Streamlining Data Transfer in Collaborative SLAM through Bandwidth-aware Map Distillation

Rui Ge, Huanghuang Liang, Zheng Gong, Chuang Hu, *Member, IEEE,*
Xiaobo Zhou, *Senior Member, IEEE,* and Dazhao Cheng, *Senior Member, IEEE*

***Abstract*—Edge intelligence offers a promising solution for Simultaneous Localization and Mapping (SLAM) in large-scale scenarios, where multiple robots collaboratively perceive the environment and upload their local maps to an edge server. However, maintaining mapping accuracy under constrained and dynamic communication resources remains a significant challenge for the practical deployment of robot swarms. Concurrent data uploads from multiple agents can exacerbate network congestion, leading to the loss of critical information, delayed updates, and, ultimately, the inconsistency of the generated maps.**

This paper presents Hermes, an edge-assisted collaborative mapping system designed for communication-constrained environments. Hermes streamlines data transfer through bandwidth-aware map distillation, ensuring only the most crucial messages are transmitted to the edge server. We quantify the importance of keyframes and landmarks based on their information entropy gain in pose estimation. By selectively sharing essential submaps, Hermes adaptively balances communication bandwidth and information richness during the mapping process. We implemented Hermes on heterogeneous platforms and conducted experiments using public datasets and self-collected campus data. Hermes exceeds SwarmMap by 50% in bandwidth utilization with similar accuracy and surpasses COVINS-G by 65% in trajectory error under highly constrained network resources.

***Index Terms*—Map Compression, Low Bandwidth, Visual SLAM, Robot Swarm.**

I. Introduction

THE advent of autonomous systems has significantly fueled interest in Simultaneous Localization and Mapping (SLAM), a critical robotic technique for constructing or updating a map of an unknown environment while tracking the agent's location within it. While traditional visual SLAM approaches largely focus on single-agent scenarios [1] [2], the expanding use of multi-robot systems in diverse fields, like search and rescue in disaster areas, has underscored the need for *Collaborative SLAM* (C-SLAM). C-SLAM involves multiple robots synchronizing their observations on a central server, enhances mapping efficiency, and reduces the uncertainties caused by individual sensor limitations in extensive scenarios, such as subterranean search&rescue [3], indoor exploration [4] [5], and city-scale patrol [6] [7].

Rui Ge, Huanghuang Liang and Dazhao Cheng are with the School of Computer Science, Wuhan University. Email: {whugerui, hhliang, dcheng}@whu.edu.cn.

Zheng Gong is with the School of Cyber Security, Tianjin University. Email: marcogong22@gmail.com.

Chuang Hu and Xiaobo Zhou are with the State Key Laboratory of Internet of Things for Smart City of the University of Macau. Email: {chuanghu, waynexzhou}@um.edu.mo.

The core functionality of C-SLAM relies on the seamless coordination of multiple agents, synchronized to achieve real-time operational precision [8] [9]. Given the substantial computational and synchronization demands on mobile platforms, offloading tasks to edge servers has become a viable solution to enhance performance by reducing the computational load on mobile devices. However, this transition to edge computing introduces significant challenges, particularly bandwidth limitations and communication instabilities that worsen as the number of robotic agents grows [10], [11]. In large-scale C-SLAM deployments, the bandwidth requirement typically increases linearly with the number of agents [12]. Additionally, a higher number of agents increases the likelihood of network collisions, leading to bandwidth spikes from concurrent data transfer and increased demands for packet retransmission.

Fig. 1 demonstrates a real-world test that highlights the linear increase in bandwidth usage and the sudden spikes that cause instant congestion as the number of agents increases. Such congestion significantly undermines the accuracy and reliability of localization efforts, consequently affecting the completeness of the global map, as illustrated in Fig. 2. Furthermore, environmental dynamics including radio frequency interference, physical barriers, and agent mobility induce fluctuations in communication channels and bandwidth [13], exacerbating these challenges, as shown in Fig. 1. These fluctuations not only reduce communication bandwidth but also impair system responsiveness and increase inaccuracies in mapping and localization, ultimately compromising the operational integrity of the system. While recent innovations in C-SLAM, such as SwarmMap [14], EdgeSLAM2 [15] and CMD-SLAM [16], attempt to address these communication challenges, they often fail to deliver consistent performance in large, dynamic environments or require additional hardware, thereby limiting their practical applicability and effectiveness.

Despite its promising potential, edge-assisted C-SLAM remains immature, driving researchers' enthusiasm for continuous improvement. Current practical implementations frequently encounter communication fluctuations that impede information transfer, leading to delays and intermittent data synchronization between agents and the edge. These issues increase mapping errors and compromise system reliability.

In this paper, we propose Hermes, an edge-assisted C-SLAM framework designed to enhance data handling in environments with variable network conditions. Hermes employs a sophisticated mechanism that not only efficiently reduces and sparsifies SLAM data but also dynamically adjusts this sparsification based on real-time bandwidth assessments. This ap-

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3549367

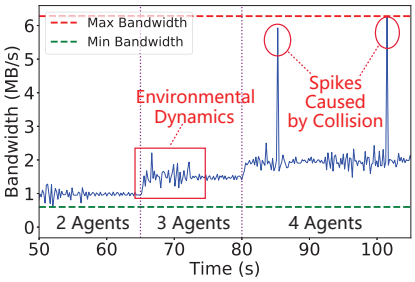

Fig. 1: Dynamic Bandwidth Demand.

Fig. 2: Map Distortion and Incompleteness.

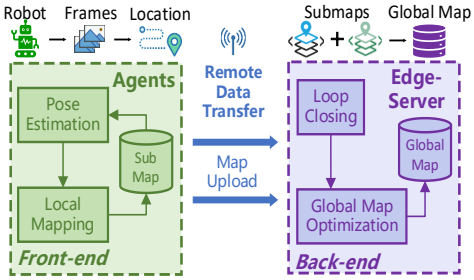

Fig. 3: Edge-assisted C-SLAM architecture.

proach ensures efficient utilization of available communication resources, thereby maximizing real-time system performance. Specifically, Hermes integrates three key design elements.

First, we enhance visual SLAM with an entropy-based keyframe (KF) designation method that focuses on scene complexity, rather than merely on landmark (LM) associations. By assessing the information entropy of scenes, this approach more accurately captures environmental changes, enabling precise and comprehensive mapping in environments with varying textures (see §IV-A).

Second, we propose a bandwidth-aware map distillation method that further refines SLAM data processing. This method uses a theoretically grounded strategy to quantify LMs based on information gain, retaining sparse yet informative LMs. It also reduces data volume by sparsifying LMs and trimming KFs, leveraging a dual-stage mixed integer linear program (MILP) to effectively address this NP-hard problem. Moreover, a bandwidth coordinator dynamically adjusts the sparsification ratio in response to current network conditions, ensuring optimal data transmission and resources utilization efficiency (see §IV-B).

Third, we present a credibility-based submap assessor designed to alleviate communication congestion in C-SLAM. This tool improves global map accuracy by prioritizing the transmission of submaps from various agents, based on criteria such as transmission stability and map connectivity. The server processes these submaps through a priority queue, ensuring that the most reliable submaps are integrated first. This method helps maintain high precision in global mapping throughout the project's lifecycle (see §IV-C).

We implemented the Hermes agent-server architecture across a diverse array of robots and personal computers in real-world environments (see §V). Our evaluation leveraged both widely recognized public datasets and proprietary data gathered from our campus to assess system efficacy. Notably, Hermes achieves a 65% reduction in trajectory error compared to the COVINS-G system [17] under similar network constraints and improves bandwidth utilization by 50% relative to SwarmMap [14], while maintaining comparable accuracy levels. These advancements mark significant progress in the efficiency and reliability of mobile computing applications in constrained network conditions (see §VI).

To further demonstrate the effectiveness of Hermes in practical scenarios, we conducted two case studies on our campus: an indoor experiment utilizing WiFi Direct and an outdoor experiment with remote connections.

II. Background and Related Work

A. Edge-assisted C-SLAM system

An edge-assisted C-SLAM system can be decoupled into front-end and back-end, as illustrated in Fig. 7. The front-end (i.e., agents) are responsible for profiling their surroundings and transmitting essential information to the edge-server. The back-end (i.e., the edge-server), on the other hand, forms a globally consistent map by stitching submaps.

Front-end. By tracking salient feature points across consecutive frames, an agent can estimate its position $\mathbf{t}$ and orientation $\mathbf{R}$ changes (6 Degrees of Freedom) by Epipolar Geometry [18] or Perspective-n-Point [19]. This procedure is defined as Visual Odometry (VO), where each agent runs tracking and local mapping threads individually to form a submap in its coordinate. A submap consists of multiple KF, each containing several LMs. KFs are crucial snapshots within the map, capturing essential environmental states, while LMs serve as recognizable reference points within those frames (see Fig. 6). KFs and LMs in the submap are uploaded to the edge-server for fine-grained optimization.

Back-end. In edge-assisted C-SLAM [20] [21], the computationally intensive and collaborate-requiring tasks, such as global optimization and loop closing, will be offloaded to an edge-server. Loop closing identifies previously visited locations, introducing extra constraints for Pose Graph Optimization (PGO). This process involves merging all submaps using the estimated transformations from loop closures.

B. Related Works

Collaborative Visual SLAM. As robots are increasingly tasked with complex functions, research institutions have shifted their focus to Multi-Agent Systems (MAS) to achieve efficient resources sharing and optimization through collaboration among multiple robots. Given their affordability and low energy consumption, cameras have become the predominant sensors in MAS, facilitating large-scale deployment and spurring advancements in various Visual-SLAM approaches as evidenced by studies such as Chang et al. [3], Xu et al. [22], and Schmuck et al. [23]. Recent studies [24] [25] integrate multi-robot systems with emerging 3D reconstruction methods, such as 3D-Gaussian-based scene representation [26], to deliver higher-quality scene models. However, these approaches incur significant bandwidth consumption when transmitting environmental details.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3549367

To address the fundamental tension between environmental representation completeness and bandwidth-efficient communication in multi-robot SLAM systems, several approaches have been proposed. COVINS-G [17] employs a generic front-end wrapper that effectively reduces data transmission by decoupling the front-end from the back-end and re-extracting landmarks (LMs) in keyframes (KFs). However, this method may compromise environmental representation because LMs are detached from the original sensor data. As a result, it becomes challenging for other robots to establish loop closure constraints effectively in sparse environmental descriptions when revisiting the area. SwarmMap [14] enhances the scalability of MAS through data and task offloading and map profiling methods. By transmitting only map modification operations rather than the entire map data, SwarmMap reduces the data transfer volume when deployed in fixed regions. However, when robots are required to explore uncharted areas, the system experiences frequent "cold starts", leading to high bandwidth consumption. Conversely, the recent Swarm-SLAM framework [4] seeks to reduce the probability of communication bottlenecks in a single node by employing a distributed system, while this reduces central reliance, the absence of a global coordinator in decentralized systems results in lower mapping accuracy. Despite these improvements, existing approaches are not tailored for field applications where bandwidth and connection stability are limited.

Addressing this crucial gap, Hermes sets itself apart by streamlining data transfers based on current network conditions, thus enabling real-time, bandwidth-aware C-SLAM in communication-constrained environments.

SLAM Map Compression. Map compression in SLAM involves two key steps: quantifying the significance of LMs and KFs, and selecting the most essential submaps. A commonly used criterion to determine the importance of an LM is the frequency of its observations; LMs observed more frequently are generally more stable across frames, as discussed by Park et al. [27] and Zhang et al. [28]. However, this rigid approach can sometimes lead to the exclusion of critical LMs. To address this issue, some researchers, such as those behind RapNet [29], have employed deep learning techniques to predict regional invariance and LM reliability, though this method incurs significant computational overhead, making it suitable mainly for long-duration LM evaluations.

Regarding the selection of map components, Carlone et al. [30] employ a greedy method to identify key features, demonstrating the submodularity of the LM selection problem. Building on this concept, AdaptSLAM [31] introduces a map uncertainty quantification method that prunes redundant KFs and supports an edge-assisted SLAM system, even under stringent communication and computation constraints. Similarly, EdgeSLAM2 [15] optimizes map compression by maintaining observation consistency, which aids in selectively retaining informative map points and thus balancing map quality with compression efficiency. Yu et al. [32] proposed an effective multi-agent communication scheme utilizing surrogate entropy minimization and the soft barrier method to reduce message entropy and enhance cooperation under limited bandwidth. However, their focus was on specific tasks such as prey or treasure-hunting, not on map compression within a mapping system. FAUMap [33] and Map++ [10] analyze bandwidth compression methods for crowdsourced map updates from the perspective of map element selection. However, they are not suitable for real-time exploratory mapping tasks.

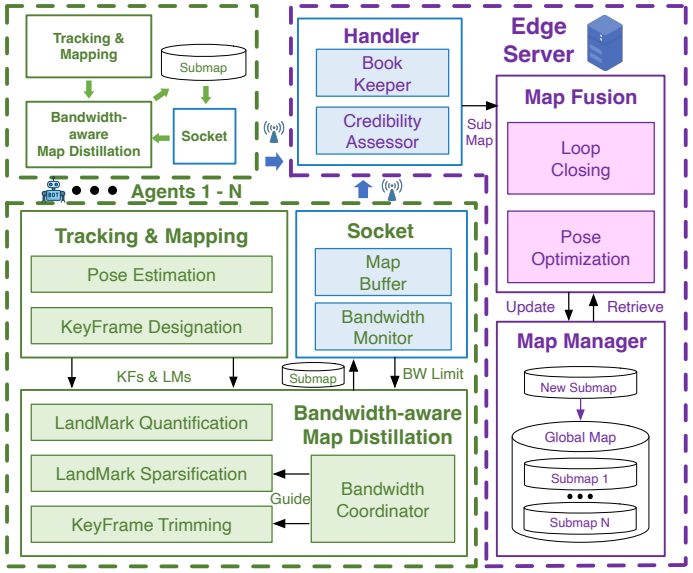

Fig. 4: The Hermes Architecture.

In contrast, Hermes deviates from these existing methodologies by introducing a theoretically grounded landmark sparsification approach for real-time map distillation that operates independently of auxiliary data like IMU or wheel speed measurements. This method is fully compatible with edge-assisted paradigms, facilitating efficient information transfer even in scenarios with unstable communications.

III. Hermes System Modeling

Hermes is composed of N individual Agents and an Edge-Server, as illustrated in Fig. 4. These components are interconnected through a Communication Pipeline, which comprises N communication sockets—each maintained by an agent—and a corresponding handler on the Edge-Server side.

Agents. To balance communication bandwidth and map quality, Hermes captures as much environmental information as possible at the front end and compresses this data before transmission. Specifically, each agent runs an independent VO front-end for the Pose Estimation module and visual SLAM program to generate KFs. Then, the Keyframe Designation module (§IV-A) selects representative KFs based on scene completeness constraints for a coarse compression. Until now, Hermes still maintains a relatively dense submap and KF to ensure the completeness of the environment profile. The filtered KFs and LMs contained within them are subsequently input to the Map Distillation component for fine sparsification, which contains attentional Landmark Sparsification and KeyFrame Trimming (§IV-B) modules. Here, Hermes can adaptively adjust the sparsification and trimming ratio of LMs and KFs based on available bandwidth following the guide of the Bandwidth Coordinator module.

Communication Pipeline. To address the dynamic communication channel conditions, Hermes optimizes the communi-

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3549367

(a) Previous KF **(b) Quantified by Stability** **(c) Quantified by Attention**

Fig. 5: Performance of different LM quantification methods when camera moving from right to left. The LMs on the chair are newly joined whereas the LMs on the TV are long-existed in the the agent's view.

cation pipeline on both the agent and server sides. On the agent side, Hermes maintains a Socket component that contains a Map Buffer, which mitigates map incompleteness resulting from communication interruptions, and a Bandwidth Monitor, which monitors the available bandwidth regularly to set a bandwidth limit for LM and KF filtering. On the server side, a Handler (§IV-C) component manages all agents' submaps, performing bookkeeping for all agents. On the Credibility Assessor, Hermes quantifies the credibility of each submap based on the transmission stability of its corresponding channel.

Edge Server. The edge server merges all submaps into a global map. For each received KF, a lightweight scene descriptor is extracted. The Map Fusion component carries out place recognition based on descriptor similarity and computes the transformation between newly received KF and the candidate KFs. Then, on the Map Manager component, each agent's submap is merged into the global map by global pose graph optimization (PGO), where map merging is performed sequentially in descending order of credibility (§ IV-D).

IV. Hermes Methods

A. Spatially-complete Keyframe Designation

In traditional visual SLAM systems, KF creation is typically triggered when the ratio of LMs in the current scene to those in the previous KF falls below a predefined threshold, indicating significant changes in the scene. However, this approach presents challenges when the camera transitions from low-texture areas, where fewer LMs (e.g., approximately 50) are observable, to high-texture regions that may feature hundreds of LMs. Under such conditions, although the scene has changed, the profusion of features in high-texture scenes makes it relatively simple to find LMs that are above the ratio threshold, consequently delaying the generation of new KFs. In texture-variant environments, the significant disparity in the number of LMs across different scenes challenges the effectiveness of the LM thresholding method in accurately representing actual scene changes. To address this, Hermes employs an entropy-based KF designation method, focusing on changes in scene complexity rather than solely on LM association. This approach ensures a more complete and accurate representation of the environment by quantifying scene information entropy.

KeyFrame Designation. As the camera navigates through varying environments, the complexity and uncertainty of the visual information within its field of view shift, resulting in fluctuations in information entropy. To capture a more comprehensive representation of scene changes, Hermes harnesses these entropy variations to enhance the KF selection strategy. New KFs are designated based on detected changes in scene information entropy, providing a richer and more detailed assessment than mere LM association. Specifically, a First-In-First-Out (FIFO) queue stores the most recent k frames. Upon the arrival of a new frame, its image information entropy is calculated and compared against the average entropy of all frames in the FIFO queue. If this ratio exceeds the predefined threshold interval $[t_l, t_h]$, it triggers the insertion of this new KF. Additionally, we observed that scene entropy will fluctuate when illumination changes. Hence, each incoming frame is pre-processed by adaptive histogram equalization [34] to reduce the illumination variation.

B. Bandwidth-aware Map Distillation

After KF designation, the data flow proceeds to the map distillation component of Hermes, where the submap is further distilled and sparsified based on available bandwidth. Our map distillation process consists of two stages: Initially, we establish a criterion to quantify all LMs and KFs based on information gain, which informs our attention mechanism. Subsequently, we conduct an adaptive screening that dynamically adjusts to the available bandwidth.

1) Landmark Quantification with Attention: In visual SLAM, a single KF contains hundreds of LMs, but not all LMs hold equal significance. In practice, we observed that the presence of many redundant LMs does not compromise accuracy. They can interfere with the pose estimation process and consume substantial transmission bandwidth, which motivates us to streamline the LMs.

Definition 1: (Landmark Stability) For an LM p_i observed by n KFs, its stability score is defined as $q_s = n$.

LM with high stability indicates its robustness against view change. Such high-stability LMs provide long-term constraints across multiple KFs in the pose graph.

Nevertheless, selecting LMs solely based on stability scores in feature tracking often results in the premature deletion of

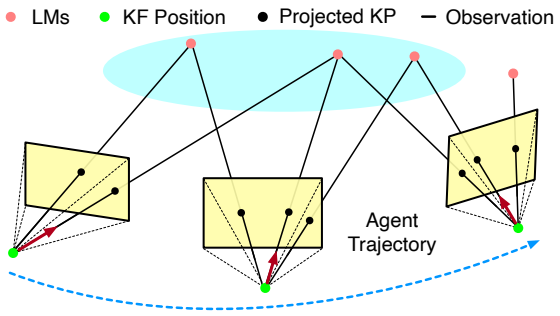

Fig. 6: Observation connection of KFs, LMs and their projected keypoints.

newly observed LMs. Fig. 5a depicts a scene captured by a moving robot. A red circle highlights a chair that has recently entered the robot's field of view, while the television has been within the field of view for an extended period. Relying solely on stability would lead to the LMs from the newly visible chair being rapidly discarded, as shown in Fig. 5b.

To mitigate the inadvertent deletion of fresh LMs, we design an LM information gain quantification method that prioritizes "future points", as shown in Fig. 5c. We incorporate *attention* to each scene of the KF, leveraging the defined information gain to shape the attention mechanism and ensure that those fresh-yet-significant LMs are retained. By retaining only the most useful LMs, we reduce the volume of data transmitted, thereby decreasing bandwidth consumption. Unlike previous approaches [30], our method does not rely on additional sensors (e.g., IMU or wheel speed) to provide supplementary motion information, only leveraging pure visual information to construct the attention.

Definition 2: (Skew Symmetrix Matrix) For vectors $\mathbf{a}$ and $\mathbf{b}$, define $\mathbf{a} \times \mathbf{b} = \mathbf{a}^{\wedge}\mathbf{b}$, where $\mathbf{a}^{\wedge}$ is the skew symmetric matrix built from $\mathbf{a}$. This operator allows the cross product $\mathbf{a} \times \mathbf{b}$ to be expressed as a linear operation.

Theorem 1: (Landmark Infomation Gain) Considering a landmark LM_l, its corresponding projected keypoint in KF kf_k is KP_{kl}. $\boldsymbol{P_l}$ and $\boldsymbol{p_{kl}}$ are vectors from the optical center of the camera to LM_l and KP_{kl}. $\hat{\boldsymbol{R}}$ and $\boldsymbol{t}$ are the last rotation matrix and translation vector estimated by VO. The information gain q_a of LM_l can be quantified with $q_a = f_{det}(\boldsymbol{Q})$, where $\boldsymbol{Q} = \boldsymbol{W^T} \cdot \boldsymbol{W}$ and $\boldsymbol{W} = [(\boldsymbol{p_{kl}})^{\wedge} \cdot (\hat{\boldsymbol{R}} \cdot \boldsymbol{P_l})^{\wedge} - (\boldsymbol{p_{kl}})^{\wedge}]$. $f_{det}(*)$ is the logarithm of the determinate function.

Proof. According to the pinhole camera model, vector $\boldsymbol{P_l}$ is ideally collinear with vector $\boldsymbol{p_{kl}}$ (as shown in Fig. 6). However, due to noise and pose estimation errors, deviations occur. This deviation can be formulated as:

$$\boldsymbol{p_{kl}} \times (\boldsymbol{R} \cdot \boldsymbol{P_l} + \boldsymbol{t}) = \boldsymbol{n_{kl}} \tag{1}$$

where vector $\boldsymbol{n_{kl}} \sim \mathcal{N}(0, \Sigma)$ denotes the projection error (noise term). The camera orientation $\boldsymbol{R}$ can be written as $\boldsymbol{R} = exp(\phi^{\wedge})exp(\psi^{\wedge})$, where ϕ represents the known rotation state, ψ is the rotation increment to be optimized. $\exp(*)$ refers to the exponential map from the Lie algebra to the Lie group, with a property $\exp(\psi^{\wedge}) \approx I$ when $\psi \to 0$. (A detailed discussion on Lie theory can be found in [35]–[37]).

The error function is defined as $f(\psi, \boldsymbol{t}) = \boldsymbol{p_{kl}} \times (\boldsymbol{R} \cdot p_i + \boldsymbol{t})$. Applying the first-order Taylor expansion, we obtain:

$$f(\psi, \boldsymbol{t}) = f(\hat{\psi}, \hat{\boldsymbol{t}}) + \frac{\partial f}{\partial \psi} \cdot (\psi - \hat{\psi}) + \frac{\partial f}{\partial \boldsymbol{t}} \cdot (\boldsymbol{t} - \hat{\boldsymbol{t}}) \tag{2}$$

For the partial derivative $\frac{\partial f}{\partial \psi}$, we have:

$$\begin{aligned} \frac{\partial f}{\partial \psi} &= \frac{\partial[(\boldsymbol{p_{kl}})^{\wedge}(\boldsymbol{R} \cdot \boldsymbol{P_l} + \boldsymbol{t})]}{\partial \psi} \\ &= (\boldsymbol{p_{kl}})^{\wedge} \frac{\partial[(exp(\phi^{\wedge}) \cdot exp(\psi^{\wedge}) \cdot \boldsymbol{P_l} + \mathbf{t})]}{\partial \psi} \\ &= (\boldsymbol{p_{kl}})^{\wedge} \lim_{\psi \to 0} \frac{(exp(\psi^{\wedge}) - \boldsymbol{I}) \cdot exp(\phi^{\wedge}) \cdot \boldsymbol{P_l}}{\psi} \\ &= (\boldsymbol{p_{kl}})^{\wedge} \lim_{\psi \to 0} \frac{\psi^{\wedge}(exp(\phi^{\wedge}) \cdot \boldsymbol{P_l})}{\psi} \\ &= -(\boldsymbol{p_{kl}})^{\wedge}(exp(\phi^{\wedge}) \cdot \boldsymbol{P_l})^{\wedge} = -(\boldsymbol{p_{kl}})^{\wedge}(\hat{\boldsymbol{R}} \cdot \boldsymbol{P_l})^{\wedge} \end{aligned} \tag{3}$$

For the partial derivative $\frac{\partial f}{\partial \mathbf{t}}$, we have:

$$\frac{\partial f}{\partial t} = \frac{\partial[(\boldsymbol{p_{kl}})^{\wedge}(\mathbf{R} \cdot \boldsymbol{P_l} + \boldsymbol{t})]}{\partial t} = (\boldsymbol{p_{kl}})^{\wedge} \tag{4}$$

Then, we can substitute Eq.(3) and Eq.(4) into Eq.(2), yielding:

$$\begin{aligned} f(\psi, \mathbf{t}) &= (\boldsymbol{p_{kl}})^{\wedge}(\hat{\boldsymbol{R}} \cdot \boldsymbol{P_l} + \hat{\boldsymbol{t}}) - (\boldsymbol{p_{kl}})^{\wedge}(\hat{\boldsymbol{R}} \cdot \boldsymbol{P_l})^{\wedge}\psi + (\boldsymbol{p_{kl}})^{\wedge}(\boldsymbol{t} - \hat{\boldsymbol{t}}) \\ &= (\boldsymbol{p_{kl}})^{\wedge} \cdot \hat{\boldsymbol{R}} \cdot \boldsymbol{P_l} - [(\boldsymbol{p_{kl}})^{\wedge} \cdot (\hat{\boldsymbol{R}} \cdot \boldsymbol{P_l})^{\wedge} \cdot (\boldsymbol{p_{kl}})^{\wedge}] \begin{bmatrix} \psi \\ \mathbf{t} \end{bmatrix} \end{aligned} \tag{5}$$

By substituting Eq.(5) to Eq.(1), we can get:

$$(\boldsymbol{p_{kl}})^{\wedge} \cdot \hat{\boldsymbol{R}} \cdot \boldsymbol{P_l} = [(\boldsymbol{p_{kl}})^{\wedge} \cdot (\hat{\boldsymbol{R}} \cdot \boldsymbol{P_l})^{\wedge} - (\boldsymbol{p_{kl}})^{\wedge}] \begin{bmatrix} \psi \\ \boldsymbol{t} \end{bmatrix} + \boldsymbol{n_{kl}} \tag{6}$$

Eq.(6) can be regarded as a linear system, where the left side of the equation is the $Measurements$, $[\psi, \boldsymbol{t}]^T$ is the $Status$ and $\boldsymbol{n_{kl}}$ is the $Noise$. The matrix $\boldsymbol{W} = [(\boldsymbol{p_{kl}})^{\wedge} \cdot (\hat{\boldsymbol{R}} \cdot \boldsymbol{P_l})^{\wedge} - (\boldsymbol{p_{kl}})^{\wedge}]$ is the information matrix that indicates how the status of the system affects the measurement of pose. To facilitate computing the feature selection problem, we employ the log-determinant function $f_{det}(*)$ as a quantification for the information gain matrix.

Relationship Between Log-Det and Pose Uncertainty. The determinant $\det(\boldsymbol{Q})$ measures the system's certainty in pose estimation. Geometrically, it is inversely proportional to the volume of the confidence ellipsoid described by the covariance matrix. Equivalently, it is directly proportional to the volume of the ellipsoid defined by the information matrix, which represents the amount of information available for estimating the pose. A larger determinant indicates a smaller confidence ellipsoid, meaning lower uncertainty and higher confidence in the estimated pose [38], [39]. This makes it ideal for evaluating information gain, as it compactly quantifies a landmark's contribution to reducing pose uncertainty.

2) Landmark Sparsification: Intuitively, by quantifying and resequencing the importance of LMs through attention and stability, we can minimize data transmission volume by selecting high-value LMs using a greedy approach. However, this method significantly increases computational complexity due to the need to compute the gain for all remaining features at each iteration. The bandwidth-aware LM selection challenge is akin to the NP-hard knapsack problem, where LMs are

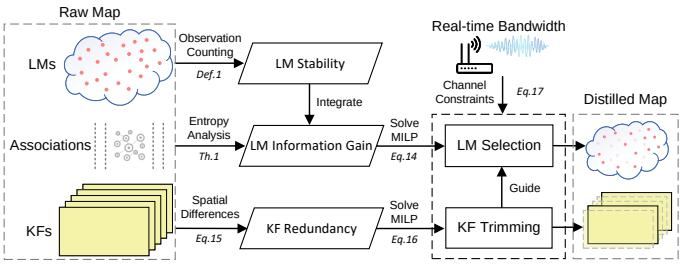

Fig. 7: Workflow of map distillation.

"items," their bandwidth costs are "weights," their contributions to SLAM are "values," and the dynamic bandwidth resources act as the capacity-varying "knapsack." The objective is to select feature points that optimize SLAM accuracy dynamically based on real-time bandwidth.

Definition 3: (Submodularity [40]) A set function $f : 2^V \rightarrow \mathbb{R}$, defined on the subsets of a finite set V, is called submodular if every two subsets A and B of V with $A \subseteq B$, and every element $x \in V \setminus B$, satisfies:

$$f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B) \tag{7}$$

Submodularity indicates the decreasing trend in the marginal gain of adding an element to a set. Thus, for a feature set exhibiting submodularity, it is feasible to replace the entire set with a subset of features while still providing desirable information in pose graph optimization.

Lemma 1: (Submodularity of Log-det Function) The set function $f_{det}(\boldsymbol{Q})$ defined in Th. 1 is monotone and submodular.

Proof. To prove that $f_{det}(\boldsymbol{Q})$ is submodular, we focus on proving that for any $A \subseteq B \subseteq N$ and $l \in N \setminus B$:

$$f(A \cup \{s\}) - f(A) \geq f(B \cup \{l\}) - f(B) \tag{8}$$

where $f(S) = \log\det(\boldsymbol{Q}_S)$ and $\boldsymbol{Q}_S$ is the submatrix of the information matrix $\boldsymbol{Q}$ corresponding to set S.

First, define the gain of adding a LM l to set S:

$$\Delta f(S, l) = f(S \cup \{l\}) - f(S) \tag{9}$$

Our goal is to demonstrate that $\Delta f(A, l) \geq \Delta f(B, l)$. The log-determinant gain can be expressed as:

$$\Delta f(S, l) = \log\left(\frac{\det(\boldsymbol{Q}_{S\cup\{l\}})}{\det(\boldsymbol{Q}_S)}\right) = \log\left(1 + \boldsymbol{q}_l^T \boldsymbol{Q}_S^{-1} \boldsymbol{q}_l\right) \tag{10}$$

Since $A \subseteq B$, we have $\boldsymbol{Q}_A^{-1} \succeq \boldsymbol{Q}_B^{-1}$, meaning that $\boldsymbol{Q}_A^{-1} - \boldsymbol{Q}_B^{-1}$ is positive semi-definite. This implies:

$$\boldsymbol{q}_l^T \boldsymbol{Q}_A^{-1} \boldsymbol{q}_l \geq \boldsymbol{q}_l^T \boldsymbol{Q}_B^{-1} \boldsymbol{q}_l \tag{11}$$

Thus:

$$\log\left(1 + \boldsymbol{q}_l^T \boldsymbol{Q}_A^{-1} \boldsymbol{q}_l\right) \geq \log\left(1 + \boldsymbol{q}_l^T \boldsymbol{Q}_B^{-1} \boldsymbol{q}_l\right) \tag{12}$$

Hence, we have:

$$\Delta f(A, l) \geq \Delta f(B, l) \tag{13}$$

This completes the proof that $f_{det}(\boldsymbol{Q})$ is submodular.

This shows that the information gain from adding a LM depends on both the new LM and the existing structure of the information matrix, capturing the combined effect of all independent information. By calculating information gain and selecting LMs with the highest gain, C-SLAM system can maintain the accuracy of pose estimation.

LandMark Selection. We formulate the LM sparsification task as a mixed integer programming problem, which is a commonly used paradigm of map sparsification in past works [28] [41]. Given a LM set $\{lm_i\}$, along with its corresponding stability vector $\boldsymbol{q}_s$ and information gain vector $\boldsymbol{q}_a$ calculated by Def.1 and Th.1:

$$\begin{aligned} &\min_{\boldsymbol{x},\lambda_1,\lambda_2} (\alpha \boldsymbol{q}_s + \beta \boldsymbol{q}_a)\boldsymbol{x} + \lambda_1 \boldsymbol{I}^T \boldsymbol{\xi} + \lambda_2 \boldsymbol{I}^T \boldsymbol{\phi} \\ &s.t. \boldsymbol{A}x + \boldsymbol{\xi} \geq K\boldsymbol{I}; \boldsymbol{B}\boldsymbol{x} + \boldsymbol{\phi} \geq \boldsymbol{I} \end{aligned} \tag{14}$$

Where α and β are the weighting parameters balancing the importance of LM stability and information gain. Hermes prioritizes LMs based on their expected contribution, selecting those with higher information gain. When the camera undergoes severe rotation (i.e., a significant difference between the rotation matrices of two keyframes), β is set to a higher value to favor LMs that can be reliably tracked over an extended period. Conversely, in scenarios where robots frequently intersect, which can be quantified by the number of loop closures, α is increased to ensure the stability of long-term landmark associations. $\boldsymbol{x}$ is a binary vector whose i^{th} element indicates whether lm_i is selected or not (lm_i will be deleted if $\boldsymbol{x}_i = 1$). $\boldsymbol{A}$ is a binary matrix in which the element $\boldsymbol{A}_{ij}$ denoting lm_i is visible or not in kf_j. K is the map compression factor indicating the maximum number of LMs that can be observed in each KF. K directly determines the sparsification level, which should be adjusted based on the current available bandwidth to achieve bandwidth-aware sparsification. Hence, it will be regulated by the Bandwidth Coordinator introduced later. B is a binary matrix in which the element $\boldsymbol{B}_{ij}$ denoting lm_i is projected in grid g_j or not. This constraint encourages the homogeneous distribution of keypoints within the frame, preventing the pose estimation problem from becoming ill-conditioned. ξ and ϕ are soft constraints to ensure the problem is solvable.

By introducing an attention mechanism that emphasizes the predictive effect on the future, Hermes ensures that more LMs important for future predictions are retained while eliminating useless LMs, thereby reducing bandwidth consumption.

3) Keyframe Trimming: In §IV-A, we outlined the strategy for KF Designation, designed to generate KFs uniformly, even in scenes with varying textures. Recall that to ensure the accuracy of local pose estimation and sufficient LM generation, we initially created a dense local map, where the KFs remain overly redundant. Therefore, to optimize bandwidth usage, we apply a secondary trimming of KFs before their transmission. This step is crucial as it complements the LM sparsification strategy implemented in the submap. By trimming the KFs before transmission, we further reduce bandwidth consumption. KF trimming is executed alternatively with LM sparsification.

KF trimming is guided by two key principles to ensure efficient data processing and accurate mapping: 1) it is governed by sparsity, which requires that KFs be either spatially distant or have significantly different viewing angles to reduce redundancy; 2) it depends on the connectivity of the remaining LMs

after sparsification, mandating that each KF captures many high-value LMs to secure reliable inter-frame associations.

KeyFrame Redundancy. We consider the spatial difference sd between two keyframes, considering both rotation and translation, is defined as:

$$sd = w_R \cdot \arccos\left(\frac{\text{trace}(R_1^T R_2) - 1}{2}\right) + w_T \cdot \|t_1 - t_2\| \quad (15)$$

where R_1, R_2 are the rotation matrices of the two KFs, and t_1, t_2 are their corresponding translation vectors. w_R and w_T are weights to normalize rotation and translation differences. A smaller sd signifies a greater spatial similarity, indicating that one KF can effectively replace the other.

KeyFrame Trimming. Given a set of KFs $\{kf_j\}$ and an entropy factor $\boldsymbol{Th_e}$, KF trimming task can be formulated as:

$$\begin{aligned} &\min_{\boldsymbol{y},\lambda} \boldsymbol{s}^T \boldsymbol{y} + \lambda \boldsymbol{\varepsilon} \\ &s.t.\ \ \boldsymbol{E}\boldsymbol{y} + \boldsymbol{\varepsilon} \geq \boldsymbol{Th_e} \end{aligned} \quad (16)$$

where $\boldsymbol{s}$ is a vector whose j^{th} element indicates the spatial difference of kf_j, larger $\boldsymbol{s}_j$ indicates that kf_j is more representative. $\boldsymbol{y}$ is a binary vector whose j^{th} element indicates whether kf_j is selected or not (kf_j will be deleted if $\boldsymbol{y}_j = 1$). ε is the soft constraint to ensure the problem is solvable. $\boldsymbol{E}$ is the observation matrix denoting the index of LMs observed by kf_j. Note that the degree of trimming is determined by $\boldsymbol{Th_e}$, which will also be guided by the bandwidth coordinator introduced later to adapted to the available bandwidth.

We can solve Eq.(16) in the same way as we do with Eq.(14), where we aim to maximize the distance between KFs while including as many effective LMs as possible.

4) Bandwidth Coordinator: Up to now, all previous discussions have been focused on addressing the distillation of LMs and KFs, independent of bandwidth considerations. Unlike the solidified map sparsification paradigm, where the map size is pre-defined prior to deployment [42] [28] [27], Hermes aims to integrate the real-time bandwidth conditions with the sparsification of LMs and trimming of KFs, allowing agents to dynamically adjust the granularity of LMs and KFs based on available bandwidth. This strategy optimizes resource usage and maximizes accuracy. Specifically, each agent maintains a bandwidth monitor to sniff real-time bandwidth situations. Then, it incorporates available bandwidth into the aforementioned selection and pruning of KFs and LMs.

Assuming an LM occupies M bytes and a KF occupies N bytes, the sniffed available bandwidth is B, we can adaptively adjust the sparsification factor K and Th_e in Eq.(14) and Eq.(16) correspondingly to meet the remaining bandwidth in wireless communication channel:

$$\frac{K}{N_{LMs}} \cdot D_{LM} + g(Th_e) \cdot D_{KF} \leq B_t \cdot \Delta t \quad (17)$$

where N_{LMs} is the average number of LMs observed by a KF in an un-distilled submap; $g(*)$ denotes the KF sparsification ratio with Th_e; D_{LM} and D_{KF} denotes the data size of original LMs or KFs; B_t is the available bandwidth at time t. Δt is the data transmission interval, in Hermes it is set to 5 seconds, which means the submap is distilled every 5 seconds and transmitted in the following 5 seconds.

Additionally, solving Eq.(14) and Eq.(16) is NP-Hard because they are MILP problems, where in the worst case the computation complexity can be $O(2^n)$, n is the number target variable (i.e., LMs and KFs). However, it is noticeable that the number of KFs is substantially smaller than the number of LMs. We address this inequality before each transmission and prioritize solving the latter problem to simplify the optimization process. Following KF trimming, any LM not observed by any KFs is removed from the local map. Then, LMs are sparsified to meet the requirements of Eq.(17). We strategically connect the key parameters of the LM and KF sparsification optimization problems (K and Th_e) to bandwidth constraints in Eq.(17). By executing these optimizations alternately, we tightly couple the sparsification processes for LMs and KFs, ultimately achieving a global optimum that aligns perfectly with the available bandwidth. Putting them together is called bandwidth-aware map distillation (see Fig. 7).

C. Congestion-response Data Transferring

While the previously proposed map distillation method reduces communication data, communication congestion remains an issue in practical scenarios. Due to low bandwidth availability in cases of congestion or connection interruption, even maps that have been extensively compressed may still fail to be transmitted, leading to discontinuous maps on the server side. In the C-SLAM paradigm, each agent's map is merged through *Loop Closing*, which computes the relative transformation based on overlapped areas. However, to maximize mapping efficiency in robot swarms, we aim to minimize overlap, making loop closure sparse. Communication congestion during loop closure can cause missed loops, resulting in map distortion and incompleteness. To address this, Hermes introduces Submap Assessor on the server side.

Credibility-based Submap Assessor (CSA). As the swarm scales, latency in processing submap merging requests increases, and communication instability can lead to temporary data association errors, correctable only with sufficient global information. To address this, we propose the Credibility-based Submap Assessor to ensure global map accuracy. Hermes evaluates submap quality using Transmission Stability (the product of successful transmission rate and bandwidth used) and Map Connectivity (average observation times of all LMs in the submap). Higher values indicate better data association and submap quality. If loops are detected, the connectivity score is set to 1, indicating strong connections. In dynamic bandwidth, submaps with continuity and coherence are prioritized for merging. A normalized weighted sum of these criteria is computed for each incoming submap, and a priority queue on the edge server determines the merging order.

CSA can be seamlessly integrated on the device side, allowing agents to locally evaluate the quality of map points based on predefined criteria. This capability enables Hermes to transcend the limitations of conventional centralized edge-assisted frameworks. Agents with enhanced communication capabilities can serve as relays, facilitating the transmission of data from other agents. Specifically, when bandwidth is available, relay agents can accept relay requests from other

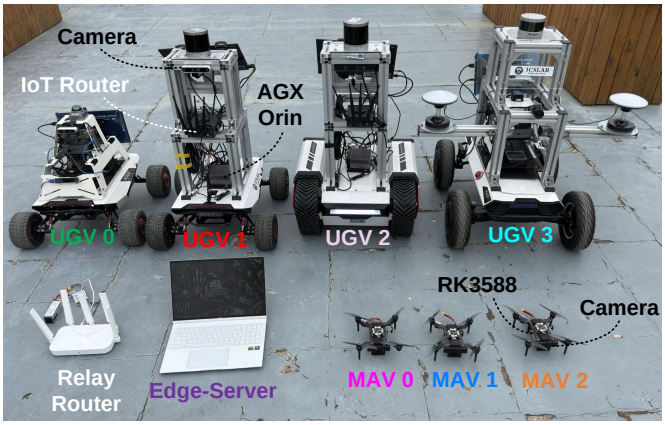

Fig. 8: The Hermes Prototype.

agents. Upon establishing a data transmission link, the relay agent first creates a map buffer for the requesting agent to store map data. Subsequently, it evaluates transmission stability and map connectivity between the two agents to dynamically adjust the bandwidth allocation strategy. To ensure data integrity, a minimum bandwidth requirement of 1 Mbps is enforced for each robot. This flexibility allows Hermes to be deployed in diverse heterogeneous robotic swarms, including scenarios involving collaborative mapping between heterogeneous robots. Experimental results are provided in §VII-B.

D. Consistency-oriented Map Merging

1) BookKeeping: Hermes manages all agents through a bookkeeper. Upon initialization, each agent attempts to connect to the edge server using a pre-stored IP address and port. Once the back-end receives at least 30 KFs from each agent, the edge server evaluates the quality of each agent's submap and selects the local coordinates of the most stable agent as the system's global coordinate.

2) Loop Detection and Map Merging: In traditional C-SLAM loop closure, geometric verification is conducted after identifying candidate loop pairs using Bag-of-Words (BoW). However, this approach can struggle to match ORB features between the current frame and the candidate KF due to a sparsified map. We leverage our accurate estimation of relative poses between adjacent KFs, obtained through the KF designation and Map Distillation strategy. For a candidate KF, we select m neighboring KFs and apply the relative transformations to construct a conceptual multi-camera system. We take advantage of the 17-point algorithm [43] to determine the relative motion between two such systems, as in [17]. Furthermore, we employ PCM [44] to eliminate outliers.

We believe the agents are only responsible for local mapping and path planning. Hence, once map synchronization is complete, previous submap components are discarded on the agents to ensure a low memory footprint.

V. Implementation

We have implemented Hermes using the state-of-the-art ORB-SLAM3 framework [2], integrated with the Robot Operating System (ROS) [45][1]. Fig. 8 illustrates the prototype system of Hermes. Within this system, the agents are responsible for running tracking and local mapping threads, while the edge-server handles loop closing and map fusion threads.

Device Side. We deploy the front-end of Hermes on each agent. The attentional landmark selection is performed in the local mapping thread, where we take advantage of Gurobi Optimizer [46] to solve Map Distillation and KeyFrame Trimming problems. The selected KFs and corresponding MPs are encoded into a binary stream using the Cereal library and uploaded to the edge-server through a 4G IoT router. We utilize the "Exactly Once" mode of MQTT protocol [47] in data transmission. The IP address of the edge server is pre-stored on each agent's local machine. We adjust the sparsification factor by monitoring whether the binary submap is successfully sent via the MQTT protocol and checking the bandwidth before sparsification. On low-speed robots, we treat bandwidth and latency as continuously varying.

Edge Side. When the edge server receives map data from an agent, it checks if the agent is registered in the bookkeeper. New agents are assigned a unique *robot_id*, while previously registered agents that reconnect after a communication interruption have their registration restored and are re-localized. If re-localization fails, the agent is treated as new.

VI. Evaluation

A. Evaluation Setup

Platform. A laptop featuring an Intel Ultra7-155H CPU, 32GB of RAM, and running Ubuntu 20.04 serves as the edge server. The agents are deployed across four unmanned ground vehicles (UGVs) and three micro aerial vehicles (MAVs). The UGVs are equipped with Nvidia Jetson AGX Orin 32GB (8 x Cortex A78) and a depth camera (either Intel Realsense D455 or Astra Pro), while the MAVs are outfitted with RockChip RK3588 (4 x Cortex A76 + 4 x Cortex A55) and a custom-built stereo camera. Map data are transmitted from the agents to the server via 4G IoT routers.

Datasets. We adopt both indoor and outdoor SLAM datasets to test the generalization capability of Hermes: 1) *ICL-NUIM (I_N)* [48] is an indoor SLAM dataset for localization and reconstruction; 2) *EuRoC* [49] is an indoor SLAM dataset collected by a MAV; and 3) *S3E* [50] is a large-scale outdoor dataset for collaborative SLAM. We reorganized it into KITTI format [51] for utilization by SwarmMap and stereo mode of Swarm-SLAM. We also use the *ContextCapture* software to construct a 3D model of our campus and import it into the Gazebo [52] simulation environment, enabling us to deploy a sufficient number of robots for scalability examination.

Baselines. We compare Hermes against two state-of-the-art SLAM systems: 1) *SwarmMap* (**NSDI'22**) [14] is a framework that enhances the scalability of C-SLAM services by optimizing collaborative strategy in edge computing environments. 2) *COVINS-G* (**ICRA'23**) [17] is a generalized back-end for C-SLAM that can be integrated with any front-end, while also considering low-bandwidth communication; 3) *Swarm-SLAM*

[1]Our code is available at https://github.com/whu-gr/Hermes.git

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3549367

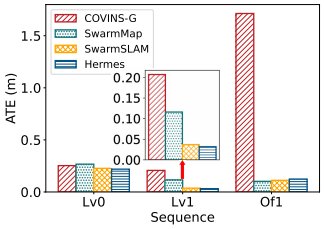

Fig. 9: ATE in I_N.

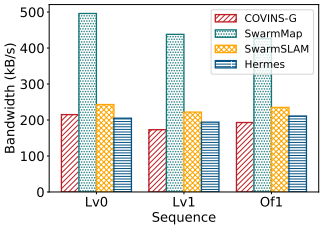

Fig. 10: Bandwidth in I_N.

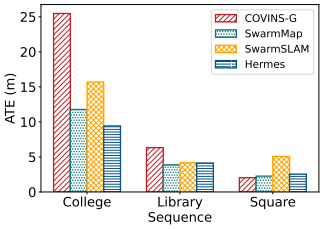

Fig. 11: ATE in S3E.

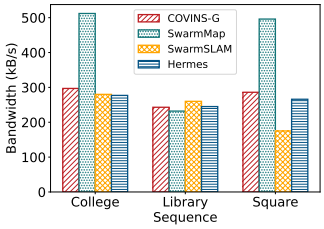

Fig. 12: Bandwidth in S3E.

(**RAL'24**) [16] is a low-bandwidth multi-modal decentralized framework for multi-robot systems.

Metrics. We evaluate the system with mapping error and communication robustness by the following metrics: 1) Trajectory Error. We use the Absolute Trajectory Error (ATE) to evaluate the discrepancy between the robot's trajectory and the ground truth trajectory, measured in meters. 2) Bandwidth Usage. 3) Agent Lost Rate. It represents the proportion of time during which an agent is lost due to either tracking failures or communication losses, relative to the total operational time of the agent throughout the mapping process.

Since Swarm-SLAM is a distributed system, we focus on comparing its ATE and single-node communication bandwidth consumption (Fig. 9 $\sim$ 12). For collaborative bandwidth comparison, which involves controlling the central router bandwidth, we use COVINS-G and SwarmMap as baseline methods representing edge-assisted approaches. (Fig. 13 $\sim$ 18)

B. Overall Performance

Indoor Scenarios.

We initially evaluated the tracking error and bandwidth usage of Hermes and three baseline systems using the indoor dataset (ICL-NUIM), as illustrated in Fig. 9 and Fig. 10. The results demonstrate that Hermes achieves lower trajectory error than COVINS-G while maintaining comparable bandwidth consumption. Additionally, it reduces bandwidth usage by 50% compared to SwarmMap, without compromising accuracy. Notably, in sequences such as Lv1 where the camera navigates through scenes with varying textures, our enhanced KF designation method more comprehensively profiles the environment. Furthermore, our motion-based attention mechanism in LM quantification effectively retains "future" LMs, establishing long-term associations across frames. These cascading optimizations in LM and KF sparsification enable Hermes to strike an effective balance between data transfer efficiency and mapping precision, ensuring both robust tracking and reduced bandwidth consumption.

Outdoor Scenarios. We then assess the tracking error and bandwidth utilization using the outdoor S3E dataset, which presents significant challenges for C-SLAM systems in large-scale scenarios. Specifically, sparse mapping tends to reduce scene representation, which causes COVINS-G to fail in detecting loop closures during the College Sequence, as depicted in Fig. 11. In contrast, Hermes excels by transferring precise and concise information that enhances overall scene perception. While SwarmMap demonstrates lower bandwidth consumption and reduced ATE in Library sequences—where three agents navigate the same area, showcasing its efficacy in repetitive scenarios such as inspection tasks—it struggles in exploratory mapping tasks in unknown environments. Here, the communication bandwidth significantly increases due to the necessity for continuous cold starts and the system's inability to eliminate redundant information effectively. In the College and Square sequences, Hermes requires only about half the bandwidth of SwarmMap, achieving substantial bandwidth efficiency without significantly compromising accuracy, as demonstrated in Fig. 12. Notably, our edge-assisted system achieves the same level of bandwidth consumption as the distributed Swarm-SLAM system on the College and Library sequences while maintaining the high-precision advantage of a centralized system. Additionally, the centralized architecture ensures that global map data can always be accessed on the edge server, rather than requiring the retrieval of all robots, as in distributed systems—a crucial advantage in scenarios such as post-disaster rescue operations.

Impact of Available Bandwidth. As depicted in Fig. 13, we evaluate these systems under varying bandwidth conditions in the Square Sequence. Without bandwidth restrictions, all three methods achieve similar mapping accuracy because they can fully synchronize information between agents and the server. However, when the bandwidth is limited to 30 Mbps, COVINS-G's accuracy declines first due to its failure to detect loops between agents and issues arising from out-of-order data transmissions caused by network congestion. Although SwarmMap also experiences performance degradation under these conditions, it retains the ability to merge all submaps. When the bandwidth is further reduced to 10 Mbps, only Hermes completes the global map merging. It does experience a slight reduction in map accuracy, a consequence of the restricted communication resources limiting the amount of constraint information available.

C. Ablation Study

Performace of KeyFrame Designation. We evaluate the impact of the KeyFrame Designation module on SLAM accuracy improvement and illustrate how it works in cascade with the KeyFrame Trimming module to balance mapping accuracy and data transmission efficiency. Specifically, we compare Hermes with the KD module (w_KD) against Hermes using the default keyframe selection method from ORB-SLAM3 (wo_KD), as shown in Fig. 14. Except for the Lv0 sequence, the integration of entropy-based KF designation consistently improves accuracy. Notably, in Lv1 and Of1—both featuring significant texture variations—KeyFrame Designation actively

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3549367

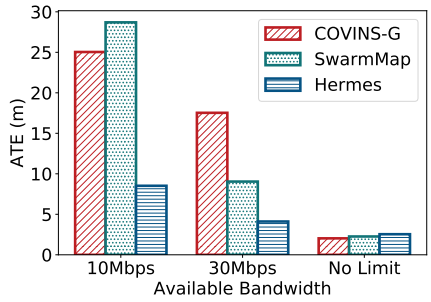

Fig. 13: ATE vs. Bandwidth.

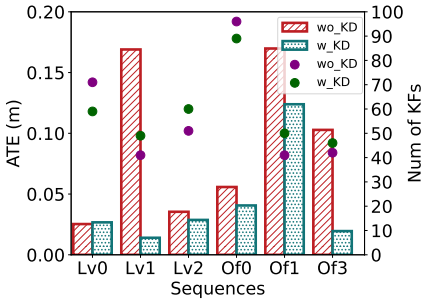

Fig. 14: Ablation for KF Designation.

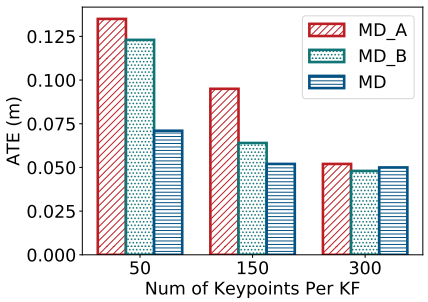

Fig. 15: Ablation for LM Quantification.

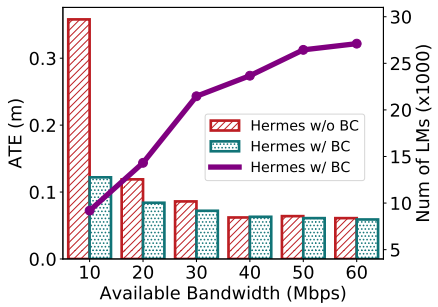

Fig. 16: Ablation for Bandwidth Coordinator.

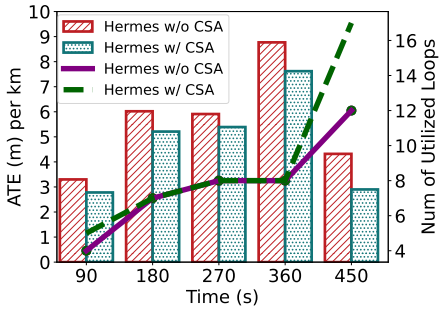

Fig. 17: Ablation for CSA.

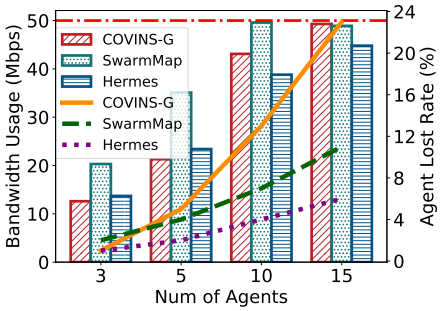

Fig. 18: Bandwidth consumption and agent lost rate with swarm scale.

creates more keyframes with a uniform distribution, effectively summarizing the environment. For instance, in the Lv1 sequence, adding only 8 additional KFs at critical locations reduced the mapping error to just 8.3% of its original value.

Performance of LM Quantification. In the Map Distillation component, we conduct ablation tests to evaluate different quantification strategies. We test two variants: MD_A where LMs are quantified by stability using a greedy approach to select the most frequently observed LMs, and MD_B, where LMs are quantified by spatial distribution, employing ANMS [53] to ensure a homogeneous spatial distribution of keypoints. The proposed LM quantification method incorporating attention is denoted as MD. We test MD, MD_A and MD_B in the MH05 sequence of the EuRoC dataset with various sparsification ratios, as illustrated in Fig. 15. Under rapid camera motion, selecting LMs based on stability risks omitting newly added LMs, leading to a trajectory drift with localization errors increasing over 150% when LMs are reduced from 300 to 50. Hermes, using the attention-based quantification method, effectively retrieves these "future" LMs, enhancing pose estimation smoothness. The effectiveness of MD_B significantly diminishes when LMs per KF drop below 50, as it fails to focus on critical minimal information.

Performance of Bandwidth Coordinator. We assess the performance of the Bandwidth Coordinator and its guidance on the LM sparsification and KF trimming modules. Specifically, we compare bandwidth-aware sparsification (*Hermes (w/BC)*) with a fixed sparsification ratio (*Hermes (w/o BC)*), where $K = 100$ is set in Eq.(14), on 5 agents. Fig. 16 depicts the ATE and corresponding LM volume under different bandwidth levels on the EuRoC dataset. At bandwidths not lower than 30 Mbps, Hermes(w/BC) utilizes sufficient bandwidth to avoid the loss of potential information, achieving a similar ATE compared to Hermes(w/o BC). As available bandwidth decreases, Hermes(w/ BC) dynamically adjusts the sparsification level, ensuring minimal map transmission requirements. The degree of LM sparsification, represented by the purple line in Fig. 16, adjusts instantly as bandwidth decreases. Despite this, ATE remains low and acceptable for C-SLAM functionality, a testament to Hermes' effective compression and pruning strategies. In comparison, due to the fixed sparsification ratio strategy, Hermes(w/o BC) quickly encounters communication bottlenecks, causing escalating latency and eventually hindering real-time message transmission.

Performance of CSA. We evaluated the effectiveness of the CSA in ensuring accurate global map merging under substantial data congestion, using a 15Mbps bandwidth with three agents in the Square sequence. In such a congested channel, unpredictable network collisions may result in data loss from any agents. To more accurately reflect real-time mapping errors, we employed a normalized ATE (i.e., ATE (m) per kilometer) to equitably assess the real-time ATE throughout the mapping. We record ATE and the number of loops every 90 seconds, with the results presented in Fig. 17. At system startup, all agents synchronously send messages to the server to ensure a robust initialization. By 90s, Hermes (w/ CSA) detects more loop closures, establishing an initial accuracy advantage. Before 360s, the robot trajectories rarely intersect, causing ATE to increase significantly due to odometry drift. In the final stage of mapping (360s–450s), the trajectories of the three robots exhibit substantial overlap, leading to a high volume of concurrent loop closure requests. Under limited

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2025.3549367

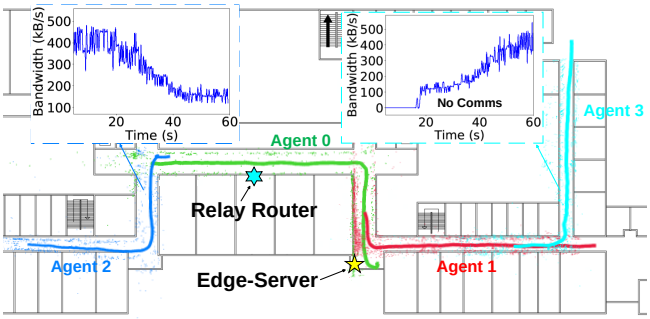

Fig. 19: Indoor collaborate mapping with four agents.

bandwidth conditions, Hermes (w/ CSA) prioritizes the more reliable submaps for global map optimization. As a result, it utilizes five additional loop constraints compared to Hermes (w/o CSA) and ultimately achieves a 32.9% reduction in error.

D. Scalability Analysis

We further investigate the scalability of Hermes by incrementally increasing the number of robots within the gazebo-simulated environment. We measure the average position update latency and bandwidth consumption of the robots, while the available bandwidth for the edge server is constrained to a maximum of 50 Mbps. As illustrated in Fig. 18, when the swarm size is small, all three methods exhibit satisfactory performance. However, as the number of robots increases, the likelihood of network congestion rises. Among the three methods, COVINS-G is the first to experience significant packet loss, leading to occasional robot disconnections. SwarmMap demonstrates the ability to prioritize "urgent" robots, ensuring low latency; however, its bandwidth usage grows linearly in exploratory scenarios, ultimately resulting in increased latency. In contrast, Hermes adopts a "yielding" behavior as the bandwidth limit is approached, dynamically adjusting the volume of uploaded data to maintain a minimal yet stable level of information synchronization.

VII. Real-World Case Studies

We conducted real-world case studies in both indoor and outdoor scenarios to demonstrate the reliability of Hermes under unstable and low-bandwidth communication conditions.

A. Indoor Experiment with WiFi Direct

We conduct mapping experiments in an indoor environment where four agents are directly connected to the edge server via a local area network. Consequently, the communication between the agents and the server is unstable and even intermittent due to signal obstruction, attenuation, and diffraction, which facilitates testing our method's awareness of fluctuating bandwidth and tolerance to packet loss.

Fig. 19 shows the collaborative mapping of four agents. We placed the edge-server in an interaction that connects two hallways and deployed a 2.4G router serving as a relay to expand communication coverage. Agents 0 and 1, being closer to the access point (AP), contributed a higher volume of map points. Agent 2, as it moved farther from the AP, utilized Hermes' capability to dynamically adjust the map's sparsification ratio in response to available bandwidth, thereby ensuring continuous mapping. Initially, outside the communication range, Agent 3 began its operation by storing map data locally in the Map Buffer. As it approached the AP and established the connection, the stored historical data was progressively uploaded to the server. This adaptation allowed for an increase in both the volume and quality of transmitted information, thereby maintaining the integrity and completeness of data synchronization across all agents.

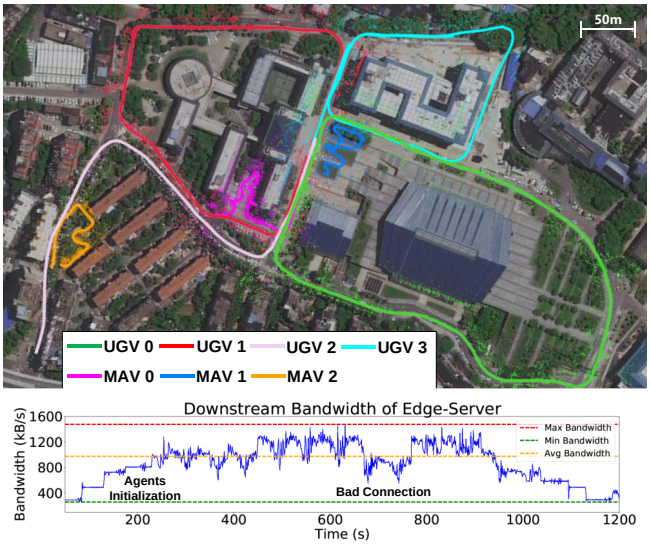

Fig. 20: Outdoor collaboration mapping with seven agents.

B. Outdoor Experiment with Remote Connection

We conducted mapping experiments on a campus with a primary emphasis on evaluating the system's bandwidth consumption and mapping quality in large-scale environments. In this setup, UGVs communicate with an edge server via SD-WAN provided by IoT routers, whereas MAVs, which are unable to carry IoT routers for remote networking, establish communication with UGVs through a wireless ad-hoc network. Consequently, MAVs must transfer mapping data to UGVs immediately upon establishing a communication link. Simultaneously, UGVs are required to efficiently allocate bandwidth between uploading their own mapping data to the edge server and relaying data from MAVs.

Fig. 20 illustrates the collaborative mapping process of our heterogeneous swarm. The trajectory spans approximately 3,100 meters, during which Hermes efficiently synchronizes data from all agents and ultimately selects 1,973 KFs and 51,190 MPs to generate the final map.

The waveform graph below depicts the downlink bandwidth utilization of the edge server. As the number of robots increases, the bandwidth demand on the edge server rises steadily before eventually stabilizing. Hermes ensures uninterrupted mapping by prioritizing the transmission of fewer but more critical LMs when communication is obstructed while conveying sufficient information to maintain mapping accuracy when communication is smooth. Throughout the

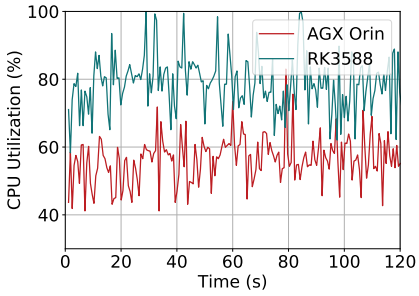

Fig. 21: CPU usage on various platforms.

TABLE I: Average runtime of Hermes' procedures.

Devices	Agent-Side				Server-Side
	Tracking	LocalBA	Distillation	Uploading	Merging
AGX Orin	24 ms	187 ms	329 ms	196 ms	1218 ms
RK3588	33 ms	225 ms	413 ms	184 ms	

mapping process, the average bandwidth utilization is 972 kB/s, with peak bandwidth usage reaching 1,477 kB/s.

CPU usage and running time during field testing are reported in Fig. 21 and Table I. Although map distillation incurs some computational overhead, it remains suitable for deployment on low-power devices and is significantly less time-consuming than the subsequent map merging process. Furthermore, since map distillation is decoupled from odometry, it only introduces a minor delay to the map upload process without directly impacting the efficiency of odometry. Additionally, as communication operates in a relatively asynchronous manner, the added computation time results in a net positive effect, outweighing the overhead caused by communication congestion.

VIII. Conclusion

In this paper, we introduced Hermes, an efficient edge-assisted C-SLAM framework with bandwidth-aware data compression. We optimized C-SLAM data transmission across agent, communication socket, and server levels using various strategies to enhance bandwidth utilization. Our bandwidth-aware distillation strategy maximizes data accuracy by optimally using available bandwidth under dynamic network conditions. Extensive evaluations on public datasets and real-world case studies demonstrate Hermes' effectiveness, reducing bandwidth usage by up to 50% compared to SOTA C-SLAM methods while maintaining mapping accuracy.

References

[1] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE transactions on robotics(TRO)*, vol. 34, no. 4, pp. 1004–1020, 2018.

[2] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics(TRO)*, vol. 37, no. 6, pp. 1874–1890, 2021.

[3] Y. Chang, K. Ebadi, C. E. Denniston, M. F. Ginting, A. Rosinol, A. Reinke, M. Palieri, J. Shi, A. Chatterjee, B. Morrell *et al.*, "Lamp 2.0: A robust multi-robot slam system for operation in challenging large-scale underground environments," *IEEE Robotics and Automation Letters(RAL)*, vol. 7, no. 4, pp. 9175–9182, 2022.

[4] P.-Y. Lajoie and G. Beltrame, "Swarm-slam: Sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 475–482, 2024.

[5] A. Cramariuc, L. Bernreiter, F. Tschopp, M. Fehr, V. Reijgwart, J. Nieto, R. Siegwart, and C. Cadena, "maplab 2.0–a modular and multi-modal mapping framework," *IEEE Robotics and Automation Letters(RAL)*, vol. 8, no. 2, pp. 520–527, 2022.

[6] Y. Tian, Y. Chang, F. H. Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems," *IEEE Transactions on Robotics(TRO)*, vol. 38, no. 4, 2022.

[7] H. Sun, Y. Qu, C. Dong, H. Dai, Z. Li, L. Zhang, Q. Wu, and S. Guo, "All-sky autonomous computing in uav swarm," *IEEE Transactions on Mobile Computing (TMC)*, 2024.

[8] A. Dhakal, X. Ran, Y. Wang, J. Chen, and K. Ramakrishnan, "Slam-share: visual simultaneous localization and mapping for real-time multi-user augmented reality," in *Proceedings of the 18th International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2022, pp. 293–306.

[9] X. Liu, S. Wen, J. Zhao, T. Z. Qiu, and H. Zhang, "Edge-assisted multi-robot visual-inertial slam with efficient communication," *IEEE Transactions on Automation Science and Engineering*, 2025.

[10] X. Zhang, H. Zhu, Y. Duan, W. Zhang, L. Shangguan, Y. Zhang, J. Ji, and Y. Zhang, "Map++: Towards user-participatory visual slam systems with efficient map expansion and sharing," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2024, pp. 633–647.

[11] C. Hu, W. Bao, and D. Wang, "Iot communication sharing: Scenarios, algorithms and implementation," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1556–1564.

[12] F. Ahmad, H. Qiu, R. Eells, F. Bai, and R. Govindan, "Carmap: Fast 3d feature map updates for automobiles," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2020, pp. 1063–1081.

[13] S. Aboagye, M. A. Saeidi, H. Tabassum, Y. Tayyar, E. Hossain, H.-C. Yang, and M.-S. Alouini, "Multi-band wireless communication networks: Fundamentals, challenges, and resource allocation," *IEEE Transactions on Communications*, 2024.

[14] J. Xu, H. Cao, Z. Yang, L. Shangguan, J. Zhang, X. He, and Y. Liu, "Swarmmap: Scaling up real-time collaborative visual slam at the edge," in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. Renton, WA: USENIX Association, 2022.

[15] D. Li, Y. Zhao, J. Xu, S. Zhang, L. Shangguan, Q. Ma, X. Ding, and Z. Yang, "Reshaping edge-assisted visual slam by embracing on-chip intelligence," *IEEE Transactions on Mobile Computing(TMC)*, pp. 1–14, 2024.

[16] Z. Jiang and Y. Shan, "Cmd-slam: A fast low-bandwidth centralized multi-robot direct stereo slam," in *2024 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2024, pp. 920–926.

[17] M. Patel, M. Karrer, P. Bänninger, and M. Chli, "Covins-g: A generic back-end for collaborative visual-inertial slam," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2076–2082.

[18] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," *Artificial intelligence(AI)*, vol. 78, no. 1-2, pp. 87–119, 1995.

[19] S. Li, C. Xu, and M. Xie, "A robust o (n) solution to the perspective-n-point problem," *IEEE transactions on pattern analysis and machine intelligence(TPAMI)*, vol. 34, no. 7, pp. 1444–1450, 2012.

[20] H. Cao, J. Xu, D. Li, L. Shangguan, Y. Liu, and Z. Yang, "Edge assisted mobile semantic visual slam," *IEEE Transactions on Mobile Computing(TMC)*, vol. 22, no. 12, pp. 6985–6999, 2022.

[21] J. Cui, S. Shi, Y. He, J. Niu, and Z. Ouyang, "Vilam: Infrastructure-assisted 3d visual localization and mapping for autonomous driving," in *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2024, pp. 1831–1845.

[22] Z. Xu, L. Zhou, S. C.-K. Chau, W. Liang, Q. Xia, and P. Zhou, "Collaborate or separate? distributed service caching in mobile edge clouds," in *IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2020, pp. 2066–2075.

[23] P. Schmuck and M. Chli, "Ccm-slam: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams," *Journal of Field Robotics*, vol. 36, no. 4, pp. 763–781, 2019.

[24] X. Liu, J. Lei, A. Prabhu, Y. Tao, I. Spasojevic, P. Chaudhari, N. Atanasov, and V. Kumar, "Slideslam: Sparse, lightweight, decentralized metric-semantic slam for multi-robot navigation," *arXiv preprint arXiv:2406.17249*, 2024.

[25] C. Zuo, Z. Feng, and X. Xiao, "Ccmd-slam: communication-efficient centralized multi-robot dense slam with real-time point cloud maintenance," *IEEE Transactions on Instrumentation and Measurement*, 2024.

[26] V. Yugay, T. Gevers, and M. R. Oswald, "Magic-slam: Multi-agent gaussian globally consistent slam," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops(CVPR)*, 2025.

[27] Y. Park and S. Bae, "Keeping less is more: point sparsification for visual slam," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 7936–7943.

[28] X. Zhang and Y. Liu, "Efficient map sparsification based on 2d and 3d discretized grids," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 12 470–12 478.

[29] D. Li, J. Miao, X. Shi, Y. Tian, Q. Long, T. Cai, P. Guo, H. Yu, W. Yang, H. Yue *et al.*, "Rap-net: A region-wise and point-wise weighting network to extract robust features for indoor localization," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1331–1338.

[30] L. Carlone and S. Karaman, "Attention and anticipation in fast visual-inertial navigation," *IEEE Transactions on Robotics(TRO)*, vol. 35, no. 1, pp. 1–20, 2018.

[31] Y. Chen, H. Inaltekin, and M. Gorlatova, "Adaptslam: Edge-assisted adaptive slam with resource constraints via uncertainty minimization," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2023, pp. 1–10.

[32] L. Yu, Q. Wang, Y. Qiu, J. Wang, X. Zhang, and Z. Han, "Effective multi-agent communication under limited bandwidth," *IEEE Transactions on Mobile Computing(TMC)*, 2023.

[33] S. Shi, C. Hu, D. Wang, Y. Zhu, and Z. Han, "Federated hd map updating through overlapping coalition formation game," *IEEE Transactions on Mobile Computing*, vol. 23, no. 2, pp. 1641–1654, 2023.

[34] Z. Wang and J. Tao, "A fast implementation of adaptive histogram equalization," in *2006 8th international Conference on Signal Processing*, vol. 2. IEEE, 2006.

[35] T. D. Barfoot, *State estimation for robotics: A matrix lie group approach*. Cambridge University Press, 2022.

[36] J. Sola, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," *arXiv preprint arXiv:1812.01537*, 2018.

[37] J. G. Mangelson, M. Ghaffari, R. Vasudevan, and R. M. Eustice, "Characterizing the uncertainty of jointly distributed poses in the lie algebra," *IEEE Transactions on Robotics (TRO)*, vol. 36, no. 5, pp. 1371–1388, 2020.

[38] I. Han, D. Malioutov, and J. Shin, "Large-scale log-determinant computation through stochastic chebyshev expansions," in *International Conference on Machine Learning*. PMLR, 2015, pp. 908–917.

[39] T. T. Cai, T. Liang, and H. H. Zhou, "Law of log determinant of sample covariance matrix and optimal estimation of differential entropy for high-dimensional gaussian distributions," *Journal of Multivariate Analysis*, vol. 137, pp. 161–172, 2015.

[40] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions," *Mathematical programming*, vol. 14, pp. 265–294, 1978.

[41] H. Soo Park, Y. Wang, E. Nurvitadhi, J. C. Hoe, Y. Sheikh, and M. Chen, "3d point cloud reduction using mixed-integer quadratic programming," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops(CVPR)*, 2013, pp. 229–236.

[42] D. Van Opdenbosch, T. Aykut, N. Alt, and E. Steinbach, "Efficient map compression for collaborative visual slam," in *IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2018, pp. 992–1000.

[43] H. Li, R. Hartley, and J.-h. Kim, "A linear approach to motion estimation using generalized camera models," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. IEEE, 2008, pp. 1–8.

[44] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, "Pairwise consistent measurement set maximization for robust multi-robot map merging," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 2916–2923.

[45] O. S. R. Foundation, "Robot operating system (ros) melodic morenia," 2018, version: Melodic Morenia. [Online]. Available: http://wiki.ros.org/melodic

[46] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2024. [Online]. Available: https://www.gurobi.com

[47] R. A. Light, "Mosquitto: server and client implementation of the mqtt protocol," *Journal of Open Source Software*, vol. 2, no. 13, p. 265, 2017.

[48] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for rgb-d visual odometry, 3d reconstruction and slam," in *2014 IEEE international conference on Robotics and automation (ICRA)*. IEEE, 2014, pp. 1524–1531.

[49] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[50] D. Feng, Y. Qi, S. Zhong, Z. Chen, Y. Jiao, Q. Chen, T. Jiang, and H. Chen, "S3e: A large-scale multimodal dataset for collaborative slam," *arXiv preprint arXiv:2210.13723*, 2022.

[51] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[52] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)(IEEE Cat. No. 04CH37566)*, vol. 3. Ieee, 2004, pp. 2149–2154.

[53] O. Bailo, F. Rameau, K. Joo, J. Park, O. Bogdan, and I. S. Kweon, "Efficient adaptive non-maximal suppression algorithms for homogeneous spatial keypoint distribution," *Pattern Recognition Letters(PRL)*, vol. 106, pp. 53–60, 2018.

Rui Ge received his B.S. degree in Automation Engineering in 2021 and his M.S. degree in Control Science and Engineering in 2024, both from University of Electronic Science and Technology of China. He is currently pursuing his Ph.D. in Computer Science at Wuhan University. His research interests include mobile computing, wireless communication, and robotics.

Huanghuang Liang received his B.S. and M.S. degrees in Automation Engineering from Anhui University of Technology in 2016 and the University of Electronic Science and Technology of China in 2019. He gained his Ph.D. in Computer Science at Wuhan University. His research interests include edge learning, federated learning/analytics, and distributed computing.

Zheng Gong received his PhD in Computing from The Hong Kong Polytechnic University in 2025. He previously received his M.Eng. in Cyberspace Security and B.Eng. in Computer Science from Xidian University in 2021 and 2018, respectively. Currently, he serves as an Associate Researcher at the School of Cyber Security, Tianjin University. His research interests encompass mobile computing, wireless communication, and edge computing.

Chuang Hu received his BS and MS degrees from Wuhan University in 2013 and 2016. He received his Ph.D. degree from the Hong Kong Polytechnic University in 2019. He is a postdoctoral fellow at the State Key Laboratory of Internet of Things for Smart City (IOTSC) of the University of Macau. His research interests include edge learning, federated learning/analytics, and distributed computing.

Xiaobo Zhou (Senior Member, IEEE) obtained the B.S., M.S., and Ph.D degrees in Computer Science from Nanjing University in 1994, 1997, and 2000, respectively. Currently, he is a Distinguished Professor at the University of Macau, Macau SAR. His research focuses broadly on distributed systems and cloud computing. He served as Chair of the IEEE Technical Community in Distributed Processing for 2020–2023.

Dazhao Cheng (Senior Member, IEEE) received his B.S. and M.S. degrees in electrical engineering from the Hefei University of Technology in 2006 and the University of Science and Technology of China in 2009. He received his Ph.D. from the University of Colorado at Colorado Springs in 2016. He is currently a professor in the School of Computer Science at Wuhan University. His research interests include big data and cloud computing.