Mr. Virgil: Learning Multi-robot Visual-range Relative Localization

Si Wang[1], Zhehan Li[2], Jiadong Lu[2], Rong Xiong[1], Yanjun Cao[2], Yue Wang[1*]

***Abstract*— Ultra-wideband (UWB)-vision fusion localization has achieved extensive applications in the domain of multi-agent relative localization. The challenging matching problem between robots and visual detection renders existing methods highly dependent on identity-encoded hardware or delicate tuning algorithms. Overconfident yet erroneous matches may bring about irreversible damage to the localization system. To address this issue, we introduce Mr. Virgil, an end-to-end learning multi-robot visual-range relative localization framework, consisting of a graph neural network for data association between UWB rangings and visual detections, and a differentiable pose graph optimization (PGO) back-end. The graph-based front-end supplies robust matching results, accurate initial position predictions, and credible uncertainty estimates, which are subsequently integrated into the PGO back-end to elevate the accuracy of the final pose estimation. Additionally, a decentralized system is implemented for real-world applications. Experiments spanning varying robot numbers, simulation and real-world, occlusion and non-occlusion conditions showcase the stability and exactitude under various scenes compared to conventional methods. Our code is available at: https://github.com/HiOnes/Mr-Virgil.**

I. Introduction

Relative localization is fundamental to multi-robot applications involving drone swarms, rescue missions and exploration tasks. One straightforward method to obtain relative estimation is the transformation of robots' global states measured by external devices such as global positioning system (GPS) [1], [2], real-time kinematic positioning system (RTK) [3], motion capture system (MCS) [4] and multi-fixed-anchor UWB system [5]. Due to the strong reliance on external infrastructure and the workspace, such solutions cannot be directly applicable to unfamiliar scenes.

To enhance scalability and accuracy, local odometry and mutual observations are integrated into the multi-agent systems, among which visual and UWB system [6]–[9] serves as typical representatives. Visual images offer neighbor observations and can be leveraged for ego-motion estimation. UWB provides omnidirectional and occlusion-resistant ranging measurements. However, the noisy UWB signals [10]–[12] and the drifting nature of visual odometry (VO) impose high demands on the fusion manner of multimodal information. Moreover, the visual detection targets are anonymous, rendering the correspondences between visual detection and UWB range a challenging data association problem. In order to address this, CREPES [6] adopts active infrared (IR) LEDs and an IR fish-eye camera to achieve identity extraction, while IR communication makes time synchronization quite cumbersome. Apart from the ID-encoded hardware approaches [6], [13], [14], many researches adhere to detection-based paradigm [7]–[9], which generates raw matches from visual detection bounding boxes, followed by hand-crafted rule-based post-processing. In general, these methods assume the resultant matches are correct and hard, making the downstream optimization fragile and irrecoverable when matching errors occur. Given the complexity of drone swarm formations, the ambiguity of robots and detection targets remains a challenging issue in multi-agent systems.

To overcome the aforementioned challenges of matching aliasing, accurate uncertainty estimates should be provided alongside data association, enabling soft constraints for subsequent optimization. Additionally, matching uncertainty should be informed by the collective structure of the multi-robot formation, rather than being limited to pairwise similarity. Beyond that, in multi-robot systems, the number of robots and visual mutual observations frequently varies, raising demand for the flexibility of the matching architecture.

In this paper, we propose Mr. Virgil, an end-to-end multi-robot visual-range relative localization system. For the front-end network, to accommodate any number of drones and visual observations, we harness the graph neural network (GNN), which has exhibited remarkable performance in the field of image feature matching [15], [16]. The GNN and the differentiable Sinkhorn algorithm [17]–[19] are utilized to solve the data association problem, yielding matching and uncertainty estimations with a global perspective. To supervise covariance, a Maximum Likelihood (ML) loss is applied to the output of the front-end network. The covariance is also incorporated as weights in the PGO back-end, enabling end-to-end learning using gradients from localization errors. The differentiable PGO back-end is employed for joint optimization and its gradient back-propagates to facilitate learning of front-end network. To achieve real-time performance in real-world applications, we also implement a decentralized system based on Robot Operating System (ROS), LibTorch and Ceres Solver. Overall, our major contributions are as follows:

- We propose an end-to-end decentralized multi-robot relative localization system, comprising a learnable front-end for data association and a differentiable PGO back-end.
- We present a GNN-based match network for multi-robot data association, realizing precise matching and reasonable uncertainty estimation, with the ability to handle an arbitrary number of drones and detections.

This work was supported in part by the National Nature Science Foundation of China under Grant 62373322 and Research and Development Project of Zhejiang Province under Grant No. 2025C01205(SD2).

[1]Institute of Cyber-Systems and Control, Zhejiang University, China.
[2]Huzhou Institute of Zhejiang University, Huzhou, China.
[*]*Corresponding author: Yue Wang.* (E-mail: ywang24@zju.edu.cn)

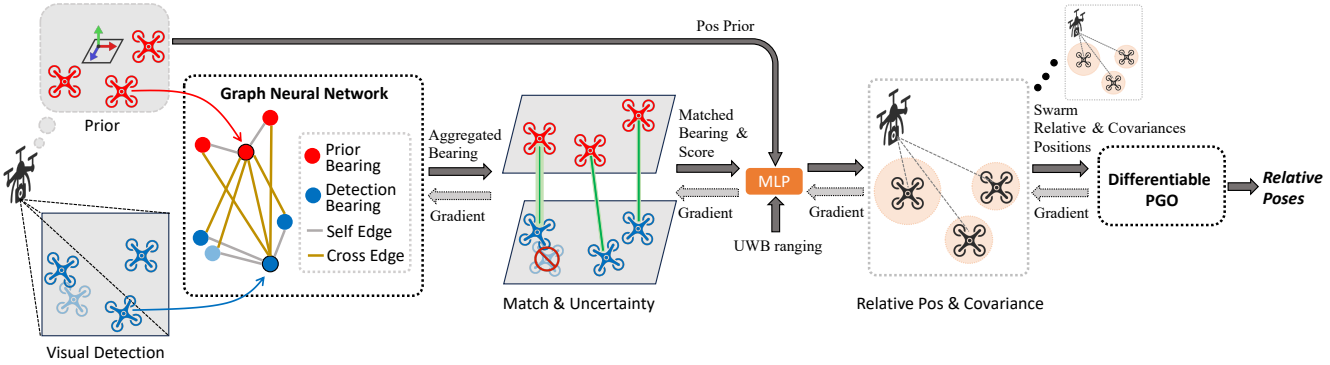


Fig. 1: The pipeline of our end-to-end multi-robot localization network. The GNN-based match net associates prior bearings and detection bearings, predicting relative positions with covariances (uncertainties of matches and positions are represented by light green lines and light pink circles). The differentiable PGO improves performance and propagates gradients back for joint error correction.

- The robustness and accuracy of our method have been verified in both simulated and real-world, occluded and non-occluded scenarios. Our network demonstrates impressive generalization ability in scenarios with limited training data, variable robot counts, and sim-to-real experiments.

II. Related Works

In this section, we discuss relative localization schemes that function without external aids like GPS or RTK, which offer superior adaptability to unfamiliar environments. Based on the sensor types, we classify the previous works into UWB-based methods and vision-based methods.

A. UWB-based Methods

A few works treat UWB as a stand-alone localization solution, owing to its cost-effectiveness and ability to offer omnidirectional ranging. For UWB systems that do not rely on external fixed anchors, each robot estimates the positions of the neighbors within its local frame, eliminating the need for a unified global coordinate system. Zhou [20] estimated the 3-DoF relative pose transformation between planar robots by leveraging inter-robot distance measurements and displacement estimation, requiring theoretical minimum UWB measurements. Fishberg [21] investigated the impact of multi-UWB tag antenna occlusion and interference on ranging errors. By applying occlusion-related weighted factors to nonlinear least squares, they achieved comparable accuracy to systems relying on continuous odometry exchange. Since only ranging information is shared, these methods incur a low communication overhead, while their stabilities are easily plagued by the noise-prone nature of UWB.

B. Vision-based Methods

The performance of the localization system can be improved by the fusion of multimodal information such as inertial data, visual detections [6]–[9], and odometry [22], among which the UWB-vision-based approach is representative. As the identity of the visual tracking target is unknown, a data association problem arises essentially. Such schemes can be divided into hardware ID-encoded-based methods and software matching-based methods depending on the ways to extract the visual identity.

ID encoding methods necessitate the arrangement of specially designed hardware devices, such as infrared LEDs [6], [14] and AprilTags [13]. Yan [14] equipped each drone with a distinct active infrared coded target and a monocular camera for detection, solving the relative transformation by PnP algorithm and Kalman filter. CREPES [6] employed programmed IR LED boards and IR cameras for encoding and decoding the identity respectively, maintaining effective relative localization in dark or partially occluded scenarios. By employing customized platforms, such methods diminished incorrect detections, but it is still encumbered by the short detection range and complex time synchronization of hardware.

For better scalability, some researchers have chosen visual detection and tracking algorithms, along with matching algorithms for data association. Xu [7] utilizes convolutional neural network (CNN) detectors and MOSSE trackers to generate bearing information and retrieve the range from the depth camera, yielding visual estimated positions. The matching process involves the comparison between the positions of the final estimates and the visual estimates, as well as a predefined threshold for outlier rejection. Based on this work, their subsequent studies Omni-swarm [8] applied the Hungarian algorithm to solve the multi-robot matching issue and introduced a sparse map for global consistency. However, the above approaches overlooked the uncertainty estimation problem with the potential of erroneous matches. Additionally, depth cameras are costly and vulnerable to changes in lighting conditions, whereas our avenue employs a bearing-only matching strategy.

III. Methodology

Our system overview is shown in Fig. 1, which mainly consists of a graph match net (front-end) and a differentiable PGO module (back-end). The GNN-based front-end combines information from priors, camera detections and UWB ranges, and the resulting 3-DoF position estimates and covariances are incorporated into our differentiable back-end, finally generating the 6-DoF state estimations.

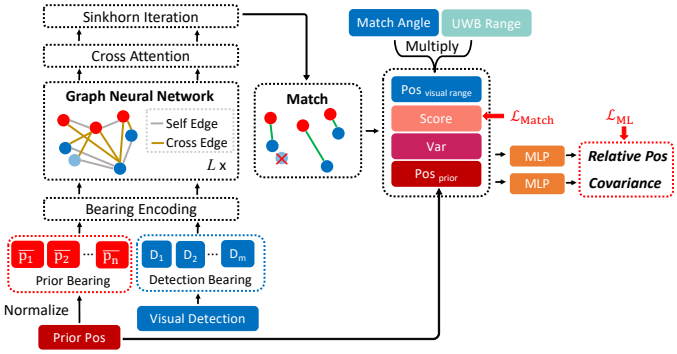


Fig. 2: The graph match net front-end architecture.

In this section, we first clarify our data association process based on GNN. Second, we describe three different constraints in PGO. Third, we introduce various loss functions for end-to-end training. Last, we illustrate the decentralized system for real-world implementation.

A. Graph Match Front-end

UWB rangings and priors of UAV are ID-aware, while visual detections are non-identified. Motivated by finding keypoints correspondences in the area of image feature matching, we solve the data association problem between priors and detections in a bearing-only manner. Our network comprises the attentional graph neural network for feature aggregation, the Sinkhorn iteration for partial assignment, and the multi-layer perceptron (MLP) for estimation of positions and uncertainties. The architecture of our graph match net is illustrated in Fig. 2.

Attentional graph neural network: A multi-layer graph attention neural network is utilized to encode a set of UAV bearing priors and a set of detection bearing outcomes, which are connected by self-edges and cross-edges, contributing to a deeper insight into the robot formation (intra-set) and the similarities among detection candidates (inter-set). Each bearing of priors and detections represents a graph node. We use a shared MLP layer to project the 3-DoF bearings $\mathbf{b}_i$ into high-dimensional space, forming the initial node embeddings ${}^{(0)}\mathbf{f}_i$ that guide the network to consider spatial information.

$$ {}^{(0)}\mathbf{f}_i = f_{encode}(\mathbf{b}_i) \tag{1} $$

The node embeddings are then aggregated through self-attention and cross-attention, realizing comprehensive message exchange between bearings of priors and detections. Residual connections are used both within the layer and between adjacent layers.

$$ \begin{aligned} {}^{(l)}\mathbf{f}_i^{self} &= {}^{(l)}\mathbf{f}_i + {}^{(l)}f_{self}([{}^{(l)}\mathbf{f}_i \,||\, {}^{(l)}\mathbf{m}_{\varepsilon_{self}}]) \\ {}^{(l+1)}\mathbf{f}_i &= {}^{(l)}\mathbf{f}_i^{self} + {}^{(l)}f_{cross}([{}^{(l)}\mathbf{f}_i^{self} \,||\, {}^{(l)}\mathbf{m}_{\varepsilon_{cross}}]) \end{aligned} \tag{2} $$

where $||$ denotes concatenation. ${}^{(l)}\mathbf{f}_i$ is the bearing embedding of layer l. The message aggregation along self edges ε_{self} and cross edges ε_{cross} are represented by ${}^{(l)}\mathbf{m}_{\varepsilon_{self}}$ and ${}^{(l)}\mathbf{m}_{\varepsilon_{cross}}$ respectively. ${}^{(l)}f_{self}$ and ${}^{(l)}f_{cross}$ are MLPs, where the weights differ across layers.

After message interaction through L GNN layers, the bearing embeddings of priors $\mathbf{f}_i^{\mathcal{P}}$ and detections $\mathbf{f}_j^{\mathcal{D}}$ are distinguishable and enriched with global information. $\mathcal{P}$ and $\mathcal{D}$ denote the set of UAV priors and detections.

Partial assignment: The score matrix $\mathbf{S} \in \mathbb{R}^{N\times M}$ can be obtained by the similarities of GNN-aggregated bearing embeddings. N and M refer to the number of drones and the maximum number of camera observations, respectively. Considering potential fake detection (the light blue drone detection in Fig. 1), M is larger than N. The pairwise score is computed by the dot-product:

$$ \mathbf{S}_{i,j} = < \mathbf{f}_i^{\mathcal{P}}, \mathbf{f}_j^{\mathcal{D}} > \tag{3} $$

In order for the network to learn to exclude mis-matching cases caused by occlusion, out-of-view and false detections, the score matrix $\mathbf{S}$ is further augmented to $\bar{\mathbf{S}} \in \mathbb{R}^{(N+1)\times(M+1)}$ by adding a dustbin row and a dustbin column for unmatched bearings. A trainable parameter is applied to represent the score of the bin row and column.

The optimal assignment can be solved by the Sinkhorn algorithm, which is differentiable, enabling end-to-end training. After several iterations, the augmented score matrix $\bar{\mathbf{S}}$ is reallocated subject to the constraint that the sums of rows and columns are equal to specific constant values.

The candidate match is derived according to the maximum score in each row and column. A match is considered valid only when the score exceeds a predefined threshold and both bearings of UAV priors and visual detections mutually consent to the match.

Position prediction: For each successfully matched UAV prior $i \in \mathcal{P}$, we construct a concatenated feature $\mathbf{feat}_i$ for estimation of positions and covariances.

$$ \mathbf{feat}_i = [{}^{vr}\mathbf{Pos}_i \,||\, {}^{p}\mathbf{Pos}_i \,||\, \mathbf{S}_i \,||\, \mathbf{Var}_i] \tag{4} $$

where ${}^{vr}\mathbf{Pos}_i$ is the raw visual ranging position, calculated by simply scaling the detection bearing $\mathbf{b}_i \in \mathbb{R}^3$ with the UWB ranging $d_i \in \mathbb{R}$. ${}^{p}\mathbf{Pos}_i$ denotes prior position. $\mathbf{S}_i$ comes from the optimal matching score of row i from the score matrix. $\mathbf{Var}_i$ is determined by the matching probability of the detection orientation.

With the input of raw estimate, pos prior, matching score and variance, two MLPs are employed for positions and covariances prediction. Positions of unmatched drones remain as the prior, with a large covariance assigned.

$$ \hat{\mathbf{t}}_i = f_{pos}(\mathbf{feat}_i),\ \hat{\mathbf{\Sigma}}_i = f_{cov}(\mathbf{feat}_i) \tag{5} $$

B. Differentiable PGO Back-end

To obtain high-precision 6-DoF poses, including the unobserved drone states, we tightly fuse sensor inputs and mutual state estimations between drones in the differentiable PGO module. Define k as the reference robot, with the relative poses of other robots to be optimized in the k coordinate system defined as:

$$ \boldsymbol{\chi} = [\hat{\mathbf{P}}_1^k, \hat{\mathbf{P}}_2^k, ..., \hat{\mathbf{P}}_N^k] \tag{6} $$

where $\hat{\mathbf{P}}_i^k$ is the same as the transformation matrix $\begin{bmatrix} \hat{\mathbf{R}}_i^k & \hat{\mathbf{t}}_i^k \\ 0 & \mathbf{1} \end{bmatrix}$, $\hat{\mathbf{R}}_i^k$ refer to the rotation matrix and $\hat{\mathbf{t}}_i^k$ refer to the translational vector. To find the optimal estimation $\boldsymbol{\chi}^*$, we attempt to minimize the combined residuals:

$$\boldsymbol{\chi}^* = argmin(\sum(C_M, C_P, C_R)) \tag{7}$$

where C_M, C_P, C_R are constraints of mutual state estimations, pose priors and range measurements respectively.

Mutual state constraint: The mutual observation between drone i and j forms a constraint edge. Since the front-end network only predicts the position, we omit the constraints on the rotation part and derive the error function in the following form:

$$\begin{aligned} \mathbf{e}_M &= \mathbf{t}_j^i - (\hat{\mathbf{R}}_i^k)^T(\hat{\mathbf{t}}_j^k - \hat{\mathbf{t}}_i^k) \\ C_M &= \mathbf{e}_M^T \hat{\boldsymbol{\Sigma}}_M^{-1} \mathbf{e}_M \end{aligned} \tag{8}$$

where $\hat{\boldsymbol{\Sigma}}_M^{-1}$ represents the information matrix, which is the inverse of the covariance matrix, where the diagonal elements are formed by the uncertainties predicted by the preceding network.

Pose prior constraint: When the number of mutual observations decreases, the optimization problem may become ill-conditioned, due to which we incorporate prior pose constraints to avoid degradation.

$$\begin{aligned} \mathbf{e}_P &= \mathbf{P}_i^k - \hat{\mathbf{P}}_i^k \\ C_P &= \mathbf{e}_P^T \hat{\boldsymbol{\Sigma}}_P^{-1} \mathbf{e}_P \end{aligned} \tag{9}$$

The diagonal entries of the covariance matrix $\hat{\boldsymbol{\Sigma}}_P$ are likewise predicted by the network.

UWB ranging constraint: The UWB measurements establish pairwise connections between drones in the cluster.

$$\begin{aligned} \mathbf{e}_R &= d_{ij} - \parallel \hat{\mathbf{t}}_i^k - \hat{\mathbf{t}}_j^k \parallel_2 \\ C_R &= \mathbf{e}_R^T \hat{\boldsymbol{\Sigma}}_R^{-1} \mathbf{e}_R \end{aligned} \tag{10}$$

where d_{ij} is the distance measurement between drone i and j. The covariance of UWB ranging $\hat{\boldsymbol{\Sigma}}_R$ is predefined.

We use second-order Levenberg-Marquardt (LM) algorithm and Cholmod sparse solver in Theseus [23] to solve the nonlinear optimization problem. The gradient of each iteration is stored and backpropagated to facilitate the training of the front-end network.

C. Loss Functions

To guide the network to produce robust matches, reliable covariance distributions and accurate state estimations, three different loss functions are applied in our training, balanced by factors λ_1, λ_2:

$$\mathcal{L} = \mathcal{L}_{Match} + \lambda_1 \mathcal{L}_{ML} + \lambda_2 \mathcal{L}_{Pose} \tag{11}$$

Match loss: For the matching item, we supervise the augmented score matrix $\bar{\mathbf{S}}$ after Sinkhorn iterations:

$$\mathcal{L}_{Match} = -\sum_{(i,j)\in\pi} \bar{\mathbf{S}}_{i,j} - \sum_{i\in\mu} \bar{\mathbf{S}}_{i,M} \tag{12}$$

where π is the set of matching bearings, μ denotes the set of unmatched UAV priors, both of which come from the ground truth labels. The column M in $\bar{\mathbf{S}}_{i,M}$ represents the dustbin column, storing the cases where camera observations are lost. The first item encourages the net to amplify the scores of correct matches, while the second term drives the network to exclude incorrect ones.

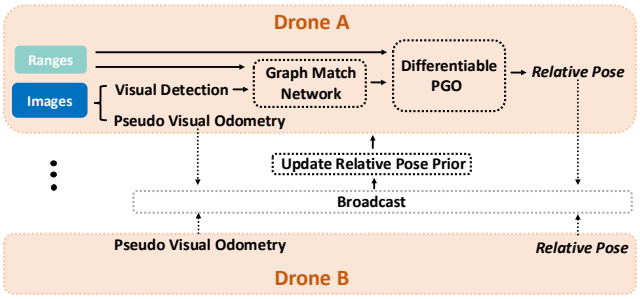


Fig. 3: Decentralized system diagram.

Maximum likelihood loss: According to the form of a multivariate Gaussian distribution, we define a negative log-likelihood covariance loss as follows:

$$\begin{aligned} \mathcal{L}_{ML} = &\frac{1}{(N+1)*N} \sum_{(i,j)\in\mathcal{S}, j\neq i} (\lambda_{det} log(det(\hat{\boldsymbol{\Sigma}}_j^i)) \\ &+ (\mathbf{t}_j^i - \hat{\mathbf{t}}_j^i)^T \hat{\boldsymbol{\Sigma}}_j^{i^{-1}} (\mathbf{t}_j^i - \hat{\mathbf{t}}_j^i)) \end{aligned} \tag{13}$$

where $\mathcal{S}$ is the set of UAV swarm, $\hat{\boldsymbol{\Sigma}}_j^i \in \mathbb{R}^{3\times3}$ is the predicted covariance matrix, $\hat{\mathbf{t}}_j^i$ is the 3-DoF position estimate of drone j with respect to drone i, while $\mathbf{t}_j^i$ is the ground truth.

Pose loss: Apart from the aforementioned two losses directly applied to the front-end net outputs, we also define the Mean Square Error (MSE) pose loss for final relative localization after graph optimization:

$$\begin{aligned} \mathcal{L}_{Pose} = &\frac{1}{(N+1)*N} \sum_{(i,j)\in\mathcal{S}, j\neq i} (\parallel \mathbf{t}_j^i - \hat{\mathbf{t}}_j^i \parallel^2 \\ &+ \lambda_q \parallel \mathbf{q}_j^i - \hat{\mathbf{q}}_j^i \parallel^2) \end{aligned} \tag{14}$$

where $\hat{\mathbf{t}}_j^i$ and $\hat{\mathbf{q}}_j^i$ denote the estimation of translation and quaternion part, $\mathbf{t}_j^i$ and $\mathbf{q}_j^i$ are their corresponding ground truths.

D. Decentralized System

For the purpose of high-performance deployment in real environments, we have realized a decentralized system (Fig. 3). The system is built upon the ROS communication framework, the graph match front-end network is implemented on LibTorch, and the optimization back-end is solved by Ceres Solver.

In the simulation experiments, the visual detection direction is computed based on the orientation and positional relationship between two drones within the field of view. In the physical experiments, we use the infrared hardware module of CREPES for recognition (without ID extraction). As drones usually fly at high speeds, we introduce pseudo visual odometry (PVO) to aid in updating the relative pose prior between consecutive frames, which is derived from the ground truth with considerable noise. The relative pose priors

are updated by the PVO and the mutual state estimation among other neighboring robots. Only the PVO and the optimized relative poses in the local frames are shared among the drones, leading to a low communication load.

IV. Experiment

We carry out experiments on both simulation datasets and self-collected real-world datasets to verify the accuracy and robustness of our proposed method, covering both line-of-sight (LOS) and non-line-of-sight (NLOS) scenarios. The details of our experiment datasets are listed in TABLE. I.

A. Experimental Settings and Datasets

A laptop equipped with a 13th Gen Intel Core i9-13900HX CPU and Nvidia RTX 4060 GPU is used to train and validate our neural network. We employ a 4-layer GNN network to aggregate the bearing features. The number of Sinkhorn iterations is set to 100. We use the Adam optimizer for training, with a learning rate of 1e-4 and a weight decay coefficient of 5e-4. The front-end network undergoes pretraining to generate accurate estimates of positions and uncertainties, which takes less than 50 epochs.

At train time, we add noise with a standard deviation of 0.1m along each axis to the ground truth position from 0.1 seconds earlier, indicating the pos prior. At test time, the prior comes from the previous prediction plus the relative odometry increment. In situations where all camera detections or UWB observations are unavailable, these instances will be omitted during training, while for testing, the position prior will be simply updated by odometry and fed into the PGO, bypassing the graph match network.

Simulation scenes: We conduct simulation experiments in a random forest environment filled with diverse obstacles like trees. A group of drones in a circular formation traverses the $70m \times 30m \times 3m$ forest. The simulated scenes with varying numbers of robots are presented in Fig. 4. Each UAV has a 180-degree field of view (FOV), and the visibility is not only affected by occlusion but also influenced by both the camera orientation of the observing drones and the direction of the IR LEDs on the drones being observed. To validate the robustness of our proposed matching method, we also randomly generate erroneous visual detections with a probability exceeding 40%.

Real-world scenes: In the physical experiments, we adopted the same hardware as CEREPS for data acquisition, while excluding its ID extraction and IMU modules. The camera features a fisheye lens with a 185-degree FOV, and the UWB module is from the NoopLoop DW1000 series. As the experiments are carried out indoors, we employ MCS to obtain the ground truth. The effectiveness of our proposal is demonstrated in both occlusion and non-occlusion scenarios.

TABLE I: Experiment Datasets.

Dataset	Drone Num	Detection Num	Occlusion	Traj Len
Sim-Forest	4/8/12/16	0∼8/12/16/20	✓	52.23m
Real-LOS	5	0∼8	✗	20.64m
Real-NLOS	5	0∼8	✓	20.63m

B. Baselines

We choose two baselines for comparison with our approach.

PVO: The odometry of each robot is derived by adding noise perturbations to the ground truth. The noise consists of a translation disturbance with a standard deviation of 0.1m and a rotation disturbance with a standard deviation of 1.0 degree, applied every 0.1s of odometry. Note that the noise magnitude is consistent with that of the other PVO-aided methods. The relative pose estimates calculated by the odometry are further passed into the PGO for optimization, with no inter-robot observations involved.

Simple Match: The data association is performed according to the closest direction between the bearings of UAV priors and visual detections. A tunable threshold is used to discard matches with large directional differences. For successfully matched drones, their relative positions are obtained by multiplying the camera directions with the UWB rangings. The covariances have a negative correlation with the cosine similarities of the matching bearings, contributing to more robust optimization compared to hard matching. This method also leverages PVO to update the state priors and enhances the overall accuracy through graph optimization.

C. Multi-Robot Localization Accuracy

For ease of visualization, estimated trajectories of other drones are transformed by adding the relative poses on the ground truth of drone 0. The estimations under ideal conditions are shown in Fig. 5(a), achieving the RMSE error of 3.9cm. In actual conditions, odometry suffers from accumulated drift, and the estimations with considerable noise added to the PVO are presented in Fig. 5(b) and Fig. 6. We evaluate 3D Relative Positioning Error (RPE) by RMSE in TABLE. II and the RPE of dataset *Real-NLOS* with respect to each robot estimated by our approach and Simple Match are depicted by heat map in Fig. 7. Notably, all error metrics are calculated in the local frame of each robot, while trajectories of in the world coordinate frame are used only for visualization.

Resistance to noise interference: The noise we introduce into the odometry is substantially greater than the errors of most state-of-the-art visual odometry methods. As shown in TABLE. II, PVO quickly diverges in all three scenarios, leading to large errors under the influence of noise. With the same noise level, the predicted trajectories of our approach exhibit some fluctuation when inter-observations decrease, but it quickly stabilizes once visibility is regained, maintaining valid and robust localization throughout the whole flight.

TABLE II: RPE evaluation under simulated and real-world datasets. Numbers in parentheses following the scene name denote the quantities of robots. Values in bold are the best.

Method	Position RMSE (m)		
	Sim-Forset(16)	*Real-LOS*(5)	*Real-NLOS*(5)
PVO	6.067	2.243	1.445
Simple Match	0.198	0.108	0.498
Ours	**0.144**	**0.090**	**0.129**

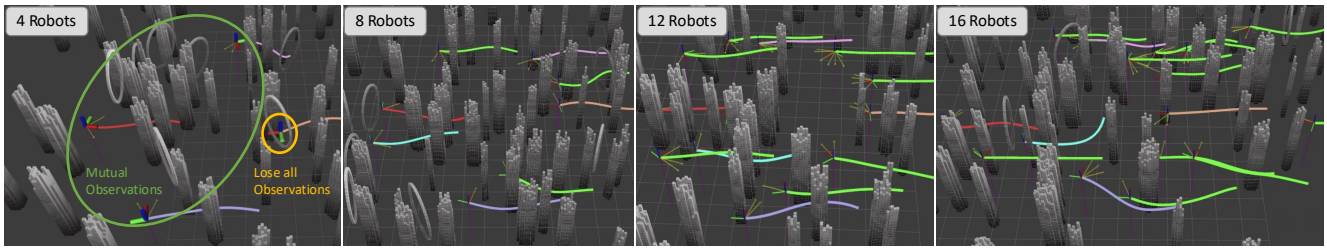
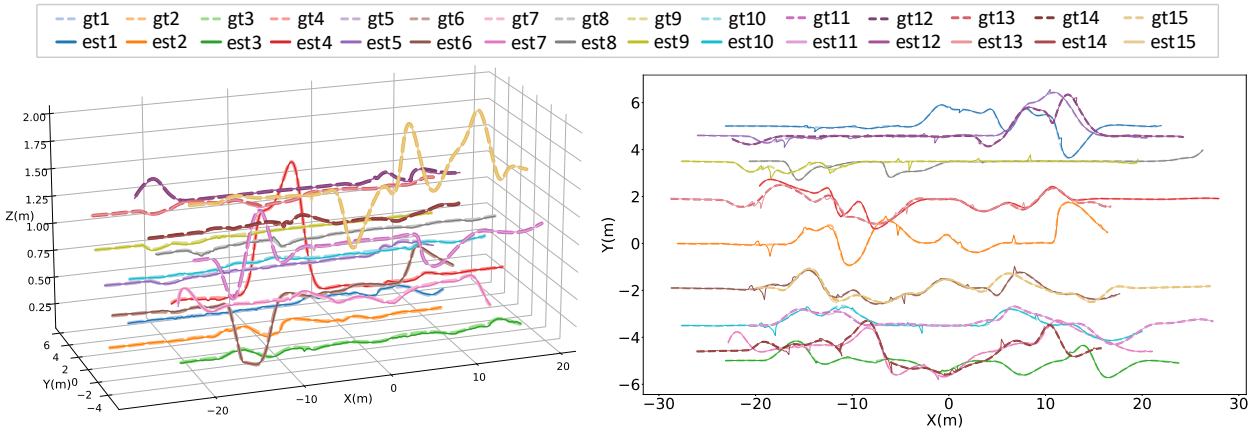


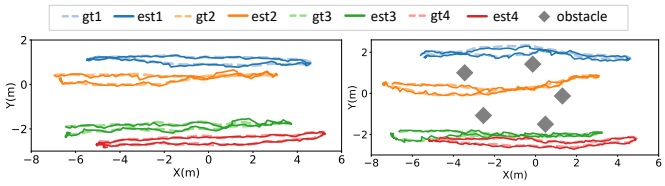
Fig. 4: The simulated random forest environment with varying robot counts. In the four-robot scenario, the three robots with mutual observations are circled by green ellipses, while the robot that lost all observations due to occlusion is highlighted by a yellow ellipse. As the number of robots increases, occlusions are more likely to occur.



(a) Estimation under ideal circumstances (PVO with no noise).

(b) Estimation under noisy circumstances (PVO with big noise).

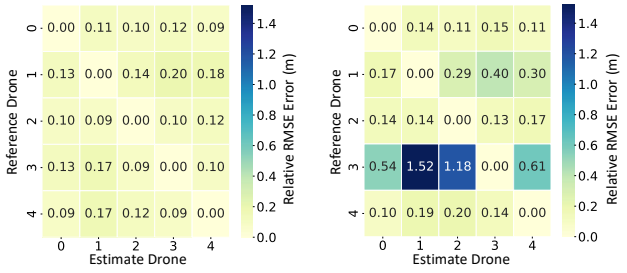
Fig. 5: The estimated trajectories of other 15 drones on simulated forest environment.



(a) *Real-LOS* scene.

(b) *Real-NLOS* scene.

Fig. 6: The estimated trajectories of other 4 drones under noisy circumstances (PVO with big noise) in real-world environment.



(a) RPE of ours.

(b) RPE of Simple Match.

Fig. 7: The RPE heat map w.r.t each drone on *Real-NLOS*.

Occlusion resistance: As indicated in TABLE. II, the Simple Match based on the closest bearing matching and the fixed threshold filter achieves comparable performance to our proposed method in non-occlusion environments (*Real-LOS*). However, a predefined threshold necessitates a trade-off between precision and recall rate. A lenient threshold parameter may result in more false matches, while a strict threshold reduces the matching recall rate. In occlusion scenarios (*Sim-Forest* and *Real-NLOS*), the Simple Match has fewer valid matches, relying solely on the noisy odometry to continuously update the state priors, causing significant performance degradation. In contrast, our graph match network focuses on the overall similarity between the set of UAV priors and visual detections in a global view, producing more successful matches and exhibiting stronger resistance to occlusion.

D. Ablation Study

In this section, we will examine the strengths of front-end learning graph networks compared to Simple Match, as well as the enhancement of accuracy achieved through back-end graph optimization.

Graph match front-end: We evaluate the graph matching network in comparison with the Simple Match under different threshold settings, measuring the matching precisions, recall rates, and F1 scores across multiple datasets. All metrics are presented in Table. III. Simple Match@0.9 indicates that only matches with a cosine similarity greater than 0.9 between the UAV priors and the detection bearings are considered valid. Likewise, Simple Match@0.99 stands for a stricter threshold parameter. The F1 score takes both the matching precision and recall rate into account, and our proposed method achieves the highest F1 score across all

TABLE III: Matching results on simulated and real-world datasets. Numbers in parentheses following the scene name denote the quantities of robots. F1 scores in bold are the best.

Method	Datasets											
	Sim-Forset (8)			*Sim-Forset* (16)			*Real-LOS* (5)			*Real-NLOS* (5)		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Simple Match@0.9	76.49%	99.45%	0.865	58.96%	97.54%	0.735	88.74%	98.85%	0.935	88.59%	99.78%	0.939
Simple Match@0.99	95.11%	98.50%	0.968	87.28%	95.94%	0.914	97.32%	92.41%	0.948	99.11%	96.14%	0.976
Ours	97.87%	98.58%	**0.982**	95.92%	91.14%	**0.935**	96.15%	98.33%	**0.972**	97.47%	99.51%	**0.985**

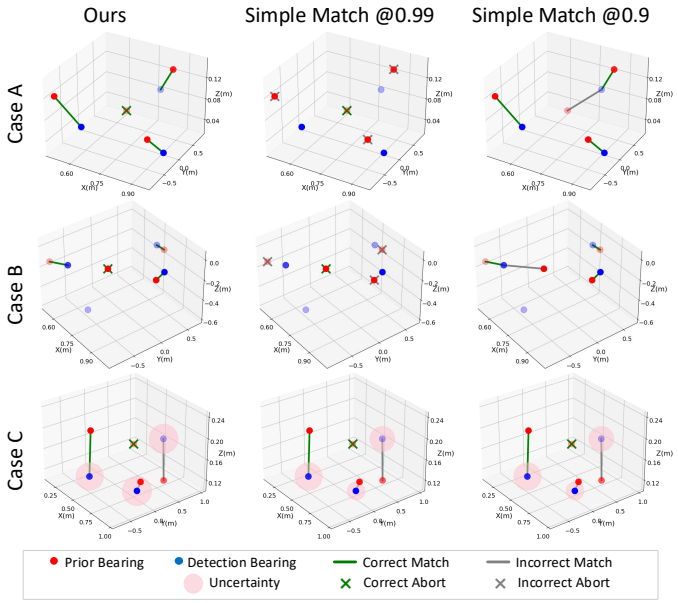


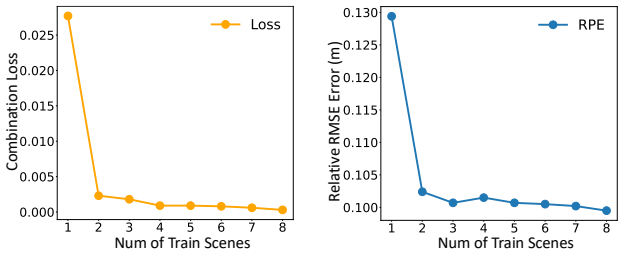
Fig. 8: Data association cases. We perform a comparison between our graph matching network and Simple Match with distinct filtering thresholds. In challenging Case C, the uncertainty of the matches is represented by pink spheres, where a larger radius indicates a higher uncertainty.

datasets, which highlights that our model strikes a good balance between incorporating anonymous observations and excluding incorrect matches. A few representative cases are displayed in Fig. 8. In Case A, there are four drones to be matched and three camera observations. Our method successfully matches all observed drones and excludes the one that is not observed, while Simple Match@0.99 rejects all matches and Simple Match@0.9 improperly associates two different drones with the same detection bearing based on the closest direction matching. A similar situation also occurs in Case B, where there are three valid camera observations and one erroneous camera detection. In the more complex and challenging Case C, all three methods encounter two incorrect matches, while our approach applies higher uncertainty to these mismatches, reducing their detrimental effect on the subsequent optimization process.

PGO back-end: To quantify the precision improvement with the incorporation of PGO, we conduct ablation experiments in a centralized inference strategy, ensuring identical communication conditions. The PVO is perturbed by noise and camera observations are interfered with random fake detections. The results listed in Table. IV demonstrate that the incorporation of PGO efficiently mitigates the localization

TABLE IV: Ablation results of PGO. The errors are evaluated by RPE RMSE.

Module			Scene					
Simple	Learned	PGO	*Sim-Forest*(16)		*Real-LOS*(5)		*Real-NLOS*(5)	
✓	✗	✗	1.572	↓46.8%	0.293	↓42.0%	0.453	↓55.8%
✓	✗	✓	0.836		0.170		0.200	
✗	✓	✗	1.280	↓89.2%	0.217	↓32.9%	0.247	↓35.2%
✗	✓	✓	0.138		0.126		0.160	



(a) Decrease of combination loss. (b) Decrease of RPE.

Fig. 9: The decrease of combination loss and RPE w.r.t number of training scenes.

error of either Simple Match or our learned front-end by over 32% across all scenarios.

E. Network Generalization Ability Analysis

In this section, we will investigate the generalization of the neural network across varying training set sizes, different numbers of robots, and simulation-to-real model transfer experiments.

Training scene numbers: We use varying numbers of scenes for training, assessing its inference performance in real-world occlusion environments. The combined loss and RPE error are presented in Fig. 9. The model reaches performance comparable to the optimal model after being trained on only two scenes.

Robot numbers: We train and test in simulated scenes with different numbers of robots, validating the model's generalization performance across varying robot nodes. To eliminate the differences in training set size caused by varying numbers of robots, we scale the training size by corresponding multiples. For example, the model for 16 robots is trained with 4 scenes, while the model for 8 robots is trained with 8 scenes. As illustrated by the error box plot in Fig. 10, networks trained with a larger number of robots achieve higher accuracy on all datasets. An increased number of robots enables the GNN to learn more intricate topological structures, offer more reasonable uncertainty estimates, and generalize to scenes with different robot quantities.

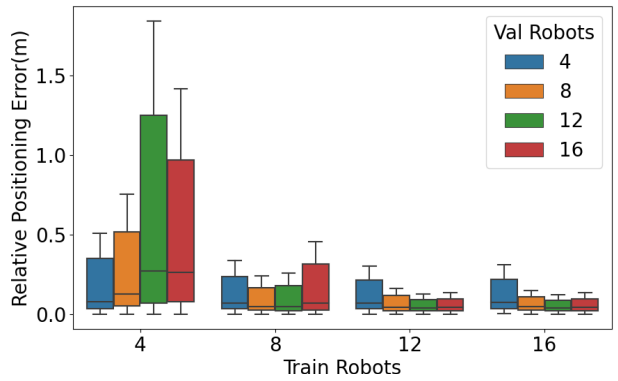


Fig. 10: RPE box plot of varying robot numbers experiment. The whisker length in the box plot is defined as 0.5 times the interquartile range (IQR).

TABLE V: The accuracy of simulation model and real model on real-world scenarios.

Model	Training Setting		Position RMSE (m)	
	Scene	Num of Robot / Cam	*Real-LOS*	*Real-NLOS*
Sim	Sim-Forest	16 / 20	0.098	0.136
Real	Real-world	5 / 8	0.090	0.129

Sim-to-real: To evaluate the network's adaptability from simulation models to real-world data, the simulation model trained on *Sim-Forest* sequences is validated in real-world occlusion and non-occlusion environments. The accuracy gap between the simulation model and the real model is less than 1cm, as shown in Table. V.

V. Conclusions

In this work, we propose an end-to-end learning visual-range framework in multi-agent relative localization system. The learnable front-end solves the data association problem through multiplex attentional graph neural network, facilitating reliable matching and uncertainty prediction. The differentiable PGO back-end collects mutual estimations and boost the overall precision.

In future work, we will consider replacing PVO with a real visual odometry system and integrating multi-frame observations into the network and optimization framework, which will further enhance the system's robustness and adaptability.

References

[1] A. Jaimes, S. Kota, and J. Gomez, "An approach to surveillance an area using swarm of fixed wing and quad-rotor unmanned aerial vehicles uav (s)," in *2008 IEEE International Conference on System of Systems Engineering*. IEEE, 2008, pp. 1–6.

[2] Y. Qi, Y. Zhong, and Z. Shi, "Cooperative 3-d relative localization for uav swarm by fusing uwb with imu and gps," in *Journal of Physics: Conference Series*, vol. 1642, no. 1. IOP Publishing, 2020, p. 012028.

[3] S. Moon, Y. Choi, D. Kim, M. Seung, and H. Gong, "Outdoor swarm flight system based on rtk-gps," *Journal of KIISE*, vol. 43, no. 12, pp. 1315–1324, 2016.

[4] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3299–3304.

[5] A. Ledergerber, M. Hamer, and R. D'Andrea, "A robot self-localization system using one-way ultra-wideband communication," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3131–3137.

[6] Z. Xun, J. Huang, Z. Li, Z. Ying, Y. Wang, C. Xu, F. Gao, and Y. Cao, "Crepes: Cooperative relative pose estimation system," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 5274–5281.

[7] H. Xu, L. Wang, Y. Zhang, K. Qiu, and S. Shen, "Decentralized visual-inertial-uwb fusion for relative state estimation of aerial swarm," in *2020 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 8776–8782.

[8] H. Xu, Y. Zhang, B. Zhou, L. Wang, X. Yao, G. Meng, and S. Shen, "Omni-swarm: A decentralized omnidirectional visual–inertial–uwb state estimation system for aerial swarms," *Ieee transactions on robotics*, vol. 38, no. 6, pp. 3374–3394, 2022.

[9] H. Xu, P. Liu, X. Chen, and S. Shen, "D^2 SLAM: Decentralized and distributed collaborative visual-inertial slam system for aerial swarm," *IEEE Transactions on Robotics*, 2024.

[10] K. Li, Z. Cao, and U. D. Hanebeck, "Continuous-time ultra-wideband-inertial fusion," *IEEE Robotics and Automation Letters*, vol. 8, no. 7, pp. 4338–4345, 2023.

[11] W. Zhao, A. Goudar, and A. P. Schoellig, "Finding the right place: Sensor placement for uwb time difference of arrival localization in cluttered indoor environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6075–6082, 2022.

[12] W. Zhao, A. Goudar, X. Qiao, and A. P. Schoellig, "Util: An ultra-wideband time-difference-of-arrival indoor localization dataset," *The International Journal of Robotics Research*, p. 02783649241230640, 2022.

[13] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 3400–3407.

[14] X. Yan, H. Deng, and Q. Quan, "Active infrared coded target design and pose estimation for multiple objects," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6885–6890.

[15] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[16] B. Roessle and M. Nießner, "End2end multi-view feature matching with differentiable pose optimization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 477–487.

[17] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in neural information processing systems*, vol. 26, 2013.

[18] G. Peyré, M. Cuturi, *et al.*, "Computational optimal transport: With applications to data science," *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[19] R. Sinkhorn and P. Knopp, "Concerning nonnegative matrices and doubly stochastic matrices," *Pacific Journal of Mathematics*, vol. 21, no. 2, pp. 343–348, 1967.

[20] X. S. Zhou and S. I. Roumeliotis, "Robot-to-robot relative pose estimation from range measurements," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1379–1393, 2008.

[21] A. Fishberg and J. P. How, "Multi-agent relative pose estimation with uwb and constrained communications," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 778–785.

[22] Z. Cao, R. Liu, C. Yuen, A. Athukorala, B. K. K. Ng, M. Mathanraj, and U.-X. Tan, "Relative localization of mobile robots with multiple ultra-wideband ranging measurements," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5857–5863.

[23] L. Pineda, T. Fan, M. Monge, S. Venkataraman, P. Sodhi, R. T. Chen, J. Ortiz, D. DeTone, A. Wang, S. Anderson, *et al.*, "Theseus: A library for differentiable nonlinear optimization," *Advances in Neural Information Processing Systems*, vol. 35, pp. 3801–3818, 2022.