Pedestrian Trajectory Prediction: Multimodal Sensor Fusion for Road Crossing

University of Turku
Department of Computing
Robotics and Autonomous Systems
Master of Science in Technology Thesis
December 2025
Danial Moradisabzevar

Supervisors:
Tomi Westerlund
Tomasz Kucner

UNIVERSITY OF TURKU
Department of Computing
Robotics and Autonomous Systems

Danial Moradisabzevar: Pedestrian Trajectory Prediction: Multimodal Sensor Fusion

for Road Crossing

Master of Science in Technology Thesis, 50 p.

December 2025

Pedestrian safety is one of the most integral research areas in the topic of autonomous vehicles (AVs). Extensive research has been conducted on the topic of predicting pedestrian trajectories, yet most of this research focuses on monosensor modalities. Meanwhile, the researches that focus on multimodal and sensor fusion approaches mainly focus on detecting pedestrians and not predicting their intended trajectory. In this research, we propose a multimodal approach for predicting pedestrian trajectories, which addresses the shortcomings of unimodal trajectory prediction by fusing LiDAR, radar, and camera data to provide a more accurate prediction of pedestrian trajectories. To achieve this, we create a pipeline that receives the inputs of Ouster OS1 LiDAR, Navtech RAS3 Radar, and Intel RealSense D415, temporally synchronizes them, calibrates them in a common coordinate frame, and fuses them into a dataset. Pedestrian annotations are obtained using PointRCNN combined with a nearest-neighbor tracker to assign consistent IDs across frames. We then evaluate state-of-the-art trajectory prediction models on the resulting dataset. Results show that a zero-shot Social-STGCNN baseline yields 0.51 ADE / 0.76 FDE, while a trained Wayformer model achieves 2.95 ADE / 2.31 FDE. [1] [2]

Keywords: Robotics, Sensor Fusion, ROS 2, Pedestrian Safety, LiDAR, radar

---

[1] AI has been used in the writing of this paper for grammar correction and paraphrasing.

[2] The code base of this research can be found at This link

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Ensuring the safety of pedestrians is a fundamental requirement for autonomous vehicles (AVs), which must function reliably and safely in complex urban environments [1]. Among the challenges they face, pedestrians remain one of the most researched and one of the most significant due to their vulnerable size, dynamic motion, and the difficulty of predicting their trajectories [2]. Pedestrians can behave abruptly, unpredictably, and sometimes even irrationally in the case of younger or under-influenced pedestrians. This makes it essential for autonomous vehicles (AVs) to detect and anticipate the movements of pedestrians. The capability of predicting the behavior and trajectory of the pedestrians becomes even more crucial in urban settings, in which human-vehicle interactions are more frequent and often occur in a more complex setting as studied in [2].

Pedestrian trajectory and intention prediction remains an active and challenging area of research, as understanding human motion in complicated urban areas is crucial for the safety of both pedestrians and AVs. Many approaches have been studied during recent years. They range from early statistical methods to advanced deep learning–based frameworks that attempt to capture the social and contextual cues of pedestrian dynamics [3]–[5]. Almost all of the early research has been relying on vision-based approaches, to be specific, monocular cameras [5]–[7] or stereo cam-

era setups that were designed to reconstruct partial 3D information of the needed pedestrian trajectories [8]. While these approaches have demonstrated effectiveness in controlled or well-illuminated environments, their inherent limitations, such as occlusions in the case of monocular cameras, constrain their robustness in real-world deployments.

Monocular methods can be susceptible to occlusions. Occlusions happen when objects, in our case, pedestrians, are partially or fully hidden and covered by surrounding objects, such as, vehicles, bicycles, cars, or other pedestrians. In addition, performance significantly degrades under adverse environmental conditions such as heavy rain, fog, or low-light scenarios or factors that are frequent in real traffic environments [9], [10]. These kinds of limitations make it difficult to ensure reliable pedestrian detection and trajectory prediction across diverse operational domains. These problems resulted in a growing recognition in the research community that relying solely on vision-based modalities is insufficient for building resilient and safety-critical autonomous systems.

To address these research questions, which are the inherent shortcomings of camera-only methods, recent research has increasingly shifted toward multimodal sensor fusion that combines complementary sensors such as LiDAR and radar. LiDAR integration is one of the prominent directions that has been explored in scientific research, where semantic features extracted from images are enriched with the precise geometric depth information that is offered by point clouds. This approach not only improves pedestrian detection accuracy but also enhances the trajectory prediction by capturing both appearance-level cues and 3D spatial relationships, as was explored in [11]. LiDAR sensors provide mostly accurate distance and structural information. This ability makes them a well-suited nominee for constructing detailed 3D maps of the surrounding environment. When this ability of LiDAR is paired with the visual cues received from the camera, the resulting multimodal rep-

resentations allow for more reliable conclusions about pedestrian motion, decision, or trajectory prediction. Especially in scenarios where the visual feed coming from the camera alone is ambiguous due to occlusions or other perspective limitations.

Going beyond LiDAR and camera-only fusion, radar has also emerged as a critical complementary modality, mostly for autonomous vehicle applications in real-life traffic. Radar sensors are less affected by harsh and adverse weather conditions such as fog, heavy rain, and snow or snowstorms. Radars provide reliable measurements at much longer ranges compared to both cameras and LiDAR. This results in radars being able to operate on ranges far beyond the reach of most sensors used in AVs. For instance, [10] introduces a large-scale dataset that incorporates synchronized camera, LiDAR, and radar streams to evaluate perception algorithms under degraded visual conditions, which highlights the unique role of radar in ensuring a robust and reliable perception.

When we combine the strength of each modality, the semantic richness that comes from the camera, geometric perception that comes from LiDAR, the weather and range-resilient measurements from radar, it becomes possible to develop an enhanced perception system that is significantly more reliable and robust across different urban environments compared to unimodal approaches using only mono cameras or LiDAR-only approaches. The integration of multimodality into a framework directly contributes to a safer pedestrian trajectory prediction or intention recognition. This reduces the likelihood of perception failure in safety-critical AV scenarios. Ultimately, this kind of sensor fusion framework represents a crucial step toward bridging the gap between controlled research settings and the unpredictable, dynamic conditions that are encountered in real-world edge-case deployments.

Despite the demonstration of the advantages of multimodal approaches in research and real-life scenarios, the majority of multimodal sensor fusion studies in autonomous driving research and literature have mainly concentrated on general

object detection and tracking tasks rather than directly addressing trajectory prediction problems. Works such as [9], [12] highlight the effectiveness of combining LiDAR, camera, and radar data for 3D object detection and multi-object tracking, showing clear improvements over unimodal baselines. These studies have been critical in the advancement of autonomous perception tasks. This enables more accurate localization for edge applications such as AVs, where they can more accurately detect and localize other vehicles, pedestrians, and static infrastructure in diverse conditions. Yet, their focus has largely remained on answering the question of *"What is where?"*, without extending to the equally critical question of *"where will it go?"*

Meanwhile, in contrast, the trajectory prediction and intention recognition studies have been less worked on in terms of multimodality, meaning the research focused on these areas has mostly been conducted in unimodal sensor areas. The dominant research studies in pedestrian trajectory prediction rely on camera-based datasets, which often use monocular or stereo cameras used for video input to capture pedestrians and their motion patterns. Foundational works such as Social-LSTM [4], which introduced socially aware recurrent models, or early behavioral intention prediction studies [3], primarily utilize RGB video streams. [8] extends the studies that are mainly focused on mono-sensor approaches by reconstructing 3D pedestrian positions using stereo-vision-based trajectory forecasting frameworks. Yet still, it operates exclusively within the domain of cameras. These approaches have yielded valuable insights into pedestrian interaction modeling, but their reliance on camera data makes them susceptible to the same limitations, such as occlusion, lighting dependency, and vulnerability to adverse weather.

The gap between multimodal detection-based research and camera-based research on trajectory prediction studies has created a significant gap in the field of AVs. On the one hand, multimodal datasets and fusion-based algorithms have matured for tasks such as detection and tracking, demonstrating that integrating

different modalities is not only feasible but also a necessary tool for the future of robot perception. On the other side, the trajectory prediction study, which is arguably one of the most safety-critical tasks for AVs, has remained mostly trained to the unimodal area of research, that being vision-only paradigms. As a result, current trajectory prediction models usually fail to fully use the complementary strengths of other sensors, such as LiDAR's geometric data, radar's resilience in harsh weather and its massive range, and camera's semantic richness. To bridge this gap is an essential topic of research for deployable edge devices and AVs. Pedestrian trajectory prediction differs fundamentally from object detection in that it requires modeling both short-term motion dynamics and longer-term behavioral intentions, often under conditions where one sensor's reliability fluctuates. A trajectory prediction system that is based on multimodal sensor fusion field could offer to fill this gap. Such research can offer the robustness of radar in rain or fog, the structural fidelity of LiDAR for accurate localization, and the contextual awareness of cameras for understanding scene semantics and social cues. Nevertheless, a handful of studies have explored this kind of integration. This motivates the need for a dedicated pipeline that not only synchronizes and fuses data from different sensors but also extends their application beyond detection and tracking into the domain of pedestrian intent understanding and their trajectory prediction.

1.2 Problem Statement

The main problem addressed in this thesis is to cover the gap between the multimodal sensor fusion and pedestrian trajectory prediction by **creating a pipeline for obtaining a structured, labeled, and multimodal dataset from the raw sensor data that can serve as a foundation for training and evaluating trajectory prediction models.**

Creating such a platform and thus bridging this gap is not only critical for

pedestrian safety research but also a step toward developing prediction frameworks that are robust, reliable, and safe for deployment in edge systems, which are, in our target case, autonomous driving systems.

This thesis addresses these problems by

- Synchronizing and calibrating heterogeneous sensor inputs.
- Detecting pedestrians and assigning pseudo-labels.
- Formatting the resulting data for use with deep learning models.
- Evaluating trajectory prediction models on the resulting dataset.

In this thesis, we propose a comprehensive pipeline that is designed to fill the aforementioned gaps. Our proposed framework creates a multimodal dataset that expands the limited research done in the area of sensor fusion for pedestrian trajectory prediction. To be more precise, the object of this project is to establish an end-to-end pipeline, which synchronizes, calibrates, and fuses the data received from each sensor, and later on, pseudo-labels that data. By providing this structured approach to creating the desired dataset, the pipeline bridges the gap between the raw multimodal recordings of the sensors and the labeled trajectory dataset ready for supervised learning in downstream applications such as trajectory prediction by the state-of-the-art models.

This pipeline receives the data from three sensors, a camera, a LiDAR, and a radar; each of which is being published as ROS2 topics. As will be discussed in the Methodology section, these data are first subjected to a temporal synchronization process to ensure that measurements captured across different modalities correspond to the same time frames. Second, a spatial calibration stage is applied, where the sensor outputs from the synchronization step are aligned in a unified coordinate system. Once synchronized and calibrated, the resulting data will go through the fusion step, which combines the data received by radar and LiDAR into a unified

point cloud structure. Camera data, being two-dimensional, is preserved and linked to the fused point cloud via shared timestamps, which allows for multimodal cross-referencing in later stages of processing.

The next stage will be the labeling stage. Here, pseudo-labeling is applied to the resulting point cloud dataset from the last section by leveraging state-of-the-art object detection algorithms. This stage lowers the need for manual labeling and, if fine-tuned, can eliminate the need for long and exhaustive manual labeling. Furthermore, a simple tracking labeling is integrated at this stage that will be suitable for trajectory models.

Finally, the dataset generated by the proposed pipeline is used to train and evaluate state-of-the-art trajectory prediction models. In this thesis, we focus particularly on transformer-based architectures and RNN-based sequence models, both of which have demonstrated strong performance in modeling spatiotemporal dependencies in pedestrian motion. We will discuss these types of models in the related works section. By benchmarking these models on the structured dataset, we not only assess the effectiveness of the proposed piepline but also provide insights into the challenges and opportunities that arise when applying trajectory forecasting algorithms to multimodal sensor data. Through this integrated pipeline, the proposed framework demonstrates a novel approach to bridging the gap between raw multimodal recordings and trajectory prediction research, which offers a scalable and reproducible methodology for advancing the field of pedestrian behavior prediction in autonomous driving contexts.

In Fig. 1.1 we can observe an overall of the first two stages. The first stage is responsible for synchronization, calibration, and sensor fusion. The second phase is the Detection or pseudo-labeling phase, in which the generated dataset will be labeled for trajectory prediction algorithms. The last stage is the evaluation of the resulting dataset and its respective ground-truth on the Trajectory prediction

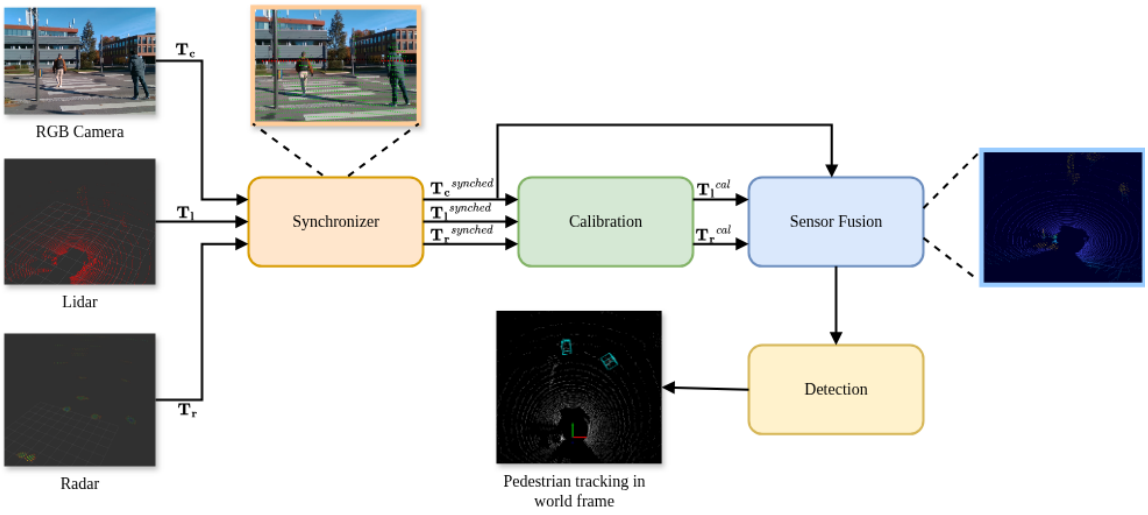

Figure 1.1: Our pipeline: on the left, the raw output of each sensor is presented. The sensor outputs are used as the inputs of the temporal synchronizer node, which aligns all of the messages chronologically. The synchronized sensor messages are then transferred in a common reference frame in the calibration node. The calibrated messages are sent to the sensor fusion node, which combines the point clouds into a single file and saves it along with the synchronized camera message. The calibrated and synchronized point clouds are then processed in the detection phase to be used as ground truth for trajectory prediction models.

model. Each stage will be discussed thoroughly in the methodology section.

1.3 Contributions

This thesis makes the following key contributions to the field of pedestrian behavior prediction and multimodal sensor fusion in autonomous driving research:

- **Multimodal Sensor Data Pipeline**: We developed a complete data preprocessing pipeline for LiDAR, radar, and camera recordings stored in ROS2 format. The pipeline handles synchronization, calibration, and transformation of data into a common reference frame.

- **Pedestrian Annotation and Trajectory Construction**: Labeled pedestrian positions using point cloud. We built a trajectory construction module

that tracks these positions across time to form temporal sequences.

- **Dataset Formatting for Trajectory Prediction**: We structured the trajectory data into input–output sequences compatible with the desired prediction model, which enables downstream training and inference for later models.

- **Initial Model Integration**: Integrated and tested trajectory prediction models using the constructed dataset to evaluate the end-to-end functionality of the pipeline.

1.4 Outline

The remainder of this thesis is organized as follows:

- **Chapter 2 – Related Works**: In Related works, we provide an overview of prior work in pedestrian trajectory prediction and multimodal sensor fusion, which are used for the detection of pedestrians. We will discuss the shortcomings and strengths of each approach.

- **Chapter 3 – System Architecture:** This section will discuss the overall setup of the sensors used in our framework, and also discuss the format of the database and the environment of ROS2 used in the development stage.

- **Chapter 4 – Methodology**: This section will discuss the algorithm and the steps taken to develop the framework in a more detailed manner. We will discuss everything related to the algorithms and the nodes comprising this framework, and furthermore, the relationships between the nodes synchronizing the data.

- **Chapter 5 – Results**: Finally, we will present qualitative and quantitative results of the trajectory prediction model and analyze the performance of the

proposed pipeline on a few selected models, and evaluate them based on FDE and ADE.

- **Chapter 6 – Conclusion and Future Work**: We will summarize the findings, reflect on limitations, and outline potential directions for future improvements.

2 Related Works

Understanding and predicting pedestrian behavior in complex traffic environments is a fundamental component of autonomous driving. In recent years, substantial progress has been made in various subfields supporting this task, including multi-modal sensor fusion, pedestrian detection, and trajectory prediction. Each of these domains tackles a specific aspect of the perception and forecasting pipeline, from acquiring accurate environmental representations to modeling human motion under uncertainty. This section reviews the most relevant prior work across these areas, with a focus on techniques that combine data from camera, LiDAR, and radar sensors, methods for detecting pedestrians from 3D point clouds, and deep learning models for forecasting pedestrian trajectories in dynamic urban scenes.

2.1 Trajectory Prediction

2.1.1 Early Approaches

Trajectory prediction studies focus on the task of predicting the future position of the dynamic agents, such as cars, vehicles, and pedestrians. Early approaches to trajectory prediction were rooted in model-based methods, where motion dynamics were explicitly encoded using mathematical formulations. Among the most widely adopted techniques were the Kalman Filter (KF) and the Extended Kalman Filter (EKF), which provided a probabilistic framework for recursively estimating an

agent's future state [13]. These filters operate by first predicting the next state of the system based on a predefined motion model—commonly constant velocity or constant acceleration, and then updating this prediction using incoming sensor measurements, while accounting for noise and uncertainty in the process. Such recursive estimation made Kalman-based approaches computationally efficient and interpretable, which led to their widespread use in early navigation and tracking systems.

However, these methods are inherently constrained by the assumptions embedded within the motion models. While effective for linear or near-linear dynamics, they struggle to capture the complex, non-linear, and highly interactive nature of pedestrian and vehicle motion in crowded urban environments. Variants such as the Unscented Kalman Filter (UKF) and Particle Filters (PFs) were later developed to extend the applicability of probabilistic filtering to non-linear domains. These approaches allowed for more flexible modeling of uncertainty but often came at the cost of increased computational complexity and still relied heavily on handcrafted models of agent behavior. Despite their limitations, the early adoption of Kalman-based approaches laid the foundation for trajectory prediction research, demonstrating the importance of sequential estimation and uncertainty modeling. Moreover, these methods highlighted the critical challenge that persists to this day: accurately predicting the trajectories of agents whose behavior is influenced not only by individual dynamics but also by interactions with other agents and the environment. This realization set the stage for the transition from purely model-based statistical methods toward data-driven approaches, where machine learning techniques began to play a central role in capturing the rich and often unpredictable patterns of human and vehicle motion.

2.1.2 Deep Learning Approaches

With the rapid advancement of deep learning, trajectory prediction research shifted from handcrafted probabilistic models toward data-driven sequence modeling approaches. Recurrent Neural Networks (RNNs) and their variants, particularly Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), became widely adopted due to their ability to capture temporal dependencies in sequential data. These models offered a significant improvement over traditional filtering methods by learning motion patterns directly from data, rather than relying on predefined motion models. For instance, in [14], a data-driven framework was introduced for forecasting pedestrian crowd movements by first transforming continuous GPS trajectories into discrete spatial cell sequences and then training an RNN with GRU units. This representation enabled the model to capture large-scale pedestrian flow dynamics, making it suitable for analyzing crowded urban environments where handcrafted models fail to generalize.

A particularly influential development came from Alahi et al. in [4], where the Social-LSTM concept was proposed. Unlike earlier models that treated each pedestrian independently, Social-LSTM incorporated the notion of social interactions by embedding the trajectories of surrounding agents into a shared representation. This allowed the model to account for mutual influence among pedestrians—for example, the tendency to avoid collisions or to follow group dynamics in shared spaces. An overview of their proposed method can be seen in Fig. 2.1. The introduction of this socially aware mechanism marked an important milestone in the field, as it shifted the focus from isolated trajectory modeling to interaction-aware prediction.

Building on this idea, later research sought to refine how models represent and prioritize these interactions. In particular, Anirudh et al. [15] extended the Social-LSTM framework by incorporating an attention mechanism that learns to weigh the relative influence of neighboring pedestrians. Instead of treating all surround-

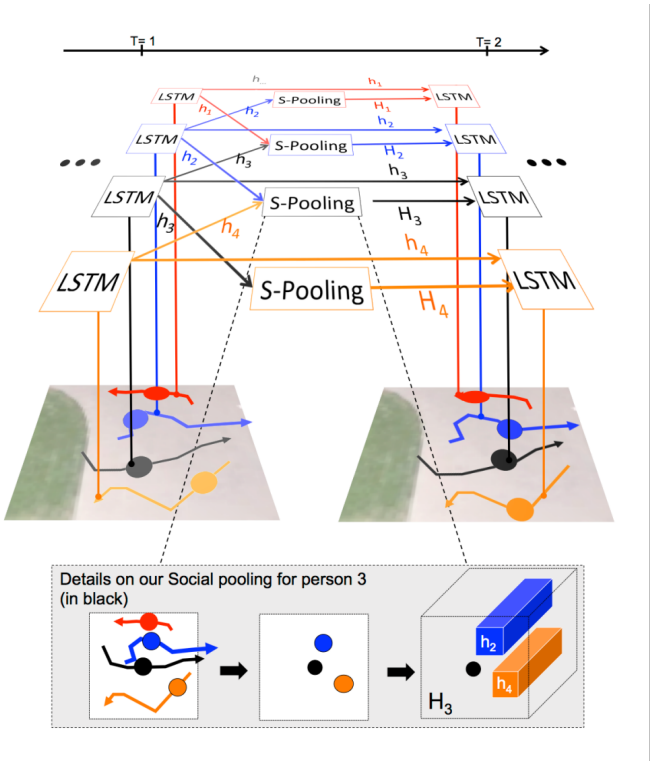

Figure 2.1: Social-LSTM proposed Architecture (©2016 IEEE)

ing agents equally, the model adaptively focused on the most relevant individuals, such as those directly in the pedestrian's path, thus improving prediction accuracy and interpretability. While this approach improved the realism of forecasts, it also introduced new challenges: the model must be able to reliably detect and track neighboring agents, as errors in this stage propagate directly into trajectory predictions.

Transformer-based architectures have been adopted more and more in recent approaches and studies, which have demonstrated success in natural language processing and computer vision studies due to their ability to model long-range dependencies using attention mechanisms. These models have an adapter for trajectory prediction to capture more complex interactions among agents in urban environments. For example, [5] proposes a camera-based transformer framework that is designed for predicting pedestrian intention of crossing the crosswalk. By leverag-

ing motion cues from the vehicles and progressively modeling interactions between pedestrians and their surroundings, the method effectively anticipates whether a pedestrian is going to cross the street or not. This represents a significant step forward in terms of contextual reasoning, as the Transformer's attention mechanism enables the model to selectively focus on the most relevant features across time and space. However, while intention prediction provides valuable insights for AV decision making, it is inherently limited in this scope since it does not provide the expected trajectory of a pedestrian but instead focuses on binary outcomes. This limitation restricts their applicability in downstream trajectory forecasting pipelines, where accurate multi-step trajectory predictions are necessary for motion planning and collision avoidance. Moreover, reliance on monocular vision alone leaves the system vulnerable to the same weaknesses discussed previously, such as occlusion sensitivity and poor performance in degraded environmental conditions.

Social-STGCNN [16] expands on socially aware trajectory modeling by introducing a spatio-temporal graph convolutional framework. In this design, each pedestrian is represented as a node and their interactions as temporal-spatial edges within a graph structure. This formulation allows the network to jointly capture both spatial relations among pedestrians at a given time and temporal dependencies across successive frames. By explicitly modeling interactions in this structured way, Social-STGCNN achieves strong trajectory forecasting performance while maintaining relatively low computational cost, making it suitable for large-scale datasets. Nevertheless, its reliance on well-defined relational structures introduces limitations: in sparse or unstructured scenarios, the model struggles to infer meaningful connections, and when trained on noisy or pseudo-labeled data, its performance degrades significantly due to the propagation of relational errors.

Wayformer [17] represents a different line of advancement by leveraging the Transformer architecture for motion prediction. Unlike graph-based approaches that

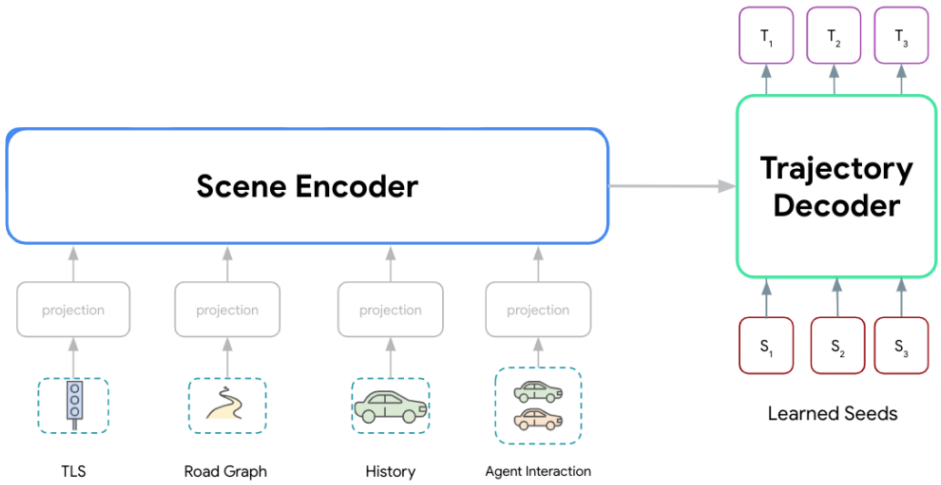

Figure 2.2: Wayformer Architecture (©2023 IEEE)

depend on predefined interaction structures, Wayformer employs an attention-based encoder–decoder framework that can flexibly learn dependencies across heterogeneous inputs. As shown in its architecture at Fig.2.2, the model is designed to unify diverse scene representations—such as agent trajectories, map elements, and multimodal sensor data—into a single forecasting pipeline. The authors explore multiple fusion strategies, including early, late, and hierarchical fusion, and demonstrate that early fusion achieves the most consistent improvements across benchmarks. Through extensive evaluation on large-scale datasets such as the Waymo Open Motion Dataset and Argoverse, Wayformer establishes state-of-the-art performance, highlighting the effectiveness of attention-driven architectures for handling multimodal information and complex interactions.

The trajectory prediction methods have evolved from interpretable but simplistic motion models to deep learning architectures capturing social interactions and contextual cues. For our research, we selected Social-STGCNN as a lightweight, socially aware baseline that efficiently models interactions between pedestrians, while being computationally affordable. In contrast, Wayformer was chosen as a state-of-the-art transformer-based framework that aligns with our framework, enabling us

to benchmark performance on a more modern architecture.

2.2 Detection and Sensor Fusion

Pedestrian detection has advanced considerably with the rise of deep learning, evolving from early image-based CNN detectors to sophisticated multimodal frameworks that incorporate LiDAR and radar information. One of the early deep learning–based approaches [18] introduced a network capable of jointly learning feature extraction, deformation handling, occlusion modeling, and classification for pedestrian detection. Although this work was novel for its time, it was constrained by the relatively shallow CNN architectures available, which limited its ability to generalize across complex urban environments.

Furthermore, the research sought to improve robustness under challenging conditions like occlusion. For instance, [19] extended the Faster R-CNN framework [20] with a guided attention mechanism that emphasized channel features associated with visible body parts. This allowed the model to better handle partial occlusions by selectively attending to regions most likely to correspond to pedestrians. While effective, this approach came with a high computational cost, which made it less practical for real-time deployment in autonomous vehicle systems.

Building on these ideas, ultimately, later studies explore more specialized usage of attention mechanisms to combine different sources of information. For example, [21] introduced an attention-based method that integrates visual features with human pose estimations. It uses the attention mechanism by focusing them on visible body parts while suppressing the background noise. The model reached an improved detection performance in highly cluttered or occluded scenarios. This work highlights the growing importance of the attention mechanism technique

All of these CNN-based approaches have a common emphasis on addressing issues such as occlusion by learning *part-aware* feature representations. Each research,

such as [18] answered the question of detecting pedestrians through implicit feature learning, in which the network was optimized for feature extraction, deformation handling, and classification. Moving on to more recent methods, [21] explicitly integrates an attention mechanism with human pose estimation, which allows the model to select which visible body section and part it is emphasizing while also suppressing the background clutter.

Altogether, these progresses represented a genuine progress towards improved visual robustness in pedestrian detection in the early years. Nevertheless, these methods are still discussed in the image domain, in which the performance can be influenced by factors such as occlusion, different situations, blurring, variations in lighting, noise, and, most importantly, adverse weather conditions. These categories of limitations will raise the need for auxiliary modules, such as point cloud-based modalities as LiDAR and radar. The combination of semantic information received from camera feed and the geometry precision of LiDAR, combined with the robustness of radar during all weather conditions, can be a robust answer to this problem. Hence, this results in multimodal fusion approaches that promise a path toward overcoming the weaknesses of relying solely on one modality.

MV3D [9] was one of the pioneers in sensor fusion. MV3D represents one of the earliest frameworks in this topic, combining LiDAR and RGB camera data to improve 3D object detection. To achieve this, they propose a method that projects LiDAR point clouds into multiple views, front view, bird's eye view, and image plane, and encodes them separately before applying deep feature fusion. Fig.2.3 demonstrates the architecture of this framework. This multimodal representation allows the network to exploit both the geometric accuracy of LiDAR and the semantic richness of RGB images, which produces 3D bounding boxes for object detection. This research was a significant improvement compared to the previous studies on 3D detections. It significantly outperformed prior methods in studies on the KITTI

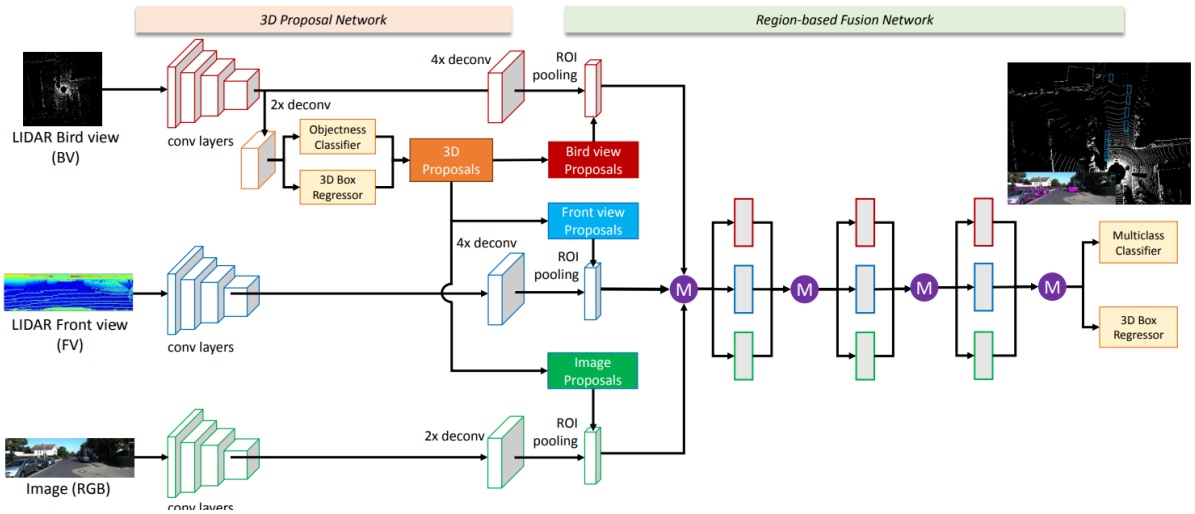

Figure 2.3: MV3D Architecture (©2025 IEEE)

benchmark, which set a new standard for multimodal detection at the time. However, its reliance on handcrafted multi-view projections caused a rigid design that is likely to lose fine-grained spatial information. Additionally, the approach is computationally demanding, which limits its applications in real-life scenarios and real-time deployments, posing an integral problem in autonomous systems.

In the same area, PointRCNN [22] was introduced. PointRCNN was the first framework to directly operate on the raw point clouds without resorting to voxelization or multi-view projection, as was the case for MV3D. PointRCNN introduces a two-stage, bottom-up pipeline. At the initial stage, a segmentation network generates high-quality 3D proposals that come directly from the raw point cloud input. Afterwards, the second stage takes these proposals, refines them in canonical coordinates using bin-based regression losses to produce accurate 3D bounding boxes. The PointRCNN method demonstrates that learning directly from raw point cloud data can achieve better accuracy when compared to projection-based approaches. Furthermore, it laid the foundation for many subsequent point cloud-based detectors. Nevertheless, its two-stage design introduces higher latency inference. Subsequently,

this leads to the same problem as before, which is a bottleneck in time-sensitive real-time applications, such as edge devices and autonomous vehicles. Later in this study, in the methodology section, we will evaluate PointRCNN along with other models on our multimodal dataset as part of the pseudo-labeling process; both to benchmark its performance in pedestrian detection and to assess its suitability for trajectory prediction pipelines. Since our dataset uses a denser point cloud structure as the raw input data, PointRCNN seems more suitable for our application.

Another one of the influential methods in our study and used in our framework is SECOND [23]. SECOND builds on earlier voxel-based detectors such as VoxelNet [24] and introduces the concept of **spatially sparse convolution**, which performs 3D convolution only on non-empty voxels that contain LiDAR points, avoiding computation over empty space and thus greatly reducing memory and runtime costs. This significantly reduces the computational cost by operating on these non-empty voxels while still maintaining high detection accuracy. This innovation allows SECOND to scale to larger scenes without a drop in memory efficiency and processing overhead of dense 3D convolution. Additionally, SECOND incorporates a sine-based angle loss, which stabilizes orientation regression, particularly for objects near angular boundaries, and employs a ground-truth sampling augmentation strategy that improves both convergence speed and model generalization. Such innovations and enhancements made SECOND one of the most practical models in the autonomous driving area of research.

Shifting away from voxel-based representations, PointPillars [25] introduces a pillar-based encoding in contrast to voxel-based representations. This approach maps the point clouds into vertical columns, or, in the paper terms, "pillars", and applies lightweight PointNet modules to learn pre-pillar features. These features are further organized into pseudo-image format, allowing the use of an efficient 2D CNN backbone for the detection stage. This approach eliminates the need for

computationally expensive 3D convolutions and make PointPillars highly time and computationally efficient. However, this comes with a downside. When reducing the detection to a pillar-based representation leads to a loss of fine-grained and small geometric detail, this gets worse when scenarios such as occlusion happen. Compared to voxel-based methods such as SECOND or more recent Transformer-based approaches, PointPillars tend to underperform in more complex urban scenarios where detailed spatial detection is crucial for safety.

Both of the last aforementioned studies, SECOND and PointPillars, represent an important milestone in the evolution of 3D object detection. This highlights the trade-off between computational efficiency and detection accuracy as expected. In the context of this thesis, these models were evaluated as part of the pseudo-labeling stage to generate pedestrian annotations from multimodal data that provides insight into their suitability for trajectory prediction tasks.

Building on earlier LiDAR and fusion-based detectors such as MV3D, He et al. [11] introduced a lightweight multimodal architecture that jointly addresses both 3D detection and trajectory prediction within a single framework. Unlike the prior approaches to the problem that treated detection and forecasting trajectory prediction as separate stages, their approach integrates these two stages. This enables real-time perception and prediction even on resource-limited platforms such as edge devices and mini-PCs. This research demonstrates a key direction in the evolution of fusion-based approaches. This direction is moving beyond large and resource-heavy pipelines towards a lighter and more compact model while maintaining the accuracy, in general models that are more practical and more suitable for tasks such as edge computing and embedded systems, and platforms.

Another important study in sensor fusion is FUTR3D [26]. This study shifted away from voxel-based model architectures and proposes a fully Transformer-driven design. FUTR3D introduces a unified, query-based sensor fusion framework that

treats inputs from multiple modalities within a shared Transformer backbone, thereby eliminating the need for handcrafted representations such as voxels or pillars. FUTR3D achieves strong performance when tested on the nuScene benchmark [27], which showcases the potential of attention-based fusion for robust multimodal sensor fusion models. However, while being effective for detection, it stays at predicting the intention of the pedestrian for crossing and does not provide feedback on the trajectory of the said pedestrian. This leaves the crucial question of how such unified fusion frameworks can be extended to downstream forecasting tasks—particularly in the context of multi-agent trajectory prediction for autonomous driving.

To summarize the literature review, trajectory prediction has come a long way from operating on simple Kalman-based approaches to transformer models using attention-based mechanisms. While being more accurate, they introduce the trade-off between complexity and accuracy. Meanwhile, while in the same scenarios, such as simpler problems or complex problems with known environmental variables, the Kalman filter and its variations, such as the extended Kalman filter, are still valuable tools.

In the topic of sensor fusion and detection, while early CNN and LiDAR-based strategies provided valuable intuition and innovation for pedestrian detection and improved the robustness under scenarios in which camera-only-based approaches fell short, each carried trade-offs in terms of accuracy, speed, and scalability. Voxel-based detectors, though effective in capturing 3D structure, often fail to preserve fine-grained details necessary for detecting smaller or partially occluded pedestrians. Pillar-based designs improve efficiency but further sacrifice detail.

These limitations have accelerated the shift toward multimodal fusion approaches, resulting in leveraging the complementary strengths of all the possible sensors, such as LiDAR, radar, and camera, and also toward Transformer-based architectures. These Transformer-based architectures provide flexible attention mechanisms capa-

ble of modeling different kinds of inputs. Together, these developments illustrate the trajectory of research moving from handcrafted and resource-heavy designs toward more general, unified frameworks that integrate detection and forecasting, ultimately paving the way for robust multimodal systems suitable for real-world deployment.

3 System Architecture

In this section, we will go through the sensor configuration and the storage strategy that were used to build the multimodal dataset for pedestrian trajectory prediction. The sensing setup combines three different modalities, 3D LiDAR, millimeter-wave radar, and an RGB camera, into one framework that serves as the backbone of the data processing pipeline. By putting these sources together, the system is able to represent dynamic urban scenes in a more complete way.

To make sure the data can be used for the later stages of the pipeline, all of the sensor streams are first time-synchronized and then recorded inside the ROS2 environment in a standardized bag file format. This way, the raw multimodal data can always be accessed, replayed, and processed consistently. The setup explained here forms the basis for everything that comes afterward, including calibration, data fusion, pedestrian detection, pseudo-labeling, and finally, trajectory prediction, which are explored in the methodology section.

3.1 Hardware Setup

Fig. 3.1 illustrates the physical placement and relative positioning of the three sensors used in the multimodal data collection system: an Ouster OS1 LiDAR [28], a Navtech RAS3 millimeter-wave radar [29], and an Intel RealSense D415 RGB camera [30]. The shown configuration in the picture ensures overlapping fields of view between all three sensors. This enables effective sensor fusion and consistent

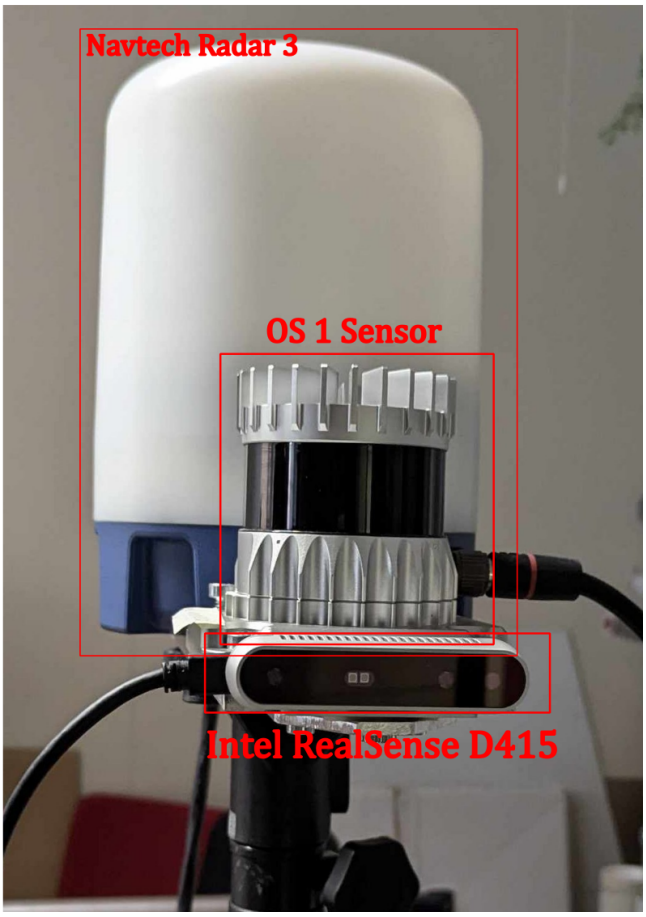

Figure 3.1: Our multimodal sensing setup: Ouster OS1 LiDAR, Navtech RAS3 radar, and Intel RealSense camera.

coverage of the scene, and also more accurate calibration in further stages. The detailed technical specifications of each sensor, including resolution, range, field of view, and operating frequency, are summarized in Table 3.1.

3.2 Data Collection

Data collection was carried out at the Aalto University Metro Center in Finland, a location chosen due to its proximity to the metro station and its usage as a busy urban intersection. Since this place experiences a continuous normal amount of diverse traffic, including pedestrians, vehicles, trams, and cyclists, it makes an ideal environment for capturing a diverse motion pattern and social interaction (as defined

Table 3.1: Sensor specifications

Sensor	Key Specifications	Rate
Ouster OS1 LiDAR	128 channels; 360° horizontal FoV; 42.4° vertical FoV; 90 m range; ±1.5 cm precision	10/20 Hz
Navtech RAS3 Radar	360° horizontal FoV; up to 270 m range; point cloud output	4/10 Hz
Intel RealSense D415	1280×720 px RGB-D; 69.4° horizontal FoV; 42.5° vertical FoV	30 Hz

Figure 3.2: Data Collection Point

in the related works section). An aerial image of the collection point is shown in Fig. 3.2

The sensors recorded the surroundings of the metro station for 47 minutes, producing a 196 GB dataset consisting of $15,492$ cumulative messages from radar, Li-

DAR, and camera streams. Camera recorded at 30 Hz, LiDAR recording at 20 Hz, and radar recording at 10 Hz.

The LiDAR and radar sensors both publish data in the `sensor_msgs/PointCloud2` message format, which in fact is the standardized representation of 3D point clouds within the ROS2 framework. In addition to point cloud data, the LiDAR also provides supplementary information topics of data, including inertial measurements (IMU), near-infrared intensity values, and other metadata, which are made available through their respective message types. The RGB camera outputs image frames using the `sensor_msgs/Image` message type, while its intrinsic calibration parameters, such as focal length, distortion coefficients, and principal point, are published under the `sensor_msgs/CameraInfo` message, under a separate topic.

To ensure spatial alignment between the heterogeneous modalities, the static extrinsic transformations that relate different sensor coordinate frames to each other are broadcast through the `tf_static` topic. These transforms encode the fixed geometric relationships between the LiDAR, radar, and camera frames, thereby enabling downstream modules to project and fuse multimodal data into a common reference frame. Together, these standardized ROS2 message types and transformation mechanisms form the foundation of the multimodal dataset, ensuring interoperability, reproducibility, and seamless integration across all stages of the proposed pipeline, which alter on will be suitable for downstream applications.

Each recording session generates a single SQLite3 `.db3` file, which encapsulates the multimodal dataset in a compact and standardized format. This SQLite3 file is comprised of not only the raw sensor measurements but also the associated metadata and precise timestamps, which ensures that the temporal relationships between modalities are preserved for downstream synchronization, calibration, and fusion. The use of a single database file per session simplifies dataset management, as it allows complete multimodal recordings to be accessed, replayed, or processed con-

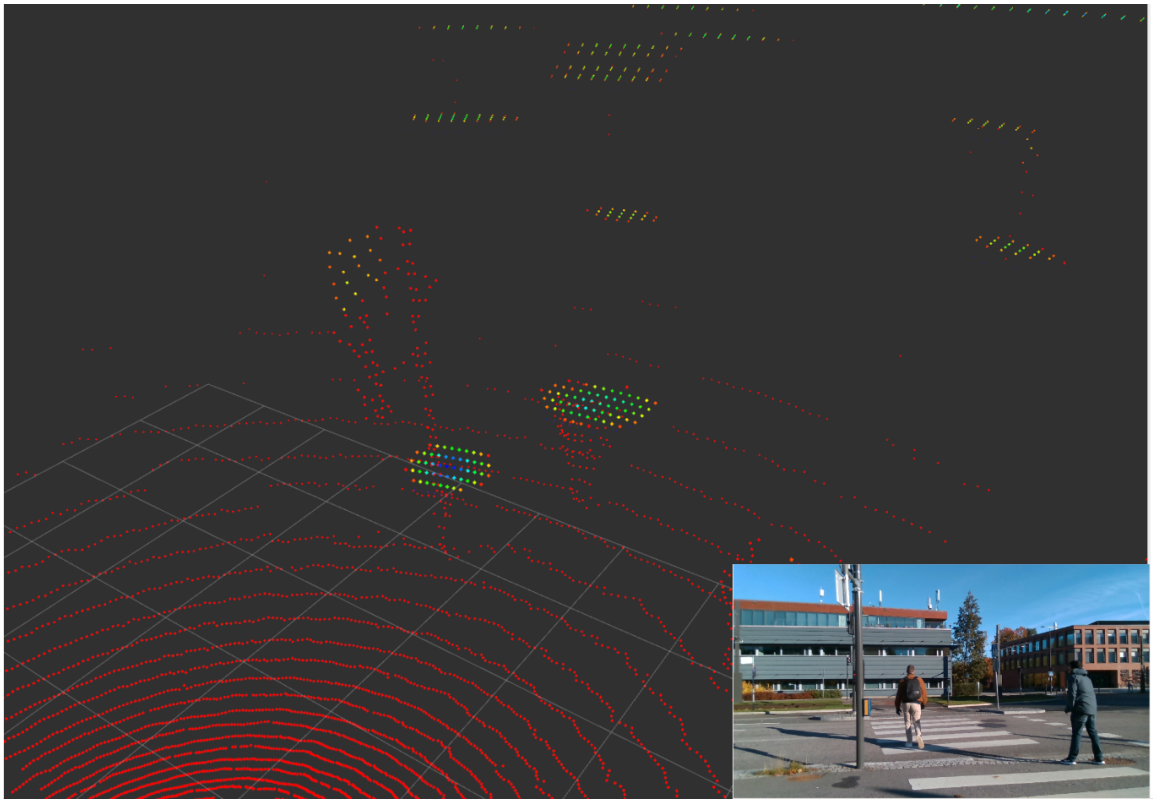

Figure 3.3: LiDAR point clouds (shown in red) and Radar point clouds (circular patches), the camera feed is also shown for the same timestamp at the bottom-right corner

sistently across different tools and environments.

An example of the raw multimodal data is visualized in the ROS visualization tool, also known as RVIZ, and is shown in Fig. 3.3. This figure illustrates the simultaneous outputs of the LiDAR, radar, and camera streams and highlights the complementary nature of the sensors. As depicted in the figure, while the LiDAR point cloud provides dense geometric information, the radar contributes long-range and weather-robust detections, and the RGB camera captures semantic data, appearance cues, and general geometrical shapes of the objects. Together, these visualizations underscore the richness of the collected data and motivate its suitability for multimodal pedestrian trajectory prediction.

4 Methodology

In this section, we will provide an overview of the three major stages that form the core of this research. In the first stage, we will discuss the preprocessing logics, which is in charge of handeling the raw model that were collected from LiDAR, radar, and camera sensor and temporally synchronizing them, spatially caliberating them into a common reference frame, and finally, organizing them, the point cloud datas in particular, into a structured dataset format suitable for downstream tasks such as of course, trajectory prediction. In this stage, we will further discuss the algorithm of each node, and ho into the details about how each of the sensors is responsible for handling input and output from the previous stages.

In the second stage, we will address the pseudo-labeling procedure and evaluate and find the best model suitable for our task at hand. This stage will take place after producing the synchronized and preprocessed data to obtain pedestrian annotations without relying on manual labeling, since the manual labeling for bigger datasets can be exhaustive. The aforementioned models will be tested on the said dataset. To enhance the robustness and add a tracker to each detection, the pseudo-labels are further refined through consistency checks across consecutive frames and by applying a nearest-neighbor-based tracker to assign stable pedestrian identities over time. Ultimately, this process results in labeled trajectories that can serve as ground truth for subsequent experiments, whether for trajectory prediction or other tasks.

In the final stage, we turn to the evaluation of the processed data on trajectory

prediction models. Here, the pseudo-labeled multimodal dataset is used for training and benchmarking different state-of-the-art forecasting architectures. We focus on different architectures, such as graph-based and transformer-based models, discussing their strengths and weaknesses when applied to multimodal input. In this stage, we also explain the design choices and experimental setup used for evaluation, while leaving the detailed presentation of both the quantitative and qualitative results for the next section.

4.1 Preprocessing

The preprocessing logic itself consists of three main steps. First, all sensor streams are synchronized chronologically to ensure temporal alignment. Next, the LiDAR and radar point clouds are transformed into the common reference frame using the extrinsics provided in `tf_static`, while the camera images are stored in the camera frame with the related calibration parameters. Ultimately, the calibrated and synced data are fused and saved in their respective formats for subsequent processing. The camera stream pictures are saved as `.png` while point cloud data are saved as `.pcd` files. The extra parameters related to camera instructions are saved as a `.yaml` file. Fig. 4.1 shows the relations between each node and the transferred messages between them.

To explain in more detail, the preprocessing logic in this work is structured around three key steps that prepare the multimodal data for downstream tasks. First, all sensor streams are synchronized chronologically to achieve temporal alignment across LiDAR, radar, and camera modalities. This ensures that measurements originating from different sensors correspond to the same real-world instant, which is critical for consistent data fusion and later trajectory analysis.

Second, using the extrinsics provided in `tf_static`, the spatial calibration is performed by transforming the LiDAR and radar data into a common reference

frame. This unifies the 3D spatial representation of the environment and enables cross-modal reasoning about object positions and movements. Camera images, in contrast, are stored within their native camera frame, but with the associated intrinsic and extrinsic calibration parameters preserved, allowing for future projection or cross-modal alignment when needed.

Finally, the fusion and storage stage consolidates the calibrated and synchronized streams. The point cloud data are fused in the common frame, while the corresponding camera images are saved simultaneously with matched timestamps. All processed data are then stored in their respective formats, forming a coherent multimodal dataset that can be reliably used for pseudo-labeling and ultimately, downstream applications such as trajectory prediction tasks. Fig. 4.1 illustrates the relations between the implemented ROS2 nodes and the flow of messages exchanged between them, highlighting how synchronization, calibration, and fusion are done in our proposed pipeline.

4.1.1 Temporal Synchronization

As illustrated in Algorithm 1, the synchronization framework consists of a chronological synchronizer as its first node, shown in Fig.4.1. The first node, referred to as the chronological synchronizer, subscribes to the three sensor streams: LiDAR (T_ℓ), radar (T_r), and camera (T_c). The synchronizer selects one of these streams as the anchor from which all timestamps are referenced; in our case, the LiDAR stream serves as the anchor. Afterwards, it initializes two buffers to store the messages of the other sensors, which are called B_r and B_c. It then subscribes to all the topics publishing the sensor messages, and creates new topics for the synchronized messages, called T_ℓ^{synced}, T_r^{synced}, and T_c^{synced}. On the arrival of the non-anchor sensor messages, the message and its timestamp are stored in their respective buffers. Upon the arrival of an anchor frame, the synchronizer searches the other buffers for

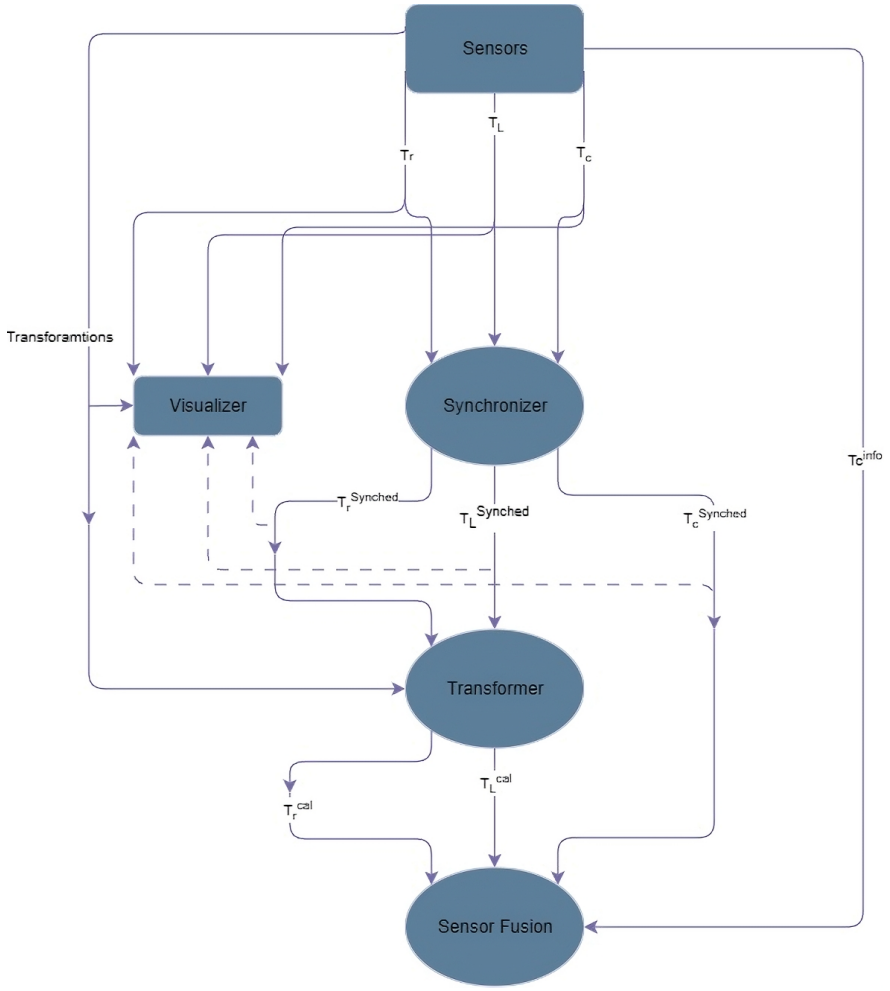

Figure 4.1: The node graph of our proposed framework, depicting the topics established between each node

the closest messages in time. Once the corresponding messages are located for each sensor and stored in $m_r^{nearest}$ for radar and $m_c^{nearest}$ for camera, the synchronized set is published together on the new topics. If only one suitable candidate is found from one of the buffers, the other message will be the last received message, but shown with a warning.

In terms of ROS2, the temporal calibrator was implemented as a ROS2 node acting as a subscriber for the raw sensor data. Ultimately, it acts as a publisher for the consecutive nodes.

Algoritm 1 Radar-LiDAR-Camera Chronological Synchronization

Require: Topics $\{T_\ell, T_r, T_c\}$;
Ensure: Topics $\{T_\ell^{synced}, T_r^{synced}, T_c^{synced}\}$
Subscribe to T_ℓ, T_r, T_c
Initialize buffer $B_r, B_c \leftarrow \emptyset$
Initialize topics T_ℓ^{synced}, T_r^{synced}, T_c^{synced}
while session active **do**
  On receive msg_r from T_r:
    $B_r \leftarrow msg_r$ and its timestamp
  On receive msg_c from T_c:
    $B_c \leftarrow msg_c$ and its timestamp
  On receive msg_ℓ from T_ℓ:
    $B_\ell \leftarrow msg_\ell$ and its timestamp
    $m_r^{nearest} \leftarrow$ nearest(B_r, t_ℓ)
    $m_c^{nearest} \leftarrow$ nearest(B_c, t_ℓ) **or** latest if camera time unreliable
  if m_r and m_c found **then**
      publish msg_ℓ in T_ℓ^{synced}
      publish $m_r^{nearest}$ in T_r^{synced}
      publish $m_c^{nearest}$ in T_c^{synced}
  end if
end while

4.1.2 Calibration

The calibrator is tasked with transforming the LiDAR and radar into a common coordinate frame. The transformations are all stored and published in `tf_static` topics. Algorithm 2 details the calibration framework applied in this research. First, the calibrator subscribes to the synchronized topics that were published by the synchronizer node. Afterwards, it initializes the listener L_tf to listen to the upcoming transforms and buffer B_{tf} and to store the incoming transforms.

During the session, all transforms between the frames are stored in tf_msg. When a new synchronized LiDAR message is received, and the corresponding transform $T_{\ell \rightarrow base\_link}$ is extracted from tf_msg and saved as a Numpy array. The same procedure is followed when a radar message is received. When both transformations are available, the LiDAR message msg_ℓ^{synced} is calibrated by applying the $T_{\ell \rightarrow base\_link}$ to it, and the same goes for radar, which results in msg_r^{calib} and msg_ℓ^{calib} Finally, the

Algoritm 2 Calibration Algorithm

Require: Topics $\{T_\ell^{synced}, T_r^{synced}, T_c^{synced}\}$
Ensure: Topics $\{T_\ell^{cal}, T_r^{cal}\}$;
Subscribe to $T_\ell^{synced}, T_r^{synced}$
Initialize a TF buffer $B_{tf} \leftarrow \emptyset$
$L_{tf} \leftarrow$ new TF listener linked to B_{tf}
while session active **do**
  $tf\_msg \leftarrow$ all transforms between *Lidar* & *base_link* using L_{tf}
  On receive T_ℓ^{synced}:
    $T_{lidar \to base\_link} \leftarrow \text{LookupTransform}(tf\_msg)$
    $T_{lidar \to base\_link} \leftarrow \text{TransformToNumpyArray}(T_{lidar \to base\_link})$
  On receive T_r^{synced}:
    $T_{radar \to base\_link} \leftarrow \text{LookupTransform}(tf\_msg)$
    $T_{radar \to base\_link} \leftarrow \text{TransformToNumpyArray}(T_{radar \to base\_link})$
  if Transforms exist **then**
    $msg_\ell^{calib} \leftarrow \text{ApplyT}(T_{lidar \to base\_link}, msg_\ell^{synced})$
    $msg_r^{calib} \leftarrow \text{ApplyT}(T_{r \to base\_link}, msg_r^{synced})$
    publish msg_ℓ^{calib} in T_ℓ^{cal}
    publish msg_r^{calib} in T_r^{cal}
  end if
end while

calibrated messages are published on their topics, T_r^{cal} and T_ℓ^{cal} respectively.

4.1.3 Sensor Fusion

As shown in algorithm 3, the sensor fusion node is responsible for combining the calibrated data and storing it. Upon initialization, it assigns frame indices for each modality and sets up caches for the latest radar and camera messages. As shown in Fig.4.1, the sensor fusion node subscribes to four topics: two topics created by the calibrator, T_r^{cal} and T_ℓ^{cal}. One topic that was created by the camera synchronizer was T_c^{synced}, and finally, an optional topic that was originally published for the camera's internal calibration info, named T_c^{info}. Camera calibration parameters (intrinsics and distortion coefficients) are extracted from T_c^{info} and saved as YAML files.

The latest messages received from T_r^{cal} and T_c^{synced} are kept in their respective variables, called msg_r^{latest} and msg_c^{latest}. Meanwhile, for each incoming LiDAR mes-

sage, the algorithm first extracts the message from the topic T_ℓ^{cal} and stores the message in msg_ℓ^{latest}. Afterwards, it retrieves the most recent radar and camera data, extracts the 3D points with intensity (x, y, z, I) from msg_ℓ^{latest} and msg_r^{latest}, and creates a unified point cloud message called P_{all} by vertically concatenating them. In the last step, the P_{all} and msg_c^{latest} are stored.

The camera stream is handled separately from the LiDAR and radar fusion process, as the two modalities differ fundamentally in their data representation. While both LiDAR and radar provide 3D point cloud measurements that can be directly merged into a unified spatial representation, the RGB camera produces 2D image data. Consequently, fusion is performed only on the 3D point clouds, resulting in a combined LiDAR–radar point cloud that is suitable for downstream tasks such as detection and trajectory prediction. The RGB images are not projected into the 3D domain at this stage; instead, they are saved in their original 2D format and stored alongside the point clouds with the same timestamps. This ensures that although the camera data remains modality-specific, it can still be used in later stages.

4.2 Pseudo-Labeling

Accurate annotation of the collected data is a critical prerequisite for the supervised training and evaluation of trajectory prediction models. To achieve this, the dataset must be created in a way that provides not only the spatial location of pedestrians but also creates unique IDs that allow each pedestrian to be continuously tracked across multiple frames uninterruptedly. Such temporal consistency is essential for creating reliable trajectories that can serve as input/output pairs for trajectory prediction models. However, as discussed in the previous sections, manual annotation of large-scale multimodal datasets is prohibitively time-consuming and resource-intensive, making the pseudo-labeling approach a viable option for this

Algoritm 3 Saving Data Algorithm

Require: Topics $\{T_\ell^{cal}, T_r^{cal}, T_c^{synced}\, T_c^{info}\}$;
Ensure: Dataset of `.pcd` (LiDAR/Radar or merged), `.png` images, and camera intrinsics (`.yaml`)

Initialize frame IDs: $i_\ell, i_r, i_m \leftarrow 0$
Initialize caches: $latest_r \leftarrow \varnothing$, $latest_c \leftarrow \varnothing$
Subscribe to $T_\ell^{cal}, T_r^{cal}, T_c^{sync}, T_c^{info}$
while session active **do**
  On receive T_c^{info}: Save $\{K, D, width, height, model\}$ as a file
  On receive T_r^{cal}: $latest_r \leftarrow$ message
  On receive T_c^{synced}: $latest_c \leftarrow$ message
  On receive T_ℓ^{cal}:
    $latest_\ell \leftarrow$ message
    $points_\ell \leftarrow$ Extracted (x, y, z, I) from $latest_\ell$
    $points_r \leftarrow$ Extracted (x, y, z, I) from $latest_r$
    $P_{all} \leftarrow$ VerticalStack$(points_\ell, points_r)$
    $pcd \leftarrow$ Construct point cloud(P_{all})
    Save pcd as `merged/frame_`i_m`.pcd`
    $i_m \leftarrow i_m + 1$
end while

study.

To overcome the challenge of creating reliable annotations without manual labeling, we employed a pseudo-labeling strategy using state-of-the-art 3D object detectors applied to our fused multimodal sensor data. These detectors automatically produced bounding box annotations for pedestrians at the frame level. However, frame-wise detections alone were insufficient for downstream trajectory-prediction tasks, where temporal consistency is required.

To bridge this gap, a lightweight nearest-neighbor tracking algorithm was integrated into the pseudo-labeling pipeline. Rather than performing complex motion modeling, the tracker simply associates each newly detected pedestrian, obtained from the PointRCNN output, with the closest existing track from the previous frame. This preserves identifier consistency across time, enabling the construction of continuous pedestrian trajectories. Despite its simplicity, this approach proved effective in producing coherent, high-quality trajectory annotations suitable for subsequent

forecasting tasks.

The tested models were implemented using the OpenPCDet framework [31], which is a toolbox for 3D detection. The code base of the OpenPCDet has been changed to be used in our framework. Most importantly, the test scripts of OpenPCDEt, which experiment on different models, have been altered to fit in the ROS2 framework

This stage was implemented as a ROS2 service, in which the model and all of its initializations have been done before the service is called. The service first loads the checkpoints from the pre-trained model. After loading the pre-trained weights, it awaits a call to the service. After receiving a call, it loads the dataset that was created in the previous stage after the sensor fusion step. Subsequently, it applies the pre-trained model that was loaded beforehand onto the dataset. The results are processed as discussed in the previous paragraph, by adding the tracking algorithm to the resulting outputs. Ultimately, after the processing of the whole dataset is finished, it is saved as a JSON file. The JSON file structure is discussed in the next paragraph.

To make sure that the output data produced by the pipeline can be easily used with common benchmarks and integrated directly into existing prediction frameworks, the pseudo-labeled results were reformatted into a standardized dataset structure. In this structure, every detected pedestrian within a frame is represented by a 3D bounding box that includes its spatial position (x, y, z), physical dimensions (dx, dy, dz), orientation yaw, detection confidence $score$, and semantic class label $label$. In addition, each detection is assigned a unique identifier ID to allow consistent tracking of the same pedestrian across consecutive frames, as discussed in the previous section. This format was intentionally chosen to resemble the structure of popular autonomous driving datasets such as KITTI [32] and NuScenes [27], so that the data produced by this pipeline can directly interface with existing trajec-

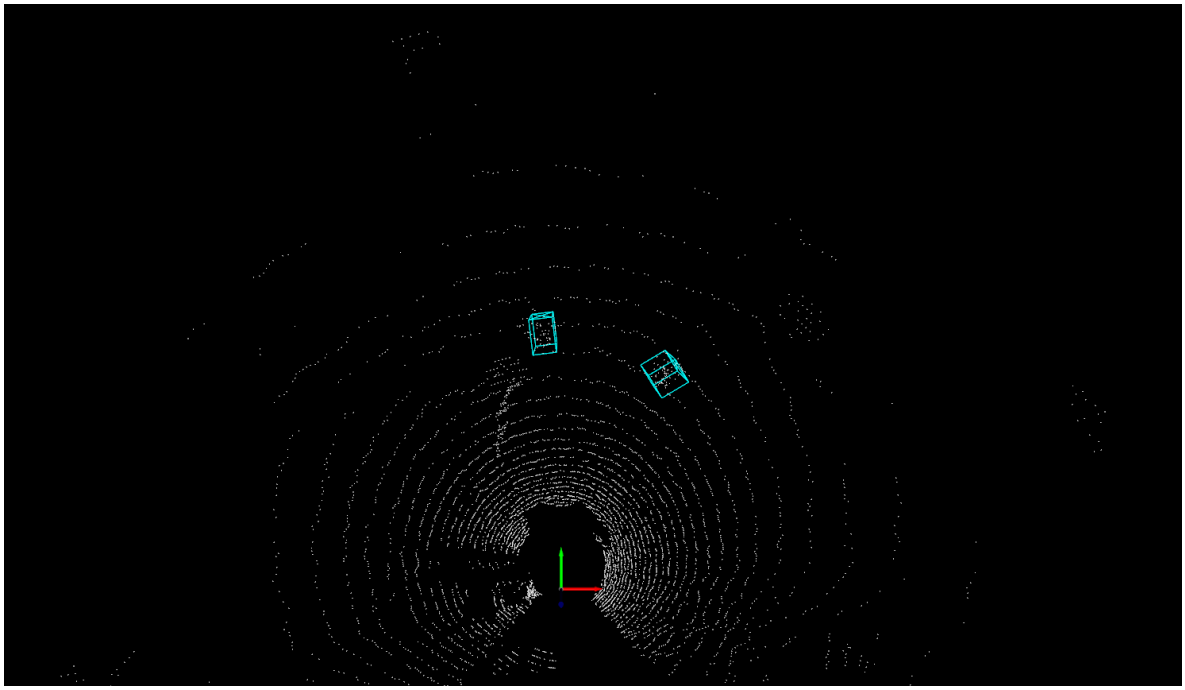

Figure 4.2: The output of the pseudo-labeler, using PointRCNN as the chosen model. The detected pedestrians are shown in blue bounding boxes.

tory prediction frameworks like Wayformer and Social-STGCNN without the need for major conversions or preprocessing. As a result, the generated dataset not only aligns with well-known research standards but also provides a flexible bridge between the pseudo-labeling module and downstream forecasting models. An example of the pseudo-labeler's output can be seen in Fig. 4.2.

The resulting dataset thus provides structured, temporally consistent annotations that serve as ground truth labels for the training and evaluation of trajectory prediction algorithms. To reduce the need for manual labeling, our framework employs a pseudo-labeling strategy that uses state-of-the-art 3D object detectors on fused multimodal sensor data. These detectors automatically generate pedestrian bounding boxes at the frame level. However, trajectory prediction models require temporal coherence, which raw frame-wise detections cannot provide.

To address this, we integrated the aforementioned lightweight nearest-neighbor

tracking algorithm into the pseudo-labeling pipeline. As mentioned, in each new frame, detected pedestrians from PointRCNN are matched with the closest track from the previous frame. This simple association preserves consistent pedestrian identities and allows continuous trajectory construction. Although minimalistic, the method proved sufficiently effective for producing trajectory annotations suitable for downstream forecasting.

4.2.1 Detection Model Comparison

One challenge in fusing radar and LiDAR data arises from the structural differences in their 3D outputs. As shown in Fig. 3.3, LiDAR produces dense and geometrically accurate point clouds, whereas radar outputs are sparse and exhibit different noise patterns and spatial distributions, often appearing as circular patches in our device. These discrepancies introduce difficulties for detection algorithms that were originally designed to operate on homogeneous point cloud data. This problem requires us to measure different models and evaluate them on our own customized dataset to see the best fit for the downstream applications.

To address this problem, we conducted experiments with multiple state-of-the-art 3D object detection algorithms to assess their performance on the fused radar–LiDAR dataset. While benchmark studies indicate that Part-A^2 [33] achieves the highest accuracy in pedestrian detection across widely used datasets, our empirical evaluation revealed that PointRCNN [22] outperformed other candidates on our customized multimodal dataset. Specifically, PointRCNN demonstrated superior robustness in handling the structural variations present in radar and LiDAR augmented point clouds, resulting in more reliable pedestrian detection in practice. Other candidates, being Part-A^2, SECOND, and Centerpoint, were also measured by the F1-score criterion, the same as PointRCNN. But the resulting score of these models was near zero. It is concluded that these models were designed for single modular datasets,

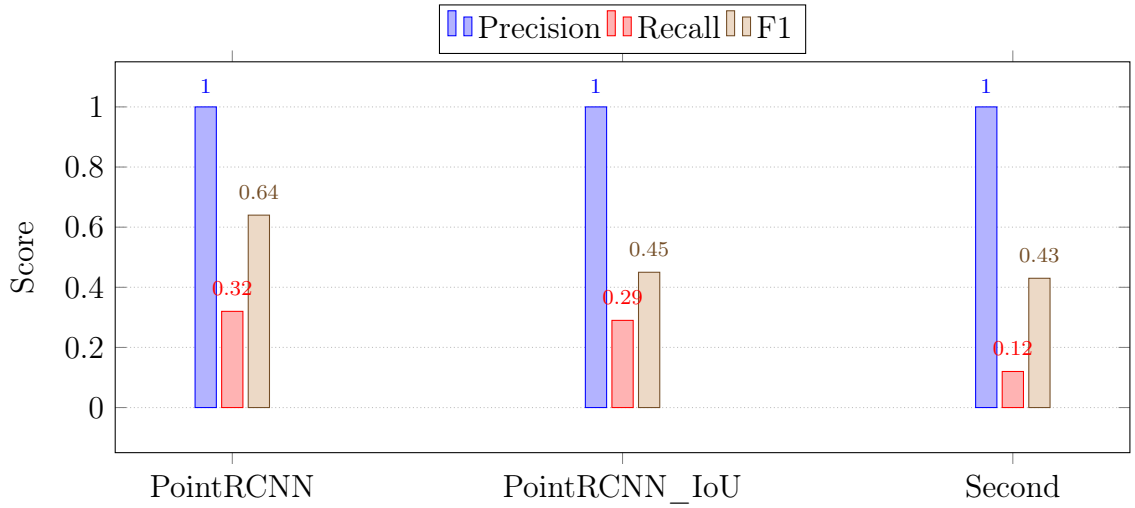

Figure 4.3: Detection performance comparison across models. Bars show Precision, Recall, and F1-scores on the same evaluation subset.

such as LiDAR-only point cloud datasets. However, the results of the PointRCNN can be seen in Fig. 4.2

Furthermore, the results of the models that had significantly above-zero F1-score are summarized in Fig. 4.3, which presents the precision, recall, and F1-score achieved by each evaluated model. These metrics highlight the trade-offs between theoretical benchmark performance and real-world applicability, emphasizing the importance of dataset-specific evaluation when selecting detection models for multimodal pedestrian trajectory prediction.

Alternative approaches, such as PointPillars [25], were also evaluated on the fused customized dataset. However, these models gave significantly lower precision and recall scores, which in some cases were near zero. This highlights their limited suitability for our customized dataset and pedestrian detection requirements of this study. This outcome can be attributed to two factors. First, PointPillars and similar architectures were originally designed and tuned primarily for vehicle detection, where objects are larger, more uniform in shape, and easier to represent within pillar-based encodings. Pedestrians, in contrast, present a higher difficulty challenge due

to their smaller size, highly variable poses, and irregular spatial structure. Second, the datasets used to train PointPillars and related methods typically feature sparser point cloud density compared to our multimodal dataset, and even compared to raw OS1 LiDAR point cloud without any preprocessing, which further reduces their ability to generalize effectively to the pedestrian-focused scenarios considered in this work.

Based on these observations, PointRCNN was ultimately selected as the most suitable detector for generating pseudo-labels in our pipeline, as it provided the most reliable balance of accuracy and robustness under multimodal fusion conditions. With the pseudo-labeled and temporally tracked pedestrian trajectories obtained, we now possess the necessary dataset to move to the next stage of the study. The following section presents the evaluation of trajectory prediction models trained on this dataset, providing insights into their performance and the effectiveness of the proposed framework.

5 Results

After generating the multimodal dataset and obtaining the pseudo-labeled annotations, the next step in our research was to evaluate how effective the proposed framework actually is for the downstream task, the pinnacle of which is the pedestrian trajectory prediction. In this stage, the synchronized and calibrated dataset, which now contains pseudo-labeled pedestrian detections from LiDAR, radar, and camera inputs, was used to train and test several trajectory prediction models. The goal here was not only to measure numerical performance but also to understand how well the pseudo-labeled multimodal data could support complex prediction tasks compared to traditional datasets. Among the evaluated models was Social-STGCNN, a spatio-temporal graph convolutional network that models the interactions between pedestrians through a graph-based structure, which was also discussed in the literature review section. This model has been widely used and benchmarked in trajectory forecasting literature, making it a suitable candidate for verifying the quality and usability of the dataset generated by the proposed pipeline.

This section is structured to give a clear overview of how the evaluation process was carried out from start to finish. We first introduce the custom visualization tool that was specifically developed to check the temporal and spatial alignment between sensors. This tool not only served as a useful diagnostic instrument during development to check the correctness of transforms stored in the raw dataset, but also helped in visually confirming that the labeling and synchronization processes

were functioning correctly. After that, we move on to describing the training setup used for the trajectory prediction models, along with the preprocessing steps and key hyperparameter configurations that were applied. Following this, we outline the evaluation metrics used to measure the models' forecasting accuracy—mainly the Average Displacement Error (ADE) and Final Displacement Error (FDE), which are widely recognized standards for assessing trajectory prediction performance. Lastly, we present a detailed comparison of how each model performed on the generated multimodal dataset, discussing their relative strengths, weaknesses, and overall suitability for predicting pedestrian motion in complex real-world scenarios.

5.1 Qualitative Verification of Data Alignment

As mentioned in the previous paragraph, to ensure the integrity of both the recorded sensor data and the transformations between sensor frames, two complementary validation tools were employed. The first tool was the ROS visualization environment, RVIZ, which enables frame-by-frame inspection of raw multimodal outputs. As illustrated in Fig. 3.3, RVIZ provides a direct means of visualizing LiDAR, radar, and camera data simultaneously, allowing qualitative assessment of sensor coverage and data consistency during and after recording sessions.

In addition to this, a custom visualizer node was implemented to further validate temporal alignment and extrinsic transformations. As shown in Fig.4.1, this node can be configured to receive input either directly from the raw sensor streams or from the chronological synchronizer node. When initialized, it receives the point cloud data of radar and LiDAR from the chosen source, and then transforms the point cloud data into the camera frame, as shown in Fig. 5.1. As mentioned, this process provides a direct verification of the correctness of extrinsic calibration by ensuring that LiDAR and radar points align accurately with the camera image plane. It also serves to confirm that the synchronizer node correctly aligns sensor messages over

time.

Aside from the technical validation, the visualizer also provides a very practical and easy-to-understand interface for human operators. It achieves this by visualizing the sensor data streams together in real time. It becomes much easier to notice problems and defects that might otherwise go unseen, such as small synchronization mismatches, calibration offsets, incompleteness of the data from one of the sensors, or transformation errors between frames. This visual layer gives an immediate sense of how well the sensors are aligned and how accurately the transformations are being applied. In this way, the visualizer goes beyond what standard ROS tools like RVIZ offer. It not only helps confirm that the transformations between coordinate frames are valid but also acts as a direct and user-friendly debugging tool for diagnosing and resolving issues within the multimodal data processing pipeline. Allowing operators to visually inspect and cross-check sensor alignment during data playback adds a layer of reliability and confidence to the entire pipeline.

5.2 Trajectory Models Evaluation

The dataset generated and refined during the pseudo-labeling phase is now used as the main input for several trajectory prediction models. Before training, a few minor adjustments were made to match the data format and input requirements of each specific model. These modifications mainly involved reorganizing the pedestrian trajectories, normalizing positional coordinates, and ensuring that each sequence follows the temporal and spatial conventions expected by the architecture of the desired model, e.g, for Social-STGCNN, the data needed to be in the format of ETH/UCY dataset. Once these preparations were complete, the dataset could be integrated into different model pipelines, allowing for a fair and consistent evaluation across multiple state-of-the-art forecasting frameworks.

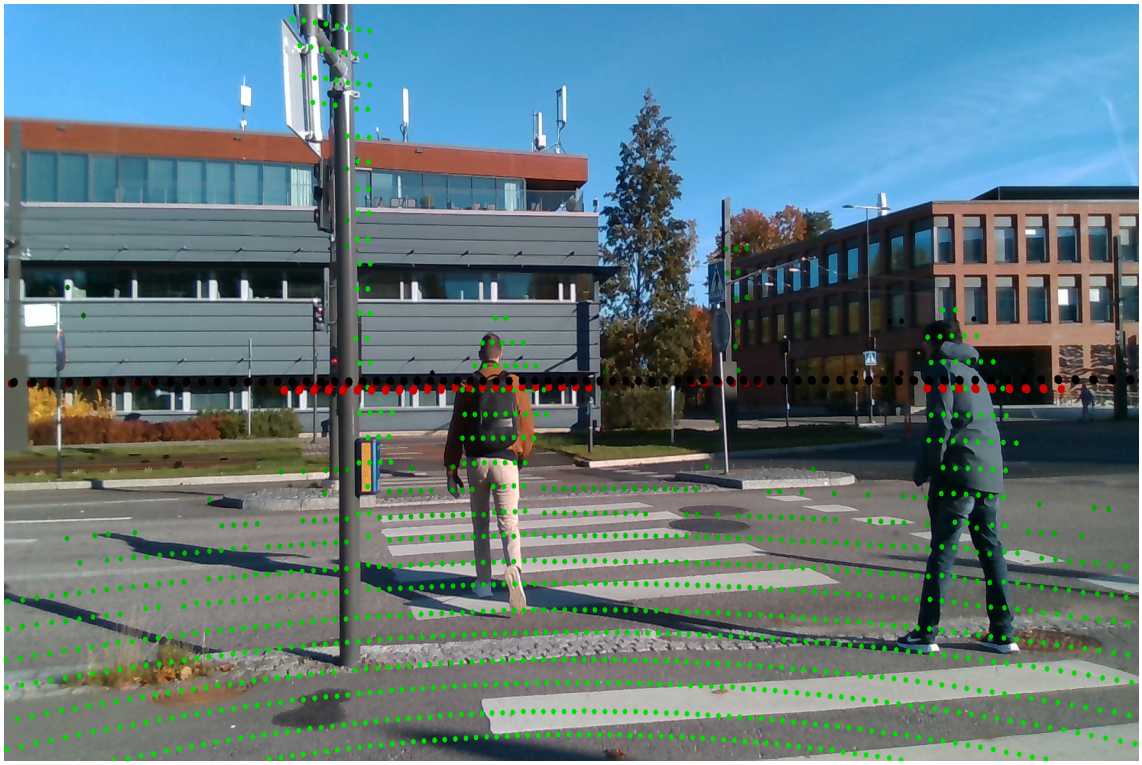

Figure 5.1: Projecting point cloud data on camera frame using TF tree

5.2.1 Evaluation Metrics

To evaluate the performance of each model, we adopted evaluation metrics that are widely recognized in state-of-the-art trajectory prediction benchmarks. These metrics assess not only the accuracy of predicted pedestrian positions but also the reliability of long-horizon forecasting, both of which are critical for safety-critical applications such as autonomous driving.

To begin, we need a definition for the ground truth. Other research, such as [34], states that forecasting accuracy is measured as the $L2$ distance between the prediction and the ground-truth trajectory, where ground truth is the precisely localized future path in metric space, which furthermore introduces ADE and FDE based on this metric space.

- **Average Displacement Error (ADE)**: Defined as the mean Euclidean dis-

tance between the predicted trajectory points and their corresponding true positions, and computed across all prediction timesteps. ADE captures the overall trajectory constancy, which provides insight into how closely the predicted path aligns with the ground truth throughout the forecast horizon. Models with lower ADE are generally better at maintaining consistent accuracy across multiple steps.

- **Final Displacement Error (FDE)**: Defined as the Euclidean distance between the predicted final position and the true final position at the last timestep of the prediction horizon. FDE emphasizes long-term forecasting accuracy, which is particularly important for downstream planning and collision avoidance in autonomous vehicle systems.

Together, ADE and FDE provide a complementary evaluation framework: ADE reflects the model's short-term and summed prediction quality, while FDE highlights its ability to accurately anticipate final pedestrian positions over longer horizons. By considering both, we obtain a comprehensive view of model performance across different temporal scales.

5.2.2 Performance Comparison

Table 5.1 presents a comparison between the pretrained Social-STGCNN and Wayformer, which were tested on the resulting dataset of the pseudo-labeling section.

Social-STGCNN achieves substantially lower errors, reaching an ADE of 0.51 and an FDE of 0.76, which indicates that it produces both more accurate short-term displacements and more reliable long-term forecasts. In contrast, Wayformer reports considerably higher errors-particularly in ADE, highlighting its difficulty in adapting to the fragmented and pseudo-labeled nature of the dataset. These results suggest that transformer-based architectures, which typically rely on well-

structured and continuous trajectory data, may experience performance degradation when faced with temporally imperfect labels or irregular sequence lengths.

The strong performance of Social-STGCNN underscores an important characteristic of graph-based models: their reliance on local spatial interactions and relative motion cues, rather than strict temporal consistency. As a result, they appear more robust to pseudo-labeling artifacts, where annotations can be sparse, discontinuous, or partially missing across frames. Additionally, the difference in spatial representation may play an influential role. Wayformer utilizes a bird's-eye-view (BEV) encoding that assumes a stable and dense occupancy structure, whereas Social-STGCNN directly operates on scene-centric 2D coordinates derived from trajectories. This suggests that models that do not depend on BEV grids or occupancy consistency may be better suited for datasets derived from multimodal pseudo-labeling pipelines, where trajectory continuity is not guaranteed.

Overall, the results highlight that Social-STGCNN offers a more resilient modeling approach under challenging annotation conditions, making it a strong candidate for forecasting pedestrian motion from multimodal sensor data with imperfect or pseudo-generated labels.

Moreover, it is noteworthy to mention that Social-STGCNN was evaluated in a zero-shot manner. The model was not trained on our dataset and was instead used directly with its publicly available ETH/UCY pretrained weights. This was due to its strict requirement for uninterrupted and fully labeled trajectory sequences, which conflict with the pseudo-labeled characteristics of our dataset, where pedestrian tracks may be fragmented or partially missing. As a result, Social-STGCNN serves only as a qualitative reference baseline, whereas the primary quantitative evaluation focuses on Wayformer, which was trained from scratch on the custom data and does not impose the same continuity constraints.

The results demonstrate that our proposed pipeline achieves its required func-

Table 5.1: Comparison of trajectory prediction models on the custom dataset.

Model	**ADE**	**FDE**
STGCNN	0.51	0.76
Wayformer	2.95	2.31

tionality and proposed initial plan, which successfully bridges the gap between raw sensor streams from different sensory inputs and downstream trajectory forecasting. Beginning with separated inputs from LiDAR, radar, and camera, the system performs synchronization, calibration, and fusion, followed by automated pseudo-labeling using the state-of-the-art models and trajectory construction. These outputs are then formatted into a standardized dataset structure suitable for training modern trajectory prediction models. The fact that Wayformer was able to train from scratch on this dataset and produce consistent ADE and FDE scores provides strong evidence of the robustness of the data preparation process.

6 Conclusion

In this work, we presented a ROS2–based pipeline that reliably converts raw and unstructured multimodal sensor streams into a structured, calibrated, and pseudo-labeled dataset that is ready for downstream applications such as trajectory prediction, thereby demonstrating the end-to-end feasibility of multimodal pedestrian trajectory forecasting. This proposed pipeline consists of three key stages.

The preprocessing stage performs chronological synchronization across several sensors, ensuring that camera, LiDAR, and radar measurements are temporally aligned. It then applies spatial calibration to transform all sensor outputs into a common reference frame, followed by structurally storing the data in standardized formats suitable for downstream processing applications. This stage establishes the foundation for consistent and reproducible multimodal datasets.

The labeling stage employs a pseudo-labeling strategy built on state-of-the-art 3D object detection models such as PointRCNN. These detectors generate pedestrian bounding boxes from the fused data streams, which serve as pseudo-ground-truth annotations. To extend these detections into temporally coherent trajectories, a lightweight nearest-neighbor tracking mechanism was integrated, assigning consistent IDs to pedestrians across frames. This process provides structured annotations that enable the supervised training of trajectory forecasting models.

Finally, the trajectory prediction stage evaluates a range of forecasting models on the generated dataset. By testing both graph-based and Transformer-based ar-

chitectures, the evaluation not only evaluates prediction accuracy through metrics such as ADE and FDE but also highlights the ability of each model to cope with pseudo-labeled, multimodal inputs. These results provide insights into the strengths and limitations of existing forecasting approaches and establish a baseline for future improvements.

Overall, this research provides a streamlined and reproducible pipeline for multimodal sensor fusion, pseudo-labeling, and trajectory forecasting, addressing a key gap in the availability of structured multimodal pedestrian datasets. Looking forward, several directions can further enhance this work. One is to expand the dataset with more diverse and robust annotations, which will further improve generalization. Second is to incorporate additional detection and tracking models that will strengthen the pseudo-labeling stage. Lastly, integrating a more general and bigger set of trajectory prediction model architectures will provide a more comprehensive evaluation. All together, these extensions will advance our framework and will provide the development of a more generalized and scalable piepline, which ultimately advances the safety and reliability of autonomous driving systems operating in the real world.

References

[1] D. Iberraken and L. Adouane, *Safety of autonomous vehicles: A survey on model-based vs. ai-based approaches*, 2023. arXiv: 2305.17941 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2305.17941.

[2] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Understanding pedestrian behavior in complex traffic scenes", *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 1, pp. 61–70, 2018. DOI: 10.1109/TIV.2017.2788193.

[3] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior", in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 206–213. DOI: 10.1109/ICCVW.2017.33.

[4] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 961–971. DOI: 10.1109/CVPR.2016.110.

[5] L. Shi, L. Wang, S. Zhou, and G. Hua, "Trajectory unified transformer for pedestrian trajectory prediction", in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 9641–9650. DOI: 10.1109/ICCV51070.2023.00887.

[6] L. Neumann and A. Vedaldi, "Pedestrian and ego-vehicle trajectory prediction from monocular camera", in *2021 IEEE/CVF Conference on Computer Vision*

and Pattern Recognition (CVPR), 2021, pp. 10 199–10 207. DOI: 10.1109/CVPR46437.2021.01007.

[7] K. Saleh, M. Hossny, and S. Nahavandi, "Real-time intent prediction of pedestrians for autonomous ground vehicles via spatio-temporal densenet", in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 9704–9710. DOI: 10.1109/ICRA.2019.8793991.

[8] J. Zhong, H. Sun, W. Cao, and Z. He, "Pedestrian motion trajectory prediction with stereo-based 3d deep pose estimation and trajectory learning", *IEEE Access*, vol. 8, pp. 23 480–23 486, 2020. DOI: 10.1109/ACCESS.2020.2969994.

[9] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving", in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6526–6534. DOI: 10.1109/CVPR.2017.691.

[10] M. Bijelic, T. Gruber, F. Mannan, *et al.*, "Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather", in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 679–11 689. DOI: 10.1109/CVPR42600.2020.01170.

[11] Y. He, L. Zhao, T. Deng, Z. Fang, and W. Chen, *Lightweight lidar-camera 3d dynamic object detection and multi-class trajectory prediction*, 2025. arXiv: 2504.13647 [cs.RO]. [Online]. Available: https://arxiv.org/abs/2504.13647.

[12] F. Drews, D. Feng, F. Faion, L. Rosenbaum, M. Ulrich, and C. Gläser, *Deepfusion: A robust and modular 3d object detector for lidars, cameras and radars*, 2022. arXiv: 2209.12729 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2209.12729.

[13] C.-Y. Lin, L.-J. Kau, and C.-Y. Chan, "Bimodal extended kalman filter-based pedestrian trajectory prediction", *Sensors*, vol. 22, no. 21, 2022, ISSN: 1424-8220. [Online]. Available: https://www.mdpi.com/1424-8220/22/21/8231.

[14] D. C. Duives, G. Wang, and J. Kim, "Forecasting pedestrian movements using recurrent neural networks: An application of crowd monitoring data", *Sensors*, vol. 19, no. 2, 2019, ISSN: 1424-8220. [Online]. Available: https://www.mdpi.com/1424-8220/19/2/382.

[15] A. Vemula, K. Mülling, and J. Oh, "Social attention: Modeling attention in human crowds", *CoRR*, vol. abs/1710.04689, 2017. arXiv: 1710.04689. [Online]. Available: http://arxiv.org/abs/1710.04689.

[16] A. A. Mohamed, K. Qian, M. Elhoseiny, and C. G. Claudel, "Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction", *CoRR*, vol. abs/2002.11927, 2020. arXiv: 2002.11927. [Online]. Available: https://arxiv.org/abs/2002.11927.

[17] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, *Wayformer: Motion forecasting via simple & efficient attention networks*, 2022. arXiv: 2207.05844 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2207.05844.

[18] W. Ouyang and X. Wang, "Joint deep learning for pedestrian detection", in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 2056–2063. DOI: 10.1109/ICCV.2013.257.

[19] S. Zhang, J. Yang, and B. Schiele, "Occluded pedestrian detection through guided attention in cnns", in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6995–7003. DOI: 10.1109/CVPR.2018.00731.

[20] R. Gavrilescu, C. Zet, C. Foșalău, M. Skoczylas, and D. Cotovanu, "Faster r-cnn:an approach to real-time object detection", in *2018 International Conference and Exposition on Electrical And Power Engineering (EPE)*, 2018, pp. 0165–0168. DOI: 10.1109/ICEPE.2018.8559776.

[21] Z. Jiang, S. Huang, and M. Li, "A pedestrian detection network based on an attention mechanism and pose information", *Applied Sciences*, vol. 14, no. 18, 2024, ISSN: 2076-3417. DOI: 10.3390/app14188214. [Online]. Available: https://www.mdpi.com/2076-3417/14/18/8214.

[22] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud", *CoRR*, vol. abs/1812.04244, 2018. arXiv: 1812.04244. [Online]. Available: http://arxiv.org/abs/1812.04244.

[23] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection", *Sensors*, vol. 18, no. 10, 2018, ISSN: 1424-8220. [Online]. Available: https://www.mdpi.com/1424-8220/18/10/3337.

[24] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection", *CoRR*, vol. abs/1711.06396, 2017. arXiv: 1711.06396. [Online]. Available: http://arxiv.org/abs/1711.06396.

[25] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds", *CoRR*, vol. abs/1812.05784, 2018. arXiv: 1812.05784. [Online]. Available: http://arxiv.org/abs/1812.05784.

[26] X. Chen, T. Zhang, Y. Wang, Y. Wang, and H. Zhao, "Futr3d: A unified sensor fusion framework for 3d detection", *arXiv preprint arXiv:2203.10642*, 2022.

[27] H. Caesar, V. Bankiti, A. H. Lang, *et al.*, "Nuscenes: A multimodal dataset for autonomous driving", *CoRR*, vol. abs/1903.11027, 2019. arXiv: 1903.11027. [Online]. Available: http://arxiv.org/abs/1903.11027.

[28] I. Ouster, "OS1 Mid-Range High-Resolution Imaging LiDAR Datasheet (Firmware Rev 7 / Hardware Rev 7.0 / Firmware v3.1)", Ouster, Inc., Tech. Rep. Rev 7, Feb. 2025, Firmware v3.1, Hardware Revision 7.0. [Online]. Available: https://data.ouster.io/downloads/datasheets/datasheet-rev7-v3p1-os1.pdf.

[29] N. R. Limited, "RAS3 Series Datasheet", Navtech Radar Limited, Tech. Rep., Jun. 2024, RAS3 Series Datasheet (updated 11 June 2024). [Online]. Available: https://navtechradar.com/wp-content/uploads/2024/06/RAS3-Series-datasheet-updated-11.06.24.pdf.

[30] I. Corporation, "Intel ® RealSense™ D400 Series Datasheet", Intel Corporation, Tech. Rep. 337029-009 (rev. 009), Jun. 2020, Intel RealSense D400 Series Product Family Datasheet. [Online]. Available: https://cdrdv2-public.intel.com/841984/Intel-RealSense-D400-Series-Datasheet.pdf.

[31] O. D. Team, *Openpcdet: An open-source toolbox for 3d object detection from point clouds*, https://github.com/open-mmlab/OpenPCDet, 2020.

[32] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset", *International Journal of Robotics Research (IJRR)*, 2013.

[33] S. Shi, Z. Wang, X. Wang, and H. Li, "Part-a^2 net: 3d part-aware and aggregation neural network for object detection from point cloud", *CoRR*, vol. abs/1907.03670, 2019. arXiv: 1907.03670. [Online]. Available: http://arxiv.org/abs/1907.03670.

[34] P. Dendorfer, V. Yugay, A. Ošep, and L. Leal-Taixé, *Quo vadis: Is trajectory forecasting the key towards long-term multi-object tracking?*, 2022. arXiv: 2210.07681 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2210.07681.