Communication-Efficient Learning for Satellite Constellations [*]

Ruxandra-Stefania Tudose [*] **Moritz H.W. Grüss** [*]
Grace Ra Kim [**] **Karl H. Johansson** [***] **Nicola Bastianello** [***]

[*] *KTH Royal Institute of Technology, Sweden*
[**] *Stanford University, United States of America*
[***] *School of Electrical Engineering and Computer Science, and Digital*
Futures, KTH Royal Institute of Technology, Sweden

Abstract: Satellite constellations in low-Earth orbit are now widespread, enabling positioning, Earth imaging, and communications. In this paper we address the solution of learning problems using these satellite constellations. In particular, we focus on a federated approach, where satellites collect and locally process data, with the ground station aggregating local models. We focus on designing a novel, communication-efficient algorithm that still yields accurate trained models. To this end, we employ several mechanisms to reduce the number of communications with the ground station (local training) and their size (compression). We then propose an error feedback mechanism that enhances accuracy, which yields, as a byproduct, an algorithm-agnostic error feedback scheme that can be more broadly applied. We analyze the convergence of the resulting algorithm, and compare it with the state of the art through simulations in a realistic space scenario, showcasing superior performance.

Keywords: satellite constellations, federated learning, compression, error-feedback, on-board satellite learning

1. INTRODUCTION

Low-Earth orbit (LEO) satellites have proved a revolutionary technology for several applications, especially positioning, Earth imaging, communications (Çelikbilek et al., 2022). In recent years, increasing numbers of LEO satellites have been placed in orbit, driven by the rise of large constellations (De Luca and Evroux, 2025). These satellite constellations then enable the collection of vast sets of data, *e.g.* Earth images, which can be used to train accurate models of different phenomena. Examples include models that can be trained for disaster navigation (Barmpoutis et al., 2020), real-time earthquake prediction (Zhai et al., 2024), disease spread (Franch-Pardo et al., 2020), food security (Aragon et al., 2018) and climate change (Shukla et al., 2021; So et al., 2022; Zhai et al., 2024).

However, enabling learning on data collected by satellites poses significant practical challenges, mainly due to the disparity between data generation and the bandwidth available to transmit the data to ground stations (GSs) on Earth (So et al., 2022). Indeed, communications to the GS are limited by sparse communication windows and delays.

The solution that has been proposed to this issue is to leverage the paradigm of federated learning (FL) (Li et al., 2020a). The idea is to process data directly onboard the satellites, and to share only the resulting models. These models are transmitted to the GS that acts as a *coordinator*, aggregating them and sending the result to the satellites for further training; see Figure 1 for a depiction of this scheme. The resulting approach reduces
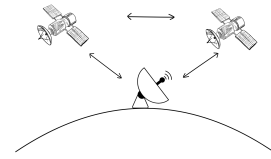

Fig. 1. Federated learning for a satellite constellation, with ground station acting as coordinator.

significantly the bandwidth required (models, and not data, are transmitted) while still training accurate models on the data collected by all satellites in the constellation. Apart from these notable features, FL brings another key contribution which is the enhanced security of the data. This is because in the FL architecture, the original data used for training never leaves the devices, only the model updates do (Uddin and Kumar, 2022).

However, even when employing a federated learning approach, the bandwidth is still limited due to sparse communication windows to the GS. In FL, several techniques have been proposed to enhance the communication efficiency of training algorithms. These can be categorized as compression/quantization (which reduces the size of

messages) (Zhao et al., 2023), partial participation (which reduces the number of agents transmitting to the coordinator) (Németh et al., 2022), and local training (which reduces the frequency of agent-coordinator communications) (Grudzień et al., 2023). In satellite applications, the use of both local training and partial participation have been tested, see (So et al., 2022; Zhai et al., 2024; Razmi et al., 2022; Han et al., 2024). In particular, satellites perform local training while they do not have a communication window to the GS. Moreover, scheduling only a limited set of satellites to communicate with the GS leads to partial participation and hence reduction of the number of messages to be transmitted (Kim et al., 2025). Although these algorithms showcase practical improvements in performance, theoretical insights are still lacking. Additionally, communication compression was not employed, while in practice it is an important measure to improve communication efficiency. In this paper we aim to incorporate compression, and provide suitable theoretical results.

However, compression can reduce the accuracy of the trained model. The idea then is to employ error feedback (EF) to reduce compression's impact on accuracy (Karimireddy et al., 2019). As the name suggests, error feedback enacts a feedback on the compression error (that is, the difference between compressed and uncompressed messages), so that this error is integrated in future messages, preventing loss of information. Integrating EF thus allows to improve the accuracy of the trained model while still having communication efficiency.

In this paper we propose a satellite-ready FL algorithm designed to guarantee both communication efficiency and training accuracy. We start from the foundation of Fed-LT[1] presented in (Bastianello et al., 2024), which uses local training and partial participation to reduce the frequency and volume of communications. To these we add compression to then reduce the size of communications. To ensure accuracy, we also implement an error feedback mechanism similar to (Cheng et al., 2025). The main contributions are summarized as follows:

- We design a novel FL algorithm that integrates a suite of communication efficiency tools (local training, partial participation, compression) while ensuring accuracy through error feedback. The compression is applied bi-directionally between agents and the coordinator.
- As a byproduct of this design, we present an algorithm-agnostic implementation of error feedback that can be plugged into any existing federated method.
- We further present a satellite-ready version of the algorithm, called Fed-LTSat, which integrates inter-satellite communications, which enhance accuracy while reducing the number of satellite-to-Earth communications.
- We test the resulting algorithm in numerical simulations implemented in FLySTacK, a platform that simulates space constraints (Kim et al., 2024), and compare with the state of the art.

---

[1] We call Fed-LT the algorithm Fed-PLT of (Bastianello et al., 2024) without privacy (P) mechanism.

2. ALGORITHM DESIGN AND CONVERGENCE ANALYSIS

We start by formalizing the problem we aim to solve, then discuss the communication-efficient Fed-LT with EF, and its satellite-ready version. We conclude by analyzing their convergence.
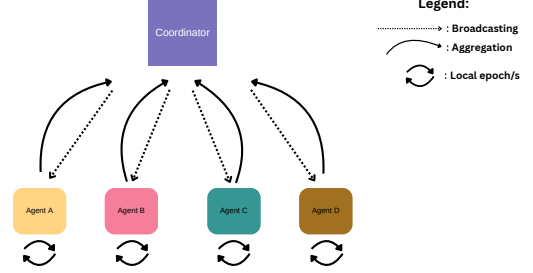
2.1 Problem formulation

Fig. 2. Scheme of the federated learning set-up.

The abstract federated learning set-up is depicted in Figure 2, with each of N agents (*e.g.* satellites) locally storing and processing data, and the coordinator (*e.g.* ground station) aggregating the local models. Formally, each agent stores a loss function $f_i : \mathbb{R}^n \to \mathbb{R}$, usually in an empirical risk form:

$$f_i(x) = \frac{1}{m_i} \sum_{h=1}^{m_i} \ell(x, d_i^h)$$

with $\{d_i^h\}_{h=1}^{m_i}$ the local data and $\ell : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$ a loss function (*e.g.* logistic loss). These losses then define the following problem:

$$\min_{x_i \in \mathbb{R}^n, \ i=1,\ldots,N} \sum_{i=1}^{N} f_i(x_i) \tag{1a}$$

$$\text{s.t. } x_1 = \ldots = x_N \tag{1b}$$

which is iteratively solved through the scheme of Figure 2.

2.2 Algorithm design: communication efficiency

The foundation of our proposed algorithm is Fed-LT from (Bastianello et al., 2024), which already incorporates local training and partial participation. The goal of this section is to further enhance Fed-LT's communication efficiency by integrating compression, while still guaranteeing accuracy through error feedback.

In Algorithm 1 we report Fed-LT (Bastianello et al., 2024) with the addition of compression on the uplink (agent-to-coordinator transmissions) and on the downlink (coordinator to agents broadcasts). During the execution of Fed-LT, each (active) agent locally trains the model x_i (using the local loss f_i and a local solver like gradient descent), and updates the auxiliary variable z_i. The latter is compressed (via compression operator C_u) and transmitted to the coordinator. The coordinator then aggregates the z_i's received (using past information for inactive agents), compresses the result with C_d, and broadcasts it to all agents.

Clearly, the bi-directional compression in Algorithm 1 causes loss of information twice, characterized by the

Algorithm 1 Fed-LT with compression

Input: For each agent initialize $x_{i,0}$ and $z_{i,0}$; choose the local solver, its parameters, and the number of local epochs N_e; the parameter $\rho > 0$; choose the uplink C_u and downlink C_d compression; let $S_k \subset \{1, \ldots, N\}$ denote the agents active at iteration k

1: **for** $k = 0, 1, 2, \ldots,$ **do**
    // coordinator update
2:    receive $z_{i,k}$ from each agent $i \in S_k$
3:    aggregate

$$y_{k+1} = \frac{1}{N}\left(\sum_{i \in S_k} z_{i,k} + \sum_{i \notin S_k} z_{i,k-1}\right)$$

4:    compress y_{k+1}, $C_d(y_{k+1})$, and transmits it to all active agents ▷ downlink compression
    // agents updates
5:    gather the active agents in S_{k+1}
6:    **for** $i = 1, \ldots, N$ **do**
7:      **if** $i \in S_{k+1}$ **then** ▷ partial participation
8:        set $w_{i,k}^0 = x_{i,k}$, $v_{i,k} = 2y_{k+1} - z_{i,k}$
9:        **for** $\ell = 0, 1, \ldots, N_e - 1$ **do** ▷ local training
10:          *e.g.* gradient descent

$$w_{i,k}^{\ell+1} = w_{i,k}^\ell - \gamma\left(\nabla f_i(w_{i,k}^\ell) + \frac{1}{\rho}(w_{i,k}^\ell - v_{i,k})\right)$$

11:        **end for**
12:        set $x_{i,k+1} = w_{i,k}^{N_e}$
13:        update the auxiliary variable
$$z_{i,k+1} = z_{i,k} + 2(x_{i,k+1} - y_{k+1})$$
14:        compress $z_{i,k+1}$, $C_u(z_{i,k+1})$, and transmit to coordinator ▷ uplink compression
15:      **else** ▷ inactive agent
16:        set $x_{i,k+1} = x_{i,k}$, $z_{i,k+1} = z_{i,k}$
17:      **end if**
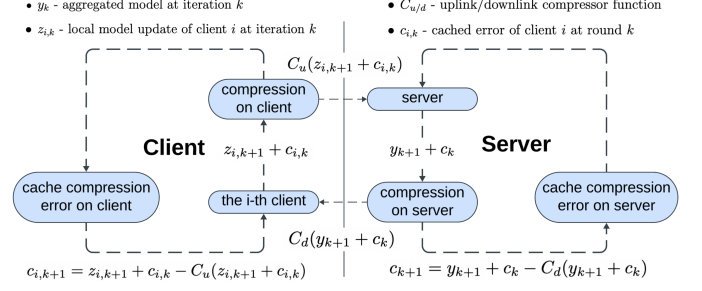18:   **end for**
19: **end for**

Fig. 3. Flow chart of the agnostic error-feedback mechanism.

Algorithm 2 Fed-LT with compression and EF

Input: For each agent initialize $x_{i,0}$ and $z_{i,0}$, and the cache $c_{i,0} = 0$; initialize the coordinator cache $c_0 = 0$; choose the local solver, its parameters, and the number of local epochs N_e; the parameter $\rho > 0$; choose the uplink C_u and downlink C_d compression; let $S_k \subset \{1, \ldots, N\}$ denote the agents active at iteration k

1: **for** $k = 0, 1, 2, \ldots,$ **do**
    // coordinator update
2:    receive $z_{i,k}$ from each agent $i \in S_k$
3:    aggregate ▷ downlink EF

$$y_{k+1} = c_k + \frac{1}{N}\left(\sum_{i \in S_k} z_{i,k} + \sum_{i \notin S_k} z_{i,k-1}\right)$$

4:    compress y_{k+1}, $C_d(y_{k+1})$, and transmits it to all active agents ▷ downlink compression
5:    cache the compression error $c_{k+1} = y_{k+1} - C_d(y_{k+1})$
    // agents updates
6:    gather the active agents in S_{k+1}
7:    **for** $i = 1, \ldots, N$ **do**
8:      **if** $i \in S_{k+1}$ **then** ▷ partial participation
9:        set $w_{i,k}^0 = x_{i,k}$, $v_{i,k} = 2y_{k+1} - z_{i,k}$
10:        **for** $\ell = 0, 1, \ldots, N_e - 1$ **do** ▷ local training
11:          *e.g.* gradient descent

$$w_{i,k}^{\ell+1} = w_{i,k}^\ell - \gamma\left(\nabla f_i(w_{i,k}^\ell) + \frac{1}{\rho}(w_{i,k}^\ell - v_{i,k})\right)$$

12:        **end for**
13:        set $x_{i,k+1} = w_{i,k}^{N_e}$
14:        update the auxiliary variable
$$z_{i,k+1} = z_{i,k} + 2(x_{i,k+1} - y_{k+1})$$
15:        compress $z_{i,k+1} + c_{i,k}$, $C_u(z_{i,k+1} + c_{i,k})$, and transmit to coordinator ▷ uplink EF
16:        cache the compression error
$$c_{i,k+1} = z_{i,k+1} + c_{i,k} - C_u(z_{i,k+1} + c_{i,k})$$
17:      **else** ▷ inactive agent
18:        set $x_{i,k+1} = x_{i,k}$, $z_{i,k+1} = z_{i,k}$, $c_{i,k+1} = c_{i,k}$
19:      **end if**
20:    **end for**
21: **end for**

---

compression errors $C_u(z_{i,k}) - z_{i,k}$ and $C_u(y_{k+1}) - y_{k+1}$. The goal then is to integrate an error feedback mechanism inspired by (Cheng et al., 2025) to improve accuracy. The idea is to cache the compression errors (both up- and downlink) to propagate them with later transmissions, reducing the loss of information. In particular, every time compression is applied, the error, *e.g.* $c_{i,k} = C_u(z_{i,k}) - z_{i,k}$, is stored locally, and added to the message to be transmitted in the following iteration $z_{i,k+1} + c_{i,k}$. The resulting message is compressed, $C_u(z_{i,k+1} + c_{i,k})$, its compression error is cached and the message is transmitted. The mechanism is applied to the up- and downlink at each iteration, ensuring that all information is ultimately transmitted. The resulting EF scheme is depicted in Figure 3. We remark that this scheme is actually *algorithm-agnostic* and can be directly plugged into alternative federated algorithms, denoting by $z_{i,k}$ and y_k the up- and downlink messages, respectively. In particular, the application of Figure 3 to Algorithm 1 yields Algorithm 2.

Remark 1. The Fed-LT framework allows for customization of the local training, by selecting *e.g.* (stochastic) gradient descent, noisy gradient descent for privacy preservation (Bastianello et al., 2024). Additionally, the aggregation step performed by the coordinator can be robustified against model attacks (Pasdar et al., 2025).

2.3 Algorithm design: satellite-ready version

Algorithm 2 in principle could be implemented over a satellite constellation to enable communication-efficient learning. However, even when employing compression the satellite-to-ground station communications still represent the main bottleneck, *e.g.* due to communication windows and delays. The idea then is to further modify Algorithm 2 to enhance its efficiency.

Surveying the literature on FL for satellite constellations, two main approaches stand out: 1) the ground station serves as coordinator (So et al., 2022), 2) aggregation is performed via Inter-Satellite Links (ISLs) (Zhai et al., 2024; Kim et al., 2025). The drawback of 1) is that the ground station is used continuously throughout training, requiring long-range communications, while 2) employs communications between satellites in closer range. However, 1) yields higher accuracy of the trained model. The idea then is to modify Algorithm 2 to employ both modes of communication, satellite-to-ground station and inter-satellite, balancing communication efficiency and accuracy. In particular, instead of requiring several satellites to directly communicate with the GS, we allow some satellites to receive the local updates of neighboring satellites in the same orbit and forward them to the GS. This way fewer satellite-to-ground station links need to be established, while still allowing the participation of several satellites. An additional augmentation to "space-ify" (Kim et al., 2025). Algorithm 2 comes in the form of *satellite-ready partial participation*. The idea is to select which satellites will connect to the GS and transmit updates based on the analysis of their orbits, in order to minimize the total time to execute a round of communication. The selected satellites can then serve to forward the model updates of neighboring satellites according to the mechanism described above.

The space-ified version of Algorithm 2 is reported in Algorithm 3, and is characterized by the features summarized as follows:

- the satellites collect and locally store data (*e.g.* Earth images);
- this data is processed onboard the satellites, so only trained models, and not large datasets, are communicated to Earth (lines 7-13);
- the schedule of satellite-to-GS communications is optimized to the time of a communication round (line 6);
- and in addition, satellites connected to the GS also serve to forward the models received through inter-satellite links (line 15).

2.4 Convergence analysis

In this section we analyze the convergence of the proposed Algorithm 2, which in turn establishes that of the space-ified version Algorithm 3. We make the following assumption.

Assumption 1. (Loss functions). The local loss functions in (1) are $\bar{\lambda}$-smooth and $\underline{\lambda}$-strongly convex.

This assumption ensures that (1) admits a unique solution; future work will look at convergence for non-convex

---

Algorithm 3 Space-ified Fed-LT (Fed-LTSat)

Input: For each agent initialize $x_{i,0}$ and $z_{i,0}$, and the cache $c_{i,0} = 0$; initialize the coordinator cache $c_0 = 0$; choose the local solver, its parameters, and the number of local epochs N_e; the parameter $\rho > 0$; choose the uplink C_u and downlink C_d compression; let $S_k \subset \{1, \ldots, N\}$ denote the agents active at iteration k

1: **for** $k = 0, 1, 2, \ldots,$ **do**
      // coordinator update
2:    receive $z_{i,k}$ from each agent $i \in S_k$
3:    aggregate   ▷ downlink EF

$$y_{k+1} = c_k + \frac{1}{N}\left(\sum_{i \in S_k} z_{i,k} + \sum_{i \notin S_k} z_{i,k-1}\right)$$

4:    compress y_{k+1}, $C_d(y_{k+1})$, and transmits it to all active agents   ▷ downlink compression
5:    cache the compression error $c_{k+1} = y_{k+1} - C_d(y_{k+1})$
      // agents updates
6:    select the active agents S_{k+1} using the scheduler of (Kim et al., 2025) (S_{k+1} includes satellites directly connected to the GS, and satellites connected to neighboring ones that forward their updates)   ▷ space-ification
7:    **for** $i = 1, \ldots, N$ **do**
8:      **if** $i \in S_{k+1}$ **then** ▷ partial participation
9:       set $w_{i,k}^0 = x_{i,k}$, $v_{i,k} = 2y_{k+1} - z_{i,k}$
10:       **for** $\ell = 0, 1, \ldots, N_e - 1$ **do** ▷ local training
11:        *e.g.* gradient descent

$$w_{i,k}^{\ell+1} = w_{i,k}^\ell - \gamma\left(\nabla f_i(w_{i,k}^\ell) + \frac{1}{\rho}(w_{i,k}^\ell - v_{i,k})\right)$$

12:       **end for**
13:       set $x_{i,k+1} = w_{i,k}^{N_e}$
14:       update the auxiliary variable

$$z_{i,k+1} = z_{i,k} + 2(x_{i,k+1} - y_{k+1})$$

15:       compress $z_{i,k+1} + c_{i,k}$, $C_u(z_{i,k+1} + c_{i,k})$, and transmit to the GS or to a neighboring satellite for forwarding   ▷ uplink EF
16:       cache the compression error

$$c_{i,k+1} = z_{i,k+1} + c_{i,k} - C_u(z_{i,k+1} + c_{i,k})$$

17:      **else** ▷ inactive agent
18:       set $x_{i,k+1} = x_{i,k}$, $z_{i,k+1} = z_{i,k}$, $c_{i,k+1} = c_{i,k}$
19:      **end if**
20:    **end for**
21: **end for**

---

problems. We introduce now the following definition of δ-approximate compressor (Karimireddy et al., 2019), which we apply to the uplink and downlink compression in Assumption 2.

Definition 1. (δ-approximate compressor). An operator $C: \mathbb{R}^n \to \mathbb{R}^n$ is a δ-approximate compressor if there exists $\delta \in (0, 1]$ such that

$$\|C(x) - x\|^2 \leq (1 - \delta)\|x\|^2, \quad \forall x \in \mathbb{R}^n.$$

Assumption 2. (Compressors). The uplink and downlink compressors, C_u and C_d, are δ-approximate.

We are now ready to state the following convergence result, proved in Appendix A.

Proposition 1. Let Assumptions 1 and 2 hold. Assume that each agent i is active at iteration k with probability $p_i \in (0, 1]$. Assume that $\|x_{i,k}\| \leq \beta$. Then

$$\lim_{k \to \infty} \|x_{i,k} - \bar{x}\| \leq 2\sqrt{\frac{\max_j p_j}{\min_j p_j}} \frac{1}{1-\sigma} \frac{\sqrt{1-\delta}}{\delta} \beta$$

where $\bar{x} = \arg\min_{x \in \mathbb{R}^n} \sum_{i=1}^N f_i(x)$, $\sigma \in (0,1)$ is the convergence rate characterized in (Bastianello et al., 2024, Proposition 2).

3. NUMERICAL RESULTS

In this section we evaluate the proposed algorithms applied to a classification task. We start in section 3.1 by comparing Algorithm 1 and Algorithm 2 to evaluate the impact of error feedback. Then in section 3.2 we compare Fed-LTSat (Algorithm 3) with state-of-the-art alternatives in the realistic space simulator FLySTacK (Kim et al., 2024).

The classification task to be solved throughout this section is problem (1) with local losses characterized by

$$f_i(x) = \frac{1}{m_i} \sum_{h=1}^{m_i} \log\left(1 + \exp(-b_{i,h} a_{i,h} x)\right) + \frac{\epsilon}{2N} \|x\|^2 \quad (2)$$

where $\epsilon = 50$, $m_i = 500$ for all agents, $n = 100$, and $N = 100$. The data are randomly generated. We use $N_e = 10$, and all other hyperparameters are tuned optimally using grid search.

As a performance metric we employ the optimality error characterized by $e_k = \sum_{i=1}^N \|x_{i,k} - \bar{x}\|^2$ where $\bar{x} = \arg\min_{x \in \mathbb{R}^n} \sum_{i=1}^N f_i(x)$. We also evaluate its asymptotic value e_K with K being the length of the simulation.

3.1 Evaluating the impact of error feedback

In this section we compare Fed-LT with bi-directional compression without EF (Algorithm 1) and with EF (Algorithm 2). For simplicity, we do not employ partial participation. The compression applied on both uplink and downlink is characterized by Definition 2.

Definition 2. (Uniform quantization). Let $x \in \mathbb{R}^n$ and $[x]_i$ denote its i-th component. The uniform quantization applies the function $q : \mathbb{R} \to \mathbb{R}$ component-wise:

$$q(x) = \Delta \cdot \left\lfloor \frac{x - V_{\min}}{\Delta} + 0.5 \right\rfloor + V_{\min} \quad (3)$$

where $\Delta = \frac{V_{\max} - V_{\min}}{L}$ is the step size, $V_{\max}$ and $V_{\min}$ the maximum and minimum values, and L the number of quantization levels. The result is the vector $(q([x]_1), \ldots, q([x]_n))$.

We average the error across 20 Monte Carlo simulations, each with $K = 500$ iterations, employing different values of L, $V_{\min}$ and $V_{\max}$. The results in terms of asymptotic error are reported in Table 1. We see that employing error feedback yields a significant performance improvement. We also remark that a more coarse quantization (smaller number of levels L) yields larger asymptotic optimality error, as more information is lost; although of course the communication size is smaller.

We conclude by presenting in Figure 4 an example of the optimality error evolution over the iterations of each

Table 1. Comparison with and without error feedback

	L	$V_{\min}, V_{\max}$	Asymptotic Error
Algorithm 1	1000	$-10, 10$	0.01192
	10	$-1, 1$	1.29873
Algorithm 2	1000	$-10, 10$	0.00348
	10	$-1, 1$	0.37752

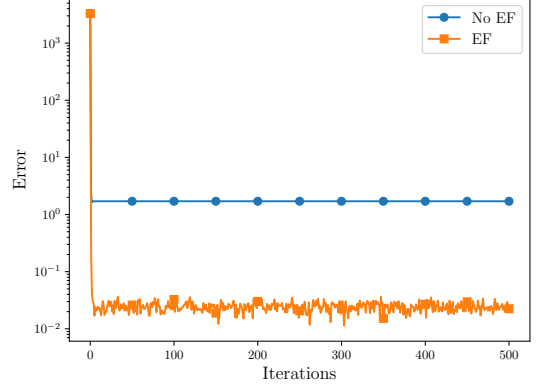algorithm. This illustrates the improvement in asymptotic optimality error brought about by using EF.

Fig. 4. Optimality error evolution over iterations, employing quantization with $V_{\min} = -1$, $V_{\max} = 1$ and $L = 10$

3.2 Comparison in a realistic space scenario

In this section we test Fed-LTSat (Algorithm 3) in the realistic space scenario of FLySTacK (Kim et al., 2024). FLySTacK simulates a satellite constellation in LEO, determining when each satellite has a communication window to the ground station (which acts as coordinator). This information is then used to schedule which satellites should send an updated model, and which should do so through inter-satellites links. In particular, we compare Fed-LTSat against the state-of-the-art alternatives Fe-dAvg (McMahan et al., 2017), FedProx (Li et al., 2020b), LED (Alghunaim, 2024), and 5GCS (Grudzień et al., 2023). These algorithms, however, were not designed for space applications, hence we space-ified them according to (Kim et al., 2025), further adding compression on the uplink and downlink communications. For a fair comparison, we apply the algorithm-agnostic error feedback scheme proposed in Figure 3, highlighting its flexibility.

We run 5 Monte Carlo simulations in the space set-up, allowing 10 satellites to participate at each round (10%). The compression applied to uplink and downlink communications is either the quantization of Definition 2 or the rand-d sparsification of Definition 3.

Definition 3. (Rand-d sparsification). Let $x \in \mathbb{R}^n$ and $[x]_i$ denote its i-th component. The rand-d sparsification is defined as follows: select uniformly at random the indices $\mathcal{I} \subset \{1, \ldots, n\}$, $|\mathcal{I}| = d$. Then return the vector $(\text{rand}([x]_1), \ldots, \text{rand}([x]_n))$ where

$$\text{rand}([x]_i) = \begin{cases} [x]_i & \text{if } i \in \mathcal{I}, \\ 0 & \text{otherwise.} \end{cases}$$

Table 2. Mean and standard deviation of the asymptotic optimality errors for the different algorithms, using 4 compressors and with 10% participation

Algorithm	Quantization ($L = 1000$, $V_{\min} = -10$, $V_{\max} = 10$)	Quantization ($L = 10$, $V_{\min} = -1$, $V_{\max} = 1$)	Rand-d ($d = 0.8n$)	Rand-d ($d = 0.2n$)
Fed-LTSat (this paper)	$\mathbf{5.91 \times 10^{-5}}$ $(\pm \mathbf{0.00 \times 10^{0}})$	$\mathbf{3.11 \times 10^{-3}}$ $(\pm \mathbf{2.77 \times 10^{-4}})$	$\mathbf{7.85 \times 10^{-3}}$ $(\pm \mathbf{2.15 \times 10^{-3}})$	1.14×10^{0} $(\pm 4.46 \times 10^{-1})$
FedAvg (McMahan et al., 2017)	4.67×10^{-4} $(\pm 7.39 \times 10^{-5})$	7.64×10^{0} $(\pm 9.48 \times 10^{-2})$	1.06×10^{0} $(\pm 6.64 \times 10^{-2})$	3.06×10^{0} $(\pm 1.47 \times 10^{-1})$
FedProx (Li et al., 2020b)	2.74×10^{-4} $(\pm 1.97 \times 10^{-5})$	1.32×10^{0} $(\pm 2.94 \times 10^{-2})$	1.40×10^{-2} $(\pm 6.45 \times 10^{-3})$	1.03×10^{0} $(\pm 3.81 \times 10^{-2})$
LED (Alghunaim, 2024)	1.63×10^{-2} $(\pm 1.38 \times 10^{-4})$	1.57×10^{0} $(\pm 0.00 \times 10^{0})$	4.51×10^{-1} $(\pm 1.88 \times 10^{-1})$	9.02×10^{-1} $(\pm 3.16 \times 10^{-5})$
5GCS (Grudzień et al., 2023)	4.09×10^{-2} $(\pm 0.00 \times 10^{0})$	3.89×10^{0} $(\pm 0.00 \times 10^{0})$	9.66×10^{-3} $(\pm 4.65 \times 10^{-3})$	$\mathbf{8.74 \times 10^{-1}}$ $(\pm \mathbf{3.63 \times 10^{-1}})$

The results of the simulations are reported in Table 2. We can see that Fed-LTSat outperforms the alternative algorithms in most scenarios, and otherwise performs very close to the alternatives. In some scenarios (*e.g.* quantization) the performance of Fed-LTSat is orders of magnitude better. When using quantization with $L = 10, V_{min} = -1, V_{max} = 1$ for example, the asymptotic optimality error, is 3 orders of magnitude lower, than any other simulated algorithm. We also remark that, similarly to the results in section 3.1, a coarser quantization/sparsification yields larger asymptotic optimality errors, due to more information being omitted from communications.

4. CONCLUSIONS AND FUTURE WORK

In this paper, we address the solution of learning problems with satellite constellations. In particular, we focus on designing a novel algorithm that ensures communication efficiency while preserving the accuracy of the trained model. We also show how to "space-ify" this algorithm for deployment in satellite constellations. We provide numerical results showcasing the performance of the proposed algorithm and comparing it with the state of the art. Future research directions include providing a broader theoretical framework for the algorithm-agnostic error feedback mechanism we proposed; and applying the proposed algorithm to non-convex problems.

Appendix A. PROOF OF PROPOSITION 1

By (Bastianello et al., 2024), we can interpret Algorithm 2 as the stochastic update

$$\begin{bmatrix} x_{i,k+1} \\ z_{i,k+1} \end{bmatrix} = \begin{cases} \begin{bmatrix} \mathcal{X}_i(x_{i,k}, z_{i,k}) \\ z_{i,k} + 2(\mathcal{X}_i(x_{i,k}, z_{i,k}) - y_{k+1}) \end{bmatrix} + e_{i,k} \\ \qquad\qquad\qquad\qquad\qquad\qquad\text{if } u_{i,k} = 1 \\ \begin{bmatrix} x_{i,k} \\ z_{i,k} \end{bmatrix} \qquad\qquad\qquad\quad \text{if } u_{i,k} = 0 \end{cases}$$
(A.1)

where $x_{i,k+1} = \mathcal{X}_i(x_{i,k}, z_{i,k})$ represents the output of the local training, $u_{i,k} \sim \text{Ber}(p_i)$ is the r.v. determining whether agent i is active, and $e_{i,k} \in \mathbb{R}^{2n}$ is the error due to compression after EF is applied.

By (Bastianello et al., 2024, Proposition 2), if $\mathbb{E}[\|e_{i,k}\|] \leq \nu$, then we know that

$$\mathbb{E}\left[\left\|\begin{bmatrix} \boldsymbol{x}_k - \bar{\boldsymbol{x}} \\ \boldsymbol{z}_k - \bar{\boldsymbol{z}} \end{bmatrix}\right\|\right] \leq \sqrt{\frac{\max_j p_j}{\min_j p_j}} \left(\sigma^k \left\|\begin{bmatrix} \boldsymbol{x}_0 - \bar{\boldsymbol{x}} \\ \boldsymbol{z}_0 - \bar{\boldsymbol{z}} \end{bmatrix}\right\| + \frac{1 - \sigma^k}{1 - \sigma}\nu\right)$$
(A.2)

where $\boldsymbol{x}$ and $\boldsymbol{z}$ are the vectors collecting x_i and z_i, respectively, and $\bar{\boldsymbol{x}} = \mathbf{1} \otimes \bar{x}$ and $\bar{\boldsymbol{z}}$ is a suitable fixed point. Therefore we need to prove that the compression error, after applying error feedback, is bounded in norm by ν.

By Jensen's inequality we have $\mathbb{E}[\|e_{i,k}\|] = \mathbb{E}[\sqrt{\|e_{i,k}\|^2}] \leq \sqrt{\mathbb{E}[\|e_{i,k}\|^2]}$ and we need to bound $\mathbb{E}[\|e_{i,k}\|^2]$. With a small modification of (Karimireddy et al., 2019, Lemma 3), using the assumption that $\|x_{i,k}\|^2 \leq \beta^2$ and that C_u, C_d are δ-approximate, yields

$$\mathbb{E}[\|e_{i,k}\|^2] \leq 4\frac{1 - \delta}{\delta^2}\beta^2. \qquad (A.3)$$

Using (A.3) into (A.2) and taking the limit $k \to \infty$ yields

$$\lim_{k \to \infty} \mathbb{E}\left[\left\|\begin{bmatrix} \boldsymbol{x}_k - \bar{\boldsymbol{x}} \\ \boldsymbol{z}_k - \bar{\boldsymbol{z}} \end{bmatrix}\right\|\right] \leq \sqrt{\frac{\max_j p_j}{\min_j p_j}} \frac{1}{1 - \sigma} \frac{\sqrt{1 - \delta}}{\delta^2}\beta.$$

The thesis follows since $\mathbb{E}[\|x_{i,k} - \bar{x}\|] \leq \mathbb{E}\left[\left\|\begin{bmatrix} \boldsymbol{x}_k - \bar{\boldsymbol{x}} \\ \boldsymbol{z}_k - \bar{\boldsymbol{z}} \end{bmatrix}\right\|\right]$. $\square$

REFERENCES

Alghunaim, S. (2024). Local Exact-Diffusion for Decentralized Optimization and Learning. *IEEE Transactions on Automatic Control*, 69(11), 7371–7386.

Aragon, B., Houborg, R., Tu, K., Fisher, J.B., and McCabe, M. (2018). CubeSats Enable High Spatiotemporal Retrievals of Crop-Water Use for Precision Agriculture. *Remote Sensing*, 10(12), 1867.

Barmpoutis, P., Papaioannou, P., Dimitropoulos, K., and Grammalidis, N. (2020). A Review on Early Forest Fire Detection Systems Using Optical Remote Sensing. *Sensors*, 20(22), 6442.

Bastianello, N., Liu, C., and Johansson, K.H. (2024). Enhancing Privacy in Federated Learning through Local Training. arXiv:2403.17572.

Cheng, Y., Shen, L., Xu, L., Qian, X., Wu, S., Zhou, Y., Zhang, T., Tao, D., and Chen, E. (2025). Communication-efficient distributed learning with local immediate error compensation. *Neural Networks*, 188, 107471.

De Luca, S. and Evroux, C. (2025). Satellites: State of play and challenges for the EU. Technical report, European Parliament. URL https://www.europarl.europa.eu/RegData/etudes/BRIE/2025/777930/EPRS_BRI(2025)777930_EN.pdf.

Franch-Pardo, I., Napoletano, B.M., Rosete-Verges, F., and Billa, L. (2020). Spatial analysis and GIS in the study of COVID-19. A review. *Science of The Total Environment*, 739, 140033.

Grudzień, M., Malinovsky, G., and Richtárik, P. (2023). Can 5th generation local training methods support client sampling? yes! In *International Conference on Artificial Intelligence and Statistics*, 1055–1092. PMLR.

Han, D.J., Hosseinalipour, S., Love, D.J., Chiang, M., and Brinton, C.G. (2024). Cooperative Federated Learning Over Ground-to-Satellite Integrated Networks: Joint

Local Computation and Data Offloading. *IEEE Journal on Selected Areas in Communications*, 42(5), 1080–1096.

Karimireddy, S.P., Rebjock, Q., Stich, S., and Jaggi, M. (2019). Error feedback fixes signsgd and other gradient compression schemes. In *International conference on machine learning*, 3252–3261. PMLR.

Kim, G., Powell, L., Svoboda, F., and Lane, N. (2024). Space for Improvement: Navigating the Design Space for Federated Learning in Satellite Constellations. arXiv:2411.00263.

Kim, G., Svoboda, F., and Lane, N. (2025). Bringing Federated Learning to Space. arXiv:2511.14889.

Li, T., Sahu, A.K., Talwalkar, A., and Smith, V. (2020a). Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine*, 37(3), 50–60.

Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2020b). Federated optimization in heterogeneous networks. In I. Dhillon, D. Papailiopoulos, and V. Sze (eds.), *Proceedings of Machine Learning and Systems*, volume 2, 429–450.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B.A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Németh, G.D., Lozano, M.A., Quadrianto, N., and Oliver, N.M. (2022). A snapshot of the frontiers of client selection in federated learning. *Transactions on Machine Learning Research*.

Pasdar, A., Liu, C., and Bastianello, N. (2025). Robust federated learning with multi-step aggregation. *IFAC-PapersOnLine*, 59(4), 181–186.

Razmi, N., Matthiesen, B., Dekorsy, A., and Popovski, P. (2022). Scheduling for Ground-Assisted Federated Learning in LEO Satellite Constellations. In *2022 30th European Signal Processing Conference (EUSIPCO)*, 1102–1106. IEEE, Belgrade, Serbia.

Shukla, S., Macharia, D., Husak, G.J., Landsfeld, M., Nakalembe, C.L., Blakeley, S.L., Adams, E.C., and Way-Henthorne, J. (2021). Enhancing access and usage of earth observations in environmental decision-making in eastern and southern africa through capacity building. *Frontiers in Sustainable Food Systems*, 5, 504063.

So, J., Hsieh, K., Arzani, B., Noghabi, S., Avestimehr, S., and Chandra, R. (2022). FedSpace: An Efficient Federated Learning Framework at Satellites and Ground Stations. arXiv:2202.01267.

Uddin, R. and Kumar, S. (2022). SDN-based Federated Learning approach for Satellite-IoT Framework to Enhance Data Security and Privacy in Space Communication. In *2022 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, 71–76. IEEE, Winnipeg, MB, Canada.

Zhai, Z., Wu, Q., Yu, S., Li, R., Zhang, F., and Chen, X. (2024). FedLEO: An Offloading-Assisted Decentralized Federated Learning Framework for Low Earth Orbit Satellite Networks. *IEEE Transactions on Mobile Computing*, 23(5), 5260–5279.

Zhao, Z., Mao, Y., Liu, Y., Song, L., Ouyang, Y., Chen, X., and Ding, W. (2023). Towards efficient communications in federated learning: A contemporary survey. *Journal of the Franklin Institute*, 360(12), 8669–8703.

Çelikbilek, K., Saleem, Z., Morales Ferre, R., Praks, J., and Lohan, E.S. (2022). Survey on Optimization Methods for LEO-Satellite-Based Networks with Applications in Future Autonomous Transportation. *Sensors*, 22(4), 1421.