Uncertainty-Aware GNN for Collaborative Robot Mapping Towards 6G-Enabled Smart Warehouses

Irfan Fachrudin Priyanta[1], Julia Freytag[2], Tobias Körner[3], M. Asfandyar Khan[1], Jérôme Rutinowski[1],
Moritz Roidl[1], Ilona Rolfes[3], Alice Kirchheim[1,2]
[1]Chair of Material Handling and Warehousing, TU Dortmund University, 44227 Dortmund, Germany
[2]Fraunhofer Institute for Material Flow and Logistics, 44227 Dortmund, Germany
[3]Institute of Microwave Systems, Ruhr University Bochum, 44801 Bochum, Germany
Email: {IrfanFachrudin.Priyanta, Asfandyar.Khan, Jerome.Rutinowski, Moritz.Roidl, Alice.Kirchheim}@tu-dortmund.de,
{Julia.Freytag, Alice.Kirchheim}@iml.fraunhofer.de, {t.koerner, Ilona.Rolfes}@rub.de

***Abstract*—Smart warehouses face rapid layout reconfigurations and frequent process adaptations, especially when using a Cyber-Physical Production Systems (CPPS) setup. These dynamic conditions introduce environmental uncertainty, making real-time navigation and obstacle avoidance a significant challenge for robot fleets. Limited sensing range and frequent occlusions further hinder local robot perception. Classic methods rely on centralized planning or vision-based systems, which struggle in low-visibility and cluttered environments. To address these gaps, we propose a graph-based collaborative perception network for the RoboFUSE (Framework for Unified Sensing and Exploration) system. Each robot operates onboard RoboFUSE with a dual-purpose waveform for sensing and communication (S&C) that emulates 6G Integrated Sensing and Communication (ISAC). This setup supports real-time data sharing across the fleet. On top of this platform, we develop RoboFUSE-Graph Neural Network (GNN), an uncertainty-aware GNN that fuses multi-robot radar data into a global spatial-semantic map. The model captures spatial relations and temporal dependencies using sliding window graphs. Experiments reveal an F1 score of 0.91 with a five-sliding window. The proposed approach enhances situational awareness, enables safe, scalable navigation, and represents a stride towards 6G-enabled smart warehouses.**

***Index Terms*—Smart Warehouses, Robotics, 6G ISAC, GNN, Collaborative Perception**

I. Introduction

The rise of smart warehouses has reshaped traditional logistics operations. One key innovation is the integration of CPPS, in which robot fleets perform material handling and inventory operations. As shown in Fig. 1, CPPS allows rapid layout reconfigurations of mobile assembly workstations and continuous process adaptations to meet varying production demands [1]. Such conditions lead to spatial and temporal uncertainty, which pose substantial challenges for real-time robot mapping, navigation, and obstacle avoidance [2].

These conditions demand precise and adaptive coordination across robot fleets. While recent multi-robot systems enable real-time mapping, many still rely on predefined or locally-built maps, limiting collaborative response to dynamic changes [3], [4]. A single robot has a limited sensor range and isolated perception, reducing the obstacles' awareness outside its surroundings. In addition, vision-based systems degrade in cluttered, low-light conditions, reducing robots' ability to react safely. [5]. Lack of global awareness leads to blind spots, inefficient path planning, and increased collision risk. When shared maps are available, they suffer latency or communication bottlenecks in large-scale deployments [6]. This restricts timely data sharing and impairs coordinated decision-making across the fleet. These limitations highlight the need for a Collaborative Perception (CP) system that fuses information across the fleet in real-time [5], [7].

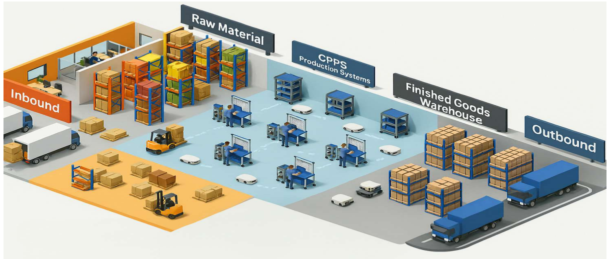


Fig. 1: Smart warehouse with CPPS setup, inspired by [1].

GNNs emerge as promising solutions for CP in multi-robot systems. They model structured interactions among robots and their environment, enabling decentralized data fusion. Recent works [3], [4], [7] demonstrate promising advances. However, these methods assume static environments, lack uncertainty modeling, or depend on Wi-Fi communication, which hinders real-time performance in dynamic smart warehouses. Emerging 6G technologies address these limitations by combining wireless transmission with ISAC, enabling high-resolution and low-latency environmental data sharing across fleets [8].

To explore this potential, our previous work introduced RoboFUSE [9], a platform emulating 6G ISAC using dual-waveform of mmWave radar sensing and onboard sub-6 GHz communication. Building upon RoboFUSE, we propose an uncertainty-aware GNN for decentralized collaborative mapping. Our GNN model fuses radar data from multiple robots, allowing fleet-wide shared situational awareness with low computational overhead. The key contributions are:

- Creating **custom datasets and preprocessing pipeline** for graph-based radar data fusion of robot fleets.
- Development of a **RoboFUSE GNN model** for decentralized multi-robot perception under uncertainty.
- **Validation** of the model in different **CPPS layouts**.

This paper is organized as follows. Related works are covered

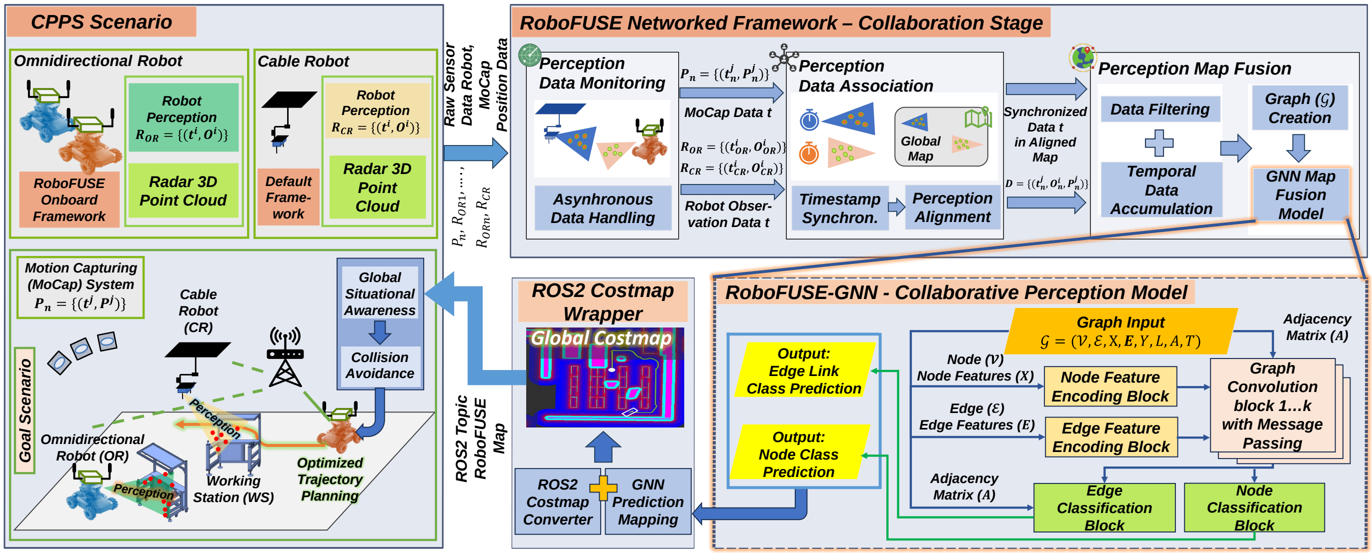

CPPS Scenario
Omnidirectional Robot
Robot Perception $R_{OR} = \{(t^i, O^i)\}$
RoboFUSE Onboard Framework
Radar 3D Point Cloud
Cable Robot
Robot Perception $R_{CR} = \{(t^i, O^i)\}$
Default Frame-work
Radar 3D Point Cloud
Motion Capturing (MoCap) System $P_n = \{(t^j, P^j)\}$
Goal Scenario
Cable Robot (CR)
Global Situational Awareness
Collision Avoidance
Perception
Omnidirectional Robot (OR)
Perception
Working Station (WS)
Optimized Trajectory Planning
Raw Sensor Data Robot, MoCap Position Data
$P_n, R_{OR1}, \cdots, R_{ORn}, R_{CR}$
RoboFUSE Networked Framework – Collaboration Stage
Perception Data Monitoring
Asynhronous Data Handling
$P_n = \{(t^j_n, P^j_n)\}$
MoCap Data t
$R_{OR} = \{(t^i_{OR}, O^i_{OR})\}$
$R_{CR} = \{(t^i_{CR}, O^i_{CR})\}$
Robot Obser-vation Data t
Perception Data Association
Global Map
Timestamp Synchron.
Perception Alignment
Synchronized Data t in Aligned Map
$D = \{(t^j_n, O^i_n, P^j_n)\}$
Perception Map Fusion
Data Filtering
Temporal Data Accumulation
Graph ($\mathcal{G}$) Creation
GNN Map Fusion Model
ROS2 Costmap Wrapper
Global Costmap
ROS2 Costmap Converter
GNN Prediction Mapping
ROS2 Topic RoboFUSE Map
RoboFUSE-GNN - Collaborative Perception Model
Graph Input $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X, \boldsymbol{E}, Y, L, A, T)$
Adjacency Matrix (A)
Output: Edge Link Class Prediction
Output: Node Class Prediction
Node ($\mathcal{V}$) Node Features (X)
Edge ($\mathcal{E}$) Edge Features (E)
Adjacency Matrix (A)
Node Feature Encoding Block
Edge Feature Encoding Block
Graph Convolution block 1…k with Message Passing
Edge Classification Block
Node Classification Block

Fig. 2: Proposed overall system architecture for collaborative robotic perception in this work. It consists of four major blocks: Physical CPPS scenario, RoboFUSE networked framework, overview of GNN-model architecture, ROS2 costmap wrapper.

in Sec. II. Sec. III specifies the key enablers for this work. The GNN approach is described in Sec. IV. Thereafter, Sec. V and Sec. VI demonstrate the experiment setups and validations. Finally, Sec. VII summarizes the paper and offers the outlook.

II. Related Work

A. Graph Neural Networks in Robotics

Recent GNN-based and collaborative mapping methods address scalability and perception sharing in multi-robot systems. Tab. I summarizes recent approaches across key performance indicators (KPIs). Authors in [7] and [10] offer decentralized perception but struggle with occlusion or inadequate uncertainty handling. Sharma et al. [11] incorporate risk-awareness yet omit collaborative mapping. Lin et al. [12] focus on central costmap fusion yet depend on vision-based sensors, which reduce robustness under occlusion. Most approaches rely on Wi-Fi and still lack a unified map. Meanwhile, RoboFUSE-GNN offers a collaborative perception framework integrating a GNN model for unified mapping using an emulated 6G ISAC.

TABLE I: GNN-based multi-robot perception approaches.

Method	RTF	UA	OAP	UM	S&C
Kimera-Multi [3]	●	○	◐	◐	Camera, Wi-Fi
Geo-GNN [4]	◐	○	○	○	Sim. LiDAR, -
LPAC [7]	●	◐	◐	○	LiDAR, Wi-Fi
Worker-Aware [10]	●	◐	●	○	Sim. LiDAR, -
Risk-Aware [11]	◐	●	◐	○	Sim. LiDAR, Wi-Fi
Collab. Nav. [12]	◐	○	○	●	LiDAR, Wi-Fi
RoboFUSE-GNN	●	●	●	●	Radar, 6G-Emulated

(*)Legend: ● = Full Support, ◐ = Partial Support, ○ = Not Addressed; RTF: Real-Time Fusion, UA: Uncertainty-Aware, OAP: Occlusion-Aware Perception, UM: Unified Mapping, S&C: Sensing and Communication Tech.

B. Towards 6G ISAC in Robotics

Radar-based systems have significantly expanded on mobile platforms, like in the automotive and robotics fields [13]. Their accurate perception in harsh environments enhances indoor and outdoor sensing reliability. Technological advances in MIMO radar systems improve the angular resolution significantly via virtual antenna arrays. Current systems reach up to 5 GHz bandwidths, enabling more precise object detection. Recent 4D radar systems measure three spatial dimensions of range, azimuth, and elevation, as well as Doppler shift [14]. This system produces a robust 3D point cloud suited for neural networks [15]. Emerging 6G technologies offer new high-resolution frequency bands but suffer from higher attenuation. This requires high-gain antennas with a limited field of view (FOV). Distributed sensor systems can compensate for this limitation [16]. Frequency-Modulated Continuous Waveform (FMCW) radar remains a prevalent method for distance measurement and object recognition in harsh environments. It complements LiDAR with robust, high-resolution sensing, well-suited for next-generation autonomous systems [17], [18].

However, current wireless technologies struggle with scalable robotics communication. Each Automated Guided Vehicle (AGV) in a smart factory may generate up to $15\,\mathrm{MB\,s^{-1}}$ of sensor data and requires sub-$1\,\mathrm{ms}$ latency real-time coordination [6], [19]. A fleet of 200 AGVs results in $\approx 3\,\mathrm{GB\,s^{-1}}$ of uplink traffic. Industrial Wi-Fi typically shows $\approx 10\,\mathrm{ms}$-$50\,\mathrm{ms}$ latency, rising beyond $250\,\mathrm{ms}$ under load [20]. Even 5G mid-band supports only $\approx 500\,\mathrm{MB\,s^{-1}}$-$800\,\mathrm{MB\,s^{-1}}$ in uplink throughput, leading to Round-Trip Time (RTT) of over $40\,\mathrm{ms}$ in high-traffic scenarios [6]. These exceed the strict latency of $\leq 20\,\mathrm{ms}$ RTT for perception-aware control [6], [9]. 6G ISAC overcomes these limits through unified sensing like radar and communication, enabling deterministic sub-ms latency and scalable data fusion across robot fleets [21].

III. Key Enablers for GNN Model Development

In this section, we provide a brief overview of the CPPS scenario [1], [22] and our previous RoboFUSE research [9]. The key enablers support the experiments and development of the GNN model for multi-robot perception using RoboFUSE's emulated 6G radar sensing.

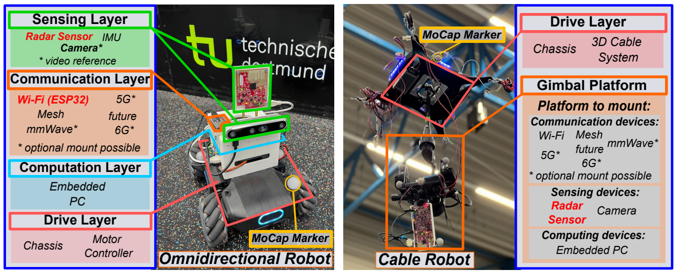


Fig. 3: Robotic platforms for dataset collection. The omnidirectional robot utilizes the RoboFUSE onboard framework [9], while the cable robot employs default framework.

A. CPPS Setup for a Perception-Aware Robotics Testbed

CPPS enables flexible manufacturing by integrating digital control, AGVs, and intelligent workstations. As shown in Fig. 1, raw materials flow from inbound zones to reconfigurable assembly stations, while finished goods are routed to outbound areas. Unlike static factories, CPPS dynamically reconfigures the workstation based on real-time production demands [1], [22]. AGVs transport material without predefined routes, navigating through the CPPS area. This introduces spatial-temporal uncertainties and dynamic obstacles. Each AGV must perceive its surroundings, adjust its behavior, and communicate with other AGVs in real-time. Such a task challenges centralized planning and highlights the need for decentralized, perception-aware collaboration.

B. Robotic Platforms for Multi-View Collaborative Sensing

The ground robot and aerial robot platforms are deployed as in Fig. 3. A ground-based Omnidirectional Robot (OR) replicates AGV, equipped with a camera and 60 GHz TI IWR6843ISK mmWave radar for floor-level perception. As proof-of-concept, an ESP32 module provides Robot-to-Robot (R2R) communication. To obtain bird's-eye view coverage, a Cable Robot (CR) employs an mmWave radar sensor This setup enables how different viewpoints contribute to shared situational awareness in cluttered and dynamic layouts. Both platforms provide synchronized multi-perspective radar data to evaluate collaborative mapping models.

C. RoboFUSE Onboard Framework: Emulating 6G ISAC

Each OR utilizes a lightweight RoboFUSE onboard framework to emulate 6G ISAC functionalities. As shown in Fig. 4, the ISAC Manager and Controller (ISMAC) controls the time-division (TD) allocation of FMCW radar sensing and Orthogonal Frequency Division Multiplexing (OFDM)-based communication. This design enables deterministic sensor data acquisition for sensing and synchronized message exchange across the fleet. Sensor outputs are encoded as single-agent perception results (SAPR) for the collaborative mapping model. The onboard module is integrated with Robot Operating System 2 (ROS2) for real-time navigation and control.

D. RoboFUSE Networked Framework Overview

In this work, we extend RoboFUSE with a networked framework that focuses on fusing multi-robot radar data and building a unified global spatial-semantic map. Each robot sends its radar SAPR data via ROS2 topic to an edge computing server, where the Collaborative Perception Management Layer (CPML) fuses SAPR data using RoboFUSE-GNN. The fused result is redistributed as a perception map with path-planning recommendations. This setup is depicted in Fig. 2.

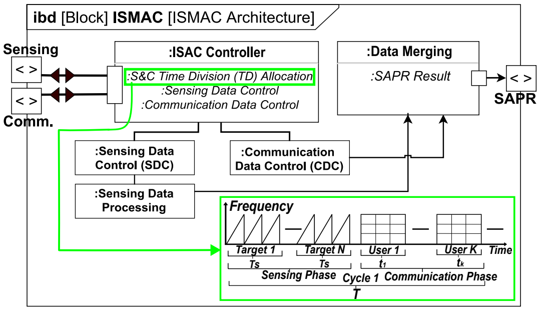


Fig. 4: ISMAC layer of RoboFUSE onboard framework orchestrates TD slicing for sensing and communication [9].

IV. GNN-Based Collaborative Robotic Perception Approach

This section outlines the steps involved in developing RoboFUSE GNN multi-robot perception model. We develop custom datasets by recording radar data using the robotic platforms in Fig. 3. Motion Capturing (MoCap) systems act as a reference system, providing the positions of marked objects with sub-mm accuracy. The collected data are then processed and structured into graphs for subsequent model training.

A. Dataset Creation Pipeline

The proposed pipeline for dataset creation is defined in Fig. 5, which is built upon the RoboFUSE Network Framework backbone in Fig. 2. First of all, radar data and MoCap data are collected asynchronously across multiple robotic platforms and a PC. As shown in Tab. II, we define $\mathcal{R}_n = \{(t^i, O^i)\}$ as the set of radar observations for robot n, and $\mathcal{P}_n = \{(t^j, P^j)\}$ as the corresponding MoCap pose references. Each radar observation is defined as $\mathcal{O}^i = (x, y, z, s, r, \phi, \theta, v)$, and each pose reference as $\mathcal{P}^j = (x, y, z, \alpha, \beta, \gamma, \psi)$. Afterward, timestamp synchronization between radar and MoCap measurements is carried out. This step is performed by matching to the closest MoCap timestamp for each robot:

$$\hat{P}_n(t_i) = \arg\min_{t_j} \left| t_i^{(R_n)} - t_j^{(M_n)} \right|, \quad \forall t_i \in \mathcal{R}_n \qquad (1)$$

Then $P_n(t_i)$ is merged with the radar robot observation N $O_n(t_i)$. To synchronize radar data across robots, all streams are aligned to a specific reference timestamp from the robot

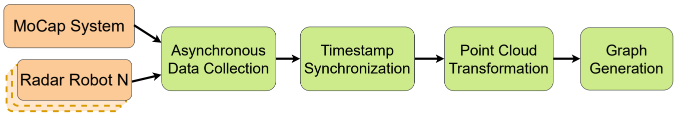


Fig. 5: Dataset creation pipeline to generate graph-based data from multi-robot radar perception and a MoCap system.

TABLE II: Sample rates, data sources, and data recorded including notations for radar robot N and MoCap system.

	Radar Robot N	**MoCap System**
Sample Rate	30 Hz	200 Hz
Data Sources	$n = \{1, \dots, 3\}$ robots $\mathcal{R}_n$: each robot's radar data	1 Tracking System
Data Recorded	3D Position (x, y, z) [m] Signal-to-Noise Ratio s (SNR) [dB] Range r [m], Azimuth ϕ [°] Elevation θ [°], Velocity v [$\mathrm{m\,s^{-1}}$]	Position (x, y, z) [m] Orientation (α, β, γ) [rad] Yaw ψ [rad]
Dataset	$\mathcal{R}_n = \{(t^0, O^0), \dots, (t^l, O^l)\}$ $O^i = (x_i, y_i, z_i, s_i, r_i, \phi_i, \theta_i, v_i)$	$\mathcal{P}_n = \{(t^0, P^0), \dots, (t^l, P^l)\}$ $P^j = (x_j, y_j, z_j, \alpha_j, \beta_j, \gamma_j, \psi_j)$
# Features/Labels	8 per radar point	7 per pose reference

with the highest sample count. Other robots' data are aligned to the nearest reference timestamp, denoted as:

$$t_{\text{ref}} = \arg\max_n |\mathcal{R}_n|, \quad \forall t_k \in \mathcal{R}_{\text{ref}}, \quad \hat{\mathcal{R}} = \bigcup_n \{\mathcal{R}_n(t_k^*)\} \quad (2)$$

$$\text{where} \quad t_k^* = \arg\min_{t_i \in \mathcal{R}_n} \left| t_k^{(\text{ref})} - t_i^{(R_n)} \right| \quad (3)$$

Consecutively, each radar point cloud is transformed into the global MoCap frame using the transformation matrix as $\mathbf{p}_{\text{g}} = \mathbf{T} \cdot \mathbf{p}_l + \mathbf{t}$, where $\mathbf{T} \in \mathbb{R}^{2\times2}$ for ground robots and $\mathbf{T} \in \mathbb{R}^{3\times3}$ for the cable robot. The final raw dataset $\mathcal{D}$ is defined as the set of synchronized and globally aligned observations:

$$\mathcal{D} = \{(t^0, \mathbf{O}_{\text{g}}^0, P^0), \dots, (t^k, \mathbf{O}_{\text{g}}^k, P^k)\}_{k=0}^{N-1}, \quad (4)$$

where $\mathbf{O}_{\text{g}}^k$ represents the observation radar data with global transformed points of the MoCap frame and its features, and P^k is the corresponding MoCap data. In the following subsection, dataset $\mathcal{D}$ serves as the input for graph construction.

B. Graph Construction and Formulation

The collaborative graph is constructed from the synchronized raw dataset $\mathcal{D}$. Alg. 1 defines the complete graph construction process. Initially, the dataset is filtered based on arena boundaries, SNR thresholds, and height constraints. To capture temporal dependencies, radar data is optionally accumulated over a sliding window of size W. This temporal accumulation enhances the graph representation by incorporating spatial and temporal context. After accumulation, each radar point cloud is mapped to a graph node v_i, characterized by a set of normalized features that include spatial coordinates, radar-specific attributes, and robot identifiers:

$$\begin{aligned} \mathbf{x}_i = [&x_i, y_i, z_i, \hat{s}_i, \hat{r}_i, \\ &\sin(\hat{\phi}_i), \cos(\hat{\phi}_i), \sin(\hat{\theta}_i), \cos(\hat{\theta}_i), R_{\text{id}}], \end{aligned} \quad (5)$$

where radar-specific attributes are normalized $(\hat{s}_i, \hat{r}_i, \hat{\phi}_i, \hat{\theta}_i)$, excluding spatial coordinates and robot identifiers (R_{id}). Semantic node labels y are derived using MoCap footprints, categorizing nodes into predefined object classes. Edges $\mathcal{E}$ are generated via K-nearest neighbors (KNN) and formulated as:

$$\mathcal{E} = \{(v_i, v_j) \mid v_j \in \mathcal{N}_k(i)\}, \quad (6)$$

where $\mathcal{N}_k(i)$ represents the set of k closest neighbors. The edge features e_{ij} of each connection are defined as:

$$e_{ij} = [\Delta x_{ij}, \Delta y_{ij}, \Delta z_{ij}, \Delta s_{ij}, \Delta t_{ij}] \quad (7)$$

Algorithm 1 Graph Construction from Filtered Dataset $\mathcal{D}$

```
Require: Filtered Dataset $\mathcal{D} = \{(t^k, O_g^k, P^k)\}_{k=0}^{N-1}$, parameters $k$, $d_{max}$, $d_{collab}$,
    temporal window size $W$
Initialize empty sets: $\mathcal{V}$ (nodes), $\mathcal{E}$ (edges), $X$ (node features), $\mathbf{E}$ (edge features),
    $Y$ (node labels), $L$ (edge labels)
for $k = 0, \dots, N-1$ do                                          ▷ Data Accumulation
    if $W > 0$ then
        $\mathcal{D}_{temp} \leftarrow \bigcup_{t=\max(0,k-W+1)}^{k} \mathcal{D}(t)$
        Set graph timestamp $t \leftarrow t_k$                       ▷ Latest timestamp
    else
        $\mathcal{D}_{temp} \leftarrow \mathcal{D}(t_k)$
        Set graph timestamp $t \leftarrow t_k$
    end if
    for all point $p \in \mathcal{D}_{temp}$ do                       ▷ Node Creation
        Extract position and normalized features: $(x, y, z, f)$
        Add $p$ to $\mathcal{V}$, and $f$ to $X$
    end for
    for all node $v_i \in \mathcal{V}$ do Node Labeling
        Assign semantic label $y_i$ based on MoCap footprint at timestamp $t$
        $Y \leftarrow Y \cup \{y_i\}$
    end for
    Run KNN($\mathcal{V}$) to determine edges $\mathcal{E}$ based on proximity    ▷ Edge Creation
    for all node pair $(v_i, v_j)$ do
        Compute edge features $e_{ij}$
        Add edge $(v_i, v_j)$ to $\mathcal{E}$, features $\mathbf{e}_{ij}$ to $\mathbf{E}$
    end for
    for all $(v_i, v_j) \in \mathcal{E}$ do                            ▷ Edge Labeling
        Compute distance $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$ and determine threshold $\delta$:
        if $R_{id}(i) = R_{id}(j)$ then
            $\delta \leftarrow d_{max}$                 ▷ Intra-robot connection (same robot)
        else
            $\delta \leftarrow d_{collab}$              ▷ Inter-robot connection (different robots)
        end if
        $l_{ij} \leftarrow \mathbf{1}(y_i = y_j \wedge d_{ij} \leq \delta)$
        $L \leftarrow L \cup \{l_{ij}\}$
    end for
end for
Construct adjacency matrix $A$ from edge set $\mathcal{E}$
Final Graph Construction:
$\hat{\mathcal{G}} \leftarrow (\mathcal{V}, \mathcal{E}, X, \mathbf{E}, Y, L, A, t)$
Ensure: Graph $\hat{\mathcal{G}}$
```

Edge creation and labeling play crucial roles in modeling an uncertainty-aware GNN by capturing intra-robot and inter-robot relationships. We introduce a novel edge labeling strategy that explicitly distinguishes between connections within the same robot (intra-robot) and across different robots (collaborative), enhancing multi-robot perception. Intra-robot edges connect nodes from the same robot within a distance threshold d_{max}, maintaining local spatial consistency and reducing uncertainty in single-robot perception. Inter-robot edges or collaborative edges connect nodes from different robots within a more relaxed distance threshold d_{collab}. These edges facilitate multi-robot data integration and address spatial uncertainty from diverse viewpoints, defined as:

$$\mathcal{E}_{\text{collab}} = \{(v_i, v_j) \in \mathcal{E} \mid R_{\text{id}}(i) \neq R_{\text{id}}(j)\} \quad (8)$$

This approach strengthens global mapping by leveraging collaborative perception. Edges' labels L are classified as: an edge is marked as 1 if the nodes share the same semantic class and the distance criterion is met. Otherwise, the edge is labeled as 0. The indicator function $1(\cdot)$ is used to express this logic in Alg. 1 succinctly. The constructed graph $\mathcal{G}$ consists of node features X, edges $\mathcal{E}$, edge attributes $\mathbf{E}$, semantic labels Y, edge labels L, adjacency matrix A, and timestamps T. Each frame $\mathcal{G}_f$ is constructed from a series of graphs $\{\mathcal{G}_1 \dots, G_k\}$. Ultimately, the final temporal graph dataset is defined as a sequence of these frames: $\mathcal{F} = \{\mathcal{G}_f^{(1)}, \dots, \mathcal{G}_f^{(T)}\}$.

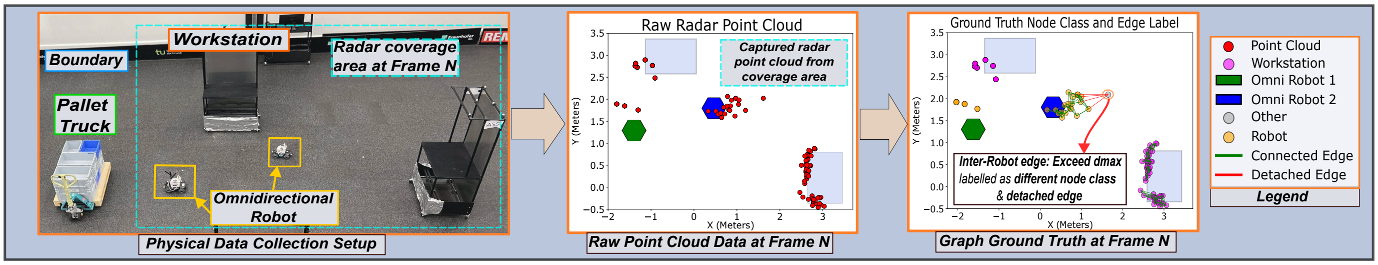

Boundary
Workstation
Radar coverage area at Frame N
Pallet Truck
Omnidirectional Robot
Physical Data Collection Setup
Raw Radar Point Cloud
Captured radar point cloud from coverage area
Y (Meters)
X (Meters)
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−2
−1
0
1
2
3
Raw Point Cloud Data at Frame N
Ground Truth Node Class and Edge Label
Inter-Robot edge: Exceed dmax labelled as different node class & detached edge
Graph Ground Truth at Frame N
Point Cloud
Workstation
Omni Robot 1
Omni Robot 2
Other
Robot
Connected Edge
Detached Edge
Legend

Fig. 6: Transforming all collected raw point clouds from multiple robots in the physical testbed into a graph using Alg. 1

C. GNN Model Architecture

We propose the GNN architecture as depicted in Fig. 2, building on the foundational concepts from this module[1]. Our approach introduces **key enhancements** to address multi-robot perception and dynamic CPPS scenario challenges. The model first takes input from frames or a set of graphs $\mathcal{G}$ and extracts the graph information using node features and edge features embedding blocks. This step transforms the input features into a higher-dimensional space, capturing more intricate representations. The extracted features are then forwarded to graph convolution blocks with message-passing modules. These blocks are connected sequentially, expanding the receptive field one hop at a time. They aggregate and update neighboring node and edge features, capturing spatial and contextual relationships.

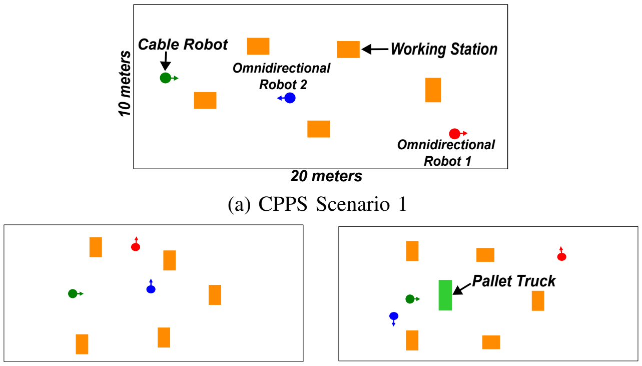
TABLE III: GNN model architecture.

Layer Type	Input Shape	Output Shape	Activation
Input	10 (Node Features)	-	-
Input	5 (Edge Features)	-	-
Node Encoding	10	64	LeakyReLU
Edge Encoding	5	64	LeakyReLU
Graph Convolution	[64, 64, 64]	64	LeakyReLU
Node Classification	64	5 Classes	Softmax
Edge Classification	64	2 Classes	Sigmoid

Tab. III represents the detailed GNN model architecture based on Fig. 2. The GNN core architecture comprises three graph convolutional layers with 64 feature dimensions. Each layer is linked to a Multilayer Perceptron (MLP) with 128 hidden layers, generating 64-dimensional outputs. The **main objective** of the GNN model is to enhance global situational awareness using a graph-based representation. The key tasks are: **(1) Node Classification**, which detects and predicts semantic classes of the warehouse objects; **(2) Edge Classification**, which predicts graph edge connections, facilitating the integration of collaborative observations and addressing environmental uncertainty. The predicted outputs are clustered to form spatial ellipsoids representing detected objects. These clusters can then be wrapped into a global ROS2-compatible costmap, incorporating spatial and semantic information. Hence, the generated costmap can support collaborative navigation and real-time trajectory planning. The proposed GNN model and ROS2 costmap are designed to be integrated into the RoboFUSE networked framework.

V. Experimental Setup

This section describes the experiment setups to validate the collaborative robotic perception architecture proposed in Fig. 2. The validation focuses primarily on developing the RoboFUSE GNN model described in Sec. IV. This work excludes the ROS2 costmap wrapper part. The experiments are carried out in the Inovationlab[2], a logistics warehouse testbed at the Technical University (TU) of Dortmund with a size of $570\,\mathrm{m}^2$. Fig. 2 and 6 portray the physical experiment setup. The experiment scenarios vary the obstacle layouts as shown in Fig. 7) to replicate dynamic workstation changes, robot movements, and pallet truck in CPPS-based warehouses.



(a) CPPS Scenario 1

(b) CPPS Scenario 2

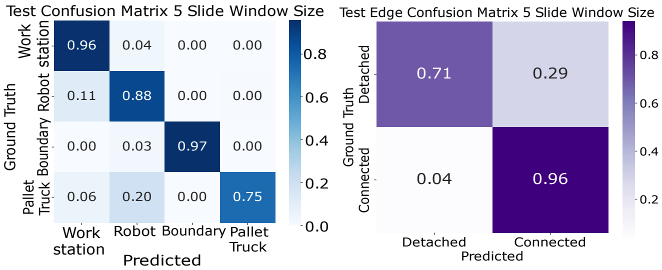
(c) CPPS Scenario 3

Fig. 7: CPPS setups at the Innovationlab TU Dortmund.

Dataset Collection: To record the data for multi-robot perception, we implement two ground ORs and an aerial CR platforms in Fig. 3. The lab features 55 marker-based MoCap cameras that accurately track and locate marked objects with sub-mm precision. In this recording, the MoCap system is the reference localization system for the robots and obstacles' positions. For each scenario, we simultaneously record each radar robot data ($\mathcal{R}_n$) and the MoCap data ($\mathcal{P}_n$) as defined in Tab. II. Each workstation has a length of $1.1\,\mathrm{m}$, a width of $0.75\,\mathrm{m}$, and a height of $2\,\mathrm{m}$. The pallet truck has a length of $1.7\,\mathrm{m}$, a width of $0.8\,\mathrm{m}$, and a height of $1\,\mathrm{m}$. MoCap also tracks the pallet truck and all workstations to provide ground truth data. The experiment area is covered by the boundary with a length of $20\,\mathrm{m}$ and a width of $10\,\mathrm{m}$. The CR is set in a static position on the left side of the arena at a height of $4\,\mathrm{m}$. Two ground robots navigate autonomously through the obstacles using predefined waypoints with a speed of $\approx 1\,\mathrm{m\,s^{-1}}$.

[1] https://github.com/UditBhaskar19/GRAPH_NEURAL_NETWORK_FOR_RADAR_PERCEPTION

[2] https://www.innovationlab-logistics.com/research-centre/



(a) Node Classification

(b) Edge Classification

Fig. 8: Confusion matrices (node classification (a) and edge classification (b)) of GNN model performances in a dataset with a sliding window size of 5.

Both robots' movements are varied with vertical, horizontal, and random patterns. This recording scenario ensures that all obstacles and the whole arena are recorded. Fig. 6 depicts an example of physical data collection using two ORs. Both robots generate raw point data, which is then transformed into a graph frame $\mathcal{G}_f$ using Alg. 1.

Training GNN model: In total, we record 37 measurement scenes covering the three CPPS scenarios. First of all, the raw data is synchronized and processed, as depicted in Fig. 5 and following the steps in Sec. IV-A. This step produces dataset $\mathcal{D}$, which is then transformed to graph $\mathcal{G}$ based on the procedures in IV-B. We enable the temporal accumulation and generate various temporal datasets $\mathcal{D}_{\text{temp}}$ in different sliding window sizes of 3, 5, 7, and 10. This method improves the GNN model's accuracy in identifying dynamic obstacles and predicting spatial-temporal interactions. Finally, $\approx 63,835$ frames are generated for each temporal sliding windows dataset. For each temporal dataset, we allocate 25 scenes for training, 7 for validation, and 5 for testing. The GNN model is trained on a high-performance PC running Ubuntu 22.04. The system features an AMD Ryzen 9 CPU with 16 cores operating at 4.2 GHz and 64 GB of RAM. The training utilizes an NVIDIA GeForce RTX 5090 GPU, offering significant computational power for handling complex GNN operations.

VI. Validation and Results

To the best of the authors' knowledge, this work introduces the first graph-based dataset for multi-robot perception mapping in smart warehouse environments. The dataset uniquely captures emulated 6G radar sensing using RoboFUSE onboard framework within multi-robot platforms. Hence, the current work offers a novel benchmark for graph-based collaborative mapping towards 6G-enabled smart warehouse systems. Our dataset for this work is publicly available on Kaggle[3]. We evaluate the proposed RoboFUSE-GNN model offline using different temporal graph datasets across dynamic CPPS layouts for robust and comprehensive validation. To evaluate the GNN model, we use test datasets that remain unseen during training for all of the following validations. This ensures the model's performance is assessed on new and unpredictable scenarios, reflecting its generalization ability.

TABLE IV: Node classification performance.

Sliding Window	Accuracy	Precision	Recall	F1 Score
3	0.9393	0.9325	0.8562	0.8927
5	0.9214	0.9314	0.8882	0.9093
7	0.9259	0.9107	0.8761	0.8931
10	0.9218	0.9184	0.8359	0.8752

TABLE V: Edge classification performance.

Sliding Window	Accuracy	Precision	Recall	F1 Score
3	0.8761	0.9863	0.8806	0.9305
5	0.9526	0.9940	0.9574	0.9754
7	0.9085	0.9896	0.9150	0.9508
10	0.9043	0.9898	0.9110	0.9488

A. *RoboFUSE-GNN Performance Evaluation*

To evaluate the performance of the proposed RoboFUSE GNN model, we use standard machine learning metrics: accuracy, precision, recall, and F1 score. These metrics are derived from the confusion matrix. Accuracy calculates the ratio of correctly predicted instances to total instances. Precision quantifies the ratio of correctly predicted positive instances to the total predicted positives. Recall indicates the ratio of correctly predicted positives to the total actual positives. The F1 score is a harmonic mean of precision and recall, providing a balanced evaluation metric. It is useful, especially when the class distribution is imbalanced.

The confusion matrix in Fig. 8 (a) shows the GNN model's node classification performance across four object classes: *Workstation, Robot, Boundary, and Pallet Truck*. The model achieves high accuracy for "Workstation" (0.96) and "Boundary" (0.97), while "Pallet Truck" has lower accuracy (0.75), with occasional misclassification as "Robot". This outcome arises from the limited occurrence of the pallet truck object. Since it is introduced only in layout 3, maintaining a consistent prediction becomes challenging. In Fig. 8 (b), the model shows high accuracy for predicting "Connected" edges (0.96). However, it struggles with "Detached" edges (0.29), indicating challenges in capturing weaker inter-robot connections.

Tab. IV and V demonstrate the RoboFUSE-GNN performance across different sliding window sizes for node and edge classification, respectively. The evaluation is derived from the average sum of all object classes using equations 9 - 12. The optimal F1 score of 0.9093 for node classification occurs at a five-sliding window size, although it has the lowest accuracy. Yet, the F1 score indicates the best balance between precision (0.9314) and recall (0.8882) among other objects. Similarly, for edge classification, the sliding window of five yields the highest F1 score (0.9754), reflecting exceptional precision (0.9940) and strong recall (0.9574). The model demonstrates robust prediction capabilities, even with unseen data. This highlights its ability to capture temporal dependencies, which are vital in dynamic CPPS environments. Optimal performance with a sliding window size of five indicates that the model uses temporal information to enhance predictive accuracy. This approach aligns with the uncertainty-aware perception concept, as the model adapts to spatial and temporal changes.

[3] https://www.kaggle.com/datasets/asfand59/robofuse-gnn-dataset

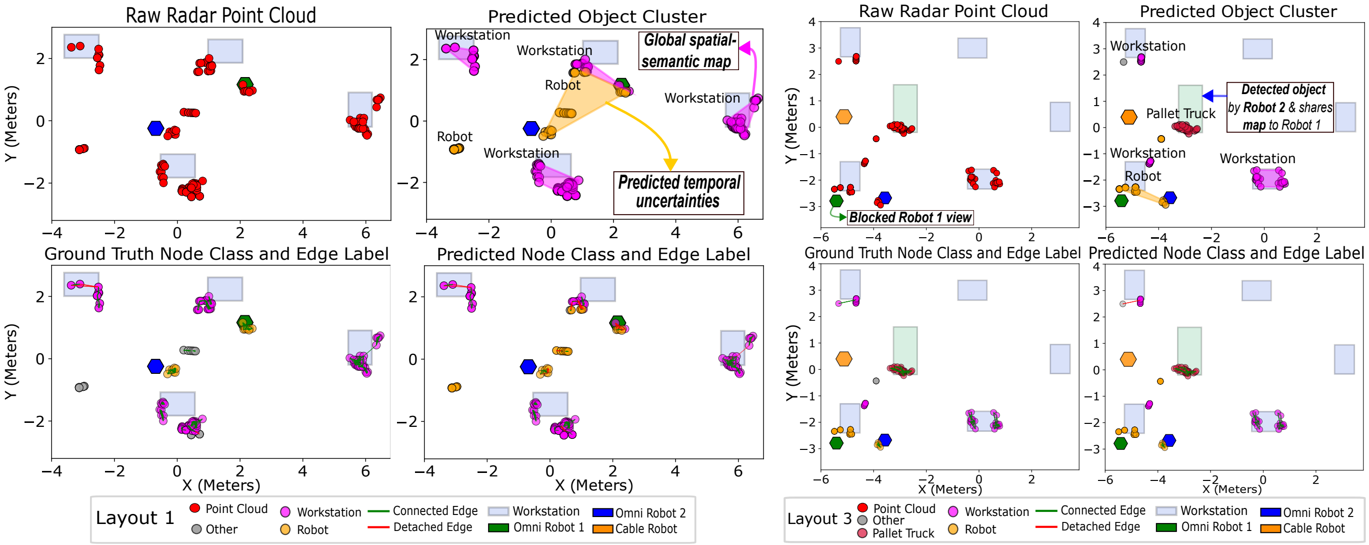


Fig. 9: Highlights of visualization results for diverse CPPS layouts using GNN model of five sliding window size and unseen test dataset: raw point cloud data, ground truth, predicted object clusters, and predicted node and edge classes.

B. Prediction Visualization

In this evaluation, we visualize and analyze the predicted node classes with their respective object clusters and the predicted edge connections. As an evaluation sample, Fig. 9 visualizes representative frames from unseen test datasets in Layout 1 and Layout 3. These layouts highlight the RoboFUSE-GNN model's ability to generalize dynamic object arrangements. Fig. 9 exhibits key visualization results of the model's performance. The raw radar point clouds emphasize the complexity and distribution of observed points. In Layout 1, the model accurately identifies most object clusters in dense point clouds, despite minor misclassifications due to dynamic robot movements. Here, the "Other" class in the ground truth is classified as "Robot" in the predicted node class. This demonstrates how GNN can capture temporal and spatial divergences in dynamic scenes. The model can distinguish between transient positional shifts and consistent spatial patterns by incorporating temporal data from sliding window sequences. Thus, the uncertainty in multi-robot perception is minimized. Such a feature is critical to ensure reliable shared situational awareness with other robots, particularly in Layout 3. Collaborative perception capability between robots is exhibited in Layout 3. In this setting, the Workstation obstructs Robot 1's view as it attempts to navigate to an unknown area. To support Robot 1, Robot 2 detects the "Pallet Truck" and shares the information through the global map containing the geometric and semantic data. Consequently, Robot 1 learns where and what type of obstacles are beyond its vision range. Finally, it can optimize its trajectory planning according to obstacle type and thus navigate safely. Additionally, the corresponding node and edge predictions (bottom row) in Fig. 9 validate the model's robust inference of semantic classes and inter-node relationships. The validation results showcase RoboFUSE-GNN's adaptability in collaborative reasoning and uncertainty-aware mapping within various warehouse layouts.

C. Mapping Accuracy Assessment

As shown in Fig. 9, the predicted object clusters can be used to reconstruct the global map. Accordingly, this section assesses the accuracy of the cluster mapping using specific evaluation metrics. To evaluate the spatial alignment between the predicted boundaries and ground truth, we introduce the Average Distance to Boundary (ADB) metric. ADB quantifies the average minimum distance from the nearest point of the predicted clusters' boundaries to each ground truth point. This metric indicates how well the predicted boundary overlaps the ground truth. As the cluster prediction covers only a partial part of the obstacles, ADB is suitable to evaluate the collaborative mapping and formulated as:

$$\mathrm{ADB} = \frac{1}{|B|} \sum_{b \in B} \min_{a \in A} \|a - b\|, \tag{9}$$

where B is the set of ground truth boundary points, A represents set of predicted boundary points, and $|a - b\|$ is the euclidean distance between a ground truth point b and a predicted boundary point a. The five sliding window size model is selected to evaluate the ADB metrics for each object class. as depicted in Fig. 10. The ADB results reveal that the "Robot" achieves the highest localization accuracy, with a mean of $39\,\mathrm{cm}$. That indicates precise boundary predictions and is closely aligned with the ground truth. As the objects get larger, like in "Pallet Truck" and "Workstation", the higher mean ADB results are obtained, with a mean of $\approx 50\,\mathrm{cm}$ for both classes. However, the "Boundary" class exhibits a higher ADB mean of $2.1\,\mathrm{m}$, indicating the lowest localization accuracy. This limitation arises primarily from the broader spatial distribution and the absence of precise ground-truth positions, as "Boundary" is beyond the MoCap range. Despite this limitation, the "Boundary" class is included to reflect realistic prediction scenarios. Other models with different sliding window sizes also demonstrate the same pattern.

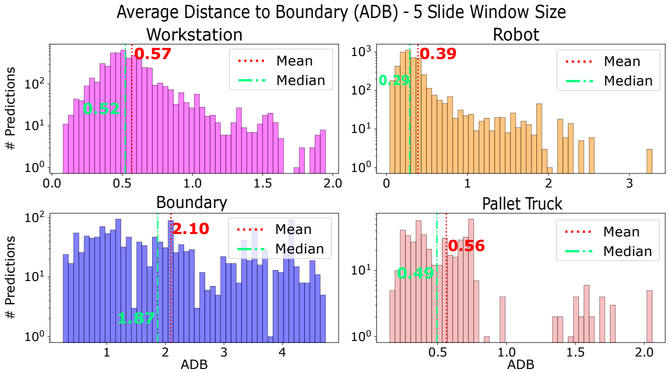


Fig. 10: ADB evaluation metrics for 5 sliding window size.

VII. Conclusion and Outlook

We present RoboFUSE-GNN, a novel uncertainty-aware GNN framework for collaborative robot mapping in dynamic CPPS warehouse environments. Leveraging radar-based sensing towards 6G ISAC principles, RoboFUSE-GNN integrates multi-robot perception with temporal graph learning. This addresses the limitations of vision-based systems, which are less effective under cluttered and low-visibility settings. Our approach models spatial-temporal uncertainties using a novel edge labeling strategy. This method characterizes intra- and inter-robot connections, enhancing the relational modeling. Therefore, custom datasets are created to develop the RoboFUSE-GNN model. Experimental results reveal that using five-temporal sliding windows optimally enhances semantic classification, cluster boundary estimation, and collaborative edge prediction. The predicted clusters of RoboFUSE-GNN can be reconstructed into the global spatial-semantic map. Sharing semantic maps enables robots to coordinate tasks through a shared understanding of the environment. Overall, RoboFUSE-GNN offers a robust, shared situational awareness framework for reliable collaborative perception under uncertainty. Future work includes real-time deployment on an edge platform using the RoboFUSE networked framework. This generates a global ROS2-compatible costmap for real-time multi-robot collision avoidance.

Concerning the 6G ISAC technology, future research will explore the integration of next-generation radar sensors operating in the 126 GHz to 182 GHz. These frequency ranges provide an increased bandwidth and the possibility of capturing Doppler information. Such enhancements enable precise distance measurement and dynamic obstacle detection, like humans and robots. The increased frequency and bandwidth improve the environmental mapping resolution. This facilitates accurate identification of smaller structures and reduces collision risks. Ultimately, these advancements promise substantial improvements in environmental perception, benefiting robust, autonomous multi-robot navigation in dynamic, 6G-enabled smart warehouse environments.

Acknowledgment

This project is funded by the German Federal Ministry of Education and Research (BMBF) under the funding announcement ”*6GEM* Research Hubs; Platform for Future Communication Technologies and 6G” under the funding codes 16KISK037, 16KISK038, and 16KISK041.

References

[1] H. Bayhan, M. Meißner *et al.*, “Presentation of a novel real-time production supply concept with cyber-physical systems and efficiency validation by process status indicators,” *The International Journal of Advanced Manufacturing Technology*, vol. 108, pp. 527–537, 2020.

[2] Y. Wang, Z. Zhou *et al.*, “Multi-robot obstacle-avoidance formation based on graph neural networks and imitation learning,” in *2024 China Automation Congress (CAC)*. IEEE, 2024.

[3] A. Rosinol, M. Abate *et al.*, “Kimera: an open-source library for real-time metric-semantic localization and mapping,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[4] E. Tolstaya, J. Paulos *et al.*, “Multi-robot coverage and exploration using spatial graph neural networks,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021.

[5] B. R. Nair, “Collaborative perception in multi-robot systems: Case studies in household cleaning and warehouse operations,” *arXiv preprint arXiv:2408.14039*, 2024.

[6] Z. Chen, K.-C. Chen *et al.*, “6G Mobile Communications for Multi-Robot Smart Factory,” *Journal of ICT Standardization*, 2021.

[7] S. Agarwal, R. Muthukrishnan *et al.*, “Lpac: Learnable perception-action-communication loops with applications to coverage control,” *arXiv preprint arXiv:2401.04855*, 2024.

[8] S. Kerboeuf, P. Porambage *et al.*, “Design methodology for 6g end-to-end system: Hexa-x-ii perspective,” *IEEE Open Journal of the Communications Society*, 2024.

[9] I. F. Priyanta, J. Freytag *et al.*, “Towards 6g-driven cooperative robot framework for unified sensing in smart warehouses,” in *2024 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2024, pp. 389–395.

[10] A. Imran, G. Beltrame, and D. St-Onge, “Gnn-based decentralized perception in multirobot systems for predicting worker actions,” *arXiv preprint arXiv:2501.04193*, 2025.

[11] V. D. Sharma, “Enhanced robot planning and perception through environment prediction,” Ph.D. dissertation, University of Maryland, College Park, 2024.

[12] Q. Lin, W. Lu *et al.*, “Efficient collaborative navigation through perception fusion for multi-robots in unknown environments,” *arXiv preprint arXiv:2411.01274*, 2024.

[13] P. Chen, X. Zhao *et al.*, “A Review of Research on SLAM Technology Based on the Fusion of LiDAR and Vision,” *Sensors*, vol. 25, no. 5, p. 1447, Feb. 2025.

[14] A. Palffy, E. Pool *et al.*, “Multi-Class Road User Detection With 3+1D Radar in the View-of-Delft Dataset,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4961–4968, Apr. 2022.

[15] D. Wu, F. Yang *et al.*, “A Survey of Deep Learning Based Radar and Vision Fusion for 3D Object Detection in Autonomous Driving,” 2024.

[16] M. Shafi, R. K. Jha, and S. Jain, “6G: Technology Evolution in Future Wireless Networks,” *IEEE Access*, vol. 12, pp. 57 548–57 573, 2024.

[17] D. Feng, C. Haase-Schutz *et al.*, “Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, Mar. 2021.

[18] Z. Chen, K.-C. Chen *et al.*, “6G Mobile Communications for Multi-Robot Smart Factory,” *Journal of ICT Standardization*, Dec. 2021.

[19] J. Fang, P. Zhu *et al.*, “Resource allocation for embb/urllc coexistence in massive mimo industrial automation,” *IEEE Internet of Things Journal*, 2025.

[20] K. Sui, M. Zhou *et al.*, “Characterizing and improving wifi latency in large-scale operational networks,” in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, 2016, pp. 347–360.

[21] W. Saad, M. Bennis, and M. Chen, “A vision of 6g wireless systems: Applications, trends, technologies, and open research problems,” *IEEE network*, 2019.

[22] L. Krämer, P. Kaiser *et al.*, “Reversing the digital twin–smart-contracting in hybrid production,” in *2023 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*. IEEE, 2023, pp. 1–6.