Semi-Supervised Federated Learning for Assessing Building Damage from Satellite Imagery

Yu Zhang[1], Yanmin Gong[1], Yuanxiong Guo[2]
[1]Department of Electrical and Computer Engineering, University of Texas at San Antonio
[2]Department of Information Systems and Cyber Security, University of Texas at San Antonio
Email: {yu.zhang@my., yanmin.gong@, yuanxiong.guo@}utsa.edu

***Abstract*—Accurate and timely building damage assessments are crucial for effective disaster response. However, traditional damage assessment methods heavily rely on manual evaluations by experts, which are labor-intensive and time-consuming. Recent research leverages machine learning (ML) and satellite remote sensing techniques to streamline the process. A major challenge of this method lies in the unlabeled nature of satellite imagery, which makes traditional ML frameworks impractical. Additionally, downloading the high-resolution satellite imagery for centralized ML is hindered by limited bandwidth and sporadic connectivity between the low Earth orbit (LEO) satellites and ground server. To address these challenges, we propose a novel semi-supervised federated learning framework named Semi-FedDA. It utilizes a small amount of labeled data on the ground server and a large amount of unlabeled data on the satellites to efficiently train a building assessment model without manual labeling. Moreover, this framework leverages intra-plane inter-satellite links (ISLs) to implement intra-orbit aggregations, which can significantly reduce the communication cost. We conduct extensive experiments on the real-world dataset. Numerical results show that our proposed framework can reduce training time by up to 94% compared with baselines, without sacrificing model accuracy.**

***Index Terms*—semi-supervised federated learning, damage assessment, satellite imagery, disaster management**

I. Introduction

Recent years have witnessed the rapid development of satellite technology, resulting in more low Earth orbit (LEO) satellites being deployed in space. These satellites continuously collect high-resolution Earth observation images and sensor data, which can support various novel applications, such as climate change monitoring [1] and disaster prediction [2]. Among them, post-disaster building damage assessment is considered one of the most important applications. Traditional building damage assessment from satellite imagery heavily relies on domain experts performing empirical analyses, which is labor-intensive and time-consuming. Therefore, there is immense potential in leveraging artificial intelligence to automate building damage assessment after natural disasters.

Recently, deep neural networks (DNNs) have been flourishing as never before in the field of object recognition and image segmentation [3]. There is substantial literature to explore how to utilize DNNs to accelerate the building damage assessment after natural disasters [4]–[8]. For example, Cooner et al. [4] evaluated the effectiveness of multilayer feedforward neural networks, radial basis neural networks, and random forests in detecting earthquake damage. However, most existing works focus on supervised learning tasks, assuming that the labels are available [4]–[7]. This assumption may not always be valid since there are a large number of satellite images generated every day (e.g., Planet Labs constellations generate more than 15 terabytes of data per day [9]), which makes manual labeling impractical. To overcome this challenge, Lee et al. [8] proposed an application of centralized semi-supervised learning (SSL) to train damage assessment models on a ground server with a small amount of labeled data and a large amount of unlabeled data. Even though utilizing SSL significantly reduces manual labeling costs, it is still challenging to download all high-resolution satellite imagery to the ground for implementing traditional centralized learning. This is primarily due to: (i) substantial propagation delay between the satellite and ground server owing to the limited bandwidth of download link; (ii) short communication time between the satellite and ground station because of the sporadic and irregular connectivity of download link; and (iii) real-time requirement of the post-disaster building damage assessment.

To overcome these challenges, federated learning (FL) is a promising approach [10]–[12]. In FL, the clients[1] collaboratively learn a shared prediction model under the orchestration of the server without sharing the raw data. Most prior works [13]–[15] on FL over LEO networks assume the clients have enough labeled data to train a global model. As aforementioned, they are not practicable since manual labeling of each satellite image is impossible. Recently, several studies [16]–[18] have explored a new framework called semi-supervised federated learning (SSFL). This framework leverages SSL to automate the imagery labeling and employs FL to reduce the communication cost. They conduct experiments on the image classification task using traditional convolutional neural networks (CNNs). However, traditional CNNs can not capture the complexity of the actual problem in assessing the post-disaster building damage. In the post-disaster building damage assessments, the dataset usually consists of two parts: pre- and post-disaster images. A primary challenge would be how to effectively model the correlation between these images for building damage assessment.

In this paper, we propose the first semi-supervised federated learning framework for assessing building damage from satellite imagery (Semi-FedDA). Particularly, the proposed

[1]Note that we use clients and satellites interchangeably in this paper.

framework takes into account the unique features of both LEO satellites (e.g., sporadic connectivity and unlabeled) and building damage assessment (e.g., pre- and post-disaster images as inputs). It significantly accelerates FL convergence by drastically reducing communication rounds without compromising model accuracy. Our main contributions are summarized as follows.

- We delineate the SSFL scenario of LEO satellite networks tailored for post-disaster damage assessment and highlight the unique challenges.
- We propose a novel framework dubbed Semi-FedDA for efficient training of post-disaster building damage assessment models using satellite imagery. This framework leverages a multitask model to perform joint building segmentation and damage classification. Furthermore, the communication cost is significantly reduced through the designed an intra-orbit aggregation strategy.
- We conduct extensive experiments based on the real-world dataset to evaluate our framework. Numerical results demonstrate that Semi-FedDA can save up to 94% training time compared with the traditional SSFL. Meanwhile, it maintains similar accuracy to supervised learning.

The rest of this paper is organized as follows. In Section II, we introduce the system model. In Section III, we present our Semi-FedDA framework for building damage assessment. Section IV shows the simulation results. Finally, the conclusion is given in Section V.

II. SYSTEM MODELING

We consider a general LEO constellation, which consists of a set of orbits $\mathcal{N} = \{1, \ldots, N\}$. Each orbit $n \in \mathcal{N}$ comprises a set of evenly distributed LEO satellites $\mathcal{I}_n = \{1, \ldots, I_n\}$. Then, the total satellite set can be denoted as $\mathcal{I} = \bigcup_{n=1}^{N} \mathcal{I}_n = \{1, \ldots, I\}$, where $I = \sum_{n=1}^{N} I_n$ is the total number of satellites. Let h_n denote the altitude of orbit n. Then, the speed and orbital period of the satellites in orbit n are expressed as $v_n = \sqrt{\frac{\mu}{h_n + r_E}}$ and $T_n = \frac{2\pi(r_E + h_n)}{v_n}$, respectively. Here, $r_E = 6371$ km is the Earth radius and $\mu = 3.98 \times 10^{14}$ m^3/s^2 is the geocentric gravitational constant. The satellites orbit the Earth and continuously capture high-resolution Earth observation images for training ML models to assess building damage after disasters. In order to cut down labeling and communication costs, we integrate the SSFL framework into this system. To that end, we assume a ground base station (BS) s orchestrates the training process.

A. SSFL at Satellites and Base Station

BS s has a small labeled data set $\mathcal{D}_s = \{(x_s^j, y_s^j)\}_{j=1}^{D_s}$, where x_s^j is a feature vector, y_s^j is the one-hot class label in a classification problem, and D_s is the size of $\mathcal{D}_s$. Moreover, each satellite $i \in \mathcal{I}$ only has an unlabeled data set $\mathcal{D}_i = \{x_i^j\}_{j=1}^{D_i}$, where x_i is a feature vector and D_i is size of $\mathcal{D}_i$. As shown in Fig. 1, the SSFL implemented in satellites and BS consists of the following steps [16]:

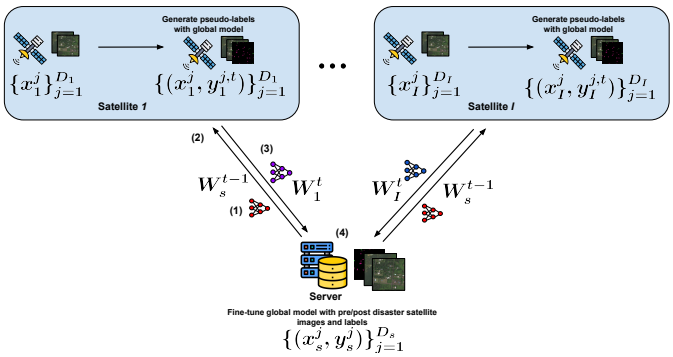


Fig. 1: An illustration of SSFL framework at satellite networks and base station

1) At round t, BS s distributes the global model W_s^{t-1} to each satellite when the satellite is in the visibility period.
2) Once satellite $i \in \mathcal{I}$ receives the global model W_s^{t-1}, it initializes the local model $W_i^t \leftarrow W_s^{t-1}$. Then, the satellite generates pseudo-labels with weakly augmented unlabeled data, i.e., $y_i^{j,t} = f(\alpha(x_i^j), W_i^t), \forall j$. Here, $\alpha(\cdot)$ denotes a weak data augmentation, e.g., random horizontal and vertical flipping, that maps one image to another. $f(x, w)$ presents how to map an input x and model parameter w to an one-hot prediction. Then, the satellite generates a high-confidence dataset $\mathcal{D}_i^{\text{fix},t} = \{(x_i^j, y_i^{j,t})$ with $\max(y_i^{j,t}) \geq \phi\}_{j=1}^{D_i}$, where $0 \leq \phi \leq 1$ is a global confidence threshold. If $\mathcal{D}_i^{\text{fix},t} = \emptyset$, the satellite stops training. Otherwise, the satellite constructs another $\mathcal{D}_i^{\text{mix},t}$ by sampling $|\mathcal{D}_i^{\text{fix},t}|$ with replacement of $\{(x_i^j, y_i^{j,t})\}_{j=1}^{D_i}$. Here, $|\mathcal{D}_i^{\text{fix},t}|$ denotes the number of elements in $\mathcal{D}_i^{\text{fix},t}$. After obtaining these two datasets, the satellite starts to train the local model for E epochs. In each epoch, the satellite randomly splits $\mathcal{D}_i^{\text{fix},t}$ and $\mathcal{D}_i^{\text{mix},t}$ into batch sets $\mathcal{B}_i^{\text{fix},t}$ and $\mathcal{B}_i^{\text{mix},t}$, respectively, each with same batch size B_i. Consequently, the satellite constructs Mixup data from one particular data batch $(x_b^{\text{fix}}, y_b^{\text{fix}}), (x_b^{\text{mix}}, y_b^{\text{mix}}) \in \mathcal{B}_i^{\text{fix},t}, \mathcal{B}_i^{\text{mix},t}$ by $x_{\text{mix}} \leftarrow \phi x_b^{\text{fix}} + (1-\phi) x_b^{\text{mix}}$, where $\phi \sim \text{Beta}(\gamma, \gamma)$, and γ is the hyperparameter. Based on the above data, the satellite calculates the following two losses [16]: $L_{\text{fix}} = \ell(f(\mathcal{A}(x_b^{\text{fix}}, W_i^t), y_b^{\text{fix}})$ and $L_{\text{mix}} = \phi \cdot \ell(f(\alpha(x_{\text{mix}}, W_i^t), y_b^{\text{fix}}) + (1-\phi) \cdot \ell(f(\alpha(x_{\text{mix}}, W_i^t), y_b^{\text{mix}})$, where $\mathcal{A}$ denotes a strong data augmentation mapping, e.g., the RandAugment [19], and ℓ denotes the cross entropy. Finally, the satellite updates the its local model as

$$W_i^t \leftarrow W_i^t - \eta \nabla_W (L_{\text{fix}} + \lambda \cdot L_{\text{mix}}), \tag{1}$$

where η is the learning rate, and $\lambda > 0$ is a hyperparameter to balance L_{fix} and L_{mix}. After training for E epochs, satellite i obtains an updated local model W_i^t.
3) Each satellite transmits the updated local model W_i^t during the visibility period. Note that we are interested in the case with full satellites participation.
4) After BS s collects all updated local models from the

satellites, it aggregates the models as

$$W_s^t = \sum_{i \in I} \frac{D_i}{D} W_i^t, \tag{2}$$

where $D = \sum_{i \in I} D_i$ is the total number of training samples. After obtaining the global model W_s^t, the BS will locally update the model for E epochs. In each epoch, the BS randomly splits its labeled data $\mathcal{D}_s$ into batch set $\mathcal{B}_s^t$. For each batch $(x_s, y_s) \in \mathcal{B}_s^t$, the BS updates the global model as

$$W_s^t \leftarrow W_s^t - \eta \nabla_W f(\alpha(x_s, W_s^t), y_s), \tag{3}$$

The above steps repeat until the global model is converged (e.g., a target loss or accuracy is achieved).

B. Computation Model

As mentioned above, each satellite $i \in \mathcal{I}$ first utilizes the initialized local model W_i^t to generate pseudo-labels at each round. The time of this process is described as

$$\tau_i^{\text{label}} = \frac{C_i^{\text{label}} S(\mathcal{D}_i)}{\nu_i}, \tag{4}$$

where C_i^{label} is the number of CPU cycles required to label a single data bit, $S(\mathcal{D}_i)$ is the size of data set $\mathcal{D}_i$ in bits, and ν_i is the CPU frequency of satellite. Next, the satellite will construct three datasets, i.e., $\mathcal{D}_i^{\text{fix},t}$, $\mathcal{D}_i^{\text{mix},t}$, and a Mixup dataset. Compared with the entire computation latency, the time to generate the datasets is very short, and therefore can be neglected. After obtaining the datasets, the satellite trains the local model for E epochs at each round. According to the linear computation time model [20], the time required by the satellite to update the local model is expressed as

$$\tau_i^{\text{train}} = \frac{E C_i^{\text{train}} S(\mathcal{D}_i)}{\nu_i} \tag{5}$$

where C_i^{train} is the number of CPU cycles required to process a single data bit.

Moreover, we assume the BS has sufficient computation resources. Thus, we ignore the computation time spent by the BS.

C. Communication Model

In an LEO satellite constellation, a satellite $i \in \mathcal{I}$ is in the visibility period of BS s at time slot τ, when $(\frac{\pi}{2}) - \angle(\vec{r}_s(\tau), \vec{r}_i(\tau) - \vec{r}_s(\tau)) \geq \alpha_e$. Here, $\vec{r}_i(\tau)$ and $\vec{r}_s(\tau)$ denote the position of the satellite and the BS, respectively, and α_e denotes the minimum elevation angle. For ease of expression, let $\omega_{i,s} = \angle(\vec{r}_s(\tau), \vec{r}_i(\tau) - \vec{r}_s(\tau))$, and we omit the time index. According to [21], the signal-to-noise ratio (SNR) between satellite i and BS s is expressed as

$$\text{SNR}_{i,s} = \begin{cases} \frac{G_i G_s P}{k_B B T \mathcal{L}_{i,s}} & \text{if } \frac{\pi}{2} - \omega_{i,s} \geq \alpha_e, \\ 0 & \text{otherwise}, \end{cases} \tag{6}$$

where G_i and G_s are the antenna gains of satellite i and BS s, respectively, P is the transmission power, k_B is the Boltzmann constant, B is the channel bandwidth, T is the receiver noise temperature, and $\mathcal{L}_{i,s}$ is the free space path loss between satellite i and BS s. Moreover, the free space path loss $\mathcal{L}_{i,s}$ is described as

$$\mathcal{L}_{i,s} = \left(\frac{4\pi f_c d(i,s)}{c} \right)^2, \tag{7}$$

where $d(i,s)$ is the distance between satellite i and the BS, f_c is the carrier frequency, and c is the speed of light.

Based on Shannon's theorem, the achievable data transmission rate from BS s to satellite i is expressed as

$$R_{i,s} = B_i \log_2(1 + \text{SNR}_{i,s}), \tag{8}$$

where B_i is the per-satellite bandwidth pre-assigned to satellite i. Then, the time for exchanging the model W between satellite i and BS s is expressed as

$$\tau_{i,s}^{\text{trans}} = \frac{S(W)}{R_{i,s}} + \frac{d(i,s)}{c}, \tag{9}$$

where $S(W)$ is the data size of model W in bits. The first term represents the required time for transmission, while the second term represents the propagation delay.

In a conventional star topology-based SSFL framework, each satellite needs to complete the SSFL tasks at every round. These tasks involve each satellite receiving the global model W_s^{t-1} from the BS, retraining the global model to obtain the local model W_i^t, and transmitting W_i^t to the BS. Based on the computation and communication models, we can calculate the total time required for each satellite i to complete these tasks as

$$\tau_i^{\text{ST}} = \tau_i^{\text{wait}} + 2\tau_{i,s}^{\text{trans}} + \tau_i^{\text{label}} + \tau_i^{\text{train}}, \tag{10}$$

where τ_i^{wait} is the waiting period for satellite i to enter the visible zone. In a global communication round, the BS needs to collect all trained local models from the satellites. Thus, the total time for a global round is expressed as

$$\tau^{\text{total}} = \max_{i \in \mathcal{I}} \tau_i^{\text{ST}}. \tag{11}$$

From the above equations, we can observe that the global training time depends on the slowest satellites in this LEO constellation. Additionally, the extensive time required for each satellite to communicate with the BS results in inefficiency. To address these issues, we propose an innovative SSFL framework, which is detailed in the following section.

III. PROPOSED FRAMEWORK

As introduced in Section I, the primary challenges to assessing building damage after disasters are twofold: the complexity of evaluating the building damages and intermittent connectivity between the satellites and BS. In this paper, we introduce a cutting-edge framework Semi-FedDA, designed to address these challenges. It comprises two main components: the Siamese U-Net model, which tackles the complexity of the building damage assessment task, and SSFL with intra-orbit aggregation, which addresses the intermittent connectivity between the satellites and BS. In the next subsections, we will introduce these two components.

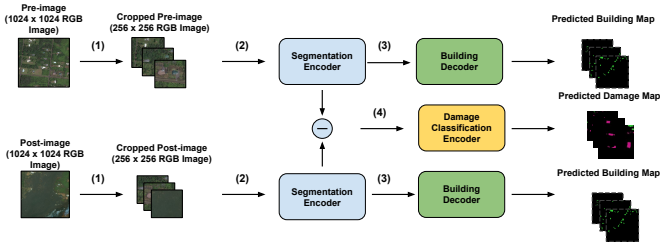


Fig. 2: An end-to-end workflow for the prediction of Siamese U-Net model

A. Siamese U-Net model

In past decades, a wide variety of deep learning architectures for semantic segmentation have been proposed [22], [23]. The majority of these models adopt the encoder-decoder framework, wherein the encoder generates representative feature maps at various spatial dimensions, and the decoder subsequently reconstructs a full-resolution semantic mask. In these traditional image segmentation tasks, the model usually inputs one task data and outputs one predicted label. However, in building damage assessment tasks, the input is a pre- and post-disaster image pair. Moreover, the model needs to output two labels: one for building segmentation and the other for damage classification. By concurrently training a single backbone on pairs of images, Siamese networks have achieved notable success in identifying similarities and differences between images [24], [25]. These networks utilize a common backbone to derive embeddings from both images, enabling the effective learning of distinctive features within a fully connected layer.

For building damage assessment in disasters, we utilize a similar Siamese U-Net model proposed in [7]. As shown in Fig. 2, the end-to-end prediction majorly consists of four steps: 1) We crop the satellite images with a resolution of 1024×1024 pixels into smaller images of 256×256 pixels. 2) Then, we feed pre- and post-disaster images through a U-Net segmentation module with shared weights (blocks with the same color in the figure share the weights). 3) The outputs of the two segmentation encoders are forwarded to the two building decoders with shared weights, respectively. These two building decoders can produce detailed per-pixel building-level predictions based on pre- and post-disaster images. 4) Besides, the outputs of the segmentation encoders are subtracted, and the result is forwarded to a damage classification decoder. This decoder produces detailed per-pixel damage level predictions.

B. SSFL with intra-orbit aggregation

Owing to the sporadic connections between the satellites and BS, there is a significant amount of idle time as satellites await entry into the visible zone. Furthermore, as illustrated by Eq. (11), the overall training speed is determined by the slowest satellite in the constellation. It is important to highlight that the intra-plane inter-satellite links (ISLs) operate at a significantly higher transmission rate compared to the connections between the satellites and BS. Similar to [13], we develop an intra-orbit aggregation strategy based on ISLs to

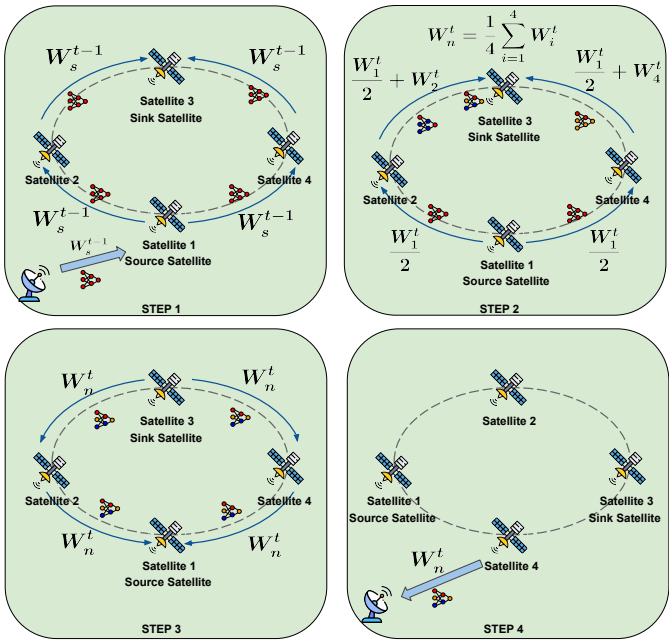


Fig. 3: An illustration of SSFL with intra-orbit aggregation

effectively tackle these challenges. As illustrated in Fig. 3, the intra-orbit aggregation strategy comprises the following steps:

1) For each orbit $n \in \mathcal{N}$, we refer to the satellite, which is the first to receive the global model W_s^{t-1} from the BS, as the source node. Upon receiving W_s^{t-1}, the source node then forwards it to its neighboring satellites. Subsequently, these neighbors further forward W_s^{t-1} to their next-hop neighbors, continuing this process until W_s^{t-1} reaches the final satellite, known as the sink node. In this process, we assume the satellites start the training process after completion of the forwarding task. The only exception is the source node, which initiates its training immediately after receiving W_s^{t-1}.
2) After completing the training process, each satellite $i \in \mathcal{I}_n$ generates a local model W_i^t. For simplicity, we assume each satellite has the same computation capacity and data size. However, it can be easily extended to the case when they are different. Consequently, under this assumption, the source node is the first to complete the training process. Then, the source node will forward the modified local model $\frac{W_s^{t-1}}{2}$ to its neighboring satellites (i.e., satellites 2 and 4). The neighboring satellites aggregate the $\frac{W_s^{t-1}}{2}$ with their own models and forward the aggregated models (i.e., $\frac{W_s^{t-1}}{2} + W_2^t$ and $\frac{W_s^{t-1}}{2} + W_4^t$) to their next-hop neighbors. This process continues until the sink node receives the aggregation models. Finally, the sink node executes the last aggregation and obtains an updated orbit model, which is expressed as
$$W_n^t = \frac{1}{I_n} \sum_{i=1}^{I_n} W_i^t. \tag{12}$$
3) After obtaining the updated orbit model W_n^t, we execute

TABLE I: Simulation Parameters

Parameters	Values
Bandwidth B	20MHz
Transmission power of BS P_s	42dBm
Transmission power of each satellite P_i	43dBm
Antenna gain of each satellite G_i	6.98dBi
Carrier frequency f_c	2.4GHz
Noise temperature T	354.81K
CPU frequency of each satellite ν_i	1GHz

a reverse relaying of Step 1. Specifically, the sink node forwards W_n^t to its neighboring satellites. Then, these neighbors further forward W_n^t to their next-hop neighbors, continuing this process until W_n^t reaches the source node.

4) Since each satellite has the same orbit model W_n^t after finishing the reverse relaying in Step 3, the first satellite entering the visible zone is then able to transmit W_n^t to the BS.

We execute the intra-orbit aggregation strategy for each orbit. When the BS collects all orbit models, it can update the global model as

$$W_s^t = \frac{1}{N}\sum_{n=1}^{N} W_n^t. \tag{13}$$

ISL plays a crucial role in this intra-orbit aggregation strategy. Since the satellites are evenly distributed in each orbit n, the time for transmitting the model W between neighboring satellites in orbit n using ISL can be presented as

$$\tau_n^{\text{ISL}} = \frac{S(W)}{R_n} + \frac{d_n}{c}, \tag{14}$$

Here, R_n is the transmission rate between neighboring satellites, which can be calculated in a similar fashion as Eq. (8). d_n is the distance between them. The first term is the transmission delay, while the second term is the propagation delay. Then, the overall time for model relays and training in orbit n can be expressed as

$$\tau_n^{\text{ISL}} = \tau_n^{\text{wait}} + 2\tau_{n,s}^{\text{trans}} + \tau_i^{\text{label}} + \tau_i^{\text{train}} + 2\left\lfloor \frac{I_n}{2} \right\rfloor \tau_n^{\text{ISL}}, \tag{15}$$

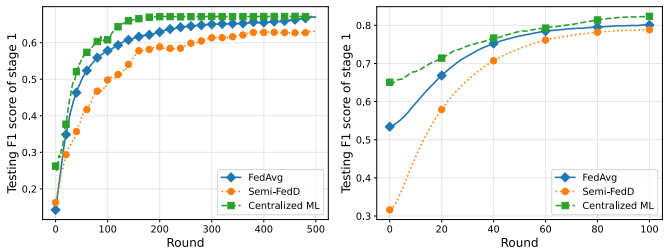
where τ_n^{wait} is the waiting period for a satellite in orbit n to enter the visible zone. Here, we ignore the aggregation time for each satellite since it is very short compared with the total processing time. Hence, the total time for a global round using the intra-orbit aggregation strategy is expressed as

$$\tau^{\text{total}} = \max_{n\in\mathcal{N}} \tau_n^{\text{ISL}}. \tag{16}$$

Since the τ_n^{wait} is much smaller than τ_i^{wait}, the proposed method can significantly reduce the communication costs. We will further demonstrate this in the next section.

IV. NUMERICAL EXPERIMENTS

In this section, we conduct extensive numerical experiments to evaluate the performance of our proposed Semi-FedDA framework. All simulations are implemented using PyTorch on an Ubuntu server with 4 NVIDIA RTX 8000 GPUs.



(a) Training stage 1 (b) Training stage 2

Fig. 4: Convergence comparison between Semi-FedDA and baselines

TABLE II: F1 score comparison between Semi-FedDA and baselines

Schemes	F_1^{bld}	F_1^{dmg}	F_1^{total}
FedAvg	0.62	0.80	0.746
Centralized ML	0.68	0.82	0.778
Semi-FedDA	0.66	0.78	0.744

A. Simulation Setup

We consider 20 satellites evenly distributed across 2 orbits at altitude of 2000 km with inclination of 80°. The BS is located at coordinates 53.0793°N latitude and 8.8017°E longitude. The minimum elevation angle of BS is 10°. Unless otherwise specified, the rest of our simulation parameters are given in Table I.

We utilize the 2018 Lower Puna eruption satellite imagery dataset [6]. After preprocessing, the dataset comprises 4660 training and 600 testing images, each with a resolution of 256×256 pixels. In this collection of images, there are a total of 3410 buildings, out of which 2831 buildings have suffered damage, while 579 buildings remain undamaged. For the Semi-FedDA setting, we assume the BS has only 400 labeled images, while each satellite has 213 unlabeled images. We use the $F1$ score as the evaluation matrix. Moreover, since our model can simultaneously implement building segmentation and damage classification, we utilize a weighted $F1$ score to evaluate the overall performance as

$$F_1^{\text{total}} = 0.3 \cdot F_1^{\text{bld}} + 0.7 \cdot F_1^{\text{dmg}}, \tag{17}$$

where F_1^{bld} and F_1^{dmg} represent the $F1$ score of building segmentation and damage classification, respectively.

To evaluate the performance of the proposed framework, we consider the following benchmarks: 1) FedAvg [10]: In this scheme, the BS has 400 labeled images, and each satellite has 213 labeled images. The full-supervised FedAvg is implemented. 2) Centralized ML [7]: We assume the BS has all labeled 4660 data. A centralized full-supervised learning is implemented in the BS. 3) SSFL: This scheme shares the same settings as Semi-FedDA but lacks the intra-orbit aggregation strategy.

TABLE III: Training time comparison between Semi-FedDA and baselines

Schemes	Training time of stage 1	Training time of stage 2
FedAvg	14399986s	3455819s
SSFL	14399923s	3455794s
Semi-FedDA	857464s	194207s

B. Evaluation Results

The training of our model consists of two stages. In the first stage, we freeze the damage classification encoder and train the segmentation encoders and building decoders. In the second stage, we freeze the segmentation encoders and building decoders and train the damage classification encoder. Fig. 4(a) and Fig. 4(b) demonstrate the convergence of training stages 1 and 2, respectively. These figures indicate that Semi-FedDA exhibits comparable performance with FedAvg and Centralized ML regarding the number of rounds needed to converge. Table II and Table III further demonstrate the details. Table II shows that Semi-FedDA experiences a minimal degradation in the F1 score, by 3.8% and 2.9% compared to Centralized ML and FedAvg, respectively. In other words, Semi-FedDA can achieve almost the same performance with only a small portion of the labeled data. Furthermore, thanks to the intra-orbit aggregation strategy, Semi-FedDA can save up to 94% training time compared with traditional SSFL, as shown in Table III.

V. CONCLUSION

In this study, we propose an innovative framework called Semi-FedDA to automate building damage assessment in disaster scenarios using satellite imagery. Compared with the traditional expert-driven labeling and assessment process, our framework can significantly reduce labor and time costs. For future works, we plan to design a more efficient satellite communication protocol for the Semi-FedDA framework.

VI. ACKNOWLEDGEMENTS

The work of Y. Zhang, Y. Guo, and Y. Gong was partially supported by NSF under grants CNS-2047761, CNS2106761, CMMI-2222670, CNS-2318683, and the gift donation from Cisco. This article solely reflects the opinions and conclusions of its authors and not the funding agents. The authors extend their gratitude to Xiangxing Guo for assisting in conducting the experiments and offering insightful feedback.

REFERENCES

[1] K. Imaoka, M. Kachi, H. Fujii, H. Murakami, M. Hori, A. Ono, T. Igarashi, K. Nakagawa, T. Oki, Y. Honda *et al.*, "Global change observation mission (GCOM) for monitoring carbon, water cycles, and climate change," *Proceedings of the IEEE*, vol. 98, no. 5, pp. 717–734, 2010.

[2] T. W. Gillespie, J. Chu, E. Frankenberg, and D. Thomas, "Assessment and prediction of natural hazards from satellite imagery," *Progress in Physical Geography*, vol. 31, no. 5, pp. 459–470, 2007.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] A. J. Cooner, Y. Shao, and J. B. Campbell, "Detection of urban damage using remote sensing and machine learning algorithms: Revisiting the 2010 Haiti earthquake," *Remote Sensing*, vol. 8, no. 10, p. 868, 2016.

[5] D. Duarte, F. Nex, N. Kerle, and G. Vosselman, "Satellite image classification of building damages using airborne and satellite image samples in a deep learning approach," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 4, pp. 89–96, 2018.

[6] R. Gupta, R. Hosfelt, S. Sajeev, N. Patel, B. Goodman, J. Doshi, E. Heim, H. Choset, and M. Gaston, "xBD: A dataset for assessing building damage from satellite imagery," *arXiv preprint arXiv:1911.09296*, 2019.

[7] S. Gholami, C. Robinson, A. Ortiz, S. Yang, J. Margutti, C. Birge, R. Dodhia, and J. L. Ferres, "On the deployment of post-disaster building damage assessment tools using satellite imagery: A deep learning approach," in *IEEE International Conference on Data Mining Workshops (ICDMW)*, 2022, pp. 1029–1036.

[8] J. Lee, J. Z. Xu, K. Sohn, W. Lu, D. Berthelot, I. Gur, P. Khaitan, K. Koupparis, B. Kowatsch *et al.*, "Assessing post-disaster damage from satellite imagery using semi-supervised learning techniques," *arXiv preprint arXiv:2011.14004*, 2020.

[9] "Explore planet products," October 2023. [Online]. Available: https://www.planet.com/products/

[10] 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.

[11] Y. Guo, Y. Sun, R. Hu, and Y. Gong, "Hybrid local sgd for federated learning with heterogeneous communications," in *International Conference on Learning Representations*, 2022.

[12] Z. Zhang, Z. Gao, Y. Guo, and Y. Gong, "Scalable and low-latency federated learning with cooperative mobile edge networking," *IEEE Transactions on Mobile Computing*, 2022.

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

[14] M. Elmahallawy and T. Luo, "AsyncFLEO: Asynchronous federated learning for LEO satellite constellations with high-altitude platforms," in *IEEE International Conference on Big Data*, 2022, pp. 5478–5487.

[15] N. Razmi, B. Matthiesen, A. Dekorsy, and P. Popovski, "Ground-assisted federated learning in LEO satellite constellations," *IEEE Wireless Communications Letters*, vol. 11, no. 4, pp. 717–721, 2022.

[16] E. Diao, J. Ding, and V. Tarokh, "SemiFL: Semi-supervised federated learning for unlabeled clients with alternate training," *Advances in Neural Information Processing Systems*, vol. 35, pp. 17 871–17 884, 2022.

[17] W. Jeong, J. Yoon, E. Yang, and S. J. Hwang, "Federated semi-supervised learning with inter-client consistency & disjoint learning," *arXiv preprint arXiv:2006.12097*, 2020.

[18] J. Bian, Z. Fu, and J. Xu, "FedSEAL: Semi-supervised federated learning with self-ensemble learning and negative learning," *arXiv preprint arXiv:2110.07829*, 2021.

[19] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 702–703.

[20] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM conference on computer communications*, 2019, pp. 1387–1395.

[21] L. J. Ippolito Jr, *Satellite communications systems engineering: atmospheric effects, satellite link design and system performance*. John Wiley & Sons, 2017.

[22] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.

[24] S. Dey, A. Dutta, J. I. Toledo, S. K. Ghosh, J. Lladós, and U. Pal, "Signet: Convolutional siamese network for writer independent offline signature verification," *arXiv preprint arXiv:1707.02131*, 2017.

[25] R. C. Daudt, B. Le Saux, and A. Boulch, "Fully convolutional siamese networks for change detection," in *IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 4063–4067.