

Federated Multi-Agent Mapping for Planetary Exploration

Tiberiu-Ioan Szatmari¹ and Abhishek Cauligi²

Abstract— Multi-agent robotic exploration stands to play an important role in space exploration as the next generation of robotic systems ventures to far-flung environments. A key challenge in this new paradigm will be to effectively share and utilize the vast amount of data generated onboard while operating in bandwidth-constrained regimes typical of space missions. Federated learning (FL) is a promising tool for bridging this gap. Drawing inspiration from the upcoming CADRE Lunar rover mission, we propose a federated multi-agent mapping approach that jointly trains a global map model across agents without transmitting raw data. Our method leverages implicit neural mapping to generate parsimonious, adaptable representations, reducing data transmission by up to 93.8% compared to raw maps. Furthermore, we enhance this approach with meta-initialization on Earth-based traversability datasets to significantly accelerate map convergence—reducing iterations required to reach target performance by 80% compared to random initialization. We demonstrate the efficacy of our approach on Martian terrains and glacier datasets, achieving downstream path planning F1 scores as high as 0.95 while outperforming on map reconstruction losses.

I. INTRODUCTION

In space exploration, the traditional paradigm of sending a single rover or spacecraft is giving way to new architectures that allow for faster, more efficient, and autonomous solutions that enable greater scientific returns and understanding. For example, the Martian *Ingenuity* helicopter scouted terrain in advance for the *Perseverance* rover with the aim of identifying scientific targets of interest in advance and allowing for faster traverses for the rover [1]. Indeed, such multi-agent mission concepts where teams of rovers or drones collaborate are poised to play a greater role in future exploration to more remote and far-flung planetary bodies in the Solar System [2], [3]. A pressing challenge in enabling such mission architectures is that of information sharing, where the traditional concept-of-operations involves sending data generated on-board back to ground station operators on Earth to act upon. This approach scales poorly with an increasing number of agents and can lead to costly time delays due to limited data transfer rates, both problems exacerbated by the increasing use of autonomy in planetary exploration [4].

In this work, we draw inspiration from the field of federated learning (FL) and demonstrate how FL can enable teams of planetary robots to effectively carry out science and exploration tasks in the presence of such operating constraints. FL is a learning-based approach in which statistical models are trained in a distributed fashion without transmitting data directly [5], [6]. Although FL has become popular in medical and consumer applications due privacy concerns that prevent the direct sharing of sensitive data,

¹Tiberiu-Ioan Szatmari, Department of Applied Mathematics and Computer Science, Technical University of Denmark & Eriksholm Research Centre (email: tibs@dtu.dk).

²Abhishek Cauligi is with the Jet Propulsion Laboratory, California Institute of Technology (email: abhishek.s.cauligi@jpl.nasa.gov).

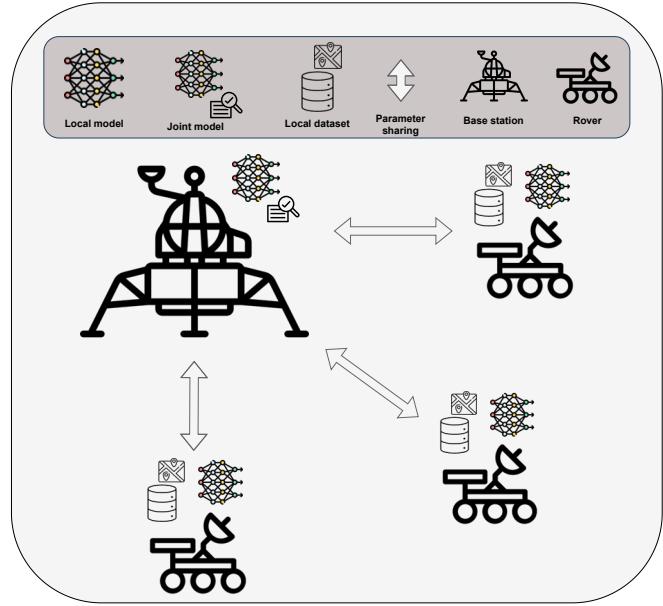


Fig. 1: In future multi-agent space exploration, balancing communication efficiency with effective collaboration is paramount. Traditional approaches that rely on sharing all raw data back to a base station quickly become infeasible due to bandwidth constraints. Federated learning addresses this by allowing rovers to learn maps and adapt their skills locally. They then share only the trained models with a base station, minimizing communication overhead. This enables efficient creation of a shared, global map representation while empowering individual rover autonomy.

it has received limited attention for applications in space robotics and planetary exploration. As such, we note that the upcoming Cooperative Autonomous Distributed Robotic Exploration (CADRE) mission ideally lends itself to the infusion of FL for the task of multi-agent distributed mapping. CADRE is a Lunar rover mission in which a team of three Lunar rovers (Figure 2) map the Lunar environment by using distributed ground penetrating radar instruments [7]. During the Exploration phase of the mission, the three rovers will compute traversability maps on-board and transmit the raw map data to a central base station, where a global map of the environment is reconciled for use in science planning.

We propose a federated learning-based approach for tackling this problem of multi-agent mapping. As shown in Figure 1, our approach targets multi-agent mission concepts such as CADRE where robots independently explore their surroundings and use on-board sensors to construct local traversability maps. Each robot supervises a neural network online using data from its own local traversability map and key to our approach is the use of implicit mapping to train a specialized 2D version of Neural Radiance Fields (NERFs) [8]. Global map construction is accomplished using federated learning, where only the trained neural network parameters from each robot are shared to the base station rather than the raw map data. This global model is distributed back to the robots for continuous refinement,



Fig. 2: CADRE rovers in a Clean Room [13].

enabling effective map generation with few-shot or even one-shot communication rounds [9], [10].

To evaluate our approach, we leverage the availability of Earth-based mapping datasets for meta-training. This process establishes a robust prior for representing map-like data. We then rigorously test the ability of our method to adapt to out-of-distribution maps simulating diverse planetary surfaces. Crucially, this demonstrates the efficacy of our proposed approach in generalizing to unforeseen and extreme environments. For validation, we employ datasets from the Athabasca Glacier in Canada [11], with its challenging icy terrain, and the DoMars16k dataset [12], which includes simulated Martian geomorphological features from locations such as Jezero Crater (the Mars 2020 landing site [4]) and Oxia Planum (the prospective ExoMars landing site).

II. RELATED WORK

A. Autonomy in Planetary Exploration

The upcoming CADRE mission exemplifies the state-of-the-art in multi-agent, spacecraft robotic autonomy. With the advent of low-power, high-performance computers such as the Qualcomm Snapdragon with integrated GPU in space robotics, missions such as CADRE [14] are capable of generating high-resolution, local traversability maps fast enough for real-time operating rates [15].

However, in a cooperative multi-agent mission, each individual agent's map offers limited situational awareness for use in science and mission planning. Instead, map information must be shared between the agents to construct a global map, whereby valuable scientific measurement targets can be identified. For CADRE, this global map construction is accomplished through the use of the MoonDB framework [16], which is a centralized database on the mission base station that stores and fuses the traversability maps generated by each agent. The key pitfall of MoonDB is that as the number of agents grow or each agent explores, the volume of data generated grows intractably and this bottlenecks global map construction.

We argue that federated learning holds significant potential for addressing such challenges. Indeed, recent years have demonstrated the efficacy of using FL to manage distributed data collected across teams of robots [17]. The authors in [18] apply FL for the task of training driver prediction models. In [18], the authors use FL to collaboratively train an imitation learning policy for robot grasping across a fleet of robots. However, applying FL to space-borne applications presents unique challenges. The extreme and diverse range of environments encountered in space missions, coupled with limited prior knowledge about their structures, makes learning particularly difficult.

B. Neural and Distributed Mapping

NeRFs [8] offer a way to create 3D representations from 2D images using deep neural networks to model volumetric density and color. A NeRF function $F_\theta : (x, d) \rightarrow (c, \sigma)$ takes a 3D location (x, y, z) and viewing direction d as input, outputting radiance (color c) and density σ . While NeRFs are used for 3D scene representation, a 2D version, also known as coordinate-based MLPs [19], applies similar principles to map image coordinates (x, y) to color value c (RGB).

In neural mapping, networks are biased toward learning low-frequency features, limiting their ability to capture high-frequency details like sharp textures [20]. To address this, mapping inputs to higher-dimensional spaces improves the modeling of high-frequency variations [21], [8]. Approaches like RNR-Map [22] use NeRFs for 3D environment embedding, aiding localization and navigation, but struggle with unseen or ambiguous areas.

Federated learning approaches presented in [23], [24] reduce communication and training costs by collaboratively training NeRFs across distributed devices. However, challenges remain in handling non-IID data, integrating diverse sensors, and adapting to dynamic environments, which are critical for multi-agent planetary exploration.

III. APPROACH

A. Overview

To address the challenges in Section II, we introduce a federated learning framework for multi-agent planetary exploration. Each agent learns local maps using 2D networks, with offline meta-training on Earth-based data. This establishes a robust prior that enables networks to rapidly learn from limited local data and viewpoints, addressing the challenges of non-IID data and ensuring local model performance, while supporting heterogeneous sensor configurations. Summarizing our contributions:

- 1) **Offline Meta-training:** Accelerates network adaptability to new maps via meta-training on traversability datasets.
- 2) **Collaborative Map Building:** Agents build a global map by exchanging learned network parameters, minimizing communication overhead.
- 3) **Map Evaluation & Refinement:** Validates refined mapping performance via image quality metrics (PSNR, SSIM) and path planning simulations (F1).

Our map refinement process uses morphological image processing (e.g. connected elements) to fill gaps and remove noise artifacts, improving map clarity. Fig. 3 illustrates the approach, and Alg. 1 provides the pseudocode.

B. Mapping the Unknown

In multi-agent space exploration, robots gather data within limited local views, resulting in incomplete global maps. Learning from this partial information can lead to inaccuracies, especially in unexplored areas. Moreover, raw coordinates (x, y) are insufficient for networks to capture the fine spatial variations needed for accurate map representation in planetary environments.

To address these challenges, we use a Gaussian encoding inspired by Fourier Feature Networks [21], transforming (x, y) coordinates into a higher-dimensional representation

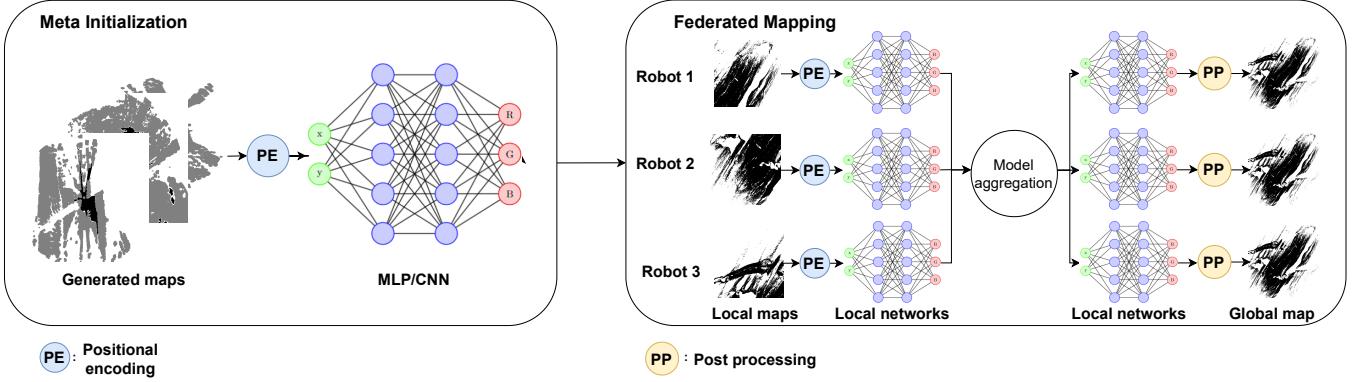


Fig. 3: Our proposed federated mapping approach begins with an offline preparation stage where a neural network is trained on an empty (unknown) grid map and then meta-trained with Reptile on a map dataset for quick adaptation (this is done only once, offline). Next, multiple agents explore within a global reference frame, generating local area maps. Each agent utilizes a neural network to learn its local map. These learned network parameters are then shared with a central server where model aggregation occurs. Finally, the updated joint model parameters, now containing global map knowledge, are distributed back to each robot. Locally, robots can further refine (e.g. remove noise) the global map generated from the updated model.

(Eq. 1), allowing the network to capture the intricate spatial details essential for space exploration. We also pre-train the network on an image representing an unexplored map (e.g., a blank image) to prevent the misinterpretation of unseen regions, ensuring the map updates accurately as data becomes available. This step, unlike meta-initialization focused on rapid adaptation, focuses on mitigating artifacts caused by the reference frames common in space missions.

The network takes positionally encoded coordinates as input and produces a map as output, where pixel values represent learned traversability information. This encoding is given by:

$$\gamma(v) = \begin{bmatrix} \cos(2\pi Bv) \\ \sin(2\pi Bv) \end{bmatrix}, \quad (1)$$

where v represents the input coordinates (x, y) and B is a matrix of random frequencies from a Gaussian distribution. The scale parameter, set to 10, adjusts the frequency matrix B , enhancing the network's ability to capture finer spatial details. We use two input channels and a mapping size of 128 to define the feature space for space exploration tasks.

C. Meta-initialization

To accelerate map learning and improve generalization for diverse terrains in space, we use a meta-initialization strategy. First, we train a neural network on the previously described empty map with encoded grid inputs. Then, using the Reptile meta-learning framework [25], we train this network on Earth-based traversability and elevation maps, such as those from the KITTI dataset [26]. We treat different categories in the dataset (urban, rural, highways, etc.) as separate tasks for meta-learning, enabling the model to generalize to new terrains. This meta-learning process aims to establish a strong prior for representing diverse map-like data, anticipating the variety of terrains encountered in space missions. Inspired by [27], this learned initialization leads to faster adaptation to new, out-of-distribution maps.

Reptile, a first-order meta-learning algorithm, finds model initialization parameters that allow rapid adaptation with few updates. It works by sampling tasks, performing gradient steps, and updating the initialization based on the difference

between initial and task-updated parameters. The update rule can be expressed as:

$$\theta \leftarrow \theta - \epsilon \sum_T (\theta - \nabla_\theta L_T(\phi_T)), \quad (2)$$

where θ represents the model parameters, T is a sampled task, and L_T is the task loss. ϵ is the step size, and ϕ_T represents task-specific updated parameters.

D. Network Definition

The network is defined as a simple convolutional architecture for 2D implicit mapping, with the default configuration consisting of four convolutional layers (kernel size 1, no padding) and an output of 3 (RGB), using ReLU activations [28] and batch normalization. This configuration functions like a fully-connected (dense) layer, making the network resemble a multi-layer perceptron (MLP). The architecture is scalable, allowing flexibility from simple linear layers to more complex convolutional ones. The parameter count is fixed, depending only on the input and output channels, not input resolution, making it independent of spatial dimensions. The network uses 256 channels to match positional encoding and is trained using the ADAM optimizer [29] with Mean Squared Error (MSE) loss.

E. Federated Mapping

Using the network from III-D, our federated mapping process starts with learning an unexplored grid (Line 2) and meta-training (Line 3), to enable swift adaptation. Each agent is initialized with meta-trained parameters (Line 4).

Agents explore independently, generating local maps (Line 6) and training the network using the ADAM optimizer and MSE loss (Line 7). Trained parameters are sent to the central server (Line 8) for aggregation (e.g., FedAvg) to create an updated global model (Line 10), which is redistributed to agents (Line 11). The global model is then used to generate a global map at each agent (Line 12), followed by gap filling and artifact removal (Line 13). Importantly, we focus on *few-shot* federated learning, aiming for effective map refinement with minimal communication rounds, iterating as agents continue exploring.

Algorithm 1 Federated Mapping with Meta-initialization

Require: M_0 an empty map, M_{global} the federated global map, D_{trav} traversability map dataset, F global reference frame

Offline Meta-Training

- 1: Train a network on the empty map
- 2: $\theta_0 \leftarrow \text{Train}(\text{Net}, M_0)$
- 3: $\theta \leftarrow \text{MetaTrain}(\text{Net}, \theta_0, D_{\text{trav}})$

Federated Map Learning

- 4: **Initialize:** Distribute meta-trained network parameters θ
- 5: **for** each agent i in parallel **do**
- 6: Explore local area, generating map M_i
- 7: $\theta_i \leftarrow \arg \min_{\theta} L(M_i, \text{Net}(\theta))$ \triangleright Local network training
- 8: Send θ_i to the central server
- 9: **end for**
- 10: $\theta_{\text{global}} \leftarrow \frac{1}{N} \sum_{i=1}^N \frac{n_k}{n} \theta_i$ \triangleright Federated Aggregation
- 11: Distribute θ_{global} to each agent
- 12: $M_{\text{global}} \leftarrow \text{Net}(\theta_{\text{global}})$ \triangleright Generate global map
- 13: $M_{\text{global}} \leftarrow \text{RefineMap}(M_{\text{global}})$ \triangleright Filter and remove noise
- 14: **Repeat** exploration, training, and aggregation as needed

F. Federated Map Refinement

In our one-shot scenario, the federated approach captures the overall environmental structure in a single communication round, minimizing data transmission between agents and the central server—a key advantage for bandwidth-limited space missions. To improve the global map, we apply efficient map refinement techniques (Fig. 6, Subfig. 6). We first scan the map to identify small gaps in navigable areas caused by sensor noise or imperfections and fill them by analyzing neighboring pixels (majority vote). This ensures continuous navigable regions for smooth path planning. Next, we remove small isolated obstacles or noise, identifying and eliminating minor pixel clusters below a certain threshold (e.g. 200) to prevent misinterpretation by path planning algorithms. These thresholds, determined through experiments on map data and fixed across our tests, reliably produce hole-free maps for downstream path planning algorithms. Should planetary terrain variations require different settings, these parameters can be dynamically adjusted.

IV. EXPERIMENTS
A. Datasets

Our evaluation strategy combines real-world Earth-based datasets for meta-initialization with simulated planetary environments for testing out-of-distribution adaptation.

For meta-initialization, we leverage the KITTI dataset [26], which includes traversability and elevation maps from urban and rural settings around Karlsruhe, Germany. KITTI provides diverse data collected using high-resolution cameras, laser scanners, and GPS. We treat different categories (Residential, Road) as separate tasks for meta-learning, creating a prior for varied terrains. While the specific categories do not occur on the Moon, they capture a broad range of surface variations—such as different textures,

TABLE I: Comparative analysis of initialization methods for map learning. We detail the PSNR after two iterations of test optimization, showing that the meta-learned initialization (Meta) outperforms other methods. Additionally, we present the avg. number of iterations required for the other methods to reach the Meta’s performance level after two iterations.

Initialization	PSNR \uparrow	# iter to match \downarrow
Random	8.16	68.68 ± 1.06
Empty map	9.37	26.98 ± 1.84
Meta	13.30	

elevation changes, and traversability cues—that provide a robust, diverse prior. We use a train/test/validation split of 2,385, 282, and 141 map samples.

For adaptability testing, we utilize the unique dataset collected by the Exobiology Extant Life Surveyor (EELS) [11] project at JPL. EELS focuses on developing “snake-like” autonomous robots to explore icy planetary bodies, a nascent area in space exploration. The dataset, collected at the Athabasca glacier in Canada, provides a large-scale point cloud converted into 2D traversability and elevation maps. The vast ice field of the glacier and crevasses simulate the extreme conditions anticipated on Europa, an icy moon of Jupiter, offering a distinctive proving ground for evaluating our approach’s ability to handle the challenges of icy world exploration. Additionally, the DoMars16k dataset [12] simulates a diverse range of Martian features, such as craters and dunes, providing a testbed for assessing our system’s generalization to the unique terrain of Mars.

This dataset strategy offers two major advantages. First, we leverage the abundance of Earth-based data for robust initialization. Second, we rigorously test the federated system’s ability to quickly adapt to new and potentially different environments to be encountered in future space exploration.

B. Metrics

We employ two main image quality metrics to evaluate our federated mapping approach. PSNR measures pixel-level differences between the reconstructed and ground truth maps, with higher values indicating better reconstruction quality. However, PSNR may not fully capture map accuracy for navigation tasks. SSIM addresses this by considering contrast, and structure, with values ranging from -1 to 1, where 1 indicates perfect similarity.

Additionally, we evaluate path planning by comparing A* paths on true versus learned maps. We assess path viability using precision, recall, and F1 scores, which provide a balanced measure of the model’s utility for autonomous navigation, as seen in prior work [30].

C. Meta-initialization Ablation Results

In Fig. 4, we compare meta-initialization with random initialization after 1000 iterations and a learning rate of $1e^{-4}$ across 200 test samples per run. The meta-initialized network quickly captures fine-grained map details, while the randomly initialized one takes significantly longer to reach similar fidelity. As shown in Table I, the meta-learned network, initiated from an empty map, achieves an average PSNR of 13.30 after just one learning step (two forward passes) on novel test maps, outperforming random initialization (8.16 PSNR) and empty map initialization (9.37

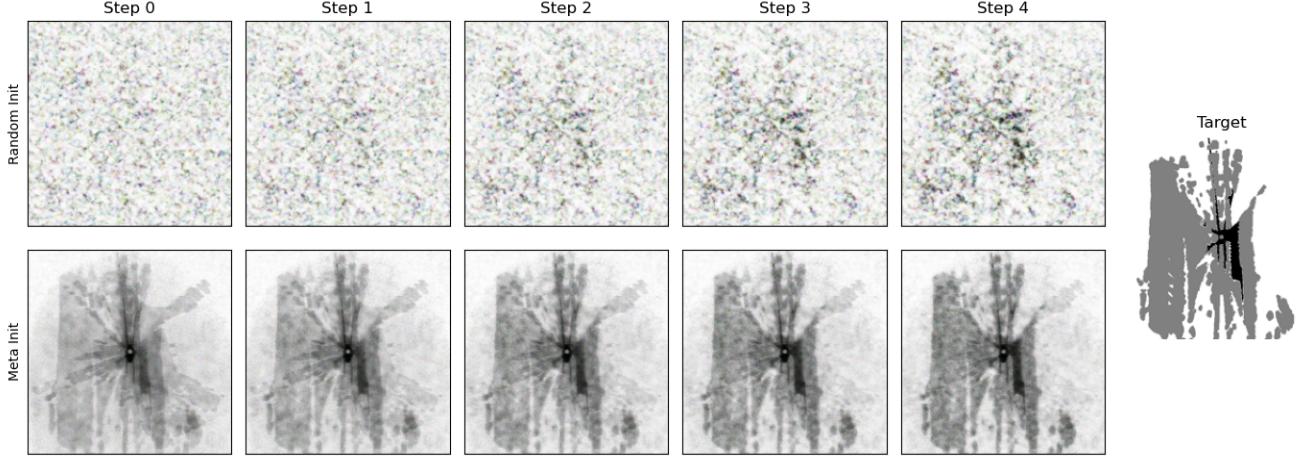
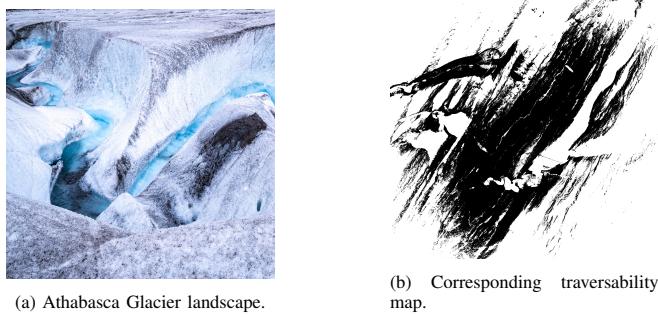


Fig. 4: Meta-initialization with Reptile enables faster network convergence for map reconstruction from KITTI data (far right). This is demonstrated by comparing a randomly initialized model (top) and its meta-initialized counterpart (bottom) from steps 0-4. Each step represents two optimization iterations.



(a) Athabasca Glacier landscape.

(b) Corresponding traversability map.

Fig. 5: Athabasca Glacier: from landscape picture to map representation.

PSNR). Random initialization requires 68.68 ± 1.06 forward passes, and empty map initialization needs 26.98 ± 1.84 to match the meta-trained network’s PSNR, demonstrating faster adaptation. Thus, we see that meta-initialization using readily available Earth-based datasets enables performant FL and fast adaptation for space-based applications.

D. Federated Mapping Results

We evaluated three federated learning algorithms—FedAvg [31], FedAdam [32] and FedYogi [32]—on multi-agent mapping in planetary exploration, focusing on their ability to handle heterogeneous, non-IID data typical of these environments. We hypothesized that FedAvg would outperform adaptive optimizers such as FedAdam and FedYogi due to challenges posed by the non-uniform data distribution across agents. FedAdam [29] and FedYogi [33] were selected to explore the potential benefits of adaptive learning rates, which adjust based on the first and second moments of the gradients. While these optimizers could theoretically offer faster convergence, they may struggle in highly diverse environments due to instability during aggregation.

To test this hypothesis, we used two out-of-distribution datasets: the Athabasca Glacier and a simulated Martian surface. As illustrated in Fig. 6, four agents explored distinct areas, learning the generated local maps ($LR = 1e^{-4}$), and sharing trained parameters in a single communication round

TABLE II: Federated mapping results on out-of-distribution datasets (Athabasca Glacier and Mars Surface) against ground truth, with performance assessed through PSNR, SSIM, and the F1 score, which is directly tied to path planning efficacy across different algorithms.

Algorithm	Athabasca Glacier			Mars Surface		
	PSNR \uparrow	SSIM \uparrow	F1 \uparrow	PSNR \uparrow	SSIM \uparrow	F1 \uparrow
FedAdam	5.27	0.31	0.1	4.4	0.25	0.03
FedYogi	10.73	0.44	0.42	9.19	0.39	0.87
FedAvg	10.85	0.47	0.95	9.26	0.42	0.94

($R = 1$). Table II provides a comprehensive evaluation of one-shot mapping performance across key metrics: PSNR (map reconstruction quality), SSIM (structural similarity) and F1 scores (path planning efficacy), using consistent settings (4 agents, 100 local epochs).

(1) FedAdam integrates Adam’s adaptive learning rates into federated learning but performs poorly in our experiments, with low PSNR, SSIM, and near-zero F1 scores. The main issue is likely its sensitivity to gradient differences across agents. In highly heterogeneous surfaces such as the glacier crevasses, where local data distributions differ significantly, the adaptive adjustments seem to have caused instability during aggregation, leading to excessive or insufficient learning rate adjustments for effective global models. (2) FedYogi, an adaptation of the Yogi optimizer for federated settings, improves over FedAdam, especially on the Mars dataset with an F1 score of 0.87. However, it still struggled with the high data heterogeneity, as the learning rate adjustments were still too reactive to the large variations in local environments. (3) FedAvg outperforms both adaptive methods, achieving the highest PSNR (10.85), SSIM (0.47), and F1 scores (0.95 on Athabasca Glacier, 0.94 on Mars). These results suggest that FedAvg’s simpler approach of averaging local updates, without dynamic learning rate adjustments, proved more stable in the tested environments, leading to superior map reconstruction and path planning across diverse terrains. In contrast, adaptive methods like FedAdam and FedYogi struggled with instability during model aggregation, making them less effective for federated learning in multi-agent planetary exploration.

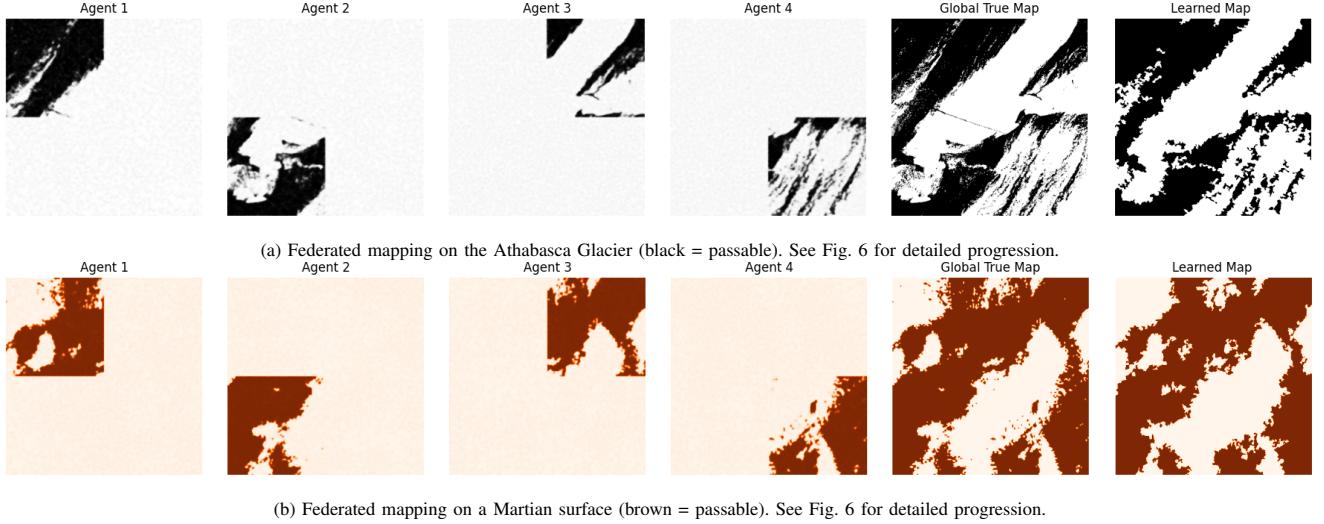


Fig. 6: Federated map learning results for both the Athabasca Glacier (top row) and a Martian surface (bottom row). For both, the figure progresses as follows: the first four subfigures display the local maps learned by individual agents. Subfigure 5 shows the ground truth global map. Subfigure 6 presents the final global map obtained by combining the agent maps through federated learning.

TABLE III: Comparison of Data Transmission Sizes between Raw Map Data and Model Parameters

Map Resolution	Data Size	Model Size	Reduction
<i>Grayscale Map</i>			
2,000 × 2,000 pixels	3.81 MB	399 KB	89.53 %
- As JPEG (Q=26)	402 kB	399 KB	0.7%
<i>OMG Map</i>			
400 × 400 cells	960 kB	399 KB	58.4 %
600 × 600 cells	2.16 MB	399 KB	81.5 %
<i>CADRE CCM</i>			
400 × 400 cells	2.88 MB	399 KB	86.1 %
600 × 600 cells	6.48 MB	399 KB	93.8 %

E. Data Transfer Comparison

Our federated mapping approach significantly reduces data transmission size compared to sending raw map data. Table III compares raw data sizes with the neural network parameters transmitted in our method. The network has approximately 200 thousand parameters: 65,792 per layer for Layers 1-3 (including biases), 512 per BatchNorm layer, and 771 for the final layer. Each parameter is stored as a 16-bit (2-byte) value, making the total model size approximately 399 KB, with the potential for further quantization.

Transmitting the full model parameters requires approx. 399 KB. This is already more bandwidth-efficient than sending raw map data for any instance of the grayscale, optimal mixture of gaussians (OMG) map proposed for Mars rotorcrafts [34], or the converging covariance map (CCM) used for CADRE [7]. In particular, we see that the CADRE CCM requires approximately 2.88 MB for a 400×400 cell map (18 bytes/cell). We note that this comparison is for a single CCM map, whereas the Exploration phase of the CADRE mission entails computing multiple CCMs as the rover drives and each one of these CCM instances is transmitted back to the base station for map merging. In addition, JPEG with quality factor (Q=26) can perform lossy compression for a raw grayscale map (2000×2000) to 402 kB -nearly matching the full model parameters. In this configuration, it

can yield high quality metrics (PSNR: 25.88, SSIM: 0.90), and a similar downstream path planning F1 score of 0.96.

However, our approach stands apart from conventional image compression like JPEG because the model can leverage gradient sparsification. Techniques such as top-k selection, as demonstrated in Deep Gradient Compression (DGC) [35], can reduce the data transmitted by roughly 99% compared to full gradient updates for CNN-based ResNet architectures. In our task, where the full model is approximately 399 kB, this implies that only about 1%—roughly 4 kB—needs to be sent per update round. This dramatic reduction in update size significantly outperforms traditional JPEG compression. Moreover, the meta-initialization strategy functions similarly to the warm-up training phase in DGC. This warm-up establishes a strong prior, such that that early updates are less noisy and more stable. Given the relatively stable nature of space environments, where maps exhibit only gradual changes, our network could produce extremely sparse incremental updates.

F. Downstream Path Planning

We evaluated the practical utility of our federated mapping by running A* path planning simulations on 75 randomly selected routes from real terrain data. In this context, a path refers to the route calculated by the A* algorithm, connecting a validated start point to an end point on either the true or learned map. Each path consists of a sequence of navigable waypoints that the robot follows while avoiding obstacles and traversing safe terrain. We compared the paths generated from the learned global maps against ground-truth traversability maps. By comparing these paths, we assessed how accurately the learned maps supported feasible route planning. Precision, recall, and F1 score were used to quantify the model’s ability to identify navigable paths and avoid obstacles. While PSNR and SSIM indicated map quality, the F1 score was key in evaluating path planning effectiveness, showing a strong correlation with improved path planning and safer navigation.

These results demonstrate the benefits of federated multi-

agent mapping, even in a one-shot scenario with minimal communication. The integration of local maps into a global traversability map enables reliable autonomous navigation, crucial for bandwidth-limited space missions.

V. CONCLUSION

In this paper, we introduce a federated multi-agent mapping strategy for space exploration robotics, utilizing implicit neural mapping for efficient and detailed map creation. This method supports rapid adaptation in diverse environments through meta-initialization and collaborative learning. Validation on mission-like datasets confirms its effectiveness. In future work, we aim to apply FL for mission concepts in which information must be shared across heterogeneous agents (e.g., a rover and a helicopter). Finally, we plan to extend our approach to 3D mapping that capture richer scene representations beyond traversability maps, further enhancing the compression benefits.

ACKNOWLEDGMENT

The research was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA (80NM0018D0004). The authors thank Jean-Pierre de la Croix, Georgios Georgakis, Shehryar Khatkak, Mike Paton, Jakob Eg Larsen and Niels H. Pontoppidan for their discussions during the development of this work.

REFERENCES

- [1] J. L. Anderson, T. L. Brown, M. Cacan, G. Kubiak, A. Jasour, and N. Z. Rothenberger, “Lessons from Ingenuity’s climb up Jezero Crater Delta,” in *IEEE Aerospace Conference*, 2024.
- [2] M. J. Schuster, M. G. Müller, S. G. Brunner, H. Lehner, and others, “The ARCHES space-analogue demonstration mission: Towards heterogeneous teams of autonomous robots for collaborative scientific sampling in planetary exploration,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5315–5322, 2020.
- [3] F. Rossi, M. Saboia, and J. Krishnamoorthy, Vander Hook, “Proximal exploration of Venus volcanism with teams of autonomous buoyancy-controlled balloons,” *Acta Astronautica*, vol. 208, no. 2, pp. 389–406, 2023.
- [4] V. Verma, M. W. Maimone, D. M. Gaines, R. Francis, T. A. Estlin, S. R. Kuhn, G. R. Rabideau, S. A. Chien, M. McHenry, E. J. Graser, A. L. Rankin, and E. R. Thiel, “Autonomous robotics is driving Perseverance rover’s progress on Mars,” *Science Robotics*, vol. 8, no. 80, pp. 1–12, 2023.
- [5] H. B. McMahan, E. Moore, D. Ramage, and A. Agüera y Arcas, “Federated learning of deep networks using model averaging,” 2016. Available at <https://arxiv.org/pdf/1602.05629v1/1000.pdf>.
- [6] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.
- [7] J.-P. de la Croix, F. Rossi, R. Brockers, D. Aguilar, K. Albee, E. Boroson, A. Cauligi, J. Delaune, and others, “Multi-agent autonomy for space exploration on the CADRE Lunar technology demonstration mission,” in *IEEE Aerospace Conference*, 2024.
- [8] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “NeRF: representing scenes as neural radiance fields for view synthesis,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [9] Y. Park, D.-J. Han, D.-Y. Kim, J. Seo, and J. Moon, “Few-round learning for federated learning,” in *Conf. on Neural Information Processing Systems*, 2021.
- [10] Y. Zhou, G. Pu, X. Ma, X. Li, and D. Wu, “Distilled one-shot federated learning,” 2021. Available at <https://arxiv.org/pdf/2009.07999.pdf>.
- [11] R. Thakker, M. Paton, B. Jones, G. Daddi, R. Royce, M. R. Swan, M. Swan, and others, “To boldly go where no robots have gone before - part 4: NEO autonomy for robustly exploring unknown, extreme environments with versatile robots,” in *AIAA Scitech Forum*, 2024.
- [12] T. Wilhelm, M. Geis, J. Pütschneider, T. Sievernich, T. Weber, K. Wohlfarth, and C. Wöhler, “DoMars16k: A diverse dataset for weakly supervised geomorphologic analysis on Mars,” *Remote Sensing*, vol. 12, no. 23, p. 3981, 2020.
- [13] NASA-JPL, “Cadre rovers, hardware in jpl clean room,” 2024. [Online; accessed Jan 2024].
- [14] J.-P. de la Croix, F. Rossi, R. Brockers, D. Aguilar, K. Albee, E. Boroson, A. Cauligi, J. Delaune, R. Hewitt, D. Kogan, et al., “Multi-agent autonomy for space exploration on the cadre lunar technology demonstration,” in *2024 IEEE Aerospace Conference*, pp. 1–14, IEEE, 2024.
- [15] L. Werner, P. Proençā, A. Nüchter, and R. Brockers, “Covariance based terrain mapping for autonomous mobile robots,” in *Proc. IEEE Conf. on Robotics and Automation*, 2024.
- [16] M. Saboia, F. Rossi, V. Nguyen, G. Lim, D. Aguilar, and J.-P. de la Croix, “CADRE MoonDB: Distributed database for multi-robot information-sharing and map-merging for lunar exploration,” in *Proc. Int. Conf. on Autonomous Agents and Multiagent Systems*, 2024.
- [17] Y. Xianjia, J. P. Queraltá, J. Heikkonen, and T. Westerlund, “Federated learning in robotic and autonomous systems,” *Procedia Computer Science*, vol. 191, pp. 135–142, 2021.
- [18] L. Wang, K. Zhang, A. Zhou, M. Simchowitz, and R. Tedrake, “Robot fleet learning via policy merging,” in *Int. Conf. on Learning Representations*, 2024.
- [19] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, “Implicit neural representations with periodic activation functions,” in *Conf. on Neural Information Processing Systems*, 2020.
- [20] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, “On the spectral bias of neural networks,” in *Int. Conf. on Machine Learning*, 2019.
- [21] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, “Fourier features let networks learn high frequency functions in low dimensional domains,” in *Conf. on Neural Information Processing Systems*, 2020.
- [22] O. Kwon, J. Park, and S. Oh, “Renderable neural radiance map for visual navigation,” in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2023.
- [23] L. Holden, F. Dayoub, D. Harvey, and T.-J. Chin, “Federated Neural Radiance Fields,” 2023. Available at <https://arxiv.org/pdf/2305.01163.pdf>.
- [24] T. Suzuki, “Federated learning for large-scale scene modeling with Neural Radiance Fields,” 2023. Available at <https://arxiv.org/pdf/2309.06030.pdf>.
- [25] A. Nichol and J. Schulman, “Reptile: a scalable metalearning algorithm,” 2018. Available at <https://arxiv.org/pdf/1803.02999.pdf>.
- [26] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” *Int. Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [27] M. Tancik, B. Mildenhall, T. Wang, D. Schmidt, P. P. Srinivasan, J. T. Barron, and R. Ng, “Learned initializations for optimizing coordinate-based neural representations,” in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2021.
- [28] A. B. Agarap, “Deep learning using rectified linear units (ReLU),” 2018. Available at <https://arxiv.org/pdf/1803.08375.pdf>.
- [29] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in *Int. Conf. on Learning Representations*, 2015.
- [30] Y. Qian, X. Wang, Z. Chen, C. Wang, and M. Yang, “Hy-seg: A hybrid method for ground segmentation using point clouds,” *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1597–1606, 2022.
- [31] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *AI & Statistics*, 2017.
- [32] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, “Adaptive federated optimization,” 2021. Available at <https://arxiv.org/pdf/2003.00295.pdf>.
- [33] M. Zaheer, S. Reddi, D. Sachan, S. Kale, and S. Kumar, “Adaptive methods for nonconvex optimization,” *Advances in neural information processing systems*, vol. 31, 2018.
- [34] P. F. Proençā, J. Delaune, et al., “Optimizing terrain mapping and landing site detection for autonomous uavs,” in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 9668–9674, IEEE, 2022.
- [35] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, “Deep gradient compression: Reducing the communication bandwidth for distributed training,” *arXiv preprint arXiv:1712.01887*, 2017.