Collaborative Exploration with a Marsupial Ground-Aerial Robot Team through Task-Driven Map Compression

Angelos Zacharia, Mihir Dharmadhikari, and Kostas Alexis

***Abstract*—Efficient exploration of unknown environments is crucial for autonomous robots, especially in confined and large-scale scenarios with limited communication. To address this challenge, we propose a collaborative exploration framework for a marsupial ground-aerial robot team that leverages the complementary capabilities of both platforms. The framework employs a graph-based path planning algorithm to guide exploration and deploy the aerial robot in areas where its expected gain significantly exceeds that of the ground robot, such as large open spaces or regions inaccessible to the ground platform, thereby maximizing coverage and efficiency. To facilitate large-scale spatial information sharing, we introduce a bandwidth-efficient, task-driven map compression strategy. This method enables each robot to reconstruct resolution-specific volumetric maps while preserving exploration-critical details, even at high compression rates. By selectively compressing and sharing key data, communication overhead is minimized, ensuring effective map integration for collaborative path planning. Simulation and real-world experiments validate the proposed approach, demonstrating its effectiveness in improving exploration efficiency while significantly reducing data transmission.**

***Index Terms*—Cooperating Robots, Motion and Path Planning**

I. Introduction

ADVANCEMENTS in robotic systems have facilitated their deployment across a wide range of autonomous missions. Both aerial and ground robots are now extensively utilized for various applications, including search and rescue [1], surveillance [2], inspection [3], and exploration [4].

Efficiently exploring unknown environments remains a significant challenge, whether navigating confined indoor spaces or traversing expansive outdoor landscapes. Single-robot systems often encounter limitations in terms of speed, sensing range, and their ability to navigate complex terrains. These challenges can be effectively addressed through the deployment of heterogeneous robot teams, where the strengths of one robot complement the limitations of the others. Marsupial ground-aerial systems exemplify a collaborative approach, combining the robustness and load capacity of ground robots with the agility and three-dimensional mobility of aerial robots. This synergy enables the strategic deployment of the aerial robot to efficiently explore large open spaces and high-ceiling environments, providing rapid coverage and accessing areas beyond the ground robot's reach for seamless mapping.

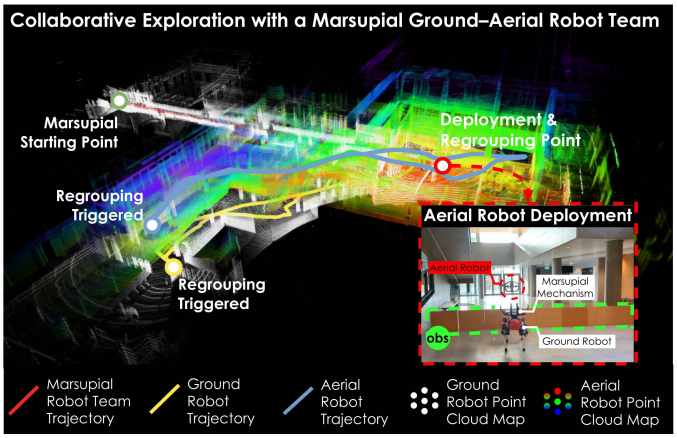

Fig. 1. Combined map from a real-world experiment, where the marsupial ground–aerial robot team collaborated in exploration. The aerial robot took over after a physical obstacle (obs) prevented the ground robot's progress.

Collaborative exploration relies on efficient data sharing between robots. Real-time exchange of sensor data, maps, and positions enables coordination and prevents redundant work. However, differences in processing power, sensor types, and communication constraints—such as limited bandwidth, high latency, and intermittent links—pose significant challenges, especially in complex environments.

To address these challenges, this paper presents a comprehensive framework that integrates planning and deployment strategies for a marsupial ground-aerial robot team. Unlike prior work that either focuses on communication-efficient compression of generic point cloud data or on heterogeneous robot coordination without scalable data sharing, our contributions are twofold. First, we propose a bandwidth-efficient, task-driven point cloud compression method tailored for volumetric map reconstruction at mission-relevant resolutions. By emphasizing occupancy-relevant structure over raw point cloud fidelity, our approach achieves high compression rates while retaining the information essential for planning. This method is open-sourced at https://github.com/ntnu-arl/pcl-vae/. Second, we introduce a decentralized collaborative exploration framework that leverages the complementary capabilities of a marsupial robot team. It features an aerial robot deployment strategy, keyframe-based map sharing for coordinated planning, and an energy-aware regrouping strategy. The framework is validated in large-scale simulations and real-world experiments, demonstrating improvements in both exploration efficiency and bandwidth usage. More experimental results from the real-world trials can be found at

Manuscript received: March, 21, 2025; Revised June, 18, 2025; Accepted September, 02, 2025.

This paper was recommended for publication by Editor Olivier Stasse upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the European Commission Horizon Europe grants SYNERGISE (EC 101121321), SPEAR (EC 101119774) and DIGIFOREST (EC 101070405).

The authors are with the Autonomous Robots Lab, Norwegian University of Science and Technology (NTNU), Norway angelos.zacharia@ntnu.no

Digital Object Identifier (DOI): see top of this page.

https://ntnu-arl.github.io/marsupial-collaborative-exploration/

The remainder of this paper is organized as follows. Section II reviews related work on marsupial robot teams and map compression and sharing techniques. Section III details the proposed framework, followed by evaluation studies in Section IV. Finally, Section V concludes the paper.

II. Related Work

A. Marsupial Systems-of-Systems

Research in marsupial robotics, particularly ground/aerial systems, has historically been sparse. However, recent advancements have emerged, notably through the efforts of several teams involved in the DARPA Subterranean Challenge [5]. The ground platforms primarily consisted of tracked vehicles [6], rovers [7], or legged robots [8], while the aerial platforms were predominantly multirotor systems. Focusing on combined docking-and-recharging, the work in [9] integrated a VTOL drone with a quadruped robot. Arguably the most well-known example, the Ingenuity helicopter, has completed multiple missions on Mars after being ferried and launched by the Perseverance rover [10].

Beyond system design, a set of works relates to the problem of planning for marsupial systems. These include works on path and trajectory planning of tethered aerial-ground systems [11], [12], and stochastic assignment for the deployment of multiple marsupial robots [13]. The deployment of marsupial robots for multi-agent exploration is studied in [8], [14], [15]. In this framework, the issue of communication constraints has repetitively attracted attention [5], [14].

B. Point Cloud Compression and Map Sharing

A set of methods have been developed to enable efficient point cloud compression, including both conventional and neural strategies. Exploiting predictive deep learning models and leveraging the image representation of LiDAR data, RIDDLE [16] achieves high-degree of compression. The contribution in [17] exploits recurrent neural networks for efficient compression, while the work in [18] leverages a convolutional autoencoder learning compact feature descriptors from point clouds. A survey on deep learning-based point cloud compression is presented in [19]. Targeting autonomous driving, [20] utilizes range image-based segmentation and clustering to reduce spatial redundancy, with video coding enhancing compression. [21] exploits spatial and temporal redundancies in point clouds for real-time, high-efficiency compression.

Beyond the general application of compression, a niche body of work exists focusing on compression for map sharing in multi-robot operations [22]. RecNet [23] transforms 3D point clouds into compact range image embeddings for efficient encoding and sharing while it serves both the goal of place recognition tasks and collaborative mapping in resource-constrained settings. The work in [24] first maps 3D point clouds into panoramas, uses event-triggered updates, and applies frequency-domain point cloud compression for efficient multi-robot systems. Departing from the current state-of-the-art, this work prioritizes a high degree of point cloud compression by encoding into a latent representation that explicitly focuses on the information necessary to reconstruct the occupancy map for planning and collision avoidance. While prior methods focus on compression ratios in the order of $10\times$ - $80\times$ [23]–[25], our approach targets and achieves $300\times$ compression, enabling efficient map sharing for collaborative exploration in communication-constrained environments.

III. Proposed Approach

This section presents the proposed methodology utilized by a marsupial robot team to collaboratively explore unknown environments through efficient, task-driven map compression. The team consists of a ground (legged) robot and an aerial robot in a marsupial configuration, where the ground robot serves as a carrier platform for the aerial system. Both robots perform graph-based exploration path planning, with the ground robot additionally assessing the deployment of the aerial robot using an exploration gain mechanism. Key to the proposed approach is a bandwidth-efficient map-sharing solution that enables the receiving robot to reconstruct the volumetric information acquired by the transmitting robot via the inter-robot communication network. This reconstruction allows the receiving robot to plan based on the volumetric map that integrates both its locally observed data and the shared information. To achieve this, each robot first compresses a selective subset of the point cloud data acquired by its onboard sensors, which is then transmitted along with the associated estimated pose transformations (keyframes). An overview of the proposed approach is presented in Figure 2.

A. Task-Driven Point Cloud Compression for Volumetric Mapping

Unlike conventional point cloud compression methods aimed at reconstructing raw input, we propose a task-specific solution focused on reconstructing volumetric information at a defined voxel resolution. This enables high compression rates by filtering out spatially insignificant details through voxelization, aligned with mission requirements. The proposed compression pipeline for LiDAR range data follows a two-step procedure: i) a remapping step that accounts for voxel map integration to retain only task-relevant information, and ii) a custom-trained Variational Autoencoder (VAE) architecture that jointly remaps and encodes the input range image, along with a corresponding decoder. The proposed architecture is shown in Figure 3.

1) Voxel-aware Range Image Generation: Range images and their associated point clouds capture intricate surface geometries, especially with modern high-resolution LiDAR sensors. However, such geometric detail imposes a challenge for compression: high-frequency features can dominate the latent representation $\mathbf{z}$, consuming capacity that would otherwise encode semantically meaningful structures. To address this, we adopt a task-specific approach that preserves only the information needed to reconstruct an accurate occupancy map, rather than the full raw geometry.

We introduce a preprocessing step that converts each range image $\mathbf{x}$ in the training set $\mathbb{D}$ into a voxel-aware version $\mathbf{x}^{vxl} \in \mathbb{D}^{vxl}$, which serves as the VAE training target. Each pixel (i, j) in $\mathbf{x}$ is projected into 3D space using LiDAR

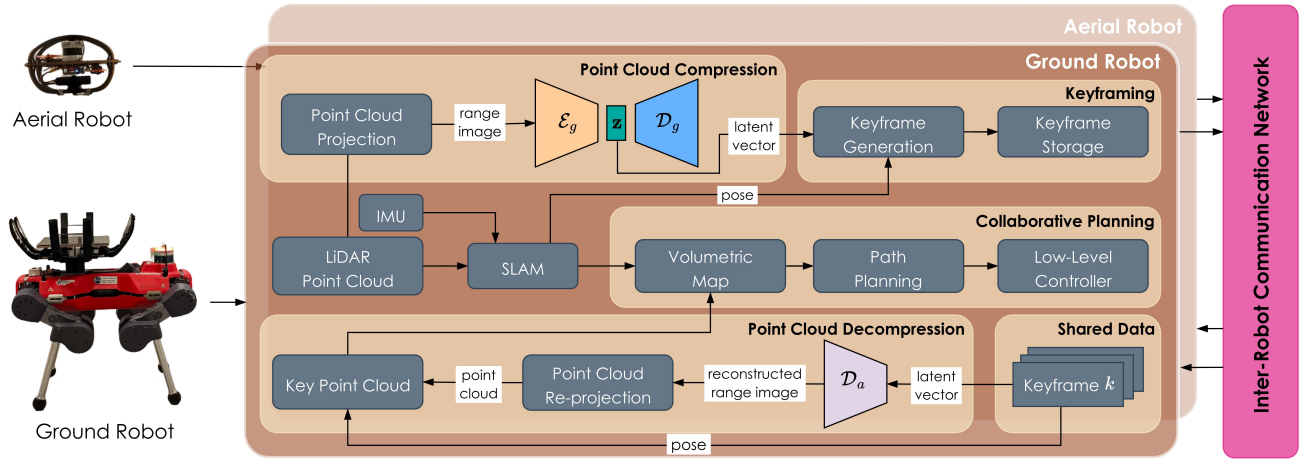

Fig. 2. Overview of the proposed collaborative exploration framework with bandwidth-efficient map sharing, employed by a marsupial ground-aerial robot team. As the ground robot explores while carrying the aerial robot, it continuously evaluates whether to deploy the aerial robot by comparing their respective exploration gains. At the same time, it compresses sparse point clouds, generating and storing keyframes. Upon deployment, a subset of keyframes is shared with the aerial robot to initialize its volumetric map. Both robots then explore independently, exchanging keyframes bidirectionally over the communication network. Each robot runs its own encoder to generate keyframes and uses the other's decoder for decompression. After a predetermined duration, both robots return to the deployment point, concluding the mission once all keyframes have been exchanged.

intrinsics and spherical-to-Cartesian conversion. The resulting 3D point cloud populates a voxelized occupancy map $\mathbf{O} \in \{0,1,2\}^{N_x \times N_y \times N_z}$ with resolution-specific voxelization s_{vxl}, labeling voxels as free (0), occupied (1), or unknown (2) via ray casting. Rays from the LiDAR origin mark traversed voxels as free and endpoints as occupied; untouched voxels remain unknown. To return to a 2D form, we re-trace each ray and assign the pixel in $\mathbf{x}^{vxl}$ the distance to the first occupied voxel it intersects, or mark it unknown if none is found. The resulting image appears discretized but retains meaningful structure, filtering out irrelevant fine details while preserving volumetric information critical for navigation and planning. This pipeline is efficiently implemented using NVIDIA Warp for large-scale GPU processing and performs the following operations:

$$\forall\, \mathbf{x} \in \mathbb{D} \underset{\text{projection}}{\mapsto} \mathbf{O}(\mathbf{x}, s_{vxl}) \underset{\text{re-projection}}{\mapsto} \mathbf{x}^{vxl} \in \mathbb{D}^{vxl}. \quad (1)$$

2) Voxel-aware Range Image Compression: Motivated by the overall success of VAEs and literature in task-driven compression for collision images [26], we utilize the VAE architecture depicted in Figure 3 to learn how to simultaneously remap and compress the input raw range images such that their voxel-aware form can be reconstructed faithfully through a particularly lightweight latent space.

Let $\mathbf{x} \in \mathbb{D}$ represent a range image and $\mathbf{x}^{vxl} \in \mathbb{D}^{vxl}$ denote its corresponding voxel-aware range image, derived by applying the operations in Eq. (1). To enable efficient dimensionality reduction of the input range image, we employ a probabilistic encoding-decoding framework that leverages the expressive power of Deep Neural Networks (DNNs) for effective compression and simultaneously learning the voxel-aware remapping. The probabilistic decoder $p_\theta(\mathbf{x}^{vxl}|\mathbf{z})$ generates a distribution over all possible values of $\mathbf{x}^{vxl}$, given the latent representation $\mathbf{z}$ with dimensions $N_\mathbf{z}$. Analogously, the probabilistic encoder $q_\phi(\mathbf{z}|\mathbf{x})$ learns to simultaneously encode and remap the raw range image $\mathbf{x}$ to a latent distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^{N_\mathbf{z}}$ and standard deviation $\boldsymbol{\sigma} \in (\mathbb{R}^+)^{N_\mathbf{z}}$. This distribution is then sampled using the reparameterization trick, $\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$ ($\odot$ is the element-wise multiplication operator) where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}_{N_\mathbf{z}}, \mathbf{I}_{N_\mathbf{z}})$ [27]. Joint training of the encoder and decoder networks is guided by the loss function

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \beta_{\text{norm}} \mathcal{L}_{\text{KL}} \quad (2)$$

where $\mathcal{L}_{\text{recon}}$ denotes the reconstruction loss and $\mathcal{L}_{\text{KL}}$ represents the KL-divergence loss, both defined as

$$\mathcal{L}_{\text{recon}}(\mathbf{x}^{vxl}, \mathbf{x}^{vxl}_{\text{recon}}) = \text{MSE}(\mathbf{x}^{vxl}, \mathbf{x}^{vxl}_{\text{recon}}) \quad (3)$$

$$\mathcal{L}_{\text{KL}}(\boldsymbol{\mu}, \boldsymbol{\sigma}) = -\frac{1}{2} \sum_{n=1}^{N_\mathbf{z}} (1 + \log(\boldsymbol{\sigma}_n^2) - \boldsymbol{\mu}_n^2 - \boldsymbol{\sigma}_n^2). \quad (4)$$

The reconstruction loss is measured as the Mean Squared Error (MSE) between the voxel-aware range image $\mathbf{x}^{vxl}$ and the reconstructed output $\mathbf{x}^{vxl}_{\text{recon}}$, excluding contributions from invalid pixels in the range image to ensure they do not affect the loss calculation. The KL-divergence loss balances the trade-off between reconstruction quality and latent space regularization by ensuring that the posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$ remains close to a predefined prior $p(\mathbf{z})$, modeled as a standard Gaussian $\mathcal{N}(\mathbf{0}_{N_\mathbf{z}}, \mathbf{I}_{N_\mathbf{z}})$ [28]. The contribution of KL-divergence loss is adjusted by the tunable hyperparameter $\beta_{\text{norm}} = \frac{\beta \cdot N_\mathbf{z}}{H \cdot W}$, where $\beta = 1$, and H and W are the height and width of the range image, respectively.

In the neural architecture of the trained compression model, the encoder comprises five convolutional layers with ELU activation functions, designed to progressively reduce the spatial dimensions of $\mathbf{x}$ while increasing feature richness. At the final stage of the encoder, a fully connected layer generates two output vectors representing the mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$, which parameterize the latent space distribution. The decoder adopts a symmetric structure to the encoder, beginning with a fully connected layer that transforms the latent vector $\mathbf{z}$ into an intermediate feature representation. This is followed by five deconvolutional layers with ReLU activation functions except for the final layer, which employs a sigmoid activation function. The latter ensures that the reconstructed data matches the range of the original data. This architecture (Fig. 3)

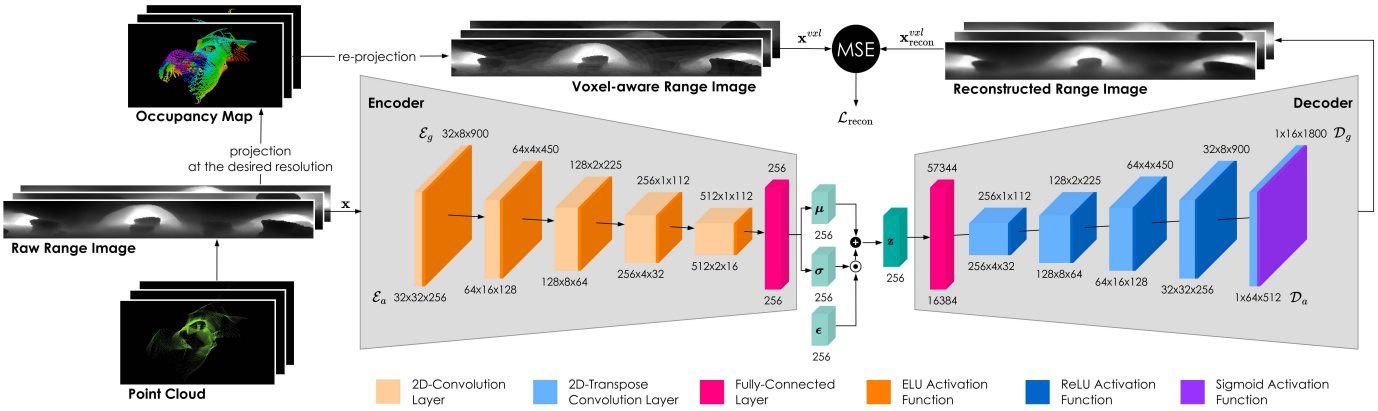

Fig. 3. The proposed network architecture is tailored to compressing and remapping the range image $\mathbf{x}$ to a latent vector $\mathbf{z}$, which is then used to generate the reconstructed voxel-aware range image $\mathbf{x}^{vxl}_{\text{recon}}$. The encoder-decoder scheme for ground robot $\{\mathcal{E}_g,\mathcal{D}_g\}$ and for the aerial robot $\{\mathcal{E}_a,\mathcal{D}_a\}$ consists of convolutional and fully connected layers along with activation functions. The output shape of each layer is indicated in the format $C \times H \times W$, representing the number of channels (C), height (H), and width (W), respectively.

achieves $2\times$ faster inference time compared to [29], through parameter reduction and the removal of residual layers, while maintaining effective compression and reconstruction.

B. Collaborative Exploration

Effective exploration of complex environments requires leveraging the strengths of heterogeneous robot teams. To fully utilize the complementary capabilities of a marsupial ground-aerial robot team, we propose a collaborative exploration path planning framework, with three phases: i) pre-deployment, ii) deployment, and iii) post-deployment, as detailed in this section. Pseudocode is provided in Algorithm 1.

The proposed framework extends GBPlanner [4], a graph-based exploration planner with local and global modules using Voxblox [30] for volumetric mapping. The local planner samples 3D points to build a dense graph $\mathbb{G}_L$ and computes shortest paths Σ_L via Dijkstra's algorithm. For ground robots, samples are projected into 2.5D to respect locomotion constraints. The best path $\sigma_{L,\text{best}}$ is selected based on exploration gain $\phi_{L,\text{best}}$. If no informative path is found, the global planner builds a sparse graph $\mathbb{G}_G$ to guide exploration and ensure return-to-home within endurance limits. Both robots independently run GBPlanner as their exploration strategy.

Pre-Deployment Phase: In this phase, the ground robot carries the aerial robot and evaluates the need for deployment during each planning iteration to enhance exploration. Specifically, the ground robot constructs a 2.5D local graph $\mathbb{G}^g_L$ and identifies the optimal path $\sigma^g_{L,\text{best}}$ along with the corresponding exploration gain $\phi^g_{L,\text{best}}$ (Algo. 1, lines 5–6) [4]. In parallel, it generates a virtual 3D local graph $\bar{\mathbb{G}}^a_L$, which approximates the graph the aerial robot would construct if it were deployed. Based on the aerial robot's sensor specifications, the ground robot selects the vertex in $\bar{\mathbb{G}}^a_L$ with the highest exploration gain $\bar{\phi}^a_{L,\text{best}}$, and designates it as the potential aerial target point $\mathbf{p}^a_{\text{target}}$ (Algo. 1, lines 7–8). The proposed deployment mechanism is triggered when the expected exploration gain in 3D significantly exceeds that in 2.5D, as defined below:

$$\mathcal{H}(\phi^g_{L,\text{best}}, \bar{\phi}^a_{L,\text{best}}) = \begin{cases} 1, & \text{if } \phi^g_{L,\text{best}} \leq e^{-\gamma_D}\bar{\phi}^a_{L,\text{best}}, \\ 0, & \text{otherwise,} \end{cases} \tag{5}$$

where a value of 1 indicates that the mechanism is triggered and $\gamma_D > 0$ controls the deployment penalty (Algo. 1, lines 9–10). In essence, the aerial robot is deployed when the potential exploration gain in 3D space exceeds that of the ground robot—even if the ground robot still has viable exploration options. This typically occurs in environments where the aerial robot can more efficiently explore complex 3D structures such as steep slopes, narrow passages, or large vertical spaces beyond the ground robot's effective reach.

Integrated with GBPlanner, the proposed compression method and keyframing strategy enable efficient map-sharing during and after deployment. Each robot $i \in \{g, a\}$ uses a VAE $\{\mathcal{E}_i, \mathcal{D}_i\}$ to encode its range image $\mathbf{x}_i$, derived from point cloud $\mathcal{P}^i_0$ projection, into a latent vector $\mathbf{z}_i$. Combined with the sensor pose ξ^i_0 at the time of capture, this forms a keyframe $k_i = \{\mathbf{z}_i, \xi^i_0\}$. A new keyframe is added to the set $\mathbb{K}_i$ whenever the robot's translation or rotation exceeds predefined thresholds τ_t or τ_r relative to the last keyframe pose ξ^i_k (Algo. 2). The ground robot maintains its keyframe set $\mathbb{K}_g$ (Algo. 1, line 11), and shares a subset to initialize the aerial robot's volumetric map upon deployment.

Deployment Phase: During the aerial robot's deployment, a co-localization technique —triggered only once at the deployment time —enables both robots to operate within a shared inertial frame $\mathcal{I}$. This allows each robot to independently update its map by incorporating both local observations and shared data during post-deployment, while leveraging the collaborative map-sharing solution. Consistent timestamping across platforms—enabled by Chrony—ensures accurate and reliable map fusion. Building on [31], the ground system shares: a) a dense local point cloud map $\mathcal{M}$, and b) its current pose ξ^g_0, which serves as an initial estimate for the aerial robot's localization. This enables co-localization by iteratively aligning the aerial robot's scan to $\mathcal{M}$ through point-to-line and point-to-plane minimization. Once co-localization is achieved, the aerial robot shares back the transform T^g_a between the two LiDAR frames (Algo. 1, line 12).

After co-localization, the ground robot sends the latest N_k keyframes to the aerial robot, which uses the ground robot's decoder $\mathcal{D}_g$ to reconstruct the corresponding point clouds. These, along with their poses, are integrated into the

Algorithm 1 Collaborative Exploration Path Planning

Phase 1: Pre-Deployment

Require: Ground robot pose ξ_0^g and its point cloud $\mathcal{P}_0^g$
1: $\xi_k^g \leftarrow \xi_0^g$
2: `DeploymentTriggered` $\leftarrow$ `false`
3: **while** `not DeploymentTriggered` **do**
4: $\quad \xi_0^g \leftarrow$ **GetCurrentConfiguration**()
5: $\quad \mathbb{G}_L^g \leftarrow$ **BuildLocalGraphGroundRobot**(ξ_0^g)
6: $\quad \sigma_{L,\text{best}}^g,\ \phi_{L,\text{best}}^g \leftarrow$ **GetBestLocalPathAndGain**($\mathbb{G}_L^g$)
7: $\quad \bar{\mathbb{G}}_L^a \leftarrow$ **BuildVirtual3DGraph**(ξ_0^g)
8: $\quad \mathbf{p}_{\text{target}}^a,\ \bar{\phi}_{L,\text{best}}^a \leftarrow$ **GetBestVertexAndGain**($\bar{\mathbb{G}}_L^a$)
9: $\quad$ **if** $\mathcal{H}(\phi_{L,\text{best}}^g, \bar{\phi}_{L,\text{best}}^a)$ **then**
10: $\quad\quad$ `DeploymentTriggered` $\leftarrow$ `true`
11: $\quad \mathbb{K}_g \leftarrow$ **Keyframing**(g, ξ_0^g, ξ_k^g) ▷ Algorithm 2

Phase 2: Deployment

12: **procedure** CO-LOCALIZATION(ξ_0^g, $\mathcal{M}$)
13: **procedure** MAP-SHARING($\mathbb{K}_g$, N_k)
14: **procedure** TARGET-SHARING($\mathbf{p}_{\text{target}}^a$)

Phase 3: Post-Deployment

Require: Set of robots $\mathcal{R} = \{g, a\}$
15: **while** `remainingTime` $> t_b$ **do**
16: $\quad$ **for all** i in $\mathcal{R}$ **do**
17: $\quad\quad \xi_0^i \leftarrow$ **GetCurrentConfiguration**()
18: $\quad\quad \mathbb{G}_L^i \leftarrow$ **BuildLocalGraph**(ξ_0^i)
19: $\quad\quad \sigma_{L,\text{best}}^i \leftarrow$ **GetBestLocalPath**($\mathbb{G}_L^i$)
20: $\quad\quad \mathbb{K}_i \leftarrow$ **Keyframing**(i, ξ_0^i, ξ_k^i) ▷ Algorithm 2
21: $\quad\quad$ **if** $\exists$ neighbor robot **then**
22: $\quad\quad\quad$ **SendUnsharedKeyframes**($\mathbb{K}_i$)
23: $\quad\quad\quad$ **if** $i = g$ **then** ▷ Ground Robot
24: $\quad\quad\quad\quad \mathbb{K}_a \leftarrow$**ReceiveKeyframes**()
25: $\quad\quad\quad\quad \mathbb{T}_g \leftarrow$ **ComputeTimesToKeyframes**($\mathbb{K}_g$)
26: $\quad\quad\quad\quad$ **SendTimesToKeyframes**($\mathbb{T}_g$)
27: $\quad\quad\quad\quad$ **ReceiveRegroupingPoint**()
28: $\quad\quad\quad$ **else if** $i = a$ **then** ▷ Aerial Robot
29: $\quad\quad\quad\quad \mathbb{K}_g \leftarrow$**ReceiveKeyframes**()
30: $\quad\quad\quad\quad \mathbb{T}_g \leftarrow$ **ReceiveTimesToKeyframes**()
31: $\quad\quad\quad\quad \mathbb{T}_a \leftarrow$ **ComputeTimesToKeyframes**($\mathbb{K}_g$)
32: $\quad\quad\quad\quad \mathbb{K}_{g,\text{best}} \leftarrow$ **GetRegroupingPoint**($\mathbb{T}_g$,$\mathbb{T}_a$)
33: $\quad\quad\quad\quad$ **SendRegroupingPoint**($\mathbb{K}_{g,\text{best}}$)
34: **ReturnToRegroupingPoint**($\mathbb{G}_G^i$), $i \in \{g, a\}$

Algorithm 2 Keyframing

1: **function** KEYFRAMING(i, ξ_0^i, ξ_k^i)
2: $\quad \Delta t,\ \Delta r \leftarrow$ **ComputePoseDifference**(ξ_0^i, ξ_k^i)
3: $\quad$ **if** $\Delta t > \tau_t$ **or** $\Delta r > \tau_r$ **then**
4: $\quad\quad \mathcal{P}_0^i \leftarrow$ **GetCurrentPointCloud**()
5: $\quad\quad \mathbf{x}_i \leftarrow$ **PointCloudProjection**($\mathcal{P}_0^i$)
6: $\quad\quad \mathbf{z}_i \leftarrow \mathcal{E}_i(\mathbf{x}_i)$
7: $\quad\quad \mathbb{K}_i \leftarrow \mathbb{K}_i \cup \{\mathbf{z}_i, \xi_0^i\}$
8: $\quad\quad \xi_k^i \leftarrow \xi_0^i$
9: $\quad$ **return** $\mathbb{K}_i$

aerial robot's volumetric map to enable collaborative planning (Algo. 1, line 13). The ground robot also shares the target point $\mathbf{p}_{\text{target}}^a$, guiding the aerial platform to its initial position. Using its local graph $\mathbb{G}_L^a$, the aerial robot then computes a path to the target and initiates exploration (Algo. 1, line 14). To prevent mapping artifacts when both robots operate in overlapping areas, Voxblox's TSDF integration filters out transient objects such as the other robot by repeatedly updating free space, ensuring only persistent structures remain in the map.

Post-Deployment Phase: After the ground robot shares the necessary data and the aerial robot reaches its target, both begin independent exploration. During this phase, they exchange keyframes when within communication range r_c and store them for later transfer when out of range. To address endurance constraints, an energy-aware regrouping strategy ensures both robots return to a common point before battery depletion. A tunable time budget t_b, set below the battery life of the robot with the least remaining capacity, guarantees a timely return. Initially, the regrouping point is set to the deployment location. When in communication range, the ground robot estimates travel times $\mathbb{T}_g$ to its keyframes using shortest paths from its global graph $\mathbb{G}_G^g$, assuming constant velocity, and shares them with the aerial robot (Algo. 1, lines 25–26). The aerial robot then computes its own times $\mathbb{T}_a$ to reachable keyframes—those with collision-free paths in $\mathbb{G}_G^a$—and selects a new regrouping point $\mathbb{K}_{g,\text{best}}$ according to (Algo. 1, lines 30-32):

$$\mathbb{K}_{g,\text{best}} = \mathbb{K}_{g,\kappa} \text{ where } \kappa = \underset{q \in \{1,\ldots,|\mathbb{K}_g|\}}{\arg\min} \left(\max\{\mathbb{T}_{g,q}, \mathbb{T}_{a,q}\} \right) \tag{6}$$

where $|\mathbb{K}_g|$ is the number of ground robot keyframes. The aerial robot shares the selected regrouping point with the ground robot (Algo. 1, lines 27, 33). At each planning step, both robots estimate the time needed to complete their next path and return. If this total exceeds the time budget, regrouping is triggered. The mission concludes once both robots return to the regrouping point and exchange all remaining data (Algo. 1, lines 34).

IV. EVALUATION STUDIES

A marsupial ground-aerial robot team was employed to evaluate the proposed approach in both simulation and real-world experiments. The team consists of a legged ground robot, ANYmal-D, and a aerial robot, RMF-Owl [32], operating in a marsupial configuration (Fig. 1). ANYmal-D measures 0.93 m × 0.53 m × 0.80 m (L × W × H), is equipped with a Velodyne VLP-16 LiDAR (FoV: [360°, 30°], range: 100 m), and serves as a carrier platform. It runs on 2× 8th Gen Intel Core™ i7 CPUs. RMF-Owl features a collision-tolerant frame (0.38 m × 0.38 m × 0.24 m) and an Ouster OS0-64 LiDAR (FoV: [360°, 90°], range: 50 m), powered by a Khadas VIM4 with 4× 2.2GHz Cortex-A73 and 4× 2.0GHz Cortex-A53 cores. It is mounted on the ground robot via a dedicated marsupial mechanism. Each robot is pre-equipped with its own trained VAE encoder for data compression and uses the other robot's decoder for decompression. Communication is handled

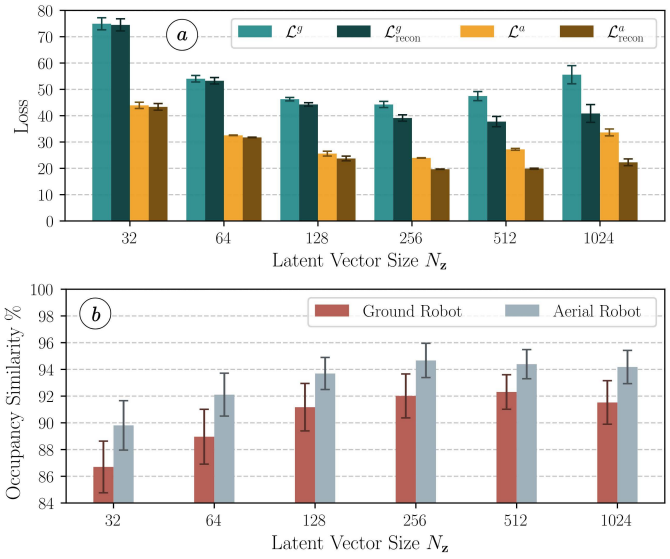

Fig. 4. Comparison of (a) overall and reconstruction losses ($\mathcal{L}^g$, $\mathcal{L}^g_{\text{recon}}$ for ground; $\mathcal{L}^a$, $\mathcal{L}^a_{\text{recon}}$ for aerial), and (b) occupancy similarity for the ground $\{\mathcal{E}_g, \mathcal{D}_g\}$ and aerial $\{\mathcal{E}_a, \mathcal{D}_a\}$ VAE models. Metrics are computed on the test set across latent sizes ($N_{\mathbf{z}}$) and voxel sizes (s_{vxl}). Each bar shows the average over three voxel sizes.

over WiFi using the NimbRo framework. All processes—co-localization, map sharing, target sharing, and collaborative exploration—run fully onboard and in real time.

A. Training Methodology

To account for differences in LiDAR characteristics, separate VAE models were trained on platform-specific datasets composed of simulated and real-world range images from diverse environments, including caves, confined spaces, and complex buildings. The aerial dataset included $\sim$ 36,000 images ($\sim$ 26,000 simulated), while the ground dataset had $\sim$ 25,000 ($\sim$ 21,000 real). Each model was trained independently for 20 epochs using the Adam optimizer (learning rate 10^{-4}, batch size 16), with a $90\% - 10\%$ train-test split.

B. Ablation Study

An ablation study was conducted to evaluate the effects of latent dimensionality and voxel resolution on volumetric map reconstruction. VAE models were trained with latent sizes $N_{\mathbf{z}} \in \{32, 64, 128, 256, 512, 1024\}$ and voxel sizes $s_{\text{vxl}} \in \{0.2, 0.3, 0.4\}$m for both robots. As shown in Figure 4a, the overall loss decreased with larger latent sizes, reaching a minimum at 256, then increased due to rapidly growing KL divergence, which led to over-regularization and degraded reconstruction quality [28].

To assess how well compressed data retains task-relevant spatial information, we introduce an occupancy similarity metric, which compares voxel-wise occupancy between maps generated from original and reconstructed range images using a k-nearest voxel approach. This metric reflects planning utility more directly than image-space fidelity. As shown in Figure 4b, occupancy similarity trends align with reconstruction loss, peaking at latent sizes of 256 (aerial) and 512 (ground), indicating that 256 is the smallest latent size offering a strong trade-off between spatial consistency and bandwidth. This supports our choice to remap data into a voxel-aware format and evaluate models using both image and task-specific metrics. The approach achieves high compression rates—337 : 1 (ground) and 384 : 1 (aerial)—enabling low-bandwidth communication (e.g., LoRa) for real-time multi-robot coordination in challenging environments.

TABLE I
PARAMETERS FOR BOTH EXPERIMENTS

Parameter	Ground Robot	Aerial Robot
Size of range image $\mathbf{x}$ $(H \times W)$	16×1800	64×512
Voxel size s_{vxl}	0.2 cm	0.2 cm
Maximum range of image $r_{\text{img}}^{\max}$	20 m	20 m
Latent space size $N_{\mathbf{z}}$	256	256
Deployment sensitivity γ_D	3.5 / 4.5	-
Translation threshold τ_t	2.0 m	3.0 m
Rotation threshold τ_r	0.785 rad	0.785 rad
Number of keyframes $N_k \subseteq \mathbb{K}_g$	300 / 10	-
Communication range r_c	50 m / 10 m	50 m / 10 m
Time budget t_b	2000 s / 300 s	2000 s / 300 s
Nominal speed	0.7 m/s	1 m/s

* Parameters of simulation and real-world experiments, represented as $\mathfrak{a}$ / $\mathfrak{b}$ for distinct values, or as $\mathfrak{c}$ for common values.

C. Simulation Studies

We validated our approach using a large-scale Gazebo simulation in a multi-level building with interconnected rooms, hallways, stairs, and ramps. The marsupial ground–aerial team coordinated successfully, meeting at an intermediate location to reduce return times and improve energy efficiency via the proposed opportunistic regrouping strategy. Robot behavior and sensors were accurately modeled using the ANYmal and RotorS simulators (see Sec. IV). Figure 5 shows the collaboratively explored volumetric maps, including key elements such as local and virtual graphs, deployment/regrouping points, shared maps, keyframes, and onboard representations. Co-localization was unnecessary as both robots operated in a shared reference frame provided by the simulation environment. The effectiveness of data sharing is highlighted by pivot points where the aerial robot redirected to unexplored areas after detecting overlap with the ground robot's map. The visible differences between the original and reconstructed range images (Fig. 5, 5a–5b) stem from the task-driven VAE's high compression ratio, which prioritizes occupancy-relevant features over raw geometric detail. Table I lists the mission parameters used during the 42-minute operation, which included 12 minutes in marsupial mode and 30 minutes of independent exploration per robot. Table II presents how compression and keyframing techniques contributed to bandwidth reduction. Latent encoding significantly lowers data rates, and keyframing ensures only essential updates are shared, minimizing communication overhead for bandwidth-constrained scenarios.

D. Experimental Results

To demonstrate the applicability of the proposed method, we conducted real-world experiments in a multi-storey university building using a marsupial ground–aerial robot team.

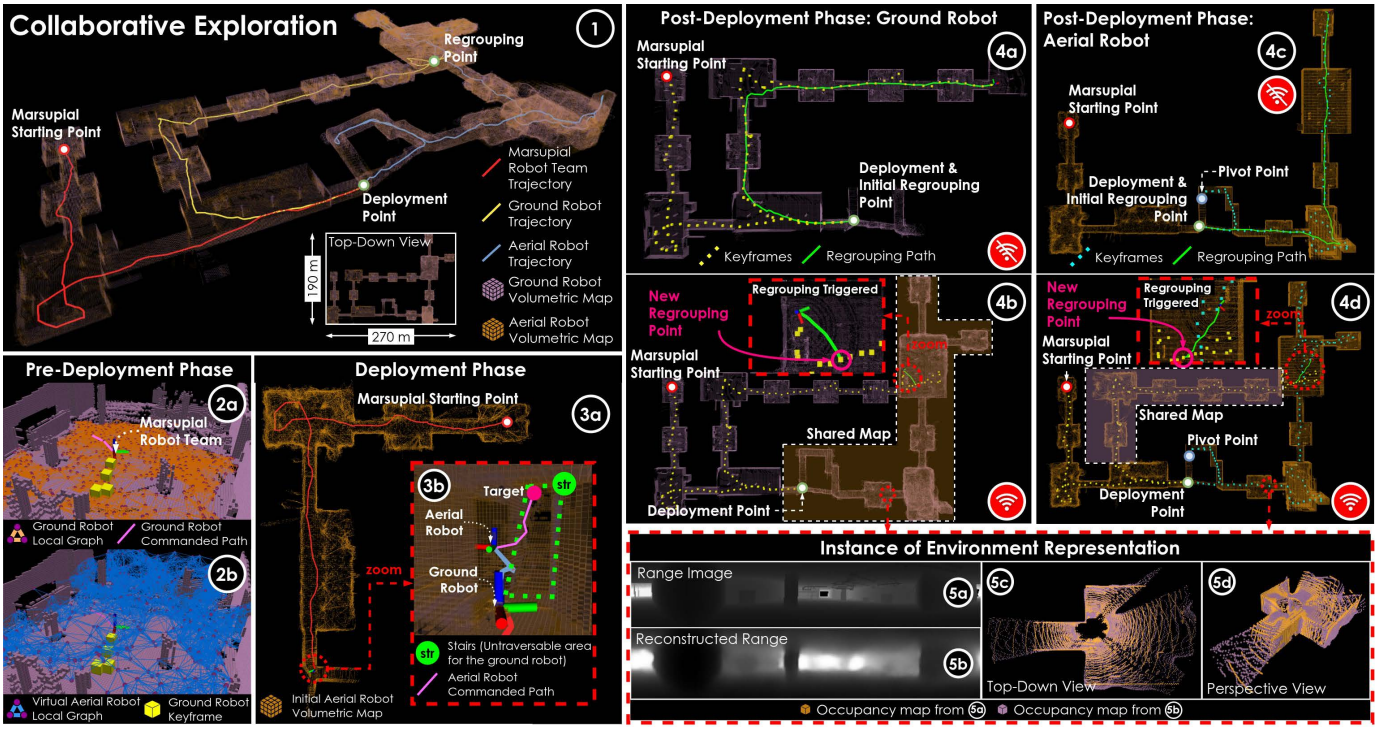

Fig. 5. Simulation results showcasing collaborative exploration by a marsupial ground–aerial robot team in a multi-level unknown environment (1). The process includes: (a) Pre-Deployment—the ground robot explores while carrying the aerial robot, builds dense graphs, evaluates potential aerial deployment locations, and stores keyframes (2a-2b); (b) Deployment—the ground robot shares the map and assigns a target to the aerial robot (3a-3b); and (c) Post-Deployment—both robots explore independently, share keyframes, and select a new regrouping point when within communication range (4a-4d). Finally, an instance of the aerial robot's range image (5a), its reconstruction by the ground robot (5b), and the corresponding occupancy maps (5c–5d) are shown.

The ground robot was equipped with a custom 3D-printed deployment mechanism (Fig. 1), featuring a flat platform, motorized brackets powered by ANYmal, and a secure mounting system for the aerial robot. The mission began at a designated start point, with the team exploring until the ground robot encountered an untraversable area due to a physical obstacle (marked "obs" in Fig. 6), triggering aerial deployment. During deployment, co-localization, map sharing, and target sharing were performed. The aerial robot processed each received keyframe in 100 ms. After deployment, both robots explored independently, exchanging data when within network range. Upon regrouping, they returned to the deployment point as they were out of range and completed the mission by merging their explored maps. The differences between original and reconstructed range images (Fig. 6, 4c–4d) reflect the aggressive task-driven compression, which retains mapping-relevant structure over visual detail. Portions of the drone cage visible in the aerial images were masked and inpainted prior to encoding to avoid corrupting the latent space. The mission lasted 10 minutes, during which the ground and aerial robots explored approximately $\sim 6,600$ m^3 and $\sim 7,500$ m^3, respectively. Table I summarizes mission settings. For each keyframe, generation took 10 ms on the ground robot and 400 ms on the aerial robot, with data rates of 0.187 kB/s and 0.276 kB/s, respectively. Figure 6 illustrates key stages of the collaborative exploration process.

Simulation and real-world results show that the proposed framework scales well in distance and application scope. The energy-aware regrouping enables flexible coordination without returning to the deployment point if it is possible, supporting longer missions. Its modular design and bandwidth-efficient map sharing make it suitable for scenarios like subterranean exploration, industrial inspection, and disaster response in GPS- or communication-limited environments.

TABLE II
QUANTITATIVE COMPARISON OF DATA TRANSMISSION RATES

Transmission Mode	Data Rates (kB/s)
Raw point cloud transmission (10 Hz)	3375
Keyframed raw point cloud transmission	58.387
Latent vector transmission (10 Hz)	10
Keyframed latent space transmission	**0.173**

V. CONCLUSION

In this paper, we presented a collaborative exploration approach for a marsupial ground-aerial robot team. Through a bandwidth-efficient, task-driven map-sharing solution, both robots can plan based on not only their local observations but also on shared information, enabling more efficient exploration. An ablation study further highlights the trade-off between the size of shared data and the quality of the reconstructed environment. Both simulation and real-world experiments were conducted to validate the proposed approach.

REFERENCES

[1] J. Delmerico *et al.*, "The current state and future outlook of rescue robotics," *Journal of Field Robotics*, vol. 36, pp. 1171–1191, 2019.
[2] B. Grocholsky *et al.*, "Cooperative air and ground surveillance," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 16–25, 2006.

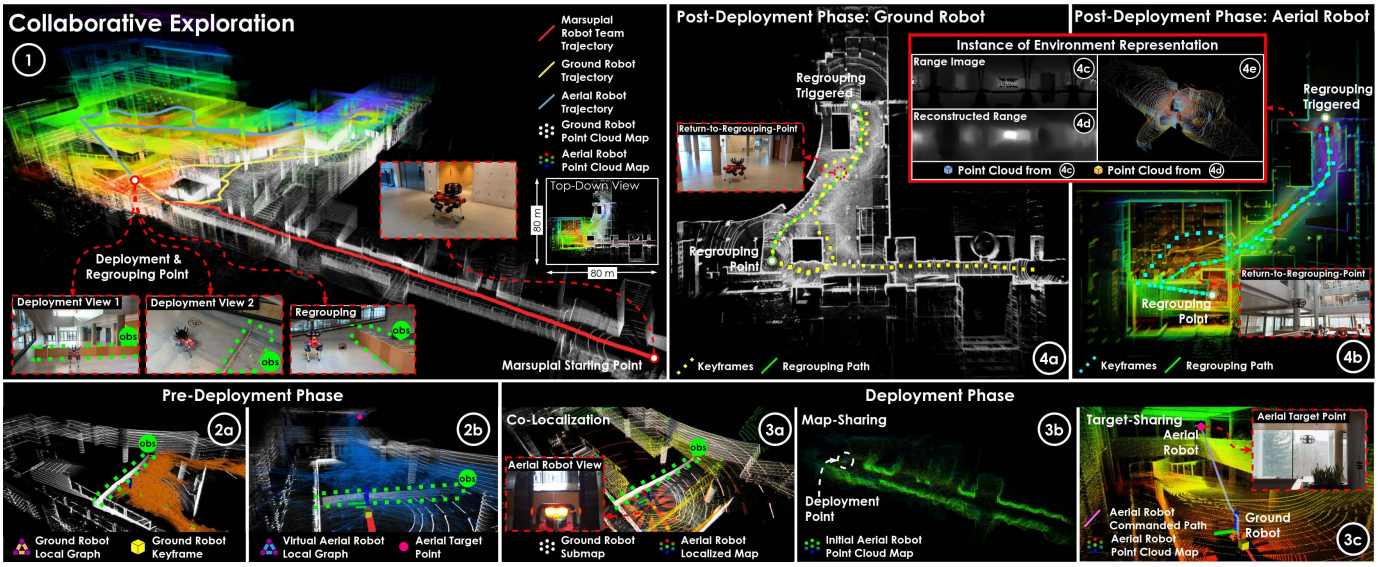

Fig. 6. Experimental results from the collaborative exploration of the marsupial ground-aerial robot team in a multi-storey university building. The combined map, explored by both robots, is shown in (1) and highlights the start, deployment, and regrouping points. The ground and virtual aerial planning graphs, generated during the pre-deployment phase and used to trigger deployment, are depicted in (2a)–(2b). The processes of co-localization during the aerial robot deployment (3a), map-sharing (3b), and target-sharing (3c) are also presented. The individual exploration maps along with regrouping path are illustrated in (4a)–(4b). An instance of the aerial robot's range image (4c), the associated reconstructed image on the ground robot (4d), and the corresponding point clouds (4e) are provided.

[3] M. Dharmadhikari *et al.*, "Semantics-aware exploration and inspection path planning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3360–3367.

[4] M. Kulkarni *et al.*, "Autonomous teamed exploration of subterranean environments using legged and aerial robots," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 3306–3313.

[5] T. H. Chung *et al.*, "Into the robotic depths: analysis and insights from the darpa subterranean challenge," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 6, no. 1, pp. 477–502, 2023.

[6] N. Kottege *et al.*, "Heterogeneous robot teams with unified perception and autonomy: How team csiro data61 tied for the top score at the darpa subterranean challenge," *Field Robotics*, pp. 313–359, 2024.

[7] C. Cao *et al.*, "Exploring the most sectors at the darpa subterranean challenge finals," *Field Robotics*, 2023.

[8] P. De Petris *et al.*, "Marsupial walking-and-flying robotic deployment for collaborative exploration of unknown environments," in *2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2022, pp. 188–194.

[9] B. Moore *et al.*, "Combined docking-and-recharging for a flexible aerial/legged marsupial autonomous system," in *2023 IEEE Aerospace Conference*. IEEE, 2023, pp. 1–9.

[10] J. Balaram *et al.*, "The ingenuity helicopter on the perseverance rover," *Space Science Reviews*, vol. 217, no. 4, p. 56, 2021.

[11] S. Martinez-Rozas *et al.*, "Path and trajectory planning of a tethered uav-ugv marsupial robotic system," *IEEE Robotics and Automation Letters*, 2023.

[12] J. Capitán *et al.*, "An efficient strategy for path planning with a tethered marsupial robotics system," *arXiv preprint arXiv:2408.02141*, 2024.

[13] C. Y. H. Lee *et al.*, "Stochastic assignment for deploying multiple marsupial robots," in *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2021, pp. 75–82.

[14] M. S. Couceiro *et al.*, "Marsupial teams of robots: deployment of miniature robots for swarm exploration under communication constraints," *Robotica*, vol. 32, no. 7, pp. 1017–1038, 2014.

[15] G. Best *et al.*, "Multi-robot, multi-sensor exploration of multifarious environments with full mission aerial autonomy," *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 485–512, 2024.

[16] X. Zhou *et al.*, "Riddle: Lidar data compression with range image deep delta encoding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 212–17 221.

[17] C. Tu *et al.*, "Point cloud compression for 3d lidar sensor using recurrent neural network with residual blocks," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 3274–3280.

[18] L. Wiesmann *et al.*, "Deep compression for dense point cloud maps," *IEEE Robotics and Automation Letters*, vol. 6, pp. 2060–2067, 2021.

[19] M. Quach *et al.*, "Survey on deep learning-based point cloud compression," *Frontiers in Signal Processing*, vol. 2, p. 846972, 2022.

[20] X. Sun *et al.*, "A novel point cloud compression algorithm based on clustering," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2132–2139, 2019.

[21] Y. Feng *et al.*, "Real-time spatio-temporal lidar point cloud compression," in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 10 766–10 773.

[22] M. T. Lazaro *et al.*, "Multi-robot slam using condensed measurements," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1069–1076.

[23] N. Stathoulopoulos *et al.*, "Recnet: An invertible point cloud encoding through range image embeddings for multi-robot map sharing and reconstruction," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 4883–4889.

[24] L. Zheng *et al.*, "Real-time efficient environment compression and sharing for multi-robot cooperative systems," *IEEE Transactions on Intelligent Vehicles*, 2024.

[25] Y. Cao *et al.*, "Real-time lidar point cloud compression and transmission for resource-constrained robots," *arXiv preprint arXiv:2502.06123*, 2025.

[26] M. Kulkarni *et al.*, "Task-driven compression for collision encoding based on depth images," in *Advances in Visual Computing*. Cham: Springer Nature Switzerland, 2023, pp. 259–273.

[27] D. P. Kingma *et al.*, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2022.

[28] I. Higgins *et al.*, "beta-VAE: Learning basic visual concepts with a constrained variational framework," in *International Conference on Learning Representations*, 2017.

[29] M. Kulkarni *et al.*, "Semantically-enhanced deep collision prediction for autonomous navigation using aerial robots," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 3056–3063.

[30] H. Oleynikova *et al.*, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1366–1373.

[31] S. Khattak *et al.*, "Complementary multi–modal sensor fusion for resilient robot pose estimation in subterranean environments," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2020, pp. 1024–1029.

[32] P. D. Petris *et al.*, "Rmf-owl: A collision-tolerant flying robot for autonomous subterranean exploration," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022, pp. 536–543.