Communication-Efficient Satellite-Ground Federated Learning through Progressive Weight Quantization

Chen Yang*, Jinliang Yuan*, Yaozong Wu, Qibo Sun, Ao Zhou, Shangguang Wang, *Senior Member, IEEE*, and Mengwei Xu

Abstract—Large constellations of Low Earth Orbit (LEO) satellites have been launched for Earth observation and satellite-ground communication, which collect massive imagery and sensor data. These data can enhance the AI capabilities of satellites to address global challenges such as real-time disaster navigation and mitigation. Prior studies proposed leveraging federated learning (FL) across satellite-ground to collaboratively train a share machine learning (ML) model in a privacy-preserving mechanism. However, they mostly focus on single unique challenges such as limited ground-to-satellite bandwidth, short connection window, and long connection cycle, while ignoring the completeness of these challenges in deploying efficient FL frameworks in space. In this paper, we propose an efficient satellite-ground FL framework, SatelliteFL, to address these three challenges collectively. Its key idea is to ensure that each satellite must complete per-round training within each connection window. Moreover, we design a progressive block-wise quantization algorithm that determines a unique bitwidth for each block of the ML model to maximize the model utility while not exceeding the connection window. We evaluate SatelliteFL by plugging an implemented FL platform into real-world satellite networks and satellite images. The results show that SatelliteFL highly accelerates the convergence by up to 2.8× and improves the bandwidth utilization ratio by up to 9.3× compared to the state-of-the-art methods.

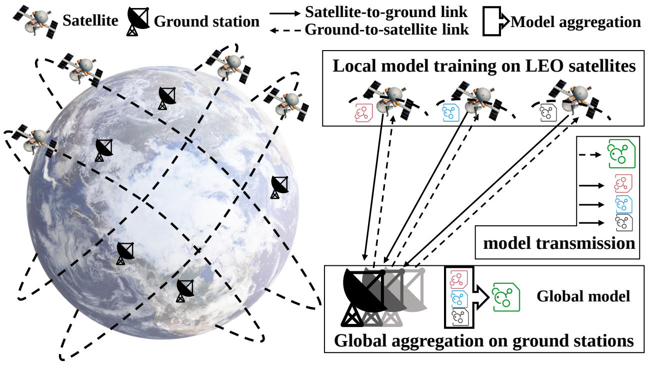
Index Terms—In-orbit computing, satellite network, federated learning.

1 Introduction

Over the years, Earth observation satellites have consistently provided rich informational support in critical areas such as food security, disaster navigation, climate change, and disease spread [1]–[3]. In recent years, significant advancements in satellite technology have substantially reduced satellite deployment costs, leading to the emergence of low Earth orbit (LEO) satellite constellations as a mainstream trend. For example, companies like Planet [4] are now able to collect over 350 million square kilometers of images daily. Massive imagery and sensor data (10s-100s of TB/day) collected by these satellites can effectively enhance machine learning (ML) capabilities to address numerous global challenges. For example, the Sunflower Satellites, with the aid of AI model, can spot wildfires in less than 1 minute, which could have been used to avoid the 2019-2020 Australian 'Black Summer' bushfire with over $70 billion in property and economic losses [5].

Traditional satellite-based machine learning tasks are mostly completed on the ground. Those collected data by LEO constellation, in Fig. 1(a), is transmitted to the ground station across satellite-to-ground link, and used to train ML models with a powerful computation cluster. However, this paradigm is becoming increasingly infeasible for the following reasons: 1) downloading the raw data collected from satellites would raise substantial overhead to the satellite-ground link. Nevertheless, the current satellite-ground link bandwidth does not even support the download of all the collected data. 2) sharing high-resolution Earth observation images, such as rare natural resources and significant economic activities may not always be feasible due to regulatory restrictions imposed by different countries [6].

There have been several efforts that dedicate to deploying in-orbit computing to address these challenges, which



(a) LEO satellite constellation (b) FL at LEO satellites and ground stations

Fig. 1: (a) LEO satellite constellations comprise many low-Earth orbit satellites, which orbit 500 to 2,000 kilometers from Earth and offer communication services collaborated with the ground stations; (b) Satellite-ground FL framework in space without downloading satellite data to the ground.

- *Chen Yang, Jinliang Yuan, Yaozong Wu, Qibo Sun, Ao Zhou, Shangguang Wang and Mengwei Xu are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. (Corresponding author: Qibo Sun) (E-mail:{yangc; yuanjinliang; yzwu; qbsun; aozhou; sgwang and mwx}@bupt.edu.cn)*

is motivated by the improvement of satellite computation capabilities. OEC [7] is a proposed orbital edge computing platform to support various computing tasks on satellites so that those collected data can be processed locally. SmallSats [8] aims to bring efficient inference and training of DNNs in space by introducing lightweight hardware accelerators and designing compact ML algorithms. However, it also faces two key challenges with the deployment of on-board training. First, centralized ML on a single satellite with limited training data is hard to obtain an advanced model, due to privacy concerns when exchanging sensitive imagery data across LEO satellites [2]. Second, considering the memory and power cost of inter-satellite communication, cross-satellite distributed ML in space is currently infeasible in LEO satellite constellations [9].

Our goal in this work is to address these challenges by designing an efficient FL framework that enables satellites to collaboratively train a shared ML model with the coordination of ground stations. Our system is designed with several steps as shown in Fig. 1(b): (1) the global model is dispatched to those connected satellites through ground-to-satellite link; (2) connected satellites run on-board training to update local models without sharing their raw data; (3) these models are transmitted to the ground station through the satellite-to-ground link for global model aggregation. These steps form per-round FL training and iterate with another batch of connected satellites in the next round until the global model converges. Some prior literature [2], [10]–[12] also introduces FL framework across satellites and ground stations to enable efficient AI applications in space. Most of them focus on tackling the straggler and staleness problems of synchronous or asynchronous FL algorithms, due to the system heterogeneity of satellites [2], [13]. Another important problem in this satellite-ground FL framework is the slow model convergence caused by limited ground-to-satellite bandwidth [10], [12]. However, it is still fundamentally challenging to apply existing solutions to this environment when considering all the unique challenges introduced by orbital dynamics of LEO satellites:

- **Limited ground-to-satellite bandwidth:** Dispatching the global model to connected satellites through the ground-to-satellite link is a key step in this environment to guarantee the success of FL training. However, the ground-satellite link is asymmetric. Specifically, the ground-to-satellite link bandwidth is extremely limited, often only about 200 Kbps [14]. Thus, even if the satellite-to-ground download bandwidth is as high as a few Gbps, the entire FL training speed will be severely constrained by the ground-to-satellite link.
- **Short connection window:** Considering the orbital dynamics of LEO satellites and limited deployment of ground stations on Earth, each satellite tends to stay a short time ($\leq$ 10 minutes [14]) in the coverage of a particular ground station before flying away. This causes the per-round FL training to be interrupted because it is often impossible to complete the above steps in such a short connection window. Especially, the first step of per-round FL training across the limited ground-to-satellite link makes this interruption more frequent.
- **Long connection cycle:** Furthermore, each satellite would only get several connection periods with ground stations in one day, and it often takes over 6 hours [14] for any two connection periods. Therefore, once a satellite fails to transmit its model update to the ground in the assigned connection window, it takes excessive time to wait till the next connection window and the update is likely to be expired.

In this paper, we present an efficient satellite-ground FL framework, namely SatelliteFL, that leverages progressive weight quantization to compress the communication data size across ground-to-satellite links. (Quantization involves representing model parameters with fewer bitwidth to reduce model size while maintaining acceptable neural network model performance degradation.) We first highlight the importance of ensuring that each satellite must complete the per-round FL training within each connection window to tackle the above three challenges. Then, we formulate this per-round FL training as a delay-constrained optimization problem to maximize the model utility while not exceeding the connection window. However, solving this optimization problem is also challenging due to the complex trade-off between model utility and constrained connection window caused by the limited decision space of quantization bitwidth. Finally, we propose a progressive block-wise quantization (ProBQ) algorithm that fine-grained quantizes each block of model with a unique quantization bitwidth, which can solve the problem and achieve near-optimal performance. The rationale is that we cherry-pick a proper quantization bitwidth for each block of ML model, instead of the whole model; it makes full use of the ground-to-satellite link bandwidth therefore retains more weight information to improve the model accuracy.

We evaluate SatelliteFL on our implemented FL platform plugged in real-world satellite networks. Extensive experiments are performed on classical ML models and open-source satellite imagery data, and we will also open-source our platform once published. The results show that, SatelliteFL can accelerate the convergence by 1.8×(on average, up to 2.8×) and improve the bandwidth utilization ratio by 4.5×(on average, up to 9.3×) compared to the state-of-the-art satellite-ground FL methods.

We make the following contributions:

- We formulate the collaborative satellite-ground training problem, with ground-to-satellite bandwidth, short connection window and long connection cycle, as a delay-constrained optimization problem. To tackle these challenges, we introduce an efficient satellite-ground FL framework that each satellite should complete per-round FL training within each connection window.
- We propose a progressive block-wise quantization algorithm to maximize the model utility while not exceeding the connection window.
- We validate the effectiveness of our solution with real-world satellite networks and satellite imagery dataset, and show that it significantly accelerates FL training and improve the bandwidth utilization ratio over the state-of-the-art methods.

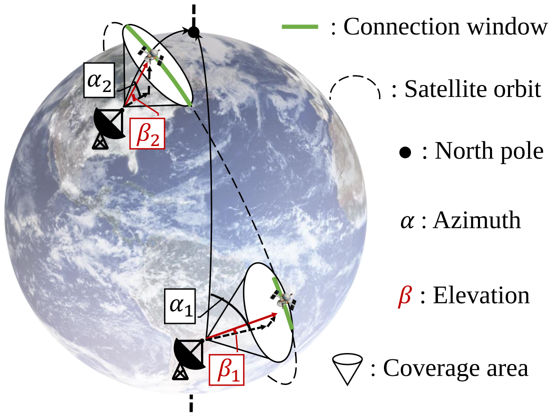


Fig. 2: LEO satellite operates in orbits with two angles and its connection window is related to the signal coverage of ground station. (1) The location of the satellite with respect to a ground station on Earth is defined by the two angles azimuth and elevation; (2) The connection window of the satellite is determined by its orbit and the signal coverage of ground station on Earth, which also indicates the connection window of this satellite at a time.

Algorithm 1: Satellite-ground FL framework

```
Input  : total satellites $\mathcal{S}$, initialized global model $\omega_0$.
Output: the global model $\omega_r$.
Ground station executes: // global coordination
    for $r \leftarrow 0$ to $R$ do
        // per-round of FL training
        $\mathcal{S}_c, \{\tau_d\} \leftarrow$ connected satellites and their duration
        $\mathcal{S}_s \leftarrow$ randomly select $K_s$ satellites from $\mathcal{S}_c$
        for each satellite $k \in \mathcal{S}_s$ with duration $\tau_d$ do
            $t_{g,s} \leftarrow$ Dispatch the $\omega_r^k$ to satellite $k$
            $\omega_{r+1}^k, t_c$ and $t_{s,g} \leftarrow$ SatelliteUpdate($\omega_r^k$)
            if $t_{s,g} + t_c + t_{g,s} > \tau_d$ then
                Satellite $k$ failed this round of FL training
                Remove $k$ from selected $\mathcal{S}_s$
        Update global $\omega_{r+1}$ as Eq. (3)
SatelliteUpdate($\omega_r^k$): // local training
    $\omega_{r+1}^k \leftarrow$ on-board training to update $\omega_r^k$ as Eq. (2)
    $t_c \leftarrow$ obtain the computation time $t_c$
    $t_{s,g} \leftarrow$ return $\omega_{r+1}^k$ and obtain communication time $t_{s,g}$
```

2 Background and Motivations

In this section, we first introduce the communication model of satellite-ground dynamic links and the system model of FL at LEO satellites. Then, we formulate the optimization model of deploying FL framework across satellites and ground stations based on the unique communication model. Finally, we highlight the key obstacles and the limitations of existing solutions to enable communication-efficient FL training in this satellite-ground cooperative environment.

2.1 Communication model of satellite-ground link

In an Earth observation scenario, we first illustrate the background of LEO satellites related to the dynamic communication link. Then, we introduce two important definitions of *connected satellite* and *connection window*.

In Fig. 2, there are two observation angles: azimuth and elevation that decide the location of a satellite with respect to a ground station on Earth. The former denotes the angle between the North pole, measured clockwise around the ground station's horizon, and the satellite, while the latter denotes the vertical angular distance between the satellite and the ground station's local horizon. Specifically, when the satellite passes directly over the ground station, the elevation angle reaches its maximum value of 90 degrees. Due to the satellite's low orbit, it often takes only a few minutes for a satellite with a low elevation (β_1 in Fig. 2) from entering the signal coverage of a ground station to leaving it. But, with a higher elevation ($\beta_2 > \beta_1$), this satellite gets closer to Earth's surface, thus delivering a longer connection window as shown in Fig. 2. These observations reveal that: (1) satellites aren't always connected to ground station, nor are they intended to be; (2) and even when they do, each connection window is short.

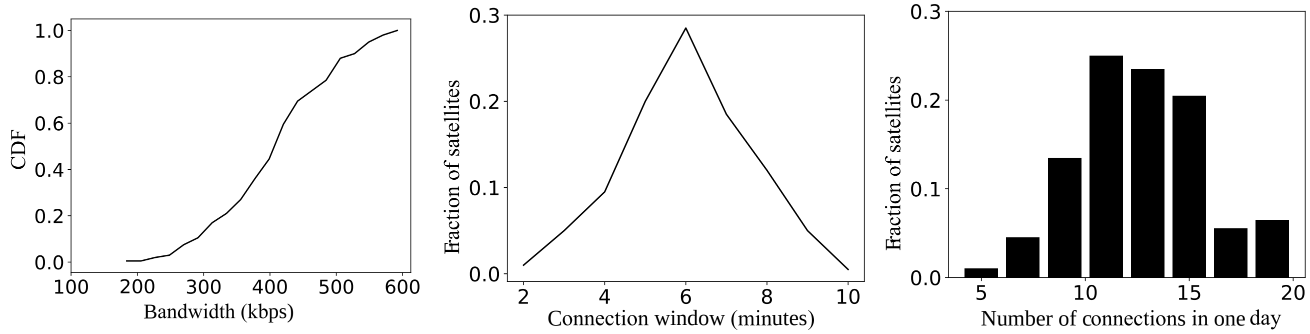
Consider a collection LEO satellites $\mathcal{S}$ and ground stations $\mathcal{G}$ for Earth observation. We introduce a continuous wall clock time t and a discrete time index $i \in \{0, 1, 2, \cdots\}$ with each adjacent time indexes having τ wall clock time interval. We denote the wall clock time interval from $i\tau$ to $(i+1)\tau$ as $[i]$. If a satellite $s \in \mathcal{S}$ is captured by a ground station $g \in \mathcal{G}$ at any time $t \in [i]$, their communication link would be established. We define the satellite s as a *connected satellite* in time interval $[i]$. As s moves out of signal coverage of g, the link breaks, and s is no longer a *connected satellite*. We define the duration of this link as *connection window*, which is denoted as τ_d, and we have $\tau_d \leq \tau$.

In this work, we assume that all the ground stations work as a powerful computing cluster without considering the transmission delay across those geographically-distributed ground stations. This assumption is theoretically sound because the fabric connectivity among them is always well-provisioned and much faster than the satellite-ground links [15]. Therefore, a satellite connected to anyone ground station can be considered to be connected to a cluster with all the ground stations $\mathcal{G}$.

2.2 Optimization model of satellite-ground FL

LEO satellites in this Earth observation scenario have collected a lot of imagery and sensor data to train a global model. Due to data privacy and bandwidth limitation concerns, a satellite-ground cooperative FL framework is proposed to collaboratively learn this model with on-board training and on-ground coordination in Fig. 1(a). We follow the classical synchronized stochastic gradient descent (SGD) method to formulate this cooperative FL process.

Consider K satellites in $\mathcal{S}$ and a cluster with ground stations $\mathcal{G}$. For each satellite $k \in \{1, 2, \cdots, K\}$, it collects and stores an imagery dataset $\mathcal{D}_k$. These satellites aim to collaboratively learn a global model ω by minimizing a



(a) Limited ground-to-satellite bandwidth (b) Short connection window (c) Long connection cycle

Fig. 3: LEO satellite's three distinctive characteristics on satellite-ground connection [17]–[19].

global objective function $F(\omega)$ as follows:

$$\min_{\omega}\{F(\omega)=\sum_{k=1}^{K}\frac{n_k}{n}f_k(\omega)\}, \tag{1}$$

where $n_k = |\mathcal{D}_k|$ denotes the samples of training data at satellite k, f_k denotes the local objective function of satellite k, and we have $n=\sum_{k=1}^{K} n_k$. The global model ω is maintained and advanced on the cluster of ground stations. We introduce the most popular FL protocol, FedAvg [16], that performs local SGD algorithm on satellites and aggregates those models of satellites on the ground to advance global model. In particular, we only randomly select K_s satellites from K_c *connected satellites* in each round, where $K_c \leq K$. The local update on each selected satellite $k \in K_s$ can be formulated as:

$$\omega_{r+1}^{k}=\omega_r-\eta\nabla f_k(\omega_r), \tag{2}$$

where η denotes the hyper-parameter learning rate and r denotes the global training round across satellites and ground stations. The global update is to aggregate the received models from K_s selected satellites, which can be formulated as:

$$\omega_{r+1}=\sum_{k\in K_s}\frac{n_k}{n}\omega_{r+1}^{k}. \tag{3}$$

Consider the wall clock time during per-round of FL training, which includes three key steps: 1) dispatching global model to each satellite; 2) on-board training to update local model; 3) sending local model to ground station. We denote the time of above three steps as $t_{g,s}$, t_c and $t_{s,g}$, respectively. In this satellite-ground FL framework, it is highly likely that some satellites' per-round of FL training will be forced to stop due to the disconnection of satellite-ground links. We describe the procedures of this framework in Algorithm 1.

2.3 Challenges of deploying efficient FL in space

As illustrated above, those participating satellites in each round are selected from those *connected satellites* with determined *connection window*, which makes the satellite-ground communication crucial in the FL optimization. However, satellite-ground communication substantially differs from traditional communication on the ground in three aspects.

Limited ground-to-satellite bandwidth. In Earth observation scenario, the communication links from the ground station to satellite are typically designed to transmit control signaling, thus resulting in very low bandwidth. In addition, expanding this ground-to-satellite bandwidth is often highly expensive due to the huge monetary cost and serious heat dissipation when replacing Earth-dial links with satellite communication links [20]–[22]. Therefore, the ground-to-satellite bandwidth is often hundreds of Kbps as shown in Fig. 3(a), although the ground station today supports Gbps satellite-to-ground bandwidth [1], [14], [23]. Our preliminary experiments show that transmitting a ResNet18 [24] model to ground station only costs 0.4 seconds, yet it takes nearly one hour for the aggregated model on ground station to be dispatched to connected satellites. The extreme ground-to-satellite bandwidth breaks down the widely-accepted assumption of symmetric upload/download bandwidth in traditional FL.

Short connection window. Traditional FL on ground runs in a well-connected distributed system with a number of high-performance network infrastructures to guarantee their superior connection conditions. However, due to the orbital dynamics of LEO satellites and limited deployment of ground stations on Earth, the established communication link usually only lasts for a short time. For example, it is often several minutes at a time, and in most cases no more than ten minutes as shown in Fig. 3(b). Our preliminary experiments on ResNet18 model show that 80% of the satellites fail to complete the per-round FL training under these stringent connection windows.

Long connection cycle. Given the frequent failure in one connection window mentioned above, many satellites would attempt to complete the previously failed steps in the next connection window. However, apart from the short connection window, the orbital dynamics of satellites and limited deployment of ground stations also lead to a very long connection cycle of satellite-ground communication links. Fig. 3(c) shows that each satellite would only get several connection periods with ground stations in one day, and it often takes many hours for any two connection periods. Therefore, this makes it infeasible to pick up the failed steps in the next connection window, due to the expired weights with large staleness.

2.4 Limitations of existing solutions

The aforementioned unique challenges have not been thoroughly studied, leaving us with ample room for improvement. This drives us to delve deeper into how to enable satellites in space with efficient intelligent processing capabilities. Based on existing efforts made from various perspectives, we group related work into three categories:

AI on satellites. Several categories of work relate to enabling AI capability of LEO satellites. OEC [7] and Tiansuan constellation [25] proposed to build the orbital edge computing platform to support various computing tasks on satellites so that those collected data can be processed in space. Model inference on satellite [8], [26]–[28] is the key to improving the image process capacity, which can help to efficiently analyze the massive collected Earth observation image. To tackle the challenges of satellite networks, some early research [29], [30] proposed to execute distributed machine learning in space. Their goal is to enable ML model to work better under a limited computing power and communication bandwidth of satellites. However, considering the memory and power cost of inter-satellite communication, cross-satellite distributed ML in space is currently infeasible in LEO satellite constellations.

Federated learning in space. Federated learning [31], [32], a collaborative machine learning paradigm, has been introduced to enable satellites to collaboratively train an ML model with the coordination of ground stations [2], [10], [11], [13], [33], [34]. Most of them focus on tackling the straggler and staleness problems of synchronous or asynchronous FL algorithms, due to the system heterogeneity of satellites [2], [13]. Other research aim to accelerate the slow model convergence caused by limited ground-satellite bandwidth [10], [12], [33], [34]. However, it is still fundamentally challenging to apply existing solutions to this environment when considering all the unique challenges introduced by orbital dynamics of LEO satellites: limited ground-satellite bandwidth, short connection window and long connection cycle. This motivates us to tackle those challenges together to enable communication efficient FL in space.

Model quantization for communication efficient FL. Quantization for FL [35], [36] is an approach that allows several devices to update models using low bitwidth gradients, maintaining accuracy while reducing their communication cost. Most existing neural quantization approaches focus on how to prune the redundant gradient information in the training processing, e.g., replacing the default numerical FP32 gradients with INT8 and even INT4 [37]–[42]. Instead of contributing a novelty accuracy-first FL quantization training algorithm, our goal is to design a generic system to efficiently support in-orbit satellite training under dynamic ground-satellite connection in reality.

In this paper, our goal is to enable satellites with effective intelligent processing capabilities under aforementioned unique challenges. Notably, some new Earth observation satellites have been launched in recent years with ground-satellite bandwidth up to tens of Mbps [43], but the majority of orbiting satellites, particularly those launched earlier, maintain only hundreds of Kbps bandwidth [44]–[47]. Besides, as the scale of a satellite network increases, the actual bandwidth allocation per satellite diminishes. Thus, deploying communication-efficient FL framework across satellites and ground stations to enable satellites with effective intelligent processing capabilities is crucial and meaningful.

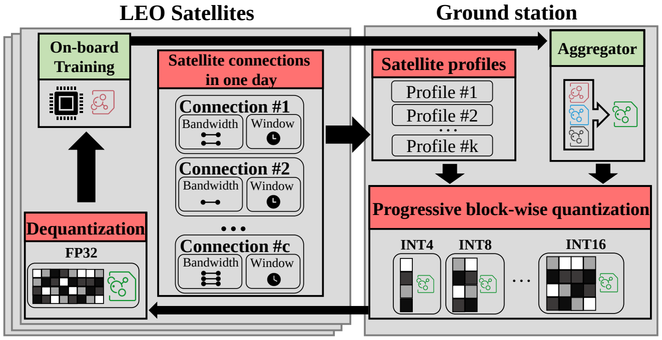


Fig. 4: Overview of our SatelliteFL framework with K satellites and a cluster of ground stations. Each satellite has c connection windows to the ground station in one day.

3 Design of SatelliteFL Framework

This section proposes a satellite-ground FL framework (SatelliteFL) with a progressive block-wise quantization algorithm (ProBQ) to improve communication efficiency during FL training across satellites and ground stations. Its key idea is to ensure that each satellite completes per-round FL training within each connection window by progressive weight quantization. § 3.1 firstly illustrates the overview of SatelliteFL. § 3.2.1 formulates the objective of SatelliteFL as a delay-constrained optimization problem. § 3.2.2 proposes the ProBQ algorithm, which can approximately solve this problem.

3.1 SatelliteFL Overview

We show the detailed illustration of our SatelliteFL framework in Fig. 4. Overall, SatelliteFL also adopts a C/S architecture, where a central server served by ground stations maintains and keeps advancing a global model, and the client refers to a collection of LEO satellites. These satellites dynamically establish communication links with ground stations in the process of orbiting Earth.

Each satellite in SatelliteFL is responsible for three tasks: (1) It dequantizes the received integer (INT) model into 32-bit float (FP32) one; (2) The dequantized FP32 model, along with the local data, are used to obtain an advanced model by on-board training; (3) It maintains the connection information, such as link bandwidth and connection window. The updated FP32 model and connection information are transmitted to the ground stations for global model updating. Considering the relatively fixed satellite orbit and ground station deployment, the daily connection of each satellite is predictable, but the connection's bandwidth and window can only be determined once the link is established. § 3.2 specifically analyzes the impact of these dynamic connections on the satellite-ground FL training.

Ground station serves as two key roles: aggregator and quantizer. The aggregator is responsible for aggregating the received FP32 models from satellites, and obtaining an advanced global model. The quantizer is responsible for quantizing the global model into multiple INT models, which are adaptive to the connection information in the satellite profiles. Its goal is to improve the model utility under the communication constraints, which is formulated as a delay-constrained optimization problem (details in § 3.2.1). Then, it adopts a progressive block-wise quantization algorithm that fine-grained quantizes each block of model with a unique quantization bitwidth to solve this problem (details in § 3.2.2). Finally, these quantized INT models are dispatched to corresponding satellites through the ground-to-satellite link.

3.2 Delay-constrained optimization problem

3.2.1 Problem formulation

Consider a connected satellite participating in this round of FL training, the ground station obtains its connection window τ_d , ground-to-satellite bandwidth $b_{g,s}$ and satellite-to-ground bandwidth $b_{s,g}$. The ground station also maintains a global model ω and launches a round of FL training with three steps. First, it would quantize this model into INT format to accelerate the communication from ground station to satellite, which can be formulated as:

$$\omega(\gamma) = quantize(\omega, \gamma), \gamma \in \mathcal{S}, \tag{4}$$

where γ is the quantization bitwidth to represent the INT format model $\omega(\gamma)$ and $\mathcal{N}$ is a set of available bitwidth specifications for hardware devices on satellite. Then, the quantized model $\omega(\gamma)$ with γ-bit INT format is sent to this satellite through the ground-to-satellite link. After receiving this quantized model, the satellite would dequantize this INT model to obtain dequantized model $\omega'(\gamma)$ with FP32 format, and run on-board training with $\omega'(\gamma)$. Finally, the locally updated model with FP32 format is transmitted to the ground station through the satellite-to-ground link. The original model ω and dequantized model are represented in FP32 format. We denote the data size of model as $|\cdot|$, thus

$$\frac{\gamma}{32} = \frac{|\omega(\gamma)|}{|\omega|},\ and\ |\omega'(\gamma)| = |\omega|. \tag{5}$$

The implication of Eq (5) is that the compression rate of the $\omega(\gamma)$ to the original ω is $\frac{\gamma}{32}$, and the data size of dequantized model $\omega'(\gamma)$ is the same as global model ω. So, a smaller γ attributes to fewer communication data, thus accelerating the model transmission from the ground station to the satellites.

Let t denote the per-round time for anyone connected satellite participating in the FL training to complete the above three steps. We do not consider the computation time of model quantization and dequantization, because it only needs to conduct a few scalar multiplications that are much less complex than local model updates. Therefore, we have the per-round time t as follows.

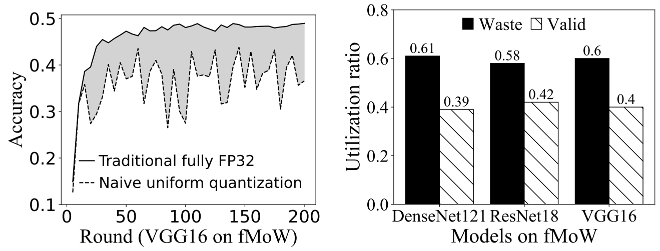
$$t = t_{g,s} + t_c + t_{s,g}, \tag{6}$$

where $t_{g,s}$ is the communication time for $\omega(\gamma)$ to be transmitted from the ground station to the satellite, t_c is the computation time for the model training on satellite, and $t_{s,g}$ is the communication time for updated model based on $\omega'(\gamma)$ to be transmitted from the satellite to the ground station. We calculate the two communication time based on the transmission data size and link bandwidth as follows:

$$t_{g,s} = \frac{|\omega(\gamma)|}{b_{g,s}} = \frac{|\omega|\cdot\gamma}{32\cdot b_{g,s}}, \tag{7}$$

and

$$t_{s,g} = \frac{|\omega'(\gamma)|}{b_{s,g}} = \frac{|\omega|}{b_{s,g}}. \tag{8}$$



(a) Low model accuracy (b) Low bandwidth utilization

Fig. 5: The deficiency of naive uniform quantization on model accuracy and bandwidth utilization.

The computation time t_c is spent updating $\omega'(\gamma)$ by SGD algorithm, which is related to the hardware devices on satellite and ω.

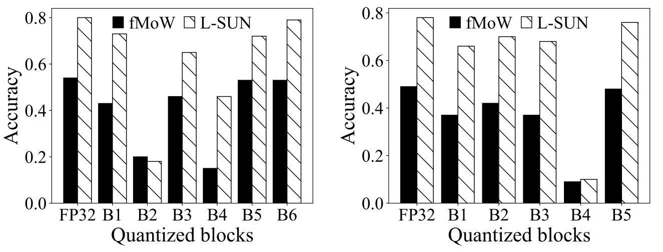
Recall the global objective function $F(\omega)$ in Eq. (1). Our goal is to minimize each $F(\omega)$ in our SatelliteFL with two key considerations: 1) ensuring that per-round FL training does not exceed the connection window τ_d of this satellite in this round; 2) the numerical error between the dequantized $\omega'(\gamma)$ and the original ω without quantization should not exceed the threshold ϵ to ensure the global model accuracy. We introduce *weight divergence* between $\omega'(\gamma)$ and ω to quantify their numerical error: $\frac{|\,||\omega||-||\omega'(\gamma)||\,|}{||\omega||}$. So, we formulate this problem:

$$\begin{aligned}\textbf{P1:}\ \min_{\gamma}\ & \{F(\omega) = \sum_{k\in\mathcal{S}} f_k(\omega'(\gamma))\} \\ s.t.\quad & \frac{|\omega|\cdot\gamma}{32\cdot b_{g,s}} + t_c + \frac{|\omega|}{b_{s,g}} \le \tau_d, \\ & \frac{|\,||\omega|| - ||\omega'(\gamma)||\,|}{||\omega||} \le \epsilon, \\ & \gamma \in \mathcal{N}.\end{aligned}$$

Solving P1 is challenging for the following reasons. First, a smaller γ attributes to fewer communication data, thus accelerating the transmission across ground-to-satellite. However, a larger γ is required to represent more information of original ω, thus reducing the numerical error introduced by quantization. Second, it is generally impossible to obtain an exact analytical relationship to connect local objective function f with the dequantized $\omega'(\gamma)$.

3.2.2 Analysis of *P1*

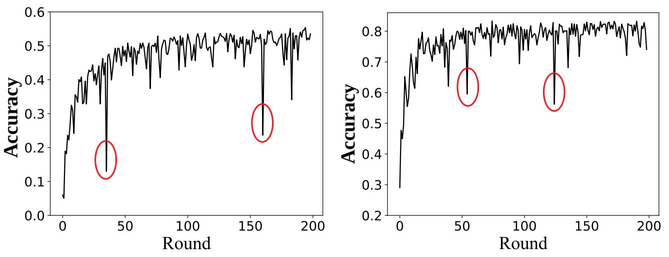
Low bitwidth quantization, while transmitting-friendly, severely limits the expressiveness of the updated information. Updating the local parameters on satellite with such limited information thus may not improve the quality of local models, even making them worse. We first analyze the



(a) DenseNet121

(b) ResNet18

Fig. 6: Comparison of traditional fully FP32 blocks (FP32) and only one quantized block with INT4 bitwidth representation (B1-B6: Block1-Block6).



(a) fMoW

(b) L-SUN

Fig. 7: Hybrid bitwidth quantization across FL training round with DenseNet121 on two datasets. The deep V regions marked in red represent the lower accuracy when using fully INT4 quantization in this round.

deficiency of naive uniform quantization to solve **P1**. Then, we dive into the sensitivity of the bitwidth of quantization to model structure, and translate the above problem to **P2**.

Naive uniform quantization is to decide a unique γ to quantize all the weights of global FP32 model, which aims to ensure that each satellite must complete per-round training within each connection window. Based on the estimation of computation time and communication time, we choose as large bitwidth as possible to quantize model, while satisfying the constraint of connection window. However, this scheme leads to a serious deficiency in the model accuracy and bandwidth utilization as shown in Fig. 5. The reason behind is that limited decision space on γ makes it difficult to achieve a complex balance between the constraints of connection window and quantization error.

Therefore, we dive into the impact of model architecture on accuracy performance when deciding different quantized bitwidth γ. Since the classical model architectures tend to have several blocks, conceptualized as analogous to the ventral visual blocks [48]. We then conduct a comprehensive measurement on each block with different low bitwidth, separately. We obtain two key observations from these experimental results that motivate our further design. The settings of measurements are consistent with § 4.1.

• **Observation 1: The impact on the accuracy varies greatly when quantizing different blocks with low bitwidth, blocks in the middle layer are less suitable for low bitwidth quantization.** Fig. 6 shows the end-to-end accuracy performance when using 4-bit quantization in each single block, while other blocks in this model use FP32 representation. Most of the blocks suffer accuracy degradation when using low bitwidth quantization. For example, block2, block3, and block4 of DenseNet121 cause up to a 35%-67% accuracy loss, while there is less than 6% accuracy loss when quantizing block1 and block6. Therefore, it is worth that we can improve the efficiency of model accuracy and bandwidth utilization (in Fig. 5) by cherry-picking suitable blocks for low bitwidth quantization.

• **Observation 2: High bitwidth quantization on blocks can make up for the accuracy loss caused by low bitwidth quantization quickly.** Fig. 7 shows the accuracy performance when using hybrid bitwidth quantization across FL round on all blocks. Note that the four marked regions denoting the lower obtained accuracy with full INT4 quantization. The lower accuracy immediately reverts to a higher level in the next few rounds when turning to INT8 (or INT16, INT32) quantization. The behind reason is that a higher bitwidth representation retains more correct gradient information than a lower one, which can revise the incorrect updating direction and mitigate the accuracy degradation. This motivates us to introduce hybrid bitwidth quantization among different blocks for efficiency improvements during the FL training.

• ***Implications.*** *In summary, low bitwidth quantization as a strategy to reduce model parameters, make an obvious accuracy degradation. However, some specific blocks will not suffer this serious performance loss, and the high bitwidth quantization can mitigate the accuracy degradation caused by the low bitwidth quantization. To enable a practical scheme with tolerable accuracy degradation and full link utilization, the quantization paradigm needs to be re-architected.*

Therefore, we introduce the block-wise decision of γ to **P1**. For anyone model ω, we have m blocks:

$$\omega = \{\omega_1, \omega_2, \cdots, \omega_m\}, \tag{9}$$

where ω_i denotes a block of ω and $i \in [1, 2, \cdots, m]$. For anyone block $\omega_i \in \omega$, we decide a unique bitwidth γ_i to quantize this block with FP32 format into INT format with γ_i bit representation. Then, we get a new decision vector Γ:

$$\Gamma = \{(\omega_1, \gamma_1), (\omega_2, \gamma_2), \cdots, (\omega_m, \gamma_m)\}. \tag{10}$$

Consider the new quantization decision on ω, we recalculate the communication time $t_{g,s}$ from ground station to satellite:

$$t_{g,s} = \frac{\sum_{i=1}^{m} |\omega_i(\gamma_i)|}{b_{g,s}} = \frac{\sum_{i=1}^{m} |\omega_i| \cdot \gamma_i}{32 \cdot b_{g,s}}, \tag{11}$$

where $\omega_i(\gamma_i)$ denotes the quantized block ω_i with bitwidth decision γ_i. We also dequantize block $\omega_i(\gamma_i)$ ($i \in [1, 2, \cdots, m]$) one by one, and organize them to obtain the dequantized model $\omega'(\Gamma)$ with FP32 format.

Finally, we translate **P1** into **P2** as:

$$\begin{aligned} \textbf{P2:} \min_{\Gamma} \ & \{F(\omega) = \sum_{k \in \mathcal{S}} f_k(\omega'(\Gamma))\} \\ s.t. \ & \frac{\sum_{i=1}^{m} |\omega_i| \cdot \gamma_i}{32 \cdot b_{g,s}} \leq \tau_d - t_c - t_{s,g}, \\ & \frac{|\,||\omega|| - ||\omega'(\Gamma)||\,|}{||\omega||} \leq \epsilon, \\ & \forall \gamma_i \in \Gamma, \gamma_i \in \mathcal{N}. \end{aligned}$$

Algorithm 2: Satellite-ground FL with ProBQ design

```
Input  : initialized global model $\omega_0(\gamma)$ and sorted
         blocks $\{\omega_1, \omega_2, \cdots, \omega_m\}$ based on the impact
         degree, decision of bitwidth
         $\gamma \in \mathcal{N} = \{b_1, b_2, \cdots, b_n\}$, ground-to-satellite
         bandwidth $b_{g,s}$, set of total satellites $\mathcal{S}$ and
         total training rounds $R$.
Output: the global model $\omega_r$.
1  Ground station executes: // global coordination
2      for $r \leftarrow 0$ to $R$ do
           // per-round of FL training
3          $\mathcal{S}_c, \{\tau_d\} \leftarrow$ connected satellites and their duration
4          Sort $\mathcal{S}_c$ by its connection duration $\tau_d$ in reverse
5          $\mathcal{S}_s \leftarrow$ select top-$K_s$ satellites from $\mathcal{S}_c$
6          for each satellite $k \in \mathcal{S}_s$ with duration $\tau_d$ do
7              $t_c, t_{s,g} \leftarrow$ estimated training and download time
8              $\Gamma \leftarrow$ ProBQ($\tau_d, t_c, t_{s,g}, b_{g,s}$)
9              Quantize $\omega_r(\gamma)$ to INT model $\omega_r^k(int)$ as $\Gamma$
10             Dispatch the $\omega_r^k(int)$ to satellite $k$
               $\omega_{r+1}^k \leftarrow$SatelliteUpdate($\omega_r^k(int)$)
11         Update global model $\omega_{r+1}$ as Eq. (3)
12 ProBQ($\tau_d, t_c, t_{s,g}, b_{g,s}$): // search a feasible decision
13     Default decision
       $\Gamma = \{(\omega_1, b_1), (\omega_2, b_1), \cdots, (\omega_m, b_1)\}$
14     for $i \leftarrow 1$ to $m$ do
15         if $t_{g,s} = \frac{\sum_{i=1}^{m} |\omega_i| \cdot \gamma_i}{32 \cdot b_{g,s}} > \tau_d - t_c - t_{s,g}$ then
16             for $j \leftarrow b_1$ to $b_n$ do
17                 $b_j = b_{j+1}$
18             Set $\gamma_i = b_j$ for $\omega_i$ and update $\Gamma$
19         else
20             return $\Gamma$
21 SatelliteUpdate($\omega_r^k(int)$): // local training
22     Dequantize $\omega_r^k(int)$ to FP32 model $\omega_r^k$
23     $\omega_{r+1}^k \leftarrow$ on-board training to update $\omega_r^k$ as Eq. (2)
24     return the updated FP32 model $\omega_{r+1}^k$
```

In problem **P2**, the objective is to minimize the each $F(\omega)$ in our SatelliteFL with two key considerations. Note that this $F(\omega)$ is controlled by the decision vector Γ in each satellite. One of the key considerations is how to ensure that per-round FL training does not exceed the transmission time $\tau_d - t_c - t_{s,g}$ of this satellite in current round, and the per-round FL training time is determined by the current decision vector Γ in each satellite model. The other key consideration is how to ensure that the numerical error between the dequantized model and original model should not exceed the threshold ϵ. Therefore, problem **P2** is a nonlinear integer dynamic optimization problem [49], [50]. To solve **P2**, we need to find out how the value of Γ affects the loss function F and the *weight divergence* $\frac{||\,||\omega|| - ||\omega'(\gamma)||\,||}{||\omega||}$. However, it has been proved that problem **P2** is a classical NP-hard problem [51] over Γ, and in reality, algorithm with reduced computation complexity is required to solve **P2**. Meanwhile, considering the relatively limited integer bitwidth supported by satellite hardware, the greedy strategy can be used to solve it well. Thus, we adopt the greedy idea to obtain a suboptimal solution in designing Algorithm 2.

Intuitively, as long as the bitwidth of quantization is larger, the practical training loss would be closer to the original FP32-based loss and the quantization error would be smaller. Therefore, the key is how to choose the maximum quantization bitwidth. As long as connection window is guaranteed by this decided bit width, the model utility can be improved to the maximum. And the available bitwidth specifications for hardware devices on satellite are very limited (like INT2, INT4, INT8, etc.), which means the decision space of γ is also small. Therefore, we can adopt a greedy algorithm to solve this problem in a short clock time.

3.3 Progressive Block-wise Quantization Algorithm

In this section, we propose a progressive block-wise quantization algorithm (ProBQ) that quantizes each block of a model with a suitable bitwidth based on the above observations. This novel design compensates for the quantization error as much as possible on the premise of guaranteeing the model update in each transmission process. Its core technical designs are two-fold: (1) adaptive quantization on fine-grained blocks to achieve a better trade-off between model utility and delay constraints; (2) FP32-guided dequantization on low bitwidth model to enable high precision on-board training. Algorithm 2 describes the workflow of our SatelliteFL with proposed ProBQ algorithm.

We first introduce two key variables in our decision vector to help describe the algorithm: $\mathcal{N}$ and ω. Consider a finite space with n positive integers $\mathcal{N} = \{b_1, b_2, \cdots, b_n\}$ that represents a set of available bitwidth specifications for hardware devices on satellite. In most cases, it supports only several types of bitwidth. (like INT2, INT4, INT8, etc.) We sort this set by the number of bitwidth in reverse, which means that the first element of $\mathcal{N}$ is the maximum bitwidth to quantize each block. Consider a general ML model ω with m blocks. Based on the prior profile information that the impact degree on the model accuracy when quantizing different blocks with a low bitwidth as shown in Fig. 6 and Fig. 7. We sort model's all blocks as this impact degree from smallest to largest to obtain a sorted list of blocks: ω = $\{\omega_1, \omega_2, \cdots, \omega_m\}$.

Then, under the satellite-ground cooperative mechanism in this framework, we focus on the introduction of the ProBQ algorithm. To reduce the failure of satellites' FL training in one round, we select K_s satellites with the longest connection time from the *connected satellites* in each round (Line 4-5). During each satellite's decision phase, we first obtain the transmission time for the quantized INT model in this *connected duration* by the estimation (Line 7), then we adopt a greedy method to search the feasible solution with the constrained available time $\tau_d - t_c - t_{s,g}$ (Line 13-21). Although this method's complexity is very high, the practical execution speed is not slow due to the small number of available satellites, bitwidth supported by the satellites, and the blocks of the model. With the obtained decision vector Γ, global model is quantified into INT format and dispatched to the corresponding satellite. Finally, in order to ensure the training accuracy, each satellite dequantizes the received INT model into FP32 model for on-board training (Line 23-25).

TABLE 1: Datasets used in experiments for one task: image classification. IC (image classification), OD (object detection), IS (instance segmentation).

Dataset	Task	Size	Samples	Labels
fMoW[52]	IC, OD, IS	resize 224*224	1 million	63
L-SUN[53]	IC	256*256	59 million	30

4 Performance Evaluation

4.1 Experimental Settings

4.1.1 Datasets and models

We adopt a real-world satellite imagery dataset and an open-source image dataset as shown in Table 1: Functional Map of the World (fMoW) [52] and L-SUN [53] on 3 classic CNN models: DenseNet121 [54], VGG16 [55] and ResNet18 [24]. These three models are popular for image classification, which contains 6, 6, and 5 blocks respectively.

- fMoW is a classic world functional map dataset, which contains more than 1 million high-resolution images with a total of 63 categories. We generate a sub-dataset by random sampling the data for each class, setting 0.2 as the fraction of data to be sampled. In sub-dataset, each high-resolution image is resize to 224*224.
- L-SUN is a classic image classification dataset, which contains approximately 59 million images with 10 scene categories and 20 object categories, and the fraction of data to be sampled is set as 0.01.

4.1.2 Simulated platform

We have implemented SatelliteFL on a simulated platform atop FedML [56], a popular FL framework with real-world communication and computation simulation. We also followed the prior work [2] to plug all the configurations as an example constellation, named Planet Lab with 12 ground stations and 191 satellites. We follow the prior work [57]–[59] to divide the two datasets into non-iid sub-dataset, and assigned them to 191 satellites. Here, we use the cote simulator to obtain the connection information same as FedSpace [2]. Specifically, to establish a clear correspondence between training rounds and the clock time, we set per training round period with τ = 15 minutes, and there are 96 training rounds in one day. The connected satellites with their connection information in each round are also obtained from this simulation platform. Moreover, to simulate the dynamic variations in the satellite-to-ground link, we attempted to model the ground-to-satellite bandwidth using a Gaussian distribution with a mean of $b_{g,s}$ and a standard deviation of $b_{g,s}/3$. All the experiments are conducted on a Ubuntu 18.04 Linux server with 8 NVIDIA A40 GPUs.

4.1.3 Metrics

Apart from the convergence accuracy of the testing data, we also report the following two metrics that closely relate to the satellites. (i) *Clock time* is the end-to-end training time perceived by the satellites, including multiple rounds of per-round FL training until model convergence. (ii) *Bandwidth utilization* of ground-to-satellite link refers to the ratio of the valid data size to the total data size that can be transmitted by the ground station during each connection duration, denoted as u_ratio. The total data size can be calculated as $b_{g,s} \cdot \tau_d$, which means if there is enough data required to be transmitted to the satellite throughout the connection duration τ_d with the ground-to-satellite bandwidth $b_{g,s}$. The valid data size here refers to the size of actual transmitted model parameters and it must be a complete one with all parameters, because part of parameters cannot be used for the satellite's on-board training. So we have:

$$u\_ratio = \frac{\alpha(|\omega|)}{b_{g,s} \cdot \tau_d},\ and\ \alpha(|\omega|) = \begin{cases} |\omega|, & success \\ 0, & fail. \end{cases} \tag{12}$$

We fix the sum of computation time t_c and communication time $t_{s.g}$ as 3 minutes based on the profiled measurement on Jetson TX2 and 1 Gbps satellite-to-ground bandwidth. Note that the average connection window is 6 minutes as shown in Fig. 3(b), and the maximum u_ratio is about 50% on average.

4.1.4 Baselines

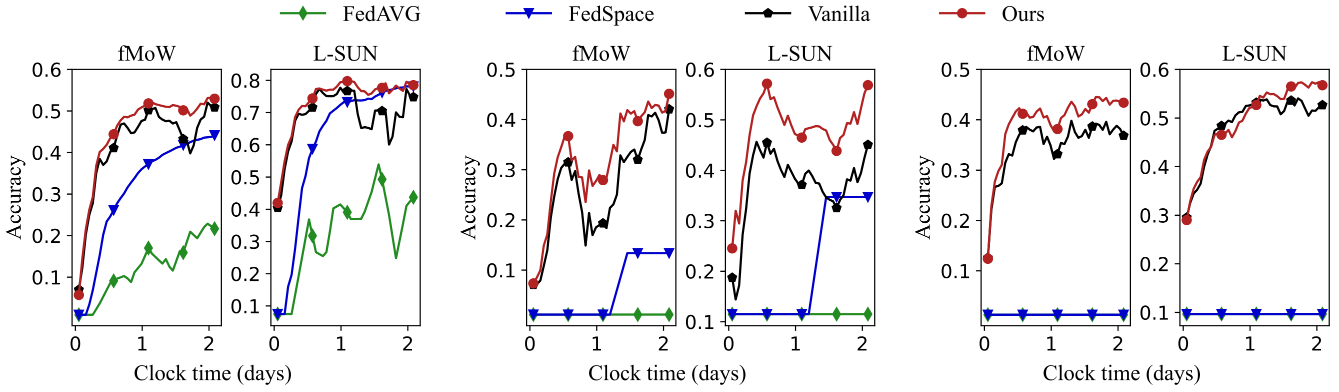
We use three baselines in experiments to demonstrate SatelliteFL's benefits are: (i) FedAvg [16]: the traditional FL protocol using fully FP32 format on the ground. (ii) Vanilla SatelliteFL (vanilla): using naive uniform bitwidth quantization in satellite-ground FL framework as described in Analysis of **P1** of § 3.2. (iii) FedSpace [2]: a state-of-the-art satellite-ground FL framework using an adaptive buffer to balance the synchronous and asynchronous training phases.

4.2 Experimental Results

4.2.1 End-to-end Performance

We show the overall results of SatelliteFL on the two metrics compared to baselines when using $b_{g,s}$=400 as the default setting. Fig. 8 shows the time-to-accuracy performance of SatelliteFL compared to three baselines, which demonstrates that our SatelliteFL framework with ProBQ algorithm greatly improves both the convergence speed and the model accuracy. Fig. 9 shows the performance of bandwidth utilization ratio, which reveals that it also highly improves the bandwidth utilization of the ground-to-satellite link.

As shown in Fig. 8, SatelliteFL greatly improves both the convergence speed and model accuracy compared with baselines, Compared to FedAvg with DenseNet121 on fMoW and L-SUN, SatelliteFL obtains 31.6% and 36.9% higher convergence accuracy, and it takes 42.1% and 44.2% fewer days to converge. Moreover, for ResNet18 and VGG16, FedAvg's accuracy is even less than 10% without any useful learned information. And FedSpace achieves the same result when using VGG16 model. This is because they are not designed for transmission delay-constrained scenarios, but only transmit current update information as possible, thus the update information is likely to be interrupted. Compared to FedSpace with DenseNet121 on fMoW and L-SUN, SatelliteFL obtains 10.7% and 1.8% higher convergence accuracy, and it takes 45.8% and 47.5% fewer days to converge. But for ResNet18 on these two datasets, FedSpace only achieves 13.3% and 34.7%, which is 33.8%



(a) DenseNet121

(b) ResNet18

(c) VGG16

Fig. 8: The time-to-accuracy performance of our SatelliteFL with ProBQ (ours) compared with three baselines: FedAvg, Vanilla SatelliteFL (vanilla) and FedSpace.

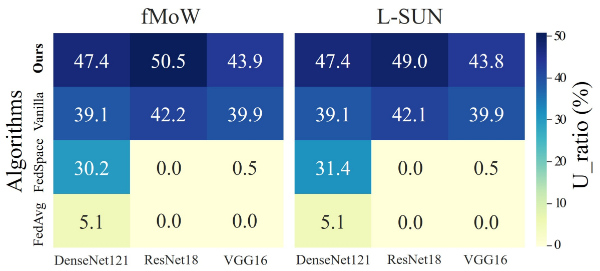
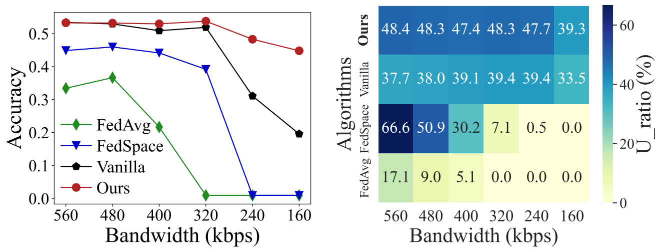


Fig. 9: The utilization ratio (u_ratio) of ground-to-satellite link bandwidth when comparing our SatelliteFL with ProBQ (ours) and three baseline algorithms using three models on two datasets. Considering the on-board training time and data transmission time across satellite-to-ground link, the maximum u_ratio is only about 50% on average (details in § 4.1).

and 23.4% lower than SatelliteFL, respectively. Vanilla SatelliteFL with uniform quantization method achieves much better performance than the other two baselines, yet, compared with our SatelliteFL with ProBQ, its time-to-accuracy performance still has obvious shortcomings. For example, using ResNet18 on fMoW and L-SUN, ours accuracy is 3.4% and 12.1% higher than Vanilla, respectively. Even on the other two models, it can achieve up to 1.7%-7.4% accuracy improvement. The reason is that ProBQ can retain more effective gradient information when the ground-to-satellite bandwidth fluctuates and even decreases. Such improvements are mostly attributed to the progressive quantization design as described in § 3.2.2.

Apart from the time-to-accuracy performance, we also evaluate the utilization ratio of ground-to-satellite link bandwidth in Fig. 9. The results show that the u_ratios of FedAvg and FedSpace are very low, almost close to 0, due to the frequent failure of ensuring the per-round FL training with one connection window. Among these two algorithms, FedSpace respectively achieves 30.2% and 31.4% with DenseNet121 on fMoW and L-SUN, but they are still 17.2% and 16% lower than our SatelliteFL. Our SatelliteFL with ProBQ can also improve the u_ratio (3.9%-8.3%) of Vanilla SatelliteFL using naive uniform quantization method. For example, using ResNet18 on fMoW, our SatelliteFL with ProBQ achieves 50.5% u_ratio, which is 8.3% higher than Vanilla SatelliteFL. The reason behind is that ProBQ designed in our SatelliteFL has progressively expanded the quantization bitwidth on the most of blocks, which can improve the system efficiency under delay constraints.



(a) Accuracy.

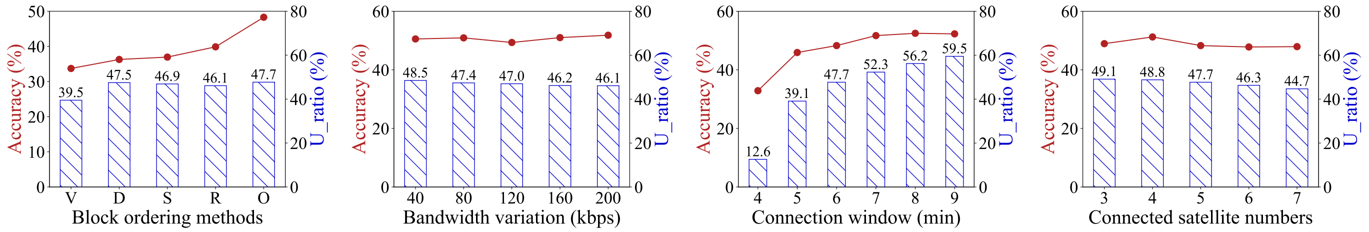
(b) U_ratio.

Fig. 10: The performance of our SatelliteFL with ProBQ (ours) compared with three baselines under varying bandwidth.

4.2.2 Link Analysis

We further study the performance of SatelliteFL under different ground-to-satellite bandwidth compared to baselines. We keep the hyper-parameter settings the same as the above end-to-end settings. Fig. 10 shows the accuracy and u_ratio of different methods under varying ground-to-satellite bandwidth. We set $b_{g,s}$= 160, 240, 320, 400, 480, and 560, and train DenseNet121 on fMoW in this experiment.

As observed, ProBQ only reduces accuracy performance by 9.1%-15.6% under poor bandwidth conditions, whereas the other three methods decrease accuracy by 19.8%-30.3%, 6.8%-44.1%, and 14.9%-36.5%, respectively. The rationale is that, compared to FedAvg and FedSpace, ProBQ strives to preserve effective update information as much as possible under limited bandwidth conditions. In contrast to Vanilla SatelliteFL's fixed quantization, ProBQ's progressive block



(a) Different block ordering. (b) Different bandwidth variation. (c) Different connection window. (d) Different connected numbers.

Fig. 11: Impacts of SatelliteFL settings. V (Vanilla). D (Default): order of original front-to-back. S (Size): order of parameter size. R (Random): order of random shuffling. O (Ours).

quantization allows sensitive blocks with lower bitwidth to retain more useful update information. More specifically, with lower bandwidth conditions such as $b_{g,s}$= 160 and 240, ProBQ reduces only 18.6% and 1.2% u_ratio, whereas FedAvg and FedSpace struggle to keep the system operational with an effective u_ratio. This is because ProBQ permits the retention of more fragmented update information and transmits it at a block-wise granularity. In addition, with $b_{g,s} > 240$, ProBQ can maintain a stable u_ratio close to 50% with consistent accuracy performance. Vanilla SatelliteFL also can keep a stable u_ratio close to 40%, while the other two baselines experience a sharp decline in u_ratio as bandwidth becomes worse. The rationale is that, within a relatively ample bandwidth range, each block can retain sufficient useful information for updates. In contrast, in more constrained bandwidth conditions, ensuring the availability of enough update information for the entire model becomes challenging.

4.2.3 Sensitivity Analysis

We then dive into the performance sensitivity of SatelliteFL under varying environments. Specifically, to further explore SatelliteFL's performance under poor bandwidth conditions, we set $b_{g,s}$=240 according to the observation drawn from the above link analysis discussed, while keeping the remaining hyper-parameters consistent with the end-to-end configuration.

Impacts of block ordering methods. Fig. 11(a) reports the impact of block ordering methods in our SatelliteFL on the performance of the above two metrics. We introduce another three kinds of ordering methods into Vanilla SatelliteFL: (i) The default method (Default) orders the blocks based on the original front-to-back order; (ii) The size method (Size) orders the blocks based on the order of parameter size; (iii) The random method (Random) orders the blocks based on a random shuffle of all the blocks. We set $b_{g,s}$=240 and train DenseNet121 on fMoW in this experiment.

The results in Fig. 11(a) show the compared performance to ProBQ ("Ours"), Vanilla, Default, Size, and Random algorithms. For the accuracy performance, ProBQ still outperforms other algorithms 8.4%-14.6%, this is because ProBQ can achieve a near-optimal choice on block priority under poor bandwidth, to give priority guarantee on the sensitive blocks remaining information with a higher bitwidth. Note that, whatever the block priority we choose in SatelliteFL, the last four methods with block-wise quantization can obtain 6.7% (on average) higher accuracy and 19.1% (on average) higher u_ratio. The reason for such performance improvement is that our block-wise strategy can make more efficient use of redundant fragmented ground-to-satellite links under poor bandwidth.

Impacts of bandwidth variation. Fig. 11(b) shows the accuracy and u_ratio on different settings of bandwidth variation. We set $b_{g,s}$= 240 and change the standard deviation = 40, 80, 120, 160, and 200 in Gaussian distribution to simulate the dynamic fluctuations of ground-to-satellite bandwidth, and then train DenseNet121 on fMoW in this experiment.

As observed, Fig. 11(b) illustrates the performance of SatelliteFL under varying degrees of bandwidth fluctuations. With an increase in the degree of bandwidth variation, the u_ratio decreased by 1.1%-2.4%, while the accuracy remained nearly constant, fluctuating within approximately 1%. The underlying reason for this phenomenon is that ProBQ can effectively adjust the different bitwidth for parameter quantization under varying degrees of dynamic bandwidth changing, thereby maintaining model accuracy as stably as possible.

Impacts of connection window. Fig. 11(c) shows the accuracy and u_ratio on different settings of connection window. We set $b_{g,s}$= 240 and train DenseNet121 on fMoW with connection window = 4, 5, 6, 7, 8, and 9 minutes in this experiment.

The results in Fig. 11(c) show the benefit of increasing the connection window time. It shows that SatelliteFL can improve accuracy by 39.5%-59.6% and 2.1×-3.7× u_ratio, which is due to ProBQ being able to retain more update information and benefit from a richer connection window time. Note that, when the connection window time $>$ 6, the u_ratio exceeds 50%, this is because ProBQ has more transmission time to transmit gradients across the ground-to-satellite link and maintain a stable accuracy performance.

Impacts of connected satellite numbers. Fig. 11(d) shows the accuracy and u_ratio on different settings of connected satellites numbers. We set $b_{g,s}$= 240 and train DenseNet121 on fMoW with connected satellites number = 3, 4, 5, 6, and 7 in this experiment.

Fig. 11(d) illustrates the performance on SatelliteFL for different numbers of connected satellites. As the number of connected satellites increases, the u_ratio abnormally decreases by 0.6%-8.9%, the accuracy first increases by 5.1%, and then also abnormally decreases by 0.9%-1.9%. The reason behind the abnormal phenomenon is that when connected satellite numbers increase, there is a greater probability of selecting satellites with worse link conditions, and these satellites have a poor u_ratio, which also affects the final accuracy.

5 Discussion and Limitation

In our research, we build upon the foundational work presented in prior work [2], which assumes the existence of a steady (with some dynamic changes) communication link during each connection window. However, it is important to acknowledge that in real-world satellite-ground communication scenarios, achieving such steadiness can be challenging due to varying distance between the satellite and the ground station or atmospheric conditions like rainfall-induced signal attenuation. The presence of an uncertain communication link can pose difficulties in meeting our objective of completing the per-round FL training within each connection window.

In our future work endeavors, we aim to enhance the success rate of parameter transmission by developing predictive models for link uncertainty. Additionally, we intend to explore redundant training strategies and missing-parameter recovery mechanisms, minimizing the need for re-transmissions following parameter transmission failures based on the concept of bounded-loss tolerance commonly applied in ML tasks [60]. Furthermore, the images collected by Earth observation satellites lack labels, posing a challenge in efficiently utilizing this raw data. We intend to explore effective methods for providing soft labels to these raw data, focusing on unsupervised training [61] and leveraging the few-shot capabilities exhibited by current large models [62].

6 Conclusion

SatelliteFL is a communication efficient FL framework across satellites and ground stations that focuses on three unique challenges introduced by the dynamic orbits of LEO satellites: limited ground-to-satellite bandwidth, short connection window and long connection cycle. It formulates the goal as a delay-constrained optimization problem to maximize the model utility while guaranteeing not to exceed the connection window. To solve this problem, it leverages a progressive weight quantization method that fine-grained quantizes each block of model with a unique quantization bitwidth. Experiments show that SatelliteFL can accelerate the convergence by 1.8×(on average, up to 2.8×) and improve the bandwidth utilization ratio by 4.5×(on average, up to 9.3×) with acceptable accuracy loss.

Acknowledgement

This work was supported by National Key R&D Program of China (No.2021ZD0113001), Shenzhen Science and Technology Innovation Commission Free Exploring Basic Research Proiect under Grant (No.2021Szvup008), NSFC (U21B2016, 62032003).

References

[1] Z. Lai, Q. Wu, H. Li, M. Lv, and J. Wu, "Orbitcast: Exploiting mega-constellations for low-latency earth observation," in *Proceedings of International Conference on Network Protocols*, 2021, pp. 1–12.

[2] J. So, K. Hsieh, B. Arzani, S. Noghabi, S. Avestimehr, and R. Chandra, "Fedspace: An efficient federated learning framework at satellites and ground stations," *arXiv preprint arXiv:2202.01267*, 2022.

[3] S. Yu, X. Gong, Q. Shi, X. Wang, and X. Chen, "Ec-sagins: Edge-computing-enhanced space-air-ground-integrated networks for internet of vehicles," *IEEE Internet of Things Journal*, pp. 5742–5754, 2021.

[4] *Remote sensing from space: What norms govern?* https://www.planet.com/our-constellations/, 2023.

[5] *Black summer*, https://www.abc.net.au/news/science/2021-03-14/bushfires-detecting-them-from-space-fireball-satellite-launch/13203470, 2021.

[6] *Remote sensing from space: What norms govern?* https://www.justsecurity.org/86114/remote-sensing-from-space-what-norms-govern/, 2023.

[7] B. Denby and B. Lucia, "Orbital edge computing: Nanosatellite constellations as a new class of computer system," in *Proceedings of International Conference on Architectural Support for Programming Languages and Operating Systems*, 2020, pp. 939–954.

[8] A. Chen, Y. Xie, Y. Wang, and L. Li, "Knowledge graph-based image recognition transfer learning method for on-orbit service manipulation," *Space: Science & Technology*, 2021.

[9] M. Handley, "Using ground relays for low-latency wide-area routing in megaconstellations," in *Proceedings of Workshop on Hot Topics in Networks*, 2019, pp. 125–132.

[10] N. Razmi, B. Matthiesen, A. Dekorsy, and P. Popovski, "Ground-assisted federated learning in leo satellite constellations," *IEEE Wireless Communications Letters*, pp. 717–721, 2022.

[11] N. Razmi, B. Matthiesen, A. Dekorsy, and P. Popovski, "On-board federated learning for dense leo constellations," in *Proceedings of International Conference on Communications*, 2022, pp. 4715–4720.

[12] F. Tang, C. Wen, X. Chen, and N. Kato, "Federated learning for intelligent transmission with space-air-ground integrated network sagin toward 6g," *IEEE Network*, 2022.

[13] J. Nguyen, K. Malik, H. Zhan, *et al.*, "Federated learning with buffered asynchronous aggregation," in *Proceedings of International Conference on Artificial Intelligence and Statistics*, 2022, pp. 3581–3607.

[14] D. Vasisht and R. Chandra, "L2d2: Low latency distributed downlink for leo satellites," in *Proceedings of Special Interest Group on Data Communication*, 2021, pp. 151–164.

[15] D. Bhattacherjee, S. Kassing, M. Licciardello, and A. Singla, "In-orbit computing: An outlandish thought experiment?" In *Proceedings of Workshop on Hot Topics in Networks*, 2020, pp. 197–204.

[16] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of

deep networks from decentralized data," in *Artificial intelligence and statistics*, 2017, pp. 1273–1282.

[17] C. Foster, J. Mason, V. Vittaldev, *et al.*, "Constellation phasing with differential drag on planet labs satellites," *Journal of Spacecraft and Rockets*, pp. 473–483, 2018.

[18] V. Ignatenko, P. Laurila, A. Radius, O. Antropov, and D. Muff, "Iceye microsatellite sar constellation status update: Evaluation of first commercial imaging modes," in *Proceedings of International Geoscience and Remote Sensing Symposium*, 2020, pp. 3581–3584.

[19] M. Safyan, "Planet's dove satellite constellation," *Handbook of Small Satellites: Technology, Design, Manufacture, Applications, Economics and Regulation*, pp. 1–17, 2020.

[20] I. Del Portillo, B. G. Cameron, and E. F. Crawley, "A technical comparison of three low earth orbit satellite constellation systems to provide global broadband," *Acta astronautica*, pp. 123–135, 2019.

[21] N. U. Hassan, C. Huang, C. Yuen, A. Ahmad, and Y. Zhang, "Dense small satellite networks for modern terrestrial communication systems: Benefits, infrastructure, and technologies," *IEEE Wireless Communications*, pp. 96–103, 2020.

[22] Y. Su, Y. Liu, Y. Zhou, J. Yuan, H. Cao, and J. Shi, "Broadband leo satellite communications: Architectures and key technologies," *IEEE Wireless Communications*, pp. 55–61, 2019.

[23] K. Devaraj, M. Ligon, E. Blossom, *et al.*, "Planet high speed radio: Crossing gbps from a 3u cubesat," 2019.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of Conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] S. Wang, Q. Li, M. Xu, X. Ma, A. Zhou, and Q. Sun, "Tiansuan constellation: An open research platform," in *Proceedings of International Conference on Edge Computing*, 2021, pp. 94–101.

[26] B. Denby and B. Lucia, "Orbital edge computing: Machine inference in space," *IEEE Computer Architecture Letters*, pp. 59–62, 2019.

[27] S. Wang, Q. Zhang, R. Xing, F. Qi, and M. Xu, "The first verification test of space-ground collaborative intelligence via cloud-native satellites," *arXiv preprint arXiv:2311.06078*, 2023.

[28] Y. Chen, Q. Zhang, Y. Zhang, X. Ma, and A. Zhou, "Energy and time-aware inference offloading for dnn-based applications in leo satellites," in *Proceedings of International Conference on Network Protocols*, IEEE, 2023, pp. 1–6.

[29] H. Guo, Q. Yang, H. Wang, *et al.*, "Spacedml: Enabling distributed machine learning in space information networks," *IEEE Network*, pp. 82–87, 2021.

[30] H. Li, C. Chen, C. Li, L. Liu, and G. Gui, "Aerial computing offloading by distributed deep learning in collaborative satellite-terrestrial networks," in *Proceedings of International Conference on Wireless Communications and Signal Processing*, 2021, pp. 1–6.

[31] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *CoRR*, vol. abs/1610.05492, 2016.

[32] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, vol. 54, 2017, pp. 1273–1282.

[33] B. Matthiesen, N. Razmi, I. Leyva-Mayorga, A. Dekorsy, and P. Popovski, "Federated learning in satellite constellations," *arXiv preprint arXiv:2206.00307*, 2022.

[34] H. Chen, M. Xiao, and Z. Pang, "Satellite-based computing networks with federated learning," *IEEE Wireless Communications*, pp. 78–84, 2022.

[35] Z. Yan, D. Li, X. Yu, and Z. Zhang, "Latency-efficient wireless federated learning with quantization and scheduling," *IEEE Communications Letters*, vol. 26, no. 11, pp. 2621–2625, 2022.

[36] P. Prakash, J. Ding, R. Chen, *et al.*, "Iot device friendly and communication-efficient federated learning via joint model pruning and quantization," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13 638–13 650, 2022.

[37] D. Xu, M. Xu, Q. Wang, *et al.*, "Mandheling: Mixed-precision on-device dnn training with dsp offloading," in *Proceedings of International Conference on Mobile Computing And Networking*, 2022, pp. 214–227.

[38] J. Yoon, G. Park, W. Jeong, and S. J. Hwang, "Bitwidth heterogeneous federated learning with progressive weight dequantization," *arXiv preprint arXiv:2202.11453*, 2022.

[39] Q. Zhou, S. Guo, Z. Qu, *et al.*, "Octo:{int8} training with loss-aware compensation and backward quantization for tiny on-device learning," in *Proceedings of Annual Technical Conference*, 2021, pp. 177–191.

[40] F. Zhu, R. Gong, F. Yu, *et al.*, "Towards unified int8 training for convolutional neural network," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1969–1979.

[41] R. Banner, I. Hubara, E. Hoffer, and D. Soudry, "Scalable methods for 8-bit training of neural networks," *Advances in neural information processing systems*, 2018.

[42] B. Jacob, S. Kligys, B. Chen, *et al.*, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2704–2713.

[43] A. Farrag, S. Othman, T. Mahmoud, and A. Y. ELRaffiei, "Satellite swarm survey and new conceptual design for earth observation applications," *The Egyptian Journal of Remote Sensing and Space Science*, pp. 47–54, 2021.

[44] *Earth observation satellites in 2022*, https://www.pixalytics.com/earth-observation-satellites-2022, 2022.

[45] D. Giggenbach, J. Horwath, and M. Knapek, "Optical data downlinks from earth observation platforms," in *Proceedings of Free-Space Laser Communication Technologies XXI*, 2009, pp. 17–30.

[46] K. Devaraj, R. Kingsbury, M. Ligon, *et al.*, "Dove high speed downlink system," 2017.

[47] *Cubesat communications system table*, https://www.klofas.com, 2019.

[48] C. Li, J. Peng, L. Yuan, *et al.*, "Block-wisely supervised neural architecture search with knowledge distillation," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, Computer Vision Foundation / IEEE, 2020, pp. 1986–1995.

[49] T. Ibaraki, "Integer programming formulation of combinatorial optimization problems," *Discrete Mathematics*, vol. 16, no. 1, pp. 39–52, 1976.

[50] D. Li, X. Sun, *et al.*, *Nonlinear integer programming*. Springer, 2006, vol. 84.

[51] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM transactions on networking*, vol. 24, no. 5, pp. 2795–2808, 2015.

[52] G. Christie, N. Fendley, J. Wilson, and R. Mukherjee, "Functional map of the world," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6172–6180.

[53] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," *arXiv preprint arXiv:1506.03365*, 2015.

[54] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2017.

[55] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[56] C. He, S. Li, J. So, *et al.*, "Fedml: A research library and benchmark for federated machine learning," *Advances in Neural Information Processing Systems, Best Paper Award at Federate Learning Workshop*, 2020.

[57] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons, "The non-iid data quagmire of decentralized machine learning," in *Proceedings of International Conference on Machine Learning*, 2020, pp. 4387–4398.

[58] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[59] T.-M. H. Hsu, H. Qi, and M. Brown, "Federated visual classification with real-world data distribution," in *Proceedings of European Conference on Computer Vision*, 2020, pp. 76–92.

[60] J. Xia, G. Zeng, J. Zhang, *et al.*, "Rethinking transport layer design for distributed machine learning," in *Proceedings of the Asia-Pacific Workshop on Networking*, 2019, pp. 22–28.

[61] H. Jang, H. Lee, and J. Shin, "Unsupervised meta-learning via few-shot pseudo-supervised contrastive learning," *arXiv preprint arXiv:2303.00996*, 2023.

[62] A. Kirillov, E. Mintun, N. Ravi, *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

ChenYang is a Ph.D. candidate in the School of Computer Science, Beijing University of Posts and Telecommunication, Beijing, China. His research interests are federated learning and edge computing.

Jinliang Yuan is a Ph.D. candidate in the School of Computer Science, Beijing University of Posts and Telecommunication, Beijing, China. His research interests are distributed machine learning, federated learning, and edge computing.

Yaozong Wu is a Master candidate in the School of Computer Science, Beijing University of Posts and Telecommunication, Beijing, China. His research interests are federated learning and resource-efficient AI systems.

Qibo Sun is an Associate Professor at the School of Computer Science and Engineering, Beijing University of Posts and Telecommunications, China. He received his Ph.D. degree in communication and electronic system from the Beijing University of Posts and Telecommunication in 2002. His research interests include services computing, satellite computing and space-ground integration network. He has published more than 100 papers. He is a member of the China Computer Federation and Chinese Association for Artificial Intelligence.

Ao Zhou received Ph.D. degree in Beijing University of Posts and Telecommunications, Beijing, China, in 2015. She is currently an Associate Professor with State Key Laboratory of Networking and Switching Technology, BUPT. She has published 50+ research papers. She played a key role at many international conferences. Her research interests include cloud computing and mobile edge computing.

Shangguang Wang is a Professor at the School of Computer Science, Beijing University of Posts and Telecommunications, China. He is the founder & chief scientist of the Tiansuan Constellation. His research interests include service computing, mobile edge computing, and satellite computing. He is currently serving as chair of IEEE Technical Committee on Services Computing, and vice chair of IEEE Technical Committee on Cloud Computing. He also served as general chairs or program chairs of 10+ IEEE conferences. He is a Fellow of the IET, senior member of the IEEE. For further information on Dr. Wang, please visit: http://www.sguangwang.com.

Mengwei Xu is an assistant professor in the computer science department at Beijing University of Posts and Telecommunications. His research interests cover the broad areas of mobile computing, edge computing, artificial intelligence, and system software.