Orchestrating Federated Learning in Space-Air-Ground Integrated Networks: Adaptive Data Offloading and Seamless Handover

Dong-Jun Han, *Member, IEEE,* Wenzhi Fang, Seyyedali Hosseinalipour, *Member, IEEE,* Mung Chiang, *Fellow, IEEE,* Christopher G. Brinton, *Senior Member, IEEE*

***Abstract*—Devices located in remote regions often lack coverage from well-developed terrestrial communication infrastructure. This not only prevents them from experiencing high quality communication services but also hinders the delivery of machine learning services in remote regions. In this paper, we propose a new federated learning (FL) methodology tailored to space-air-ground integrated networks (SAGINs) to tackle this issue. Our approach strategically leverages the nodes within space and air layers as both (i) edge computing units and (ii) model aggregators during the FL process, addressing the challenges that arise from the limited computation powers of ground devices and the absence of terrestrial base stations in the target region. The key idea behind our methodology is the adaptive data offloading and handover procedures that incorporate various network dynamics in SAGINs, including the mobility, heterogeneous computation powers, and inconsistent coverage times of incoming satellites. We analyze the latency of our scheme and develop an adaptive data offloading optimizer, and also characterize the theoretical convergence bound of our proposed algorithm. Experimental results confirm the advantage of our SAGIN-assisted FL methodology in terms of training time and test accuracy compared with various baselines.**

***Index Terms*—Federated learning, Space-air-ground integrated networks, LEO satellites, Data offloading and handover**

I. Introduction

As the proliferation of edge devices, including mobile phones, smart vehicles, and Internet-of-Things (IoT) sensors, continues to escalate, they generate vast quantities of data at the wireless edge. In response to this surge, federated learning (FL) [1]–[3] has emerged as a powerful method for harnessing these distributed data sources to train machine learning (ML) models. Over recent years, FL has garnered significant attention and has been rigorously explored across various configurations: from single-server environments [1], [4], hierarchical structures [5], [6], to decentralized networks [7], [8]. This body of research, spanning foundational studies to implementations, underscores the adaptability and potential of FL in optimizing data-driven insights at the network edge.

This work was supported in part by the Defense Advanced Research Projects Agency (DARPA) under Grant D22AP00168, in part by the National Science Foundation (NSF) under Grant CNS-2212565, and in part by the Office of Naval Research (ONR) under Grant N000142112472.

D.-J. Han is with the Department of Computer Science and Engineering, Yonsei University, Seoul, South Korea (email: djh@yonsei.ac.kr).

W. Fang, M. Chiang and C. G. Brinton are with the Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, USA (e-mail: {fang375, chiang, cgb}@purdue.edu).

S. Hosseinalipour is with the Department of Electrical Engineering, University at Buffalo–SUNY, NY, USA (email: alipour@buffalo.edu).

A. Motivation and Key Questions

Despite the advances in FL frameworks, they mostly rely heavily on terrestrial communication infrastructures for model aggregation during the training process. This reliance renders most existing FL methods unsuitable for areas lacking terrestrial communication facilities. Specifically, many remote regions of the Earth, such as mountains, forests, deserts, and coastal areas, do not have well-developed base stations, even though they are home to numerous ground devices, such as IoT sensors, that collect valuable data. The data gathered in these locations are essential for the development of intelligent services tailored to a variety of applications: (i) Disaster predictions in coast, mountain, and forest areas that lack a base station. To achieve this, FL over data samples collected from various types of sensor devices in these remote regions is required. (ii) Autonomous vehicle applications in rural regions. Since these regions have different traffic patterns compared to urban areas, FL needs to be conducted over data samples of vehicles in rural regions. (iii) Medical applications, which is one of the key use cases of FL. Hospitals located in different areas of the world may want to collaboratively train a global model for disease prediction. In such cases, hospitals located in rural regions can take advantage of satellites based on our approach. (iv) Smart agriculture across farms where a well-developed terrestrial base station is unavailable. In this use case, FL should be conducted using data samples collected from different farms. Decentralized FL methods [7], [8], although designed to mitigate some of these challenges, encounter significant obstacles in environments where communication links between devices are unreliable or non-existent, as often found in disaster-affected or maritime regions. Consequently, there is a need for an FL methodology that is specifically tailored to remote areas, ensuring that the distributed data collected in those regions can be leveraged to develop intelligent services.

Space-air-ground integrated networks (SAGINs) have recently emerged as a groundbreaking solution within the wireless communications community [9], [10], aimed at extending wireless coverage across the globe, particularly in isolated and remote regions. In addition to the terrestrial nodes located at the *ground layer*, SAGINs leverage satellites in the *space layer* and air nodes, such as unmanned aerial vehicles (UAVs), in the *air layer*. This multi-layered architecture enables SAGINs to either complement or entirely supplant traditional terrestrial networks in delivering communication services. Furthermore, SAGINs

are not limited to providing mere connectivity; they also have the potential to act as edge computing platforms [11]–[13]. In particular, they can undertake computation tasks offloaded from terrestrial, resource-constrained devices, such as IoT sensors. The integration of SAGINs thus promises not only to bridge the connectivity gap in underserved areas but also to enhance the computational capabilities at the network edge, opening new avenues for advanced applications and services.

Inspired by the capabilities of SAGINs, this paper sets out to explore the orchestration of FL within SAGINs to facilitate FL in remote areas. This brings forth a set of novel challenges that are absent in conventional FL implementations over terrestrial networks. Our investigation is driven by research questions aimed at unlocking the full potential of FL in the context of SAGINs. First, how should we optimally utilize the unique components of SAGINs, including satellites, air nodes, and ground devices, during the FL process? Second, how should we address the network dynamics in SAGINs (e.g., dealing with the mobility, varying computation capacities, and the inconsistent coverage times provided by satellites) during FL? Third, can we theoretically guarantee the convergence of FL despite the inherent challenges of SAGINs, such as variable network conditions and limited connectivity? Despite the importance of deploying FL in remote regions for intelligent service development, these questions have been largely overlooked in existing research. Our goal is thus to fill this gap by providing insights and solutions that enable effective FL over SAGINs.

B. Main Contributions

In this paper, we propose a FL methodology that takes advantage of both computation and communication resources of space/air/terrestrial nodes in SAGINs to provide intelligent ML services over remote areas. Compared to prior FL methods that rely on base stations, our approach strategically leverages the space and air nodes as both (i) edge computing units and (ii) ML model aggregators during the FL process, to address the challenges arising from the limited computation powers of ground devices and the absence of terrestrial base stations in the target remote region. Under this framework, we propose an adaptive approach to *optimize data offloading* depending on the network dynamics of SAGINs, including the inconsistent computation capabilities and coverage times of low-earth orbit (LEO) satellites. Considering the mobility of LEO satellites, we also propose an *optimized data/model handover strategy* where each satellite transmits the trained model and its dataset to the next incoming satellite to ensure a seamless ML model training process. By incorporating the handover delay into our latency modeling, we optimize the amount of data being offloaded across the layers in SAGINs during the FL process.

Overall, our contributions can be summarized as follows:

- **New methodology:** We introduce a new SAGIN-based FL methodology with adaptive data offloading and handover, which facilitates intelligent ML services in remote areas without the need for terrestrial communication infrastructures. Our scheme strategically utilizes the space and air nodes as edge computing units and model aggregators, and captures the key features of SAGINs including mobility of satellites, time-varying resources and coverage times of incoming satellites, hierarchical architecture, and computation resources of space/air/terrestrial nodes.
- **Analysis and optimization:** We analyze the latency of the proposed algorithm, and propose an optimized inter-layer data offloading scheme and an intra-layer data handover strategy for the space layer to minimize the delay. This optimization process takes into account the data transmission delay, data processing delay, and model aggregation delay altogether, as well as various network dynamics in SAGINs. We also analytically characterize the convergence bound of our algorithm, and show that the model converges to a stationary point for non-convex loss functions even when adaptive data offloading is applied.
- **Simulations under practical modeling:** We provide extensive experiments using three FL benchmark datasets. To simulate real-world scenarios, we adopt the Walker-Star function to model a satellite constellation and measure the coverage time of each satellite over the target region. Experimental results demonstrate that the proposed methodology can achieve the target accuracy much faster with less training latency compared to various baselines.

To the best of our knowledge, this is one of the earliest works to successfully integrate FL with adaptive data offloading/handover optimization across space-air-ground layers, while accounting for various network dynamics specific to SAGINs.

C. Related Works

FL over terrestrial networks: FL has been actively studied in terrestrial networks where the server (e.g., base station) aggregates the client models in the system. Most of them consider a single-server setup [1], [4], [14]–[17] while some researchers also study multi-server scenarios [5], [6], [18]–[20]. In [7], [8], [21], [22], the authors investigate decentralized FL where each client aggregates the models via device-to-device communications with its adjacent clients, without relying on the server. Data offloading strategies are also studied in FL where each client offloads a portion of its local dataset to the server [23]–[25]. However, prior works on FL are mostly inapplicable in remote regions, where well-developed base stations are not available and communication links between devices are unstable (e.g., disaster or maritime regions). Compared to these works, we facilitate FL in remote areas by strategically leveraging non-terrestrial network elements, specifically SAGINs.

FL with UAVs or satellites: Another line of research has explored FL over UAVs [26]–[28] or satellites [29]–[36], where either UAVs or satellites collect their own datasets and are considered as clients. After the local training procedure at UAVs or satellites, model aggregation and synchronization are conducted relying on the ground base station [26], [27], [30]–[33], [36] or directly at the UAVs/satellites [34], [35]. The problem setup of these studies differs from ours as we focus on FL over data samples collected at ground devices located in remote regions. This necessitates interaction between ground devices and nodes in the space/air layers not only for model aggregation (to address the lack of base stations in remote regions) but also for computation offloading (to tackle the limited computation capabilities of ground devices).

Some previous works [37]–[40] have focused on the setting where ground devices collect data and conduct FL assisted by the UAVs/satellites, similar to our problem setup. Specifically in [39], the satellite aggregates the models of ground devices via over-the-air aggregation, without requiring any base stations. The authors of [37] focus on solving the maze problem using the deep Q network assisted by the satellites. In [38], [41], data offloading has been studied for satellite-assisted FL by considering only the space layer. While these works do not consider SAGINs, a recent work [40] specifically studied FL considering space-air-ground layers. However, the nodes in space and air layers are only used as model aggregators, not as edge computing units. Compared to all prior works, the contribution of this work is to adaptively optimize data offloading across different layers and handover within the space layer, while taking into account the network dynamics specific to SAGINs (e.g., heterogeneous coverage time and resource availability of current/incoming satellites) during FL.

Space-air-ground integrated networks (SAGINs): Motivated by the potential for providing wide wireless coverage across the Earth, SAGINs [9] have been actively studied in the literature. Outage analysis is conducted for SAGINs in [10], while network control methodologies for SAGINs are considered in [42], [43]. In [11]–[13], [44], the authors focused on edge computing in SAGINs, where ground devices offload their computation tasks to the space and air layers. Compared to existing studies on SAGINs, the unique position of this work lies in the integration of distributed/federated ML, SAGINs, and adaptive data offloading/handover. Beyond enhancing wireless coverage, we provide additional guidelines for intelligent ML services in remote areas with the assistance of SAGINs.

The rest of the paper is organized as follows. We describe the problem setup in Section II, followed by an overview of the methodology in Section III. In Section IV, we analyze the latency of our scheme and optimize data offloading. Theoretical convergence results are provided in Section V, and numerical results are presented in Section VI. Finally, we draw conclusion and future directions in Section VII.

II. Problem Setup

We consider a SAGIN that is composed of space, air, and ground layers. We let $\mathcal{G}$ be the set that consists of K terrestrial devices located at a specific target region that lacks a base station. We denote $D_k = D_k^l \cup D_k^o$ as the local dataset of device k with $D_k^l \cap D_k^o = \emptyset$, where D_k^l is the privacy-sensitive dataset that should be kept locally at each device, while D_k^o consists of non-sensitive samples that can be offloaded to other nodes. We define $\alpha_k = |D_k^o|/|D_k|$ as the portion of non-sensitive data samples at ground device k, where $|D|$ represents the number of samples in dataset D. This problem setting covers various applications, including (i) autonomous vehicles or mobile phones that collect data with both non-sensitive classes (e.g., traffic lights, trees) and sensitive classes (e.g., humans), (ii) hospitals with data of patients who have agreed with the privacy policy and who have not agreed with it, (iii) sensor devices for disaster predictions in coastal regions that mostly collect non-sensitive samples.

In the air layer, we consider a set $\mathcal{A}$ with N air nodes (e.g., UAVs) covering the target region. Each air node n is associated with the device set $\mathcal{G}_n$, where $\mathcal{G} = \cup_{n=1}^{N} \mathcal{G}_n$ holds with $\mathcal{G}_{n_1} \cap \mathcal{G}_{n_2} = \emptyset$ if $n_1 \neq n_2$. In the space layer, we consider LEO satellites that are moving according to their own orbits. Each ground device can directly communicate with the corresponding air node in the air layer, while each air node can communicate with the satellite that is covering the target region. Fig. 1 illustrates the overview of our system model.

The goal is to train a shared global model $\mathbf{w}^*$ tailored to the datasets collected at ground devices in $\mathcal{G}$. Specifically, we aim to minimize the following objective function:

$$F(\mathbf{w}) = \sum_{k=1}^{K} \lambda_k F_k(\mathbf{w}), \tag{1}$$

where $\lambda_k = \frac{|D_k|}{\sum_{j\in\mathcal{G}} |D_j|}$ is the relative dataset size of device k. $F_k(\mathbf{w})$ is the local loss function of device k defined as $F_k(\mathbf{w}) = \frac{1}{|D_k|} \sum_{x\in D_k} \ell(x; \mathbf{w})$, where $\ell(x; \mathbf{w})$ is the loss (e.g., cross-entropy loss) obtained with data sample x and model $\mathbf{w}$.

There are several key challenges in achieving the above goal in remote areas. First, it is difficult to aggregate the trained models within such regions without a base station. Secondly, the terrestrial devices (e.g., IoT sensors) are often equipped with low computation capabilities, significantly slowing down the training process. In this work, we use space and air nodes as model aggregators to solve the first challenge, and also use them as edge computing units to process data samples offloaded from the ground layer, to tackle the second challenge.

III. Methodology Overview

In this section, we provide an overview of our methodology that achieves the aforementioned objectives in SAGINs. The proposed algorithm consists of R global rounds indexed by $r = 0, 1, \ldots, R-1$. In the following, we focus on a specific round r to describe the process of our scheme.

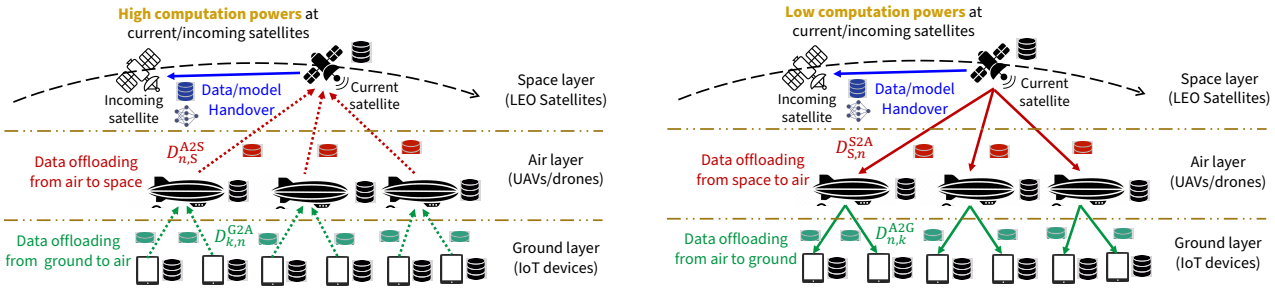
A. Adaptive Inter-Layer Data Offloading

Let $D_{\mathsf{G},k}^{(r)}$, $D_{\mathsf{A},n}^{(r)}$, and $D_{\mathsf{S}}^{(r)}$ denote the local datasets at node $k \in \mathcal{G}$ in the ground layer, node $n \in \mathcal{A}$ in the air layer, and the satellite that is currently serving the targeting region, respectively, at the beginning of round r. Note that we have

$$D_{\mathsf{G},k}^{(0)} = D_k, \ \forall k \in \mathcal{G}, \ D_{\mathsf{A},n}^{(0)} = \emptyset, \ \forall n \in \mathcal{A}, \ D_{\mathsf{S}}^{(0)} = \emptyset \tag{2}$$

for $r = 0$ since data samples are generated at the ground devices.

Depending on various system environments, inter-layer data offloading is first performed across the network to obtain the updated datasets $D_{\mathsf{G},k}^{(r+1)}$, $D_{\mathsf{A},n}^{(r+1)}$, and $D_{\mathsf{S}}^{(r+1)}$ at the nodes in each layer. Fig. 1 illustrates example scenarios of adaptive data offloading depending on the computation capabilities of the satellites. Intuitively, if the current/incoming satellites have relatively high computation powers, more data samples can be offloaded to the space layer. Otherwise, data samples should be transmitted from the space layer to other layers for load balancing. The data offloading solution is also affected by the coverage times of the satellites over the target region.



(a) High computation powers at current/incoming satellites.

(b) Low computation powers at current/incoming satellites.

Fig. 1: Overview of adaptive data offloading/handover during FL over SAGINs, depending on the current resource availability.

We describe the detailed optimization procedure for our adaptive data offloading strategy later in Section IV, as it is built upon the analysis provided in the following subsections.

B. Local Training at Ground and Air Layers

Based on the updated datasets $D_{\mathsf{G},k}^{(r+1)}$, $D_{\mathsf{A},n}^{(r+1)}$, and $D_{\mathsf{S}}^{(r+1)}$ obtained from Section III-A, the nodes in the system conduct local model updates. We first describe the local training process at the ground and air layers. At the beginning of global round r, all nodes in the system have the synchronized model represented by $\mathbf{w}^{(r)}$. Starting from the initial model $\mathbf{w}_{\mathsf{G},k}^{(r,0)} = \mathbf{w}_{\mathsf{A},n}^{(r,0)} = \mathbf{w}^{(r)}$, each ground device k and air node n updates its model for H local iterations according to

$$\mathbf{w}_{\mathsf{G},k}^{(r,h+1)} = \mathbf{w}_{\mathsf{G},k}^{(r,h)} - \eta_{\mathsf{G},k}^{(r)} \tilde{\nabla} \ell_{\mathsf{G},k}^{(r+1)}(\mathbf{w}_{\mathsf{G},k}^{(r,h)}), h = 0, \dots H-1, \quad (3)$$

$$\mathbf{w}_{\mathsf{A},n}^{(r,h+1)} = \mathbf{w}_{\mathsf{A},n}^{(r,h)} - \eta_{\mathsf{A},n}^{(r)} \tilde{\nabla} \ell_{\mathsf{A},n}^{(r+1)}(\mathbf{w}_{\mathsf{A},n}^{(r,h)}), h = 0, \dots H-1, \quad (4)$$

where $\mathbf{w}_{\mathsf{G},k}^{(r,h)}$ and $\mathbf{w}_{\mathsf{A},n}^{(r,h)}$ are the models after h local iterations at global round r, $\ell_{\mathsf{G},k}^{(r+1)}(\cdot) = \frac{1}{|D_{\mathsf{G},k}^{(r+1)}|} \sum_{x \in D_{\mathsf{G},k}^{(r+1)}} \ell(x;\cdot)$ and $\ell_{\mathsf{A},n}^{(r+1)}(\cdot) = \frac{1}{|D_{\mathsf{A},n}^{(r+1)}|} \sum_{x \in D_{\mathsf{A},n}^{(r+1)}} \ell(x;\cdot)$ are the local loss functions defined at the corresponding nodes. Also, $\tilde{\nabla} \ell_{\mathsf{G},k}^{(r+1)}(\cdot)$ and $\tilde{\nabla} \ell_{\mathsf{A},n}^{(r+1)}(\cdot)$ denote the computed mini-batch gradients, where the size of the mini-batch can be set based on the size of the local dataset. $\eta_{\mathsf{G},k}^{(r)}$ and $\eta_{\mathsf{A},n}^{(r)}$ represent the learning rates at ground device k and air node n, respectively.

The required local computation times (in seconds) at ground device k and air node n for model updates are expressed as

$$\tau_{\mathsf{G},k}^{\mathsf{local},(r)} = \frac{m_{\mathsf{G},k} |D_{\mathsf{G},k}^{(r+1)}|}{f_{\mathsf{G},k}}, \quad \tau_{\mathsf{A},n}^{\mathsf{local},(r)} = \frac{m_{\mathsf{A},n} |D_{\mathsf{A},n}^{(r+1)}|}{f_{\mathsf{A},n}}, \quad (5)$$

respectively, where $f_{\mathsf{G},k}$, $f_{\mathsf{A},n}$ are the CPU frequencies (in cycles/sec) and $m_{\mathsf{G},k}$, $m_{\mathsf{A},n}$ are the numbers of required CPU cycles to update the model with one data sample (in cycles/sample) at the corresponding nodes.

C. Satellite-Side Training and Data/Model Handover

In parallel with the local training process at the air/ground layers, the current satellite also updates the model using dataset $D_{\mathsf{S}}^{(r+1)}$. Starting from $\mathbf{w}_{\mathsf{S}}^{(r,0)} = \mathbf{w}^{(r)}$, the model update process at the satellite can be written as follows:

$$\mathbf{w}_{\mathsf{S}}^{(r,h+1)} = \mathbf{w}_{\mathsf{S}}^{(r,h)} - \eta_{\mathsf{S}}^{(r)} \tilde{\nabla} \ell_{\mathsf{S}}^{(r+1)}(\mathbf{w}_{\mathsf{S}}^{(r,h)}), h = 0, \dots H-1, \quad (6)$$

where $\tilde{\nabla} \ell_{\mathsf{S}}^{(r+1)}(\cdot)$ and $\eta_{\mathsf{S}}^{(r)}$ are the satellite-side stochastic mini-batch gradient and learning rate, respectively. The size of the mini-batch is set to $|D_{\mathsf{S}}^{(r+1)}|/H$ so that all data samples in $D_{\mathsf{S}}^{(r+1)}$ can be processed in H iterations.

Data/model handover: In satellite networks, satellites are perceived as non-stationary units, where at each snapshot of the network each LEO satellite covers a different region compared to other LEO satellites and may have its own specific task tailored to its coverage area (e.g., edge computing, FL, or communication services). In our setup, the current satellite that is covering the target area is responsible for conducting FL over that region. However, a key challenge is that each satellite has a limited coverage time over the target region due to the mobility. This motivates us to consider an intra-layer data/model handover strategy within the space layer, to ensure a seamless FL process. Specifically, the current satellite transmits the updated model and dataset to the incoming satellite before leaving the target region, so that this new satellite can continue model training in the space layer using dataset $D_{\mathsf{S}}^{(r+1)}$ during its coverage period over the target region. These local training and handover steps are repeated until all data samples in $D_{\mathsf{S}}^{(r+1)}$ are processed, based on a series of incoming satellites that will cover the target region.

The handover delay between the i-th and $(i+1)$-th satellites at global round r can be written as follows:

$$\tau_{i,i+1}^{\mathsf{hand},(r)} = \frac{Q(\mathbf{w}) + q|D_{\mathsf{S}}^{(r+1)}|}{Z_{i,i+1}^{\mathsf{ISL},(r)}}, \quad (7)$$

where $Q(\mathbf{w})$ is the model size (in bits), q is the size of each data sample (in bits) and $Z_{i,i+1}^{\mathsf{ISL}}$ is the transmission rate for inter-satellite link (ISL) communications between i-th and $(i+1)$-th satellites. Referring to [31], [45], we have $Z_{i,i+1}^{\mathsf{ISL},(r)} = B \log_2(1 + \frac{p_{\mathsf{S},i}^{(r)} A_i^{\mathrm{Tx}} A_{i+1}^{\mathrm{Rx}}}{C_{i,i+1} N_0})$, where B is the bandwidth, $p_{\mathsf{S},i}^{(r)}$ is the transmit power of the i-th satellite, A_i^{Tx} and A_{i+1}^{Rx} are the Tx and Rx gains of antenna, $C_{i,i+1}$ is the free space path loss between satellites, N_0 is the noise power density.

Latency at the space layer: Based on the above data/model handover strategy, we now characterize the training latency at the space layer. Let $f_{\mathsf{S},i}^{(r)}$ represent the CPU frequency of the i-th satellite covering the target region at global round r. We also denote $m_{\mathsf{S},i}^{(r)}$ as the number of CPU cycles required to process one data sample at the i-th satellite at round r. Moreover, let $T_i^{(r)}$ denote the delay until the i-th satellite leaves the coverage

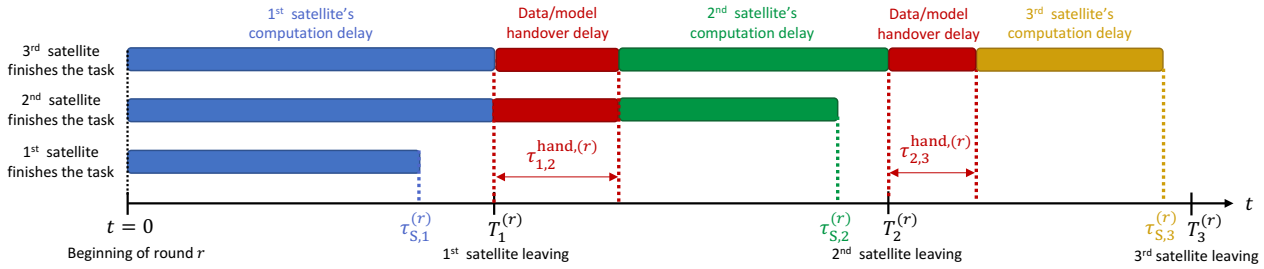


Fig. 2: Illustration of model training and intra-layer data/model handover procedures at the space layer. If the current satellite is not able to complete the task within its coverage time over the target region, the next incoming satellite continues local training after receiving the dataset and the model from the previous satellite to ensure a seamless FL process.

of the target region, measured from the moment when global round r has started. Trivially, for the satellites that do not join or leave the region in round r, $T_i^{(r)}$ becomes infinity.

To gain insights, we start with some examples illustrated in Fig. 2. Suppose that the first satellite is able to process the whole dataset $D_{\text{S}}^{(r+1)}$ within the time duration $T_1^{(r)}$. Then, the local computation delay $\tau_{\text{S},1}^{(r)}$ (in seconds) at the space layer can be written as follows:

$$\tau_{\text{S},1}^{(r)} = m_{\text{S},1}^{(r)} |D_{\text{S}}^{(r+1)}| / f_{\text{S},1}^{(r)}. \tag{8}$$

However, if $\tau_{\text{S},1}^{(r)} > T_1^{(r)}$, indicating that the first satellite is unable to complete the computation before leaving the target region, data/model handover from the first satellite to the second satellite is conducted. Note that the number of data samples that can be processed at the first satellite within time duration $T_1^{(r)}$ is $(f_{\text{S},1}^{(r)}/m_{\text{S},1}^{(r)})T_1^{(r)}$. Hence, the amount of data samples that should be processed at the satellites other than the first one becomes $|D_{\text{S}}^{(r+1)}| - (f_{\text{S},1}^{(r)}/m_{\text{S},1}^{(r)})T_1^{(r)}$. Now suppose that the second satellite can process all $|D_{\text{S}}^{(r+1)}| - (f_{\text{S},1}^{(r)}/m_{\text{S},1}^{(r)})T_1^{(r)}$ data samples before leaving the target region. Then the computation time at the second satellite to finish local training can be expressed as $m_{\text{S},2}^{(r)}(|D_{\text{S}}^{(r+1)}| - \frac{f_{\text{S},1}^{(r)}}{m_{\text{S},1}^{(r)}} T_1^{(r)})/f_{\text{S},2}^{(r)}$. This leads to the following latency result:

$$\tau_{\text{S},2}^{(r)} = T_1^{(r)} + \tau_{1,2}^{\text{hand},(r)} + \frac{m_{\text{S},2}^{(r)}(|D_{\text{S}}^{(r+1)}| - \frac{f_{\text{S},1}^{(r)}}{m_{\text{S},1}^{(r)}} T_1^{(r)})}{f_{\text{S},2}^{(r)}}. \tag{9}$$

The result in (9) incorporates the computation time at the first satellite, i.e., $T_1^{(r)}$, the handover delay, i.e., $\tau_{1,2}^{\text{hand},(r)}$, and the computation time at the second satellite, i.e., the last term.

However, if $\tau_{\text{S},2}^{(r)} > T_2^{(r)}$, the local training cannot be completed before the second satellite leaves the target area. In this case, the third satellite processes the remaining data after receiving the information from the second satellite via ISL communication. Overall, we obtain the following result:

$$\tau_{\text{S}}^{(r)} = \begin{cases} \tau_{\text{S},1}^{(r)}, & \text{if } \tau_{\text{S},1}^{(r)} < T_1^{(r)} \ (1^{\text{st}} \text{ satellite finishes the task}) \\ \tau_{\text{S},2}^{(r)}, & \text{if } \tau_{\text{S},2}^{(r)} < T_2^{(r)} \ (2^{\text{nd}} \text{ satellite finishes the task}) \\ \tau_{\text{S},3}^{(r)}, & \text{if } \tau_{\text{S},3}^{(r)} < T_3^{(r)} \ (3^{\text{rd}} \text{ satellite finishes the task}) \\ \vdots \end{cases} \tag{10}$$

where $\tau_{\text{S},1}^{(r)}$ and $\tau_{\text{S},2}^{(r)}$ are defined in (8) and (9) while $\tau_{\text{S},3}^{(r)}$ is written as follows:

$$\tau_{\text{S},3}^{(r)} = T_2^{(r)} + \tau_{2,3}^{\text{hand},(r)} + \frac{m_{\text{S},3}^{(r)}\left(|D_{\text{S}}^{(r+1)}| - \frac{f_{\text{S},1}^{(r)}}{m_{\text{S},1}^{(r)}} T_1^{(r)} - \frac{f_{\text{S},2}^{(r)}}{m_{\text{S},2}^{(r)}}(T_2^{(r)} - T_1^{(r)} - \tau_{1,2}^{\text{hand},(r)})\right)}{f_{\text{S},3}^{(r)}}. \tag{11}$$

As illustrated in Fig. 2, the term $T_2^{(r)}$ in (11) captures the delay until the second satellite leaves the target region, $\tau_{2,3}^{\text{hand},(r)}$ is the handover delay, and the last term is the delay for the third satellite to complete the remaining tasks. For an arbitrary $i \geq 2$, we can generalize the result as follows:

$$\tau_{\text{S},i}^{(r)} = T_{i-1}^{(r)} + \tau_{i-1,i}^{\text{hand},(r)} + \frac{m_{\text{S},i}^{(r)}(|D_{\text{S}}^{(r+1)}| - \Omega_i^{(r)})}{f_{\text{S},i}^{(r)}}, \tag{12}$$

where $\Omega_i^{(r)}$ is the amount of data samples processed prior to the i-th satellite at round r. Fig. 2 summarizes the idea of the repeated local training and data/model handover processes at the space layer.

D. Model Aggregation

After local updates are completed according to Sections III-B and III-C, model aggregation is conducted to obtain a new global model. Specifically, each air node n aggregates the models $\{\mathbf{w}_{\text{G},k}^{(r+1)}\}_{k\in\mathcal{G}_n}$ sent from the ground devices in its coverage and the model $\mathbf{w}_{\text{A},n}^{(r+1)}$ trained by its own. Then, each air node n sends the aggregated model to the current satellite for global aggregation. The final global model becomes

$$\mathbf{w}^{(r+1)} = \sum_{k\in\mathcal{G}} \lambda_{\text{G},k}^{(r+1)} \mathbf{w}_{\text{G},k}^{(r,H)} + \sum_{n\in\mathcal{A}} \lambda_{\text{A},n}^{(r+1)} \mathbf{w}_{\text{A},n}^{(r,H)} + \lambda_{\text{S}}^{(r+1)} \mathbf{w}_{\text{S}}^{(r,H)}, \tag{13}$$

where $\lambda_{\text{G},k}^{(r+1)} = \frac{|D_{\text{G},k}^{(r+1)}|}{\sum_{j\in\mathcal{G}}|D_j|}$, $\lambda_{\text{A},n}^{(r+1)} = \frac{|D_{\text{A},n}^{(r+1)}|}{\sum_{j\in\mathcal{G}}|D_j|}$, and $\lambda_{\text{S}}^{(r+1)} = \frac{|D_{\text{S}}^{(r+1)}|}{\sum_{j\in\mathcal{G}}|D_j|}$ are the portions of data samples at each node.

The delay for uploading the model from ground device k to air node n can be written as follows:

$$\tau_{k,n}^{\text{G2A},(r)} = \frac{Q(\mathbf{w})}{Z_{k,n}^{\text{G2A},(r)}}, \tag{14}$$

where $Z_{k,n}^{\mathsf{G2A},(r)}$ is the uplink communication rate between ground device k and air node n expressed as[1]

$$Z_{k,n}^{\mathsf{G2A},(r)} = \mathbb{E}\Big[b_{k,n}^{(r)} \log_2(1 + \frac{p_{\mathsf{G},k}|h_{k,n}^{(r)}|^2}{b_{k,n}^{(r)} N_0})\Big]. \tag{15}$$

Here, $p_{\mathsf{G},k}$ is the transmit power of ground device k, $b_{k,n}^{(r)}$ is the bandwidth, and $h_{k,n}^{(r)} = \beta_0/(d_{k,n}^{(r)})^{\gamma^{\mathsf{G2A}}} g$ is the channel between device k and air node n, which is defined with the distance $d_{k,n}^{(r)}$, pathloss exponent between ground and air γ^{G2A}, channel gain β_0 at the reference distance of 1 meter, and Rayleigh fading parameter g. Similarly, we can also define the model upload delay from air node n to the current satellite, i.e., $\tau_{n,\mathsf{S}}^{\mathsf{A2S},(r)}$, based on the communication rate $Z_{n,\mathsf{S}}^{\mathsf{A2S},(r)}$ between air node n and the current satellite covering the target region[2].

IV. Adaptive Data Offloading Optimization

In this section, we provide details for our data offloading step outlined in Section III-A. This process aims to construct $\{D_{\mathsf{G},k}^{(r+1)}\}_{k\in\mathcal{G}}$, $\{D_{\mathsf{A},n}^{(r+1)}\}_{n\in\mathcal{A}}$, and $D_{\mathsf{S}}^{(r+1)}$ from $\{D_{\mathsf{G},k}^{(r)}\}_{k\in\mathcal{G}}$, $\{D_{\mathsf{A},n}^{(r)}\}_{n\in\mathcal{A}}$, and $D_{\mathsf{S}}^{(r)}$, at the beginning of global round r.

A. Characterization of Data Transmission Direction

Latency without data offloading: The first step of our approach is to characterize the direction of data transmission. We start by deriving the latency without data offloading, to see which layer causes more delay. When data offloading is not considered, the overall delay at round r can be written as

$$\tau^{(r)} = \max\Big\{\tau_S^{(r)}, \max_{n\in\mathcal{A}}\{\tau_{\mathsf{A},n}^{(r)} + \tau_{n,\mathsf{S}}^{\mathsf{A2S},(r)}\}\Big\}, \tag{16}$$

where $\tau_S^{(r)}$ is the completion time at the space layer defined in (10) and $\tau_{n,\mathsf{S}}^{\mathsf{A2S},(r)}$ is the model transmission delay from air node n to the current satellite, similar to (14). $\tau_{\mathsf{A},n}^{(r)}$ is the delay until air node n aggregates its own updated model with the models sent from the devices in its coverage area $\mathcal{G}_n$:

$$\tau_{\mathsf{A},n}^{(r)} = \max\Big\{\tau_{\mathsf{A},n}^{\mathsf{local},(r)}, \max_{k\in\mathcal{G}_n}\{\tau_{\mathsf{G},k}^{\mathsf{local},(r)} + \tau_{k,n}^{\mathsf{G2A},(r)}\}\Big\}. \tag{17}$$

Here, $\tau_{\mathsf{A},n}^{\mathsf{local},(r)}$ and $\tau_{\mathsf{G},k}^{\mathsf{local},(r)}$ are the local computation times at air node n and ground device k, respectively, as described in (5). Here, we note that all notations in (16) and (17) are defined with the datasets before data offloading, i.e., $\{D_{\mathsf{G},k}^{(r)}\}_{k\in\mathcal{G}}$, $\{D_{\mathsf{A},n}^{(r)}\}_{n\in\mathcal{A}}$, $D_{\mathsf{S}}^{(r)}$, to characterize the data offloading direction.

Data transmission scenarios: Our adaptive data offloading method is motivated by the dynamic nature of SAGINs, including the computation capabilities as well as the coverage times of current/incoming satellites. We consider two different scenarios depending on the direction of data transmission.

[1]In scenarios where instantaneous channel is available via feedback, the latency can be written without the expectation.

[2]Following [46]–[48], Rayleigh fading can be adopted between the ground device and the air node, considering obstacles in remote areas such as forests and mountainous regions. In scenarios where the line-of-sight link is dominant, we can use the free-path space loss model by setting $h_{k,n}^{(r)} = \beta_0/(d_{k,n}^{(r)})^2$ as in [49], [50]

(i) Case I: $\tau_S^{(r)} > \max_{n\in\mathcal{A}}\{\tau_{\mathsf{A},n}^{(r)} + \tau_{n,\mathsf{S}}^{\mathsf{A2S},(r)}\}$ ***(Offloading from space to air/ground).*** Case I considers the scenario where the current and the next few incoming satellites have relatively low computation/communication capabilities. In this case, we allow data samples to be transmitted from the space layer to air/ground layers for load balancing.

(ii) Case II: $\tau_S^{(r)} < \max_{n\in\mathcal{A}}\{\tau_{\mathsf{A},n}^{(r)} + \tau_{n,\mathsf{S}}^{\mathsf{A2S},(r)}\}$ ***(Offloading from air/ground to space).*** In this case, the current/incoming satellites have relatively large computation powers. Hence, we propose data transmission from air/ground layers to the space layer for load balancing.

Objective: Our objective is to adaptively optimize data offloading across space-air-ground layers to minimize the latency. By incorporating the data offloading delay, we can rewrite the overall latency in (16) into the following form:

$$\bar{\tau}^{(r)} := \max\Big\{\bar{\tau}_S^{(r)}, \max_{n\in\mathcal{A}}\{\bar{\tau}_{\mathsf{A},n}^{(r)} + \tau_{n,\mathsf{S}}^{\mathsf{A2S},(r)}\}\Big\}. \tag{18}$$

In (18), $\bar{\tau}_S^{(r)}$ is the new delay at the space layer and

$$\bar{\tau}_{\mathsf{A},n}^{(r)} := \max\Big\{\bar{\tau}_{\mathsf{A},n}^{\mathsf{local},(r)}, \max_{k\in\mathcal{G}_n}\{\bar{\tau}_{\mathsf{G},k}^{\mathsf{local},(r)} + \tau_{k,n}^{\mathsf{G2A},(r)}\}\Big\} \tag{19}$$

is the new completion time at air node n until all the models in its coverage are aggregated, considering data offloading. $\bar{\tau}_{\mathsf{A},n}^{\mathsf{local},(r)}$ and $\bar{\tau}_{\mathsf{G},k}^{\mathsf{local}}$ are the updated delays to finish local training at air node n and ground device k, respectively, under this data offloading framework.

In the following, we will characterize the new delays $\bar{\tau}_S^{(r)}$, $\bar{\tau}_{\mathsf{A},n}^{\mathsf{local},(r)}$, and $\bar{\tau}_{\mathsf{G},k}^{\mathsf{local}}$ in (18) and (19) by considering data offloading. Then, we will optimize the amount of data being offloaded across the layers to minimize $\bar{\tau}^{(r)}$.

B. Case I: Data Offloading from Space to Air/Ground

We first consider Case I. Let $D_{\mathsf{S},n}^{\mathsf{S2A},(r)}$ be the dataset sent from the space layer to air node n in the air layer.

Dataset and latency characterization at the space layer: Then the updated dataset $D_{\mathsf{S}}^{(r+1)}$ at the space layer after data offloading satisfies the following criterion:

$$|D_{\mathsf{S}}^{(r+1)}| = |D_{\mathsf{S}}^{(r)}| - \sum_{n\in\mathcal{A}} |D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|. \tag{20}$$

Accordingly, we can obtain the updated satellite-side delay $\bar{\tau}_{\mathsf{S}}^{(r)}$ by inserting $|D_{\mathsf{S}}^{(r+1)}| = |D_{\mathsf{S}}^{(r)}| - \sum_{n\in\mathcal{A}} |D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|$ to (10). In (20), $\{|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|\}_{n\in\mathcal{A}}$ is the set of parameters that we would like to optimize. We also aim to optimize the load balancing between air and ground layers. To achieve this, we will first study the load balancing between air node n and the associated ground devices in $\mathcal{G}_n$ when $|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|$ is given. After that, we focus on the load balancing between the space and air layers.

We first characterize the direction of data transmission between the air and ground. If (i) $|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|$ is provided from the space layer to air node n, and (ii) data offloading between air and ground layers is not performed, the local computation delay at air node n can be rewritten as follows:

$$\tau_{\mathsf{A},n}^{\mathsf{local},(r)} = \max\{\frac{m_{\mathsf{A},n}|D_{\mathsf{A},n}^{(r)}|}{f_{\mathsf{A},n}}, \frac{q|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|}{Z_{\mathsf{S},n}^{\mathsf{S2A},(r)}}\} + \frac{m_{\mathsf{A},n}|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|}{f_{\mathsf{A},n}}. \tag{21}$$

The result in (21) can be interpreted as follows. At the beginning of round r, the current satellite transmits dataset $D_{\mathsf{S},n}^{\mathsf{S2A},(r)}$ to air node n. This incurs delay of $\frac{q|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|}{Z_{\mathsf{S},n}^{\mathsf{S2A},(r)}}$, where $Z_{\mathsf{S},n}^{\mathsf{S2A},(r)}$ is the downlink communication rate between the current satellite and air node n. In parallel, air node n conducts local update based on the dataset $D_{\mathsf{A},n}^{(r)}$, causing delay of $\frac{m_{\mathsf{A},n}|D_{\mathsf{A},n}^{(r)}|}{f_{\mathsf{A},n}}$. When both processes are completed, air node n can update the model using dataset $D_{\mathsf{S},n}^{\mathsf{S2A},(r)}$ received from the satellite, which is captured in the last term of (21).

Now if $\tau_{\mathsf{A},n}^{\mathsf{local},(r)} > \max_{k\in\mathcal{G}_n}\{\tau_{\mathsf{G},k}^{\mathsf{local},(r)} + \tau_{k,n}^{\mathsf{G2A},(r)}\}$, i.e., if the computation time at air node n is larger than the completion time at the ground layer in its associated region, we let air node n transmit data samples to the ground layer for load balancing. Otherwise, i.e., if $\tau_{\mathsf{A},n}^{\mathsf{local},(r)} < \max_{k\in\mathcal{G}_n}\{\tau_{\mathsf{G},k}^{\mathsf{local},(r)} + \tau_{k,n}^{\mathsf{G2A},(r)}\}$, we let air node n receive data samples from the corresponding ground devices for load balancing. In the following, we describe our method assuming $\tau_{\mathsf{A},n}^{\mathsf{local},(r)} > \max_{k\in\mathcal{G}_n}\{\tau_{\mathsf{G},k}^{\mathsf{local},(r)} + \tau_{k,n}^{\mathsf{G2A},(r)}\}$, where the result for the second case can be obtained in a similar way.

Dataset and latency characterization at air/ground layers: We define $D_{n,k}^{\mathsf{A2G},(r)}$ as the dataset that is sent from air node n to ground device $k \in \mathcal{G}_n$ at global round r. Then, the following holds for the updated dataset $D_{\mathsf{A},n}^{(r+1)}$ at air node n:

$$|D_{\mathsf{A},n}^{(r+1)}| = |D_{\mathsf{A},n}^{(r)}| + |D_{\mathsf{S},n}^{\mathsf{S2A},(r)}| - \sum_{k\in\mathcal{G}_n} |D_{n,k}^{\mathsf{A2G},(r)}|, \quad (22)$$

which is obtained after receiving $|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|$ samples from the satellite and sending $\sum_{k\in\mathcal{G}_n}|D_{n,k}^{\mathsf{A2G},(r)}|$ samples to the ground devices in $\mathcal{G}_n$. For each ground device $k \in \mathcal{G}_n$, we can write

$$|D_{\mathsf{G},k}^{(r+1)}| = |D_{\mathsf{G},k}^{(r)}| + |D_{n,k}^{\mathsf{A2G},(r)}|, \quad (23)$$

after receiving data from the corresponding air node n.

From the above definitions on the updated datasets at each layer, we can write $\bar{\tau}_{\mathsf{A},n}^{\mathsf{local},(r)}$ in (19), which represents the delay for air node n to finish computation, as follows:

$$\bar{\tau}_{\mathsf{A},n}^{\mathsf{local},(r)} = \begin{cases} \dfrac{m_{\mathsf{A},n}|D_{\mathsf{A},n}^{(r+1)}|}{f_{\mathsf{A},n}}, & \text{if } |D_{\mathsf{A},n}^{(r+1)}| \le |D_{\mathsf{A},n}^{(r)}| \\ \max\left\{\dfrac{m_{\mathsf{A},n}|D_{\mathsf{A},n}^{(r)}|}{f_{\mathsf{A},n}}, \dfrac{q|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|}{Z_{\mathsf{S},n}^{\mathsf{S2A},(r)}}\right\} & \\ \quad + \dfrac{m_{\mathsf{A},n}(|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}| - \sum_{k\in\mathcal{G}_n}|D_{n,k}^{\mathsf{A2G},(r)}|)}{f_{\mathsf{A},n}}, & \text{otherwise} \end{cases} \quad (24)$$

In (24), if $|D_{\mathsf{A},n}^{(r+1)}| \le |D_{\mathsf{A},n}^{(r)}|$, air node n can finish computation without waiting for dataset $D_{\mathsf{S},n}^{\mathsf{S2A},(r)}$ from the satellite. On the other hand, if $|D_{\mathsf{A},n}^{(r+1)}| > |D_{\mathsf{A},n}^{(r)}|$, it indicates that air node n also needs to process data samples received from the satellite. For both cases, air node n transmits $|D_{n,k}^{\mathsf{A2G},(r)}|$ data samples to ground device k after receiving data from the satellite.

Hence, for ground device k, we can write the completion time in (19) as follows:

$$\bar{\tau}_{\mathsf{G},k}^{\mathsf{local},(r)} = \max\Bigg\{ \underbrace{\frac{m_{\mathsf{G},k}|D_{\mathsf{G},k}^{(r)}|}{f_{\mathsf{G},k}}}_{\text{Comp. with original data}}, \underbrace{\frac{q|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|}{Z_{\mathsf{S},n}^{\mathsf{S2A},(r)}} + \frac{q|D_{n,k}^{\mathsf{A2G},(r)}|}{Z_{n,k}^{\mathsf{A2G},(r)}}}_{\text{Comm. for receiving data samples}} \Bigg\} + \underbrace{\frac{m_{\mathsf{G},k}|D_{n,k}^{\mathsf{A2G},(r)}|}{f_{\mathsf{G},k}}}_{\text{Comp. with received data from air node } n}. \quad (25)$$

Algorithm 1 Load Balancing Between Air Node n and the Ground Devices in $\mathcal{G}_n$

Input: $\nu_{L,1} = \nu_{L,2} = 0$, an appropriate ν_{U_1}, ν_{U_2}, and small ϵ_1, ϵ_2. Initialized $|D_{n,k}^{\mathsf{A2G},(r)}| = 0$ for all $k \in \mathcal{G}_n$. Fixed $|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|$.
Output: Optimal data allocation $\{|D_{n,k}^{\mathsf{A2G},(r)}|\}_{k\in\mathcal{G}_n}$ between air node n and ground devices in $\mathcal{G}_n$.
while $\nu_{U,1} - \nu_{L,1} \ge \epsilon_1$ **do**
  Set $Y_n = (\nu_{U,1} + \nu_{L,1})/2$
  Obtain $\{|D_{n,k}^{\mathsf{A2G},(r)}|\}_{k\in\mathcal{G}_n}$ based on the following while loop:
  Set an appropriate ν_{L_2} and ν_{U_2}.
  while $\sum_{k\in\mathcal{G}_n}|D_{n,k}^{\mathsf{A2G},(r)}| < (1-\epsilon_2)Y_n$ or $\sum_{k\in\mathcal{G}_n}|D_{n,k}^{\mathsf{A2G},(r)}| > (1+\epsilon_2)Y_n$ **do**
    for each $k \in \mathcal{G}_n$ **do**
      Compute $|D_{n,k}^{\mathsf{A2G},(r)}|$ to make $\bar{\tau}_{\mathsf{G},k}^{\mathsf{local},(r)} + \tau_{k,n}^{\mathsf{G2A},(r)}$ in (25) and $\frac{1}{2}(\nu_{U,2} + \nu_{L,2})$ as close as possible within range $|D_{n,k}^{\mathsf{A2G},(r)}| \in [0, \min\{|D_{\mathsf{A},n}^{(r)}|, Y_n\}]$ using bisection search.
    end for
    if $\sum_{k\in\mathcal{G}_n}|D_{n,k}^{\mathsf{A2G},(r)}| \le (1-\epsilon_2)Y_n$ **then**
      $\nu_{L,2} \leftarrow \frac{1}{2}(\nu_{U,2} + \nu_{L,2})$
    else
      $\nu_{U,2} \leftarrow \frac{1}{2}(\nu_{U,2} + \nu_{L,2})$
    end if
  end while
  Compute $\max_{k\in\mathcal{G}_n}\{\bar{\tau}_{\mathsf{G},k}^{\mathsf{local},(r)} + \tau_{k,n}^{\mathsf{G2A},(r)}\}$ based on (25) and the obtained $\{|D_{n,k}^{\mathsf{A2G},(r)}|\}_{k\in\mathcal{G}_n}$
  Compute $\bar{\tau}_{\mathsf{A},n}^{\mathsf{local},(r)}$ according to (24)
  if $\bar{\tau}_{\mathsf{A},n}^{\mathsf{local},(r)} \ge \max_{k\in\mathcal{G}_n}\{\bar{\tau}_{\mathsf{G},k}^{\mathsf{local},(r)} + \tau_{k,n}^{\mathsf{G2A},(r)}\}$, set $\nu_{L,1} = Y_n$.
  else set $\nu_{U,1} = Y_n$.
end while

Specifically, each ground device starts computation with its original data when round r begins, and in parallel, waits until data samples from air node n arrives. Then, each device finishes computation using data samples received from air node n.

Load balancing between air/ground layers: For load balancing between air and ground layers, we first optimize $\{|D_{n,k}^{\mathsf{A2G},(r)}|\}_{k\in\mathcal{G}_n}$ that minimizes $\bar{\tau}_{\mathsf{A},n}^{(r)}$ in (19), by solving

$$\min_{\{|D_{n,k}^{\mathsf{A2G},(r)}|\}_{k\in\mathcal{G}_n}} \max\left\{\bar{\tau}_{\mathsf{A},n}^{\mathsf{local},(r)}, \max_{k\in\mathcal{G}_n}\{\bar{\tau}_{\mathsf{G},k}^{\mathsf{local},(r)} + \tau_{k,n}^{\mathsf{G2A},(r)}\}\right\} \quad (26)$$

when $|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|$ is given. Note that the completion time at the ground layer, i.e., $\max_{k\in\mathcal{G}_n}\{\bar{\tau}_{\mathsf{G},k}^{\mathsf{local},(r)} + \tau_{k,n}^{\mathsf{G2A},(r)}\}$, is an increasing function of $|D_{n,k}^{\mathsf{A2G},(r)}|$ while the computation delay at the air layer, i.e., $\bar{\tau}_{\mathsf{A},n}^{\mathsf{local}}$, is a decreasing function of $|D_{n,k}^{\mathsf{A2G},(r)}|$. Hence, as described in Algorithm 1, we can use bisection search to make $\bar{\tau}_{\mathsf{A},n}^{\mathsf{local},(r)}$ and $\max_{k\in\mathcal{G}_n}\{\bar{\tau}_{\mathsf{G},k}^{\mathsf{local},(r)} + \tau_{k,n}^{\mathsf{G2A},(r)}\}$ as

Algorithm 2 Load Balancing Across Space-Air-Ground Layers

Input: $\nu_{L,1}=\nu_{L,2}=0$, an appropriate ν_{U_1}, ν_{U_2}, and small ϵ_1, ϵ_2. Initialized $|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|=0$ for all $n\in\mathcal{A}$.
Output: Optimal data allocations $\{|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|\}_{n\in\mathcal{A}}$ and $\{|D_{n,k}^{\mathsf{A2G},(r)}|\}_{k\in\mathcal{G}_n}$ for all $n\in\mathcal{A}$.
while $\nu_{U,1}-\nu_{L,1}\geq\epsilon_1$ **do**
  Set $X=(\nu_{U,1}+\nu_{L,1})/2$
  Obtain $\{|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|\}_{n\in\mathcal{A}}$ based on the following while loop:
  while $\sum_{n\in\mathcal{A}}|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|<(1-\epsilon_2)X$ or $\sum_{n\in\mathcal{A}}|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|>(1+\epsilon_2)X$ **do**
    for each $n\in\mathcal{A}$ **do**
      Compute $|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|$ to make $\bar{\tau}_{\mathsf{A},n}^{(r)}+\tau_{n,\mathsf{S}}^{\mathsf{A2S},(r)}$ and $\frac{1}{2}(\nu_{U,2}+\nu_{L,2})$ as close as possible within range $|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|\in[0,\min\{|D_{\mathsf{S}}^{(r)}|,X\}]$, using bisection search and $\{|D_{n,k}^{\mathsf{A2G},(r)}|\}_{k\in\mathcal{G}_n}$ obtained from **Algorithm 1**.
    end for
    if $\sum_{n\in\mathcal{A}}|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|\leq(1-\epsilon_2)X$ **then**
      $\nu_{L,2}\leftarrow\frac{1}{2}(\nu_{U,2}+\nu_{L,2})$
    else
      $\nu_{U,2}\leftarrow\frac{1}{2}(\nu_{U,2}+\nu_{L,2})$
    end if
  end while
  Compute $\bar{\tau}_{\mathsf{A},n}^{(r)}$ in (19) based on the obtained $\{|D_{n,k}^{\mathsf{A2G},(r)}|\}_{k\in\mathcal{G}_n}$ for all $n\in\mathcal{A}$ and $\{|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|\}_{n\in\mathcal{A}}$.
  Compute $\bar{\tau}_{\mathsf{S}}^{(r)}$ according to (10) and $|D_{\mathsf{S}}^{(r+1)}|$ in (20)
  if $\bar{\tau}_{\mathsf{S}}^{(r)}\geq\max_{n\in\mathcal{A}}\{\bar{\tau}_{\mathsf{A},n}^{(r)}+\tau_{n,\mathsf{S}}^{\mathsf{A2S},(r)}\}$, set $\nu_{L,1}=X$.
  else set $\nu_{U,1}=X$.
end while

close as possible, by controlling our optimization parameters $\{|D_{n,k}^{\mathsf{A2G},(r)}|\}_{k\in\mathcal{G}_n}$. In Algorithm 1, we first solve

$$\min_{\{|D_{n,k}^{\mathsf{A2G},(r)}|\}_{k\in\mathcal{G}_n}}\max\left\{\bar{\tau}_{\mathsf{A},n}^{\mathsf{local},(r)},\max_{k\in\mathcal{G}_n}\{\bar{\tau}_{\mathsf{G},k}^{\mathsf{local},(r)}+\tau_{k,n}^{\mathsf{G2A},(r)}\}\right\}$$
$$\text{subject to: }\sum_{k\in\mathcal{G}_n}|D_{n,k}^{\mathsf{A2G},(r)}|=Y_n \tag{27}$$

for a given Y_n, and then optimize Y_n to minimize $\bar{\tau}_{\mathsf{A},n}^{(r)}$ in (19), by implementing bisection search in a hierarchical way.

Load balancing across space-air-ground layers: Now we revisit our final goal, which is to jointly optimize $\{|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|\}_{n\in\mathcal{A}}$ and $\{|D_{n,k}^{\mathsf{A2G},(r)}|\}_{k\in\mathcal{G}_n}$ for all $n\in\mathcal{A}$, to minimize the overall latency $\bar{\tau}^{(r)}$ in (18) based on the obtained $\bar{\tau}_S^{(r)}$, $\bar{\tau}_{\mathsf{A},n}^{\mathsf{local},(r)}$, and $\bar{\tau}_{\mathsf{G},k}^{\mathsf{local}}$. The overall optimization procedure is summarized in Algorithm 2. Specifically, we solve

$$\min_{\{|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|\}_{n\in\mathcal{A}},}\max\left\{\bar{\tau}_S^{(r)},\max_{n\in\mathcal{A}}\{\bar{\tau}_{\mathsf{A},n}^{(r)}+\tau_{n,\mathsf{S}}^{\mathsf{A2S},(r)}\}\right\}, \tag{28}$$

for load balancing between space and air layers. During optimization, Algorithm 1 is adopted to obtain $\{|D_{n,k}^{\mathsf{A2G},(r)}|\}_{k\in\mathcal{G}_n}$ for load balancing between air and ground layers and to compute $\bar{\tau}_{\mathsf{A},n}^{(r)}$, for a given $|D_{\mathsf{S},n}^{\mathsf{S2A},(r)}|$. Overall, we make $\bar{\tau}_S^{(r)}$ and $\max_{n\in\mathcal{A}}\{\bar{\tau}_{\mathsf{A},n}^{(r)}+\tau_{n,\mathsf{S}}^{\mathsf{A2S},(r)}\}$ as close as possible by applying bisection search in a hierarchical way.

Remark 1. *In practice, Algorithm 1 and Algorithm 2 can be implemented at the nearest gateway to obtain optimized data offloading solutions. The solutions are subsequently sent to the corresponding nodes to execute the data offloading process.*

C. Case II: Data Offloading From Air/Ground to Space

Now we consider Case II, where data samples are transmitted from air/ground to space. Let $|D_{\mathsf{n,S}}^{\mathsf{A2S},(r)}|$ be the number of data samples sent from the air node n to the current satellite.

Dataset and latency characterization at the space layer: The satellite-side dataset after data offloading satisfies:

$$|D_{\mathsf{S}}^{(r+1)}|=|D_{\mathsf{S}}^{(r)}|+\sum_{n\in\mathcal{A}}|D_{n,\mathsf{S}}^{\mathsf{A2S},(r)}|. \tag{29}$$

The satellite-side delay $\bar{\tau}_{\mathsf{S}}^{(r)}$ can be updated accordingly based on $|D_{\mathsf{S}}^{(r+1)}|=|D_{\mathsf{S}}^{(r)}|+\sum_{n\in\mathcal{A}}|D_{n,\mathsf{S}}^{\mathsf{A2S},(r)}|$ and (10).

As in Case I, we start by characterizing the data transmission direction between air and ground layers. Without any data transmission between air and ground layers, the completion time at air node n can be written as follows:

$$\tau_{\mathsf{A},n}^{\mathsf{local},(r)}=\max\left\{\frac{m_{\mathsf{A},n}(|D_{\mathsf{A},n}^{(r)}|-|D_{n,\mathsf{S}}^{\mathsf{A2S},(r)}|)}{f_{\mathsf{A},n}},\frac{q|D_{n,\mathsf{S}}^{\mathsf{A2S},(r)}|}{Z_{n,\mathsf{S}}^{\mathsf{A2S},(r)}}\right\}, \tag{30}$$

when $|D_{n,\mathsf{S}}^{\mathsf{A2S}(r)}|$ is given. Different from Case I, in (30), both the computation time and the data offloading delay contribute to $\tau_{\mathsf{A},n}^{\mathsf{local},(r)}$. This is because the air node can upload the model to the satellite only when all data samples in $|D_{n,\mathsf{S}}^{\mathsf{A2S},(r)}|$ are transmitted to the satellite.

Now we consider the following two cases, depending on whether air node n should transmit data to the ground layer or collect data from the ground layer. If $\tau_{\mathsf{A},n}^{\mathsf{local},(r)}<\max_{k\in\mathcal{G}_n}\{\tau_{\mathsf{G},k}^{\mathsf{local},(r)}+\tau_{k,n}^{\mathsf{G2A},(r)}\}$, we let devices in $\mathcal{G}_n$ offload data to the associated air node n for load balancing. Otherwise, we let air node n transmit data samples to the corresponding ground devices. We consider the first case for description. The result for the second case can be obtained in a similar way.

Dataset and latency characterization at air/ground layers: Let $D_{k,n}^{\mathsf{G2A},(r)}$ be the dataset that is sent from ground device $k\in\mathcal{G}_n$ to air node n. Then, we have

$$|D_{\mathsf{A},n}^{(r+1)}|=|D_{\mathsf{A},n}^{(r)}|-|D_{n,\mathsf{S}}^{\mathsf{A2S},(r)}|+\sum_{k\in\mathcal{G}_n}|D_{k,n}^{\mathsf{G2A},(r)}| \tag{31}$$

at each air node n, after transmitting $|D_{n,\mathsf{S}}^{\mathsf{A2S},(r)}|$ samples to the satellite and receiving $\sum_{k\in\mathcal{G}_n}|D_{k,n}^{\mathsf{G2A},(r)}|$ samples from ground devices in cluster n. For each ground device $k\in\mathcal{G}_n$, we obtain

$$|D_{\mathsf{G},k}^{(r+1)}|=|D_{\mathsf{G},k}^{(r)}|-|D_{k,n}^{\mathsf{G2A},(r)}| \tag{32}$$

after transmitting data to the associated air node.

From these definitions, we obtain the following result:

$$\bar{\tau}_{\mathsf{A},n}^{\mathsf{local},(r)}= \tag{33}$$
$$\begin{cases}\max\left\{\frac{m_{\mathsf{A},n}|D_{\mathsf{A},n}^{(r+1)}|}{f_{\mathsf{A},n}},\frac{q|D_{n,\mathcal{S}}^{\mathsf{A2S},(r)}|}{Z_{n,\mathcal{S}}^{\mathsf{A2S},(r)}}\right\}, & \text{if } |D_{\mathsf{A},n}^{(r+1)}|\leq|D_{\mathsf{A},n}^{(r)}|\\ \max\left\{\max\left\{\frac{m_{\mathsf{A},n}|D_{\mathsf{A},n}^{(r)}|}{f_{\mathsf{A},n}},\max_{k\in\mathcal{G}_n}\{\frac{q|D_{k,n}^{\mathsf{G2A},(r)}|}{Z_{k,n}^{\mathsf{G2A},(r)}}\}\right\}\right. & \\ \left.+\frac{m_{\mathsf{A},n}(\sum_{k\in\mathcal{G}_n}|D_{k,n}^{\mathsf{G2A},(r)}|-|D_{n,\mathsf{S}}^{\mathsf{A2S},(r)}|)}{f_{\mathsf{A},n}},\frac{q|D_{n,\mathcal{S}}^{\mathsf{A2S},(r)}|}{Z_{n,\mathcal{S}}^{\mathsf{A2S},(r)}}\right\}, & \\ \text{otherwise.}\end{cases}$$

We note that air node n is ready to transmit the model to the satellite when data offloading to satellite is also completed. This is captured in the latency result above.

At each ground device k, we can write

$$\bar{\tau}_{\mathsf{G},k}^{\mathsf{local},(r)} = \max\left\{\frac{m_{\mathsf{G},k}(|D_{\mathsf{G},k}^{(r)}| - |D_{k,n}^{\mathsf{G2A},(r)}|)}{f_{\mathsf{G},k}}, \frac{q|D_{k,n}^{\mathsf{G2A},(r)}|}{Z_{k,n}^{\mathsf{G2A},(r)}}\right\}, \tag{34}$$

In (34), we take the maximum of local computation time and data offloading delay. Again, this is because the ground device can start uploading the updated model only if both the local computation and data offloading processes are completed.

Load balancing between air/ground layers: For load balancing between air and ground layers, our goal is to optimize $\{|D_{k,n}^{\mathsf{G2A},(r)}|\}_{k\in\mathcal{G}_n}$. It can be seen that the completion time at the ground layer, i.e., $\max_{k\in\mathcal{G}_n}\{\bar{\tau}_{\mathsf{G},k}^{\mathsf{local},(r)} + \tau_{k,n}^{\mathsf{G2A},(r)}\}$, is a decreasing function of $|D_{k,n}^{\mathsf{G2A},(r)}|$ if $|D_{k,n}^{\mathsf{G2A},(r)}| \leq \frac{m_{\mathsf{G},k}Z_{k,n}^{\mathsf{G2A},(r)}|D_{\mathsf{G},k}^{(r)}|}{m_{\mathsf{G},k}Z_{k,n}^{\mathsf{G2A},(r)}+qf_{\mathsf{G},k}}$, and an increasing function of $|D_{k,n}^{\mathsf{G2A},(r)}|$ otherwise. Also, the delay $\bar{\tau}_{\mathsf{A},n}^{\mathsf{local},(r)}$ at the air layer is an increasing function of $|D_{k,n}^{\mathsf{G2A},(r)}|$. Hence, similar to Algorithm 1, we can find $\{|D_{k,n}^{\mathsf{G2A},(r)}|\}_{k\in\mathcal{G}_n}$ by using bisection search in a hierarchical way to make $\bar{\tau}_{\mathsf{A},n}^{(r)}$ and $\max_{k\in\mathcal{G}_n}\{\bar{\tau}_{\mathsf{G},k}^{\mathsf{local},(r)} + \tau_{k,n}^{\mathsf{G2A},(r)}\}$ as close as possible within the range following range:

$$|D_{k,n}^{\mathsf{G2A},(r)}| \in \left[0, \min\left\{\frac{m_{\mathsf{G},k}Z_{k,n}^{\mathsf{G2A},(r)}|D_{\mathsf{G},k}^{(r)}|}{m_{\mathsf{G},k}Z_{k,n}^{\mathsf{G2A},(r)} + qf_{\mathsf{G},k}}, |D_{\mathsf{G},k}^{(r)}| - |D_k^l|\right\}\right]. \tag{35}$$

Recall that $|D_k^l|$ is the number of privacy-sensitive samples at ground device k. Hence, $|D_{\mathsf{G},k}^{(r)}| - |D_k^l|$ represents the amount of non-sensitive data of ground device k at round r, which captures the feasible number of samples for offloading.

Load balancing across space-air-ground layers: Finally, we optimize $\{|D_{n,\mathsf{S}}^{\mathsf{A2S},(r)}|\}_{n\in\mathcal{A}}$ and $\{|D_{k,n}^{\mathsf{G2A},(r)}|\}_{k\in\mathcal{G}_n}$ for all $n \in \mathcal{A}$, to minimize the overall latency $\bar{\tau}^{(r)}$ in (18) based on the obtained $\bar{\tau}_S^{(r)}$, $\bar{\tau}_{\mathsf{A},n}^{\mathsf{local},(r)}$, and $\bar{\tau}_{\mathsf{G},k}^{\mathsf{local}}$. Similar to Algorithm 2 for Case I, we can make $\bar{\tau}_S^{(r)}$ and $\max_{n\in\mathcal{A}}\{\bar{\tau}_{\mathsf{A},n}^{(r)} + \tau_{n,\mathsf{S}}^{\mathsf{A2S},(r)}\}$ as close as possible by applying bisection search, where the solution for load balancing between air and ground layers is adopted during this process.

D. Complexity Analysis

Algorithm 1 involves load balancing between an air node and the associated ground devices, utilizing nested loops and bisection searches. The primary loop, governed by the variables $\nu_{L,1}$ and $\nu_{L,2}$, iterates using a bisection method until a specified precision ϵ_1 is achieved, contributing a complexity of $\mathcal{O}(\log(\frac{1}{\epsilon_1}))$ [51], [52]. Within this loop, an inner loop also utilizes bisection search to meet a precision ϵ_2, adding a complexity of $\mathcal{O}(\log(\frac{1}{\epsilon_2}))$. The for-loop iterates over n ground devices, with each iteration involving a bisection search that contributes $\mathcal{O}\left(\log\left(\min\{|D_{\mathsf{A},n}^{(r)}|, Y_n\}\right)\right)$ complexity [53], [54]. Summing these, the overall time complexity of Algorithm 1 can be written as $\mathcal{O}\left(\log(\frac{1}{\epsilon_1}) \times \log(\frac{1}{\epsilon_2}) \times |\mathcal{G}_n| \times \log\left(\min\{|D_{\mathsf{A},n}^{(r)}|, Y_n\}\right)\right)$, reflecting the combined logarithmic and linear components of the nested operations. Similarly, the complexity of Algorithm 2 becomes $\mathcal{O}\left(\log(\frac{1}{\epsilon_1}) \times \log(\frac{1}{\epsilon_2}) \times |\mathcal{A}| \times \log\left(\min\{|D_{\mathsf{S}}^{(r)}|, X\}\right)\right)$.

V. Convergence Analysis

In this section, we investigate the convergence property of the proposed algorithm. After data offloading is performed in the r-th training round, the global loss function defined in (1) can be rewritten in the following form:

$$F(\mathbf{w}) = \sum_{k\in\mathcal{G}} \lambda_{\mathsf{G},k}^{(r)} \ell_{\mathsf{G},k}^{(r+1)}(\mathbf{w}) + \sum_{n\in\mathcal{A}} \lambda_{\mathsf{A},n}^{(r)} \ell_{\mathsf{A},n}^{(r+1)}(\mathbf{w}) + \lambda_{\mathsf{S}}^{(r)} \ell_{\mathsf{S}}^{(r+1)}(\mathbf{w}). \tag{36}$$

We note that the global loss function $F(\mathbf{w})$ is time-invariant because the global dataset does not change; rather only the data samples are exchanged among the nodes. On the other hand, the local losses, i.e., $\ell_{\mathsf{G},k}^{(r+1)}(\mathbf{w})$, $\ell_{\mathsf{A},n}^{(r+1)}(\mathbf{w})$, and $\ell_{\mathsf{S}}^{(r+1)}(\mathbf{w})$, vary over time. Our goal is to analyze the evolution of $\|\nabla F(\mathbf{w}^{(r)})\|$ to characterize the convergence behavior for non-convex loss functions. We rely on the following assumptions.

Assumption 1. *$\ell_{\mathsf{G},k}^{(r+1)}(\mathbf{w})$, $\ell_{\mathsf{A},n}^{(r+1)}(\mathbf{w})$ and $\ell_{\mathsf{S}}^{(r+1)}(\mathbf{w})$, are L-smooth for any $k \in \mathcal{G}$, $n \in \mathcal{A}$, and for any r.*

Assumption 2. *The mini-batch gradients $\tilde{\nabla}\ell_{\mathsf{G},k}^{(r+1)}(\mathbf{w})$, $\tilde{\nabla}\ell_{\mathsf{A},n}^{(r+1)}(\mathbf{w})$, and $\tilde{\nabla}\ell_{\mathsf{S}}^{(r+1)}(\mathbf{w})$ are unbiased estimates of $\nabla\ell_{\mathsf{G},k}^{(r+1)}(\mathbf{w})$, $\nabla\ell_{\mathsf{A},n}^{(r+1)}(\mathbf{w})$, and $\nabla\ell_{\mathsf{S}}^{(r+1)}(\mathbf{w})$, respectively. The variance is bounded as $\mathbb{E}\|\tilde{\nabla}\ell_{\mathsf{G},k}^{(r+1)}(\mathbf{w}) - \nabla\ell_{\mathsf{G},k}^{(r+1)}(\mathbf{w})\|^2 \leq \sigma_g^2$, $\forall k \in \mathcal{G}$, which also holds for $\tilde{\nabla}\ell_{\mathsf{A},n}^{(r+1)}(\mathbf{w})$ and $\tilde{\nabla}\ell_{\mathsf{S}}^{(r+1)}(\mathbf{w})$.*

Assumption 3. *The gradient dissimilarity between each local loss function and the global loss function $F(\mathbf{w})$ is bounded as $\mathbb{E}\left\|\nabla\ell_{\mathsf{G},k}^{(r+1)}(\mathbf{w}) - F(\mathbf{w})\right\|^2 \leq c_r\|F(\mathbf{w})\|^2 + \delta_r^2$, $\forall k \in \mathcal{G}$. This holds for $\ell_{\mathsf{A},n}^{(r+1)}(\mathbf{w})$, $\forall n \in \mathcal{A}$ and $\ell_{\mathsf{S}}^{(r+1)}(\mathbf{w})$ as well.*

Assumptions 1–3 are standard and have been widely adopted in the analyses of existing works [21], [24], [25], where Assumption 3 specifically quantifies the data heterogeneity in each round r. We present our main theorem below.

Theorem 1. *Suppose that Assumptions 1–3 hold and the learning rates satisfies*

$$\eta_{\mathsf{G},k}^{(r)} = \eta_{\mathsf{A},n}^{(r)} = \eta_{\mathsf{S}}^{(r)} = \eta^{(r)} \leq \frac{1}{2\sqrt{1+c_r}HL}, \tag{37}$$

where H denotes the number of local iterations at each node per global round. Then under non-convex settings, our algorithm satisfies the following convergence result:

$$\begin{aligned}\frac{1}{\Gamma_R}\sum_{r=0}^{R-1}\eta^{(r)}\mathbb{E}\left\|\nabla F\left(\mathbf{w}^{(r)}\right)\right\|^2 &\leq 4\frac{F\left(\mathbf{w}^{(0)}\right) - F^*}{H\Gamma_R} \\ &+ \frac{4L}{\Gamma_R}\sum_{r=0}^{R-1}(\eta^{(r)})^2\Big(\sum_{k\in\mathcal{G}}(\lambda_{\mathsf{G},k}^{(r)})^2 + \sum_{n\in\mathcal{A}}(\lambda_{\mathsf{A},n}^{(r)})^2 + (\lambda_{\mathsf{S}}^{(r)})^2\Big)\sigma_g^2 \\ &+ \frac{2H^2L^2\sigma_g^2}{\Gamma_R}\sum_{r=0}^{R-1}(\eta^{(r)})^3 + \frac{4H^2L^2}{\Gamma_R}\sum_{r=0}^{R-1}(\eta^{(r)})^3\delta_r^2,\end{aligned} \tag{38}$$

where F^* is the minimum value that $F(\mathbf{w})$ can achieve and $\Gamma_R = \sum_{r=0}^{R-1} \eta^{(r)}$ is the summation of learning rates.

Proof. See Appendix A □

The impact of data heterogeneity after each round of data offloading is reflected both in the learning rate condition (37) and the last term of (38) in the convergence bound. From (37), we see that as the extent of data heterogeneity after data offloading gets larger, a smaller learning rate is required to guarantee the convergence of the algorithm. We also observe from (38) that the bound increases as the heterogeneity of data distributions across the nodes grows. The second term of the right-hand side of (38) captures the effect of the portion of data samples at each node on the convergence bound, which is time-varying due to data offloading. Additionally, by selecting an appropriate learning rate that satisfies $\sum_{r=0}^{R-1}(\eta^{(r)})^2 \to 0$, $\sum_{r=0}^{R-1}(\eta^{(r)})^3 \to 0$ and $\Gamma_R \to \infty$ for $R \to \infty$, the upper bound will diminish to zero. In particular, we can either adopt a decaying learning rate according to $\eta^{(r)} = \frac{\eta^{(0)}}{r+1}$ or keep it constant as $\eta^{(r)} = \frac{1}{\sqrt{HR}}$. This guarantees convergence to a stationary point of the non-convex loss function.

VI. Experimental Results

In this section, we provide experimental results to validate the effectiveness of the proposed methodology in SAGINs.

A. Simulation Setup

Dataset and model: We consider the following benchmark datasets for FL: MNIST, FMNIST, and CIFAR-10. Using MNIST and FMNIST, we train a convolutional neural network with two convolutional layers and two fully connected layers, and a convolutional neural network with two convolutional layers and one fully connected layer, respectively. Using CIFAR-10, we train the VGG-11 model. We conduct FL using the training set of each dataset, and evaluate the performance of the constructed global model using the testing set.

SAGIN setting: We consider $K = 50$ ground devices located at a squared target region of 1200 m $\times$ 1200 m. There are $N = 5$ air nodes at a height of 20 km above the target area, each serving 10 ground devices without overlapping. A series of LEO satellites cover the target region in each global round, where we adopt the `walkerStar` function in MATLAB to construct a constellation model. Fig. 3 shows the created satellite constellation, where 80 LEO satellites are distributed evenly across 5 different orbits with altitude of 800 km and inclination of 85°. We set the minimum elevation angle to communicate to 15°, and the latitude and longitude of the target region are 40° N and 86° W, respectively. We use `accessIntervals` function to calculate the coverage time of each satellite over the target region. Referring to the settings of prior works [31], [37], [40], we adopt the following parameter values for simulations: $f_{\mathsf{G},k} = 10^8$ Hz, $f_{\mathsf{A},n} = 10^9$ Hz, $f_{\mathsf{S},i} \in [1, 10] \times 10^9$ Hz, $m_{\mathsf{G},k} = m_{\mathsf{A},n} = m_{\mathsf{S}} = 3 \times 10^9$ cycles/sample, $p_{\mathsf{G},k} = 0.1$ W, $p_{\mathsf{A},n} = 1$ W, $p_{\mathsf{S},i} = 10$ W, $Z_{i,i+1}^{\mathsf{ISL},(r)} = 3.125$ Mbps, $N_0 = 3.98 \times 10^{-21}$ W/Hz. Here, to model the time-varying resource availability at the space layer over the target region, the CPU frequencies of satellites $f_{\mathsf{S},i}$ are sampled from a specific uniform distribution $[1, 10] \times 10^9$.

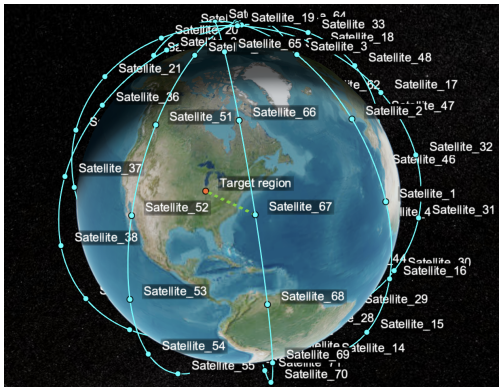


Fig. 3: Illustration of the satellite constellation constructed based on the `walkerStar` function.

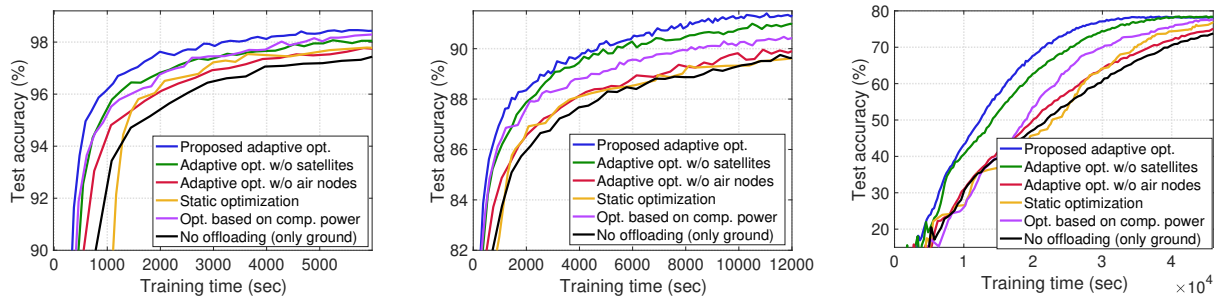
The training set of each dataset is distributed to the ground devices in two different scenarios: IID (independent and identically distributed) and non-IID cases. For the IID case, we allocate the training samples to the ground devices uniformly at random. For the non-IID scenario, we sort the training set according to each sample's class, split the sorted dataset into 200 shards, and then randomly assign 4 shards to each ground device. This introduces heterogeneous data distributions among ground devices. Note that the nodes in the space and air layers do not hold data at the beginning. We set the portion of non-sensitive to $\alpha_k = \alpha = 0.8$ for all ground devices, and also study the effect of α in Section VI-C.

Comparison schemes: For baselines, we first consider the scheme where only the ground devices process data without any data offloading, to see the advantage of adopting nodes in space and air layers as edge computing units. Satellites and air nodes are only used to aggregate the updated models. This baseline represents the majority of existing works that do not involve data offloading. Secondly, we consider optimizing data offloading only between the air and ground layers. Hence, the satellite-side computation power is not utilized during local model updates. Similarly, we optimize data offloading only between ground and space layers, without using the computational capabilities of air nodes during local update. We also consider the static optimization scheme, which applies our optimization strategy only at the initial global round and keeps the same solution throughout the remaining FL process. This baseline utilizes the computational resources of all three layers of SAGINs and is considered to see the impact of adaptive data offloading instead of using a fixed solution. Finally, we consider another baseline that utilizes the resources of all layers of SAGINs, where the number of data samples processed at each node is proportional to its computational power. For a fair comparison, we use FedAvg to aggregate the models in all baseline schemes and our methodology.

B. Main Experimental Results

We first observe Fig. 4, which reports the accuracy versus training time plots in different settings. Our key takeaways are as follows. First, the scheme without data offloading achieves

This article has been accepted for publication in IEEE Journal on Selected Areas in Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAC.2024.3459090



(a) MNIST, IID (b) FMNIST, IID (c) CIFAR-10, IID

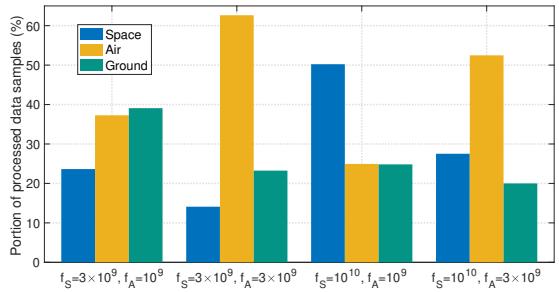
(d) MNIST, Non-IID (e) FMNIST, Non-IID (f) CIFAR-10, Non-IID

Fig. 4: Accuracy versus training time plots. For the static optimization scheme, we apply our inter-layer data offloading scheme only in the first global round and keep the intra-layer data fixed throughout the remaining rounds. The results show the advantage of adaptive data offloading optimization considering both space and air layers.

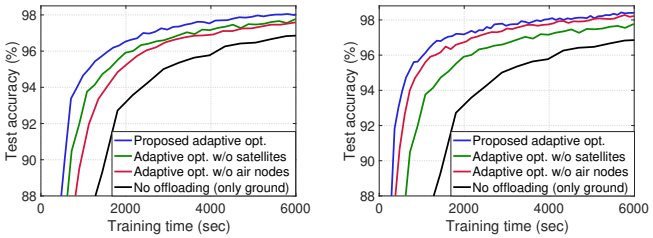
slow convergence, since the computation resources of space and air nodes are not utilized in this method. Utilizing only the computation resources of ground devices causes delays. We also observe that the fixed data offloading scheme achieves relatively low performance since the varying resource availability at the satellites are not considered in the scheme. If too many or too few data samples are offloaded to the space layer, the training process can be slowed down. This highlights the importance of adaptively optimizing data offloading, instead of relying on a fixed solution. We see that our approach, which leverages both the space and air layers, attains superior performance compared to the baselines that utilize only one of these layers. The proposed scheme also outperforms the scheme with optimized fixed data offloading and the baseline that conducts data offloading proportional to the computational power of each node in SAGINs. Further ablation studies on the effect of each layer are provided in the next subsection. The overall results highlight the significance of (i) inter-layer data offloading across space-air-ground, and (ii) adaptively conducting this to account for the network dynamics in SAGINs.

C. Varying System Parameters

Effect of computation powers of space and air nodes: In Fig. 5, we investigate the effect of computational capabilities at different layers, which can be adapted based on the battery constraint of each node. In extreme cases, the CPU frequency can drop to 0 if the battery is close to 0, and it can reach the maximum CPU frequency if the battery is sufficient. MNIST is considered in a non-IID setup. For these experiments, we set the CPU frequencies of space and air nodes (i.e., f_{S} and f_{A}, respectively) to the values depicted in the figure. Fig. 5a first shows the portion of data samples processed at each layer in our solution, depending on f_{S} and f_{A}. In the first case with $f_{\mathsf{S}} = 3 \times 10^9$ Hz and $f_{\mathsf{A}} = 10^9$ Hz (a scenario where both space and air nodes have insufficient battery), a relatively large number of data samples are allocated to the ground layer due

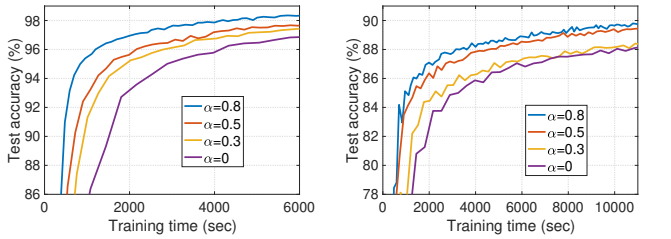


(a) Portion of data samples processed at each layer. We increase the CPU frequency of the (i) air node, (ii) space node, and (iii) both.



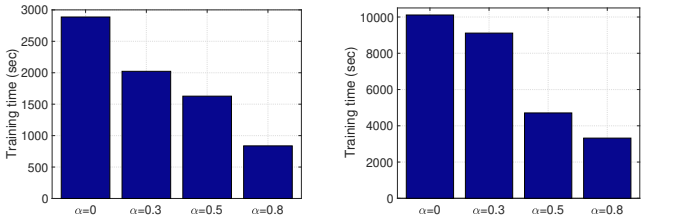
(b) Performance with $f_{\mathsf{S}} = 3{\times}10^9$ Hz, $f_{\mathsf{A}} = 10^9$ Hz. (c) Performance with $f_{\mathsf{S}} = 10^{10}$ Hz, $f_{\mathsf{A}} = 10^9$ Hz.

Fig. 5: Effect of computation capabilities of space/air nodes.



(a) Accuracy versus training time on MNIST.

(b) Accuracy versus training time on FMNIST.



(c) Training time to achieve 95% accuracy on MNIST.

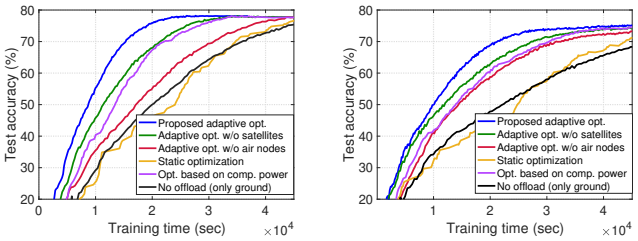
(d) Training time to achieve 88% accuracy on FMNIST.

Fig. 6: Effect of the portion of non-sensitive samples on our solution ($\alpha = 0$ reduces to no data offloading).

to the limited batteries at the space and air nodes. The air layer is allocated with more data samples than the space layer, indicating that the air nodes are considered more important than the satellites. This can be also confirmed from the accuracy curve in Fig. 5b, by comparing the scheme without satellites and the one without air nodes. Now if f_{A} increases from 10^9 Hz to 3×10^9 Hz (i.e., a scenario where the air node has more battery compared to the previous case), the portion of data samples processed at the air node becomes more dominant. On the other hand, if we increase f_{S} from 10^9 Hz to 10^{10} Hz (i.e., if the satellite has sufficient battery) while setting $f_{\mathsf{A}} = 10^9$ Hz, the role of the space layer becomes crucial, as also verified in Fig. 5c. Finally, when both space and air layers have sufficient resources ($f_{\mathsf{S}} = 10^{10}$ Hz and $f_{\mathsf{A}} = 3 \times 10^9$ Hz), only 20% of data samples are allocated to the ground layer. This allocation is the minimum amount of data that should be processed at the ground layer considering the portion of non-sensitive samples ($\alpha = 0.8$). Again, the results underscore the significance of taking advantage of the computation resources across all layers in SAGINs during the FL process.

Effect of the portion of non-sensitive data: In Fig. 6, we also study how the portion of non-sensitive samples α in each ground device's local dataset, affects the FL performance. If all data samples are privacy-sensitive (i.e., $\alpha = 0$), the setting reduces to conventional FL with no data offloading. Accuracy curves and the training time required to achieve the target accuracy are reported under the non-IID setting. We see that our methodology achieves the target accuracy faster as α increases, since a larger α provides a more flexible data offloading solution for our scheme.

Experiments with free-space path loss model: In practice, there often exists a line-of-sight link between the ground device and the air node. To validate the effectiveness of our approach under this setting, we use the free-space path loss model



(a) CIFAR-10, IID

(b) CIFAR-10, Non-IID

Fig. 7: Experimental results using the free-space pathloss model with a dominant line-of-sight link.

between the ground device and the air node, as adopted in [49], [50], considering that the line-of-sight link is dominant. We also adopt this free-space path loss model for satellite communication, where there is always a line-of-sight link. Fig. 7 shows the results using the CIFAR-10 dataset in both IID and non-IID scenarios. Compared to the setting with Rayleigh fading in Fig. 4, all schemes in Fig. 7 achieve faster convergence with less training time due to the reduced communication delay. It can be seen that our scheme consistently outperforms existing baselines by strategically taking advantage of the resources across space-air-ground integrated networks. The overall results further confirm the effectiveness and applicability of our method.

VII. Conclusion and Future Directions

In this paper, we proposed a distributed ML methodology that orchestrates FL in space-air-ground integrated networks. The core idea was to take advantage of both computation and communication resources of different layers in SAGINs to facilitate/accelerate FL in remote regions. We analytically characterized the latency of our method, and proposed an adaptive data offloading solution to minimize the training time depending on the current resource availability. We also derived the convergence bound of the scheme and guaranteed convergence to a stationary point for non-convex loss functions. The advantages of the proposed method as well as the effects of system parameters are investigated via simulations.

There are several promising directions for future research in this domain. One direction is to optimize the trajectories of air nodes to achieve a better performance within our framework. Another direction is to introduce an additional layer by considering the base stations or geostationary earth orbit satellites that can connect to the LEO satellites, to further enhance the performance.

Appendix

A. Proof of Theorem 1

For ease of notation, we adopt the following equivalent form for the global loss function:

$$F(\mathbf{w}) \triangleq \sum_{i \in \mathcal{P}} \lambda_i^{(r)} \ell_i^{(r+1)}(\mathbf{w}), \tag{39}$$

where $\mathcal{P} := \{(\mathsf{G}, k) \mid k \in \mathcal{G}\} \cup \{(\mathsf{A}, n) \mid n \in \mathcal{A}\} \cup \{\mathsf{S}\}$ and $\sum_{i \in \mathcal{P}} \lambda_i^{(r)} = 1$. Additionally, we define $\Phi_r =$

$\sum_{h=0}^{H-1}\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\mathbb{E}\left\|\mathbf{w}_i^{(r,h)}-\mathbf{w}^{(r)}\right\|^2$, where $\mathbf{w}_i^{(r,h)}$ represents the models parameter of device i after h local iterations within the r-th round. It denotes the intermediate model in (3), (4), or (6). To prove the convergence of the proposed algorithm, we first investigate how each round of training reduces the global loss, as formalized in Lemma 1.

Lemma 1. *Under Assumptions 1-3 and $\eta^{(r)}\leq\frac{1}{2HL}$, we have*

$$\mathbb{E}\Big[F\left(\mathbf{w}^{(r+1)}\right)\Big]\leq\mathbb{E}\left[F\left(\mathbf{w}^{(r)}\right)\right]-\frac{\eta^{(r)}H}{2}\mathbb{E}\left\|\nabla F\left(\mathbf{w}^{(r)}\right)\right\|^2+\frac{\eta^{(r)}L^2}{2}\Phi_r+(\eta^{(r)})^2HL\sigma_g^2\sum_{i\in\mathcal{P}}(\lambda_i^{(r)})^2. \quad (40)$$

To characterize the evolution of $\mathbb{E}\left\|\nabla F\left(\mathbf{w}^{(r)}\right)\right\|^2$ as shown in Theorem 1, we need to further bound the term $\Phi_r=\sum_{h=0}^{H-1}\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\mathbb{E}\left\|\mathbf{w}_i^{(r,h)}-\mathbf{w}^{(r)}\right\|^2$ that appears in Lemma 1. We establish an upper bound for Φ_r in Lemma 2.

Lemma 2. *Under Assumptions 1-3 and $\eta^{(r)}\leq\frac{1}{2HL}$, we have*

$$\Phi_r\leq 2(1+c_r)H^3(\eta^{(r)})^2\mathbb{E}\left\|\nabla F(\mathbf{w}^{(r)})\right\|^2+\frac{2}{3}H^3(\eta^{(r)})^2(\sigma_g^2+3\delta_r^2).$$

The proofs of Lemmas 1 and 2 are provided in Appendix B. Combining Lemmas 1 and 2, we obtain

$$\begin{aligned}\mathbb{E}\Big[F\left(\mathbf{w}^{(r+1)}\right)\Big]&\leq\mathbb{E}\Big[F\left(\mathbf{w}^{(r)}\right)\Big]-\frac{\eta^{(r)}H}{2}\mathbb{E}\left\|\nabla F\left(\mathbf{w}^{(r)}\right)\right\|^2\\&+(\eta^{(r)})^2HL\sigma_g^2\sum_{i\in\mathcal{P}}(\lambda_i^{(r)})^2+\frac{\eta^{(r)}L^2}{2}\Big\{2(1+c_r)H^3(\eta^{(r)})^2\\&\times\mathbb{E}\left\|\nabla F(\mathbf{w}^{(r)})\right\|^2+\frac{2}{3}H^3(\eta^{(r)})^2(\sigma_g^2+3\delta_r^2)\Big\}.\end{aligned}$$

Reorganizing the above inequality and utilizing (37) give rise to the following result:

$$\begin{aligned}\eta^{(r)}\mathbb{E}\left\|\nabla F\left(\mathbf{w}^{(r)}\right)\right\|^2&\leq 4\frac{\mathbb{E}\Big[F\left(\mathbf{w}^{(r)}\right)\Big]-\mathbb{E}\left[F\left(\mathbf{w}^{(r+1)}\right)\right]}{H}\\&+4(\eta^{(r)})^2L\sigma_g^2\sum_{i\in\mathcal{P}}(\lambda_i^{(r)})^2+2(\eta^{(r)})^3H^2L^2\sigma_g^2+4(\eta^{(r)})^3H^2L^2\delta_r^2.\end{aligned}$$

By telescopic expansion of the above inequality from $r=0$ to $R-1$, we can obtain the result shown in Theorem 1.

B. Proof of Lemmas

1) Proof of Lemma 1: For ease of notation, we denote $\boldsymbol{e}_i^{(r,h)}$, $i\in\mathcal{P}=\{(\mathsf{G},k)\mid k\in\mathcal{G}\}\cup\{(\mathsf{A},n)\mid n\in\mathcal{A}\}\cup\{\mathsf{S}\}$ as a mini-batch gradient $\tilde{\nabla}\ell_{\mathsf{G},k}^{(r+1)}(\mathbf{w}_{\mathsf{G},k}^{(r,h)}),k\in\mathcal{G}$, $\tilde{\nabla}\ell_{\mathsf{A},n}^{(r+1)}(\mathbf{w}_{\mathsf{A},n}^{(r,h)}),n\in\mathcal{A}$, or $\tilde{\nabla}\ell_{\mathsf{S}}^{(r+1)}(\mathbf{w}_{\mathsf{S}}^{(r,h)})$.

Due to the smoothness of local loss functions described in Assumption 1, the global loss function $F(\mathbf{w})$ is L-smooth as well. Based on the iteration $\sum_{h=0}^{H-1}\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\boldsymbol{e}_i^{(r,h)}=\mathbf{w}^{(r+1)}-\mathbf{w}^{(r)}$, we have

$$\mathbb{E}\Big[F\left(\mathbf{w}^{(r+1)}\right)\Big]\leq\mathbb{E}\Big[F\left(\mathbf{w}^{(r)}\right)\Big]+(\eta^{(r)})^2\frac{L}{2}\underbrace{\mathbb{E}\left\|\sum_{h=0}^{H-1}\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\boldsymbol{e}_i^{(r,h)}\right\|^2}_{\Psi_2}\underbrace{-\eta^{(r)}\mathbb{E}\left\langle\nabla F\left(\mathbf{w}^{(r)}\right),\sum_{h=0}^{H-1}\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\boldsymbol{e}_i^{(r,h)}\right\rangle}_{\Psi_1}. \quad (41)$$

We next bound Ψ_1 and Ψ_2. First, for Ψ_1, we have

$$\Psi_1=-\eta^{(r)}H\mathbb{E}\left\langle\nabla F\left(\mathbf{w}^{(r)}\right),\frac{1}{H}\sum_{h=0}^{H-1}\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\nabla\ell_i^{(r+1)}(\mathbf{w}_i^{(r,h)})\right\rangle.$$

Due to $-\langle\mathbf{a},\mathbf{b}\rangle=\frac{1}{2}\|\mathbf{a}-\mathbf{b}\|^2-\frac{1}{2}\|\mathbf{a}\|^2-\frac{1}{2}\|\mathbf{b}\|^2$, we have

$$\Psi_1=-\frac{\eta^{(r)}H}{2}\left\{\mathbb{E}\left\|\frac{1}{H}\sum_{h=0}^{H-1}\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\nabla\ell_i^{(r+1)}(\mathbf{w}_i^{(r,h)})\right\|^2+\mathbb{E}\left\|\nabla F\left(\mathbf{w}^{(r)}\right)\right\|^2\right\}+\frac{\eta^{(r)}H}{2}\underbrace{\mathbb{E}\left\|\frac{1}{H}\sum_{h=0}^{H-1}\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\nabla\ell_i^{(r+1)}(\mathbf{w}_i^{(r,h)})-\nabla F(\mathbf{w}^{(r)})\right\|^2}_{\Psi_3}.$$

Now based on $\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\nabla\ell_i^{(r+1)}(\mathbf{w}^{(r)})=\nabla F(\mathbf{w}^{(r)})$, $\sum_{i\in\mathcal{P}}\lambda_i^{(r)}=1$, the Jensen's inequality, and Assumption 1, we can bound Ψ_3 as

$$\Psi_3\leq\frac{L^2}{H}\sum_{h=0}^{H-1}\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\mathbb{E}\left\|\mathbf{w}_i^{(r,h)}-\mathbf{w}^{(r)}\right\|^2,$$

where the inequality comes from Assumption 1.

For Ψ_2, by using the Cauchy-Schwartz inequality, we have

$$\begin{aligned}\Psi_2&\leq 2\mathbb{E}\left\|\sum_{h=0}^{H-1}\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\left(\tilde{\nabla}\ell_i^{(r+1)}(\mathbf{w}_i^{(r,h)})-\nabla\ell_i^{(r+1)}(\mathbf{w}_i^{(r,h)})\right)\right\|^2\\&+2\mathbb{E}\left\|\sum_{h=0}^{H-1}\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\nabla\ell_i^{(r+1)}(\mathbf{w}_i^{(r,h)})\right\|^2\\&\leq 2H\sigma_g^2\sum_{i\in\mathcal{P}}(\lambda_i^{(r)})^2+2\mathbb{E}\left\|\sum_{h=0}^{H-1}\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\nabla\ell_i^{(r+1)}(\mathbf{w}_i^{(r,h)})\right\|^2.\end{aligned}$$

Utilizing $\eta^{(r)}\leq\frac{1}{2HL}$ and combining Ψ_1, Ψ_2, and Ψ_3 with (41) give rise to Lemma 1.

2) Proof of Lemma 2: We first denote $s^{(r,\tau)}=\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\mathbb{E}\left\|\mathbf{w}_i^{(r,\tau)}-\mathbf{w}^{(r)}\right\|^2$, which can be bounded as

$$\begin{aligned}s^{(r,\tau)}&=(\eta^{(r)})^2\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\mathbb{E}\left\|\sum_{k=0}^{\tau-1}\boldsymbol{e}_i^{(t,k)}\right\|^2\leq\tau(\eta^{(r)})^2\sum_{h=0}^{\tau-1}\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\mathbb{E}\left\|\boldsymbol{e}_i^{(r,h)}\right\|^2\\&=\tau(\eta^{(r)})^2\sum_{h=0}^{\tau-1}\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\mathbb{E}\left\|\boldsymbol{e}_i^{(r,h)}-\nabla\ell_i^{(r+1)}(\mathbf{w}_i^{(r,h)})\right\|^2\\&+\tau(\eta^{(r)})^2\sum_{h=0}^{\tau-1}\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\mathbb{E}\left\|\nabla\ell_i^{(r+1)}(\mathbf{w}_i^{(r,h)})\right\|^2\\&\leq\tau(\eta^{(r)})^2\sum_{h=0}^{\tau-1}\underbrace{\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\mathbb{E}\left\|\nabla\ell_i^{(r+1)}(\mathbf{w}_i^{(r,h)})\right\|^2}_{\Psi_4}+(\eta^{(r)})^2\tau^2\sigma_g^2. \quad (42)\end{aligned}$$

Next, we establish an upper bound for Ψ_4 as

$$\begin{aligned}\Psi_4&=\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\mathbb{E}\left\|\nabla\ell_i^{(r+1)}(\mathbf{w}_i^{(r,h)})\mp\nabla\ell_i^{(r+1)}(\mathbf{w}^{(r)})\mp\nabla F(\mathbf{w}^{(r)})\right\|^2\\&\leq 3L^2\sum_{i\in\mathcal{P}}\lambda_i^{(r)}\mathbb{E}\left\|\mathbf{w}_i^{(r,h)}-\mathbf{w}^{(r)}\right\|^2+(3+3c_r)\mathbb{E}\left\|F(\mathbf{w}^{(r)})\right\|^2+3\delta_r^2,\end{aligned}$$

where the last inequality comes from Assumption 3. By plugging the upper bound of Ψ_4 into (42) and taking summation over τ from 1 to $H-1$, we obtain

$$\sum_{\tau=1}^{H-1} s^{(r,\tau)} \leq 2H^2L^2(\eta^{(r)})^2 \sum_{h=0}^{H-1} s^{(r,h)} + (1+c_r)H^3(\eta^{(r)})^2 \times \mathbb{E}\left\|\nabla F(\mathbf{w}^{(r)})\right\|^2 + H^3(\eta^{(r)})^2(\frac{1}{3}\sigma_g^2 + \delta_r^2), \quad (43)$$

where we utilize the property of arithmetic sequence. Utilizing $s^{(r,0)} = 0$ and rearranging (43), we have

$$(1-2H^2L^2(\eta^{(r)})^2) \sum_{\tau=0}^{H-1} s^{(r,\tau)} \leq (1+c_r)H^3(\eta^{(r)})^2\mathbb{E}\left\|\nabla F(\mathbf{w}^{(r)})\right\|^2 + H^3(\eta^{(r)})^2(\frac{1}{3}\sigma_g^2 + \delta_r^2).$$

Since $\eta^{(r)} \leq \frac{1}{2HL}$ holds, we have $(1-2H^2L^2(\eta^{(r)})^2) \geq \frac{1}{2}$. Scaling the above inequality gives rise to Lemma 2.

REFERENCES

[1] B. McMahan, E. Moore *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[2] P. Kairouz, H. B. McMahan *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[3] T. Li, A. K. Sahu *et al.*, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[4] S. Wang, T. Tuor *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[5] L. Liu, J. Zhang *et al.*, "Client-edge-cloud hierarchical federated learning," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[6] M. S. H. Abad, E. Ozfatura *et al.*, "Hierarchical federated learning across heterogeneous cellular networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8866–8870.

[7] A. G. Roy, S. Siddiqui *et al.*, "Braintorrent: A peer-to-peer environment for decentralized federated learning," *arXiv preprint arXiv:1905.06731*, 2019.

[8] A. Lalitha, S. Shekhar *et al.*, "Fully decentralized federated learning," in *Third workshop on Bayesian Deep Learning (NeurIPS)*, 2018.

[9] J. Liu, Y. Shi *et al.*, "Space-air-ground integrated network: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2714–2741, 2018.

[10] J. Ye, S. Dang *et al.*, "Space-air-ground integrated networks: Outage performance analysis," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 7897–7912, 2020.

[11] B. Shang, Y. Yi *et al.*, "Computing over space-air-ground integrated networks: Challenges and opportunities," *IEEE Network*, vol. 35, no. 4, pp. 302–309, 2021.

[12] S. Yu, X. Gong *et al.*, "Ec-sagins: Edge-computing-enhanced space–air–ground-integrated networks for internet of vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5742–5754, 2021.

[13] Y. Liu, L. Jiang *et al.*, "Energy-efficient space–air–ground integrated edge computing for internet of remote things: A federated drl approach," *IEEE Internet of Things Journal*, vol. 10, no. 6, pp. 4845–4856, 2022.

[14] H. H. Yang, Z. Liu *et al.*, "Scheduling policies for federated learning in wireless networks," *IEEE transactions on communications*, vol. 68, no. 1, pp. 317–333, 2019.

[15] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3546–3557, 2020.

[16] M. Chen, Z. Yang *et al.*, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2020.

[17] M. Chen, H. V. Poor *et al.*, "Convergence time optimization for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2457–2471, 2020.

[18] W. Y. B. Lim, J. S. Ng *et al.*, "Decentralized edge intelligence: A dynamic resource allocation framework for hierarchical federated learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 3, pp. 536–550, 2021.

[19] ——, "Dynamic edge association and resource allocation in self-organizing hierarchical federated learning networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3640–3653, 2021.

[20] D.-J. Han, M. Choi *et al.*, "Fedmes: Speeding up federated learning with multiple edge servers," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3870–3885, 2021.

[21] J. Wang, A. K. Sahu *et al.*, "Matcha: Speeding up decentralized sgd via matching decomposition sampling," in *2019 Sixth Indian Control Conference (ICC)*. IEEE, 2019, pp. 299–300.

[22] A. Koloskova, N. Loizou *et al.*, "A unified theory of decentralized sgd with changing topology and local updates," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5381–5393.

[23] N. Huang, M. Dai *et al.*, "Wireless federated learning with hybrid local and centralized training: A latency minimization design," *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[24] B. Ganguly, S. Hosseinalipour *et al.*, "Multi-edge server-assisted dynamic federated learning with an optimized floating aggregation point," *IEEE/ACM Transactions on Networking*, 2023.

[25] S. Hosseinalipour, S. Wang *et al.*, "Parallel successive learning for dynamic distributed model training over heterogeneous wireless networks," *IEEE/ACM Transactions on Networking*, 2023.

[26] Y. Wang, Z. Su *et al.*, "Learning in the air: Secure federated learning for uav-assisted crowdsensing," *IEEE Transactions on network science and engineering*, vol. 8, no. 2, pp. 1055–1069, 2020.

[27] H. Zhang and L. Hanzo, "Federated learning assisted multi-uav networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 14 104–14 109, 2020.

[28] T. Zeng, O. Semiari *et al.*, "Federated learning in the sky: Joint power allocation and scheduling with uav swarms," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[29] B. Matthiesen, N. Razmi *et al.*, "Federated learning in satellite constellations," *IEEE Network*, 2023.

[30] J. So, K. Hsieh *et al.*, "Fedspace: An efficient federated learning framework at satellites and ground stations," *arXiv preprint arXiv:2202.01267*, 2022.

[31] N. Razmi, B. Matthiesen *et al.*, "On-board federated learning for dense leo constellations," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 4715–4720.

[32] ——, "Scheduling for ground-assisted federated learning in leo satellite constellations," in *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 1102–1106.

[33] ——, "Ground-assisted federated learning in leo satellite constellations," *IEEE Wireless Communications Letters*, vol. 11, no. 4, pp. 717–721, 2022.

[34] M. Elmahallawy and T. Luo, "Fedhap: Fast federated learning for leo constellations using collaborative haps," in *2022 14th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2022, pp. 888–893.

[35] Z. Zhai, Q. Wu *et al.*, "Fedleo: An offloading-assisted decentralized federated learning framework for low earth orbit satellite networks," *IEEE Transactions on Mobile Computing*, 2023.

[36] M. Elmahallawy and T. Luo, "Optimizing federated learning in leo satellite constellations via intra-plane model propagation and sink satellite scheduling," *arXiv preprint arXiv:2302.13447*, 2023.

[37] T. K. Rodrigues and N. Kato, "Hybrid centralized and distributed learning for mec-equipped satellite 6g networks," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 4, pp. 1201–1211, 2023.

[38] H. Chen, M. Xiao *et al.*, "Satellite-based computing networks with federated learning," *IEEE Wireless Communications*, vol. 29, no. 1, pp. 78–84, 2022.

[39] Y. Wang, C. Zou *et al.*, "Federated learning over leo satellite," in *2022 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2022, pp. 1652–1657.

[40] Q. Fang, Z. Zhai *et al.*, "Olive branch learning: A topology-aware federated learning framework for space-air-ground integrated network," *IEEE Transactions on Wireless Communications*, vol. 22, no. 7, pp. 4534 – 4551, 2023.

[41] D.-J. Han, S. Hosseinalipour *et al.*, "Cooperative federated learning over ground-to-satellite integrated networks: Joint local computation and data offloading," *IEEE Journal on Selected Areas in Communications*, 2024.

[42] N. Kato, Z. M. Fadlullah *et al.*, "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 140–147, 2019.

[43] F. Tang, C. Wen *et al.*, "Federated learning for intelligent transmission with space-air-ground integrated network (sagin) toward 6g," *IEEE Network*, 2022.

[44] A. Paul, K. Singh *et al.*, "Digital twin-assisted space-air-ground integrated networks for vehicular edge computing," *IEEE Journal of Selected Topics in Signal Processing*, 2023.
[45] I. Leyva-Mayorga, B. Soret *et al.*, "Inter-plane inter-satellite connectivity in dense leo constellations," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3430–3443, 2021.
[46] X. Pang, N. Zhao *et al.*, "Irs-assisted secure uav transmission via joint trajectory and beamforming design," *IEEE Transactions on Communications*, vol. 70, no. 2, pp. 1140–1152, 2021.
[47] Y. Guo, R. Zhao *et al.*, "Distributed machine learning for multiuser mobile edge computing systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 3, pp. 460–473, 2022.
[48] D. Callegaro and M. Levorato, "Optimal edge computing for infrastructure-assisted uav systems," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1782–1792, 2021.
[49] M. Fu, Y. Shi *et al.*, "Federated learning via unmanned aerial vehicle," *IEEE Transactions on Wireless Communications*, 2023.
[50] Q. Wu, Y. Zeng *et al.*, "Joint trajectory and communication design for multi-uav enabled wireless networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 2109–2121, 2018.
[51] K. Sikorski, "Bisection is optimal," *Numerische Mathematik*, vol. 40, pp. 111–117, 1982.
[52] Z. Wang, Y. Zhou *et al.*, "Interference management for over-the-air federated learning in multi-cell wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 8, pp. 2361–2377, 2022.
[53] I. Flores and G. Madpis, "Average binary search length for dense ordered lists," *Communications of the ACM*, vol. 14, no. 9, pp. 602–603, 1971.
[54] Y. Shi, J. Cheng *et al.*, "Smoothed l_p-minimization for green cloud-ran with user admission control," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 4, pp. 1022–1036, 2016.

Dong-Jun Han (Member, IEEE) is an Assistant Professor at the Department of Computer Science and Engineering at Yonsei University, South Korea. Previously, he was a postdoctoral researcher in the School of Electrical and Computer Engineering at Purdue University. He received the B.S. degrees in mathematics and electrical engineering, and the M.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2016, 2018, and 2022, respectively. His research interest is at the intersection of communications, networking, and machine learning, specifically in distributed/federated machine learning and network optimization.

Wenzhi Fang (Graduate Student Member, IEEE) received his B.S. degree from Shanghai University in 2020 and completed his master's degree at ShanghaiTech University in 2023. Currently, he is pursuing a PhD in electrical and computer engineering at Purdue University, West Lafayette, US. His research interests focus on optimization theory and its applications in machine learning, signal processing, and wireless networks.

Seyyedali Hosseinalipour (Member, IEEE) received the B.S. degree in electrical engineering from Amirkabir University of Technology, Tehran, Iran, in 2015 with high honor and top-rank recognition. He then received the M.S. and Ph.D. degrees in electrical engineering from North Carolina State University, NC, USA, in 2017 and 2020, respectively. He was the recipient of the ECE Doctoral Scholar of the Year Award (2020) and ECE Distinguished Dissertation Award (2021) at North Carolina State University. He was a postdoctoral researcher at Purdue University, IN, USA from 2020 to 2022. He is currently an assistant professor at the Department of Electrical Engineering at the University at Buffalo (SUNY). He has served as the TPC Co-Chair of workshops and symposiums related to distributed machine learning and edge computing held in conjunction with IEEE INFOCOM, IEEE GLOBECOM, IEEE ICC, IEEE/CVF CVPR, IEEE MSN, and IEEE VTC. Also, he has served as the guest editor for IEEE Internet of Things Magazine. His research interests include the analysis of modern wireless networks, synergies between machine learning methods and fog computing systems, distributed/federated machine learning, and network optimization.

Mung Chiang (Fellow, IEEE) is the 13th President of Purdue University and the Roscoe H. George Distinguished Professor of Electrical and Computer Engineering. Previously he was the Arthur LeGrand Doty Professor of Electrical Engineering at Princeton University, where he founded the Princeton Edge Lab in 2009 and cofounded several startups spun out from there. The 2013 NSF Alan T. Waterman Awardee, he also received a Guggenheim Fellowship, the IEEE Kiyo Tomiyau Award, the IEEE INFOCOM Achievement Award, and is a member of the National Academy of Inventors and Royal Swedish Academy of Engineering Science. He served as the Science and Technology Adviser to the U.S. Secretary of State.

Christopher G. Brinton (Senior Member, IEEE) is the Elmore Rising Star Associate Professor of Electrical and Computer Engineering (ECE) at Purdue University. His research interest is at the intersection of networking, communications, and machine learning, specifically in fog/edge network intelligence, distributed machine learning, and AI/ML-inspired wireless network optimization. Dr. Brinton is a recipient of five of the US top early career awards, from the National Science Foundation (CAREER), Office of Naval Research (YIP), Defense Advanced Research Projects Agency (YFA and Director's Fellowship), and Air Force Office of Scientific Research (YIP), the IEEE Communication Society William Bennett Prize Best Paper Award, the Intel Rising Star Faculty Award, the Qualcomm Faculty Award, and roughly $17M in sponsored research projects as a PI or co-PI. He has also been awarded Purdue College of Engineering Faculty Excellence Awards in Early Career Research, Early Career Teaching, and Online Learning. He currently serves as an Associate Editor for IEEE/ACM Transactions on Networking, and previously was an Associate Editor for IEEE Transactions on Wireless Communications. Prior to joining Purdue, Dr. Brinton was the Associate Director of the EDGE Lab and a Lecturer of Electrical Engineering at Princeton University. He also co-founded Zoomi Inc., a big data startup company that holds US Patents in machine learning for education. His book The Power of Networks: 6 Principles That Connect our Lives and associated Massive Open Online Courses (MOOCs) reached over 400,000 students. Dr. Brinton received the PhD (with honors) and MS Degrees from Princeton in 2016 and 2013, respectively, both in Electrical Engineering.