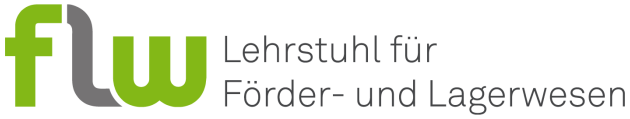
TU DORTMUND UNIVERSITY, DORTMUND, GERMANY

Faculty of Mechanical Engineering



Chair of Material Handling and Warehousing

Master Thesis

Development of a Framework for Collaborative Perception Management Layer for Future 6G-Enabled Robotic Systems

First Examiner: Prof.'in Dr.-Ing. Alice Kirchheim

Second Examiner: Irfan Fachrudin Priyanta, M.Sc.

Presented by: Thiyanayugi Mariraj

Matriculation number: 241940

Course: Master of Science in Automation and Robotics

Issue date: 06. 01. 2025

Submission date: 07. 07. 2025

Dortmund, Germany

Abstract

Robotic perception systems in modern warehouse automation face previously unknown challenges due to blind spots, occlusions, and the dynamic environmental conditions common in industrial logistics facilities. This thesis creates and assesses a thorough framework for collaborative perception management in warehouse robotics. This uses millimeter-wave (mmWave) radar technology, as a proof of concept for upcoming Sixth Generation wireless technology (6G)-enabled integrated sensing and communication capabilities.

This research presents a comprehensive evaluation of a complete framework for collaborative perception management in a swarm of Autonomous Mobile Robot (AMR)s. The proof of concept for 6G employs mmWave radar as its principal sensor technology. This use of mmWave radar serves as an indirect path for evaluating the future Integrated Sensing and Communication (ISAC) capabilities of the 6G wireless standard.

The thesis presents a novel preprocessing pipeline that turns unrefined collaborative sensor data into structured graph representations that can be processed by Graph Neural Network (GNN)s. Fundamental issues are addressed by this methodology. These issues include spatial alignment across robot-centric coordinate frames and temporal synchronization between heterogeneous sensors.

Additionally, the construction of collaborative features specifically captures two robot sensor fusion patterns. The framework uses high-precision Vicon Motion Capture System (Vicon) as ground truth in realistic warehouse scenarios. This ground truth is utilized for coordinate transformation and labeling mmWave radar point clouds from two autonomous robot platforms.

A systematic assessment of six different GNN architectures establishes clear performance hierarchies for collaborative robot occupancy prediction. This work compares two families of network architectures, Graph Attention Network version 2 (GATv2), and Edge-Conditioned Convolution (ECC). They are evaluated across two temporal configurations that involve sliding window of size 3 and 5.

The experimental verification includes 34 recording sessions that produce collaborative graph frames. These frames extend over three warehouse layouts, constituting a complete dataset for collaborative graph-based perception. GNNs can learn authentic dual-robot sensor fusion patterns instead of single-robot observations. This is achieved through novel collaborative features that quantify individual robot contributions within shared spatial voxels.

Key contributions are the creation of a preprocessing framework for collaborative robotics research that can be used in further research, and the systematic architectural evaluation of GNNs for collaborative perception. The successful demonstration of mmWave radar capabilities provides a strong basis for 6G ISAC paradigms, placing collaborative robotics at the core of next-generation autonomous warehouse systems.

Acknowledgements

I extend my heartfelt gratitude to all those who have contributed to the successful completion of this thesis on Development of a Framework for Collaborative Perception Management Layer for Future 6G-Enabled Robotic Systems.

First and foremost, I would like to express my profound appreciation to my supervisors, Prof.'in Dr.-Ing. Alice Kirchheim and Irfan Fachrudin Priyanta, M.Sc., whose exceptional guidance and mentorship have been instrumental throughout this research journey. Their expertise in collaborative robotics and warehouse automation provided invaluable insights that shaped the direction and quality of this work. The constructive feedback during our regular meetings significantly enhanced the rigor and clarity of this thesis, while their encouragement to explore innovative approaches to Graph Neural Network architectures has been truly inspiring.

I am deeply grateful to the Chair of Material Handling and Warehousing (FLW) at TU Dortmund University for providing me with this remarkable research opportunity and access to state-of-the-art facilities. The experimental infrastructure, including the Robomaster platforms, mmWave radar sensors, and Vicon Motion Capture (MoCap) system, enabled the comprehensive data collection and validation that forms the backbone of this thesis.

Special thanks go to my family for their unwavering support and understanding throughout the demanding periods of this research. Their encouragement during challenging phases of data processing and model development provided the emotional strength necessary to persevere through complex technical obstacles. I am also grateful to my friends who provided valuable perspectives and helped clarify complex concepts through constructive discussions. This thesis represents not only my individual effort but also the collective support and contributions of all these remarkable individuals and institutions.

Contents

1 Introduction

The world of logistics has changed significantly in recent times. In particular, warehouses have emerged from being simple areas to store things to being complex automated hubs that are the backbone of global supply chains. These contemporary structures use Cyber-Physical Production Systems (CPPS) to seamlessly combine digital intelligence with physical processes. This makes it possible for a swarm of Autonomous Mobile robots (AMR) to move around in complicated spaces. These include high-bay storage spaces, mobile workstations, and places where humans work with machines [1]. This evolution is a big step away from traditional manual operations and towards fully autonomous logistics systems that must meet efficiency standards in more complex environments.

For years, warehouse automation has primarily depended on single-agent robotic systems performing independently in these intricate environments. These successful systems cut down on harmful warehouse practices by using sensor suites made up of cameras, Light Detection and Ranging (LiDAR), and radar technologies. These allow individual robots to create comprehensive environmental models by performing advanced sensor fusion techniques [2]. The core architecture of these systems is centered on autonomous decision-making. Each robot navigates using only its own perception abilities and processes sensory data locally, negating the need for coordination or information exchange with other robots [3]. This method has worked well in controlled settings with predictable layouts and little fluctuation in the surroundings.

Nevertheless, the operational requirements of modern warehouse settings have started to highlight notable drawbacks of single-agent strategies. Modern facilities have high-bay configurations that produce complicated occlusion patterns and increasingly narrow aisles designed to maximize storage density. These environments also experience frequent layout changes to accommodate shifting inventory requirements, along with the continuous movement of employees, machinery, and goods [4]. Individual robotic systems that rely on their sensor capabilities to perceive and react to rapidly changing environments face challenges due to these dynamic conditions. Robots deal with more complexity when they come across situations that need them to quickly identify and avoid obstacles. These include both static infrastructure components, such as shelving units, and dynamic elements, like moving individuals and machinery.

When analysing the performance characteristics of individual sensor technologies under demanding warehouse conditions, the limitations become specifically evident. While cameras offer detailed visual information, they can struggle in changing lighting, blocked areas, and environments with repetitive patterns that impact depth accuracy. Even though LiDAR sensors typically perform reliably, they encounter difficulties in environments with dust, a common issue in warehouse settings. These sensors also struggle in areas with highly reflective surfaces that can generate incorrect readings. Additionally, in heavily congested spaces, interpreting and processing dense point cloud data efficiently becomes problematic [5]. The isolated sensing

capabilities of individual robots are insufficient to compensate for these sensor-specific vulnerabilities, which result in critical blind spots and perception gaps.

These basic limits of single-agent methods have led to the emergence of the collaborative perception approach. This new method enables robots to collect and share sensory data. By working together, they form a much more complete model of their environment, which goes beyond the capabilities of any individual agent [6]. This approach combines sensory data from various platforms. As a result, it shifts the perception problem from individual sensor limitations to distributed sensing capabilities. This results in a single environmental model with significantly fewer blind spots and improved situational awareness [5]. The collaborative framework makes use of the idea that information exchange between robotic agents can get around the limitations imposed by independent sensors and computational resources.

1.1 Technological Context and Challenges

The effectiveness of collaborative perception fundamentally depends on two critical technological components: robust sensing capabilities and a reliable communication infrastructure. Although conventional sensors, such as cameras and LiDAR, have proven effective in controlled environments, they face limitations in the real-world setting of a warehouse. Cameras have trouble with different lighting and obstacles, while LiDAR systems face challenges with dust, shiny surfaces, and the heavy clutter found in warehouses. On the other hand, mmWave radar sensors have shown great potential for warehouse use as a proof-of-concept for 6G [4]. They provide strong sensing abilities even when optical and laser-based sensors do not work well. Operating in the 60-64 GHz frequency band for industrial purposes, mmWave radar delivers dependable performance in tough situations such as dust, smoke, changing lighting, and partial blockages.

Collaborative perception can only reach its full potential with the right infrastructure for communication. Early wireless technologies such as Wireless Fidelity (Wi-Fi) have proven insufficient for the demanding requirements of real-time coordination among robotic platforms in dense warehouse environments [4]. While 5G technology has made major strides in latency, bandwidth, and reliability, it still falls short of meeting the Ultra-Reliable Low-Latency Communication (URLLC) standards. These standards are essential for safety-critical robotic applications in fast-changing warehouse settings. The arrival of 6G technologies aims to solve these communication challenges with improvements in wireless networking. Beyond just enhancing 5G specifications, 6G introduces groundbreaking ideas like ISAC, which combines sensing and communication into a single framework [7]. This is especially important for mmWave-based systems, where the same hardware can handle both high-resolution sensing and high-bandwidth communication. This dual capability allows for effective collaborative perception in tight spaces [8].

1.2 Problem Statement

Critical gaps remain in the development of efficient collaborative perception systems specific for warehouse robotics applications. This is despite notable advancements in individual robotic

perception capabilities and the encouraging potential of collaborative approaches.Current collaborative frameworks, which were mainly created for automotive applications, do not take into account the particular difficulties that arise in indoor warehouse settings. These difficulties include the need for constant adaptation to shifting inventory layouts and operational patterns. Additionally, they encompass complex three-dimensional storage structures, higher agent density requirements, and frequent occlusions brought on by storage infrastructure [9]. Current frameworks fall short in addressing the fundamental differences between indoor warehouse operations and outdoor automotive scenarios, which call for specialised approaches.

Integrating mmWave radar technology into collaborative robotic systems for warehouse applications is an area not explored very much in current research. Current radar processing methods majorly focus on single-platform applications. As a result, they do not address the challenges of merging sparse radar point clouds from multiple mobile platforms with varying perspectives, motion dynamics, and temporal synchronization requirements. To enable the practical deployment of radar-based collaborative perception systems in warehouse environments, this presents a significant research gap that needs to be solved.

Existing perception approaches generally treat sensing, communication, and decision-making as separate problems. This division misses opportunities for cross-layer optimization, which could greatly increase the overall system performance. ISAC technologies in 6G offer a potential pathway to enable such integration. However, the development of practical frameworks that make effective use of these capabilities in the context of warehouse robotics remains undeveloped [10].

1.3 Research Objectives and Contributions

This thesis addresses these challenges by introducing a structured framework for collaborative perception in warehouse robotics. It leverages the complementary strengths of mmWave radar sensing and emerging 6G communication technologies. The ultimate objective is to bridge the gap between current collaborative perception frameworks and the unique operational demands of modern warehouse environments.

The specific research objectives are:

1. Conducting a comprehensive analysis of state-of-the-art collaborative perception frameworks functions and identifying major limitations in their application in warehouse automation, particularly through mmWave radar technology.
2. Designing a novel preprocessing pipeline to transform raw sensor readings across different robotic platforms to structured representations suitable for collaborative perception algorithms.
3. Developing and validating GNN architectures specially designed for spatial reasoning in collaborative robotics applications.
4. Building a perception dataset from the Robomaster platform mounted with mmWave realistic warehouse settings.
5. Formulating a comprehensive evaluation framework that measures model performance across different dimensions like spatial accuracy and temporal consistency.

The core contributions of this thesis are the creation of a preprocessing pipeline that addresses the intrinsic challenges in the radar data fusion from different robotic platforms, such as temporal synchronization, coordinate conversion, and data labeling requirements. The thesis proposes the systematic analysis of GNN architectures for collaborative robot occupancy prediction. It compares six different models to establish absolute performance hierarchies and architectural principles.

1.4 Thesis Organization

The rest of this thesis is organized to provide a detailed review at collaborative perception in warehouse robotics. Chapter 2 lays out the basic theoretical background, covering mmWave radar principles, platform specifications, and GNNs. Chapter 3 offers a complete review of the latest developments in collaborative perception, warehouse automation, and related technologies. It also highlights key research gaps that creates a foundation for this work. Chapter 4 details the proposed methodology, which includes a thorough preprocessing pipeline and the design of the Graph Neural Network models created for this application. Chapter 5 describes the experimental setup, data collection methods, and implementation details. It also includes an evaluation of results that features performance comparisons, spatial analysis, and insights on the architecture. Finally, Chapter 6 wraps up the thesis by summarizing the main contributions and suggesting future research directions that could build on the foundations established here.

This research marks an important step toward achieving fully autonomous warehouse operations through collaborative perception technologies. It offers both theoretical insights and practical solutions for using coordinated robotic systems in complex industrial settings. By focusing on the key areas of sensing, communication, and machine learning technologies, this work contributes to the larger goals of Industry 4.0. It also aids in the development of smart, flexible manufacturing and logistics systems that can handle the needs of modern global supply chains.

2 Fundamentals

This chapter presents the basic theories that are crucial for comprehending how collaborative perception works in modern robotic systems. The foundations include mmWave radar sensing concepts and GNN structures, which serve as a basis for the methodological contributions in later chapters. Understanding these fundamental ideas is essential for creating collaborative perception frameworks. These frameworks must perform well in dynamic warehouse settings, where conventional sensing modalities frequently encounter major difficulties.

2.1 mmWave Radar Sensing Fundamentals

mmWave radar systems, particularly that works in the 60-64 GHz bands, have become vital to technologies for working in industrial and robotic applications [11]. Unlike optical sensors, that struggle under varying lighting conditions, and LiDAR systems, which are affected by dust and reflective surfaces, mmWave radar offers a robust and reliable sensing foundation. This makes it a strong candidate for robotic applications in harsh or unpredictable environments [12]. The capability of the sensor to simultaneously measure range, velocity, and angle through dust, smoke and other visual blockages makes mmWave radar an ideal sensor for autonomous warehouse and industrial applications.

The Frequency Modulated Continuous Wave (FMCW) radar architecture has been common in this sector due to its high range resolution and cost-effectiveness in terms of computation. Compared to pulsed radar systems, FMCW radar continuously sends electromagnetic waves, which enable accurate measurements at modestly low power consumption and fewer hardware requirements. This continuous process enables the possibility of extracting rich environmental information through sophisticated signal processing techniques that transform raw electromagnetic reflections into consumable perception data [13].

2.1.1 Range Measurement

The fundamental principle underlying the operation of FMCW radar involves the transmission of frequency modulated continuous waves. Additionally, it includes the analysis of the received signals that are returned from the targets to extract information about the targets. The signals that are transmitted in FMCW systems are continuous waves whose frequency varies linearly with time. These signals are referred to as chirp signals. By measuring the frequency differences between transmitted and received signals, this linear frequency modulation allows the radar to determine the target's range [13]. Figure 2.1 shows the complete chain of the chirp generation through range-Doppler-angle processing.

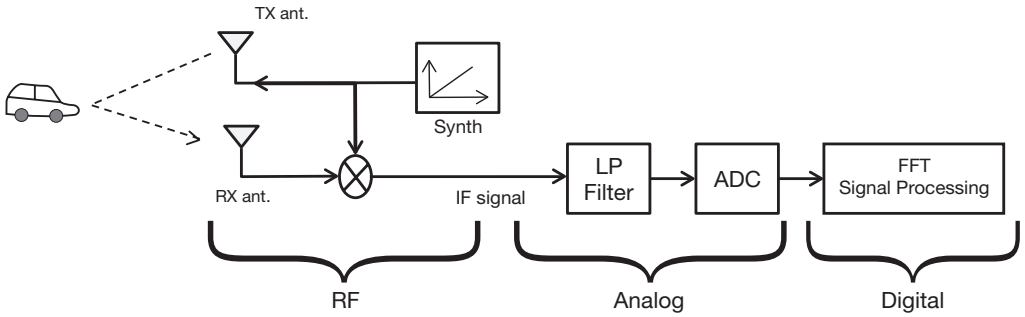


Figure 2.1: FMCW radar signal processing pipeline showing transformation from analog RF signals through digital processing stages [13]

The mathematical foundation of FMCW radar starts from the transmitted chirp signal, whose instantaneous frequency varies linearly over the chirp duration. This frequency-time relationship can be detailed as:

$$f(t) = f_c + \frac{B}{T_c} \cdot t \tag{2.1.1}$$

where f_c is the starting frequency of the chirp, B indicates the total bandwidth swept during the chirp duration, and T_c is the time duration of a single chirp. The linear term $\frac{B}{T_c}$, often denoted as the chirp slope S, directly affects the radar's range resolution capabilities and establishes the rate of frequency change.

When this chirped signal strikes an object that is at a distance d and travels back to the radar receiver, it has experienced a round-trip propagation delay. This round-trip time delay, the foundation of range measurement, is calculable as:

$$\tau = \frac{2d}{c} \tag{2.1.2}$$

where c is the speed of light in the propagation medium. This delay appears as a difference in frequency between the instantaneous frequency of the transmitted signal and the frequency of the received echo at any specified moment in time.

The radar receiver's mixing process brings together the transmitted and received signals, yielding an Intermediate Frequency (IF) signal whose frequency reflects the range to the target. For stationary targets, this beat frequency remains constant throughout the chirp duration and is expressed by the following equation.

$$f_{\text{IF}} = \frac{2Sd}{c}, \quad \text{where } S = \frac{B}{T_c} \tag{2.1.3}$$

The target range can be calculated directly from the measured beat frequency by manipulating this relationship algebraically:

$$d = \frac{f_{\text{IF}} \cdot c \cdot T_c}{2B} \tag{2.1.4}$$

The FMCW radar system's range resolution, which defines its ability to discriminate between two narrowly spaced targets in the range dimension, is fundamentally constrained by the transmitted chirp's bandwidth. This resolution is stated by:

$$\Delta R = \frac{c}{2B} \tag{2.1.5}$$

The inverse relationship between bandwidth and range resolution underscores a vital design consideration in FMCW radar systems. Fine range resolution is obtained by larger bandwidth allocations, which can be restricted both by regulatory limitations as well as hardware limitations [11]. Modern mmWave radar systems typically operate with bandwidths in the several gigahertz range to facilitate centimeter-scale range resolution. These systems are well-suited to the task of detailed environmental mapping and obstacle detection in robotics applications. However, with the increasing spread of radar systems in automotive and industrial applications comes the problem of mutual interference between co-located radar systems operating on the same frequency bands [14].

2.1.2 Velocity and Angle Measurement

By using the Doppler effect, FMCW radar systems are excellent at measuring target velocity in addition to range determination [15]. When moving targets reflect electromagnetic waves, the reflected signal's frequency shift is caused by the relative motion. The target's radial velocity component in relation to the radar can be directly measured through this Doppler frequency shift.

The Doppler frequency shift for a target travelling at radial velocity v is as follows:

$$f_d = \frac{2vf_c}{c} \tag{2.1.6}$$

where f_c is the carrier frequency of the radar signal. This frequency shift results from the compression or expansion of the electromagnetic waves caused by the relative motion between the radar and the target.

In real-world FMCW applications, phase analysis of several consecutive chirps is used to measure velocity instead of direct frequency measurement. For velocity estimation, the phase difference between subsequent chirps offers a more accurate and computationally effective approach. The following is an expression for this phase progression:

$$\Delta\phi = \frac{4\pi v T_c}{\lambda} \tag{2.1.7}$$

where $\lambda = c/f_c$ is the wavelength of the radar signal. This approach based on phases allows for very precise measurements of the velocity even for targets that are moving slowly.

The velocity estimation formula can be obtained by rearranging the phase relationship:

$$v = \frac{\lambda\Delta\phi}{4\pi T_c} \tag{2.1.8}$$

The maximum velocity that can be measured is fundamentally limited by the discrete sampling of digital radar systems and the intrinsic 2π periodicity of phase measurements. This constraint on velocity ambiguity is provided by:

$$v_{\max} = \pm \frac{\lambda}{4T_c} \tag{2.1.9}$$

This constraint presents a compromise between velocity resolution and maximum velocity that must be carefully weighed in system design based on the expected target velocity ranges in the application environment.

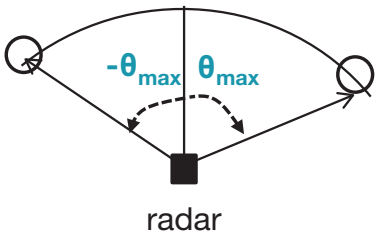


Figure 2.2: Maximum angular Field of View (FoV) determination [13]

Angle estimation for FMCW radar systems makes use of multiple receive antenna elements phased in specific geometric configurations to determine the direction of arrival of reflected signals [16]. The majorly used configuration employs a Uniform Linear Array (ULA) where antenna elements are positioned at equal intervals in a straight line. As the electromagnetic wavefront with a specific angle θ to the array normal approaches the antenna array, it encounters variations in path lengths. Consequently, the wavefront arrives at the different antenna elements at slightly different instances.

The difference in path length between neighboring antenna elements produces a measurable phase shift that can be used to estimate the angle. For a ULA with element spacing d, the phase difference between elements is given by:

$$\Delta\phi_{\text{ant}} = \frac{2\pi d \sin(\theta)}{\lambda} \tag{2.1.10}$$

The angle of arrival can be estimated using the inverse relationship by measuring this phase difference across the antenna array:

$$\theta = \arcsin\left(\frac{\lambda \Delta\phi_{\text{ant}}}{2\pi d}\right) \tag{2.1.11}$$

The number of antenna elements N, their spacing, and the Signal-to-Noise Ratio (SNR) are some of the variables that affect the radar system's angular resolution, which establishes its capacity to discriminate between targets at various angles. One way to approximate the theoretical angular resolution is as follows:

$$\Delta\theta \approx \frac{\lambda}{Nd\cos(\theta)} \tag{2.1.12}$$

The relationship shows that angular resolution gets better with larger antenna apertures (through more elements or wider spacing) and gets worse at larger off-boresight angles where $\cos(\theta)$ term becomes significant.

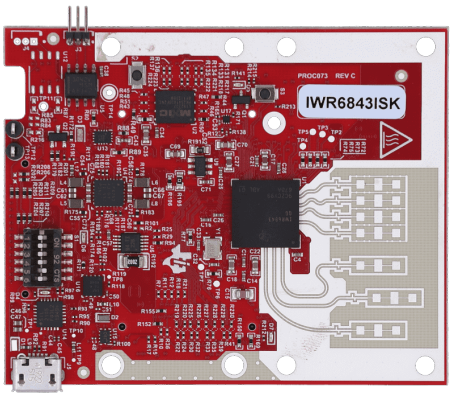


Figure 2.3: Texas Instruments IWR6843 ISK Module [17]

Current mmWave radar modules, including the Texas Instruments IWR6843 shown in Figure 2.3, are made with compact parts that integrate antennas for transmitting and receiving signals and do sophisticated signal processing [13]. Complex Multiple-Input Multiple-Output (MIMO) radar configurations that improve angular resolution and allow elevation angle estimation using virtual antenna array techniques are made possible by these integrated solutions [16].

2.1.3 Radar Signal Processing and Point Cloud Generation

Sophisticated digital signal processing pipelines that extract valuable information from complex electromagnetic returns are necessary to convert raw radar measurements into actionable perception data [15]. To extract range, velocity, and angular information, the processing chain begins with digitising the received IF signals and then moves on to spectral analysis.

Range extraction begins and ends with the use of Fast Fourier Transform (FFT) operations on the digitized signals. Before the FFT generates the range profile, which has each frequency bin mapping to a specific range via Equation 2.1.4, windowing functions such as Hamming or Blackman are used to reduce spectral leakage. The result, range-FFT, is computed for each chirp to form a range-time matrix.

The range-Doppler map is obtained by performing Doppler-FFT across multiple chirps for each range bin in order to extract velocity [13]. The number and spacing of chirps determine the velocity resolution, which strikes a balance between update rates and precision.

Angular information takes advantage of having multiple receive antennas. Phase differences across the array are used with beamforming methods, such as Bartlett, Capon, or Multiple Signal Classification (MUSIC), to estimate angles [16].

Stages of this process yield radar point clouds that encode the 3D positions and velocities of the targets being tracked. Each point in the point cloud contains Cartesian coordinates (x, y, z).

These coordinates are derived from spherical coordinates (r, θ, ϕ) as:

$$x = r\sin(\theta)\cos(\phi) \tag{2.1.13}$$
$$y = r\sin(\theta)\sin(\phi) \tag{2.1.14}$$
$$z = r\cos(\theta) \tag{2.1.15}$$

Advanced methods such as Constant False Alarm Rate (CFAR) detection adaptively set thresholds for robust detection. Maintaining target identities is the job of multi-target tracking, and machine learning models refine classification and reduce false alarms [15, 18].

Careful temporal and spatial alignment is required in sensor fusion, owing to the differing update rates and degree of sparsity in comparison to LiDAR [18, 16]. Real-time robotic applications impose very strict requirements that dictate the use of hardware accelerators such as Field-Programmable Gate Array (FPGA)s and Digital Signal Processor (DSP)s. A device such as the Texas Instruments IWR6843 integrates dedicated accelerators for tasks such as FFT computation. This enables the instrument to produce point clouds at a sufficiently high rate and low power, making it useful for dynamic navigation [13].

2.2 Graph Neural Network Concepts

GNNs are a robust class of neural networks that operate on data structured as graphs. They recover dependencies and relationships in such data. This capability has pushed the state of the art forward in many different domains, including social network analysis [19], recommendation systems [20], molecular chemistry [21], and computer vision. GNNs are one of the most powerful and effective tools we have today for turning graph-based data into useful insights [22].

Conventional convolutional networks typically operate on standard grid formations, such as images. In contrast, GNNs are purpose-built for irregular, non-Euclidean data structures. These include radar point clouds, sparse sensor readings, or closely packed arbitrary spatial arrangements of objects.That makes GNNs particularly good candidates for addressing robotic perception challenges, in which the relevant data naturally consists of interconnected entities rather than rigid grids [23].

2.2.1 The role of GNNs in advancing Collaborative Perception

GNNs have novel strengths for collaborative perception in their native representation of spatial relations and ability to handle non-uniform data structures. Radar point clouds, voxelized or clustered, naturally form sparse, non-uniform graphs wherein vertices are spatial regions and edges specify proximity relations [24]. The graph-based representation allows GNNs to operate natively on such structures without loss of fundamental geometric and topological information which would be sacrificed in the conventional grid-based approach. Differing from typical convolutional architectures designed for uniformly structured grid data from cameras, GNNs are well-equipped to accommodate the naturally sparse and non-uniform distribution of radar sensor data. They can learn informative spatial features without dense, uniform scanning of the entire sensing space [23].

GNNs' contextual reasoning capability stems from their sophisticated message passing schemes that enable each encoded spatial region as a node to summarize and process information from its neighborhood. This environmental knowledge is critical to differentiate between valid objects and ambient noise and define proper boundaries among filled and unoccupied space in complex warehouse environments. The paradigm of message passing inherently provides scalability in dynamic environments where the number and spatial arrangement of observed objects can change continuously. This is because the graph structure can dynamically change for various numbers of nodes and edges without undergoing architectural modifications [25].

In collaborative systems, GNNs show a special ability to accommodate information from many different sensor sources across several robotic platforms. When we treat the radar voxels and spatial observations from different robots as nodes in a single graph, GNNs let us perform collaborative spatial reasoning that extends well beyond any one robot's limited viewpoint. This becomes especially valuable in places like warehouses. In these environments, robots must coordinate their perception in order to navigate around dynamic obstacles and other autonomous agents while maintaining accurate global maps [12, 26].

The majority of GNN architectures are based on the principle of message passing. This framework provides a unified understanding of how GNNs learn node representations through the iterative exchange of information. The core notion is that each node iteratively collects information from its direct neighbors and combines it with its own features to reformulate its representation. In order to create complex spatial and contextual representations, this process is repeated across several layers, enabling nodes to collect data from progressively more distant parts of the graph [27].

2.2.2 General Mechanisms in GNNs

While the paradigm of message passing is universal, GNNs use many clever mechanisms to process both spatial and contextual information in systems where collaborative perception occurs. A key development is adaptive neighbour weighting, in which GNNs dynamically determine the relative significance of each neighbor's contribution to the feature update of a particular node [28]. In warehouse environments, some spatial relationships are more important than others. This adaptive mechanism allows the network to selectively focus on the most relevant spatial regions or features, rather than giving equal weight to all nearby nodes. GNNs can discriminate between less significant environmental noise and important spatial features, like robot-to-robot proximity relationships. This discrimination is a result of their capacity to learn these attention patterns.

The relationship features incorporation represents another vital mechanism through which GNNs use the attributes of the connections between nodes to make spatial reasoning more capable. The attributes of edges can include geometric properties like the distance between nodes, relative angles, or even semantic relationships like collaborative robot interactions [25]. GNNs formally encode these edge properties directly into the message passing model. Such direct relationship feature modeling allows GNNs to learn high-level spatial dependencies that would be difficult for node features alone. This is particularly true in systems in which geometric and collaborative relations among perception regions hold critical information for successful occupancy prediction.

Learning to represent the hierarchy of information with multi-layer GNN architectures allows us to progressively abstract the representations of spatial information. It enables a transition from local patterns to a global understanding of the environment. Initial layers usually capture fine-grained local spatial patterns and immediate neighbour relationships. In contrast, deeper layers integrate information over increasingly wider receptive fields to develop a more comprehensive understanding of the scene [27]. Effective coordination in complex warehouse environments is made possible by this hierarchical processing mechanism. It is crucial for collaborative perception tasks where robots must simultaneously reason about local spatial occupancy and the global environmental context [12]. These broad ideas serve as the conceptual foundation for comprehending how GNNs support spatial reasoning in collaborative perception systems. The next chapters will discuss in detail about particular architectural implementations of these mechanisms.

This chapter has set up the foundational theoretical elements that are necessary for grasping the idea of collaborative perception in collaborative systems. The principles of mmWave The principles of mmWave radar provide the sensor-level foundation for good perception of the environment. Meanwhile, the fundamentals of signal processing set up the real mathematical basis for extracting spatial and kinematic information from radar measurements. The GNN concepts provide the computational frameworks for doing spatial reasoning, with discussions on message passing and general mechanisms for processing irregular spatial data structures. All of these interdisciplinary elements i.e. the sensor physics, graph theory, and machine learning come together in this chapter. They create the comprehensive theoretical background that is necessary to understand the methodological innovations and experimental contributions of the following chapters.

3 State of the Art

This chapter discusses the current research state in Warehouse automation collaborative perception, with a focus on mmWave radar-based sensing and GNN approaches to environment understanding. The analysis highlights the prominent gaps between the existing automotive perception paradigms and warehouse specific demands, which serve as the foundation for the methodological contributions of this thesis.

3.1 Warehouse Automation

Contemporary warehouse automation has transformed from basic conveyor systems to intricate robotic setups where AMRs glide through dynamic conditions [1, 29]. Compared to other robotic domains, deploying robot swarms in warehouse settings poses special difficulties. These include high operational densities, frequent obstructions from storage infrastructure, and the requirement for constant adjustment to shifting inventory layouts.

Today's robotic systems predominantly use individual robot perception, accomplished with a variety of sensors, such as laser range finders, cameras, and ultrasonic sensors [6]. However, in crowded warehouse settings, these single-agent methods have serious drawbacks. The dependability of individual perception systems is greatly impacted by blind spots produced by tall storage racks.

Dynamic occlusions from other robots and human workers also affect these systems. Additionally, changing lighting conditions further compromise their reliability [5]. Research on collaborative perception frameworks, in which several robots combine their senses to create a single, cohesive representation of the environment, has been motivated by these constraints.

The integration of collaborative behaviors in warehouse robotics extends beyond perception to include coordinated path planning, task allocation, and swarm intelligence [30, 31]. Advances in distributed algorithms have enabled scalable coordination, but the perception layer remains a central bottleneck to achieving fully autonomous warehouse operations [32]. The issue is also compounded by the real-time nature of warehouse operations, where delays in perception or decisions propagate to cause significant losses in productivity.

3.2 Collaborative Perception Frameworks

Collaborative perception has basically developed as a transformative paradigm shift in multi-agent robotics. It is revolutionizing how autonomous systems approach environmental understanding by using robots to systematically transcend the inherent limitations of individual sensing capabilities through information sharing protocols and advanced multi-modal fusion techniques [33, 9]. The basic operational principle of these systems involves the systematic aggregation and integration of heterogeneous observations collected from multiple distinct viewpoints and sensor configurations. This process constructs a comprehensive and accurate environmental model that surpasses the perceptual capabilities that any single autonomous agent could achieve through independent operation. While utilizing the complementary strengths of distributed sensing networks, this collaborative approach helps to improve spatial coverage, temporal consistency and measurement redundancy. This way it effectively addresses key limitations that affect individual robot perception systems. These include challenges such as occlusion handling, sensor range constraints, field-of-view restrictions and vulnerabilities due to single points of failure.

Earlier, collaborative perception emphasized relatively simple and well-understood sensor fusion techniques. These took streams of raw measurement data from multiple robots of various kinds, combined them, and processed them using probabilistic methods like Kalman filtering, particle filtering, and Bayesian inference [34]. However, these foundational approaches faced scalability challenges and struggled with substantial bandwidth constraints. This highly increased computational complexity and communication overhead as the number of participating agents increased within the collaborative network.

When multiple robots work simultaneously, they often need to share high-dimensional raw sensor data. This includes point clouds, high-resolution images, and dense occupancy grids. Transmitting this data between robots quickly results in significant bandwidth limitations. Meanwhile, the computational load rises quickly as more robots are introduced. This is because each robot must communicate with every other robot, which causes pairwise communication overhead. In place of bandwidth-intensive raw sensor data streams, modern collaborative perception frameworks have systematically developed to incorporate intelligent communication protocols. They also include deep learning-based feature extraction algorithms and highly efficient transmission of semantically meaningful information representations [35, 36]. These approaches employ convolutional neural networks, attention mechanisms, and graph neural networks to extract compact, yet information-rich feature representations from learned models. These representations preserve critical perceptual information while exponentially reducing communication requirements.

With Vehicle-to-Everything (V2X) communication protocols, the automotive industry has spearheaded the development of collaborative perception [37]. These protocols allow connected autonomous vehicles to exchange real-time perception data, localisation information, and behavioural intentions for enhanced safety and coordination. Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Pedestrian (V2P), and Vehicle-to-Network (V2N) communication are all included in V2X systems, which build entire interconnected ecosystems.

Prominent systems like Cooper, F-Cooper, and DiscoNet [38] have shown that multi-vehicle collaboration protocols can improve detection accuracy by 15–30%. However, direct application to dense warehouse environments with high agent density, complex multi-level infrastructure, and unpredictable human-robot interactions is fundamentally problematic. This is because

automotive frameworks are optimised for outdoor environments with sparse agent distributions, predictable motion patterns, and defined infrastructure.

Current research has identified intermediate representation sharing, where robots interchange processed feature representations, compressed semantic maps and high-level abstractions instead of raw data streams [39, 40]. This method brings down communication overhead while keeping critical perceptual information. The When2com framework [39] brought in attention-based mechanisms for optimal communication timing and content selection based on uncertainty estimates and information gain metrics. V2VNet [40] utilized spatially aware graph neural networks for message passing and feature aggregation, facilitating selective incorporation of relevant collaborator information.

Even with technological progress, current frameworks are mainly aimed at automotive usages and do not have adaptation mechanisms for the spatial constraints specific to warehouses. They also lack support for the way they operate at high-density, for the patterns of dynamic obstacles that they have, or for their requirement to navigate across multiple levels. This necessitates the creation of collaborative perception architectures specific to warehouses.

3.3 mmWave Radar for Robotic Perception

mmWave radar technology has become a significant sensing method for robotic applications and offers distinct benefits over traditional sensors in difficult conditions [41, 42]. For industrial robotics, operating in the 60-64 GHz band, mmWave radar provides a robust performance when other methods like optical sensors or lidar struggles.

The application of mmWave radar in robotics has progressed from basic obstacle detection to complex mapping of environments and classifying objects within them [43, 44]. Recent technological advancements in radar signal processing have enabled the extraction of rich spatial information from radar returns in point cloud format, similar to LiDAR data. This also includes velocity information via Doppler processing [45]. The inclusion of this extra velocity information is particularly valuable within dynamic warehouse environments. In such settings, the isolation of static infrastructure from the moving agents becomes critical.

Deep learning methods have been successfully used for processing radar data. Convolutional Neural Network (CNN)-based models have shown great results in object detection and semantic segmentation tasks [43, 45]. However, most current research focuses on automotive applications, which have different operational needs than warehouse robotics. The high-resolution spatial data from modern MIMO radar systems, along with their ability to perform well in various environmental conditions, makes them suitable for collaborative perception in warehouses. This is especially important in environments where traditional sensors might have difficulties.

Integration challenges for mmWave radar in collaborative systems include aligning coordinate frames, synchronizing time and managing sparse point cloud data. Recent research has tackled some of these issues using calibration techniques and probabilistic data association methods [46, 47]. However, considerable work still needs to be done to apply these solutions to the scale and complexity of warehouse operations.

3.4 Graph Neural Networks for Spatial Reasoning

GNNs have revolutionized the processing of non-Euclidean data structures, offering powerful tools for reasoning about spatial relationships and dependencies [48, 19]. GNNs offer an ideal foundation for encoding environmental structures perceived by robots as graphs. In these graphs, the nodes denote spatial locales or entities, and the edges describe their connections [49, 25].

The shift from primitive graph convolutional networks [50] to advanced structures such as Graph Attention Network (GAT) [28] and their improved versions GATv2 [51] has given us more diverse and capable means. These advancements help in dealing with the spatial aspects of a problem. GATv2 in particular improves upon its predecessor by altering the attention mechanism it uses to be more dynamic and lay more emphasis on the qualities of the graph in problem-solving. This leads to using GATv2 in a more diverse set of spaces, such as in robotic scenarios.

ECC networks represent another significant advancement, explicitly incorporating edge attributes into the message passing process [52, 53]. This proficiency is especially pertinent for tasks of spatial perception in which critical information is conveyed through geometric relationships between entities. In a warehouse, edges can specify not just the identity of an obstacle, but also its position relative to the robot, distances, and potential collision risks between robots and obstacles.

In the field of robotics, GNNs have shown promise in successfully applying them to a number of tasks, from scene understanding [54, 55] all the way to multi-agent coordination [56]. Their ability to capture both local and global spatial patterns makes them a prime candidate for working with collaborative perception. In this setting, information from multiple viewpoints must be integrated in order for a coherent scene understanding to be achieved [57]. However, there are still a lot of gaps in the application of GNNs to collaborative perception in the real world. This is because the majority of current work concentrates on single-agent scenarios or assumes perfect communication.

3.5 SLAM and Mapping

Simultaneous Localization and Mapping (SLAM) in multi-agent systems has unique challenges beyond single-agent SLAM, requiring coordination of local maps, loop closure across robots, and handling of relative positioning uncertainty [58, 59]. The early centralized approaches of distributed SLAM algorithms have evolved into modern decentralized systems that allow for a large number of robots to work together [33, 60].

Current frameworks such as DOOR-SLAM [33] and Kimera-Multi [9] have shown themselves to be quite capable of distributed SLAM, working effectively in real time and handling outlier rejection quite well. The methods, however, are predominantly visual or LiDAR-based, with radar still largely in the prototypical stage. The integration of multi-agent semantic information into the SLAM process has been shown to have a very positive effect upon the data association problem. This, in turn, has shown great promise for increasing computational efficiency [61].

Multi-agent SLAM faces particular difficulties because of the distinctive features of warehouse environments such as dynamic obstacles, repetitive structures and Global Positioning System (GPS)-denied operation. Perceptual aliasing in symmetric environments and preserving consistent maps when inventory configurations change are frequent challenges for existing solutions. This brings in the need for specialised methods that make use of collaborative tactics and the complementary advantages of various sensing modalities.

3.6 Communication for Collaborative Robotics

The effectiveness of collaborative perception depends on the communication infrastructure. Traditional wireless technologies like Wi-Fi are not enough for the needs of robot coordination in busy warehouse environments [12]. Although 5G technology has made big improvements in latency, bandwidth, and reliability [62], it still does not meet the URLLC requirements for safety-critical robotic applications.

Emerging 6G technologies aim to fix these limitations by offering new features like ISAC, which combines sensing and communication into one framework [7, 8]. This combination is especially important for mmWave-based systems, where the same hardware can handle both high-resolution sensing and high-bandwidth communication [10]. The chance to optimize sensing and communication resources together creates new opportunities for effective collaborative perception in environments with limited bandwidth.

Research on communication efficient collaborative perception has found several strategies, including adaptive communication scheduling [39], learned compression of perceptual features [40], and priority-based information sharing [63]. However, these methods have mainly been assessed in vehicle settings, which have different spatial and temporal characteristics compared to warehouse environments.

3.7 Research Gaps and Opportunities

Despite important progress in individual research areas, gaps still exist in creating effective collaborative perception systems for warehouse robots. The recent collaborative perception frameworks meant for automotive use do not meet the specific challenges found in indoor warehouse settings. These include a higher density of agents, complex 3D structures, and frequent obstructions from storage equipment. The size and geometric complexity of warehouses need completely different methods for data association and map representation.

Although mmWave radar presents a powerful opportunity for warehouse sensing, its incorporation into collaborative systems is barely touched upon in the literature. Most of the signal processing for current mmWave radars focuses on single platform applications. Therefore, it does not address the quite different problem of fusing together point clouds from several mobile radar platforms that have different viewpoints and motion dynamics. Developing specialized algorithms for collaborative radar perception represents a rich research opportunity.

Current GNN architectures for spatial reasoning have not been fully assessed for collaborative robotic perception tasks. There are still questions about the best graph representations for sen-

sor data from multiple robots. We also need to consider the trade-offs between different GNN designs for real-time performance. Additionally, we should find ways to include temporal dynamics in graph-based spatial reasoning. The unique needs of warehouse environments, such as the ability to differentiate between static structures and moving agents, require new methods for building graphs and constructing features.

The integration of perception, communication and decision-making in collaborative systems lacks unified frameworks that can optimize these areas together. Existing methods often view these as separate issues. This leaves opportunities for optimization across different layers, which could enhance system performance. The rise of ISAC technologies brings in a possible way to achieve this integration. However, practical frameworks for warehouse robotics are still underdeveloped.

These research gaps motivate the development of specialized frameworks for collaborative perception in warehouse environments. These frameworks aim to use the advantages of mmWave radar sensing and GNNs while addressing the operational limitations of modern logistics. This thesis aims to address these gaps through a comprehensive approach encompassing preprocessing pipelines, GNN architectures dedicated to collaborative perception, and evaluation.

4 Methodology

This chapter lays out the comprehensive framework and algorithmic fundamentals for converting sensor data into graph structures fit for collaborative perception. The methodology addresses critical challenges in processing sparse, diverse sensor data in interconnected stages. These include raw data extraction and standardisation, multi-modal synchronisation, coordinate transformation, quality improvement, data labeling, graph generation, and GNN design. This chapter also introduces a multidimensional evaluation framework. This framework is designed to examine spatial reasoning abilities required for collaborative robotics.

The proposed methodology enables collaborative perception by addressing three major obstacles. First, it addresses temporal differences among sensors. Second, it addresses spatial inconsistencies between robot-centric coordinate frames. Finally, it turns sparse point clouds into organised formats suitable for Machine Learning (ML).

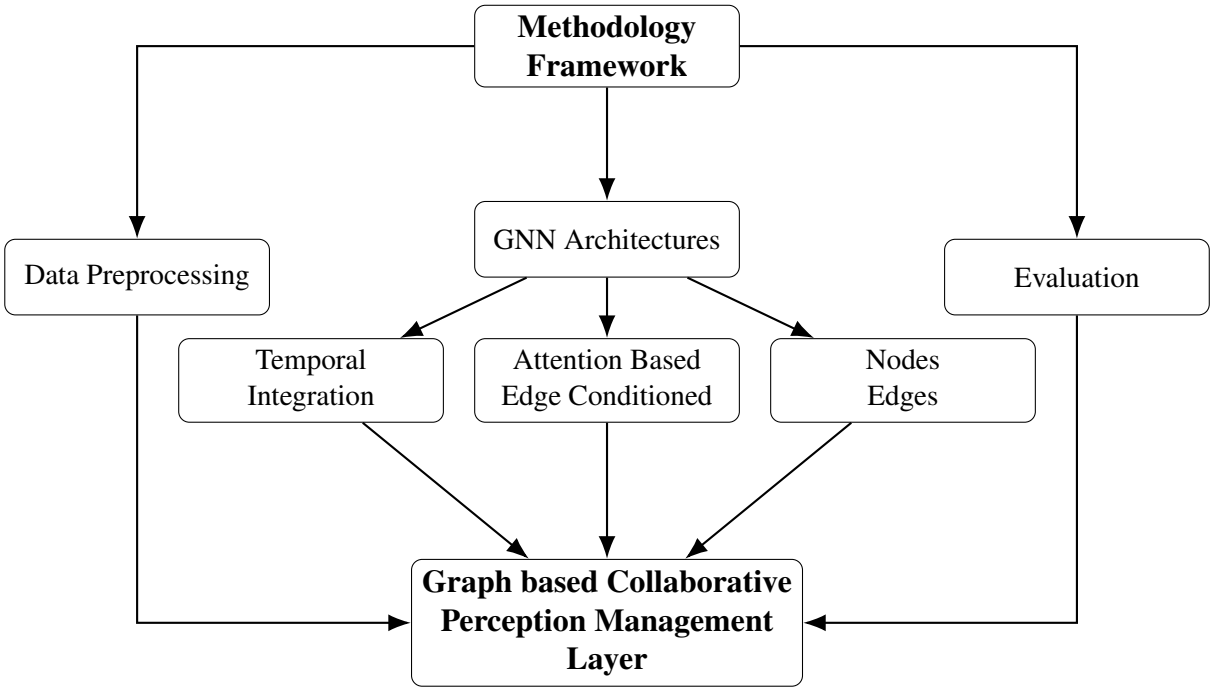


Figure 4.1: Methodology Framework for Collaboration Perception Management Layer (CPML).

4.1 Data Preprocessing Framework

This part presents the theoretical base for transforming raw sensor data from numerous robots into a unified, synchronized format. This format is ready for advanced processing, specifically for GNN application. This theoretical framework discusses raw data extraction principles, standardisation methods, synchronisation algorithms, coordinate system transformation mathematics, data quality enhancement strategies, data labeling frameworks, and graph structure generation theory.

4.1.1 Raw Data Extraction and Standardization

Heterogeneous raw sensor data are standardized during the preprocessing stage. The data are unified in terms of format and structure while maintaining the requisite spatial and temporal precision necessary for sensor fusion [64].

The Vicon MoCap system offers the high-frequency, high-accuracy ground truth positional and orientational data [65]. It creates a reference frame by analysing timestamped Degrees of Freedom (DoF) pose information. It can manage fluctuating object counts, missing data periods, and timestamp formats that are inconsistent. Temporal validation ensures monotonicity. Quality evaluation entails evaluating uncertainty during static phases and validating trajectories during motion.

mmWave radar sensors create point clouds that carry spatial coordinates, signal strength, and angle information [41]. The extraction process handles `sensor_msgs/PointCloud2` messages in Robot Operating System 2 (ROS2) [66], retrieving attributes such as range, SNR, and angle. High-precision timestamps from ROS2 headers and databases allow for seamless synchronization.

The data extracted is converted into CSV using a standard format: coordinates in meters, time in Unix epoch seconds, and angles in radians. Validation covers range checks, timestamp consistency and statistical outlier detection.

4.1.2 Data Synchronization

One of the difficult problems in multi-agent sensor fusion is temporal alignment. The timings used by each sensor system varies over time. These timing discrepancies must be rectified by the synchronisation technique. At the same time, it must maintain the connections within each data stream and guarantee the precision required for accurate collaborative perception [67].

From a methodological perspective, it is very important to identify and correct clock drift and offset problems. These problems occur because each sensor system keeps its own internal clock, leading to systematic but not uniform time differences between the sensors. While systems like Vicon work with accurate timing, robot internal clocks can shift in relation to the Vicon system and to one another. Delays in network communication and processing times create different time offsets that need to be considered during synchronization.

Another challenge is posed by variations in the sampling rate. Different types of sensor systems operate at different native frequencies and may experience variable data rates due to processing load or communication constraints. A Vicon system, for instance, might operate at a fixed, very high frequency, while certain types of radar systems might have variable update rates. The availability of data may also differ across types of sensor systems.

The synchronization approach adopts a multi-step process that systematically tackles each temporal alignment problem, while preserving data quality and temporal relationships. The core of this approach is described in Algorithm 1.

Notation used in Algorithm 1:

Symbol	Description
R_{D1}, R_{D2}	Raw radar datasets from robots 1, 2
V_D	Raw Vicon MoCap data
C_P	Configuration parameters
R_{F1}, R_{F2}	Temporally filtered radar data
V_R	Resampled Vicon data at fixed interval
S_R	Synchronized radar dataset
S_F	Final synchronized dataset
Δt_{fixed}	Fixed resampling interval
δt_{thresh}	Synchronization threshold
$t(r), t(v)$	Timestamp functions for radar/Vicon data
$t_{\text{span}}(V_R)$	Temporal bounds of Vicon data
$\arg\min$	Argument of minimum (nearest neighbor)
$\lvert\cdot\rvert$	Absolute difference operator

Algorithm 1 Multi-Modal Data Synchronization

Require: R_{D1}, R_{D2}, V_D, C_P
Ensure: Synchronized dataset S_F

Phase 1: Temporal Alignment

$V_R \leftarrow \text{Resample}(V_D, \Delta t_{\text{fixed}})$ ▷ Fixed interval resampling
$R_{F1}, R_{F2} \leftarrow \text{Filter}(R_{D1}, R_{D2}, t_{\text{span}}(V_R))$ ▷ Temporal bounds

Phase 2: Cross-Robot Synchronization

$S_R \leftarrow \emptyset$
for $r_i \in R_{F1}$ **do**
  $r_j \leftarrow \arg\min_{r \in R_{F2}} |t(r_i) - t(r)|$ ▷ Nearest neighbor
  if $|t(r_i) - t(r_j)| \leq \delta t_{\text{thresh}}$ **then**
    $S_R \leftarrow S_R \cup \{(r_i, r_j)\}$
  end if
end for

Phase 3: Vicon-Radar Alignment

for $(r_i, r_j) \in S_R$ **do**
  $v_k \leftarrow \arg\min_{v \in V_R} |t(r_i) - t(v)|$
  $S_F \leftarrow S_F \cup \{(r_i, r_j, v_k)\}$
end for
return $S_F = \{(r_1, r_2, v) : |t(r_1) - t(r_2)| \leq \delta t, |t(r_1) - t(v)| \leq \delta t\}$

The establishment of reference time starts with identifying the best available timing source among all sensing systems. In this instance, the Vicon system functions as the temporal reference, marking the master timeline T_V.

Vicon data resampling deals with the high and variable sampling frequency of MoCap data by resampling it to a consistent temporal grid $\{t_i\}_{i=1}^{N}$. This process uses interpolation techniques suitable for position $\mathbf{p}(t)$ and orientation $\mathbf{q}(t)$ data, such as forward-fill and linear interpolation. These techniques help maintain smooth trajectories while ensuring consistent temporal spacing. The choice of resampling frequency is a balance between time resolution and computational efficiency.

Radar data temporal filtering limits radar measurements to Vicon temporal bounds through $R_{F1}, R_{F2} = \{r \in R_{D1}, R_{D2} \mid t(r) \in [t_{\min}(V_R), t_{\max}(V_R)]\}$, ensuring all radar data has its corresponding MoCap information. Cross-robot radar synchronization aligns multiple radar streams by matching timestamp correspondence within tolerance δt_{thresh}, where $|t(r_i) - t(r_j)| \leq \delta t_{\text{thresh}}$, typically through nearest-neighbor temporal matching.

Vicon-radar temporal alignment synchronizes radar dataset S_R with Vicon timeline T_V using $s_f = (s_r, v)$, where $v = \arg\min_{v' \in V_R} |t(s_r) - t(v')|$, accounting for systematic time offsets Δt_{sys} between clocks. Movement detection identifies active motion periods by using velocity thresholds $v(t) = \| \frac{d\mathbf{p}(t)}{dt} \|$, with motion detected when $v(t) > v_{\text{th}}$. Activity segmentation divides sessions into distinct phases defined as Segment $= \{t \mid v(t) > v_{\text{th}}$ and other criteria$\}$, based on sustained motion, proximity or task execution phases.

4.1.3 Coordinate Transformation

To effectively fuse sensors across multiple platforms, collaborative perception systems must transform sensor measurements. These measurements, taken in different local coordinate frames, need to be unified into a single global reference system. This is done through the use of hierarchical reference frames, temporal synchronization and geometric relationships between sensors, robots and the global system.

The hierarchy of coordinate frames starts with the **Global Reference Frame**, given by the Vicon MoCap system, which serves as the spatial authority with a fixed origin and laboratory-aligned axes. **Robot Body Frames** are set for each robotic platform and are usually centered at the robot's geometric center, following standard robotics guidelines (X-forward, Y-left, Z-up). These frames move as robots navigate, with poses tracked continuously by MoCap. **Sensor Local Frames** are individual coordinate systems for the sensing devices mounted on the robots. For the radar sensors, the origin is defined at the sensor's phase center, and the axes align with the physical mounting and beam direction. Only the robot body frame is tracked by the MoCap system. The sensor local frame is related to the robot body frame by a fixed mounting offset along the axes.

The systematic transformation chain moves from the Sensor's Local Frame to the Robot Body Frame to the Global Reference Frame. It compensates for sensor mounting offsets and orientations, for the dynamic body poses from motion capture, and for temporal synchronization between measurements and pose estimates. The mathematical foundation changes a point P_{local} in sensor coordinates to P_{global} in the global frame. It does this through a series of rigid body transformations that account for both rotation and translation across different reference frames.

$$P_{global} = R_{robot} \cdot (R_{sensor} \cdot P_{local} + T_{sensor}) + T_{robot} \tag{4.1.1}$$

where P_{global} denotes the point coordinates in the global reference frame, P_{local} is the original point coordinates in the sensor local frame, R_{robot} and T_{robot} expresses the robot's orientation (3×3 rotation matrix) and position (3×1 translation vector) obtained from the MoCap system and R_{sensor} and T_{sensor} represent the sensor's orientation and position relative to the robot body frame, obtained by means of calibrating processes [68].

For collaborative perception systems that use rigidly mounted sensors, the rotational offset R_{sensor} can be pre-calibrated and integrated into the sensor measurements. Alternatively, it can be treated as identity for sensors aligned with the robot body frame. In such cases, the transformation equation simplifies to:

$$P_{global} = R_{robot} \cdot (P_{local} + T_{sensor}) + T_{robot} \tag{4.1.2}$$

This simpler form minimizes computing requirements while keeping transformation accuracy for systems with well-defined sensor mounting arrangements.

Computation of the rotation matrix requires converting orientation data from MoCap systems. This data is normally given in either Euler angles, axis-angle representations, or quaternion formats, and must be converted into proper rotation matrices suitable for coordinate transformations. Careful attention must be given to rotation order rules, such as ZYX and XYZ and sign rules. This is important for maintaining consistency throughout the transformation process. For robotic systems that primarily operate in planar environments, the transformation can be further simplified to 2D rotations that involve only the yaw angle:

$$R_{2D} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{4.1.3}$$

where θ is the yaw angle of the robot platform relative to the global reference frame. This 2D formulation greatly reduces the computational complexity for ground-based robotic applications. It keeps the spatial accuracy needed for successful collaborative perception tasks.

4.1.4 Data Filtering and Quality Enhancement

Real-world radar data contain more noises and many features that must be dealt with to achieve a reasonable level of reliable performance for ML. Radar-specific noise sources include thermal noise from receiver electronics and environmental clutter from static reflectors like walls and furniture. Multi-path reflections create ghost detections from indirect signal paths (Non-Line of Sight), and side-lobe contamination can lead to angular measurement errors. Additional challenges are from range and Doppler ambiguities, electromagnetic interference and motion-induced artifacts. These issues can highly reduce data quality if not handled properly.

A filtering pipeline tackles these challenges with the help of a series of specialized techniques. Operational Arena Filtering applies spatial bounding box limits based on known environment geometry to remove clearly incorrect measurements. SNR Filtering eliminates detections below

adjustable thresholds, typically between 6 and 15 dB, using either fixed or adaptive methods that depend on local noise analysis. Statistical Outlier Detection uses k-nearest neighbor analysis to find measurements that do not match their surrounding points. Points are marked as outliers when their average neighbor distance is more than a certain number of standard deviations above global statistics. Height-Based Filtering removes measurements outside relevant vertical ranges to get rid of ground reflections and ceiling artifacts. FoV Filtering limits detections based on documented sensor specifications, considering azimuth and elevation constraints and platform orientation for mobile systems. Temporal Consistency Filtering checks the persistence of measurements across consecutive frames to eliminate random false detections and includes motion checks to confirm reasonable velocity patterns.Implementation aspects include choosing the best order for filters to ensure efficiency. Additionally, memory-efficient chunked processing is crucial for handling large datasets. Careful parameter tuning through cross-validation is necessary to find the right balance between noise reduction and signal preservation.

4.1.5 Data Labeling and Ground Truth Generation

Generating accurate ground truth labels for supervised learning tasks like binary occupancy prediction is necessary. This stage outlines methods of using knowledge about the environment, geometric models, and robot positioning to automatically generate accurate labels.

The labeling of objects in this step falls into four main categories: specific workstations, robots, arena boundaries, and unknown objects. Ground truth generation uses known reference positions for workstations, robot locations from Vicon tracking, and predefined arena boundary coordinates. The method combines these static geometric models with real-time position data. It processes each timestamp independently, without relying on previous data.

The detection of workstations employs geometric models using predefined dimensions (1.0m $\times$ 0.65m) and configuration file positions. The method focuses on radar edge detection features through specific proximity testing with a tolerance of 20 cm. Point association uses first-match sequential testing for each workstation. Successful proximity matches decide the label assignment.

Current-frame Vicon positioning is used in the detection of robots and with it rectangular models (32cm $\times$ 24cm) can be created that have a tolerance parameter of 12cm. Robot coordination prevents self-detection but keeps the ability to detect other robots with the help of geometric boundary testing. The coordination mechanism works by processing each radar detection point against the known positions of both robots in the collaborative system. For any given timestamp, the system retrieves the exact global coordinates and orientation angles for Robot 1 and Robot 2 from the synchronized Vicon data. When evaluating whether a particular radar detection corresponds to a robot, the system tests the point against both robot positions using the geometric boundary testing algorithm. The critical insight is that Robot 1's radar data is being processed, so detections near Robot 1's own position are filtered out as self-detections. Meanwhile, detections near Robot 2's position are correctly labeled as legitimate robot detections. The system works with two robots at the same time and does not rely on motion prediction or temporal smoothing.

Boundary objects define arena limits as rectangular regions (x: -9.1 to 10.2m, y: -4.42 to 5.5m) with different tolerance settings: 25cm for straight edges and 35cm for corners. The classification involves direction and corner identification to understand complex radar reflection patterns.

The processing architecture implements a proximity to occupied points and then points which are still unknown. Dataset configuration management extracts workstation positions from Excel files. It uses pattern matching with backup methods for temporal association. Output generation produces two labeling columns, one for specific identifiers and another for general categorical labels.

4.1.6 Graph Structure Generation

GNNs relies on the data structured as mathematical graphs with nodes, edges, and associated feature vectors. In this stage, we convert labeled point cloud data from the collaborative robots into graph representations. These graphs retain the spatial relationships of the points while incorporating the sensor information and temporal dynamics.

A collaborative perception graph G is defined as:

$$G = (V, E, \mathbf{X}, \mathbf{A}) \tag{4.1.4}$$

where $V = \{v_1, v_2, ..., v_n\}$ is the set of nodes corresponding to spatial voxels having aggregated sensor data from both robots. The edge set $E \subseteq V \times V$ gives the spatial relationships between neighboring nodes, while the node feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ has d-dimensional feature vectors capturing spatial, collaborative and temporal information. The adjacency matrix $\mathbf{A} \in \{0,1\}^{n \times n}$ encodes graph connectivity based on spatial proximity relationships. Graph construction processes use pre-labeled point cloud data from both collaborative robots. They gather point-level labels during voxelization to create node-level classification information.

Table 4.1: Graph Component Specifications

Component	**Definition**	**Dimensions**
Nodes (V)	Spatial voxels containing radar points from one or both robots	$\lvert V\rvert \in [2, 500]$
Edges (E)	Fully connected topology: all nodes connect to all other nodes	$\lvert E\rvert = \frac{\lvert V\rvert(\lvert V\rvert-1)}{2}$
Node Features	[spatial_features(13), robot_counts(2), collaborative_score(1), temporal_offset(1)]	$\mathbb{R}^{17}$
Node Labels	Classes: 0=unoccupied, 1=occupied (Workstation, Robot, Boundary)	$\{0,1\}$
Adjacency Matrix	Symmetric binary matrix with complete connectivity (excluding self-loops)	$\{0,1\}^{\lvert V\rvert \times \lvert V\rvert}$

The voxelization procedure transforms the unstructured point cloud data from the two collaborative robots into a regular, unified 3D grid format. This format is necessary for subsequent processing with GNNs. Each radar point $p = [x, y, z]$ from either robot is mapped to voxel indices. This is done based on the chosen voxel resolution δ by using floor operations:

$$voxel\_x = \lfloor x/\delta \rfloor, \quad voxel\_y = \lfloor y/\delta \rfloor, \quad voxel\_z = \lfloor z/\delta \rfloor \tag{4.1.5}$$

The aggregation process usually combines the valid radar points from both robots in the same voxel structure. It keeps the individual contributions of each robot and creates a unified spatial representation. The graph representation employs spatial voxels as nodes, with each 0.1m³ voxel containing radar points from Robot 1, Robot 2, or both. The maximum node count of 2,500 per graph was established to accommodate the expected spatial density of 0.1-meter voxels containing radar points across the collaborative workspace. This limit also computational tractability for real-time graph neural network inference.

$$E = \{(v_i, v_j) \mid i \neq j\} \tag{4.1.6}$$

The fully connected edge layout creates complete graphs where each node links to every other node within a frame. This enables thorough modeling of spatial relationships and global message passing. Complete connectivity allows GNNs to capture local spatial patterns and long-range dependencies, both crucial for collaborative perception tasks. The enhanced representation of node features reflects both spatial relationships and collaboration patterns through 17 dimensional vectors. These vectors include normalized positions, raw coordinates, relative positions, distances, and novel robot collaboration features. Label aggregation works at the voxel level by using majority voting among the pre-labeled points in each voxel.

$$label(v_i) = \arg \max_{c \in Classes} |\{p \in v_i : label(p) = c\}| \tag{4.1.7}$$

Each node v_i is associated with a 17-dimensional feature vector capturing spatial and collaborative information:

$$\mathbf{x}_i = [\mathbf{s}_i^{spatial}, \mathbf{s}_i^{collaborative}]^T \tag{4.1.8}$$

$$\mathbf{x}_i = \left[x_n, y_n, z_n, x_g, y_g, z_g, x_{\text{rel}}, y_{\text{rel}}, z_{\text{rel}}, x_0, y_0, z_0, d_c, n_1, n_2, C_s, t_{\text{offset}}\right]^T \tag{4.1.9}$$

Here, x_n, y_n, and z_n are normalized coordinates bounded within $[0,1]$; x_g, y_g, and z_g represent the global coordinates in the world reference frame; x_{rel}, y_{rel}, and z_{rel} are relative coordinates measured from the frame center; x_0, y_0, and z_0 denote origin-based coordinates; $d_c = \|v_i - c^{\text{frame}}\|_2$ is the Euclidean distance from point v_i to the frame center; n_1 and n_2 are robot-specific point counts within the voxel; $C_s \in [0, 0.5]$ is the collaboration score metric; and t_{offset} indicates the temporal offset, which takes values in $\{0,1,2\}$ for Sliding window size 3 (T3) or in $\{0,1,2,3,4\}$ for Sliding window size 5 (T5).

The spatial feature component $\mathbf{s}_i^{spatial} \in \mathbb{R}^{13}$ includes several spatial representations. These consist of normalized position coordinates for scale-invariant spatial encoding and raw position coordinates for absolute spatial locations. They also include position relative to the spatial center for local context, position relative to the coordinate system origin for global reference, and distance to the spatial center as a measure of centrality. Collaborative features $\mathbf{s}_i^{collaborative} \in \mathbb{R}^3$ represent the unique aspect of this work. They quantify the contributions of individual robots within each spatial voxel. This includes features that count the number of radar points contributed by each robot. Additionally, it includes a collaboration score that measures spatial overlap between robots, calculated as the minimum contribution ratio of each robot within each voxel.

Table 4.2: Node Feature Definitions

Index	Feature	Description
0-2	Normalized coordinates	X, Y, Z normalized to frame bounds [0,1]
3-5	Global coordinates	Raw X, Y, Z in global reference frame
6-8	Relative coordinates	X, Y, Z relative to frame center
9-11	Origin coordinates	X, Y, Z relative to coordinate origin
12	Distance to center	Euclidean distance to frame center
13-14	Robot 1 & 2 point count	Number of points from Robot 1 & 2 in voxel
15	Collaboration score	Multi-robot collaboration metric [0, 0.5]
16	Temporal offset	Temporal sequence position (T3/T5 only)

Spatial features (indices 0-12) encode normalized position coordinates, raw spatial coordinates, position relative to the frame center and origin, and distance to the spatial center. Robot collaboration features (indices 13-15) measure the contributions of Robot 1 and Robot 2 within each spatial voxel. This allows GNNs to model cooperation patterns and spatial coverage redundancy. For temporal configurations T3 and T5, an additional temporal offset feature (index 16) indicates the causal position within past sequences. This keeps a strict temporal order for real-time deployment scenarios.

The data cleaning stage (Subsection 4.1.4) applied SNR filtering, field-of-view filtering, height-based filtering, and statistical outlier detection to ensure high-quality input data. By applying these radar-specific filters during preprocessing rather than incorporating them as node features, the framework separates data quality assurance from spatial reasoning. The focus on spatial and collaborative features ensures that the Graph Neural Network learns occupancy patterns based on spatial relationships. It also enables the network to understand collaborative coverage patterns rather than relying on sensor-specific signal characteristics. This approach enhances the generalizability of the framework across different radar configurations. It also maintains computational efficiency by concentrating on the most relevant features for spatial occupancy prediction tasks.

This graph architecture does not use edge features, instead relying on complete connectivity and comprehensive spatial information encoded in node features. In the fully connected topology, every spatial voxel connects to every other voxel, creating a symmetric adjacency matrix representing pure spatial connectivity. The Edge-Conditioned Convolution architecture leverages this design by learning adaptive edge transformations through concatenated node features. This approach allows dynamic computation of spatial relationships from the rich 17-dimensional node representations. This eliminates the need for explicit edge features such as Euclidean distances or geometric angles, as these relationships are inherently captured in node features and learned implicitly during message passing. The absence of explicit edge features enhances computational efficiency by reducing memory requirements and avoiding redundant spatial information. This allows the Graph Neural Network to focus on learning complex spatial and collaborative patterns through node feature interactions.

The adjacency matrix $\mathbf{A}$ uses a completely connected graph structure to capture spatial relationships between all the voxels in each frame. For each node v_i, connections are made to all other nodes:

$$A_{ij} = \begin{cases} 1 & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \tag{4.1.10}$$

This connectivity in its most complete form allows the preservation of spatial relationships between all pairs of voxels during processing by a GNN, ensuring comprehensive collaborative perception modeling across the entire spatial domain.

The temporal modeling of collaborative perception is done with the help of a sliding window that incorporates past and current time steps. The causal temporal window $W_t^{(w)}$ for a window size w is given as:

$$W_t^{(w)} = \{F_{t-w+1}, F_{t-w+2}, \ldots, F_{t-1}, F_t\} \tag{4.1.11}$$

For $w = 5$,

$$W_t^{(5)} = \{F_{t-5+1}, F_{t-5+2}, F_{t-5+3}, F_{t-5+4}, F_t\} = \{F_{t-4}, F_{t-3}, F_{t-2}, F_{t-1}, F_t\}, \tag{4.1.12}$$

which contains 5 frames.

In temporal window models, nodes get temporal offset features that tell their causal position in the historical sequence; they get relative indices, with negative values indicating historical frames and zero representing the current time step.

Figure 4.2: Point Cloud to Graph Conversion Pipeline.

Notation used in Algorithm 2:

Symbol	Description
$G = (V, E, \mathbf{X}, \mathbf{A})$	Collaborative perception graph
Pr^t	Radar point cloud from robot r at time t
$p \in \mathbb{R}^3$	Individual radar point coordinates
δ	Voxel resolution parameter
$vid = \lfloor p/\delta \rfloor$	Voxel identifier from point coordinates
V	Set of spatial voxel nodes, $\lvert V\rvert \in [2, 500]$
E	Set of edges, $\lvert E\rvert = \lvert V\rvert(\lvert V\rvert - 1)/2$
$\mathbf{X} \in \mathbb{R}^{\lvert V\rvert \times d}$	Node feature matrix
$\mathbf{A} \in \{0,1\}^{\lvert V\rvert \times \lvert V\rvert}$	Adjacency matrix
$\mathbf{x}_i \in \mathbb{R}^{17}$	Feature vector for node i
$\mathbf{s}_{\text{spatial}} \in \mathbb{R}^{13}$	Spatial feature components
$c_{\text{r1}}, c_{\text{r2}}$	Point counts from robots 1 and 2
$s_{\text{collab}} = \frac{\min(c_{r1}, c_{r2})}{c_{r1} + c_{r2}}$	Collaboration score
t_{offset}	Temporal sequence position
$\mathbf{y} \in \{0,1\}^{\lvert V\rvert}$	Binary occupancy labels
$W_t^{(w)}$	Temporal window of size w

Algorithm 2 Point Cloud to Graph Conversion

Require: Point clouds P_1^t, P_2^t, voxel resolution δ
Ensure: Graph $G = (V, E, \mathbf{X}, \mathbf{A})$ with labels $\mathbf{y}$

Phase 1: Spatial Voxelization

$V_{\text{map}} \leftarrow \emptyset$ ▷ Initialize voxel mapping
for $r \in \{1,2\}$, $p \in P_r^t$ **do**
  $v_{id} = \lfloor p_x/\delta \rfloor, \lfloor p_y/\delta \rfloor, \lfloor p_z/\delta \rfloor$
  $V_{\text{map}}[v_{id}] \leftarrow V_{\text{map}}[v_{id}] \cup \{(p, r, label(p))\}$
end for

Phase 2: Node Feature Construction

for $v_i \in V$ **do**
  $\mathbf{s}_i^{(spatial)} \leftarrow [\mathbf{p}_{norm}, \mathbf{p}_{raw}, \mathbf{p}_{rel}, \mathbf{p}_{origin}, d_{center}]$ ▷ 13D
  $c_{r1}, c_{r2} \leftarrow$ point counts from robots 1, 2
  $s_{collab} = \min(c_{\text{r1}}, c_{\text{r2}})/c_{\text{r1}} + c_{\text{r2}}$ if $c_{\text{r1}} + c_{\text{r2}} > 1$, else 0
  $\mathbf{x}_i \leftarrow [\mathbf{s}_i^{(spatial)}, c_{\text{r1}}, c_{\text{r2}}, s_{collab}]^T \in \mathbb{R}^{16}$
  $y_i \leftarrow \text{majority\_vote}(\{label(p) : (p, r, l) \in v_i\})$
end for

Phase 3: Graph Topology

$\mathbf{A}_{ij} = 1$ if $i \neq j$, else 0 ▷ Complete connectivity
$E = \{(i, j) : \mathbf{A}_{ij} = 1\}$, $|E| = |V|(|V| - 1)$
return $G = (V, E, \mathbf{X}, \mathbf{A})$ with $\mathbf{y} \in \{0,1,2,3,4\}^{|V|}$

Collaborative graph generation has computational complexity that scales with the total number of points from both robots during the voxelization process($O(|P_1| + |P_2|)$). It also scales quadratically with the number of resulting voxels during full connectivity construction ($O(|V|^2)$). The space complexity involves storing 17-dimensional node features ($O(|V| \times 17)$) and dense edge representations ($O(|V|^2)$).

A strong data segmentation strategy is essential for developing and assessing models. Preserving temporal integrity is vital for time-series datasets. Partitioning must be done at the level of experimental sessions or sequences to avoid data leakage. Data leakage can occur when temporally sequential measurements are divided between different splits (e.g., training and testing). The concept of experimental diversity distribution seeks to ensure that every partition (train, validation, test) includes representative samples of varied experimental conditions, behaviors, and setups. This approach helps to aid model generalization. The statistical balance requirements aim to maintain uniform statistical attributes across partitions. This includes balanced class distribution and comprehensive spatial/temporal coverage to guarantee an equitable evaluation.

4.2 Graph Neural Networks

This section discusses the detailed theoretical foundations of the GNN architectures that are suitable for collaborative perception tasks. It also details the specific architectures used in this research. The theoretical framework includes attention-based mechanisms, edge-conditioned convolutions, and design principles for spatial reasoning.

4.2.1 Graph Feature Representation

The theoretical foundations of graph feature representation for collaborative perception consist of node and edge design principles that encode spatial and temporal information within graph structures. Node features link spatial coordinates with voxel qualities to achieve precise localization. They include various information for context, collaborative context via relative positions among multiple robots, and temporal dependencies for dynamic modeling. These geometric grounds enable coordinated interpretation in shared environments.

Relational information is captured by edges through geometric relationships, encodings of distance and angle, and patterns of connectivity. Systematic edge formation, like that in k-nearest neighbor and radius-based approaches, is based on proximity criteria. These approaches also rely on theoretical frameworks of edge formation. To adapt to changing relationships in both spatial and temporal contexts and to edge formation in theoretical frameworks, dynamic edge weights are used. These weights reflect mathematically modeled relationships that consider both the relevant geometric constraints and the temporal evolution of the features.

Three fundamental paradigms, which offer complementary takes on spatial reasoning, form the basis of collaborative perception GNN architectures. The first paradigm is attention-based architectures. The neighbors are weighted dynamically and through learned mechanisms that establish a framework in which the neighboring information can be focused on. Edge-conditioned architectures include spatial relationships using edge-specific transformations. These transformations adjust computations. Temporal integration frameworks model changing relationships over time and how spatial configurations evolve. Together, these approaches effectively handle complex spatial and temporal dynamics in collaborative perception situations.

These theoretical foundations guide the selection and implementation of specific GNN architectures for collaborative perception, as detailed in the following sections.

4.2.2 Architecture Selection

The selection of specific GNN architectures for collaborative perception requires balancing computational efficiency, expressiveness, and spatial relationship modeling capabilities in dynamic warehouse environments.

Attention mechanisms, originally designed for tasks involving sequence-structured data (e.g., machine translation [69]), has now been successfully applied to graph-structured data. The main idea behind the mechanism is to allow each node to weight the importance of its neighbors differently during the process of information aggregation. This approach contrasts with treating all neighbors equally. This adaptive weighting provides a more subtle and powerful approach to capturing relevant local context.

GATv2 was selected over the original Graph Attention Networks (GAT) architecture due to fundamental improvements in attention mechanism expressiveness. Standard GAT computes static attention where ranking is unconditioned on the query node, limiting its ability to express simple graph problems. GATv2 implements dynamic attention through a modified scoring function: $e_{i,j} = \mathbf{a}^\top \text{LeakyRectifiedLinearUnit}(\text{LeakyReLU})\left(\mathbf{W}[\mathbf{h}_i \,\|\, \mathbf{h}_j]\right)$, enabling every node to attend to any other node [51].

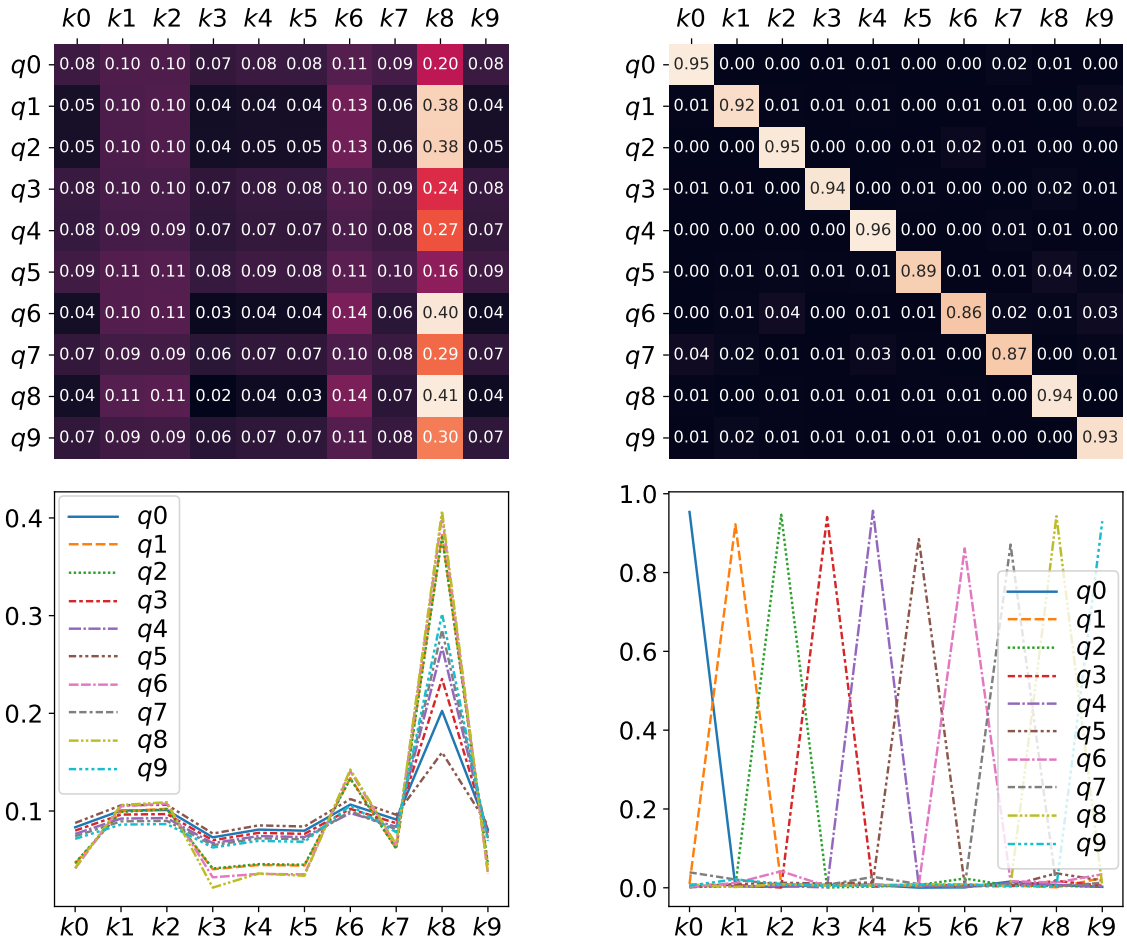


Figure 4.3: Attention mechanisms comparison: (a) Standard GAT with static attention, (b) GATv2 with dynamic attention [51].

The fundamental difference between the standard GAT and GATv2 is depicted in Figure 4.3. Within the experimental context of a complete bipartite graph, the standard GAT (Figure 4.3a) displays static attention characteristics. All query nodes show nearly uniform attention patterns towards a common group of key nodes. This constraint occurs because the attention ranking remains global and is not tailored to the unique features of each query node. Conversely, GATv2 (Figure 4.3b) supports dynamic attention, allowing each query node to generate its own distinct ranking of attention coefficients for the shared key nodes. This query-centric behavior permits spatial locations within collaborative perception scenarios to dynamically concentrate on relevant neighboring areas. This is done according to their individual spatial context and specific collaborative needs.

$$\alpha_{ij} = \frac{\exp\left(\mathbf{a}^T \text{LeakyReLU}\left(\mathbf{W}[h_i \parallel h_j]\right)\right)}{\sum_{k \in \mathcal{N}(i)} \exp\left(\mathbf{a}^T \text{LeakyReLU}\left(\mathbf{W}[h_i \parallel h_k]\right)\right)} \tag{4.2.1}$$

where α_{ij} is the attention weight that node i assigns to information from node j, h_i and h_j are the feature vectors of nodes i and j respectively, $\mathbf{W}$ is a learnable shared weight matrix, LeakyReLU is the non-linear activation function, $\mathbf{a}$ is a learnable attention vector, $\mathcal{N}(i)$ denotes the neighborhood of node i, and $\parallel$ represents concatenation [28].

The mechanism of multi-head attention expands this idea by performing multiple independent functions of attention in parallel. This allows the model to capture all kinds of spatial relationships at once. Each attention head can specialize in different aspects. This includes proximity-based attention for local neighborhood aggregation and for object-level reasoning. It also encompasses long-range attention for global context and directional attention for oriented spatial relationships. The multi-head formulation is given by:

$$h_i' = \text{Concat}_{k=1}^{K} \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^k \mathbf{W}^k h_j \right) \tag{4.2.2}$$

where K is the number of independent attention heads, α_{ij}^k is the attention weight for the k-th head and $\mathbf{W}^k$ is the weight matrix for the k-th head [28].

Algorithm 3 Attention Computation for a Single GATv2 Head k

Require: Node features for central node i (h_i) and neighbor j (h_j); Weight matrix $\mathbf{W}_k$; Attention vector $\mathbf{a}_k^T$.
Ensure: Aggregated features for node i from head k, denoted h_i''.

for all node x in the graph **do**
    Transform node features: $h_x' \leftarrow \mathbf{W}_k h_x$
end for
for all node i **do**
    Initialize: $h_i'' \leftarrow \mathbf{0}$
    Compute attention logits: $e_{ij} \leftarrow \mathbf{a}_k^T \text{LeakyReLU}([h_i' \parallel h_j'])$ for $j \in \mathcal{N}(i)$
    Normalize: $\alpha_{ij} \leftarrow \frac{\exp(e_{ij})}{\sum_{l \in \mathcal{N}(i)} \exp(e_{il})}$
    Aggregate: $h_i'' \leftarrow \sum_{j \in \mathcal{N}(i)} \alpha_{ij} h_j'$
end for
return h_i''

The final node representation combines data from all attention heads using various composition methods. Concatenation retains all attention-specific information while increasing dimensionality. In contrast, averaging reduces dimensionality while preserving diversity from various attention perspectives. Alternatively, learned combination approaches such as attention layers over head outputs or weighted sums with learnable weights can be utilised.

Effective GATv2 architectures brings in several key design principles. Normalization strategies follow a two-step approach: Batch Normalization after linear transformations to stabilize activations, and Layer Normalization after aggregating attention to manage varying sizes and scales. This method enhances training stability and convergence, crucial for complex architectures with many parameters and irregular graph-structured data. Standard input scaling like 0 to 1 is insufficient alone. As data moves through layers, its distribution changes. Applying Batch Normalization post-linear transformation stabilizes activations, while Layer Normalization post-aggregation normalizes features per node. This two-stage process ensures well-behaved activations, improving stability, speed, and performance, especially in complex, large-parameter architectures.

Skip connections are important for training deeper models. They help prevent the problem of gradient vanishing. These connections are often set up as residual connections: $h_i^{(l+1)} = h_i^{(l)} +$

GATv2Layer$(h_i^{(l)}, \{h_j^{(l)}\}_{j\in\mathcal{N}(i)})$. Some advanced versions might include gated skip connections or attention-weighted skip connections.

Regularisation strategies prevent overfitting in a variety of ways. Dropout is often used with attention coefficients, hidden representations following GNN layers, and input features. DropEdge removes edges randomly during training to reduce the co-adaptation of neighbouring nodes. Attention dropout focusses on attention weights, α_{ij}. When combined with adequate normalisation, these regularisation methods allow training of deep GATv2 architectures. They preserve the model's capacity to grasp complicated spatial relationships in collaborative perception settings.

ECC was selected for its capacity to incorporate spatial relationships into the message-passing process. By conditioning the convolution operation on pairwise geometric information, ECC can better capture distinctions such as direct adjacency or overlapping coverage. This capability is crucial for accurate occupancy prediction in structured warehouse environments.

The fundamental idea behind ECC is that spatial relationships between nodes can be modeled through adaptive transformations during message passing [53]. This enables context-specific modifications to node features before they are aggregated. The update rule for a node h_i' can be expressed as:

$$h_i' = \text{Act}\left(\frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)} F_\Theta(h_i,h_j)h_j + \mathbf{b}\right) \tag{4.2.3}$$

where $F_\Theta(h_i,h_j)$ is a neural network that generates adaptive weights based on node feature combinations, $\mathbf{b}$ is a learnable bias term, $|\mathcal{N}(i)|$ is the cardinality of the neighborhood for normalization, and Act is a non-linear activation function.

The conditioning network F_Θ processes concatenated node features through several fully connected layers [53]:

$$F_\Theta(h_i,h_j) = \mathbf{W}_L \cdot \text{ReLU}(\mathbf{W}_{L-1} \cdot \text{ReLU}(...\mathbf{W}_1 \cdot [h_i||h_j] + \mathbf{b}_1)... + \mathbf{b}_{L-1}) + \mathbf{b}_L \tag{4.2.4}$$

This design allows ECC to learn complex, nonlinear transformations based on spatial relationships, capturing geometric patterns that fixed aggregation schemes might miss. For collaborative perception tasks, the conditioning network learns to distinguish between different types of spatial configurations through node feature interactions.

The conditioning network computes adaptive weights based on node feature combinations. It accepts concatenated node features as input through fully connected layers with Rectified Linear Unit (ReLU) activations and dropout regularization. The output generates parameters that define the adaptive transformation matrix $\mathbf{W}_{ij}$, allowing the network to modify its computational behavior based on specific node pair characteristics.

ECC naturally supports permutation equivariance by symmetrically treating node pairs and generating adaptive weights. The architecture implicitly learns spatial relationships from node feature interactions. This enables the effective modeling of collaborative perception scenarios. However, it requires careful memory management for scalable implementation. GATv2 and

ECC operate within the message passing framework [53], updating node states through:

$$\mathbf{m}_{i\leftarrow j}^{(l+1)} = \psi\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}\right) \tag{4.2.5}$$

$$\mathbf{m}_i^{(l+1)} = \bigoplus_{j\in\mathcal{N}(i)} \mathbf{m}_{i\leftarrow j}^{(l+1)} \tag{4.2.6}$$

$$\mathbf{h}_i^{(l+1)} = \phi\left(\mathbf{h}_i^{(l)}, \mathbf{m}_i^{(l+1)}\right) \tag{4.2.7}$$

where ψ is the message function, $\bigoplus$ is aggregation, and ϕ is the update function.

GATv2 uses the message function $\alpha_{ij}^k \mathbf{W}^k h_j^{(l)}$ with learned attention weights, weighted sum aggregation and multi-head concatenation. ECC uses $F_\Theta(h_i^{(l)}, h_j^{(l)})h_j^{(l)}$ with transformations based on node feature combinations, mean/sum aggregation and activation with bias. GATv2 excels at dynamic neighbor weighting through attention mechanisms. In contrast, ECC specializes in adaptive transformations based on node interactions, enabling context-specific message passing without requiring explicit edge attributes.

While Graph Sample and Aggregate (GraphSAGE) offers computational advantages through neighbor sampling [20], its sampling mechanism can be counterproductive for collaborative perception applications. GraphSAGE samples only neighbor subsets rather than considering complete spatial relationships, potentially losing important spatial context critical for precise occupancy reasoning. Standard Graph Convolutional Network (GCN)s treat all neighbors equally through uniform aggregation [50], limiting their ability to model heterogeneous spatial relationships in collaborative perception. The attention mechanisms in GATv2 and edge-conditioning in ECC provide more sophisticated approaches to handle spatial heterogeneities and collaborative coordination requirements.

The selection of both GATv2 and ECC provides complementary benefits: GATv2 excels at adaptive importance weighting among spatial regions, while ECC provides superior geometric relationship modeling. This architectural diversity strengthens experimental validity. It ensures that conclusions about collaborative perception effectiveness are not dependent on the characteristics of a single GNN variant.

4.2.3 Class Imbalance Handling

Collaborative perception in warehouse environments suffers from a severe class imbalance because the occupancy patterns are spatially sparse. The vast majority of the operational space is unoccupied, with concentrated activity around workstations, boundaries, and robot pathways. This distribution favors the majority classes and lowers sensitivity to detecting minority classes. As a result, models will reach high overall accuracy by just predicting the majority class, but they will struggle to learn important characteristics of minority classes.

By assigning different weights to minority and majority classes during training, weighted loss functions address this issue. To solve distributional imbalance, the weighted Binary Cross-Entropy (BCE) loss modifies the conventional formula:

$$\mathscr{L}_{weighted} = -\frac{1}{N}\sum_{i=1}^{N}\left[w_{pos}\cdot y_i \cdot \log(\sigma(\hat{z}_i)) + w_{neg}\cdot(1-y_i)\cdot\log(1-\sigma(\hat{z}_i))\right] \tag{4.2.8}$$

where $\hat{z}_i$ represents raw logits, y_i denotes ground truth binary labels, σ is the sigmoid function, and w_{pos}, w_{neg} are class weights [70]. The positive weight is calculated using inverse class frequency:

$$w_{pos} = \frac{N_{unoccupied}}{N_{occupied}} \tag{4.2.9}$$

This weighting makes sure that minority classes get proper attention during backpropagation. It helps prevent too much bias toward the majority class while keeping overall accuracy intact. The implementation works smoothly with PyTorch Geometric by using `BCEWithLogitsLoss` with determined positive weights. This approach makes sure the stability in numbers and efficiency in computation. It maintains the ability for collaborative learning while creating balanced sensitivity to both occupied and unoccupied areas. This balance is neccessary for accurate navigation and obstacle avoidance in changing warehouse environments.

4.3 Evaluation Framework for Collaborative Perception

This study carries out an in-depth evaluation of occupancy prediction using GNNs in collaborative environments. It uses a traditional metrics-based approach yet complements it with a unique spatial reasoning perspective to offer a more nuanced performance analysis across six GNN architectures. The framework meets the needs of collaborative robotics. It shows how good occupancy prediction affects navigation safety, coordination efficiency and operational reliability. This is identified through several analytical views including classification performance, confusion matrix analysis, spatial accuracy evaluation and distance-based assessment.

4.3.1 Confusion Matrix Analysis

Confusion matrix analysis allows for a detailed evaluation of discrimination across different architectural families. It highlights behavior relevant to collaborative robotics. The framework looks at the following components:

Table 4.3: Confusion Matrix Terminology for Occupancy Prediction

Metric	Description
True Positive (TP)	Occupied regions correctly detected as occupied. Crucial for reliable obstacle detection and collision avoidance.
True Negative (TN)	Free space correctly identified as free. Enables safe and efficient navigation in dynamic environments.
False Positive (FP)	Free space incorrectly classified as occupied. Can lead to unnecessary avoidance maneuvers and reduced efficiency.
False Negative (FN)	Occupied regions incorrectly classified as free. Poses safety risks by allowing potential collisions with undetected obstacles.

The confusion matrix is defined as:

$$\begin{bmatrix} \text{TN} & \text{FP} \\ \text{FN} & \text{TP} \end{bmatrix}$$

Safety-critical evaluation focuses mainly on reducing FN to make sure all occupied areas are detected for effective collision avoidance. The framework tries to maintain a balanced trade-off between sensitivity and the FP rate. It does so by examining discrimination trends across different GNN variants. Additionally, it classifies deployment readiness based on safety and operational efficiency.

4.3.2 Classification Evaluation Metrics

The classification evaluation uses traditional metrics to allow for standardized architectural comparison and to evaluate the suitability of models for use in resource-limited situations. The main metrics include:

Table 4.4: Evaluation Metrics

Metric	**Formula**	**Description**
Accuracy (Acc)	$\text{Acc} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}$	Proportion of correctly classified voxels (both occupied and free) out of all predictions.
Precision (P)	$P = \frac{\text{TP}}{\text{TP}+\text{FP}}$	Indicates the fraction of predicted occupied voxels that are truly occupied.
Recall (R)	$R = \frac{\text{TP}}{\text{TP}+\text{FN}}$	Measures the ability to detect actual occupied voxels.
F1-score (F_1)	$F_1 = 2 \times \frac{P \times R}{P+R}$	Harmonic mean of precision and recall, balancing both FP and FN.

These metrics give a clear understanding of how the model performs. They balance prediction accuracy and computational cost. This balance is neccessary for using the model in real-world robotics and spatial perception tasks.

4.3.3 Spatial Accuracy Evaluation

Evaluating the spatial accuracy of a collaborative robotic system requires more than just traditional classification methods to assess spatial reasoning capabilities. Distance-based assessment addresses robotics-specific spatial accuracy requirements through:

$$\text{Distance Accuracy}(\tau) = \frac{|\{p \in P_{pred} : \min_{g \in G_{truth}} ||p-g||_2 \leq \tau\}|}{|P_{pred}|} \tag{4.3.1}$$

where P_{pred} is the predicted occcupied points, G_{truth} is the ground truth occupied points, τ represents distance tolerance and $||p-g||_2$ represents Euclidean distance. The method uses three tolerance levels (0.15m, 0.25m and 0.35m) [71].

The methodology presented in this chapter establishes a comprehensive framework for transforming the complex challenge of collaborative perception into a tractable machine learning problem. The preprocessing pipeline systematically addresses the fundamental obstacles of temporal synchronization, spatial alignment, and data quality. These challenges have historically limited the effectiveness of collaborative perception systems. The graph neural network architectures provide complementary approaches to spatial reasoning, each optimized for different aspects of collaborative perception challenges. The evaluation framework ensures that model performance is assessed not merely through traditional classification metrics. It is also evaluated using robotics-specific measures that directly relate to operational safety and efficiency. This integrated methodology forms the foundation for the experimental validation presented in subsequent chapters. It provides both theoretical rigor and practical applicability for real-world collaborative robotics deployment.

5 Experiments and Results

This chapter discusses the experimental setup and assessment of the collaborative perception framework created in this research. The experiments depict the performance of the preprocessing pipeline and GNN architectures working in real-world warehouse settings. They set performance standards for collaborative robotics applications. By evaluating six different GNN models, this research finds the key insights about attention-based and edge-conditioned methods.

5.1 Experimental Setup

The collaborative perception framework was experimentally evaluated in a well-structured testbed. This testbed closely mimicked actual warehouse operations and provided controlled conditions for systematic assessment of the algorithms. The experimental infrastructure includes precision tracking systems, realistic environmental constraints and advanced robotic platforms. This setup allows to have a detailed evaluation of collaborative perception algorithms in various operational scenarios.

5.1.1 Warehouse Environment

A set of experiments was conducted in a controlled warehouse environment specifically designed to replicate real-world collaborative robotics scenarios. This sophisticated testbed integrates high-precision MoCap systems (Vicon) with realistic industrial infrastructure to create that optimal environment for collaborative perception research.

The experimental arena, shown in Figure 5.1, covers a rectangular area of 20.7 m × 9.92 m (x-axis: -11.1 m to 9.6 m, y-axis: -4.42 m to 5.5 m). This area provides space for two robot collaborative operations. Inside the arena, five workstations are placed as obstacles to mimic real warehouse operations. This setup allows collaborative robots to encounter various scenarios. These include wide navigation corridors, narrow passages between storage units, and complex multi-obstacle areas that are common in today's automated warehouses. The testbed's modular design allows to systematically reconfigure it for various experimental layouts.

The environmental infrastructure includes a Vicon MoCap system which provides sub-millimeter position accuracy at frequencies of upto 200 Hz. This precise tracking ability lays as the groundwork for training and validating collaborative perception algorithms. It also allows for detailed analysis of robot coordination dynamics and spatial relationship modeling.

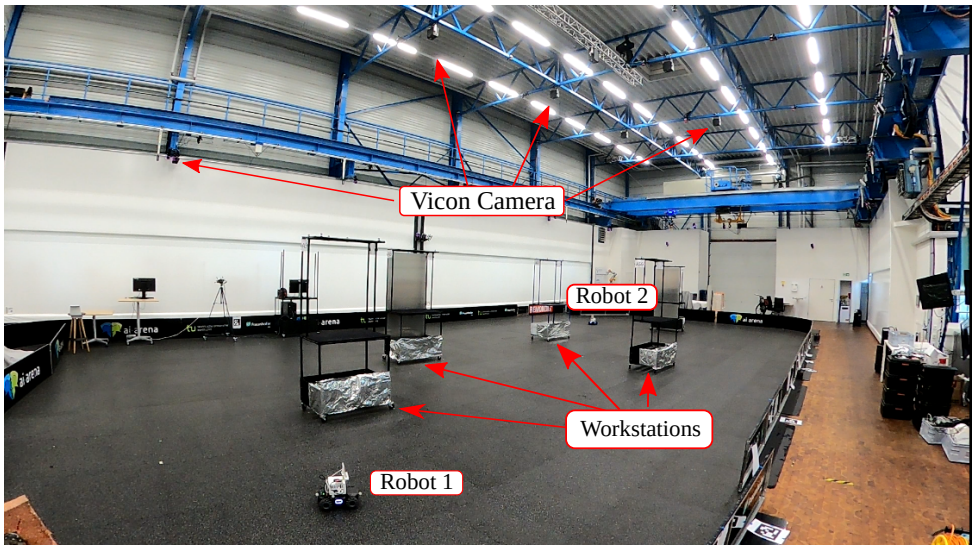


Figure 5.1: Experimental warehouse arena (20.7m × 9.92m) featuring two Robomaster platforms with mmWave radar sensors and five workstation obstacles. The controlled environment includes overhead Vicon MoCap cameras providing high-precision ground truth for collaborative perception validation.

5.1.2 Robotic Platform Configuration

The experimental framework uses two autonomous Robomaster platforms. Each one has a set of sensors designed for working together on perception tasks. The system combines sensing, communication, computation and mobility subsystems to create an advanced collaborative robotics platform.

Each robotic platform has four key subsystems. The sensing subsystem uses mmWave radar sensors (IWR6843) to provide high-resolution spatial and velocity measurements. It is further supported by visual cameras and Inertial Measurement Unit (IMU)s for better understanding of the environment. The communication subsystem uses ESP32 Wi-Fi modules which enables real-time coordination and data sharing between collaborative platforms with low-latency performance. The computation subsystem runs on NVIDIA Jetson AGX Orin processors. Finally, the mobility subsystem allows omnidirectional navigation with strong obstacle avoidance features, providing reliable operation in various warehouse settings.

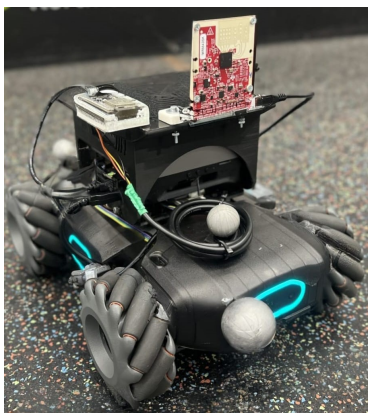


Figure 5.2: Robomaster platform equipped with sensors and MoCap markers

The configuration of the sensors allows the robots to create point cloud data that represents the obstacles and features of the environment around them. Each robot can sense within its own range and this distributed sensing is the basis for the next step in robot perception. By sharing and fusing the data they generate, individual robots can collectively understand the environment around them, which is far beyond the capabilities of any single platform.

Each experimental trial involves moving both robotic platforms along set paths. During this time, the system continuously gathers sensor data and ground truth information from the MoCap setup. The experimental protocol confirms complete coverage of diverse collaborative scenarios such as cooperative navigation, distributed sensing and dynamic obstacle avoidance tasks.

Figure 5.3 illustrates a comprehensive experimental pipeline outlining the complete workflow that encompasses model development, the training framework, and evaluation.

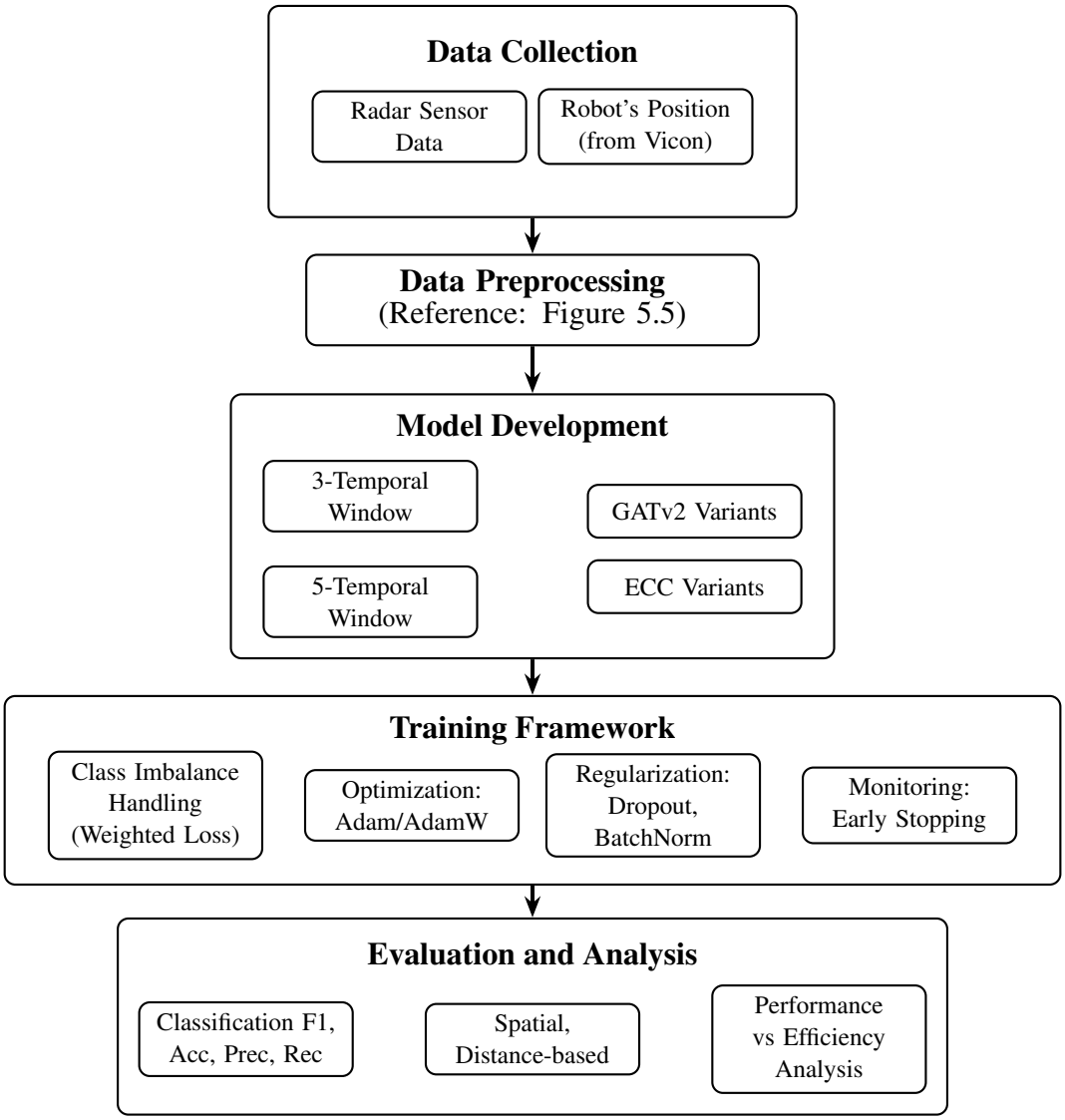


Figure 5.3: Experimental Pipeline

5.1.3 Data Collection

Three distinct experimental layouts were designed with varying spatial complexities and collaborative scenarios. Each layout reflects a different operational phase that is typical of warehouse automation. The three layouts are shown in Figure 5.4.

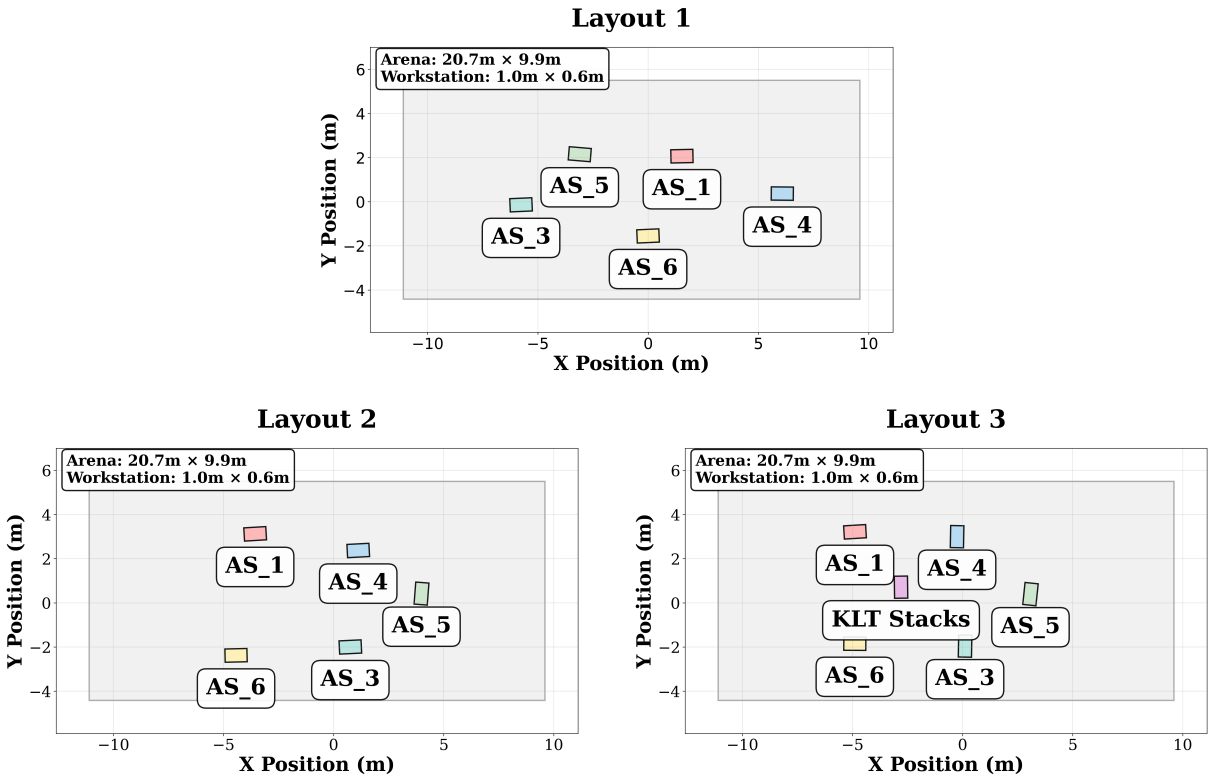


Figure 5.4: Arena Layout Configuration

The framework encompasses distinct movement scenarios designed to evaluate collaborative perception capabilities across varying warehouse automation contexts. These scenarios reflect real-world operational phases commonly encountered in modern automated warehouses. In these settings, multiple AMRs must coordinate their sensing and navigation activities within shared workspace environments.

Horizontal scenarios in warehouses involve robots in parallel corridors, coordinating movements and maintaining awareness around shared workstations. This setup enhances collaborative perception due to overlapping sensor ranges.

Diagonal scenarios require more complex navigation at intersecting paths, challenging perception algorithms and offering complementary sensor views from different angles. Vertical scenarios focus on sequential operations along shared corridors, testing the perception system's ability to handle temporal offsets and distinguish between current and cleared areas.

The dataset extends to hybrid scenarios where robots combine movement patterns. These include horizontal-diagonal, with one robot in parallel corridors and another at intersecting paths; horizontal-vertical, aligning parallel with sequential movements; and vertical-horizontal, reversing these patterns. Also, randomized scenarios were captured, with robots operating freely to challenge perception algorithms in dynamic, unstructured settings.

Overall, The thesis includes 34 individual datasets across three different experimental layouts. These sessions were captured during an intensive three-month period from February through May 2025. Each session represents a full collaborative scenario that lasted for 3 to 4 minutes. They create temporal sequences with enough duration and complexity to train strong Collaborative perception algorithms with GNNs. Specifically, Layout 1 contains 20 recordings, Layout 2 includes 9 recordings, and Layout 3 consists of 5 recordings.

For this thesis, each dataset contains one complete recording session with synchronized sensor data and detailed ground truth labels. This definition helps to keep the timing consistent, which is considered to be important for developing collaborative perception algorithms. It also enables a clear organization across the three experimental layouts.

The preprocessing pipeline ensures that each collaborative dataset contains synchronized radar point clouds from both robots at 30 Hz. It also includes the use of high-precision Vicon MoCap data (200Hz) that provides robot pose information, environmental context, scenario metadata and labels for workstations, robots, boundaries and unknown areas. Then, it takes point cloud data from both robots and combines them into unified spatial voxels. This helps to create the graph representations that capture true collaborative sensor fusion instead of single-robot approaches. This dual-robot integration allows for the extraction of new collaborative features. These features quantify spatial overlap, coverage redundancy, and coordinated perception patterns that are important for collaborative warehouse automation.

5.2 Preprocessing Implementation

The preprocessing implementation uses the methodology to change raw sensor data into graph representations for GNN training. The pipeline processes synchronized radar point clouds and Vicon MoCap data. It converts raw measurements into spatially-aligned datasets that maintain the spatial relationships necessary for occupancy prediction. This approach overcomes the major challenges of combining data from different sensors in dynamic warehouse environments. Precise timing and spatial alignment are important for successful collaborative perception.

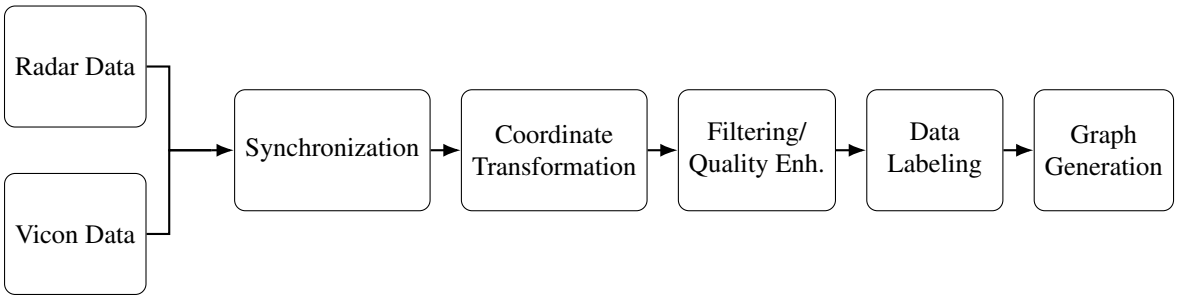


Figure 5.5: Data processing pipeline

The data extraction process varied across the different layouts, requiring distinct extraction approaches for each configuration. For Layout 1, sensor data was captured and stored in ROS bag files (rosbags), which provided a standardized format for multi-sensor data synchronization and replay capabilities. However, for Layout 2 and Layout 3, the data collection process utilized plain log files instead of the ROS bag format. This difference in data storage formats necessitated specific coordinate system transformations during the data extraction phase for Layout

2 and Layout 3. Specifically, the coordinate transformation $x = -y$ and $y = -x$ was applied during preprocessing to ensure spatial consistency across all experimental layouts, allowing for meaningful comparison of results despite the different data extraction method employed.

The synchronization stage connects multiple sensor streams from various robotic platforms while keeping data accurate during the transformation process. The implementation manages the challenges of heterogeneous sensor data fusion. This includes radar measurements that have different sampling rates, Vicon MoCap data that offers sub-millimeter positioning accuracy, and the temporal synchronization needed to ensure spatial consistency in collaborative robotic operations. The synchronization process used a fixed Vicon resampling interval of $\Delta t_{\text{fixed}} = 25,\text{ms}$ and a cross-modal synchronization threshold of $\delta t_{\text{thresh}} = 0.5,\text{s}$. Nearest-neighbor matching based on absolute timestamp differences $|t_i - t_j|$ was used to align radar data from both robots and the Vicon system within these temporal bounds.

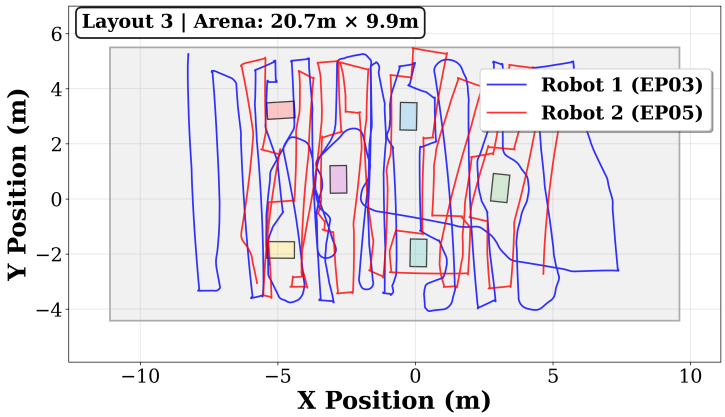


Figure 5.6: Parsed Vicon data showing robot trajectory from an experimental run

Coordinate transformation is the basis of the entire preprocessing pipeline. It converts radar measurements from individual robot frames into a single global coordinate system, using Vicon data as the reference point. Figure 5.6 shows the precision and continuity of the MoCap system's tracking abilities throughout an experimental run. This provides the spatial backbone that allows precise coordinate transformation of all sensor measurements. The smooth, continuous trajectory indicates successful tracking without any interruptions or positioning errors. This is crucial for maintaining the spatial integrity of the transformed sensor data.

The transformation process includes complex mathematical operations that takes into account the robot's dynamic movements such as position updates, changes in orientation and the temporal synchronization requirements. The trajectory data showcases movement patterns typical of collaborative robotics. These include coordinated maneuvers, overlapping coverage areas, and dynamic positioning changes that characterize real-world scenarios. This detailed spatial information helps the preprocessing pipeline accurately transform sensor measurements while considering the dynamic nature of the collaborative workspace.

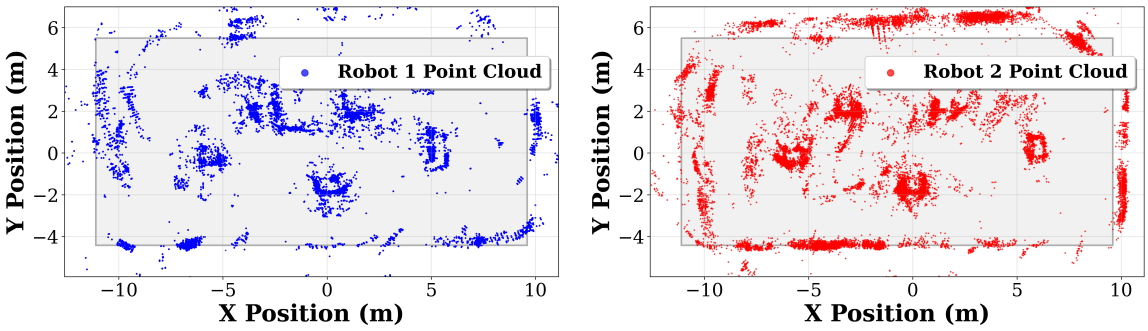


Figure 5.7: Coordinate transformation results showing radar point clouds from both robots aligned in global reference frame. Left panel displays Robot 1's measurements, right panel shows Robot 2's measurements.

Figure 5.7 shows successful transformation outcomes with radar point clouds from both the robots aligned in the global frame. The left panel presents Robot 1's sensor data, while the right panel displays Robot 2's measurements, both now aligned within the same global reference frame. Blue and red clusters signify distinct measurements from each robot. This illustrates how the transformation process maintains the spatial distribution of sensor detections while providing a unified spatial view of the collaborative workspace.

The visualization highlights the complementary nature of collaborative sensor coverage, where each robot adds unique spatial data which then enhances the overall perception of the system. The successful alignment confirms the coordinate transformation's accuracy. It also ensures that the spatial relationships between sensor detections have been maintained throughout the transformation process. The different clustering patterns in each robot's data reflect varying perspectives and sensor orientations. This illustrates how collaborative perception achieves comprehensive workspace coverage that single sensor systems cannot achieve.

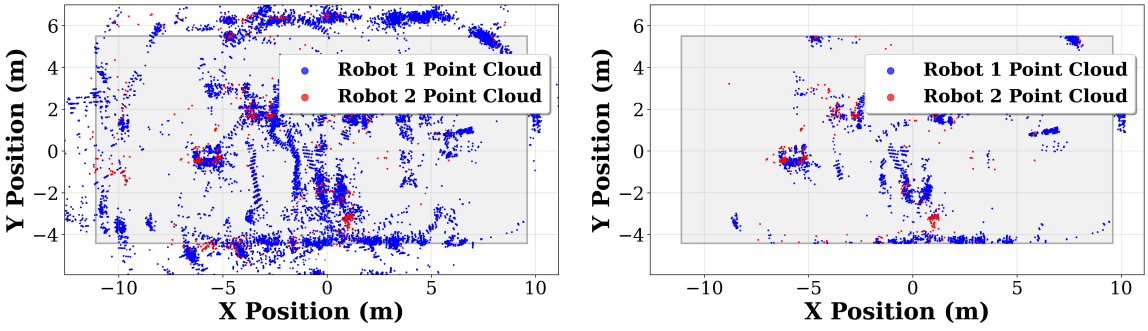


Figure 5.8: Point Cloud Cleaning Comparison. Left: Original data. Right: Cleaned data

The data quality enhancement phase utilises the advanced filtering techniques to boost the SNR while keeping the integrity of true target detections. Figure 5.8 dramatically shows the filtering effectiveness, transforming noisy raw sensor data into clean, processed measurements that are suitable for GNN training. The left panel shows the original data. It has visible noise artifacts, spurious detections, and environmental clutter typical of raw radar measurements in complex indoor settings. The right panel displays the cleaned data after applying the filtering process. It demonstrates much less noise while keeping the important spatial structure of true target detections.

The filtering process takes a multi-layered approach to address the challenges of radar sensor data in warehouse environments. The SNR filtering part uses a minimum SNR threshold of 5.0 dB, effectively getting rid of low-quality measurements while keeping high-confidence detections. Spatial boundary filtering limits the workspace to specific operational boundaries. It uses an X-range from -10.0 to 10.5 meters and a Y-range from -5.0 to 6.0 meters, ensuring measurements match the actual collaborative workspace. Height filtering restricts detections to the operational zone between 0.05 and 2.5 meters, which removes ground reflections and ceiling artifacts.

Statistical outlier detection is the most advanced part of the filtering process. It uses k-nearest neighbor analysis with k set to 20 neighbors and a standard deviation threshold of 3.0. This method calculates the mean distance to the nearest neighbors for each point and finds outliers based on statistical deviation from the global distribution. The field-of-view filtering limits detections to a 120-degree angle, ensuring measurements come from the intended sensor coverage area.

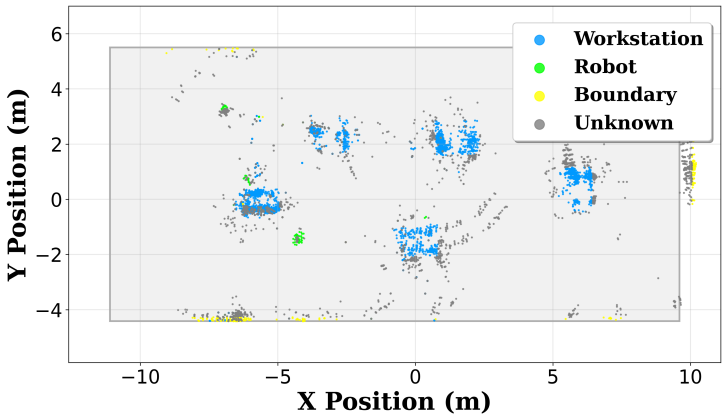


Figure 5.9: Labeled Data

The data labeling implementation generated accurate ground truth labels from processed sensor measurements. Figure 5.9 shows the labeling achieved with the help of this implementation. It illustrates how raw sensor measurements have been organized into meaningful occupancy categories. The visualization uses different colors to represent various objects, including workstations, boundaries, robots, and unknown/unoccupied areas. This provides the essential ground truth for supervised learning in GNNs.

Table 5.1: Object Distribution Analysis

Object	**Count**	**Percentage**	**Binary Label**	**Binary Class**
Workstation	247,847	52.30%	1	Occupied
Boundary	35,667	7.53%	1	
Robot	9,305	1.96%	1	
KLT Stacks	111	0.02%	1	
Occupied Subtotal	**292,930**	**61.81%**	**1**	
Unknown/Unoccupied	180,981	38.19%	0	Unoccupied
Total Dataset	**473,911**	**100.00%**	–	–
Class Imbalance Ratio: 0.618:1 (Unoccupied:Occupied)				

A statistical analysis of the collaborative dataset was conducted to measure class imbalance and find suitable weighting parameters for optimizing the loss function. The entire dataset consists of 473,911 labeled data points spread across warehouse layouts and scenarios. Table 5.1 offers a detailed breakdown of the class distribution in the collaborative perception dataset, highlighting the spatial sparsity typical of warehouse environments.

The analysis shows that occupied areas make up 292,930 voxels (61.8%), while unoccupied areas comprise 180,981 voxels (38.2%). In the occupied category, workstation areas account for 52.3% of the total dataset, indicating the warehouse's focus on manufacturing operations. Based on this analysis, the positive weight for the weighted loss function is calculated as $w_{pos} = N_{unoccupied}/N_{occupied} = 180,981/292,930 = 0.618$. This results in a class imbalance ratio of 0.618:1, which needs methodological intervention to ensure balanced learning performance.

Implementing the weighted loss function involves calculating class weights and applying them to the BCEWithLogitsLoss function. Specifically, the implementation uses `pos_weight = torch.tensor([0.618]).to(device)`. This weighted loss approach aims to reduce model bias and improve generalization capabilities across different warehouse operational scenarios. It prevents major class overfitting and ensures steady performance throughout the collaborative perception framework.

The graph generation converts synchronized point cloud measurements from two robots into organized graph representations. These graphs capture spatial relationships and robot cooperation patterns essential for occupancy prediction tasks. Each frame processes combined point clouds with robot source tracking. This allows for novel collaboration features that highlight individual robot contributions within shared spatial areas. The 0.1m voxel resolution strikes a good balance between spatial detail preservation and computational tractability. Processing times increase linearly with the complexity of the dual-robot input.

The voxelization merges radar points from both robots into unified spatial voxels while maintaining robot source identification. Each voxel collects spatial data from both robots and computes collaboration features that measure individual contributions, spatial overlap patterns, and coverage redundancy. These elements are important for robust perception. Voxel label assignment uses majority voting from all robot contributions. This ensures consistent class labeling while leveraging improved spatial coverage from dual-robot sensor fusion.

The following adjacency matrix shows the fully connected layout using actual GNN frame data (file: `1746370721.936099.pt` from the temporal_5 dataset). This complete graph has 5 voxels and a total of 20 connections, with 4 occupied voxels and 1 unoccupied voxel (Figure 5.10).

$$\mathbf{A}_{5\times5} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

The matrix shows symmetry ($A_{ij} = A_{ji}$), confirming undirected edges. It has no self-loops ($A_{ii} = 0$) and gets an average degree of 9.0, which shows full connectivity between all node pairs. Each node processes 17-dimensional features, including spatial coordinates, relative positions, distance measurements, robot contribution counts, and temporal offset indicators.

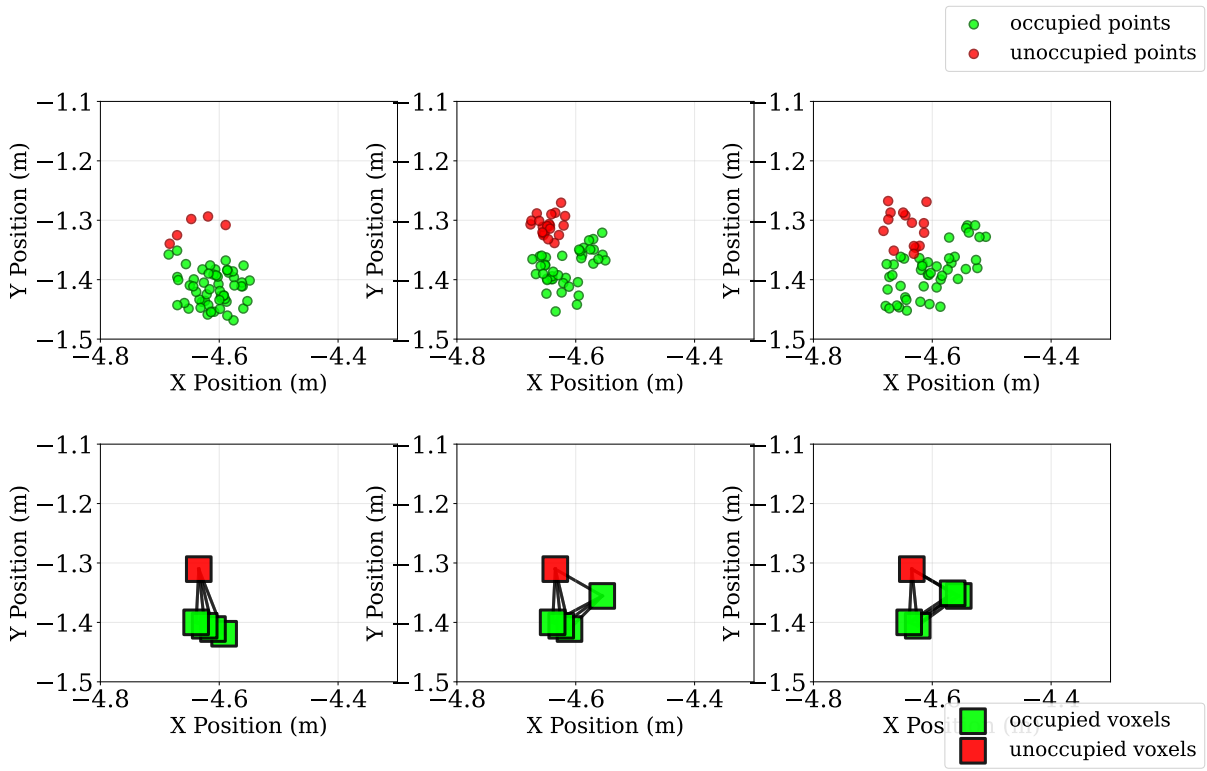


Figure 5.10: Comparison of raw point cloud data (top) and corresponding graph structure representation (bottom) for three consecutive frames. Green indicates occupied space, red indicates unoccupied space. Graph structure visualization showing fully connected topology for collaborative perception with spatial voxels as nodes colored by occupancy status.

The temporal integration of collaborative data into graph representations occurs through causal windowing and collaborative feature aggregation. The framework processes temporal sequences using strictly causal windows that exclude future information, ensuring real-time deployment compatibility for autonomous robot systems.

The implementation of different temporal window configurations changes the structure and availability of training data in the collaborative perception framework. The temporal modeling method includes single-frame processing (T1), sliding window size 3 (T3) ($[t-2, t-1, t]$), and sliding window size 5 (T5) ($[t-4, t-3, t-2, t-1, t]$). Each of these has specific requirements for contextual information, which directly affects the total number of training samples. This phenomenon occurs because longer temporal sequences require more historical context to keep the temporal dependencies intact. As a result, there are fewer complete sequences for model training and evaluation.

Table 5.2: Temporal Window Impact on Frame Availability

Layout	**Temporal Window 1**	**Temporal Window 3**	**Temporal Window 5**
Layout 1	7,138 frames	7,098 frames (-40)	7,058 frames (-80)
Layout 2	8,454 frames	8,436 frames (-18)	8,418 frames (-36)
Layout 3	4,323 frames	4,313 frames (-10)	4,303 frames (-20)
Total	**19,915**	**19,847**	**19,779**

Overall, the impact across all layouts shows that while the reduction in available training data is measurable, it stays within acceptable limits for effective model development. The full dataset contains 19,915 frames for single-frame processing. This number drops to 19,847 frames for sliding window size 3 and 19,779 frames for sliding window size 5. The largest decrease is only 136 frames, which is about 0.68% of the total dataset. This slight reduction allows all temporal modeling approaches to maintain enough statistical power for reliable performance evaluation. It also keeps the necessary temporal dependencies intact for collaborative perception tasks.

The dataset partitioning strategy used in this collaborative perception framework aims to prevent temporal information leakage. It also ensures a good distribution of data across training, validation, and testing stages. Instead of randomly assigning individual frames or sequences, the partitioning algorithm works at the dataset level. It guarantees that complete recording sessions go to specific splits. This method is crucial for maintaining the temporal integrity of the collaborative perception data. It helps to avoid situations where the model learns patterns that cross different evaluation phases. Such scenarios could result in inflated performance metrics and a weaker generalization assessment.

Table 5.3: Dataset Split Distribution

Layout	**Training**	**Validation**	**Testing**	**Total**
Layout 1	14 datasets	3 datasets	3 datasets	20 datasets
Layout 2	6 datasets	1 dataset	2 datasets	9 datasets
Layout 3	2 datasets	1 dataset	2 datasets	5 datasets
Total	**22 datasets**	**5 datasets**	**7 datasets**	**34 datasets**

This dataset-level partitioning approach makes sure that no temporal information from individual recording sessions overlaps between different splits. This separation is crucial for evaluating collaborative perception algorithms without bias. The careful distribution maintains variety in layout across all partitions. This prevents any single experimental setup from overshadowing the evaluation process. It also provides enough representation for dependable statistical analysis across various collaborative perception scenarios.

Table 5.4: Frame Distribution Across Splits

Split	**Datasets**	**T1 Frames**	**T3 Frames**	**T5 Frames**
Training	22 (64.7%)	14,356	14,312	14,268
Validation	5 (14.7%)	2,944	2,934	2,924
Testing	7 (20.6%)	2,615	2,601	2,587
Total	**34 (100%)**	**19,915**	**19,847**	**19,779**

The resulting frame distribution across different temporal configurations shows the steady influence of the dataset-level partitioning strategy while keeping the effects of the temporal windows that were established earlier. The training partition has 14,356 frames for single-frame processing, 14,312 frames for sliding window size 3, and 14,268 frames for sliding window size 5. In total, there are 42,936 frames across all temporal setups, which is 72.1% of the combined dataset. This large allocation for training offers strong statistical power for model learning. It also ensures that the number of available frames decreases systematically as the size of the temporal window increases. The validation partition contains 2,944, 2,934, and 2,924 frames for the corresponding temporal configurations, adding up to 8,802 frames or 14.8% of the total dataset. This provides enough data for reliable tuning of hyperparameters and model selection for all time modeling approaches.

The testing partition features 2,615, 2,601 and 2,587 frames for the three temporal windows, totaling 7,803 frames or 13.1% of the complete dataset. This distribution guarantees thorough evaluation capabilities while maintaining the temporal dependency constraints inherent in each window configuration. The consistent percentage reduction across temporal windows in each split indicates that the partitioning strategy maintains the essential traits of temporal modeling needs. This strategy also ensures balanced representation for reliable comparative analysis.

The importance of this partitioning framework goes beyond simple data distribution considerations. It addresses the core requirements for evaluating collaborative perception. The framework ensures temporal independence between splits, while also maintaining layout diversity and temporal window consistency. This approach lays a solid foundation for unbiased assessments of collaborative perception algorithms across various temporal modeling strategies, spatial settings, and interaction patterns.

5.3 GNN Architecture Implementation

Six different GNN architectures were implemented and tested for collaborative perception tasks. These include Standard GATv2, Complex GATv2, and ECC. Each architecture used two temporal window configurations, Temporal-3 and Temporal-5, to examine how temporal modeling affects spatial reasoning performance. The architectures vary in complexity, number of parameters, and attention mechanisms while keeping the same input-output specifications for comparative evaluation.

5.3.1 Standard GATv2 Architecture Implementation

The Standard GATv2 architecture uses multi-head attention mechanisms for graph-based spatial reasoning tasks. It processes 17-dimensional node features through three GATv2 layers, maintaining consistent 64-dimensional hidden representations.

Table 5.5: Standard GATv2 - Architecture Specifications

Layer	**Operation**	**Input**	**Output**	**Heads**	**Dropout**	**Parameters**
Input	Node Embedding	17	64	-	0.0	1,088
GATv2-1	Multi-Head Attention	64	64	4	0.2	8,448
GATv2-2	Multi-Head Attention	64	64	4	0.2	8,448
GATv2-3	Multi-Head Attention	64	64	4	0.2	8,448
BatchNorm	Normalization	64	64	-	-	771
MLP Hidden	Linear + ReLU	64	128	-	0.5	8,256
Classifier	Linear	128	1	-	0.0	65
					Total Parameters:	**35,524**

This design strikes a good balance between model capability and computational needs for collaborative perception tasks. It employs four attention heads per layer, allowing the model to recognize various spatial relationship patterns simultaneously while maintaining computational tractability. This multi-head attention mechanism helps the model focus on different aspects of spatial relationships in each layer. These aspects include proximity-based connections, similarities, and directional relationships between nodes that represent voxelized workspace regions.

The three-layer architecture provides enough depth for learning hierarchical spatial features while avoiding excessive computational overhead that could hinder real-time deployment. The consistent 64-dimensional hidden representation across the attention layers keeps the model expressive for spatial reasoning while ensuring memory efficiency. With a parameter count of 35,524, this architecture is well-suited for resource-limited collaborative robotics applications where multiple models might need to operate simultaneously.

The dropout regularization strategy, set at 0.2 for attention layers and 0.5 for the Multi-Layer Perceptron (MLP) hidden layer, is designed to prevent overfitting on the spatial reasoning task while allowing enough model capacity to learn complex spatial patterns. The batch normalization layer following the attention stack enhances training stability and convergence, which is particularly important when dealing with irregular graph structures that vary in size and connectivity. The mean pooling aggregation strategy effectively summarizes node-level spatial representations into graph-level features, maintaining spatial context and enabling binary occupancy classification through the final linear classifier.

5.3.2 Complex GATv2 Architecture Implementation

The Complex GATv2 architecture is a high-capacity model that captures complex spatial relationships in collaborative perception scenarios. The increased hidden dimensionality of 128 greatly boosts its representational capacity compared to the standard version. This allows the model to learn more detailed spatial features and intricate interaction patterns from collaborative robot observations. The eight attention heads in each layer enable the model to focus on various types of spatial relationships simultaneously, such as geometric proximity, similarity, temporal consistency, and collaboration patterns.

Table 5.6: Complex GATv2 - Architecture Specifications

Layer	**Operation**	**Input**	**Output**	**Heads**	**Dropout**	**Parameters**
Input	Node Embedding	17	128	-	0.0	2,176
GATv2-1	Multi-Head Attention	128	128	8	0.3	33,280
BatchNorm-1	Batch Normalization	128	128	-	-	513
GATv2-2	Multi-Head Attention	128	128	8	0.3	33,280
BatchNorm-2	Batch Normalization	128	128	-	-	513
GATv2-3	Multi-Head Attention	128	128	8	0.3	33,280
BatchNorm-3	Batch Normalization	128	128	-	-	513
GATv2-4	Multi-Head Attention	128	128	8	0.3	33,280
BatchNorm-4	Batch Normalization	128	128	-	-	513
LayerNorm	Layer Normalization	128	128	-	-	1,024
MLP Hidden	Linear + ReLU	128	256	-	0.5	32,896
Classifier	Linear	128	1	-	0.0	129
					Total Parameters:	**171,397**

With four layers, the model enables hierarchical learning of spatial features. Early layers capture local relationships, like immediate neighborhood patterns and basic geometric shapes. In contrast, deeper layers integrate global spatial context and learn complex multi-scale spatial dependencies. This hierarchy is especially useful for collaborative perception tasks, where grasping both local occupancy patterns and the global workspace structure is essential for accurate spatial reasoning.

The normalization strategy, which uses batch normalization after each attention layer and layer normalization before the final MLP, greatly improves training stability and convergence. This is crucial for the Complex GATv2 architecture due to its large number of parameters and the irregular nature of graph-structured spatial data. The batch normalization layers stabilize attention weight distributions across different graph sizes and connectivity patterns. Meanwhile, layer normalization ensures consistent feature scaling before the final classification.

The higher dropout rate of 0.3 in the attention layers balances the model's increased capacity and helps prevent overfitting on spatial reasoning tasks. The larger MLP hidden layer, with 256 dimensions and 0.5 dropout, improves feature transformation before binary classification. This allows the model to develop more complex decision boundaries for occupancy prediction. Even with a significant increase in parameters, the Complex GATv2 architecture remains computationally feasible while offering improved spatial reasoning capabilities for challenging collaborative perception scenarios.

5.3.3 ECC Architecture Implementation

The Edge-Conditioned Convolution architecture uses a message passing method for spatial reasoning in collaborative perception tasks with fully connected graph structures. Unlike attention-based methods, the ECC architecture handles spatial relationships through direct message passing between all node pairs. This allows for better integration of spatial information across the entire graph structure.

Table 5.7: ECC - Architecture Specifications

Layer	**Operation**	**Input**	**Output**	**Dropout**	**Activation**	**Parameters**
Input	Node Embedding	17	64	0.0	–	1,088
ECC-1	Edge-Conditioned Conv	64	64	0.2	ReLU	139,392
BatchNorm-1	Batch Normalization	64	64	–	–	257
ECC-2	Edge-Conditioned Conv	64	64	0.2	ReLU	139,392
BatchNorm-2	Batch Normalization	64	64	–	–	257
ECC-3	Edge-Conditioned Conv	64	64	0.2	ReLU	139,392
BatchNorm-3	Batch Normalization	64	64	–	–	257
GlobalPool	Mean Aggregation	64	128	–	–	0
Classifier	Linear	128	1	0.0	Sigmoid	129
					Total Parameters:	**420,164**

The ECC implementation features a graph topology where every node is linked to every other node. This design allows spatial information to flow directly between any two points in the collaborative workspace. To pass messages between connected nodes, the architecture combines the features of source and target nodes and processes them with shared neural networks. This concatenation-based approach helps the model learn spatial relationships through the combined representation of node feature pairs, without needing to compute geometric relationships or distances explicitly.

With 420,164 parameters, the model reflects the complexity of the neural networks that learn to process concatenated node feature pairs and create meaningful spatial representations. The three-layer ECC architecture allows for the gradual refinement of spatial features through several rounds of message passing. Each layer lets nodes gather information from all other nodes in

the graph, with deeper layers understanding more complex spatial dependencies and multi-hop relationships. The consistent 64-dimensional hidden representation throughout the architecture balances expressiveness with computational tractability for the fully connected graph structure.

Batch normalization layers after each ECC operation enhance training stability. This stability is crucial due to the dense connectivity and high message volume in fully connected graphs. The mean aggregation method in the global pooling layer effectively summarizes the spatial representations learned through the message passing process. The sigmoid activation in the final classifier produces a probabilistic output for binary occupancy prediction. The fully connected architecture ensures that classification decisions take into account spatial context from the entire collaborative workspace.

5.3.4 Training Infrastructure

Model training took place on a high-performance Personal Computer (PC) with an NVIDIA GeForce RTX 4070 Graphics Processing Unit (GPU) (device 0) that has 12 GB GDDR6X memory. GPU acceleration was enabled through Compute Unified Device Architecture (CUDA) 12.6, which ensured compatibility with deep learning frameworks.

The computational environment used Python 3.12.8 along with PyTorch 2.7.0 and PyTorch Geometric 2.6.1 frameworks, supplemented by Numerical Python (NumPy) 2.2.2, Pandas 2.2.3, and Scikit-learn 1.6.1 for data processing and analysis. GPU utilization was optimized with mixed precision training, using torch.cuda.amp.autocast() to lower memory consumption while keeping numerical stability. The PyTorch setup supported CUDA 12.6 with cuDNN version 90501 for faster tensor operations.

Standardized training procedures were followed across all graph neural network architectures to ensure reproducible results and fair performance comparisons. Training settings included early stopping with a 20-epoch patience based on validation loss. Cosine annealing learning rate scheduling with warm restarts was also employed. Adaptive batch sizes were used, adjusting between 8-32 graphs depending on model complexity and available GPU memory (11.6 GB effective capacity). Memory management was optimized effectively with gradient accumulation strategies and periodic garbage collection to avoid memory fragmentation during long training sessions.

Optimization strategies using grid search for hyperparameter tuning were applied to specific architectures. GATv2 variants employed the Adaptive Moment Estimation (Adam) optimizer with learning rates between [0.001, 0.003], beta parameters (0.9, 0.999), and an epsilon value of 1e-8 for steady convergence. ECC architectures used the Adam with Weight Decay (AdamW) optimizer with a fixed learning rate of 0.0005. A weight decay coefficient of 0.01 was employed for L2 regularization. The architectures also utilized identical beta parameters to enhance convergence stability and improve generalization performance.

Training convergence monitoring was implemented by logging training and validation metrics at each epoch. Automatic model checkpointing occurred whenever there was an improvement in validation loss, which helped preserve optimal model weights and prevent overfitting. To account for differences in model performance from random initialization and stochastic training processes, GAT model variants were trained 2-3 times with different random seeds. The best-performing model based on validation metrics was then selected for final evaluation and comparison.

5.3.5 Architecture Comparison Summary

Table 5.8: GNN Architecture Comparison Summary

Architecture	Parameters (T3/T5)	Layers	Attention Heads	Hidden Dimensions
Standard GATv2	35.5K	3	4	64
Complex GATv2	171.4K	4	8	128
ECC	420.2K	3	–	64

The three architectures showcase different methods for graph-based spatial reasoning, each with its own level of complexity. Standard GATv2 offers good computational efficiency with a moderate number of parameters. Complex GATv2 increases representational capacity by using a deeper architecture and more attention heads. ECC brings in novel edge-conditioned processing to model spatial relationships adaptively. All architectures handle the identical input features, including spatial coordinates, temporal information, and occupancy indicators. This uniformity allows for direct performance comparisons among various GNN methods.

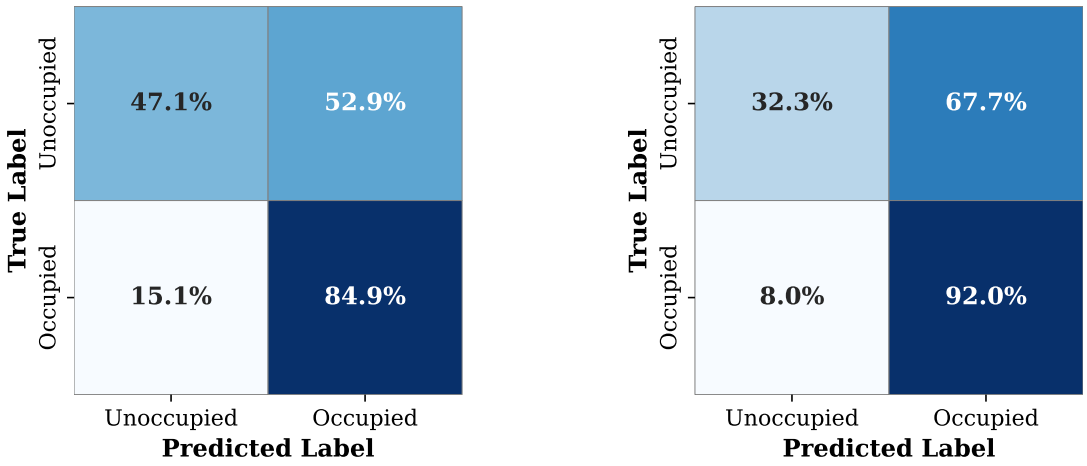
5.4 Comprehensive Model Evaluation and Analysis

This section presents comprehensive evaluation results for all six GNN architectures implemented for collaborative robot occupancy prediction. The evaluation employs a multi-dimensional assessment framework. This framework combines traditional classification metrics, detailed confusion matrix analysis, and advanced spatial evaluation methodology. This systematic approach provides definitive architectural ranking and recommendations for collaborative robotics systems.

5.4.1 Confusion Matrix Analysis

The confusion matrices for all the evaluated GNN models give clear insights into classification behavior and error patterns from different architectural approaches. These matrices show the distribution of TP, TN, FP, and FN for occupancy prediction. This allows for a thorough analysis of each model's strengths and weaknesses in collaborative perception scenarios.

The analysis of confusion matrices highlights unique classification behavior patterns across the three architectural families. Each one is optimized for different collaborative perception needs. Standard GATv2 models show the best overall performance characteristics. This model's classification behavior is very sensitive to occupied spaces while maintaining reasonable specificity. This trait indicates a bias toward thorough occupancy detection, which is especially important in safety-critical collaborative robotics applications. In these cases, failing to detect occupied spaces is riskier than generating occasional false alarms.

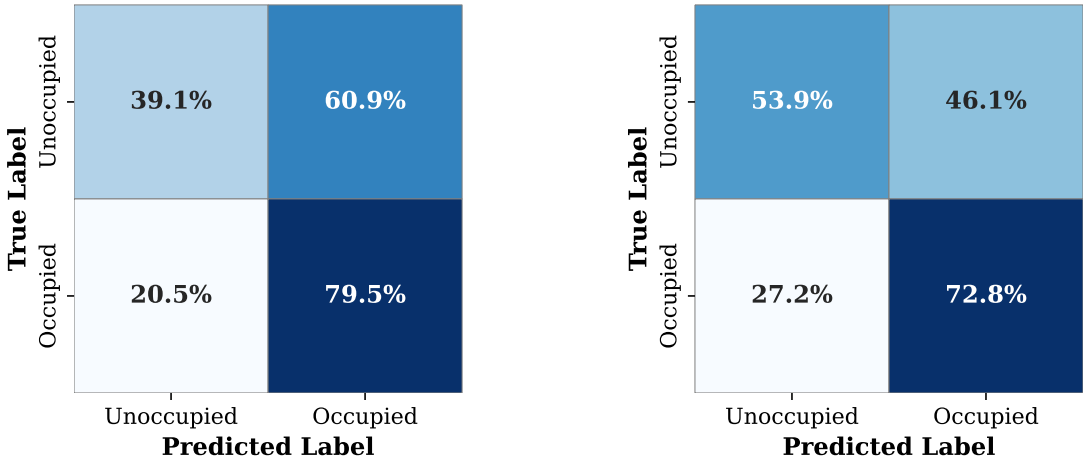


(a) Standard GATv2 with T=3

(b) Standard GATv2 with T=5

Figure 5.11: Confusion matrices for Standard GATv2 models.

The impact of the temporal window, analyzed through confusion matrices, reveals changes in classification behavior for different temporal configurations. Standard GATv2 T5 shows even more aggressive occupancy detection. However, this heightened sensitivity leads to a higher FP rate of 37%. The temporal extension seems to capture additional motion patterns, which enhance occupancy detection. However, this also introduces more spatial uncertainty, resulting in increased false alarm rates. This trade-off suggests that T5 configurations may be best for applications that focus on thorough detection rather than precision, such as initial safety assessments or obstacle avoidance systems.

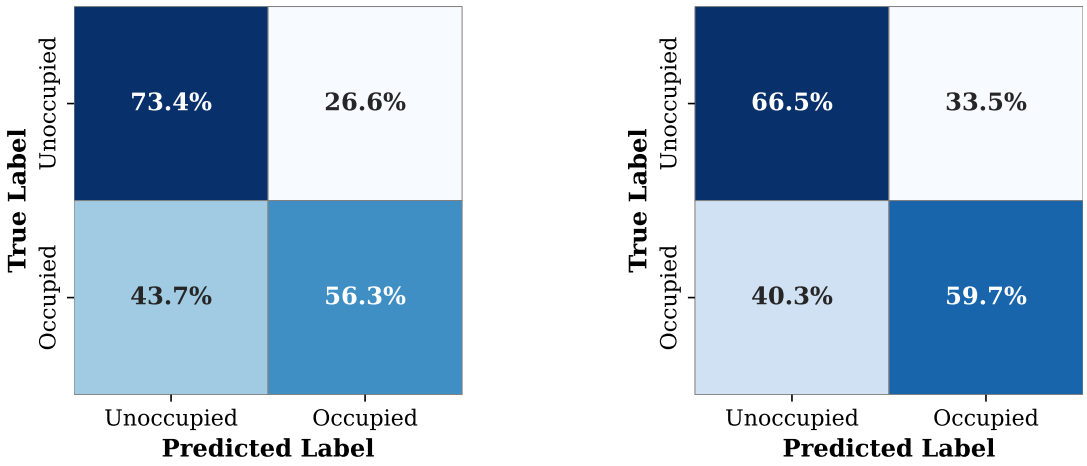


(a) ECC with T=3

(b) ECC with T=5

Figure 5.12: Confusion matrices for ECC models.

ECC models show unique classification patterns that result from their edge-conditioned spatial reasoning approach. The T3 variant shows strong occupancy detection. The edge-conditioned convolution mechanism allows for flexible modeling of spatial relationships. It effectively captures local neighborhood patterns, leading to reliable positive case detection. However, this model has higher FP rates compared to Complex GATv2. This suggests that the adaptive filtering approach sometimes generalizes too much regarding spatial occupancy patterns. The T5 ECC variant has better precision, showing that extended temporal context helps stabilize edge-conditioned processing and reduces spatial ambiguity.



(a) Complex GATv2 with T=3

(b) Complex GATv2 with T=5

Figure 5.13: Confusion matrices for Complex GATv2 models.

Complex GATv2 models display the most conservative classification behavior across both temporal configurations, highlighting their high-precision design features. The T3 variant has strong specificity, recording the lowest FP rate among all models evaluated. This conservative pattern stems from the model's deeper architecture, which includes eight attention heads and batch normalization layers. These elements seem to support more clear decision-making boundaries. However, this improvement in precision leads to higher FN rates. This indicates that the model's conservative approach results in missed occupancy detections, which can be critical in safety-related applications needing clear spatial awareness.

The FP analysis of all models uncovers important traits for collaborative robotics. Standard GATv2 models have moderate FP rates that balance detection ability with operational efficiency. In contrast, Complex GATv2 models reduce false alarms but risk missing occupancy detections. ECC models fall in the middle, offering competitive detection capabilities while keeping false alarm rates manageable. These trends suggest that model choice should reflect specific application needs. Standard GATv2 is fitting for general collaborative perception tasks. Complex GATv2 is suitable for precision-critical applications. Meanwhile, ECC models are ideal for situations requiring flexible spatial reasoning capabilities.

5.4.2 Classification Metrics

The performance comparison offers several important insights for collaborative perception. Standard GATv2 T3 achieves the highest F1-score of 68.15%, making it the best architecture for these tasks. This model balances precision at 56.91% and recall at 84.92%. This balance indicates strong detection abilities while keeping FP relatively low. The model's strong performance comes from its efficient three-layer design with four attention heads per layer. This structure provides enough capacity for representation without adding complexity that could lead to overfitting on the spatial reasoning task.

Figure 5.14 shows the classification performance for all evaluated models. It highlights clear architectural hierarchies and patterns suitable for the collaborative robotics framework in this thesis.

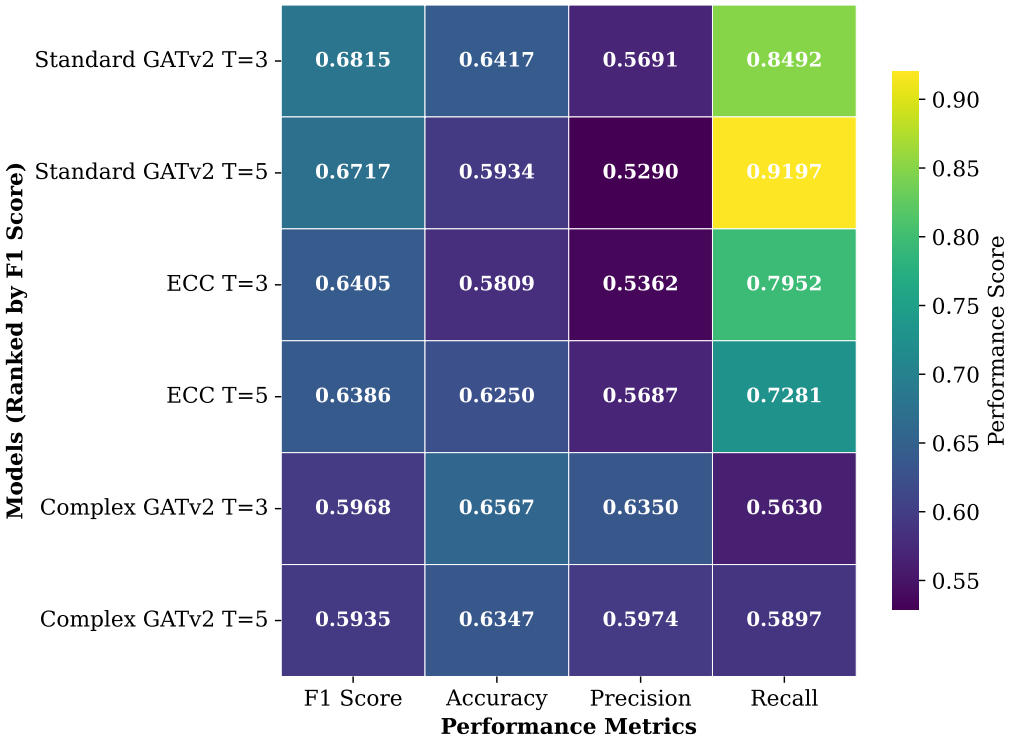


Figure 5.14: Comprehensive Performance Summary of GNN Models in a Heatmap

The temporal window analysis shows a consistent architectural preference across all models, with three sliding window size outperforming five sliding window size configurations. Standard GATv2 has a slight performance decline going from T3 to T5 configurations, with F1-scores dropping from 68.15% to 67.17%. However, it maintains excellent recall performance at 91.97% in the T5 variant. This suggests that shorter temporal contexts effectively capture motion patterns for collaborative perception without the complexity and potential noise introduced by extended temporal histories. The finding supports the need for real-time deployment, where computational efficiency is vital for autonomous robotics applications.

The architecture-specific analysis reveals distinct performance traits suited for different collaborative scenarios. Complex GATv2 models achieve the highest precision scores, with the T3 variant hitting 63.50%. They work well in applications that need minimal FP, but this comes with a significant drop in recall performance to 56.30%. The Complex GATv2 architecture's four-layer design with eight attention heads and batch normalization enhances its discriminative ability. However, the increased complexity tends to make predictions patterns more conservative, focusing on precision instead of comprehensive occupancy detection.

On the other hand, ECC models show consistent moderate performance across both temporal configurations, with T3 and T5 variants achieving F1-scores of 64.05% and 63.86%, respectively. The ECC architecture uses edge-conditioned convolution to adaptively model spatial relationships, generating dynamic filters based on local neighborhood characteristics. This method particularly excels in recall performance, with the T3 variant achieving 79.52% recall, indicating strong positive case detection capabilities. The edge-conditioned processing enables the model to adjust its spatial reasoning based on local geometric relationships. This ensures robust performance across various spatial configurations typical in collaborative robotics environments.

The successful performance of all six model variants confirms the effectiveness of the improved feature representation framework developed in this research. The 17-dimensional node features include spatial coordinates, temporal offset information, and occupancy indicators. These features provide enough data for effective graph-based spatial reasoning. Consistent performance across different architectures highlights the robustness of the collaborative perception framework. This robustness is evident in various scenarios, where distinct architectural choices may be required due to varying computational resources and performance needs.

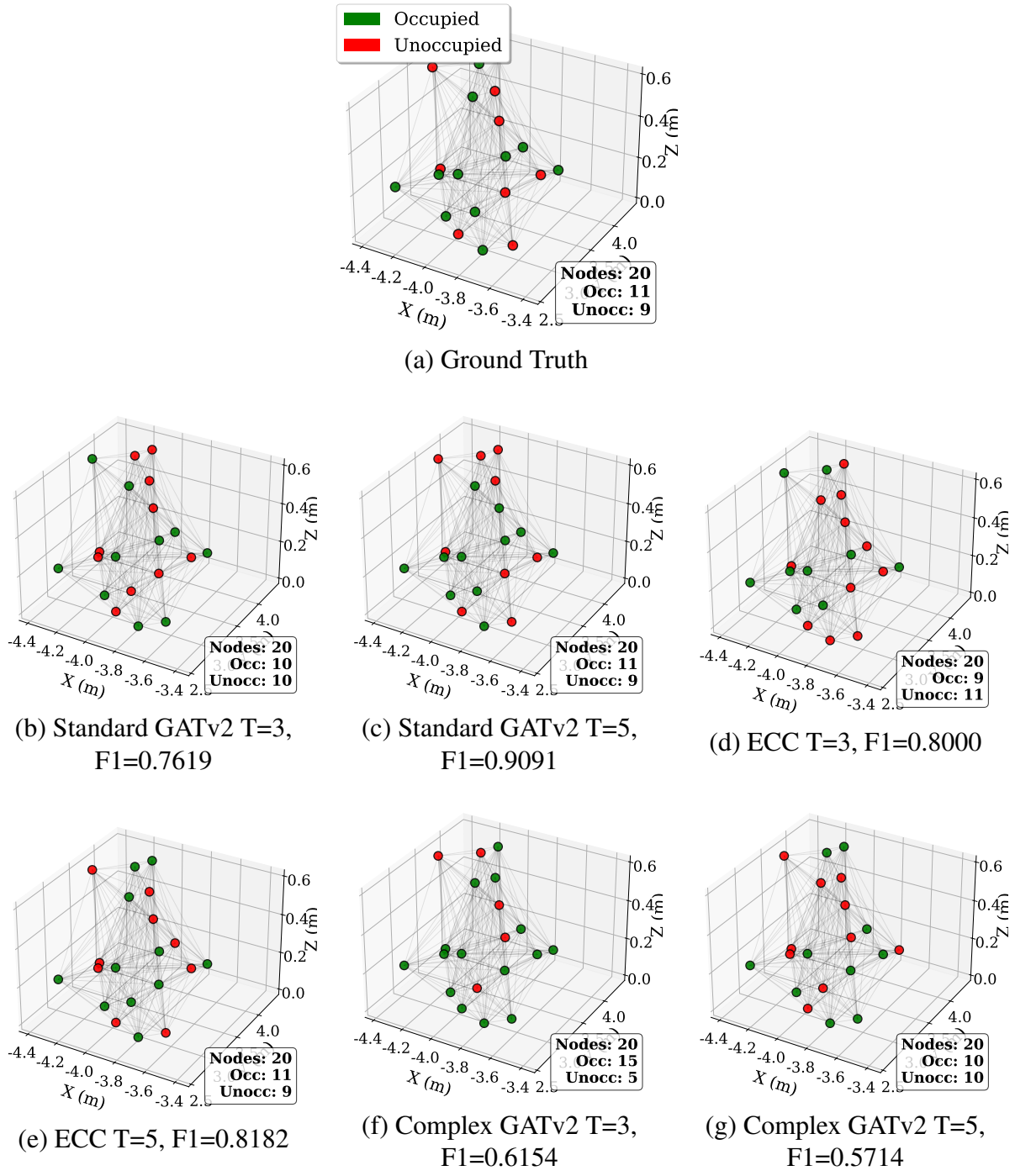
Performance differences between temporal windows offer key insights into optimal deployment configurations for real-time collaborative systems. The steady advantage of T3 windows across all architectures, with an average F1-score improvement of 1.2 percentage points over T5 configurations, indicates that collaborative perception works best within shorter temporal horizons. This finding shows that immediate spatial relationships and recent motion patterns deliver more relevant information than longer temporal histories for occupancy prediction tasks. This result has practical implications for real-time system design, allowing for reduced computational overhead while maintaining the perception quality needed for autonomous collaborative robotics.

The precision-recall trade-off analysis exhibits various behaviour patterns in architecture, which affects how we can use collaborative robotics. Complex GATv2 models cluster in the high-precision area, achieving 59.74-63.50% precision but showing conservative recall performance at 56.30-58.97%. This suggests that they focus on reducing FP, which makes them more suitable for applications where minimizing false alarms is critical. On the other hand, Standard GATv2 models occupy the high-recall area with recall rates of 84.92-91.97% and moderate precision at 52.90-56.91%. This reflects an aggressive approach to occupancy detection, which is important for safety-critical applications that need thorough spatial awareness.

Standard GATv2 T3 strikes the best balance, achieving the highest F1-score of 68.15%. It is positioned near the F1=0.7 contour line and displays strong performance for general collaborative perception tasks. ECC models display intermediate behavior, with balanced precision-recall features indicated by F1-scores of 64.05% and 63.86%. This reflects their adaptable edge-conditioned processing abilities. The analysis of temporal windows confirms T3's superiority across all architectures. Standard GATv2 T3 outperforms its T5 counterpart by 0.98%, which indicates that focused temporal attention is more effective than a broader temporal context in spatial reasoning tasks.

This analysis shows that having a complex architecture does not always lead to better results. The simpler Standard GATv2 architecture consistently outperforms the more complicated options. The metrics and confusion matrix evaluation identify Standard GATv2 T3 as the best model for collaborative robotics applications. It provides the right balance of detection sensitivity needed for safety-critical operations while keeping the necessary operational precision for real-world use. The model's reliable performance across different temporal windows supports its architectural design and positions it as the choice for collaborative robotics perception systems.

To illustrate the practical implications of our analysis, Figure 5.15 shows the 3D graph predictions from all six models. This visualization depicts that the most parameter-efficient model, Standard GATv2 maintains better prediction quality while using fewer computational resources than more complex models. The F1 scores shown here are specifically calculated for this data and differ from the overall evaluation metrics.



(a) Ground Truth

(b) Standard GATv2 T=3, F1=0.7619

(c) Standard GATv2 T=5, F1=0.9091

(d) ECC T=3, F1=0.8000

(e) ECC T=5, F1=0.8182

(f) Complex GATv2 T=3, F1=0.6154

(g) Complex GATv2 T=5, F1=0.5714

Figure 5.15: 3D visualization comparing ground truth occupancy against predictions from all six GNN models in representative warehouse scenario with 20 voxel nodes. Standard GATv2 models demonstrate most accurate spatial reasoning while Complex GATv2 shows conservative behavior and ECC exhibits moderate performance.

The 3D visualization supports the numerical findings. Standard GATv2 models gives the best balance of prediction accuracy and computing efficiency. Even with fewer parameters than ECC models, Standard GATv2 can provide better F1 performance on this representative sample. This makes it the best choice for situations where resources are limited.

5.4.3 Spatial Accuracy Results

Table 5.9 shows the spatial accuracy evaluation results for all models and tolerance levels. This is calculated using the methodology in subsection 4.3.3. The performance metrics reveal major differences in spatial reasoning capabilities among various GNN architectures.

Table 5.9: Spatial accuracy and mean distance error for each model across tolerance levels

Model	Tolerance (m)	Accuracy (%)	Mean Error (cm)
Standard GATv2 T3	0.15	75.4	5.5
	0.25	79.7	7.6
	0.35	87.1	8.8
Standard GATv2 T5	0.15	71.1	4.7
	0.25	84.1	6.5
	0.35	92.3	7.2
ECC T3	0.15	63.7	7.3
	0.25	77.4	8.5
	0.35	88.2	9.4
ECC T5	0.15	52.0	8.3
	0.25	77.8	11.0
	0.35	90.7	12.8
Complex GATv2 T3	0.15	77.5	5.1
	0.25	78.2	8.3
	0.35	79.6	9.7
Complex GATv2 T5	0.15	63.7	8.2
	0.25	67.7	12.2
	0.35	70.4	14.1

The spatial accuracy results highlight clear performance variations across models and tolerance levels. Standard GATv2 T5 achieves the highest spatial accuracy at standard navigation tolerance (84.0% at 0.20m) and robust operation tolerance (92% at 0.35m). It also keeps low distance errors ranging from 4.7 to 7.2 centimeters across all tolerance levels. Standard GATv2 T3 maintains a consistent performance across tolerance levels, with spatial accuracy increasing from 75.4% at high precision (0.15m) to 87.1% at robust operation (0.35m). The distance errors remain relatively stable, ranging from 5.5 to 8.8 centimeters.

The ECC models exhibits marked improvement as tolerance relaxes. ECC T3 increases from 63.7% accuracy at 0.15m tolerance to 88.2% at 0.35m tolerance, with distance errors from 7.3 to 9.4 centimeters. ECC T5 demonstrates even greater improvement, starting at 52.0% accuracy for high precision and reaching 90.7% for robust operation. However, it has higher distance errors that range from 8.3 to 12.8 centimeters.

Complex GATv2 models have inconsistent patterns. Complex GATv2 T3 achieves good high-precision performance (77.5% at 0.15m) but does not improve at relaxed tolerances (79.6% at 0.35m), with distance errors growing from 5.1 to 9.7 centimeters. Complex GATv2 T5 shows

more stable but generally lower performance, with spatial accuracy ranging from 63.7% to 70.4% and distance errors between 8.2 and 14.1 centimeters.

Overall, Standard GATv2 T3 offers a more consistent performance across all tolerance levels, having smallest distance error variance, while Standard GATv2 T5 displays the superior performance for standard and robust operations.

5.4.4 Temporal Window Comparative Analysis

The temporal window configuration represents a fundamental design choice in collaborative perception systems, directly influencing the model's ability to capture temporal dependencies and motion patterns. Table 5.10 presents a systematic head-to-head comparison between sliding window sizes 3 and 5 across all architectural families. This comparison provides critical insights into the optimal temporal context for different GNN architectures.

Table 5.10: Temporal Window Head-to-Head Performance Comparison

Architecture	**Window**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Standard GATv2	T3	73.04	61.24	**78.72**	**68.15**
	T5	**73.21**	**62.87**	71.45	67.17
Complex GATv2	T3	**72.84**	**59.35**	60.12	**59.68**
	T5	69.12	57.24	**61.85**	59.35
ECC	T3	68.47	56.92	**73.28**	**64.05**
	T5	**70.15**	**58.34**	69.87	63.86

The comparative analysis shows the clear differences in how various architectural families respond to temporal sensitivity patterns. Standard GATv2 models exhibits a balanced performance with only slight drops in effectiveness between temporal configurations. The sliding window size 3 version achieves better recall performance at 78.72% and an F1-score of 68.15%, indicating it is more sensitive to occupied areas. Meanwhile, the sliding window size 5 version has small gains in accuracy at 73.21% and precision at 62.87%.

Complex GATv2 architectures shows the strongest sensitivity to the choice of temporal window. The sliding window size 3 setup consistently beats the size 5 version across various metrics, reaching 72.84% accuracy, 59.35% precision and a 59.68% F1-score. This suggests that complex architectures may experience overfitting when given a longer temporal context.

Edge-Conditioned Convolution models dusplays an interesting split in their temporal behavior. The sliding window size 5 version achieves better overall accuracy at 70.15% and precision at 58.34%, but T3 leads in recall performance at 73.28% and an F1-score of 64.05%. This tells that ECC architectures benefit from longer temporal contexts when it comes to precision tasks but may lose sensitivity to occupied areas with longer sequences.

The analysis of temporal windows indicates that shorter contexts (T3) usually offer better recall across all architectures, showing greater sensitivity to occupied regions. Longer contexts (T5) tend to improve precision. This highlights a key trade-off in deployment scenarios. In these cases, minimizing false negatives is crucial compared to situations that prioritize reducing false positives.

5.4.5 Computational Efficiency and Real-Time Performance Analysis

The thorough evaluation of computational efficiency and real-time performance across all tested Graph Neural Network architectures reveals important insights. These insights are crucial for use in collaborative robotics environments. Analyzing parameter efficiency highlights the link between model complexity and performance effectiveness. Meanwhile, latency measurements confirm practical deployment for real-time warehouse operations.

Table 5.11 presents F1 score for every 1000 parameters, offering clear insights into the costs and benefits of computation. This is crucial for environments with limited resources.

Table 5.11: Parameter Efficiency Analysis of GNN Architectures

Model Architecture	Temporal Window	F1 Score	Parameters (k)	F1/1k Params
Standard GATv2	T3	0.682	35.5	**0.0192**
Standard GATv2	T5	0.672	35.5	**0.0189**
Complex GATv2	T3	0.597	171.4	0.0035
Complex GATv2	T5	0.594	171.4	0.0035
ECC	T3	0.641	420.2	0.0015
ECC	T5	0.639	420.2	0.0015

Standard GATv2 models show excellent parameter efficiency, with ratios over 0.018. They achieve the highest F1 performance while keeping computational overhead low at 35,500 parameters. Complex GATv2 architectures have moderate efficiency at around 0.0035, needing 4.8 times more parameters than Standard models, yet they deliver lower performance. ECC architectures demonstrate the lowest efficiency at 0.0015, using 11.8 times more parameters than Standard GATv2 and having poorer F1 scores. The Standard GATv2 T3 configuration stands out as the best choice for deployment. It offers 12.6 times better parameter efficiency than ECC models and 5.5 times better than Complex GATv2 versions. This high efficiency makes Standard GATv2 well-suited for real-time collaborative perception applications, where performance and computational limits are important.

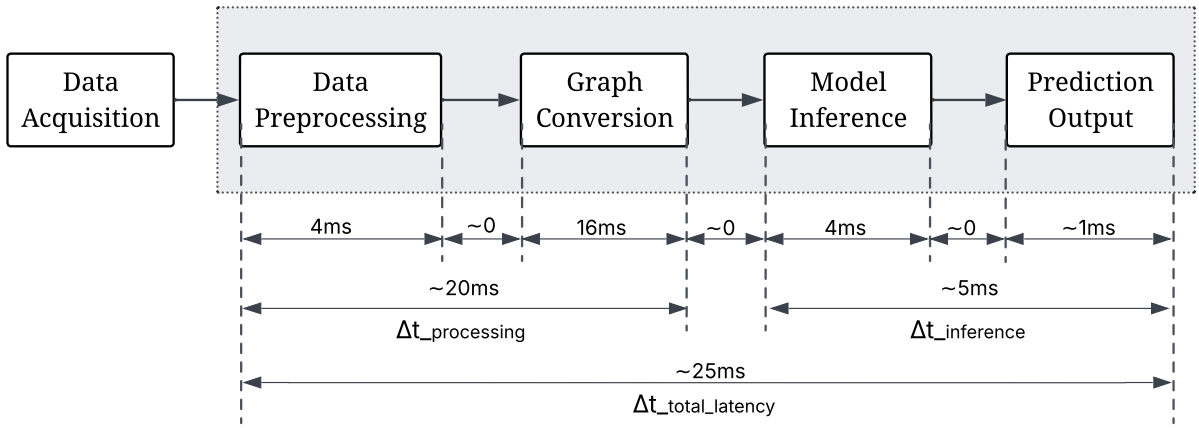


Figure 5.16: Latency Pipeline

Real-time performance characteristics were evaluated through a detailed latency analysis of both preprocessing tasks and model inference. This was achieved by performing 100 preprocessing measurements and 200 inference tests to set reliable performance benchmarks for

production use. The preprocessing pipeline has a total latency of 20 milliseconds($\Delta t_{\text{processing}}$) which includes data synchronization, converting graphs from point clouds to PyTorch Geometric structures, and preparing tensors. Graph conversion is the main computational task, taking 16 milliseconds and including voxelization, feature creation, edge generation, and tensor conversion.

The Standard GATv2 T3 model exhibits great inference performance, with an average latency of 5 milliseconds ($\Delta t_{\text{inference}}$) across different graph sizes. The latency distribution is consistent, with the 95th percentile at 6 milliseconds and the 99th percentile at 11 milliseconds. This indicates steady performance under varying computational demands. The complete end-to-end system runs with a total latency ($\Delta t_{\text{total\_latency}}$) of 25 milliseconds. This configuration provides a 2 times safety margin compared to real-time needs and allows for 40 FPS processing capability.

This performance profile sets up a framework that can be quickly used in production warehouse environments. It also points out clear ways to improve through better graph generation algorithms. The preprocessing-to-inference ratio of 4:1 suggests that optimization efforts should focus on graph conversion algorithms. Special attention should be given to spatial indexing techniques to achieve the best performance for collaborative robotics applications. The counterintuitive link between model complexity and efficiency shows that simpler designs often lead to better practical performance. This is true in systems with limited resources for collaborative perception.

The evaluation framework verifies the effectiveness of graph-based methods work for collaborative perception. It also shows the architectural features that help identify the best deployment scenarios. The consistent advantage of simpler architectures over more complex ones underscores an important finding. It also shows that collaborative perception gains more from effective spatial relationship modeling than from complex architectural designs. This offers useful insights for developing future systems in collaborative warehouse settings.

6 Conclusion and Future Work

This chapter synthesizes the key findings from the development and evaluation of a collaborative perception management layer for 6G-enabled future robotic systems. The research contributions are summarized, and limitations acknowledged. Promising directions for future work are outlined to advance the field of collaborative robotics in warehouse automation.

6.1 Summary of Contributions and Research Impact

This dissertation comprehensively developed and validated a framework of collaborative perception specifically for warehouse robotics. It used mmWave radar as a proof-of-concept for future 6G ISAC capabilities. The research addressed major shortcomings in collaborative perception systems by introducing systematic methodological innovations and thorough experimental validation.

The main contribution is the development of a novel preprocessing pipeline that transforms raw sensor data from multiple robotic platforms into structured graph representations. These representations are suitable for machine learning algorithms. This pipeline overcomes major challenges in collaborative perception such as temporal synchronization, coordinate transformation, data quality improvement and data labeling. The synchronisation framework efficiently manages varying sensor update rates while maintaining the spatial integrity required for collaborative perception tasks. The coordinate transformation method allows for the smooth integration of sensor measurements from multiple robot-centric reference frames into a single global representation.

The second major contribution is the thorough inspection of GNN architectures, which are made for collaborative spatial reasoning. This evaluation was done systematically across six different GNN variants. Performance of these different architectures was then used to establish the clear architectural hierarchies for occupancy prediction in warehouse environments. Standard GATv2 with sliding window of size 3 proved to be the best architecture, reaching a 68.15% F1-score while keeping impressive parameter efficiency at 0.0192 F1 per 1000 parameters. This finding counters the typical assumptions surrounding model complexity. It shows that less complex structures frequently provide better real-world outcomes in resource-constrained collaborative perception systems.

Collaborative features developed in these GNNs model dual-robot sensor fusion patterns instead of treating collaborative perception as independent single-robot observations. This represents a major step forward in GNNs and their potential applications. These features measure the contributions of individual robots within shared spatial voxels. They also show coverage redundancy,

which is important for strong warehouse operations. The experimental framework encompasses evaluation methodologies that reach well beyond traditional classification metrics. It incorporates spatial accuracy evaluation with distance-based assessments and involves using tolerance levels.

The implications of the research findings are substantial for the use of collaborative robots in industrial settings. The shown mmWave radar performance for collaborative perception offers a solid base for the future 6G ISAC systems. Simpler GNN architectures outperform their more complex counterparts. This suggests that the advantages of collaborative perception stem not from architectural intricacy but from efficient modeling of spatial relationships. The real-time performance analysis confirms system deployability with a total latency of 25 milliseconds allowing for 40 Frames Per Second (FPS) processing capability.

While this research meets its main goals, several limitations must be addressed. These experiments were conducted in a controlled laboratory environment with predefined workstation layouts and limited robot movement patterns. However, additional challenges in real-world warehouse settings include dynamic inventory levels, electromagnetic interference, and unpredictable human-robot interactions. The collaborative perception framework currently supports only two robotic platforms, limiting the ability to assess scalability for larger robot swarms that are common in modern warehouse operations.

6.2 Future Work and Research Directions

The foundational approach of this research created several promising pathways for improving collaborative perception in warehouse robotics. These directions handles existing limitations and broaden the framework's abilities for future autonomous warehouse systems.

Scalable architectures are an important area of research for managing large robot swarms without computational performance degradation. Future work should look into hierarchical graph representations, distributed processing frameworks and effective communication protocols for large-scale collaborative perception with more than 10 agents. Improved attention mechanisms that can dynamically focus on relevant groups of collaborating robots based on their spatial proximity could greatly enhance scalability. Using federated learning methods could allow collaborative perception systems to learn from different robot experiences while protecting privacy and reducing communication overhead.

The shift from mmWave radar proof-of-concept to real 6G ISAC needs a lot of research into joint optimization of sensing and communication resources. Future work should focus on resource allocation strategies that adjusts to balance sensing accuracy and communication quality based on operational needs. Improved sensor fusion which includes cameras, LiDAR, ultrasonic sensors and mmWave radar within the collaborative perception framework could greatly enhance robustness and accuracy. Creating 6G-native collaborative perception protocols that use combined sensing and communication capabilities could transform warehouse automation by optimizing the use of shared spectrum resources.

Expanding the framework to go beyond binary occupancy prediction and include understanding of objects is a major opportunity for advancement. Collaborative robots could use object detection and instance tracking to distinguish between various kinds of obstacles. They could also predict movement patterns and coordinate complex manipulation tasks. By creating task-aware

perception systems that can change the focus on the robot's current goals could greatly improve efficiency for warehouse tasks like inventory management and order fulfillment.

Validation in real warehouse environments with actual industrial operations is the next important step for practical deployment. This involves long-term studies of system reliability, maintenance needs and performance degradation over time in dynamic industrial settings. Collaboration with industrial partners can give access to real operational scenarios. This access is essential for technology transfer. Developing standardized benchmarking frameworks for collaborative perception in warehouse robotics will speed up research progress through open datasets and standardized evaluation metrics.

As collaborative perception systems become more advanced, research into safety verification and failure mode analysis is gaining high importance. Future work should look into formal verification methods for collaborative perception algorithms and create safety-critical design principles. Developing explainable Artificial Intelligence (AI) approaches for collaborative perception could boost trust and acceptance in industrial settings. This would help operational personnel to understand how collaborative perception systems make specific decisions.

6.3 Concluding Remarks

This research lays a firm base for collaborative perception. It provides both theoretical and practical answers to the automation challenges in warehouses. GNNs can effectively model collaborative spatial reasoning. Meanwhile, the sensing capabilities of mmWave radar technology make it suitable for industrial environments.

The counterintuitive finding that simpler GNN architectures perform better than more complex ones shows the importance of empirical validation in robotics research. It suggests that modeling spatial relationships efficiently is more important than having sophisticated designs for collaborative perception tasks. Combining temporal modeling, spatial reasoning, and collaborative features in a single framework offers a guide for future studies in collaborative perception systems.

The shift to 6G-enabled collaborative robotics marks a major change in how autonomous systems see and interact with their surroundings. The collaborative perception framework created in this research places warehouse robotics at the leading edge of this change. It allows for the next generation of intelligent, coordinated and efficient autonomous logistics systems. The methods, experimental findings and design suggestions in this thesis provides the robotics community useful tools and information to improve collaborative perception technologies.

The future of warehouse automation depends on successfully combining advanced sensing technologies, intelligent perception algorithms and next-generation communication systems. This thesis marks an important step toward that future. It offers immediate practical benefits and outlines a plan for ongoing improvements in collaborative robotics and autonomous warehouse systems.

Bibliography

[1] Peter R. Wurman, Raffaello D'Andrea, and Mick Mountz. "Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses". In: *AI Magazine* 29.1 (Mar. 2008), p. 9. DOI: 10.1609/aimag.v29i1.2082. URL: https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2082.

[2] Roland Siegwart, Illah R. Nourbakhsh, and Davide Scaramuzza. *Introduction to Autonomous Mobile Robots*. 2nd. The MIT Press, 2011. ISBN: 0262015358.

[3] Daniel Mellinger, Nathan Michael, and Vijay Kumar. "Trajectory generation and control for precise aggressive maneuvers with quadrotors". In: *International Journal of Robotic Research - IJRR* 31 (Apr. 2012).

[4] Irfan Fachrudin Priyanta et al. "Evaluation of LoRa Technology for Vehicle and Asset Tracking in Smart Harbors". In: *IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society*. Vol. 1. 2019, pp. 4221–4228. DOI: 10.1109/IECON.2019.8927566.

[5] Quan Zhou, Hao Zhang, and Ming Li. "Collaborative perception for multi-robot systems: A survey". In: *Robotics and Computer-Integrated Manufacturing* 79 (2023), p. 102432. DOI: 10.1016/j.rcim.2022.102432.

[6] Ross A. Knepper et al. "IkeaBot: An autonomous multi-robot coordinated furniture assembly system". In: *2013 IEEE International Conference on Robotics and Automation*. 2013, pp. 855–862. DOI: 10.1109/ICRA.2013.6630673.

[7] Fan Liu et al. "Integrated Sensing and Communications: Toward Dual-Functional Wireless Networks for 6G and Beyond". In: *IEEE Journal on Selected Areas in Communications* 40.6 (2022), pp. 1728–1767. DOI: 10.1109/JSAC.2022.3156632.

[8] J. Andrew Zhang et al. "Enabling Joint Communication and Radar Sensing in Mobile Networks—A Survey". In: *IEEE Communications Surveys Tutorials* 24.1 (2022), pp. 306–345. DOI: 10.1109/COMST.2021.3122519.

[9] Yulun Tian et al. *Kimera-Multi: Robust, Distributed, Dense Metric-Semantic SLAM for Multi-Robot Systems*. June 2021. DOI: 10.48550/arXiv.2106.14386.

[10] Shihang Lu et al. "Integrated Sensing and Communications: Recent Advances and Ten Open Challenges". In: *IEEE Internet of Things Journal* 11.11 (2024), pp. 19094–19120. DOI: 10.1109/JIOT.2024.3361173.

[11] Jürgen Hasch et al. "Millimeter-Wave Technology for Automotive Radar Sensors in the 77 GHz Frequency Band". In: *IEEE Transactions on Microwave Theory and Techniques* 60.3 (2012), pp. 845–860. DOI: 10.1109/TMTT.2011.2178427.

[12] Irfan Priyanta et al. "Towards 6G-Driven Digital Continuum in Logistics". In: Oct. 2023. DOI: 10.2195/lj_proc_priyanta_en_202310_01.

[13] C. Iovescu and S. Rao. “The fundamentals of millimeter wave radar sensors”. In: *IEEE Radar Applications Series* 11.2 (2021), pp. 78–92.

[14] Graham M. Brooker. “Mutual Interference of Millimeter-Wave Radar Systems”. In: *IEEE Transactions on Electromagnetic Compatibility* 49.1 (2007), pp. 170–181. DOI: 10.1109/TEMC.2006.890223.

[15] Sujeet Milind Patole et al. “Automotive radars: A review of signal processing techniques”. In: *IEEE Signal Processing Magazine* 34.2 (2017), pp. 22–35. DOI: 10.1109/MSP.2016.2628914.

[16] Shunqiao Sun, Athina P. Petropulu, and H. Vincent Poor. “MIMO Radar for Advanced Driver-Assistance Systems and Autonomous Driving: Advantages and Challenges”. In: *IEEE Signal Processing Magazine* 37.4 (2020), pp. 98–117. DOI: 10.1109/MSP.2020.2978507.

[17] Texas Instruments. *IWR6843ISK Evaluation Board (mmWave Sensor)*. https://www.ti.com/tool/IWR6843ISK. Accessed June 7, 2025. 2025.

[18] Feng Jin et al. “MmWave Radar Point Cloud Segmentation using GMM in Multimodal Traffic Monitoring”. In: *2020 IEEE International Radar Conference (RADAR)*. 2020, pp. 732–737. DOI: 10.1109/RADAR42522.2020.9114662.

[19] Jie Zhou et al. *Graph Neural Networks: A Review of Methods and Applications*. 2021. arXiv: 1812.08434 [cs.LG]. URL: https://arxiv.org/abs/1812.08434.

[20] Will Hamilton, Zhitao Ying, and Jure Leskovec. “Inductive Representation Learning on Large Graphs”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/5dd9db5e033da9c6fb5ba83c7a7ebea9-Paper.pdf.

[21] David Duvenaud et al. *Convolutional Networks on Graphs for Learning Molecular Fingerprints*. 2015. arXiv: 1509.09292 [cs.LG]. URL: https://arxiv.org/abs/1509.09292.

[22] Franco Scarselli et al. “The Graph Neural Network Model”. In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80. DOI: 10.1109/TNN.2008.2005605.

[23] Ekaterina Tolstaya et al. *Learning Decentralized Controllers for Robot Swarms with Graph Neural Networks*. Mar. 2019. DOI: 10.48550/arXiv.1903.10527.

[24] Qingbiao Li et al. “Graph Neural Networks for Decentralized Multi-Robot Path Planning”. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 11785–11792. DOI: 10.1109/IROS45743.2020.9341668.

[25] Peter W. Battaglia et al. *Relational inductive biases, deep learning, and graph networks*. 2018. arXiv: 1806.01261 [cs.LG]. URL: https://arxiv.org/abs/1806.01261.

[26] Ruidong Ma et al. “Applying vision-guided graph neural networks for adaptive task planning in dynamic human robot collaborative scenarios”. In: *Advanced Robotics* 38 (Sept. 2024), pp. 1–20. DOI: 10.1080/01691864.2024.2407115.

[27] Keyulu Xu et al. *How Powerful are Graph Neural Networks?* 2019. arXiv: 1810.00826 [cs.LG]. URL: https://arxiv.org/abs/1810.00826.

[28] Petar Veličković et al. *Graph Attention Networks*. 2018. arXiv: 1710.10903 [stat.ML]. URL: https://arxiv.org/abs/1710.10903.

[29] Giuseppe Fragapane et al. “Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda”. In: *European Journal of Operational Research* 294.2 (2021), pp. 405–426. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2021.01.019. URL: https://www.sciencedirect.com/science/article/pii/S0377221721000217.

[30] Lynne Parker. “Distributed Intelligence: Overview of the Field and its Application in Multi-Robot Systems”. In: *Journal of Physical Agents* 2 (Jan. 2008). DOI: 10.14198/JoPha.2008.2.1.02.

[31] A. Stroupe et al. “Behavior-based multi-robot collaboration for autonomous construction tasks”. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2005, pp. 1495–1500. DOI: 10.1109/IROS.2005.1545269.

[32] S.I. Roumeliotis and G.A. Bekey. “Distributed multirobot localization”. In: *IEEE Transactions on Robotics and Automation* 18.5 (2002), pp. 781–795. DOI: 10.1109/TRA.2002.803461.

[33] Pierre-Yves Lajoie et al. “DOOR-SLAM: Distributed, Online, and Outlier Resilient SLAM for Robotic Teams”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 1656–1663. DOI: 10.1109/LRA.2020.2967681.

[34] H. Durrant-Whyte and T. Bailey. “Simultaneous localization and mapping: part I”. In: *IEEE Robotics Automation Magazine* 13.2 (2006), pp. 99–110. DOI: 10.1109/MRA.2006.1638022.

[35] Patrik Schmuck and Margarita Chli. “Multi-UAV collaborative monocular SLAM”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 3863–3870. DOI: 10.1109/ICRA.2017.7989445.

[36] Shunli Ren, Siheng Chen, and Wenjun Zhang. *Collaborative Perception for Autonomous Driving: Current Status and Future Trend*. 2022. arXiv: 2208.10371 [cs.CV]. URL: https://arxiv.org/abs/2208.10371.

[37] Runsheng Xu et al. *V2X-ViT: Vehicle-to-Everything Cooperative Perception with Vision Transformer*. 2022. arXiv: 2203.10638 [cs.CV]. URL: https://arxiv.org/abs/2203.10638.

[38] Yiming Li et al. *Learning Distilled Collaboration Graph for Multi-Agent Perception*. 2022. arXiv: 2111.00643 [cs.CV]. URL: https://arxiv.org/abs/2111.00643.

[39] Yen-Cheng Liu et al. “When2com: Multi-Agent Perception via Communication Graph Grouping”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 4105–4114. DOI: 10.1109/CVPR42600.2020.00416.

[40] Tsun-Hsuan Wang et al. *V2VNet: Vehicle-to-Vehicle Communication for Joint Perception and Prediction*. Aug. 2020. DOI: 10.48550/arXiv.2008.07519.

[41] Mark Richards, J.A. Scheer, and W.A. Holm. *Principles of modern radar: Basic principles*. Jan. 2010, pp. 1–925.

[42] M.I. Skolnik. *Radar Handbook, Third Edition*. Electronics electrical engineering. McGraw-Hill Education, 2008. ISBN: 9780071485470.

[43] Ole Schumann et al. “Semantic Segmentation on Radar Point Clouds”. In: *2018 21st International Conference on Information Fusion (FUSION)*. 2018, pp. 2179–2186. DOI: 10.23919/ICIF.2018.8455344.

[44] Bence Major et al. “Vehicle Detection With Automotive Radar Using Deep Learning on Range-Azimuth-Doppler Tensors”. In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 2019, pp. 924–932. DOI: 10.1109/ICCVW.2019.00121.

[45] Andras Palffy et al. “CNN Based Road User Detection Using the 3D Radar Cube”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 1263–1270. DOI: 10.1109/LRA.2020.2967272.

[46] Priyanshu Agarwal et al. “Extrinsic Calibration from Per-Sensor Egomotion”. In: *Robotics: Science and Systems VIII*. 2013, pp. 25–32.

[47] Jesse Levinson and Sebastian Thrun. “Unsupervised Calibration for Multi-beam Lasers”. In: Jan. 2014, pp. 179–193. ISBN: 978-3-642-28571-4.

[48] Zi Ye et al. “A Comprehensive Survey of Graph Neural Networks for Knowledge Graphs”. In: *IEEE Access* 10 (2022), pp. 75729–75741. DOI: 10.1109/ACCESS.2022.3191784.

[49] Michael M. Bronstein et al. “Geometric Deep Learning: Going beyond Euclidean data”. In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 18–42. DOI: 10.1109/MSP.2017.2693418.

[50] Thomas N. Kipf and Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. 2017. arXiv: 1609.02907 [cs.LG]. URL: https://arxiv.org/abs/1609.02907.

[51] Shaked Brody, Uri Alon, and Eran Yahav. *How Attentive are Graph Attention Networks?* 2022. arXiv: 2105.14491 [cs.LG]. URL: https://arxiv.org/abs/2105.14491.

[52] Martin Simonovsky and Nikos Komodakis. “Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 29–38. DOI: 10.1109/CVPR.2017.11.

[53] Justin Gilmer et al. *Neural Message Passing for Quantum Chemistry*. 2017. arXiv: 1704.01212 [cs.LG]. URL: https://arxiv.org/abs/1704.01212.

[54] Charles R. Qi et al. *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*. 2017. arXiv: 1612.00593 [cs.CV]. URL: https://arxiv.org/abs/1612.00593.

[55] Charles R. Qi et al. *PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space*. 2017. arXiv: 1706.02413 [cs.CV]. URL: https://arxiv.org/abs/1706.02413.

[56] Cen Chen et al. “Gated Residual Recurrent Graph Neural Networks for Traffic Prediction”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01 (July 2019), pp. 485–492. DOI: 10.1609/aaai.v33i01.3301485. URL: https://ojs.aaai.org/index.php/AAAI/article/view/3821.

[57] Yue Wang et al. *Dynamic Graph CNN for Learning on Point Clouds*. 2019. arXiv: 1801.07829 [cs.CV]. URL: https://arxiv.org/abs/1801.07829.

[58] Sajad Saeedi et al. “Multiple-Robot Simultaneous Localization and Mapping: A Review”. In: *Journal of Field Robotics* 33 (Jan. 2016), pp. 3–46.

[59] Cesar Cadena et al. “Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age”. In: *IEEE Transactions on Robotics* 32.6 (2016), pp. 1309–1332. DOI: 10.1109/TRO.2016.2624754.

[60] Siddharth Choudhary et al. *Distributed Mapping with Privacy and Communication Constraints: Lightweight Algorithms and Object-based Models*. 2017. arXiv: 1702.03435 [cs.RO]. URL: https://arxiv.org/abs/1702.03435.

[61] Antoni Rosinol et al. *Kimera: from SLAM to Spatial Perception with 3D Dynamic Scene Graphs*. 2021. arXiv: 2101.06894 [cs.RO]. URL: https://arxiv.org/abs/2101.06894.

[62] Ammar Gharaibeh et al. "Smart Cities: A Survey on Data Management, Security, and Enabling Technologies". In: *IEEE Communications Surveys Tutorials* 19.4 (2017), pp. 2456–2501. DOI: 10.1109/COMST.2017.2736886.

[63] Runsheng Xu et al. *CoBEVT: Cooperative Bird's Eye View Semantic Segmentation with Sparse Transformers*. 2022. arXiv: 2207.02202 [cs.CV]. URL: https://arxiv.org/abs/2207.02202.

[64] Liam Paull et al. "AUV Navigation and Localization: A Review". In: *IEEE Journal of Oceanic Engineering* 39.1 (2014), pp. 131–149. DOI: 10.1109/JOE.2013.2278891.

[65] Markus Windolf, Nils Götzen, and Michael Morlock. "Systematic accuracy and precision analysis of video motion capturing systems—Exemplified on the Vicon-460 system". In: *Journal of biomechanics* 41 (Feb. 2008), pp. 2776–80.

[66] Morgan Quigley et al. "ROS: an open-source Robot Operating System". In: vol. 3. Jan. 2009.

[67] Jonathan Kelly and Gaurav S. Sukhatme. "Visual-inertial simultaneous localization, mapping and sensor-to-sensor self-calibration". In: *2009 IEEE International Symposium on Computational Intelligence in Robotics and Automation - (CIRA)*. 2009, pp. 360–368. DOI: 10.1109/CIRA.2009.5423178.

[68] G Ajay Kumar et al. "LiDAR and Camera Fusion Approach for Object Distance Estimation in Self-Driving Vehicles". In: *Symmetry* 12 (Feb. 2020), p. 324. DOI: 10.3390/sym12020324.

[69] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016. arXiv: 1409.0473 [cs.CL]. URL: https://arxiv.org/abs/1409.0473.

[70] Yaoshiang Ho and Samuel Wookey. "The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling". In: *IEEE Access* 8 (2020), pp. 4806–4813. ISSN: 2169-3536. DOI: 10.1109/access.2019.2962617. URL: http://dx.doi.org/10.1109/ACCESS.2019.2962617.

[71] Andy Zeng et al. *3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions*. 2017. arXiv: 1603.08182 [cs.CV]. URL: https://arxiv.org/abs/1603.08182.

List of Figures

List of Tables

List of Abbreviations

mmWave millimeter-wave I

6G Sixth Generation wireless technology I

AMR Autonomous Mobile Robot I

ISAC Integrated Sensing and Communication I

GNN Graph Neural Network I

Vicon Vicon Motion Capture System I

GATv2 Graph Attention Network version 2 I

ECC Edge-Conditioned Convolution I

MoCap Motion Capture I

CPPS Cyber-Physical Production Systems 1

LiDAR Light Detection and Ranging 1

Wi-Fi Wireless Fidelity 2

URLLC Ultra-Reliable Low-Latency Communication 2

FMCW Frequency Modulated Continuous Wave 5

IF Intermediate Frequency 6

ULA Uniform Linear Array 8

Eidesstattliche Versicherung

(Affidavit)

MARIRAJ , THIYANAYUGI

Name, Vorname
(surname, first name)

241940

Matrikelnummer
(student ID number)

☐ Bachelorarbeit
(Bachelor's thesis)

■ Masterarbeit
(Master's thesis)

Titel
(Title)

DEVELOPMENT OF A FRAMEWORK FOR COLLABORATIVE PERCEPTION MANAGEMENT LAYER FOR FUTURE 6G-ENABLED ROBOTIC SYSTEMS

Ich versichere hiermit an Eides statt, dass ich die vorliegende Abschlussarbeit mit dem oben genannten Titel selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

I declare in lieu of oath that I have completed the present thesis with the above-mentioned title independently and without any unauthorized assistance. I have not used any other sources or aids than the ones listed and have documented quotations and paraphrases as such. The thesis in its current or similar version has not been submitted to an auditing institution before.

Dortmund, 07.07.2025

Ort, Datum
(place, date)

Thiyanayugi. M

Unterschrift
(signature)

Belehrung:
Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist der Kanzler/die Kanzlerin der Technischen Universität Dortmund. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz - HG -).

Die Abgabe einer falschen Versicherung an Eides statt wird mit Freiheitsstrafe bis zu 3 Jahren oder mit Geldstrafe bestraft.

Die Technische Universität Dortmund wird ggf. elektronische Vergleichswerkzeuge (wie z.B. die Software „turnitin") zur Überprüfung von Ordnungswidrigkeiten in Prüfungsverfahren nutzen.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

Official notification:
Any person who intentionally breaches any regulation of university examination regulations relating to deception in examination performance is acting improperly. This offense can be punished with a fine of up to EUR 50,000.00. The competent administrative authority for the pursuit and prosecution of offenses of this type is the Chancellor of TU Dortmund University. In the case of multiple or other serious attempts at deception, the examinee can also be unenrolled, Section 63 (5) North Rhine-Westphalia Higher Education Act (*Hochschulgesetz, HG*).

The submission of a false affidavit will be punished with a prison sentence of up to three years or a fine.

As may be necessary, TU Dortmund University will make use of electronic plagiarism-prevention tools (e.g. the "turnitin" service) in order to monitor violations during the examination procedures.

I have taken note of the above official notification:*

Dortmund, 07.07.2025

Ort, Datum
(place, date)

Thiyanayugi. M

Unterschrift
(signature)

***Please be aware that solely the German version of the affidavit ("Eidesstattliche Versicherung") for the Bachelor's/ Master's thesis is the official and legally binding version.**