Enhancing Object Detection with Transformer-Based Adaptive Sensor Fusion

by

Reza Sadeghian

Thesis submitted to the University of Ottawa in partial fulfillment of the requirements for the degree of Doctorate in Philosophy in Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Examining Committee

The following served on the Examining Committee for this thesis.

External Member: James Clark
Professor, Department of Electrical and Computer Engineering
McGill University

Internal Member(s): Burak Kantarci
Professor, School of Electrical Engineering & Computer Science
University of Ottawa

Paula Branco
Associate Professor, School of Electrical Engineering & Computer Science
University of Ottawa

Marzieh Amini
Associate Professor, Department of Systems & Computer Engineering
Carleton Univeristy

Supervisor(s): WonSook Lee
Professor, School of Electrical Engineering & Computer Science
University of Ottawa

Chris Joslin
Professor, Department of Systems & Computer Engineering
Carleton Univeristy

Amy Felty
Professor, School of Electrical Engineering & Computer Science
University of Ottawa

Declaration of Authorship

I hereby certify that this thesis is entirely my own original work except where otherwise indicated. I am aware of the University of Ottawa regulations concerning plagiarism, including those regarding consequent disciplinary actions. Any use of the works of any other author, in any form, is properly acknowledged at their point of use.

Abstract

Achieving reliable perception in dynamic environments while enabling real-time decision-making is critical for practical deployment in autonomous vehicles. The objective of this research is to enhance the accuracy, robustness, and computational efficiency of object detection systems for autonomous driving.

To address the need for efficient and low-latency, we first developed TransfuseNet, a lightweight LiDAR-camera fusion network specifically designed for 2D object detection. TransfuseNet optimizes computational efficiency by leveraging self-attention mechanisms for mid-level feature fusion and introducing a Multi-Convolutional Fusion (MCF) operator that prioritizes essential features. With its compact model architecture and reduced resource consumption, TransfuseNet achieves inference latency below 40ms, making it well-suited for real-time applications where rapid action is required. However, while TransfuseNet effectively balances accuracy and efficiency, it does not explicitly account for sensor reliability variations or provide mechanisms to adapt to degraded sensor inputs.

To overcome these limitations, we introduced ReliFusion, a reliability focused LiDAR-camera fusion framework for 3D object detection. ReliFusion was designed as a more advanced fusion model that integrates LiDAR and camera data for enhanced perception and dynamically adjusts sensor contributions based on real-time reliability assessments. Unlike conventional fusion strategies that assume equal reliability of all modalities, ReliFusion incorporates adaptive mechanisms to ensure robustness under sensor degradation, occlusions, and environmental challenges. It integrates a Spatio-Temporal Feature Aggregation (STFA) module to improve temporal consistency, a Reliability module based on Cross-Modality Contrastive Learning (CMCL) to quantify the trustworthiness of sensor inputs, and a Confidence-Weighted Mutual Cross-Attention (CW-MCA) module to refine fusion weights according to estimated reliability scores. This adaptive approach enables ReliFusion to maintain stable detection performance even in challenging real-world conditions.

Experimental evaluations on the KITTI and nuScenes datasets demonstrate that both TransfuseNet and ReliFusion achieve improved detection accuracy compared to existing fusion-based methods. While TransfuseNet provides an efficient solution for real-time 2D detection, ReliFusion advances multimodal 3D detection by addressing sensor degradation and incorporating dynamic reliability-driven fusion strategies. The findings of this research contribute to the design of sensor fusion-based object detection systems that enhance multimodal perception in autonomous vehicles by addressing key challenges such as sensor degradation, occlusions, and dynamic environmental conditions.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my late supervisor, Professor WonSook Lee, whose guidance, support, and mentorship over the course of four years were instrumental in shaping this research. Her dedication and belief in my work continue to inspire me. Her passing was a profound loss, and I will always be grateful for the opportunity to learn under her supervision.

Following her passing, I was fortunate to have Professor Chris Joslin step in as my supervisor. Although he had been co-supervising my work for the last nine months, his role became vital in helping me navigate this difficult period. His support, guidance, and thoughtful advice were essential in completing this thesis. I truly could not have finished this journey without him.

I would also like to sincerely thank Professor Amy Felty and Professor Herna Viktor for their emotional and financial support during a time of uncertainty. Their kindness and encouragement made a lasting difference in helping me stay focused and complete my studies.

I am equally grateful to my family — my mother, my father, my brother, and my wife — for their unwavering support, love, and patience throughout this long journey. Their belief in me never faltered, and this achievement would not have been possible without them.

Table of Contents

List of Tables

List of Figures

Abbreviations

AP Average Precision 28

BEV Bird's EyeView 60, 62, 68, 75, 90–92, 94, 95

CNN Convolutional Neural Network xix, 38, 42

FV Frontal view 76, 87, 90, 91, 95

IoU Intersection over Union 26, 27, 85, 86

LiDAR Light Detection And Ranging xxi, 21, 74, 78, 80, 82

mAP mean Average Precision 28

MCF Multi-Convolutional Fusion 82, 91, 92, 94, 95

MFB multi-modal factorized bilinear pooling 91, 92, 94

NDS nuScenes Detection Score 29

NMS Non-Maximum Suppression 84, 85

RADAR Radio Detection and Ranging 23, 31

ReLU Rectified Linear Unit 82

RPN Region Proposal Network 83–85

Chapter 1

Introduction

1.1 Motivation and Challenges

Accurate and robust object detection plays a foundational role in enabling safe navigation for autonomous driving systems. Object detection allows self-driving vehicles to perceive and interpret dynamic road environments, identifying obstacles such as vehicles, pedestrians, cyclists, and traffic signs under a wide range of conditions. Reliable perception is critical, particularly when facing complex and unpredictable real-world environments.

Achieving robust object detection is inherently challenging. Environmental factors such as fog, heavy rain, snow, and varying illumination (as shown in Figure 1.1 , either extremely bright sunlight or low-light nighttime conditions) significantly impact the performance of perception systems. These conditions can obscure objects, introduce noise into sensor data, and degrade the ability of a vehicle to correctly classify and localize surrounding elements. For example, fog scatters light, reducing the visibility of distant objects; heavy rain introduces noise into LiDAR returns and camera images; snow and bright sunlight can cause glare and occlusions, confusing perception systems. Consequently, a reliable object detection system must be able to operate robustly across

these adverse scenarios to ensure safe autonomous driving.

Moreover, in critical scenarios, the speed at which objects are detected directly influences the available reaction time for decision-making and collision avoidance. Delays in perception can significantly reduce the vehicle's ability to respond safely. Therefore, timely and real-time object detection is essential, and the specific real-time requirements for autonomous driving will be examined later in this chapter.

Figure 1.1: Challenging conditions in object detection. Left: Overexposed Illumination. Right: High Occlusion.

In addition to environmental challenges, the inherent limitations of individual sensors also present significant obstacles to achieving robust and reliable perception. Cameras and LiDAR, the primary sensors employed in autonomous vehicles, each exhibit specific weaknesses that can compromise detection accuracy under various conditions. Understanding these limitations further motivates the need for advanced sensor fusion strategies that leverage the complementary strengths of multiple modalities.

Cameras, while providing high-resolution semantic information, suffer from poor depth estimation and significant sensitivity to lighting variations. Under low-light conditions, at night, or during adverse weather such as fog or heavy rain, camera-based object detection performance degrades substantially. Motion blur and occlusions can further lead to missed or misclassified objects. These issues become particularly critical in high-speed environments, where reduced reaction time amplifies the consequences of

delayed or inaccurate perception.

LiDAR sensors, on the other hand, provide accurate three-dimensional spatial information independent of ambient lighting. However, LiDAR also exhibits limitations, particularly when encountering transparent or highly reflective surfaces such as glass or wet roads, where laser beams may pass through or scatter unpredictably. Additionally, LiDAR data is often sparse due to the limited number of laser beams and their fixed angular resolution, which results in lower point density, especially at greater distances. It is also subject to degradation under harsh environmental conditions like heavy rain, snow, and fog, where particulate matter interferes with laser returns.

Real-world incidents underline the consequences of perception system failures. In 2016 in Williston, Florida [47], a Tesla Model S operating in Autopilot mode failed to detect a white semi-truck crossing the highway, resulting in a fatal crash . The vehicle's camera system misclassified the truck's surface as part of the bright sky, demonstrating vulnerability under challenging lighting conditions. Similarly, in 2018 in Tempe [48], Arizona, an Uber self-driving test vehicle struck a pedestrian at night due to perception errors. Although LiDAR detected the pedestrian, misclassification and poor camera visibility contributed to delayed braking, ultimately leading to a fatal outcome.

These examples highlight the urgent need for object detection systems capable of maintaining high accuracy and reliability even under adverse environmental conditions. Building such systems requires not only leveraging complementary sensing modalities but also developing adaptive mechanisms that can dynamically respond to sensor degradation and environmental complexity. Addressing these challenges is central to advancing autonomous driving technology toward safer and more reliable real-world deployment.

1.2 Importance of Detecting Specific Object Classes

In autonomous driving, detecting specific object types is critical for safety, situational awareness, and effective decision-making. This research focuses on detecting eleven ob-

ject classes: cars, trucks, trailers, buses, bicycles, motorcycles, pedestrians, construction vehicles, traffic cones, and barriers. Each class represents elements commonly encountered in urban, suburban, and highway environments, and each poses unique risks and challenges for autonomous navigation.

Detecting vehicles such as cars, trucks, trailers, and buses is essential for collision avoidance, lane management, and traffic flow integration. Misidentifying or failing to detect these objects can lead to accidents, especially in dense traffic or at high speeds. Accurate localization and classification of these objects allow autonomous systems to predict their trajectories and make informed driving decisions, such as lane changes, braking, and acceleration.

Pedestrians and cyclists are particularly vulnerable road users. Their motion is less constrained by traffic rules and is highly variable—for example, mid-block crossings, sudden starts/stops, lateral weaving, riding/walking against traffic, and abrupt changes in direction. Because these behaviors reduce available reaction time, failures in detection translate directly into elevated collision risk. Detection systems must be sensitive to variations in posture, size, and movement patterns, particularly in environments where occlusions or low visibility conditions are present.

Traffic cones and barriers serve as temporary or permanent indicators of road construction, lane closures, or accident scenes. Their detection is crucial for enabling autonomous vehicles to adapt their paths dynamically, ensuring compliance with temporary traffic regulations and avoiding restricted areas.

Construction vehicles introduce additional complexity due to their varying shapes, slow and unpredictable movements, and frequent operation in work zones with non-standard traffic patterns. Recognizing these vehicles allows autonomous systems to exercise appropriate caution in construction zones.

Moreover, accurate and rapid classification of detected objects is essential not only for recognizing their presence but also for enabling timely and appropriate responses. In emergency situations where unavoidable collisions might occur, distinguishing between

critical objects such as pedestrians and non-critical objects such as traffic cones becomes vital for ethical decision-making and risk minimization. Thus, fast and precise object classification significantly enhances the system's ability to ensure safety in complex real-world environments.

These eleven classes were selected because they combine frequency, safety impact, and diverse interaction patterns, which together drive the operational complexity of perception and decision-making.

1.3 Latency Requirements for Autonomous Driving Perception

As mentioned, accurate and timely perception is fundamental for safe autonomous driving. In this section, we define *latency* as the total time elapsed between the beginning of sensor data acquisition (e.g., a full LiDAR sweep or camera frame exposure) and the availability of processed detection output. This is distinct from *frame rate*, which specifies how frequently sensors produce new measurements.

In multimodal systems where sensors operate at different rates (e.g., camera at 30 Hz and LiDAR at 20 Hz), synchronization must occur at the slower frequency. Therefore, we treat the LiDAR rate (20 Hz) as the governing cycle, which imposes a 50 ms inter-sweep period. This defines the maximum available throughput budget per perception cycle.

To enable real-time operation, a complete perception–decision–actuation loop must fit within this safety window. We explicitly consider three latency phases:

- **Capture latency** (t_{cap}): Time required for a sensor to acquire one full measurement. For cameras, this is the exposure and readout time (a few ms), while for spinning LiDARs it is approximately one sweep duration (e.g., 50 ms at 20 Hz).

- **Inference latency** (t_{inf}): Processing time from when the sensor data is available until detection outputs are produced (e.g., neural network inference, fusion, post-processing).
- **Control latency** (t_{ctrl}): Time from when detections are available until the vehicle executes a response, including planning, decision-making, and actuator delays (braking or steering).

In practice, capture of the next input overlaps with inference on the previous one. Thus, the constraints are:

$$\begin{aligned} \textbf{Throughput condition:} \quad & t_{\text{inf}} \leq T_{\text{sensor}} \\ \textbf{End-to-end latency:} \quad & t_{\text{E2E}} = t_{\text{cap}} + t_{\text{inf}} + t_{\text{ctrl}} \end{aligned}$$

Definitions and Scope

- **Frame rate** (f_{sensor}): Number of sensor updates per second (Hz).
- **Inter-frame interval** (T_{sensor}): Time between two sensor measurements, computed as:
$$T_{\text{sensor}} = \frac{1}{f_{\text{sensor}}}.$$
- **Latency budget**: The maximum allowable end-to-end latency is limited by the stopping distance constraint (derived below).

Typical Automotive Sensor Rates

Modern automotive perception stacks fuse data from cameras and LiDAR:

- *Cameras*: 25–30 Hz ($T_{\text{camera}} = 33$–40 ms)

- *Spinning LiDARs*: 10–20 Hz ($T_{\text{LiDAR}} = 50$–100 ms)

To synchronize both modalities, the system aligns to the slower LiDAR rate (20 Hz), implying a maximum throughput cycle of 50 ms.

Stopping Distance Model

A vehicle's total stopping distance d_{total} consists of:

$$d_{\text{total}} = d_{\text{capture}} + d_{\text{inf}} + d_{\text{braking}} \tag{1.1}$$

where:

$$d_{\text{capture}} = v \cdot t_{\text{cap}} \qquad \text{(distance traveled while capturing sensor data)} \tag{1.2}$$

$$d_{\text{inf}} = v \cdot t_{\text{inf}} \qquad \text{(distance traveled during inference)} \tag{1.3}$$

$$d_{\text{braking}} = \frac{v^2}{2a} \qquad \text{(distance required to decelerate to zero)} \tag{1.4}$$

with v as the vehicle speed in m/s, and $a = 9$ m/s^2 representing the magnitude of deceleration (aggressive ABS braking [46]).

To ensure safety, the vehicle must detect obstacles early enough to allow completion of all phases of the stopping process—sensor capture, perception/inference, and physical braking. We define d_{detect} as the distance from the object at which the perception system first initiates sensing.

Thus, to avoid a collision:

$$d_{\text{detect}} \geq d_{\text{total}} = d_{\text{capture}} + d_{\text{inf}} + d_{\text{braking}} \tag{1.5}$$

This constraint ensures that the perception system and control stack operate within a safe stopping margin for all operating speeds and latency conditions.

Representative Driving Scenarios

Given a detection range d_{detect}, the available reaction time budget is:

$$t_{\text{budget}} = \frac{d_{\text{detect}} - d_{\text{braking}}}{v} \tag{1.6}$$

The perception budget is then:

$$t_{\text{perc}} = t_{\text{budget}} - t_{\text{cap}} - t_{\text{ctrl}} \tag{1.7}$$

To ensure safety, the inference time must satisfy $t_{\text{inf}} \leq t_{\text{perc}}$. We use $t_{\text{cap}} = 0.05$ s (50 ms) and $t_{\text{ctrl}} = 0.6$ s as conservative planning/actuation overhead [3].

Table 1.1 summarizes representative scenarios. The detection distances were chosen to reflect conservative yet realistic operating conditions. On highways, 100 m corresponds to approximately 2.8 seconds of time-to-collision at 130 km/h, a range that is reliably achievable by modern camera–LiDAR systems though many sensors exceed 150–200 m. For rural arterials, 45 m at 80 km/h reflects conditions with limited sightlines (e.g., curves or vegetation), yielding a reaction budget similar to the highway case. In urban driving, where occlusions from buildings or parked vehicles often restrict visibility, 22 m at 50 km/h is typical and leaves only 0.16 s for inference. This illustrates that, while urban visibility constraints are severe, the highway scenario imposes the tightest perception budget in our settings, with urban remaining highly constrained due to occlusions.

A visual overview of the stopping-distance components and per-scenario budgets is shown in Figure 1.2.

Table 1.1: Perception latency budgets across driving scenarios. t_{budget} is the maximum available reaction time, and t_{perc} is the remaining inference budget after capture and control are subtracted. Detection distances represent conservative yet realistic values for each driving environment.

Scenario	v (km/h)	d_{detect} (m)	t_{budget} (s)	t_{perc} (s)
Highway	130	100	0.76	0.11
Rural Arterial	80	45	0.79	0.14
Urban	50	22	0.81	0.16

* Assumes $t_{\text{cap}} = 0.05$ s, $t_{\text{ctrl}} = 0.6$ s.

Braking vs. Steering Trade-Off

When $d_{\text{detect}} < d_{\text{braking}}$, pure braking is insufficient. The system must initiate emergency steering, which imposes stricter timing. Steering control often requires an additional $\sim$200 ms, reducing the available perception budget. In such cases:

$$t_{\text{perc}}^{\text{steer}} = t_{\text{budget}} - t_{\text{cap}} - (t_{\text{ctrl}} + 0.2). \tag{1.8}$$

For example, under the highway scenario ($t_{\text{budget}} \approx 0.76$ s), subtracting 0.05 s capture and $(0.6 + 0.2)$ s control yields a negative perception budget, i.e., the maneuver is infeasible at that detection range. This highlights the need for either longer detection distances, reduced speeds, or reduced actuation overheads in close-range steering situations.

Summary of Latency Requirements

- **Sensor throughput:** 30 Hz camera $\Rightarrow$ inference $<$ 33 ms; 20 Hz LiDAR $\Rightarrow$ inference $<$ 50 ms.
- **Brake-limited:** Under nominal highway conditions (130 km/h and 100 m detection), $t_{\text{perc}} < 110$ ms.

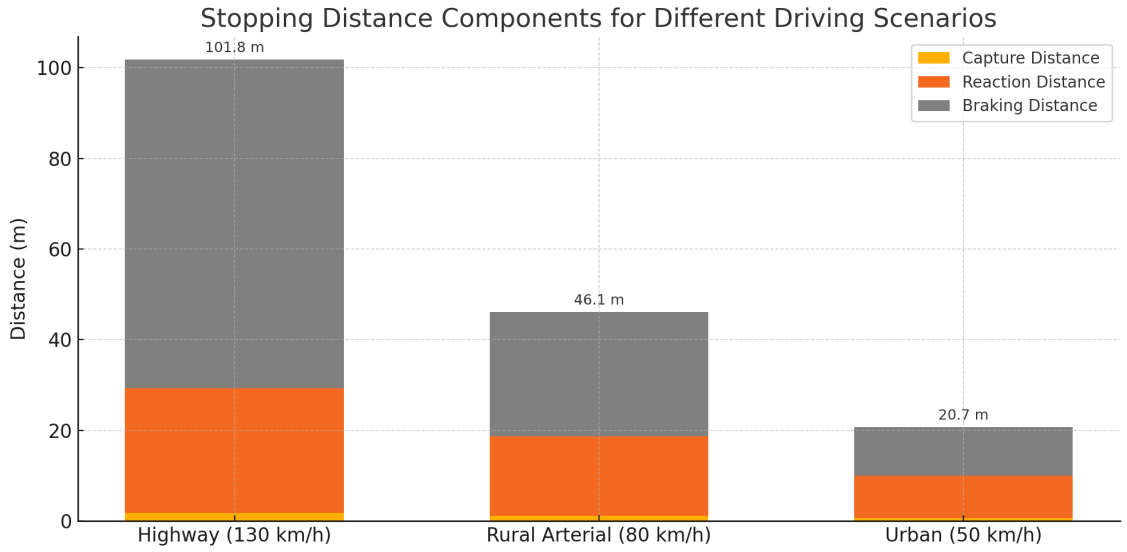

Figure 1.2: Stopping distance components for highway, rural, and urban scenarios, showing contributions from capture (t_{cap}), inference (t_{inf}), control (t_{ctrl}), and braking.

- **Steer-limited:** In close-range emergencies, perception is often infeasible without either earlier detection or faster control.

Together, these analyses show that although a sub-200 ms target may suffice for basic highway braking, robust performance across all driving conditions—especially short-range and emergency steering—demands perception latencies on the order of tens of milliseconds.

Note on Sensor Synchronization

In multimodal perception systems, cameras and LiDAR often operate at different frame rates (e.g., 30 Hz and 20 Hz), which can introduce timestamp offsets between modalities. In this work, we do not model a separate synchronization delay for two reasons. First, modalities are aligned by timestamps to the LiDAR cadence (20 Hz) without imposing

a blocking barrier; capture and inference are overlapped, and fusion is performed at the *feature* level using temporally aggregated camera features (ReliFusion), which is robust to small intermodal skew. Second, the evaluation datasets (KITTI and nuScenes) provide timestamped, approximately synchronized pairs, and standard LiDAR motion compensation (deskew) is applied. This corrects for within-sweep skew, i.e., the fact that a spinning LiDAR records different points at slightly different times during a single revolution, which would otherwise cause motion-induced distortion. After deskewing, the effect is negligible at the operating rates considered.

Under these conditions, residual offsets are small relative to our perception budgets. We conservatively bound synchronization overhead by $t_{\text{sync}} \leq 5$ ms (i.e., $\leq$5% of our tightest budget, $t_{\text{perc}} = 0.11$ s on highway), so conclusions are unchanged. If larger or more variable offsets were present in a deployment (e.g., unsynchronized multi-camera rigs or clock drift), a synchronization term can be included without altering the framework:

$$t'_{\text{E2E}} = (t_{\text{cap}} + t_{\text{sync}}) + t_{\text{inf}} + t_{\text{ctrl}} \quad \text{and} \quad t'_{\text{perc}} = t_{\text{perc}} - t_{\text{sync}}.$$

For example, $t_{\text{sync}} = 5$ ms would reduce the highway perception budget from 0.11 s to 0.105 s. Given timestamp alignment to the LiDAR cadence, feature-level temporal fusion, and deskewed, approximately synchronized datasets, omitting an explicit t_{sync} is justified.

1.4 Research Objectives

The primary objective of this research is to develop robust, efficient, and reliable perception models for autonomous driving. Specifically, this thesis aims to:

- **Achieve real-time performance:** Ensure perception latency of approximately 110 milliseconds or less per input frame under highway driving scenarios, enabling timely decision-making for collision avoidance.

- **Improve robustness under adverse conditions:** Maintain high detection accuracy despite environmental challenges such as fog, rain, snow, glare, and low-light conditions, as well as sensor degradation.
- **Accurately detect critical object classes:** Build models capable of detecting and classifying eleven key object categories—vehicles, pedestrians, cyclists, traffic cones, barriers, and construction vehicles—that represent the most common and safety-critical participants in real-world driving environments.
- **Develop adaptive multimodal fusion:** Leverage complementary strengths of LiDAR and camera sensors through novel fusion strategies that dynamically assess sensor reliability and adapt to degraded inputs.
- **Advance practical deployment:** Deliver models that balance accuracy, robustness, and computational efficiency, ensuring suitability for real-world autonomous vehicle applications.

1.5 Proposed Solution and Contributions

This research adopts a solution based on the fusion of camera and LiDAR data, combined with machine learning techniques, to achieve robust object detection for autonomous driving. By integrating complementary sensing modalities, the perception system can leverage both high-resolution semantic information from cameras and accurate spatial localization from LiDAR.

To address challenges posed by adverse environmental conditions and sensor reliability degradation, this research introduces two fusion-based approaches: an initial lightweight 2D object detection method and a primary 3D reliability-driven methodology.

This thesis makes the following contributions to multimodal perception for autonomous driving:

1. **Formulation of Reliability-Driven Fusion in BEV Space.** We introduce a model-level reliability estimation framework that quantifies the trustworthiness of LiDAR and camera features in a shared embedding space and integrates these estimates directly into a confidence-weighted cross-attention fusion mechanism. Unlike conventional fixed-weight fusion strategies, this approach enables adaptive modality prioritization under sensor degradation.

2. **Integration of Contrastive Learning for Sensor Reliability Estimation.** We propose the use of Cross-Modality Contrastive Learning (CMCL) to align multimodal BEV representations and implicitly detect degraded sensor inputs through embedding consistency. This formulation extends contrastive learning beyond representation alignment to reliability estimation within perception models.

3. **Spatio-Temporal Feature Aggregation for Robust Multimodal Detection.** We design a Spatio-Temporal Feature Aggregation (STFA) module that sequentially applies spatial inter-view attention and temporal cross-frame attention in BEV space. This enhances detection stability and robustness in dynamic and partially occluded environments.

4. **Latency-Constrained Perception Analysis for Autonomous Driving.** We provide a quantitative stopping-distance-based latency formulation that links perception inference time to vehicle dynamics and safety constraints. This establishes explicit perception latency budgets for highway, rural, and urban driving scenarios.

5. **Empirical Validation under Sensor Degradation.** We conduct systematic corruption-based evaluation on benchmark datasets to analyze performance under sensor malfunction scenarios, demonstrating that reliability-driven fusion provides consistent robustness improvements over fixed-weight baselines.

6. **Development of Two Complementary Fusion Architectures.** We design and evaluate both TransfuseNet (a lightweight 2D fusion architecture achieving sub-40 ms inference) and ReliFusion (a 3D reliability-aware architecture achieving robust performance under degraded sensing), providing a progression from efficient fusion to adaptive reliability-driven perception.

1.6 Thesis structure

The remainder of this thesis is organized as follows:

Chapter 2 introduces transformer architectures and their role in addressing complex computer vision challenges, laying the theoretical foundation for subsequent methods.

Chapter 3 presents a comprehensive literature review of recent methods in single-sensor and multimodal object detection, examining existing approaches to address autonomous perception challenges. Alternative fusion strategies are also reviewed.

In Chapter 5 introduces TransfuseNet, our initial fusion method for efficient real-time 2D object detection. Its architecture, implementation, and contributions toward improving speed and efficiency in autonomous driving perception are presented.

Building upon TransfuseNet's limitations, Chapter 6 introduces ReliFusion, a novel and advanced reliability-driven framework for robust 3D object detection. ReliFusion dynamically assesses sensor reliability and adaptively fuses sensor data, overcoming limitations identified in earlier approaches.

Chapter 7 comprehensively evaluates ReliFusion. Extensive experiments are conducted on the nuScenes dataset, including rigorous comparisons with state-of-the-art methods under normal and degraded sensor conditions. Detailed quantitative and qualitative results showcase the robustness and superior performance of ReliFusion.

Finally, Chapter 8 summarizes the research contributions, highlights significant findings, and outlines future research opportunities to further enhance multimodal sensor

fusion for autonomous driving perception.

Related Publications

The following publications are directly related to the work presented in this thesis:

- R. Sadeghian, N. Hooshyaripour, W. S. Lee, "Transformer-based RGB and LiDAR Fusion for Enhanced Object Detection," *International Conference on Pattern Recognition (ICPR)*, pp. 445–460, Springer Nature Switzerland, Dec. 2024.
- R. Sadeghian, N. Hooshyaripour, C. Joslin, W. S. Lee, "Reliability-Driven LiDAR-Camera Fusion for Robust 3D Object Detection," *Canadian Conference on Artificial Intelligence (Canadian AI)*, May 2025. Received the Best Paper Award.
- R. Sadeghian, C. Joslin, "ReliFusion: Reliability-Driven Multimodal Fusion for Real-Time 3D Object Detection in Autonomous Driving" in preparation for submission to *Machine Vision and Applications*, 2025.

Chapter 2

Background

2.1 Introduction

This chapter establishes the theoretical foundations required to understand transformer-based methods and multimodal data fusion for object detection. First, the evolution of object detection methods, particularly the transformative impact of deep learning and transformers, is discussed. Subsequently, key concepts of transformer architectures, their adaptation to vision tasks, and their specific benefits for object detection are presented. Then, we introduce LiDAR technology and discuss its significance and working principles, providing the context for its integration with camera data. Following this, we explain data fusion methodologies, emphasizing the rationale and importance of combining LiDAR with camera data. Lastly, essential metrics used to evaluate object detection performance are introduced, setting the stage for the methodologies described in subsequent chapters.

2.2 Transformers and Their Role in Object Detection

2.2.1 Introduction to Transformers

The transformer architecture, originally introduced by Vaswani et al. in 2017 [67], fundamentally changed the approach to handling sequential data by introducing attention mechanisms. Unlike traditional recurrent neural networks (RNNs) and convolutional networks (CNNs), transformers utilize attention mechanisms to efficiently handle long-range dependencies within sequences. Their design eliminates recurrence and convolution, which enables parallel processing of sequences and overcomes limitations associated with handling long-range dependencies. Initially designed for Natural Language Processing (NLP) tasks, the transformer architecture's success motivated its adaptation to other domains, including computer vision.

2.2.2 Core Components of Transformer

The transformer architecture comprises sophisticated components designed to manage long-range dependencies and effectively represent sequential data. The main components include self-attention mechanisms, multi-head attention, positional encoding, feedforward neural networks, and normalization layers, as illustrated in Figure 2.1.

Self-Attention Mechanism: Self-attention allows transformers to dynamically evaluate the relevance of each token in relation to others within a sequence. Formally, attention scores between tokens are computed as:

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

where Q, K, and V are Query, Key, and Value vectors, respectively.

Multi-Head Attention: To capture multiple types of relationships simultaneously, multi-head attention employs several parallel self-attention operations. Each attention

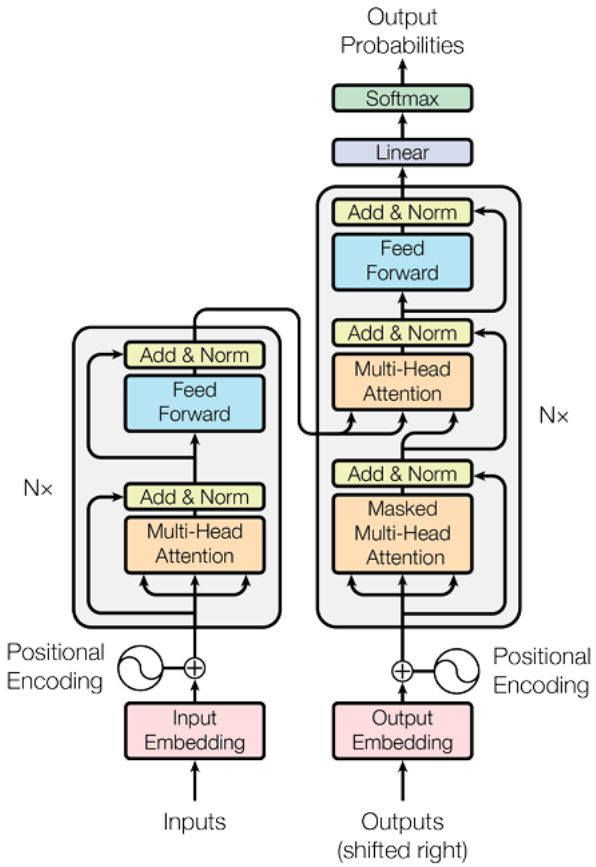

Figure 2.1: The Transformer Architecture. This image is reprinted from [67].

head calculates distinct attention scores, enabling the model to handle diverse dependencies concurrently (Figure 2.2).

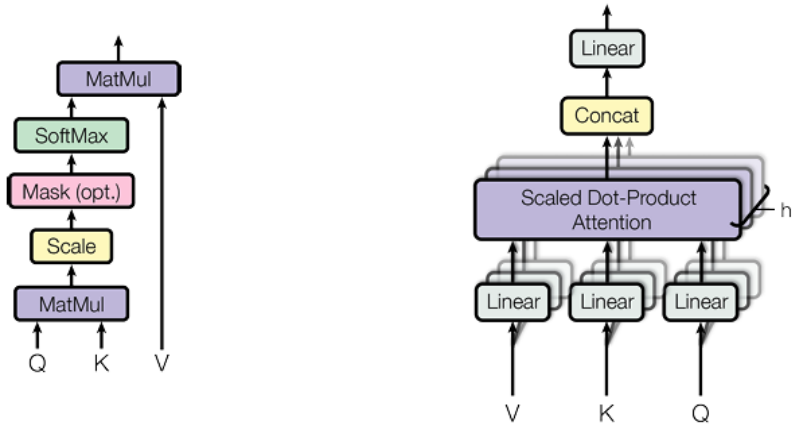

Figure 2.2: Left: Scaled Dot-Product Attention. Right: Multi-Head Attention . This image is reprinted from [67].

Positional Encoding: Since the transformer architecture lacks inherent positional information, positional encoding is added to input embeddings to indicate token positions. Positional encodings are generated using sine and cosine functions defined as:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right), \quad PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{2.2}$$

Here, *pos* denotes the index of the token in the input sequence (e.g., the position of a word or image patch), and i is the index of the embedding dimension. Even dimensions $(2i)$ are assigned sine values and odd dimensions $(2i+1)$ cosine values. The term d_{model} represents the embedding size, while the denominator $10000^{2i/d_{\text{model}}}$ controls the wavelength of the sinusoid so that lower-index dimensions encode fine-grained positional variations (shorter wavelengths) and higher-index dimensions capture broader trends (longer wavelengths). Together, these sinusoidal encodings give each token a unique numerical signature that reflects both its absolute position in the sequence and its relative distance to other tokens. This enables the transformer to distinguish whether two tokens

(or image patches) are adjacent or far apart, even though the architecture itself does not impose any sequential or spatial structure. For vision tasks, this mechanism preserves spatial layout information that would otherwise be lost when an image is flattened into a sequence of patches.

Feedforward Neural Networks: Transformers integrate feedforward neural networks after attention layers, enabling intricate feature transformations that enhance data representation.

Residual Connections and Layer Normalization: Residual connections facilitate smoother gradient flow through the network, and layer normalization stabilizes training, enabling deeper and more efficient architectures.

2.2.3 Adaptation of Transformers to Vision Tasks

To adapt transformers from NLP to computer vision, an input image is first divided into non-overlapping patches (e.g., 16×16 pixels). Each patch is then flattened into a vector and projected through a linear layer to obtain a fixed-length embedding. These patch embeddings, together with positional encodings, form a sequence that can be processed by the transformer in the same way as a sequence of words in NLP. This approach, introduced by the Vision Transformer (ViT) [12], converts two-dimensional spatial information into a one-dimensional sequence representation suitable for attention-based processing. While ViT demonstrated strong performance on vision benchmarks, transformer-based vision models typically require significantly larger training datasets and greater computational resources than convolutional neural networks (CNNs), since CNNs benefit from inductive biases such as locality and translation invariance that reduce data requirements.

2.2.4 Benefits of Transformers in Object Detection

The adoption of transformers in object detection offers several distinct advantages:

Efficient Global Context Modeling: The self-attention mechanism enables transformers to model global context by effectively capturing relationships between spatially distant regions of an image, improving object detection performance through better context modeling.

Adaptability and Scalability: Transformers naturally accommodate varying input sizes and object scales because they divide the image into fixed-size patches and process these patches uniformly through self-attention. Unlike CNNs, which rely on convolutional kernels with fixed receptive fields and often require handcrafted mechanisms such as image pyramids or multi-scale feature maps, transformers can directly attend across the entire image. This property allows them to capture both small and large objects within the same architecture without extensive reconfiguration.

Simplified Pipeline and End-to-End Learning: Transformers enable streamlined, end-to-end model architectures that integrate multiple steps within a unified framework. For example, detection transformers (such as DETR) eliminate the need for region proposal networks, anchor design, and non-maximum suppression—components commonly required in CNN-based detectors. Instead, transformers learn object queries and directly predict final bounding boxes and classes in a single optimization process. This simplifies the overall detection pipeline and allows all components to be trained jointly from raw image data to final predictions.

2.3 LiDAR Technology

2.3.1 Introduction to LiDAR and Its Significance in Object Detection

LiDAR is a remote sensing technology that measures distances by emitting laser pulses and calculating the time it takes for these pulses to reflect back from surrounding surfaces. The result is a dense three-dimensional (3D) point cloud that encodes the geom-

etry of the environment with high accuracy. Compared to cameras, which capture rich texture and color but lack reliable depth information, LiDAR directly provides metric depth, making it particularly valuable for object localization and size estimation. Unlike RADAR, which is robust to weather but offers coarse resolution, LiDAR achieves fine-grained spatial detail that supports the detection of small or partially occluded objects. This combination of precision and robustness makes LiDAR especially beneficial for safety-critical tasks such as autonomous driving, robotics, and environmental monitoring, where reliable perception of the surrounding scene is essential.

The key benefits of integrating LiDAR into object detection systems include:

- **High Precision and Accuracy:** LiDAR's capability to provide precise spatial measurements ensures accurate object localization, essential for reliable detection performance.
- **Robustness Under Varied Conditions:** LiDAR maintains effectiveness across varying lighting conditions and environmental challenges, such as nighttime, fog, or dust, overcoming limitations typical in camera-based systems.
- **Detailed 3D Data Representation:** LiDAR provides comprehensive 3D point cloud data, capturing detailed geometric structures essential for accurate object localization and classification.

2.3.2 Working Principles of LiDAR

The basic operational principle of LiDAR technology involves emitting laser pulses, receiving their reflections from objects, and calculating distances based on the measured return times. This process can be broken down into the following essential steps:

1. **Laser Emission:** The LiDAR sensor emits laser pulses into the environment, which propagate through space and strike surrounding surfaces.

2. **Reflection of Laser Pulses:** The emitted pulses are reflected back from object surfaces and received by the LiDAR sensor.

3. **Distance Calculation:** The sensor measures the time elapsed between the emission of a laser pulse and the reception of its reflection, known as the time-of-flight (ToF). The distance (D) to the reflecting surface is calculated as:

$$D = \frac{c \cdot t}{2} \tag{2.3}$$

where c is the speed of light and t is the measured round-trip time.

By combining the measured distance with the known orientation of the emitted laser beam (determined by the sensor's scanning mechanism), the LiDAR system computes the precise three-dimensional coordinates (x, y, z) of the reflecting point relative to the sensor. Repeating this process for thousands of laser beams per second and across different scanning angles generates a dense 3D point cloud that captures the geometry of the surrounding environment. Depending on the sensor design, this can be achieved using rotating mechanical mirrors (spinning LiDARs) or solid-state phased arrays, each with trade-offs in field of view, resolution, and robustness.

2.4 Data Fusion Principles

Data fusion involves integrating information from multiple sensor modalities, such as LiDAR, cameras, or Radio Detection and Ranging (RADAR), to produce a comprehensive environmental representation for object detection. This process leverages the strengths of each sensor to mitigate their individual limitations, such as limited depth accuracy in camera-based systems or low spatial resolution in RADAR. Furthermore, effective fusion methods provide redundancy, improving system robustness against sensor malfunctions or degraded environmental conditions. By managing and reducing uncertainty inherent to individual sensors, data fusion contributes to more accurate object localization and

reliable decision-making in complex autonomous driving scenarios. The methodologies behind different fusion approaches are discussed in subsequent sections.

2.4.1 Early Fusion, Late Fusion, and Mid-level Fusion

There are three primary types of data fusion regarding when to apply the fusion operation: Early Fusion, Late Fusion, and Mid-level Fusion. Each type has distinct approaches and implications for integrating information from multiple sources effectively.

Early fusion, also known as feature-level fusion (Figure 2.3), involves merging information from different modalities at the earliest stage of processing. In this approach, the raw data or features from each modality are combined to form a unified and comprehensive feature representation. This fused representation is then used as input for subsequent processing steps, such as object detection algorithms [25,69]. Early fusion integrates data from multiple sensor modalities at the initial processing stage, constructing a unified feature representation from the combined input data. This method preserves complementary information across modalities, facilitating a comprehensive representation for subsequent analysis and detection tasks.

Late fusion, or decision-level fusion (Figure 2.4), takes a different approach by processing each modality independently up to a certain point in the analysis. The outputs or decisions from each modality are then combined or fused at a later stage, typically at the decision-making level. This fusion can occur using various strategies, including combining decision scores, probabilities, or classification results. Late fusion allows for specialized processing of each modality up to the point of fusion, enabling the system to leverage the unique characteristics and strengths of each source individually. This approach is particularly useful when modalities have distinctive features or when they require specialized processing methods [73, 81].

Late fusion is versatile and adaptable to scenarios where modalities have varying levels of reliability or when combining modalities at an earlier stage might lead to infor-

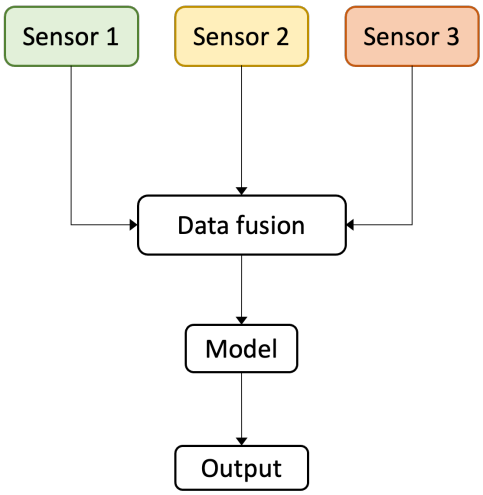

Figure 2.3: Early fusion (feature-level fusion) combines raw data or features from different modalities into a unified representation.

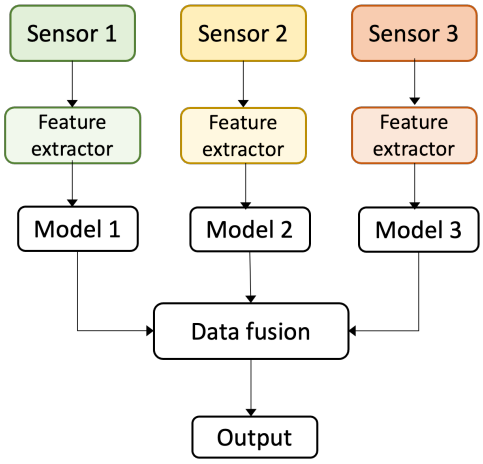

Figure 2.4: Late fusion (decision-level fusion) processes each modality separately, combining outputs or decisions at a later stage, often in decision-making.

mation loss or confusion.

Mid-level fusion, as the name suggests, combines elements of both early and late fusion approaches (Figure 2.5). It involves creating a fused representation at an intermediate level of processing, where features from different modalities are combined. Mid-level fusion integrates modality-specific information at an intermediate processing stage, after which each modality undergoes further separate processing. This strategy balances the comprehensive representation achieved by early fusion and the modality-specific benefits provided. Mid-level fusion is designed to capture the synergies between modalities while allowing for specialized analysis, providing an adaptable and efficient fusion strategy [83].

The choice of fusion type is a critical decision in the design of a multimodal object detection system and depends on various factors. These factors include the nature of the application, the characteristics of the data from each modality, the computational complexity of the system, and the desired balance between combining information early

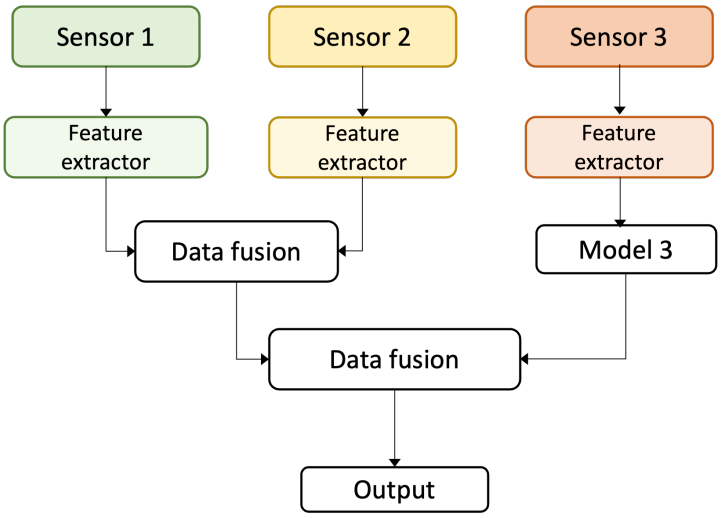

Figure 2.5: Mid-level Fusion. Mid-level fusion blends elements of both early and late fusion by combining features from different sources at an intermediate stage

versus late in the processing pipeline. Selecting the appropriate fusion strategy is essential to maximize the benefits of multimodal data integration based on the objectives and constraints of the specific application.

2.5 Evaluation Metrics for Object Detection

Intersection over Union (IoU) [26] is a fundamental metric used to measure the overlap between the predicted bounding boxes and the ground truth bounding boxes for the objects. It quantifies the extent to which the predicted and ground truth bounding boxes align. The IoU is calculated by dividing the area of overlap between the predicted and ground truth bounding boxes by the area of their union.

A high IoU value indicates that the predicted bounding box accurately aligns with the ground truth, and it is computed using the following formula:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \tag{2.4}$$

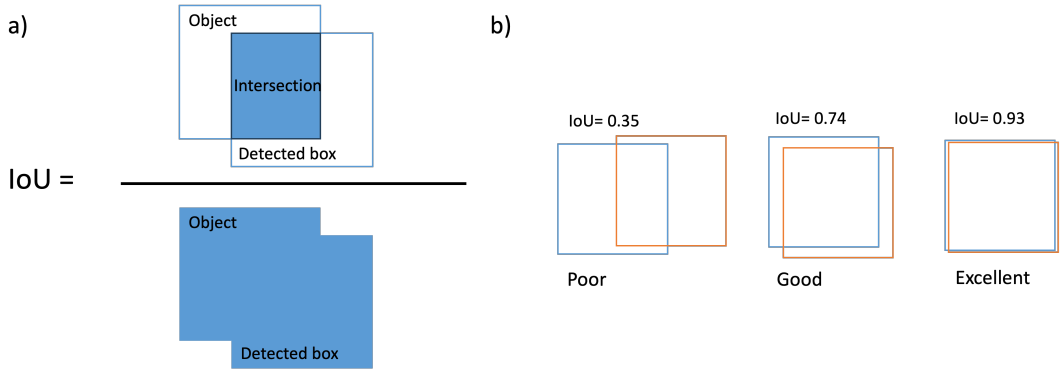

Figure 2.6: (a) Calculation of IoU involves dividing the intersecting area of two boxes by their combined area. (b) Demonstrations of IoU values for various box alignments.

IoU values closer to 1 imply better object localization accuracy, while values closer to 0 indicate poor localization (Figure 2.6). Models often use IoU as a threshold to filter out false positives, setting a minimum IoU to consider a detection as valid. Moreover, IoU is central to non-maximum suppression, a technique used to merge and eliminate redundant bounding boxes, retaining the most accurate detection.

Recall, also known as sensitivity or true positive rate, measures the model's ability to capture all the actual positives. It signifies the proportion of true positive predictions to the total actual positives, including both true positives and false negatives. High recall implies a low false negative rate, ensuring the model captures a significant portion of actual positives. The formula to calculate recall is as follows:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2.5}$$

Precision is a metric that measures the accuracy of the positive predictions made by the model. It represents the proportion of true positive predictions to the total predicted positives, including both true positives and false positives. High precision implies a low false positive rate, ensuring that most predicted positives are true positives.

Precision is defined mathematically as:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2.6}$$

Average Precision (AP) in object detection quantifies a model's ability to accurately identify and locate objects across various classes. It does so by balancing precision, the model's correctness in identifying relevant objects, with recall, its completeness in finding all relevant instances. For each object class, AP is calculated by plotting a precision-recall curve based on the model's predictions at different confidence thresholds, then computing the area under this curve. This process ensures that the model's performance is evaluated over a range of conditions, reflecting its sensitivity and specificity.

mean Average Precision (mAP), an extension of AP, averages the AP scores across all object classes, providing a single comprehensive metric of overall detection performance. By aggregating performance metrics across multiple object classes, mAP provides a comprehensive measure of detection effectiveness, facilitating objective comparisons between models and datasets.

nuScenes Detection Score (NDS) is the composite metric adopted in the nuScenes benchmark to provide a more holistic evaluation of object detection performance. It balances detection accuracy with the quality of localization, scale, orientation, velocity, and attribute estimation. Unlike conventional benchmarks that rely solely on IoU–based mAP, nuScenes evaluates detections using center distance in the ground plane, thereby decoupling localization accuracy from bounding box geometry and attributes.

In addition to mAP, NDS incorporates several true–positive (TP) quality metrics for predictions matched to ground truth within 2 meters:

- **ATE (Average Translation Error):** Euclidean distance error between predicted and ground-truth object centers in the ground plane, measured in meters [m].
- **ASE (Average Scale Error):** $1 -$ IoU after aligning the predicted and ground-truth boxes in translation and yaw. This metric penalizes mismatches in object size rather than location.
- **AOE (Average Orientation Error):** Absolute yaw (heading) difference between predicted and ground-truth boxes, expressed in radians [rad].
- **AVE (Average Velocity Error):** Difference in velocity magnitude between predicted and ground-truth objects, measured in meters per second [m/s].
- **AAE (Average Attribute Error):** $1-$ attribute accuracy, where "attributes" refer to object state properties such as whether a vehicle is moving or stopped, or whether a pedestrian is standing or sitting. This metric quantifies errors in predicting such categorical attributes.

The overall NDS is defined as

$$\mathrm{NDS} = \frac{1}{10}\left[5 \cdot \mathrm{mAP} + \sum_{m\mathrm{TP} \in \{m\mathrm{ATE},\, m\mathrm{ASE},\, m\mathrm{AOE},\, m\mathrm{AVE},\, m\mathrm{AAE}\}} \big(1 - \min(1, m\mathrm{TP})\big)\right]. \quad (2.7)$$

where mAP denotes mean Average Precision, and the additional terms correspond to the mean values of the true–positive quality metrics: mATE (Average Translation Error), mASE (Average Scale Error), mAOE (Average Orientation Error), mAVE (Average Velocity Error), and mAAE (Average Attribute Error). Thus, NDS $\in [0, 1]$, where mAP contributes half of the score and the remaining half reflects how well detections preserve geometric and dynamic fidelity, including accurate localization, orientation, velocity, scale, and attribute estimation. A higher NDS value indicates not only that objects are detected, but also that their predicted positions, sizes, and dynamic states closely match the ground truth, making it a more comprehensive benchmark than mAP alone.

2.6 Summary

This chapter introduced the theoretical foundations for the methods developed in this thesis. Transformer architectures were reviewed, highlighting their key components—self-attention, multi-head attention, and positional encoding—and their adaptation from NLP to vision tasks. The principles and advantages of LiDAR technology were then described, emphasizing its ability to provide precise 3D spatial information for object detection. Different data fusion strategies, namely early, late, and mid-level fusion, were outlined, together with their respective benefits and trade-offs. Finally, evaluation metrics including IoU, precision, recall, AP, mAP, and the nuScenes Detection Score (NDS) were defined, establishing the criteria used to assess object detection models in later chapters.

Chapter 3

Literature Review

Object detection has grown significantly over the past two decades due to advancements in sensing technologies and machine learning methods. Early computer vision research largely targeted 2D bounding box detection due to widespread applications such as face recognition, security surveillance, and image categorization. However, the growing demand for precise spatial understanding in domains such as robotic manipulation, augmented reality, industrial automation, medical imaging, and particularly autonomous driving has shifted research focus toward accurate three-dimensional (3D) object detection. Autonomous driving has notably accelerated this transition, emphasizing rigorous 3D localization of various dynamic objects, including vehicles, pedestrians, and cyclists, to enable safe navigation.

This shift towards comprehensive spatial perception required integrating diverse sensor modalities beyond conventional cameras. Modern 3D object detection systems now commonly leverage multiple complementary sensing modalities—RGB cameras, LiDAR, and RADAR—each characterized by distinct strengths and inherent limitations that profoundly influence detection methodologies. Cameras provide dense, high-resolution visual data essential for semantic interpretation, LiDAR offers precise depth measurements independent of ambient lighting, and RADAR ensures robust detection under

adverse weather conditions while providing direct velocity measurements.

Given the complementary characteristics of these sensors, extensive research has explored their combined use, aiming to mitigate individual modality limitations and achieve reliable, accurate, and robust object detection across diverse environmental and operational scenarios. The following subsections provide an overview of these sensor modalities, highlighting their strengths, constraints, and foundational roles within contemporary object detection frameworks discussed comprehensively in subsequent sections.

3.1 Camera-Based Object Detection

Object detection using camera imagery has been extensively studied in computer vision and remains widely used in autonomous perception systems. Cameras provide dense, high-resolution visual data that include semantic cues such as color, texture, and shape, which are important for detecting traffic participants and environmental features, including vehicles, pedestrians, traffic signs, and lane markings. Their relatively low cost compared to LiDAR and ubiquity make them suitable for many real-time applications in autonomous vehicles.

This section reviews major developments in camera-only object detection models, organized into two primary groups: (i) convolutional neural network (CNN)-based methods, including both two-stage and single-stage frameworks, and (ii) transformer-based approaches that employ self-attention mechanisms to capture spatial relationships. Each group is examined in terms of architectural design, computational cost, and detection performance under various conditions.

The limitations commonly observed in camera-based systems—such as reduced detection accuracy in the presence of occlusions, scale variation, or low illumination—are highlighted in benchmark studies. These issues are particularly evident in speed–accuracy

trade-offs and robustness under different environmental conditions. These limitations provide the motivation for multi-sensor fusion approaches discussed in later sections.

3.1.1 Convolutional Methods for 2D Detection

Early advances in deep learning-based object detection were primarily driven by convolutional neural networks (CNNs), which replaced handcrafted feature extraction with learned representations [18, 29]. CNN-based methods for 2D object detection are commonly categorized into two-stage and single-stage approaches, based on whether they use an explicit region proposal step.

3.1.1.1 Two-Stage CNN-Based Methods

The Region-based Convolutional Neural Network (R-CNN) [18] was one of the first methods to apply CNNs for object detection. It generated candidate object regions using the Selective Search algorithm [66], extracted features from each region using a CNN, and classified them using support vector machines (SVMs). While R-CNN significantly improved detection accuracy compared to traditional methods, it was computationally inefficient due to redundant CNN evaluations on each proposal.

Fast R-CNN [17] addressed this inefficiency by applying the CNN to the entire image once and using a Region of Interest (RoI) pooling layer to extract features for each proposal. This approach reduced computation time while maintaining accuracy. However, it still relied on external region proposal methods such as Selective Search, which remained a bottleneck.

Faster R-CNN [55] introduced the Region Proposal Network (RPN), integrating proposal generation directly into the CNN pipeline. This unified framework improved both speed and accuracy and became a widely adopted baseline. Nevertheless, Faster R-CNN relies on predefined anchor boxes with fixed scales and aspect ratios. These anchor set-

tings require manual tuning and may perform suboptimally in scenes with significant object scale variation or occlusion.

3.1.1.2 Single-Stage CNN-Based Methods

Single-stage detectors simplify the pipeline by predicting object classes and bounding boxes directly from feature maps, eliminating the need for a region proposal stage. YOLO (You Only Look Once) [52] was among the first single-stage models and introduced a grid-based prediction scheme for real-time performance. Subsequent YOLO versions introduced several architectural improvements:

- YOLOv2 [53] introduced anchor boxes, predefined bounding boxes with fixed scales and aspect ratios used as references for object localization. During training, the network learns to adjust these anchors to match ground-truth objects, which significantly improves performance across varying object sizes.
- YOLOv3 [54] adopted feature pyramid networks (FPNs) to enable multi-scale detection, enhancing the ability to detect small and large objects simultaneously.
- YOLOv4 [2] further improved detection by incorporating advanced training optimizations, including the CSPDarknet53 backbone and Complete IoU (CIOU) loss [85], which improves bounding-box regression stability.
- YOLOv5 [27] refined the framework by adding automatic anchor computation, improved training pipelines, and novel data augmentation strategies such as mosaic augmentation, where multiple images are combined to increase data diversity.
- YOLOv6 [31] emphasized deployment efficiency by incorporating quantization-aware training and optimized inference pipelines, making it suitable for industrial applications.

- YOLOv7 [68] introduced the Extended Efficient Layer Aggregation Network (E-ELAN) and auxiliary training heads, striking a balance between speed and accuracy, and achieving state-of-the-art results on several benchmarks.
- YOLOv8 [65] transitioned to an anchor-free detection head, reducing the reliance on manually defined anchors. It also integrated improved model scaling strategies and advanced data augmentation, offering a more flexible framework for diverse real-time tasks.

Despite these advances, the YOLO family of models still faces notable challenges. They remain sensitive to partial occlusion, where objects are only partly visible (e.g., a pedestrian behind a parked car), leading to detection failures. Small objects also remain difficult to detect due to the coarse grid representation in earlier versions, although FPNs have partially mitigated this issue. Additionally, YOLO models are affected by class imbalance, where abundant categories such as cars dominate training data and reduce detection accuracy for less frequent classes such as cyclists.

In summary, YOLO models represent a major step forward in single-stage detection, offering real-time speed with competitive accuracy. However, their limitations in handling occlusion, small objects, and class imbalance highlight the need for complementary or hybrid approaches, motivating continued research into transformer-based and multimodal fusion methods.

The Single Shot MultiBox Detector (SSD) [39] addressed some of YOLO's early limitations by using multiple feature maps of different resolutions to detect objects at varying scales. SSD introduced a set of predefined anchor boxes—also known as default boxes—with fixed scales and aspect ratios at each feature map location. These anchors serve as reference bounding boxes that the network adjusts during training to match ground-truth object locations. However, the use of fixed anchor configurations imposes constraints: it assumes prior knowledge of the object size and shape distributions in the dataset and lacks adaptability to scenes with high object scale variation or uncommon

aspect ratios. As a result, SSD's detection accuracy is reduced in scenarios involving small, overlapping, or irregularly shaped objects.

RetinaNet [37] introduced the Focal Loss function to address the extreme class imbalance in dense detection tasks. This loss function reduces the contribution of easy negatives—background anchors or regions that are trivially classified as non-objects—during training, thereby preventing them from overwhelming the gradient updates and improving performance on crowded scenes. However, RetinaNet's added complexity results in higher inference time compared to simpler detectors.

EfficientDet [63] applied compound scaling from EfficientNet [62], jointly optimizing depth, width, and resolution. EfficientDet offers a balanced trade-off between accuracy and computational cost but continues to face challenges in detecting occluded or small objects and maintaining performance under low-light conditions.

While CNN-based 2D detectors have achieved notable progress, their performance often degrades under challenging conditions. For example, occlusion (e.g., a pedestrian partly hidden behind a vehicle) prevents CNNs from capturing complete object features. Cluttered scenes introduce overlapping textures and background noise, which can confuse convolutional filters and increase false detections. Scale variation, where objects appear at vastly different sizes depending on their distance from the sensor, is difficult to handle with fixed receptive fields and anchor configurations. These limitations highlight the need for models that can capture global context and adapt more flexibly to varying input conditions, motivating the development of transformer-based methods, discussed in the following subsection.

3.1.2 Transformer-Based Camera Methods

The integration of transformers in vision tasks has advanced object detection paradigms, providing mechanisms to capture global context effectively through self-attention operations. The introduction of the Vision Transformer (ViT) [12] marked a turning point,

demonstrating that self-attention could replace convolutions by treating images as sequences of non-overlapping patches and enabling direct modeling of long-range dependencies (Figure 3.1). Although initially designed for image classification, ViT inspired a range of detection models that extended its patch-based representation to localization tasks.

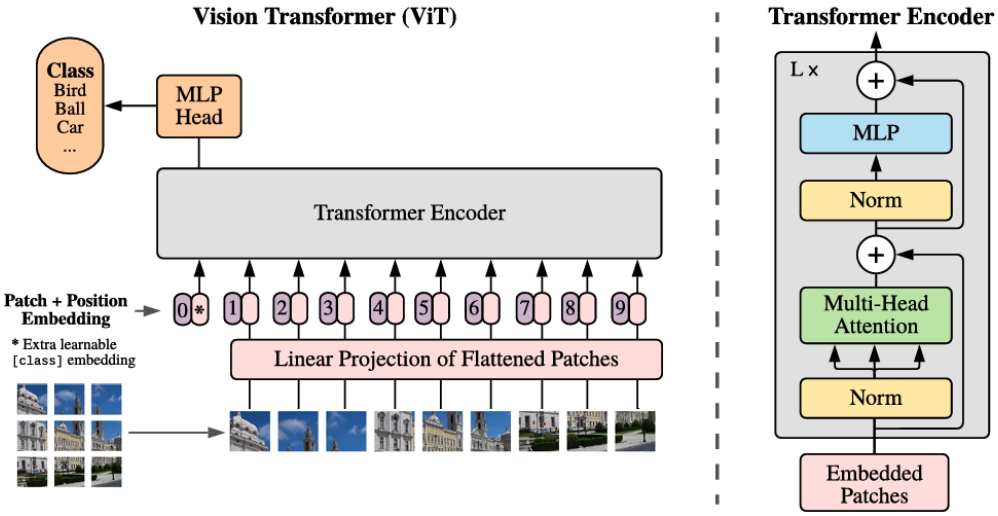

Figure 3.1: ViT architecture. An image is divided into uniform-sized patches, each linearly embedded. Position embeddings are added, and the resultant sequence of tokens is processed by a stack of L transformer encoder layers, each consisting of multi-head attention, a normalization layer (Norm), and a feed-forward multi-layer perceptron (MLP). A special learnable classification token is used for the final classification task. This image is reprinted from [12].

In object detection, ViT-based backbones have been adopted to improve global context modeling compared to CNNs. DETR [6] represented a major shift by eliminating handcrafted anchors and region proposals, instead formulating detection as a *set prediction problem*. In this paradigm, the model directly predicts a fixed-size set of bounding boxes and class labels in a single forward pass (Figure 3.2). A bipartite matching step, implemented with the Hungarian algorithm, then pairs each predicted box with a ground-truth object in a one-to-one manner, ensuring that every prediction is uniquely assigned.

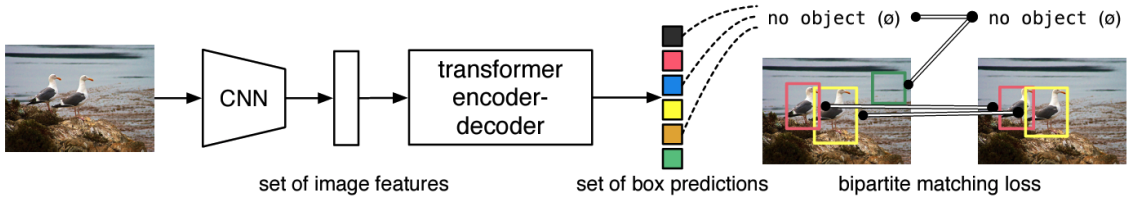

Figure 3.2: DETR architecture. DETR predicts the final set of detections by integrating a standard CNN with a transformer structure. Predictions that do not correspond to any match are expected to result in a prediction of the "no object" class. This image is reprinted from [6].

This reformulation simplified the detection pipeline and provided competitive accuracy, but it also introduced challenges such as slow convergence and reduced sensitivity to small objects due to coarse spatial representations.

To address these limitations, Deformable DETR [88] introduced sparse attention mechanisms that accelerate training and improve detection of smaller objects. Similarly, hybrid approaches such as the Swin Transformer [40] combined transformer self-attention with CNN-like hierarchical structures, achieving strong results on benchmarks like COCO by capturing both local and global contexts efficiently. Despite these advances, such models remain computationally expensive and highly data-dependent, limiting deployment in real-time or resource-constrained environments.

Expanding into 3D, DETR3D [72] adapted transformer-based methodologies to monocular object detection by directly predicting 3D bounding boxes from single-camera images. By leveraging explicit 3D positional encodings and multi-scale feature maps, DETR3D improved spatial localization, though challenges such as depth ambiguity, occlusions, and calibration inaccuracies persist.

In summary, transformer-based methods have reshaped the design of modern detection architectures by enabling global context modeling and reducing reliance on handcrafted components such as anchors or proposal networks. Models such as ViT, DETR, and their variants have demonstrated strong accuracy gains on benchmarks, particu-

larly in complex scenes where long-range dependencies and cross-object relationships are critical. However, these benefits come at a cost: training requires large-scale annotated datasets to prevent underfitting, inference is computationally expensive due to quadratic attention complexity, and performance often degrades on small objects or under occlusion because of coarse patch-level representations. Furthermore, most transformer-based detectors struggle with real-time deployment, as their latency and memory footprints exceed the constraints of automotive-grade hardware.

These limitations highlight several open research challenges: improving data efficiency through pretraining or domain adaptation, designing lightweight or hierarchical attention mechanisms that scale better with resolution, and integrating transformers with convolutional backbones to balance global reasoning with local detail preservation. Addressing these challenges is important for advancing domain-adaptive and resource-aware transformer models that are robust enough for safety-critical applications such as autonomous driving.

3.2 LiDAR-Based Object Detection

LiDAR sensors have been widely adopted in autonomous driving due to their ability to capture precise three-dimensional spatial information by emitting laser pulses and measuring their time-of-flight. Compared to camera sensors, LiDAR provides reliable depth measurements largely unaffected by lighting conditions, which has motivated extensive research into 3D localization, obstacle detection, and scene reconstruction [50, 87].

Despite these advantages, LiDAR data introduces challenges that have shaped detection research. Point clouds are sparse and non-uniform, with density decreasing at greater distances, and the unordered structure makes them incompatible with standard convolutional operations. Moreover, the lack of appearance cues (e.g., color, texture) limits semantic understanding.

To address these challenges, various methods have been proposed to interpret LiDAR

point clouds for object detection. These methods can be broadly categorized into three groups:

1. **Point-based methods**, which operate directly on the raw point cloud without intermediate representations [50, 51, 57, 70, 75, 84];
2. **Voxel-based methods**, which convert the 3D space into structured volumetric grids to leverage convolutional architectures [21, 43, 56, 78, 86, 87];
3. **Projection-based methods**, which map point clouds into 2D representations to exploit mature 2D CNN backbones [7, 9, 13, 15, 36, 61].

Each category is reviewed in the following subsections, with a focus on model architecture, feature representation strategies, computational efficiency, and detection accuracy in various driving scenarios. Particular attention is given to how these methods manage point sparsity, object scale variation, and occlusion—factors that significantly affect LiDAR-based detection performance.

3.2.1 Point-Based Methods

Point-based methods operate directly on raw LiDAR point clouds without requiring intermediate representations such as voxels or images. These approaches aim to preserve the full geometric fidelity of the data and avoid the quantization errors associated with grid-based transformations.

Earlier work by Rusu et al. [57] explored segmentation strategies that cluster point clouds based on spatial proximity or surface properties, such as grouping nearby points into clusters or aggregating regions with similar surface normals. Other approaches attempted to detect simple geometric primitives like planes. While effective in reducing computational complexity for downstream tasks, these methods relied heavily on hand-engineered rules (e.g., manually set distance thresholds or curvature limits) and therefore lacked generalization across diverse scenes.

Guo et al. [20] reviewed various local geometric descriptors that capture properties such as surface curvature, normal orientation, and spatial arrangement. These descriptors are typically computed from neighborhoods of points, encoding geometric relationships such as angle distributions or curvature changes. While they provide valuable features for classification and segmentation tasks, they often require manual parameter tuning (e.g., neighborhood size) and remain sensitive to point density variations, which can degrade robustness in large-scale or sparse environments.

PointNet [50] introduced a significant advancement by proposing a deep learning architecture that directly consumes unordered point sets. It uses symmetric aggregation functions (e.g., max pooling) to achieve permutation invariance. Although PointNet improved classification and segmentation performance, its lack of local neighborhood modeling limits its effectiveness in detecting fine-grained or partially occluded objects.

To address this, extensions such as PointNet++ [51] introduced hierarchical feature learning by grouping points into local neighborhoods and recursively aggregating features. This improved the ability to capture fine-grained geometric structures but came at the cost of repeated neighborhood search and grouping operations. As a result, PointNet++ suffers from high computational overhead and memory usage, making it difficult to scale efficiently to large outdoor point clouds with millions of points, such as those in autonomous driving datasets.

Octree-based CNN (O-CNN) [70] proposed using octree structures to enable hierarchical spatial partitioning of point clouds. This approach reduced memory consumption and allowed CNNs to be applied more efficiently. However, managing dynamic scene content with octrees requires frequent tree updates and traversal, which increases computational overhead, and their hierarchical structure is less flexible for real-time applications where rapid adaptation to changing point densities is needed.

More recent approaches aimed to improve detection in sparse or occluded settings. For example, BtcDet [75] employs a shape occupancy network to infer missing object parts from partial observations. While this improves accuracy in occlusion-heavy scenes,

it introduces significant computational overhead due to the complexity of shape completion.

Instance-Aware Single-Stage Detector (IA-SSD) [84] adopts instance-aware downsampling to preserve key structural information during point selection. By using class-aware and centroid-aware sampling strategies, it improves detection of small or distant objects. However, the method's performance is contingent on accurate initial semantic segmentation, which may be degraded under sparse or noisy conditions.

Overall, point-based methods maintain the advantages of working directly on unstructured data but often encounter trade-offs between accuracy, efficiency, and scalability. These limitations motivate hybrid approaches that incorporate structured representations, as discussed in the following subsection.

3.2.2 Voxel-Based Methods

Voxel-based methods transform irregular point clouds into structured volumetric grids, allowing the application of standard CNN operations. These approaches benefit from well-established 3D convolution techniques but introduce trade-offs related to memory usage, spatial resolution, and quantization artifacts.

VoxNet [43] was one of the first models to discretize 3D space into uniform voxel grids and apply 3D CNNs for object recognition. While this method demonstrated competitive performance in controlled environments, the fixed grid resolution resulted in increased in memory consumption and introduced discretization errors, particularly for small or thin structures.

To improve memory efficiency, OctNet [56] adopted a sparse voxel representation based on octrees. This hierarchical structure reduces the number of occupied voxels by focusing on regions with data, enabling deeper networks under limited resource constraints. However, octree traversal introduces computational overhead and complicates parallelization, which can limit real-time applicability.

VoxelNet [87] proposed an end-to-end architecture that combines voxelization with 3D feature learning and region proposal generation. By learning voxel-wise features through Voxel Feature Encoding (VFE) layers, it achieves better spatial representation. However, voxel size selection influences performance: smaller voxels improve resolution but increase computational cost, while larger voxels reduce detail and may hinder small object detection.

CenterPoint [80] reformulates 3D object detection as a center-based task, predicting object centers on a BEV heatmap and regressing attributes such as size, orientation, and velocity. This anchor-free design improves rotational invariance and avoids manually tuned anchor settings. Built on voxelized backbones such as VoxelNet or PointPillars, CenterPoint also extends naturally to tracking by associating objects across frames using predicted velocities, offering a simple yet effective alternative to motion models. While it delivers high detection and tracking accuracy with real-time efficiency, its reliance on accurate BEV heatmap centers can degrade performance on very small or heavily occluded objects, and the approach remains sensitive to point cloud sparsity at long ranges.

More recent approaches explore hybrid representations to enhance context aggregation. PVT-SSD [78] integrates voxel and point features through a transformer-based Point-Voxel module, combining voxel-level geometry with fine-grained point-level information. As part of its pipeline, sparse voxel features are transformed into dense bird's-eye view (BEV) feature maps ("To Dense" step), enabling standard 2D convolution operations and facilitating spatial alignment for query initialization. This dual representation improves detection accuracy, particularly in sparse or complex environments. However, the additional transformer components and dense feature processing substantially increase computational and memory demands, limiting deployment on embedded systems (Figure 3.3).

VoxSeT [21] introduced voxel-based set attention modules that support parallel computation with linear complexity. By using voxel sets as input tokens, it balances efficiency

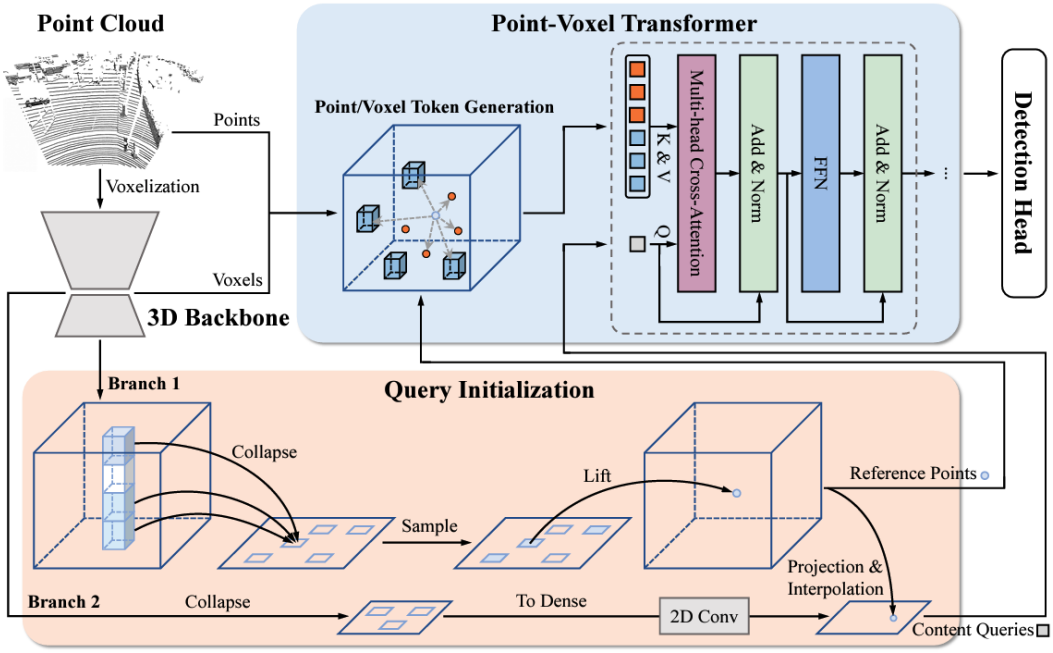

Figure 3.3: PVT-SSD architecture. Raw point clouds are voxelized for sparse convolution input. The module identifies reference points by processing non-empty voxels and extracts queries from the dense BEV feature map. These queries undergo adaptive fusion of voxel and point features via the Point-Voxel Transformer. Finally, the detection head performs classification and regression using these fused features. This image is reprinted from [78].

and global context modeling. However, its performance may degrade in highly dynamic scenes due to limited temporal modeling and potential over-smoothing of sparse inputs.

Octr [86] proposed a hierarchical transformer architecture based on octrees, incorporating coarse-to-fine attention across spatial levels. Its OctAttn mechanism dynamically constructs sparse attention patterns over octants, while hybrid positional embeddings encode semantic and geometric cues. Although Octr achieved strong results in detecting distant and partially visible objects, the dynamic octree construction introduces complexity and latency that pose challenges for real-time applications.

In summary, voxel-based methods offer a compromise between raw point fidelity and convolutional efficiency. Their performance depends on voxel resolution, memory optimization strategies, and the integration of contextual information — that is, combining cues from neighboring voxels or across multiple scales to capture object shape and scene layout more effectively.

3.2.3 Projection-Based Methods

Projection-based methods transform unstructured 3D point clouds into 2D grid representations, enabling the use of efficient 2D convolutional neural networks (CNNs) developed for image processing. These approaches offer improved computational efficiency and memory utilization compared to voxel-based and point-based methods. However, projection often results in the loss of fine-grained geometric detail, especially in the vertical dimension, and can introduce distortion depending on the projection strategy.

The most common projection paradigms are the Bird's Eye View (BEV) and the Frontal View. Each offers distinct trade-offs in terms of spatial resolution, semantic richness, and suitability for different object scales and orientations.

3.2.3.1 Bird's Eye View (BEV) Projection

BEV projection compresses the vertical axis to generate a top-down 2D map of the scene. This representation preserves horizontal spatial relationships and is well-suited for navigation tasks and road scene understanding.

Vote3Deep [13] applies sparse convolutional voting on BEV maps to efficiently encode spatial features. This approach achieves real-time inference and performs well in structured urban scenes. However, its reliance on manually designed voting functions limits adaptability to diverse object shapes and densities.

AVOD (Aggregate View Object Detection) [9] combines BEV and image features in a two-stage pipeline. BEV-based proposals are refined using semantic cues from image data, improving object classification. While AVOD demonstrates competitive performance, it is sensitive to sensor calibration errors and incurs additional latency because its multi-stream architecture processes BEV and image features through separate pipelines before fusion, requiring extra computation and synchronization. PIXOR [77] presents a single-stage detection framework that directly regresses oriented bounding boxes from BEV feature maps. By treating BEV as a 2D image, it leverages standard CNN backbones for efficient processing. Nonetheless, the vertical compression inherent in BEV reduces the model's ability to resolve object height and distinguish between overlapping vertical structures, particularly in cluttered scenes.

3.2.3.2 Frontal View Projection

Frontal or range view projection maps LiDAR points into a cylindrical or spherical 2D grid based on angular resolution and radial distance, aligning with the sensor's acquisition geometry. This view retains depth continuity and angular information but can introduce spatial distortion, especially at oblique angles or close distances.

RSN (Range Sparse Net) [61] employs a two-stage architecture that performs semantic segmentation on range images followed by sparse voxel refinement. This hybrid design

enhances long-range detection performance but relies heavily on accurate foreground-background segmentation, and the voxelization step may reintroduce discretization artifacts.

To the Point [7] uses range image projections in conjunction with graph convolutional operations to recover local geometric context. This method supports cross-modal fusion with camera data. However, the use of graph-based kernels increases computational complexity, and the projection process distorts spatial relationships in occluded regions.

RangeDet [15] addresses geometric distortion through several innovations: range-conditioned feature pyramids, coordinate-adaptive meta-kernel convolutions, and weighted non-maximum suppression. These components improve robustness to depth variation and object scale. Nevertheless, performance remains sensitive to angular resolution and requires careful tuning of projection parameters.

RangeIoUDet [36] introduces point-based Intersection-over-Union (IoU) supervision and a hybrid Generalized IoU (GIoU) loss to better align detection predictions with ground-truth objects. Unlike standard IoU, which only evaluates overlap, GIoU also penalizes the distance between non-overlapping boxes, providing a more informative gradient for bounding box regression. This design improves localization precision, though it increases training complexity and scales poorly with very high-resolution inputs.

In summary, projection-based methods provide a computationally efficient pathway to object detection by leveraging 2D convolutional architectures. However, they often involve trade-offs in geometric accuracy, especially under occlusion or when operating on highly non-uniform point clouds. Hybrid approaches that combine projection with point-wise refinement have shown promise but come with increased architectural and training complexity.

3.3 RADAR-Based Object Detection

Automotive RADAR sensors are increasingly used in autonomous driving due to their ability to operate reliably in adverse weather and low-visibility conditions, such as fog, rain, and darkness. Radar sensors emit radio frequency (RF) signals, typically in the millimeter-wave band (24–77 GHz), and measure their reflections to estimate object range and relative velocity using the Doppler effect. Compared to LiDAR, RADAR offers longer detection ranges and is more robust to environmental interference. However, RADAR signals are inherently sparse and noisy, and the data exhibits low angular resolution and high variability in reflectivity (RADAR Cross Section, or RCS) depending on object material, shape, and orientation.

These characteristics pose several challenges for high-precision object detection, especially for shape estimation and classification. Recent research has explored RADAR-based detection frameworks, either using RADAR alone or in conjunction with other modalities. This section reviews key RADAR-only methods and outlines the rationale for excluding RADAR from the experimental design of this thesis.

RODNet [71] processes time-series micro-Doppler spectrograms using a multi-branch encoder-decoder architecture based on U-Net. It employs temporal and spatial attention to extract fine-grained motion features for distinguishing object classes. While RODNet performs well in near-field detection tasks (up to 15 meters), it depends on high-resolution range-Doppler maps that are typically unavailable in commercial automotive RADAR systems, limiting its practical applicability in highway or long-range scenarios.

RadarNet [42] introduces a spatiotemporal convolutional model that directly consumes raw RADAR cube inputs. It encodes motion and shape cues to support object tracking and classification. Although the network demonstrates effective detection of moving objects, it underperforms on static objects due to weak reflections and is prone to false positives caused by clutter or multi-path effects. Additionally, the method requires

RADAR hardware capable of consistent RADAR cube formatting, which varies across manufacturers.

RadarPillars [45] adapts the PointPillars [30] architecture to RADAR data by replacing LiDAR inputs with RADAR detections consisting of range, azimuth, Doppler, and RCS. The detections are projected into BEV and converted into pseudo-images for processing via 2D convolutions. While this design benefits from RADAR's inherent motion information, it exhibits reduced performance in crowded or static scenes due to the low spatial resolution and limited elevation field of view inherent in RADAR sensors.

Despite these developments, RADAR-only detection models continue to face notable limitations:

- **Sparse and Noisy Data:** RADAR measurements are often degraded by environmental noise and clutter, leading to reduced object localization precision.
- **Low Angular Resolution:** The wide beamwidth of RADAR sensors hinders the discrimination of closely spaced objects, particularly in urban settings.
- **Inconsistent Reflectivity (RCS):** RADAR returns vary based on material composition, surface roughness, and aspect angle, resulting in detection inconsistencies.
- **Limited Dataset Availability:** Public datasets that provide synchronized RADAR, camera, and LiDAR measurements with high-quality 3D annotations are scarce. For example, nuScenes [5] includes RADAR data, but with limited labeled samples and range coverage.

Given these challenges, this thesis does not incorporate RADAR as a primary sensing modality. The focus remains on LiDAR-camera fusion, which offers higher geometric accuracy and semantic richness. Nonetheless, RADAR remains a promising complementary modality for future work, particularly in scenarios involving sensor redundancy or degraded environmental visibility.

3.4 Fusion-Based Object Detection

The limitations of single-sensor systems in object detection have motivated the use of multi-modal fusion techniques in autonomous perception. While cameras provide high-resolution semantic information, they lack reliable depth estimation. LiDAR offers accurate 3D geometry but is sparse and lacks appearance cues. RADAR is robust under adverse weather but provides low spatial resolution. Each modality alone is insufficient for consistently reliable object detection across diverse environmental and operational conditions.

Sensor fusion aims to exploit the complementary characteristics of different modalities by combining them at various stages in the perception pipeline. Depending on the fusion point, methods are typically categorized into:

- **Early fusion**, where raw data or low-level features are combined before further processing;
- **Mid-level fusion**, where intermediate features from each modality are integrated;
- **Late fusion**, where high-level decisions or predictions from each modality are merged;
- **Transformer-based fusion**, where attention mechanisms dynamically model relationships across modalities. Unlike early fusion, which concatenates raw data, or mid/late fusion, which merges fixed feature representations, transformer-based fusion enables cross-attention between modalities. This allows the model to selectively focus on the most relevant regions (e.g., aligning LiDAR points with corresponding image features) and adaptively integrate information, rather than relying on predefined fusion rules.

This section reviews representative models within each of these categories, focusing on their fusion strategies, architectural design, and performance characteristics. Particular attention is given to alignment accuracy, robustness under sensor degradation (e.g.,

reduced LiDAR point density at long ranges or image noise in low light), computational efficiency, and reliance on calibration or synchronization. These factors are critical for evaluating the suitability of each method for deployment in real-time autonomous driving systems.

3.4.1 Early Fusion Techniques

Early fusion techniques integrate sensor data at the raw data level or at low-level feature stages before significant independent processing occurs. This approach enables the network to learn joint representations from the outset, potentially allowing better exploitation of complementary cues between modalities. However, early fusion methods typically require precise calibration between sensors and are sensitive to noise or degradation in any single modality.

EPNet [25] proposes a LiDAR-guided image fusion framework that aligns 2D image features with 3D point cloud coordinates in a point-wise fashion. The model introduces a LI-Fusion module that combines semantic features from camera images with the geometric structure of LiDAR data. Additionally, it incorporates a Consistency Enforcing (CE) loss, which encourages consistency between object classification scores and bounding box regression accuracy during training. This improves robustness against ambiguous detections and spatial misalignment.

EPNet's point-wise fusion mechanism is dependent on accurate extrinsic calibration. Even small misalignments can lead to incorrect feature associations, degrading performance. Furthermore, the use of dense image features increases memory usage and computational cost, limiting scalability to real-time applications. The CE loss also adds complexity to the training process and may require careful parameter tuning to generalize across datasets or sensor setups. In low-light or poor weather conditions, the reliance on RGB image features may further degrade performance.

PointAugmenting [69] adopts a different early fusion strategy by augmenting each

raw LiDAR point with point-wise semantic features extracted from RGB images (as shown in Figure 3.4). The method first projects the 3D point cloud onto the 2D image plane and retrieves the corresponding image features using a pretrained CNN backbone. These features are then concatenated with the original LiDAR points to form enriched point-level inputs. Fusion is applied prior to BEV encoding via skip connections and concatenation modules within the backbone network.

To improve robustness, the model applies cross-modal data augmentation that synchronizes geometric transformations (e.g., rotation, flipping) across image and LiDAR domains. This technique simulates diverse conditions and enhances generalization. Experimental results on the nuScenes benchmark show a significant improvement over LiDAR-only baselines, with a reported increase of 6.5% in mean Average Precision (mAP).

Despite its effectiveness, PointAugmenting also has several limitations. The projection from 3D points to the 2D image plane requires highly accurate sensor calibration; otherwise, semantic features may be incorrectly assigned. The model also assumes that the camera features are of sufficient quality, which may not hold under poor lighting or inclement weather. Additionally, the concatenation of image-derived features with LiDAR features increases the dimensionality of the input, which in practice leads to higher memory usage and computational cost during training and inference. The cross-modal augmentation pipeline introduces additional training complexity and requires careful design to ensure consistency across modalities.

MVX-Net [60] extends VoxelNet to multimodal detection through two strategies for early fusion of RGB and LiDAR. In PointFusion, image features from a pre-trained detector are concatenated to corresponding LiDAR points before voxelization, while in VoxelFusion, pooled image features are appended to voxel embeddings. These designs enhance semantic understanding and improve classification accuracy compared to LiDAR-only detectors, particularly for small objects. However, MVX-Net is heavily dependent on precise camera–LiDAR calibration, and PointFusion in particular introduces

computational overhead during point-to-image projection. Furthermore, its reliance on pre-trained 2D features reduces flexibility, and performance degrades in poor lighting or adverse weather when RGB inputs are less reliable.

Overall, early fusion approaches benefit from joint feature representation and improved semantic-geometric alignment, but their performance is often constrained by calibration sensitivity, high memory usage, and reliance on consistent image quality. These challenges have motivated the development of alternative strategies such as mid-level and attention-based fusion, which attempt to balance early interaction with greater flexibility and robustness.

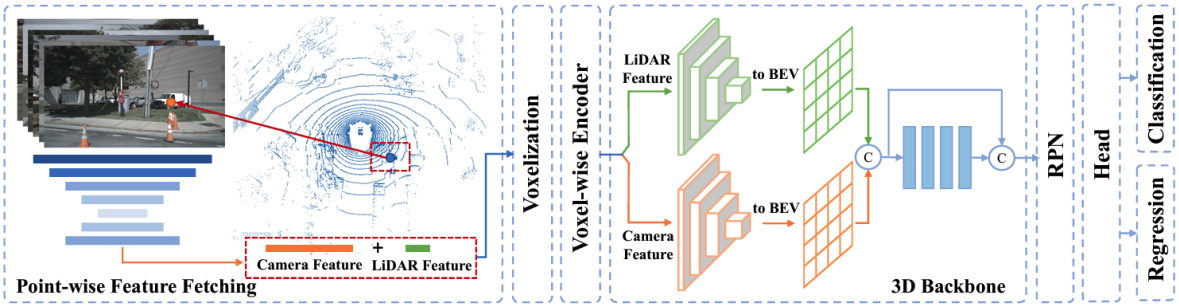

Figure 3.4: Overview of Point Augmenting: The framework is structured in two phases. Firstly, point-wise feature retrieval involves projecting LiDAR points onto the image plane, followed by the augmentation with point-wise CNN features. Secondly, for 3D detection, the CenterPoint model is enhanced by integrating an additional 3D sparse convolution stream for camera features, with modalities merged using a straightforward skip and concatenation method in BEV maps. This image is reprinted from [69].

3.4.2 Mid-Level Fusion Methods

Mid-level fusion combines intermediate features extracted separately from each modality, aiming to preserve modality-specific strengths while enabling joint reasoning across spatial and semantic domains. Compared to early fusion, mid-level methods are generally more flexible, as they allow independent preprocessing pipelines and enable the network

to align higher-level features adaptively. However, these approaches must still address modality alignment, synchronization, and computational complexity, particularly when feature representations are heterogeneous.

CAT-Det [83] adopts a dual-stream architecture in which LiDAR and image features are extracted independently through two modality-specific backbones. The authors design a Pointformer [83] branch tailored for LiDAR, which captures both local geometric details and global context from point clouds, and an Imageformer [83] branch for camera inputs, which models semantic and spatial relationships in images. These high-level features are fused using a Cross-Modal Transformer (CMT) [83], which applies cross-attention mechanisms to model fine-grained inter-modal correlations. The CMT dynamically selects and combines complementary features across modalities, improving semantic and spatial alignment for 3D object detection.

To improve robustness during training, CAT-Det introduces a One-way Multi-modal Data Augmentation (OMDA) strategy, which applies geometric and semantic transformations only to the LiDAR modality. This encourages the camera features to adapt implicitly to perturbed LiDAR features, enhancing alignment without requiring direct supervision between modalities.

CAT-Det shows improved performance in occluded or cluttered scenarios on the KITTI dataset. However, the asymmetric augmentation strategy limits the exposure of image features to real-world distortions such as motion blur, low illumination, or weather-related degradation. Additionally, the use of separate backbones and the transformer-based fusion module increases model complexity and training time, potentially hindering deployment in latency-sensitive applications.

DeepFusion [33] presents a mid-level fusion framework designed to address feature alignment issues caused by geometric data augmentations and domain mismatch. Unlike earlier methods that fuse raw or shallow features (low-level edges, textures, or geometric cues), DeepFusion focuses on aligning deep features (high-level semantic representations extracted in later network layers) through two key innovations: InverseAug and Learn-

ableAlign.

InverseAug explicitly reverses spatial augmentations (e.g., rotation, flipping, scaling) applied during training by applying the inverse transformations to the image features before fusion. This ensures that camera and LiDAR features remain geometrically consistent. LearnableAlign uses a cross-attention module to learn dynamic correspondences between the two feature spaces during training. This module models the spatial relationship between LiDAR points and image pixels without relying on hard geometric constraints. The fusion process is illustrated in Figure 3.5, emphasizing its distinction from early fusion techniques that directly concatenate raw point coordinates with image features.

DeepFusion demonstrates improved performance on the Waymo Open Dataset, particularly in scenarios involving long-range detection and sensor perturbations. The architecture generalizes well to out-of-distribution data and maintains accuracy in low-visibility conditions. However, several limitations remain. The method performs less reliably on dynamic objects due to motion inconsistencies between sensors. Furthermore, DeepFusion lacks explicit temporal modeling, which affects detection stability across frames. The combination of deep fusion, inverse augmentation, and attention mechanisms increases model size and inference time, making real-time deployment more difficult.

In summary, mid-level fusion strategies offer a compromise between early integration and late decision-level fusion. They allow for more informed cross-modal reasoning but introduce challenges related to alignment accuracy, increased architectural complexity, and scalability to time-sensitive or resource-constrained environments.

3.4.3 Late Fusion Techniques

Late fusion methods integrate modality-specific predictions or high-level features after separate processing pipelines have been applied to each sensor. This approach minimizes

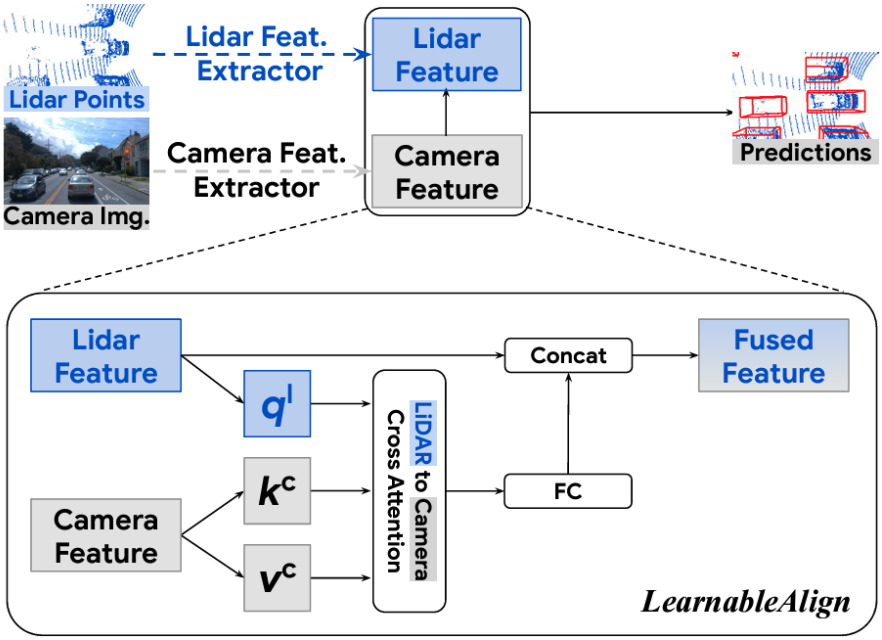

Figure 3.5: DeepFusion merges two modalities at the deep feature level, unlike earlier methods such as PointAugmenting [69] that apply camera features to LiDAR points at the input stage. To address deep feature fusion's modality alignment challenges, it introduces InverseAug and LearnableAlign—a cross-attention-based module where q, k, and v denote the standard query, key, and value matrices used in attention, and FC refers to a fully connected layer. This image is reprinted from [33].

cross-modal interference during feature extraction and simplifies the fusion process, making it relatively robust to sensor noise or degradation. However, the separation of streams may limit the ability to model fine-grained cross-modal interactions and can introduce misalignment between detection outputs, particularly in spatially complex environments.

3D-CVF [81] implements a late fusion strategy that projects image features into the Bird's Eye View (BEV) domain using learned geometric transformation parameters. These transformed features are fused with LiDAR-derived BEV features through an adaptive gated module that emphasizes the most informative modality depending on local context. The gating mechanism allows the model to adjust its reliance on each modality dynamically, improving robustness in scenarios where one sensor is unreliable or occluded.

While 3D-CVF demonstrates improved detection performance on long-range and partially occluded objects, it depends on accurate sensor calibration to perform spatial alignment between modalities. Errors in calibration or slight temporal desynchronization can lead to feature misalignment, reducing the effectiveness of the fusion. Additionally, the dual-path structure increases computational and memory requirements.

Xie et al. [73] propose a dual-stream detection framework in which LiDAR and camera inputs are processed independently through modality-specific CNN backbones. Final detection results are obtained by concatenating features at the detection head stage. This architecture reduces early-stage interference between modalities and preserves the unique characteristics of each feature stream.

Despite its simplicity, this design faces several limitations. First, spatial misalignment between sensor streams is not corrected during the fusion process, potentially degrading detection accuracy. Second, using separate backbones increases the number of parameters and inference time, making real-time deployment more challenging. Third, the lack of explicit interaction between modalities during feature extraction can limit the model's ability to exploit complementary information, especially in difficult scenarios involving occlusion.

Overall, late fusion methods offer implementation simplicity and resilience to modality-specific noise but often underutilize inter-modal correlations. Their performance depends heavily on reliable geometric calibration and may be suboptimal in cluttered or dynamic environments where early or mid-level interaction between modalities is advantageous.

3.4.4 Transformer-Based Fusion Methods

Transformer-based fusion methods leverage self-attention and cross-attention mechanisms to dynamically align and integrate features from different sensor modalities. Unlike traditional fusion techniques that rely on fixed spatial correspondences or early-stage concatenation, transformers allow flexible, content-aware interactions between modalities, meaning that features from one modality are selectively combined with the most relevant features from another based on their information content. These architectures have shown strong performance across a range of benchmarks but introduce computational complexity.

TransFusion [1] proposes a two-stage transformer-based detection pipeline with separate backbones for LiDAR and camera features. The first decoder layer generates coarse 3D bounding box proposals using LiDAR features and learned object queries. In the second layer, these proposals are refined by attending to corresponding image features via Spatially Modulated Cross-Attention (SMCA), which modulates attention weights based on the predicted 3D locations from LiDAR. Additionally, an image-guided query initialization mechanism is used to seed object queries based on camera features, improving recall for small or visually ambiguous objects (Figure 3.6).

TransFusion addresses several limitations of point-level early fusion by adopting a soft-association strategy, avoiding rigid pixel-to-point correspondences. This improves robustness in scenarios where sensor misalignment or occlusion degrades precise matching. However, the model's dual-decoder structure and attention modules increase latency and memory consumption. Moreover, its performance depends on the quality of initial

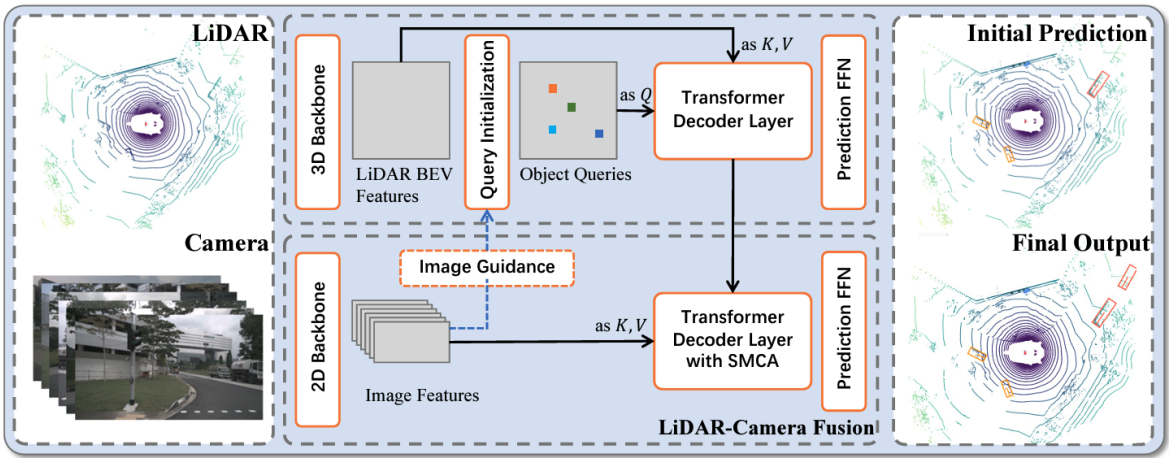

Figure 3.6: The TransFusion model uses 3D and 2D backbones to extract LiDAR and image features, with a two-layer transformer decoder detection head. The first layer creates initial 3D bounding boxes from object queries, while the second fuses these with image features for enhanced detection. It incorporates a Spatially Modulated Cross-Attention mechanism for better image focus and an image-guided query initialization to improve object detection in sparse LiDAR data. Here, Q, K, and V denote query, key, and value embeddings used in the attention mechanism, while FFN refers to a feed-forward network. This image is reprinted from [1].

LiDAR-based proposals; inaccurate initial detections may reduce the effectiveness of subsequent image-guided refinement.

M3DETR [19] proposes a comprehensive transformer-based architecture for 3D object detection using LiDAR point clouds. The framework centers around a unified mechanism to capture diverse spatial structures by simultaneously modeling multi-representation, multi-scale, and mutual-relation features of the input data, as illustrated in Figure 3.7. This design allows M3DETR to reason over voxel, point-wise, and Bird's EyeView (BEV) embeddings concurrently, thereby capturing both fine-grained geometric cues and high-level spatial context.

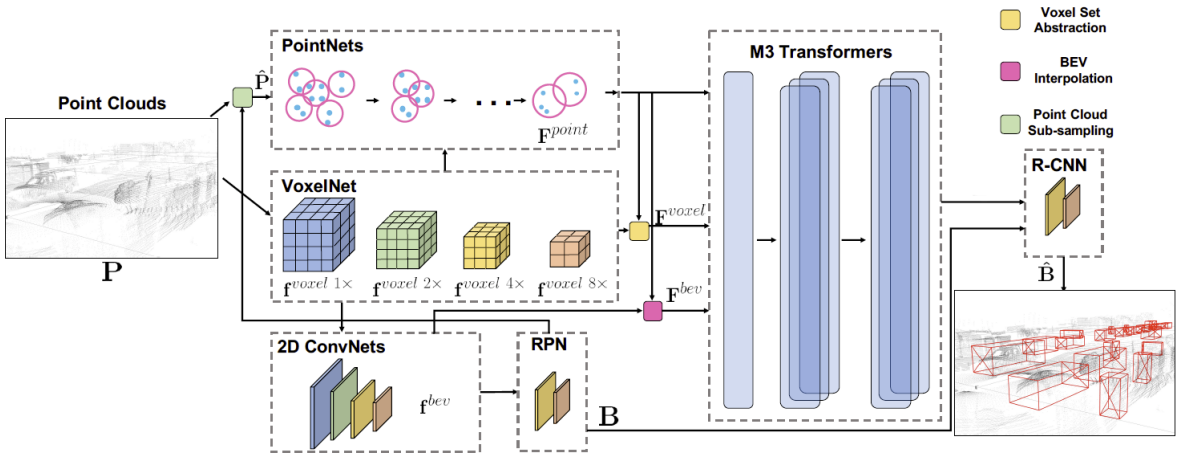

Figure 3.7: M3DETR framework employs a transformer-based approach for object detection, using a coarse-to-fine method. It integrates PointNets, VoxelNet, and 2D ConvNets for diverse feature learning. M3 Transformers focus on multi-representation and multi-scale features, while the Region Proposal Network (RPN) generates initial proposals. The R-CNN module refines these features from M3 transformer outputs, enhancing detection outcomes. This image is reprinted from [19].

The architecture first encodes raw point clouds into three distinct representations—voxel grids, point-based features, and BEV projections—each emphasizing different structural priors. These embeddings are then processed by the core M3 Transformer, which attends across these modalities to learn cross-representation, cross-scale, and cross-point

interactions. This multi-perspective fusion enhances the model's ability to interpret complex 3D scenes, especially in cluttered urban environments with overlapping objects or varying scales.

For detection, M3DETR adopts a two-stage pipeline comprising a Region Proposal Network (RPN) followed by an R-CNN-style refinement module. These modules leverage the rich fused features from the M3 Transformer to generate and refine accurate 3D bounding boxes, improving localization precision and class discrimination.

Prior 3D detectors are constrained by two primary challenges: the three common point-cloud representations (voxels, raw points, and BEV) each demand distinct neural-network architectures and costly format conversions—creating semantic gaps that make effective fusion non-trivial—and their multi-scale feature pyramids rely on bilinear down-/up-sampling plus simple concatenation, which cannot simultaneously preserve high resolution and broad contextual receptive fields, yielding suboptimal detection accuracy.

M3DETR addresses these issues with its M3 Transformer blocks but introduces its own limitations: its multi-branch feature extraction and cross-modal attention mechanisms substantially increase computational and memory overhead—hindering real-time deployment on embedded or low-power systems; aligning disparate representations (e.g., voxels and points) requires precise calibration, where even small misalignments can propagate through the transformer and degrade performance; and although it achieves strong results on KITTI and Waymo benchmarks, its robustness under adverse weather or sensor degradation remains untested, leaving real-world reliability an open question. BEVFusion [35] addresses the limitations of query-based fusion methods such as TransFusion by disentangling LiDAR and camera streams and projecting both into the same BEV space, where they are fused with a lightweight attention module. This design avoids dependence on LiDAR queries for extracting image features and improves robustness under sensor degradation, as each modality can operate independently. On nuScenes, BEVFusion achieves state-of-the-art accuracy while maintaining efficiency, and it demonstrates resilience under simulated LiDAR or camera failures. Nonetheless, BEVFusion

still inherits challenges from BEV-based designs, including reliance on accurate depth estimation for camera features, limited vertical resolution, and performance sensitivity to imperfect calibration between modalities. FUTR3D [10] proposes a unified transformer-based framework that flexibly supports diverse sensor configurations, including cameras, LiDAR, and radar. A Modality-Agnostic Feature Sampler (MAFS) allows object queries to interact with heterogeneous sensor features in a shared space without explicit view transformations. This design makes FUTR3D inherently scalable to different modality combinations, ranging from dense camera–LiDAR setups to sparse or cost-sensitive sensor configurations. On benchmarks such as nuScenes, FUTR3D achieves competitive performance across multiple settings. However, the approach introduces additional computational overhead due to multi-modal attention layers and remains dependent on precise temporal synchronization, while still trailing LiDAR-only methods in fine-grained geometric accuracy.

Cross-Modal Transformer (CMT) [76] introduces a fully end-to-end transformer framework for LiDAR–camera fusion that directly aligns image and point cloud features through position-guided encoding, avoiding explicit view transformations. A Coordinates Encoding Module (CEM) injects 3D positional information into both modalities, enabling object queries to interact consistently with image and LiDAR tokens for unified prediction. Robustness is further improved via masked-modal training, where one modality is intermittently dropped, allowing the model to function under sensor failure. While CMT achieves strong accuracy and robustness on benchmarks such as nuScenes, it comes with high memory and compute demands due to cross-modal attention and still exhibits reduced precision in long-range depth estimation compared to LiDAR-only models.

BEVFusion [41] proposes a unified multi-modal fusion framework that transforms both LiDAR and image data into a shared BEV representation for efficient and scalable 3D object detection. Unlike traditional methods that fuse features in their native domains (e.g., 2D for images, 3D for LiDAR), BEVFusion leverages the geometric consistency of the BEV space to enable tight spatial alignment and homogeneous processing

of heterogeneous data.

The architecture begins by extracting high-level features from both camera images and LiDAR point clouds using dedicated 2D and 3D backbones. Camera features undergo view transformation via a learned depth-guided warping mechanism, which lifts 2D features into 3D space before projecting them into BEV. Similarly, LiDAR features are voxelized and encoded into BEV using sparse convolutional encoders. These two sets of BEV features are then fused using a simple but effective additive fusion scheme, which preserves modality complementarity while allowing efficient computation.

A critical innovation in BEVFusion is the adoption of a BEV-centric design, where the entire perception pipeline—including feature extraction, fusion, and detection—is unified under the BEV space. This ensures consistent spatial reasoning, seamless integration of different modalities, and compatibility with map-based priors or planning systems. The model is illustrated in Figure 3.8, highlighting the transformation and fusion process across modalities.

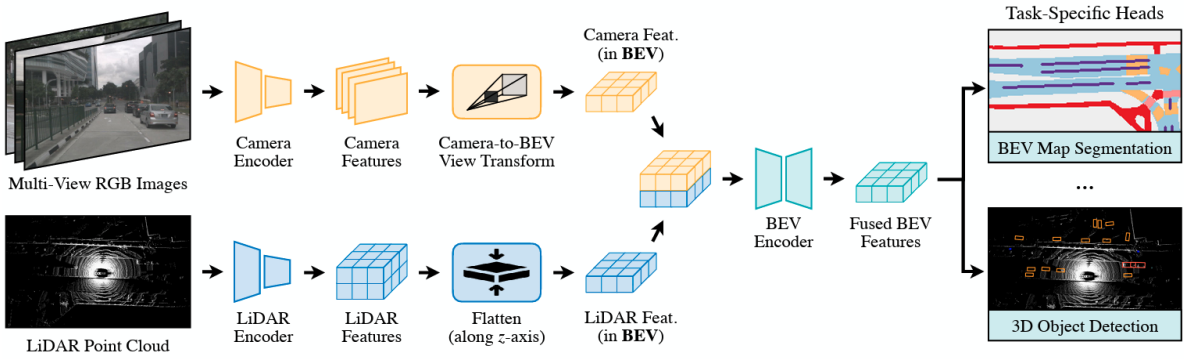

Figure 3.8: BEVFusion transforms both LiDAR and image features into the BEV space using sparse voxel encoders and a depth-guided image warping module. The features are fused via element-wise addition and processed through a unified BEV encoder for final detection. This architecture eliminates the need for modality-specific detection heads. Image reprinted from [41].

BEVFusion achieved state-of-the-art performance on the nuScenes benchmark, demonstrating strong improvements in both detection accuracy and runtime efficiency over

previous fusion frameworks. The unified BEV representation allowed for seamless integration of additional data sources such as RADAR or maps, showcasing its extensibility for multi-modal perception tasks in autonomous driving.

BEVDet [24] establishes a high-performance paradigm for multi-camera 3D object detection by explicitly projecting image features into the Bird's-eye-view (BEV) space, where planning and localization tasks are naturally defined. The framework uses an image-view encoder, a view transformer to lift features into BEV, a BEV encoder, and a detection head adapted from CenterPoint. This modular design improves geometric reasoning compared to image-view detectors. Representative augmentations in both image and BEV space further strengthen generalization. BEVDet demonstrates that conducting detection directly in BEV space leads to higher accuracy and efficiency than view-based methods. However, it still suffers from limited depth perception, sensitivity to calibration errors between cameras, and degraded performance on small or distant objects due to the lack of LiDAR depth cues.

BEVFormer [34] builds upon the BEVDet paradigm by incorporating spatiotemporal transformers to aggregate information across frames and enhance global reasoning. Camera features are lifted into BEV space using deformable attention, where object queries interact with both spatial and temporal contexts. This improves long-range perception and stability, enabling better detection of dynamic objects compared to BEVDet. Despite these advances, BEVFormer introduces heavy computational cost due to multi-frame attention and relies strongly on accurate ego-motion for temporal alignment. It also struggles with fine-grained localization under sparse supervision and remains more resource-intensive than LiDAR-based counterparts, limiting real-time deployment.

Multimodal sensor fusion boosts 3D object-detection accuracy but faces three practical limits. First, it depends on highly precise extrinsic and intrinsic calibration between sensors; small shifts caused by vibration or temperature changes can lower accuracy. Second, the fusion networks—especially those with attention layers—require extra computation and memory, slowing inference on embedded hardware. Third, LiDAR, RADAR,

and cameras capture data at different frame rates and with different delays, making real-time time-alignment difficult and allowing residual timing errors to pass through the detection pipeline. Further work on these issues is necessary, particularly for urban driving scenes where single-sensor systems degrade under occlusion or poor lighting.

3.5 Research Gaps and Thesis Positioning

Multimodal 3D object-detection pipelines face several constraints that limit deployment in production vehicles. Current fusion backbones—especially those based on global attention—incur memory and latency costs that exceed the capabilities of automotive-grade hardware. In addition, several key limitations remain:

- **Calibration sensitivity:** Modest inaccuracies in extrinsic calibration and sub-frame sampling offsets between LiDAR, camera, and RADAR introduce projection errors that propagate through the network and reduce detection accuracy.
- **Environmental vulnerability:** Under adverse weather or partial sensor occlusion, one modality can become unreliable, yet most existing methods fuse sensor outputs with fixed weights or heuristic confidence scores.
- **Data limitations:** While large annotated sets exist for camera and LiDAR, RADAR labels remain scarce, hindering supervised learning and cross-domain transfer.
- **Uncertainty and interpretability:** Current deep fusion models usually act as "black boxes." They provide predictions such as bounding boxes but rarely indicate how confident these predictions are or explain why certain features (e.g., LiDAR vs. camera) were prioritized. Without calibrated uncertainty estimates or interpretable intermediate outputs, it becomes difficult to verify and validate these models in safety-critical applications like autonomous driving.

These gaps have motivated a variety of recent approaches, but important limitations remain. Recent studies employing BEV projection networks and transformer-based cross attention have partially addressed spatial alignment and context aggregation. However, these designs generally assume static calibration, operate on individual frames, and lack mechanisms for dynamic modality re-weighting, leaving them sensitive to calibration drift, sensor degradation, and domain shift across weather conditions or sensor configurations.

This thesis addresses these limitations through a resource-aware transformer framework. The proposed model retains separate LiDAR and RGB encoders, employs multi-scale mutual cross-attention to learn spatial correspondences, and integrates a reliability-aware contrastive objective to down-weight degraded channels. Temporal consistency is enforced via long-range attention across successive frames, reducing prediction variance under occlusion and asynchronous sampling. The framework is trained end-to-end on the nuScenes benchmark and evaluated under synthetic calibration offsets, and simulated sensor dropouts to assess efficiency, calibration tolerance, and domain generalisation relative to established baselines. Full architectural and implementation details are provided in Chapter 6.

Chapter 4

Datasets

The choice of datasets is critical for advancing research in 3D object detection for autonomous driving. Datasets determine not only the scale of training but also the diversity of conditions under which models are evaluated. In this study, we focus on two benchmarks that have become standards in the field: KITTI [16] and nuScenes [5]. Both datasets provide multimodal sensor inputs, high-quality annotations, and task benchmarks essential for evaluating transformer-based object detection methods. Unlike general-purpose 2D datasets such as PASCAL VOC [14] or COCO [38], these two datasets are explicitly designed for autonomous driving scenarios, offering 3D spatial information, temporal consistency, and metadata that are indispensable for sensor fusion frameworks.

4.1 KITTI Dataset

The KITTI dataset [16], collected in Karlsruhe, Germany, is one of the most widely used benchmarks for autonomous driving research. It provides synchronized multimodal data from a vehicle equipped with:

- **Cameras:** Two high-resolution RGB cameras mounted on the roof with a stereo baseline of 54 cm. Stereo refers to capturing the same scene simultaneously from two horizontally displaced viewpoints, enabling the recovery of depth information from images.
- **LiDAR:** A Velodyne HDL-64E rotating laser scanner with 64 beams, operating at approximately 10 Hz. Each LiDAR sweep generates about 100,000 3D points, covering a full 360° horizontal field of view.
- **GPS/IMU:** Global Positioning System (GPS) and Inertial Measurement Unit (IMU) sensors providing precise localization and motion data, including position, velocity, and acceleration of the ego vehicle.

Dataset Size and Structure

The KITTI dataset consists of:

- **Object Detection Benchmark:** 7,481 frames for training/validation and 7,518 frames for testing.
- **Annotations:** More than 80,000 objects are annotated across three benchmark classes: Car, Pedestrian, and Cyclist.
- **Tasks Provided:** KITTI provides benchmarks for multiple perception tasks including 2D object detection, 3D object detection, BEV detection, multi-object tracking, depth estimation (stereo and monocular), and semantic/instance segmentation.
- **Image Resolution:** Camera images are 1242×375 pixels on average.
- **LiDAR Resolution:** Each sweep generates dense 3D point clouds with a vertical field of view of $-24.9°$ to $+2°$ and a range of up to 120 meters.

Difficulty Levels

KITTI annotations are divided into three levels of difficulty based on object size, occlusion, and truncation:

- **Easy:** Objects larger than 40 pixels in height, less than 15% occluded, and less than 15% truncated.
- **Moderate:** Objects larger than 25 pixels, less than 30% occluded, and less than 30% truncated.
- **Hard:** Objects larger than 25 pixels, but with occlusion up to 50% and truncation up to 50%.

These categories allow researchers to evaluate not only overall detection accuracy but also robustness under challenging visual conditions.

Advantages and Limitations

- **Advantages:** KITTI provides high-quality annotations, precise calibration, and multimodal sensor data. It is compact yet sufficiently diverse across urban, rural, and highway environments. Its stereo imagery supports depth-based reasoning, while its 64-beam LiDAR provides dense 3D perception.
- **Limitations:** KITTI is relatively small compared to modern datasets. It lacks environmental variations such as night-time or adverse weather, and its object categories are limited. This restricts its ability to benchmark robustness in complex real-world conditions.

KITTI is selected in this research for evaluating TransfuseNet, as it provides a well-established benchmark for multimodal 2D and 3D detection, ensuring comparability with a large body of existing literature.

4.2 nuScenes Dataset

The nuScenes dataset [5], released by Motional (formerly Aptiv), was designed to address limitations of earlier datasets like KITTI. It provides greater scale, multimodal sensor coverage, diverse driving conditions, and a broader set of annotated object categories.

Sensor Setup

- **Cameras:** Six RGB cameras covering the full 360° field of view around the ego vehicle, with image resolution of 1600×900 pixels.
- **LiDAR:** A 32-beam LiDAR operating at 20 Hz, capturing point clouds with a maximum range of 100 meters.
- **RADAR:** Five automotive RADARs (front, rear, and sides), providing complementary velocity and range information.
- **GPS/IMU:** Global Positioning System and Inertial Measurement Unit sensors for precise localization, orientation, and vehicle dynamics.

Dataset Size and Structure

- **Scenes:** 1,000 driving sequences, each lasting approximately 20 seconds, collected in Boston and Singapore to ensure geographic and traffic diversity.
- **Keyframes:** Each scene is sampled at 2 Hz (every 0.5 seconds), resulting in 40,000 annotated keyframes. Intermediate frames are available at 20 Hz for LiDAR and RADAR data.
- **Annotations:** Over 1.4 million 3D bounding boxes across 10 object classes: Car, Truck, Bus, Trailer, Construction Vehicle, Pedestrian, Bicycle, Motorcycle, Bar-

rier, and Traffic Cone. Each annotation includes position, dimensions, orientation, velocity, and object attributes.

- **Splits:** 700 scenes for training, 150 for validation, and 150 for testing.
- **LiDAR Samples:** Approximately 1.4 million LiDAR sweeps across the dataset.
- **Image Samples:** About 1.4 million images across all cameras.

Tasks Provided

nuScenes provides benchmarks for:

- 3D object detection
- Tracking and trajectory forecasting
- Instance and semantic segmentation
- Motion prediction and planning
- Sensor fusion tasks (camera + LiDAR + RADAR)

Advantages and Limitations

- **Advantages:** nuScenes offers multimodal diversity, temporal structure, and challenging environmental conditions such as night and rain. It covers more object classes than KITTI and provides attributes (e.g., object movement state). Its 360° coverage ensures no blind spots.
- **Limitations:** The 32-beam LiDAR provides sparser point clouds than KITTI's 64-beam LiDAR. Although LiDAR runs at 20 Hz, annotations are sampled at 2 Hz, which limits temporal granularity for fine-grained motion tasks.

nuScenes is employed to evaluate the proposed ReliFusion framework. Unlike KITTI, it covers multiple environments (urban, suburban, highways), different weather conditions (clear, rainy), and illumination settings (day, night). This makes it a more rigorous benchmark for testing model reliability and robustness. The large scale of nuScenes ensures that the proposed method generalizes well beyond limited scenarios, validating the adaptability of ReliFusion in realistic autonomous driving contexts.

4.3 Dataset Comparison

A side-by-side comparison of the KITTI and nuScenes datasets is provided in Table 4.1.

Feature	**KITTI**	**nuScenes**
Collection Location	Karlsruhe, Germany	Boston (USA), Singapore
Numbre of Samples	14,999 frames	1,000 scenes (∼40,000 keyframes)
Object Classes	3 categories	10 categories
LiDAR	64-beam LiDAR (10 Hz)	32-beam LiDAR (20 Hz)
Cameras	2 RGB, 1242×375	6 RGB, 1600×900
RADARs	None	5
GPS/IMU	Yes	Yes
Annotations	2D + 3D boxes, difficulty levels	3D boxes + velocity + attributes
Environmental Diversity	Urban, rural, highway (daytime, clear)	Urban, suburban, highway (day/night, clear/rain)

Table 4.1: Comparison of KITTI and nuScenes datasets.

In summary, KITTI provides a compact yet high-quality dataset well-suited for benchmarking baseline methods such as TransfuseNet, while nuScenes offers large-scale, multimodal, and environmentally diverse data that enables the evaluation of advanced approaches such as ReliFusion, particularly in terms of robustness and reliability.

Chapter 5

Methodology Phase 1: TransfuseNet

5.1 Introduction

The objective of this research is to enhance accuracy, robustness, and computational efficiency of object detection systems for autonomous driving. Achieving reliable perception in dynamic environments and enabling real-time decision-making are critical for practical autonomous vehicle applications. Our research is structured into two distinct phases. In the first phase, we developed TransfuseNet, a lightweight fusion network specifically optimized for real-time performance and computational efficiency. By incorporating a compact model architecture, reducing resource consumption, and minimizing inference latency to under 40ms, TransfuseNet addresses the demanding requirements of high-speed autonomous driving scenarios.

In the second phase (presented in the next chapter), we extend our research by introducing ReliFusion, a framework specifically designed to address limitations of TransfuseNet related to robustness against sensor malfunctions and environmental challenges. ReliFusion incorporates adaptive fusion strategies to ensure consistent detection performance under sensor limitations and occlusions.

In this chapter, we provide a detailed description of TransfuseNet, its components, and experimental validations.

5.2 TransfuseNet

TransfuseNet involves multiple key components, including data representation techniques, a fusion network, late fusion strategies, and the proposal generation and detection head. Each of these components plays a key role in our approach to achieve accurate object detection.

5.2.1 Data Representation

Our research focuses on integrating data from two distinct sensor types: camera RGB images and LiDAR point cloud.

5.2.1.1 Camera Data Representation

The input for the camera stream consists of RGB images from the KITTI dataset. These RGB images have dimensions of 1242×375×3. To ensure consistency and facilitate subsequent processing, we perform min-max normalization on the pixel values. This normalization scales the pixel values to the range of 0 to 1, ensuring that the data is appropriately prepared for fusion with LiDAR information.

5.2.1.2 LiDAR Data Representation

The LiDAR stream, on the other hand, is represented in two distinctive ways: Bird's Eye View (BEV) and Frontal View (FV). These representations serve as a bridge between the raw LiDAR point cloud data and our fusion network, simplifying object detection and fusion processes. Our choice of data representation techniques, Bird's Eye View (BEV)

and Frontal View (FV), serves as a fundamental step in preparing sensor inputs for fusion and object detection. BEV provides a structured and precise view of the environment, while FV aligns LiDAR data with camera perspectives, enabling multimodal fusion and improving object detection. These representations lay the foundation for the subsequent stages of our initial methodology.

5.2.1.3 Bird's Eye View (BEV)

The BEV representation provides a compact yet comprehensive view of the environment by encoding the 3D point cloud into a 2D image-like format. This transformation offers several advantages:

- Compact and Structured Representation: BEV simplifies the complex 3D LiDAR point cloud into a 2D grid structure, making it efficient for storage and processing.
- Precise Spatial Information: BEV retains the precise spatial relationships between objects, preserving position and orientation information.
- Complementary to Camera Data: BEV complements the information captured by cameras, providing depth-based perspectives that enhance our understanding of the environment.

The BEV representation spans a physical space of [0, 100] meters along the x-axis, [-30, 30] meters along the y-axis, and [-0.6, 3.5] meters along the z-axis. It is discretized into a 1242×375 image with 7 distinct channels dedicated to height information, further enriching the data for fusion (Figure 5.1(b)).

5.2.1.4 Frontal View (FV)

Frontal View (FV) representation as shown in Figure 5.1(c) aligns LiDAR data with the camera's perspective, making it compatible for fusion with camera data. FV involves

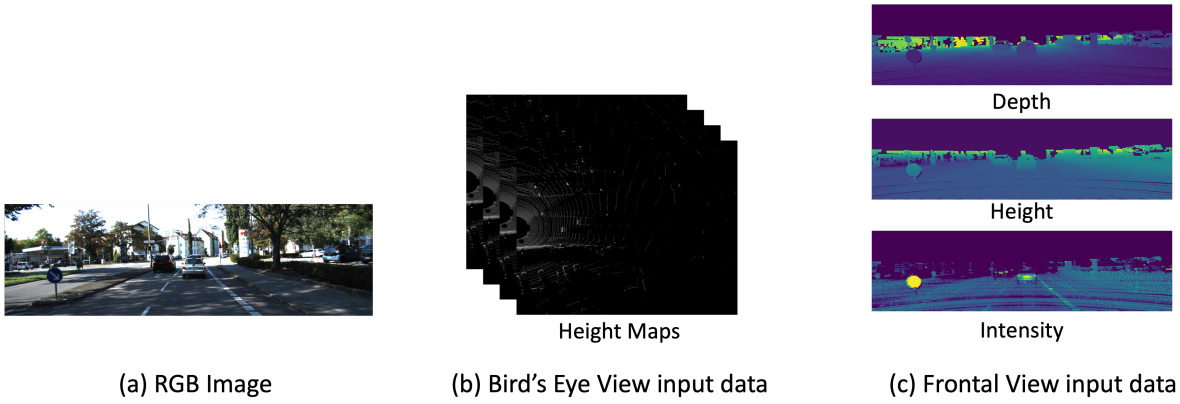

Figure 5.1: Input data of the our TransfuseNet. (b) Bird's Eye View aligns the data from the top overview while (c) Frontal View aligns LiDAR data with the camera's perspective.

deriving three essential features from sparse LiDAR point clouds: intensity, depth, and height maps. These features are then projected into a format that matches the camera's viewpoint. FV representation offers several advantages:

- Multimodal Information Fusion: FV facilitates the fusion of LiDAR and camera data at the feature level, enhancing the comprehensiveness of information.
- Rich Feature Set: FV includes intensity, depth, and height maps, each providing valuable insights about the environment. Intensity maps capture reflectance values, depth maps indicate distances and height maps represent object heights.
- Improved Object Detection: The combination of FV with camera data enhances object detection accuracy, particularly in challenging scenarios.

The three features in Frontal view (FV) representation are concatenated in the channels' dimensions to create a frontal view feature image with dimensions of $1242 \times 375 \times 3$. This encoding of FV data is normalized between 0 and 1, ensuring consistency and compatibility with other data modalities.

5.2.2 Fusion Network

In this section, We explore the details of our Fusion Network, a key component of our proposed initial methodology for Transformer-based LiDAR-Camera Fusion for Object Detection. The Fusion Network comprises two essential stages: mid-level fusion, which leverages the Transformer architecture, and late fusion, offering flexibility through both non-learnable and learnable fusion techniques. In our method, mid-level and late fusion techniques were chosen over early fusion to capitalize on their strengths in handling complex spatial relationships and sensor-specific features more effectively. Mid-level fusion allows for the integration of detailed, modality-specific features at an intermediate stage, enabling the transformer models to exploit spatial and temporal correlations across modalities for improved accuracy. Late fusion, on the other hand, combines decision-making outputs from different sensors at a later stage, enhancing the system's reliability by leveraging diverse perspectives. This approach ensures a more robust and versatile object detection capability, as it benefits from both the detailed feature integration of mid-level fusion and the comprehensive decision validation offered by late fusion, while avoiding the potential drawbacks of early fusion, such as the loss of modality-specific information and increased computational complexity

5.2.2.1 Mid-Level Fusion

Our Fusion Network is designed to harness the power of self-attention mechanisms found in transformers [67] to facilitate mid-level fusion. This stage plays a pivotal role in our methodology, as it allows us to incorporate global context into both the image and LiDAR modalities. Let's examine the critical elements of mid-level fusion:

Self-Attention Mechanism: The self-attention mechanism is the core of Mid-level fusion which was described in Equation (2.1) in details. In our Fusion Network, this mechanism allows us to capture the intricate relationships between LiDAR and camera data, enhancing the fusion process.

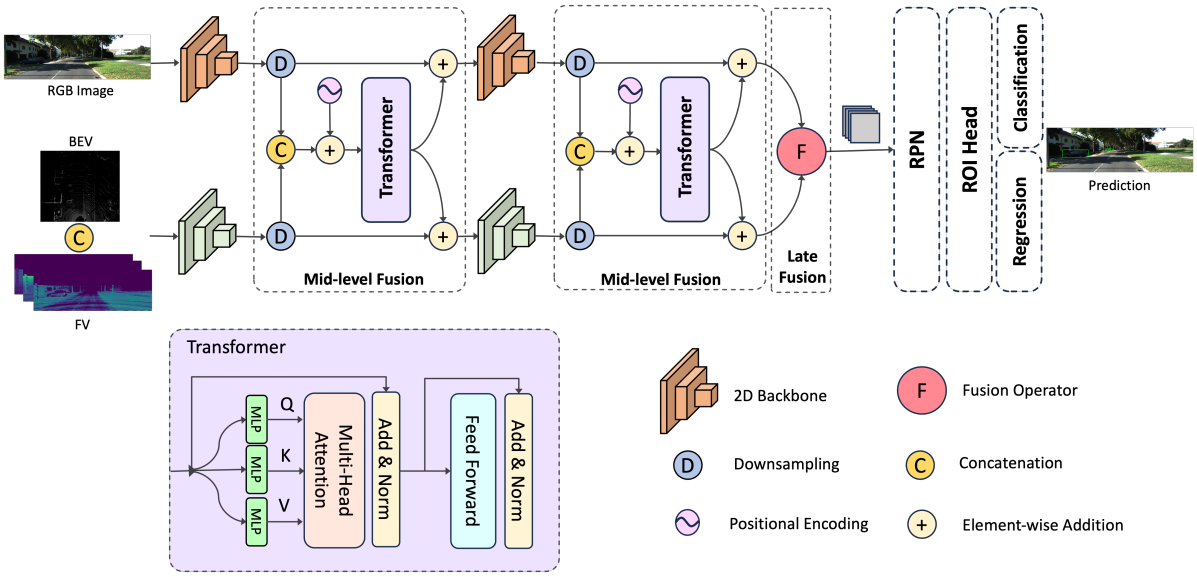

Figure 5.2: The overall architecture of our proposed TransfuseNet with single-view RGB and LiDAR BEV/FV inputs. The system employs multiple transformer layers for intermediate feature map fusion, followed by a late fusion operator. These fused features are input to a Region Proposal Network and a subsequent detection head for bounding box prediction.

Intermediate-Level Feature Maps: Our Fusion Network operates on intermediate-level feature maps, which are represented as 3D tensors with dimensions $H \times W \times C$. These feature maps capture rich information from both LiDAR and camera streams.

Concatenation of Features: To merge information from both modalities, we concatenate the individual features from each stream in the channels' dimensions. This results in a tensor with dimensions $(2 \times H \times W) \times C$.

Learnable Positional Embeddings: To further enhance spatial awareness, we employ learnable positional embeddings. These embeddings are added to the concatenated features, allowing the network to understand the spatial relationships among input tokens. This incorporation of positional information is crucial for improved performance and interpretation.

Transformer Processing: The prepared tensor, with positional embeddings, serves

as the input to the transformer. The transformer processes this input and produces an output tensor of the same size, creating context-aware representations.

Feature Map Integration: The output from the transformer is shaped into two feature maps, each with dimensions $H \times W \times C$. These feature maps represent a refined fusion of LiDAR and camera data, capturing relevant information at this intermediate level.

Resolution Management: Handling high spatial resolution feature maps can be computationally demanding. To address this challenge, we employ strategic techniques:

1. Average Pooling: We downsample higher resolution feature maps from the early encoder blocks to a fixed resolution of H = W = 8. This operation helps manage computational complexity while preserving essential information.

2. Bilinear Interpolation: After transformer processing, we upsample the output to the original resolution before element-wise summing with the existing feature maps. This ensures that the fused information aligns with the original resolution.

In summary, our Fusion Network's mid-level fusion stage, empowered by the Transformer architecture, grid structure feature, and fusion at different levels and resolutions, plays a pivotal role in understanding and integrating information from both LiDAR and camera modalities. The strategic use of learnable positional embeddings enhances spatial awareness, contributing to improved performance in object detection.

In the subsequent sections, we will explore the late fusion stage of TransfuseNet, detailing both non-learnable and learnable fusion techniques, and their impact on object detection accuracy.

5.2.2.2 Late Fusion Strategies

Late fusion marks the second pivotal stage in TransfuseNet, following mid-level fusion. This phase plays a vital role in integrating the comprehensive context-aware representa-

tions derived from both LiDAR and camera modalities. TransfuseNet adopts a sequential fusion strategy, transitioning seamlessly from mid-level fusion to late fusion. Within the late fusion stage, we explore two distinct approaches: (1) Non-learnable Fusion and (2) Learnable Fusion Operators. Each of these strategies contributes unique advantages to the overall framework, enhancing the experimental process and object detection accuracy.

I. Non-learnable Fusion

The non-learnable fusion strategy prioritizes efficiency and computational speed by avoiding the introduction of trainable weights. Instead, it relies on straightforward mathematical operations that can be performed rapidly. In our implementation, we employ two fundamental non-learnable fusion operators:

1) Elemental Addition

One of the non-learnable fusion operators utilized in TransfuseNet is elemental addition. This operator combines feature maps from the camera and LiDAR modalities by element-wise addition, as represented in Equation 5.1. This operation aggregates information without introducing any learnable parameters, resulting in accelerated processing times.

$$F_{\text{add}} = I_{\text{Camera}} \oplus I_{\text{LiDAR}} \tag{5.1}$$

2) Elemental Multiplication

The second non-learnable fusion operator employed in TransfuseNet is elemental multiplication. Similar to elemental addition, this operator combines feature maps element-wise, albeit through multiplication, as depicted in Equation 5.2. This operator offers another lightweight fusion mechanism suitable for applications where computational efficiency is a priority.

$$F_{\text{mul}} = I_{\text{Camera}} \otimes I_{\text{LiDAR}} \tag{5.2}$$

5.2.2.3 Learnable Fusion

In contrast to non-learnable fusion, the learnable fusion strategy introduces layers with trainable parameters, offering the network the capability to adapt and capture intricate interactions among features from different input modalities. TransfuseNet incorporates two learnable fusion operators, each designed to harness the learning potential inherent in the fusion process:

1) Multi-modal Factorized Bilinear Pooling (MFB) [82]

MFB [82] is one of the learnable fusion operators integrated into TransfuseNet. It enables the network to dynamically learn complex relationships between features extracted from camera and LiDAR data. This operator employs convolution layers to achieve fusion, maintaining the model's complexity while enabling feature learning (Figure 5.3). The use of batch normalization techniques ensures stability and efficient training.

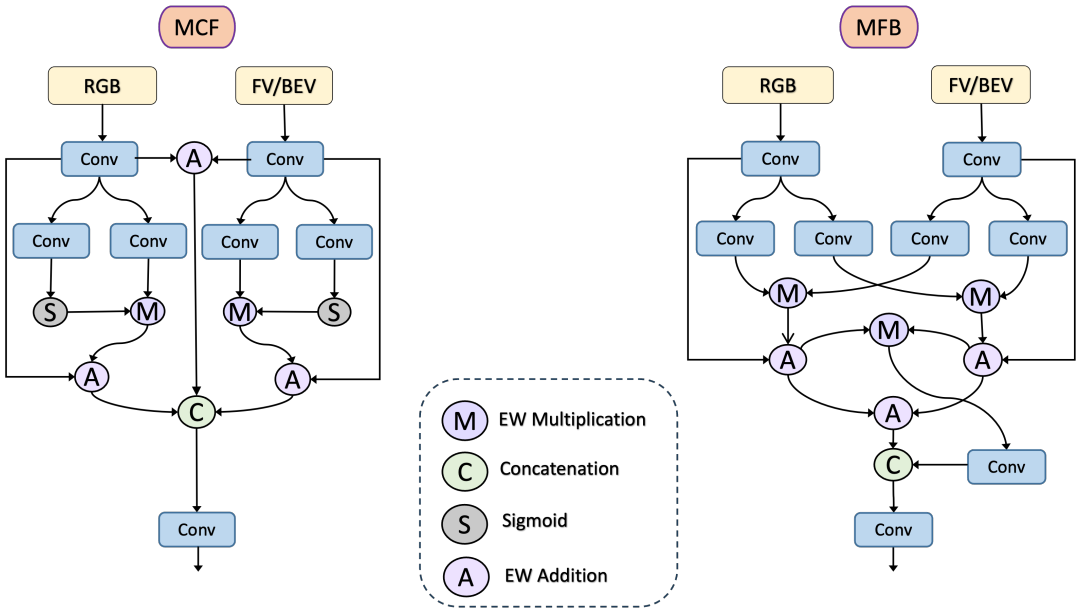

Figure 5.3: Learnable fusion operators. Left: Our proposed Multi-convolutional Fusion operator (MCF). Right: Multi-modal factorized Bilinear pooling fusion operator (MFB) [82].

2) Multi-Convolutional Fusion (MCF)

TransfuseNet introduces a novel learnable fusion operator called Multi-Convolutional Fusion (MCF). This fusion operator is designed to capture nuanced interactions between camera and LiDAR features through the application of convolution layers (Figure 5.3). By utilizing convolutional operations, Multi-Convolutional Fusion (MCF) facilitates feature learning and adaptation, enabling the model to understand complex cross-modal relationships.

MCF leverages the Sigmoid layer to determine the relevance of individual features. The sigmoid layer outputs a probability value for each feature, signifying its importance. Features with higher probability scores are deemed more important and are thus prioritized during the fusion process with subsequent layers. The application of Rectified Linear Unit (ReLU) as the primary activation function throughout TransfuseNet, along with specific parameter configurations such as padding value and kernel size, ensures the effective operation of these learnable fusion operators.

Learnable Vs Non-Learnable Fusion strategies: In summary, the late fusion strategies in TransfuseNet, whether non-learnable or learnable, play a pivotal role in synthesizing context-aware representations from camera and LiDAR modalities. The non-learnable fusion operators prioritize computational efficiency, making them suitable for applications with resource constraints. On the other hand, the learnable fusion operators introduce adaptability and the capacity to capture intricate feature interactions, enhancing object detection accuracy.

5.2.3 Region Proposal Generation and Detection Head

The Proposal Generation and Detection Head follows a two-stage paradigm inspired by Faster R-CNN [55], where the network first generates region proposals and then classifies the detected objects.

5.2.3.1 Feature Extraction and Proposal Generation

In the first stage, the Feature Extraction module extracts high-level features from the input data. For this purpose, TransfuseNet employs ConvMixer [64], a light and SOTA convolutional neural network architecture known for its effectiveness in feature extraction tasks. The Feature Extraction module processes the transformed LiDAR and camera data, capturing essential patterns and information required for subsequent stages.

Following feature extraction, the network moves to the Proposal Generation stage. This phase is responsible for generating region proposals that potentially contain objects of interest. To accomplish this, TransfuseNet employs a Region Proposal Network (RPN) that operates in conjunction with the extracted feature maps.

In the Region Proposal Network (RPN), a set of anchor boxes is used to propose potential object regions. These anchor boxes span different scales and aspect ratios, allowing the network to consider a wide range of object sizes and shapes. In the TransfuseNet experiments, nine anchor boxes are used, formed by combining three scales and three aspect ratios. The RPN evaluates each anchor box and predicts two critical pieces of information for each one:

Objectness Score: This score represents the probability of an anchor box containing an object. It helps the network differentiate between regions that are likely to contain objects and those that are not.

Bounding Box Offsets: These offsets adjust the dimensions of the anchor box to better fit the object's actual location and size within the region. These offsets are essential for refining the proposal's accuracy.

The combination of objectness scores and bounding box offsets allows the RPN to generate region proposals that adapt to the specific objects present in the input data.

5.2.3.2 Detection Head Module

Following the proposal generation stage, TransfuseNet proceeds to the Detection Head module. This module is responsible for further refining the region proposals and classifying objects within these proposals.

The Detection Head module consists of three convolutional layers followed by a dropout layer. These convolutional layers are designed to process the proposed regions and extract more refined features. The dropout layer introduces regularization, helping prevent overfitting and enhancing the network's generalization capabilities.

Of the three convolutional layers, the last two are particularly used to detect objects. These layers focus on identifying the objects' precise locations and predicting their associated objectness scores and class labels.

Bounding Box Refinement: The second convolution layer refines the bounding boxes generated by the RPN. It adjusts the positions and sizes of the proposals to align more accurately with the actual objects present in the region.

Objectness Score Prediction: The last convolution layer predicts the objectness scores for the refined proposals. These scores represent the likelihood of each proposal containing a real object. High objectness scores indicate high confidence in the presence of an object.

Object Classification: In addition to objectness scores, the Detection Head module is responsible for object classification. For each refined proposal, it predicts the class label of the object contained within the proposal. In this context, the primary class of interest is "Car," but the network can be adapted to detect other classes as well.

5.2.3.3 Non-Maximum Suppression (NMS)

To produce a reliable set of object detections during inference, TransfuseNet employs a post-processing technique known as Non-Maximum Suppression (NMS) [18]. NMS is

applied to the region proposals generated by the RPN and refines them by selecting a subset of high-scoring proposals while discarding redundant or overlapping ones.

The core idea of NMS is to suppress proposals with low objectness scores and retain those with high confidence. This operation helps remove duplicate detections of the same object and ensures that only the most accurate and promising proposals are considered as final detections.

In practice, NMS operates by iteratively selecting the proposal with the highest objectness score, discarding nearby proposals that overlap with it significantly (based on a specified IoU threshold), and moving on to the next highest-scoring proposal. This process continues until all proposals have been evaluated.

By incorporating NMS, TransfuseNet refines its object detections, eliminates redundancies, and produces a concise and accurate set of detected objects, ultimately enhancing the overall performance of the system.

In the next section, we will discuss the experimental setup, where we will provide a comprehensive overview of the datasets used for training and evaluation, detail the training procedures, and discuss the key evaluation metrics employed to assess the performance of TransfuseNet in the context of object detection.

5.3 Experiments on the TransfuseNet

In this section, we present the experimental methodologies and evaluations specifically conducted on the KITTI dataset, suitable for evaluating the computational efficiency and accuracy goals of the TransfuseNet approach. Later, in the context of ReliFusion, we will extend our evaluation to the nuScenes dataset, which provides a broader and more challenging range of scenarios. By comparing our method with existing state-of-the-art techniques, we aim to clearly present the improvements and progress our method brings to the field. To ensure the robustness and validity of our design decisions, we conduct thorough ablation studies, delving into the nuances of each choice.

5.3.1 Dataset and Metric

Our choice of the KITTI object detection benchmark [16] for the evaluation of TransfuseNet is influenced by its rich repository of data in object detection. It includes 7,481 annotated training images and 7,518 test images, accompanied by their respective point clouds. Altogether, this dataset features 80,256 objects that have been labeled. Given the constraints and structure of the KITTI benchmark, We decided to split the training images into two groups: training and validation.

In our study, we specifically focus on the 'car' category within the KITTI dataset for two main reasons. Firstly, this category contains a large number of instances, providing the extensive amount of data needed for the thorough training of deep network-based methods. Secondly, by concentrating on this category, we aim to improve our network's performance in detecting cars, enhancing its precision and accuracy.

To better understand our results, the KITTI dataset divides objects into three difficulty levels: easy, moderate, and hard. This categorization is based on factors like the size of the object, how much it's blocked from view, and how much it's cut off from the frame. This detailed way of classifying objects helps us evaluate how strong and flexible our detection models are under different levels of complexity.

To quantify the efficacy of our network in detecting cars across these three difficulty tiers, we employ the Average Precision (AP%) metric. This metric, widely recognized for providing a balanced measure of a model's precision and recall, is evaluated using an IoU threshold of 0.7. This IoU threshold ensures that only detections closely matching the ground truth are considered accurate, maintaining the strictness of our evaluations.

5.3.2 Experimental Setting

This research integrates multiple sensor modalities, including camera RGB images and LiDAR point cloud data. Utilizing these sensor types not only enriches the input data but also enables our model to capitalize on the distinct advantages offered by each

modality. The detailed and complex features shown in RGB images complement the depth information and spatial accuracy inherent to LiDAR point clouds. Converting these sensor inputs into the FV/BEV format standardizes the input structure, enabling consistent processing and reducing inconsistencies from varying data formats.

5.3.3 Hardware and Software Configuration

Our experiments utilize a system equipped with an Intel Core i9-10900K CPU (10 cores, 20 threads), 32 GB of memory, and an NVIDIA GeForce RTX 3090 GPU with 24 GB of dedicated GDDR6X memory, running on Ubuntu 20.04 with CUDA 11.3 and cuDNN 8.2. This configuration supports the computational requirements of the research.

5.3.4 Data Preprocessing

We use the ConvMixer [64] model for feature extraction due to its effectiveness in capturing complex patterns. For camera RGB images, ConvMixer extracts relevant features for object identification. The same model is applied to LiDAR data to maintain a consistent feature space and ensure a unified extraction process across both modalities.

5.3.5 Hyperparameter Tuning

In our experiments, we use the Adam optimizer [28] for its adaptive learning rate and efficient convergence. The initial learning rate is set to 0.001, balancing learning speed and stability. A weight decay of 0.00001 is applied to mitigate overfitting and enhance generalization.

The training process runs for 150 epochs with a batch size of eight, balancing computational efficiency and learning stability. To refine feature representation and enhance the network's ability to process multi-modal data, we integrate two transformers with

four attention heads each. These transformers are designed to handle sequential data and capture long-term dependencies, improving the network architecture.

5.3.6 Loss Function

The training process follows principles from the Faster R-CNN architecture [55], incorporating both object detection and classification. The loss function consists of two interconnected elements to optimize both tasks effectively

- **Classification Loss**

  It evaluates the model's ability to differentiate between classes, particularly in categorizing detected objects as a 'Car' or not. For this, the cross-entropy loss function is :

$$L_{cls} = -\sum [y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)] \tag{5.3}$$

  Here, y_i represents the ground truth label, taking the value 1 if the detected object is indeed a 'Car' and 0 otherwise. On the other hand, p_i stands for the model's predicted probability for the object being a 'Car'.

- **Regression Loss**

  Contrary to the classification loss, which operates on a categorical level, the regression loss concentrates on the accuracy of the model in determining the exact bounding box coordinates of the detected object. To optimize the precise location of detected objects, we incorporate the smooth L1 loss function [17], which is particularly favored for its stability and resistance to outliers. Mathematically, the smooth L1 loss function is defined as:

$$L1_{smooth} = \begin{cases} 0.5(y_i - \hat{y})^2/\beta & \text{if } |y_i - \hat{y}| < \beta \\ |y_i - \hat{y}| - 0.5 * \beta & \text{otherwise} \end{cases} \tag{5.4}$$

In this context, y_i stands for the ground truth coordinates, and $\hat{y}$ is the model's predicted coordinate values. The hyperparameter β, which we've set to 1 for our experiments, regulates the transition between the L2 loss and L1 loss, striking a balance between the two to ensure optimal performance.

The total loss, combining classification and regression components, optimizes the model during training and is defined as:

$$L_{total} = L_{cls} + \lambda \cdot L_{reg} \tag{5.5}$$

in which the coefficient λ regulates the balance between classification and regression loss. It adjusts the emphasis between the accuracy of object classification and the precision of bounding box predictions. The choice of λ adjusts the balance between classification and regression, preventing either from dominating the training process. This ensures the model effectively identifies and localizes objects. The optimal λ value is determined empirically based on the specific task requirements.

5.3.7 Results

Our results are derived from testing on the KITTI dataset, specifically focusing on how different input modalities influence the detection capability of our algorithm.

5.3.8 Different Input modalities

The baseline configuration utilizes only RGB data from camera images, excluding LiDAR information. This setup establishes a reference point for assessing the contribution of

Input Data	2D AP (%)		
	Easy	*Moderate*	*Hard*
RGB	87.08	83.13	77.07
RGB + BEV	91.58	85.49	80.28
RGB + FV	92.19	86.93	79.94
RGB + BEV +FV	**93.85**	**89.93**	**83.17**

Table 5.1: Average Precision (AP%) of TransfuseNet with respect to input data. Element-wise addition is employed as the late fusion operator in this table. The 2D AP refers to the Average Precision for two-dimensional image data, which is a measure of the model's accuracy in detecting objects in standard camera images.

additional sensor data.

Integrating LiDAR data introduces a significant improvement in detection accuracy. LiDAR provides depth information and precise spatial representation, enhancing detection performance across all object categories. As shown in Table 5.1, the incorporation of LiDAR consistently improves detection results, underscoring its importance in 2D object detection. For objects that are more readily detected—specifically, those within the 'easy' and 'moderate' categories—FV consistently outperforms BEV. The higher level of detail in FV provides a clearer representation, facilitating more effective object detection.

However, for objects in the 'hard' category, the combination of RGB and BEV yields better detection performance. Objects in the 'hard' category present significant detection challenges due to their smaller size or heavy occlusion. The top-down perspective of BEV mitigates occlusion issues that often affect other views, providing a more reliable representation in such cases.

Experiments with different input modalities revealed that combining BEV and FV enhances detection performance. This approach leverages BEV's ability to handle occlusion and FV's detailed representation, resulting in improved detection across all difficulty levels.

5.4 Summary

Our quantitative analysis on the KITTI dataset provides insights into the suitability of different input modalities for our network. The choice of modality depends on the specific challenges associated with each object category. While BEV proves effective in handling occlusion in the 'hard' category, FV demonstrates superior performance in detecting objects in the 'easy' and 'moderate' categories.

5.4.1 Different Fusion Strategies

The complexity of object detection necessitates diverse data fusion strategies, particularly when integrating multiple input modalities. To develop an effective detection system, it is essential to evaluate the performance of different data fusion approaches.

This section presents a detailed evaluation of various late fusion operators within the architectural framework. A quantitative analysis compares the proposed Multi-Channel Fusion (MCF) method with the established Multi-Fusion Block (MFB) approach. As shown in Table 5.2, the study examines a range of fusion strategies, including both non-learnable techniques (such as element-wise addition and multiplication) and learnable fusion methods.

Our results clearly demonstrate that learnable fusion methods outperform non-learnable counterparts in terms of detection performance. This difference is particularly evident in the detection of smaller and more complex objects, categorized as 'hard' in our study. In this category, the limitations of non-learnable fusion methods become more apparent, emphasizing the advantages of learnable approaches in challenging object detection scenarios.

It is important to highlight that our findings consistently show the superior performance of the MCF method compared to the multi-modal factorized bilinear pooling (MFB) method. In the 2D object detection task, MCF achieves performance gains of

Fusion operator	2D AP (%)			BEV AP (%)		
	Easy	*Moderate*	*Hard*	*Easy*	*Moderate*	*Hard*
Add	93.85	89.93	83.17	86.87	80.27	74.83
Mul	94.39	90.78	83.96	90.46	84.03	78.77
MFB	96.27	94.72	89.92	94.24	89.16	85.03
MCF (ours)	**97.53**	**94.92**	**91.65**	**94.93**	**91.02**	**87.12**

Table 5.2: Performance evaluation of TransfuseNet using Average Precision (AP%). The table results are based on inputs of RGB images and concatenated BEV and FV representations. The 2D AP refers to the Average Precision for two-dimensional image data, which is a measure of the model's accuracy in detecting objects in standard camera images. BEV AP stands for Bird's Eye View Average Precision, indicating the accuracy of object detection when data is represented in a top-down view.

+1.26, +0.20, and +1.73 percentage points for the easy, moderate, and hard categories, respectively. For the Bird's Eye View (BEV) detection task, the advantage of MCF is even more pronounced, surpassing MFB by +0.69, +1.86, and +2.09 percentage points across different scenarios. These improvements are primarily attributed to MCF's use of a sigmoid layer, which effectively identifies and enhances significant features within the fusion process, thereby improving overall detection performance.

For non-learnable fusion methods, our evaluation indicates that element-wise multiplication consistently outperforms addition across all scenarios. This improved performance can be attributed to its ability to emphasize dominant features while simultaneously suppressing less relevant ones before passing them to subsequent model layers.

These findings reinforce the effectiveness of the MCF fusion method, particularly in comparison to MFB. The results establish MCF as a viable and effective approach for both 2D and BEV object detection tasks, representing a notable advancement in multimodal fusion strategies.

Method	Input Data		2D AP (%)			BEV AP (%)		
	LiDAR	Image	*Easy*	*Moderate*	*Hard*	*Easy*	*Moderate*	*Hard*
OMNI3D [4]	-	✓	95.78	92.72	84.81	31.70	21.20	18.43
MonoNeRD [74]	-	✓	94.60	86.89	77.23	31.13	23.46	20.97
NeurOCS [44]	-	✓	96.39	91.08	81.20	32.27	24.49	20.89
Pseudo-Stereo [11]	-	✓	95.75	90.27	82.32	32.64	23.76	20.64
SNVC [32]	-	✓	96.33	93.33	85.81	86.88	73.61	64.49
IA-SSD [84]	✓	-	96.26	93.54	88.49	92.79	89.33	84.35
BtcDet [75]	✓	-	96.23	93.47	88.55	92.81	89.34	84.55
CT3D [58]	✓	-	96.28	93.30	90.58	92.36	88.83	84.07
Pointpillars [30]	✓	-	94.00	91.19	88.17	90.07	86.56	82.81
PointRCNN [59]	✓	-	95.92	91.90	87.11	92.13	87.39	82.72
SVGA-Net [22]	✓	-	96.05	94.67	91.86	92.07	89.88	85.59
CAT-Det [83]	✓	✓	95.97	94.71	**92.07**	92.59	90.07	85.82
3D-CVF [81]	✓	✓	96.87	93.36	86.11	93.52	89.56	82.45
EPNet [25]	✓	✓	96.25	94.44	89.99	94.22	88.47	83.69
STD [79]	✓	✓	96.14	93.22	90.53	94.74	89.19	86.42
M3DETR [19]	✓	✓	97.39	94.83	91.10	94.41	90.37	85.98
TransfuseNet w/ MFB	✓	✓	96.27	94.72	89.92	94.24	89.16	85.03
TransfuseNet w/ MCF	✓	✓	**97.53**	**94.92**	91.65	**94.93**	**91.02**	**87.12**

Table 5.3: Evaluation results on KITTI 2D and BEV object detection benchmark (car). We evaluated TransfuseNet against the latest state-of-the-art results on the test set, using mean Average Precision measured at 40 recall positions for comparison. The best results appear in bold.

Methods	CAT-Det [83]	M3DETR [19]	BtcDet [75]	TransfuseNet w/ MCF
# Param.	30M	76M	35M	**7M**
Inference (ms)	60	180	80	**20**

Table 5.4: Comparative analysis of the number of parameters and inference time, evaluated on an NVIDIA GeForce RTX 3090 GPU with batch size 1.

5.4.2 State-of-the-Art Comparison

As shown in Table 5.3, TransfuseNet, utilizing our proposed learnable fusion operator MCF, is evaluated against state-of-the-art networks in both BEV and the easy category of 2D object detection. Additionally, when compared to other leading approaches categorized by input data type, TransfuseNet demonstrates superior performance, particularly when compared to models relying solely on image or LiDAR data. Furthermore, in 2D object detection, the integration of MCF and MFB in TransfuseNet results in improved performance over existing state-of-the-art models in the easy and moderate categories, respectively.

In addition to its accuracy, TransfuseNet achieves significantly lower inference time compared to all other methods, demonstrating its computational efficiency—an essential factor for safe autonomous driving. As shown in Table 5.4, TransfuseNet is three times faster than CAT-Det and nine times faster than M3DETR, both of which are transformer-based methods, while also maintaining a considerably lower parameter count.

5.4.3 Qualitative Results

As shown in Figure 5.4, TransfuseNet effectively detects small and occluded objects, a task that remains challenging for RGB-only models. This highlights the significance of incorporating LiDAR data for improved 2D object detection. Additionally, Figure 5.5 demonstrates the robustness of TransfuseNet, where the network correctly identifies

objects that are not labeled as ground truth. While these detections are classified as false positives, they are in fact true positives.

5.4.4 Ablation Study

In this section, we conduct ablation studies on various components, including input data types, backbone models, fusion techniques, and Transformer structure.

Initially, we evaluate the effectiveness of multi-view features across different input modalities and late fusion operators. The results, summarized in Table 5.1, demonstrate that the integration of RGB, BEV, and FV consistently outperforms other combinations, regardless of the late fusion operator used. Furthermore, our proposed MCF method achieves superior performance compared to alternative fusion techniques in all scenarios.

We also examine the impact of varying the number of Transformer blocks within TransfuseNet. As shown in Table 5.5, optimal performance is achieved with two Transformer blocks. The absence of Transformer blocks, and consequently mid-level fusion, significantly degrades results, highlighting their importance in improving fusion effectiveness. Additionally, diminishing returns observed with three blocks suggest that increasing the number of parameters may negatively affect performance. Moreover, Table 5.6 illustrates the superior effectiveness of Transformers as mid-level fusion operators compared to addition or multiplication methods.

Finally, as indicated in Table 5.7, we compare our model across various parameters, including the number of attention heads and layers, as well as different backbone models. Our default configuration includes two Transformer layers, eight attention layers, four attention heads, and ConvMixer for BEV and RGB feature extraction. ConvMixer is selected as the primary backbone due to its superior accuracy compared to alternatives such as VGG-16 and ResNet-34. Notably, ConvMixer features a simpler architecture and requires significantly fewer parameters, approximately one-tenth of those in ResNet-34.

Network Parameter	Value	2D AP (%)		
		Easy	*Moderate*	*Hard*
Attention layer	2	95.76	91.25	86.61
	4	95.62	93.93	86.92
	6	96.49	93.01	88.88
Attention head	2	96.12	93.48	87.75
Backbone	VGG-16	96.24	91.65	90.43
	ResNet-34	96.97	93.26	88.54
Default Config	-	**97.53**	**94.92**	**91.65**

Table 5.7: Ablation study of 2D object detection. Comparison of different model structures' results on the KITTI validation set.

5.5 Limitations and Motivation for Further Improvement

TransfuseNet demonstrates efficient and real-time performance in LiDAR-camera fusion with satisfactory detection accuracy. However, despite these achievements, the approach has inherent limitations related to its robustness against sensor malfunctions and degradation scenarios. Specifically, the fixed fusion weights utilized by TransfuseNet limit its adaptability in handling scenarios where sensors experience partial or complete failure. These challenges underline the necessity for a more robust, adaptive, and reliability-driven fusion framework, which we address in the subsequent chapter by introducing ReliFusion.

Ground Truth

2D Prediction of TransfuseNet

BEV Prediction of TransfuseNet

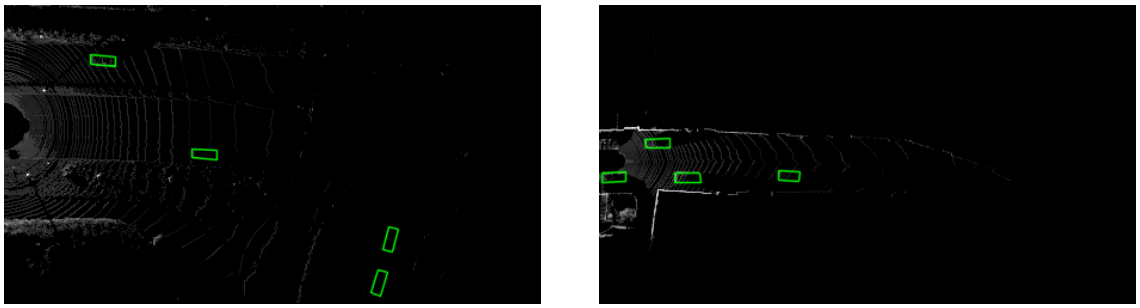

Figure 5.4: Qualitative detection results of our TransfuseNet on KITTI validation samples. Green and blue bounding boxes are True positive detection and Ground truth, respectively.

Figure 5.5: Sample from the KITTI dataset illustrating the capability of our network to accurately detect an object despite incorrect annotation. The green bounding box indicates true positive detection, while the blue bounding box represents ground truth.

# Transformer block	2D AP (%)		
	Easy	*Moderate*	*Hard*
0	87.97	84.92	81.67
1	95.75	90.93	85.03
2	**97.53**	**94.92**	**91.65**
3	96.88	94.53	89.94

Table 5.5: Evaluating the impact of the number of transformer blocks employed in TransfuseNet.

Mid-level fusion operator	2D AP (%)		
	Easy	*Moderate*	*Hard*
No Mid-fusion	87.97	84.92	81.67
Add	88.71	86.52	81.79
Mul	89.93	87.67	83.41
Transformer	**97.53**	**94.92**	**91.65**

Table 5.6: The effectiveness of different Mid-level fusion operator with two blocks utilized in TransfuseNet w/ MCF as late fusion operator.

Chapter 6

Methodology Phase 2: ReliFusion

6.1 Introduction

In the preceding chapter, we introduced TransfuseNet, demonstrating efficient real-time fusion of LiDAR and camera data for object detection. Despite achieving high computational efficiency and satisfactory detection accuracy, TransfuseNet's fixed fusion weights limit its robustness and adaptability, particularly under scenarios involving sensor degradation, occlusions, or malfunctions.

To overcome these limitations, this chapter introduces ReliFusion, our advanced fusion framework specifically designed to enhance robustness and reliability in 3D object detection by dynamically adapting to sensor reliability. ReliFusion quantifies the trustworthiness of sensor inputs through confidence scores, which dynamically guide the fusion process. By incorporating adaptive confidence-weighted fusion mechanisms, ReliFusion maintains robust and accurate detection even under severe sensor malfunctions.

This chapter comprehensively describes each component of ReliFusion, including feature extraction, spatio-temporal aggregation, reliability-driven fusion, and end-to-end optimization, highlighting how our approach addresses limitations identified from our initial TransfuseNet methodology.

Unlike early fusion strategies that combine raw sensor representations and late fusion approaches that merge independent predictions, ReliFusion adopts a mid-level fusion paradigm in BEV space. This design choice allows both modalities to learn rich modality-specific representations prior to interaction, while still enabling cross-modal reasoning at the feature level. The selection of mid-level placement was informed by both literature review (Chapter 3) and preliminary empirical comparisons in Phase 1 (Chapter 5).

6.2 ReliFusion

ReliFusion is a novel LiDAR-camera fusion framework designed to enhance the robustness of 3D object detection by dynamically adapting to sensor reliability. Unlike conventional fusion approaches that assume equal reliability across modalities, ReliFusion introduces a reliability-driven fusion mechanism that quantifies the trustworthiness of sensor inputs and adjusts fusion weights accordingly. The proposed framework consists of multiple key components. First, the multi-view image and LiDAR feature extraction module processes camera and LiDAR data in the Bird's Eye View (BEV) space to ensure spatial alignment. Then, the Spatio-Temporal Feature Aggregation (STFA) module integrates self-attention mechanisms to model both spatial correlations across multiple camera views and temporal dependencies across consecutive frames, enhancing detection stability. To further account for varying sensor reliability, the Reliability Module utilizes Cross-Modality Contrastive Learning (CMCL) to align LiDAR and camera features in a shared embedding space and assigns confidence scores to quantify each modality's reliability. These confidence scores are then leveraged in the Confidence-Weighted Mutual Cross-Attention (CW-MCA) module, where attention-based feature fusion dynamically prioritizes the more reliable modality while mitigating the impact of sensor degradation. Finally, the entire network undergoes a multi-stage training process, where individual modules are pre-trained before end-to-end fine-tuning using a multi-task loss function that jointly optimizes detection accuracy, feature alignment, and confidence prediction.

By integrating feature extraction, spatio-temporal modeling, reliability assessment, and adaptive fusion, ReliFusion enhances robustness in challenging scenarios, including partial LiDAR occlusions, adverse weather conditions, and sensor malfunctions.

6.2.1 Multi-View Image and LiDAR Feature Extraction

ReliFusion utilizes a multi-view camera setup and LiDAR point clouds to generate a robust feature representation for 3D object detection. To effectively integrate these inputs, the method employs Bird's Eye View (BEV) transformation for both LiDAR and image data, ensuring spatial alignment before fusion.

6.2.1.1 LiDAR Feature Extraction

The LiDAR point cloud is processed using VoxelNet [87] as 3D backbone, which encodes voxelized input into high-level LiDAR feature representations denoted as F_{LiDAR}. To enhance feature extraction, the point cloud is voxelized into uniform grid cells of size $0.075m \times 0.075m \times 0.2m$, similar to CenterPoint [80], allowing efficient sparse convolutional processing.

Although voxel-based encoding provides strong geometric structure and compatibility with BEV representations, voxelization and sparse 3D convolution remain computationally intensive stages in the LiDAR processing pipeline. The discretization of the point cloud into fine-grained voxels introduces preprocessing overhead and increases memory access complexity, particularly when high spatial resolution is used.

Alternative representations may reduce this bottleneck. Pillar-based encoders collapse the vertical dimension to simplify computation, while point-based architectures operate directly on raw points without voxel discretization. Hybrid approaches combining dynamic voxelization with learned sampling strategies may further improve efficiency.

In the context of ReliFusion, voxelization was selected to maintain compatibility with BEV-aligned mid-level fusion and established detection backbones. However, future

deployment-oriented implementations could replace full voxelization with lighter-weight pillar encodings or hardware-optimized sparse convolution libraries to reduce latency without fundamentally altering the fusion strategy.

6.2.1.2 Multi-View Image Feature Extraction

Each camera image I_k (where $k = 1, ..., 6$ for the six-camera setup in nuScenes) is processed using a ConvMixer backbone [64]. ConvMixer combines patch embedding with depthwise and pointwise convolutions, acting as a lightweight hybrid between CNNs and vision transformers. This design enables extraction of high-level semantic features while maintaining computational efficiency. The resulting per-view feature maps F^k_{Camera} are subsequently aligned with the LiDAR BEV representation.

6.2.1.3 Transformation to BEV

To achieve spatial consistency between LiDAR and camera features, ReliFusion employs the Lift-Splat-Shoot (LSS) transformation [49], which projects multi-view image features into BEV space. This transformation consists of three main steps:

1. **Lifting**: Each image pixel is unprojected into 3D space using known camera intrinsics.
2. **Splatting**: The projected 3D features are accumulated onto a common BEV grid.
3. **Shooting**: The BEV feature map is generated by aggregating splatted features.

The resulting BEV-transformed image feature map, denoted as $F_{\text{Camera-BEV}}$, aligns with F_{LiDAR} in the BEV space, ensuring a unified feature representation for subsequent processing.

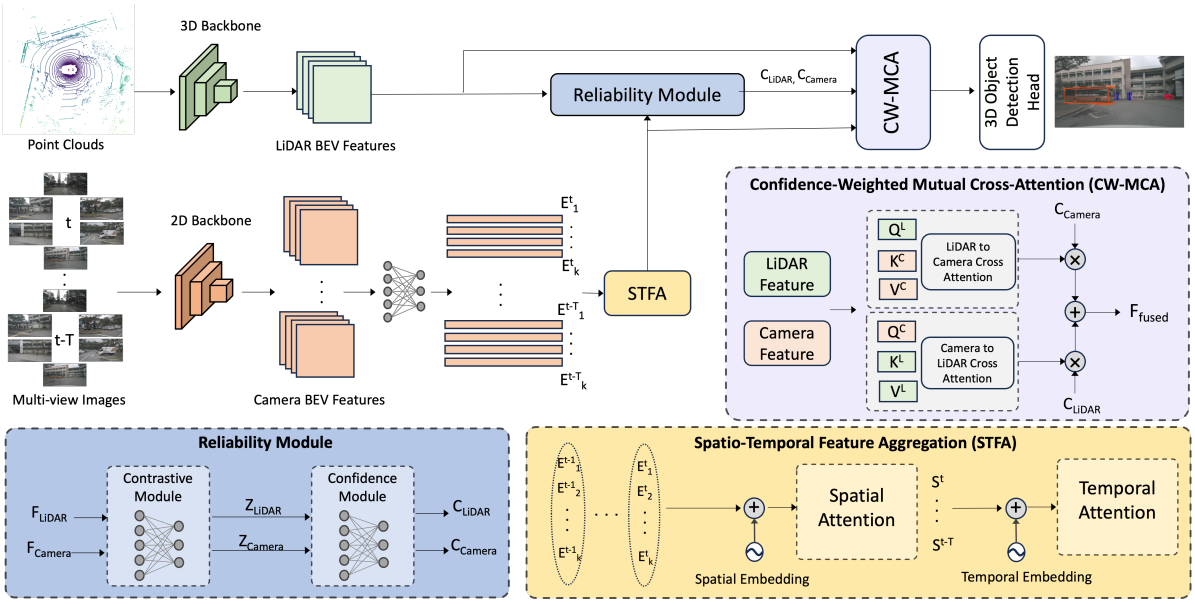

Figure 6.1: ReliFusion architecture explicitly designed to overcome limitations of fixed-weight fusion by integrating adaptive confidence-based cross-attention mechanisms for robust LiDAR-camera fusion in BEV representation.

6.2.2 Spatio-Temporal Feature Aggregation (STFA)

To overcome the limitations of TransfuseNet, which lacked temporal context and adaptability, ReliFusion incorporates the STFA module. The STFA module enhances robustness by sequentially applying spatial and temporal attention, allowing the network to capture relationships across multiple camera views at each time step as well as dependencies across consecutive frames. By integrating learnable spatial embeddings and temporal encodings, the module ensures effective feature aggregation and improves detection stability in dynamic environments.

Temporal modeling is applied to camera BEV features before fusion because camera features exhibit higher temporal variability and benefit more from motion-based aggregation. LiDAR features, while sparse, are geometrically stable and already aligned in BEV space, reducing the marginal benefit of additional temporal modeling relative to computational cost.

6.2.2.1 Spatial Attention for Inter-View Aggregation

At each time step t, multi-view image features $\{F_k^t\}_{k=1}^6$, where k indexes the views, are extracted using the ConvMixer [64] as backbone . These BEV features $F_k^t \in \mathbb{R}^{C\times H\times W}$ are flattened and projected into an embedding space using a linear transformation:

$$E_k^t = W_s^\top \cdot \text{Flatten}(F_k^t) + b_s \tag{6.1}$$

In Equation (6.1), $W_s^\top \in \mathbb{R}^{d\times(C\cdot H\cdot W)}$ is a learnable weight matrix, $b_s \in \mathbb{R}^d$ is a bias vector, and Flatten$(\cdot)$ reshapes the feature map into a sequence.

To model spatial relationships across different views at the same time step, ReliFusion employs a spatial self-attention mechanism. For each view k (query), the embeddings E_j^t from all views $j \in \{1, ..., 6\}$ (keys and values) are considered. The attention weights are computed using the Equation (6.2):

$$\text{Attention}(Q_k, K_j) = \text{Softmax}\left(\frac{Q_k K_j^\top}{\sqrt{d}}\right) \tag{6.2}$$

where:

$$Q_k = W_q E_k^t, \quad K_j = W_k E_j^t, \quad V_j = W_v E_j^t \tag{6.3}$$

with learnable projection matrices $W_q, W_k, W_v \in \mathbb{R}^{d \times d}$. The attended feature representation for view k at time t is calculated in Equation (6.4).

$$S_k^t = \sum_{j=1}^{6} \text{Attention}(Q_k, K_j) V_j \tag{6.4}$$

Finally, the aggregated spatial feature representation at time t is obtained by combining all attended views (Equation (6.5)).

$$S^t = \{S_k^t \mid k = 1, ..., 6\} \tag{6.5}$$

This process allows each view to incorporate information from all others, mitigating occlusions and enhancing spatial consistency.

6.2.2.2 Temporal Attention for Cross-Time Dependency

To maintain detection stability over time, the spatially aggregated features $\{S^t\}_{t=1}^T$ (where T denotes the number of time steps) are processed using temporal attention mechanisms. First, temporal embeddings P_t are added to encode sequential order (Equation (6.6)).

$$\tilde{S}^t = S^t + P_t(t) \tag{6.6}$$

where $P_t(t) \in \mathbb{R}^d$ is a learnable temporal encoding for each time step. Temporal attention (Equation (6.7)) is then applied across all time steps to capture dependencies:

$$\text{Attention}_t(Q, K, V) = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V \tag{6.7}$$

where:

$$Q = W_q\tilde{S}^t, \quad K = W_k\tilde{S}^{t'}, \quad V = W_v\tilde{S}^{t'} \tag{6.8}$$

for $t' \neq t$, with learnable projection matrices $W_q, W_k, W_v \in \mathbb{R}^{d\times d}$. The final temporally aggregated feature representation is computed as Equation (6.9).

$$T = \sum_{t=1}^{T} \text{Attention}_t(Q, K, V) \tag{6.9}$$

This mechanism enhances the model's ability to handle motion and occlusions by capturing long-term dependencies between frames.

6.2.2.3 Refinement with Layer Normalization

To stabilize training and ensure smooth feature aggregation, each attention module is followed by Layer Normalization (LN) and Residual Connections which is formulated in Equation (6.10).

$$\hat{T} = \text{LayerNorm}(T + \text{MLP}(T)) \tag{6.10}$$

where MLP is a two-layer feedforward network with GELU activation [23]. This normalization step prevents gradient explosion and improves feature consistency.

The STFA module outputs the refined feature representation $\hat{T}$, which is then passed into the Reliability Module for further processing. This refined representation effectively

encodes spatio-temporal consistency, ensuring that ReliFusion maintains robust performance even under sensor malfunctions and dynamic scene variations.

6.2.3 Reliability Module

Reliable perception in multimodal systems requires the ability to detect and adapt to degraded or malfunctioning sensor inputs. Rather than assuming that sensors remain reliable under all conditions, the proposed framework explicitly estimates the trustworthiness of each modality and adjusts the fusion process accordingly. ReliFusion employs a Reliability Module that quantifies confidence in LiDAR and camera data using Cross-Modality Contrastive Learning (CMCL). This module generates dynamic confidence scores, enabling adaptive fusion and reducing the impact of corrupted sensor inputs.

6.2.3.1 Cross-Modality Contrastive Learning (CMCL)

To ensure consistent feature alignment, CMCL maps LiDAR and camera features into a shared embedding space, distinguishing between reliable and corrupted inputs. As formulated in Equation (6.11), the BEV-transformed camera features $F_{\text{Camera-BEV}}$ and LiDAR features F_{LiDAR} are projected using multi-layer perceptrons (MLPs):

$$z_{\text{LiDAR}} = \text{MLP}_{\text{LiDAR}}(F_{\text{LiDAR}}), \quad z_{\text{Camera}} = \text{MLP}_{\text{Camera}}(F_{\text{Camera-BEV}}) \tag{6.11}$$

Positive feature pairs (unaltered data) are encouraged to remain close in the embedding space, while negative pairs (corrupted data) are pushed apart using the contrastive loss function in Equation (6.12).

$$L_{\text{contrast}} = -\log \frac{\exp(\text{sim}(z_{\text{LiDAR}}, z_{\text{Camera}})/\tau)}{\sum_{i=1}^{K} \exp\Big(\text{sim}(z_{\text{LiDAR}}, z_{\text{Camera}}^{(i)})/\tau\Big)} \tag{6.12}$$

where $\text{sim}(z_{\text{LiDAR}}, z_{\text{Camera}})$ denotes the cosine similarity, τ is a temperature parameter, and K represents the batch size. This contrastive learning process ensures that features from both modalities remain aligned even under sensor degradation.

6.2.3.2 Reliability Scoring

The Reliability Module assigns confidence scores to each modality based on the quality of its embedding. The confidence scores for LiDAR and camera are computed as:

$$C_{\text{LiDAR}} = \sigma(W_{\text{LiDAR}} z_{\text{LiDAR}} + b_{\text{LiDAR}}) \tag{6.13}$$

$$C_{\text{Camera}} = \sigma(W_{\text{Camera}} z_{\text{Camera}} + b_{\text{Camera}}) \tag{6.14}$$

where W and b are learnable parameters, and σ represents the sigmoid activation function. These confidence scores, C_{LiDAR} and C_{Camera}, are used in the next stage to dynamically adjust fusion weights, ensuring that unreliable sensor data contributes less to the final object detection.

The confidence scores C_{LiDAR} and C_{Camera} directly modulate the feature representations prior to and during cross-attention fusion. Let $F_{\text{LiDAR}} \in \mathbb{R}^{C \times H \times W}$ and $F_{\text{Camera-BEV}} \in \mathbb{R}^{C \times H \times W}$ denote the modality-specific BEV features.

Thus, unreliable modalities are explicitly attenuated in magnitude before fusion. These scaled features are subsequently used as inputs to the mutual cross-attention mechanism, ensuring that modality contributions are proportionally adjusted according to their predicted reliability.

Generalization to Unseen Degradations. It is important to note that the Reliability Module does not explicitly learn degradation categories (e.g., fog, noise, occlusion). Instead, it learns to estimate modality confidence based on feature-level consistency in a shared embedding space. Because the confidence predictor operates on intermediate

BEV feature representations rather than raw sensor inputs, it captures distribution-level feature quality signals instead of degradation-specific artifacts.

Consequently, when encountering unseen perturbations that similarly distort feature statistics, the module can adaptively down-weight unreliable modalities based on learned feature alignment and embedding consistency. This design promotes robustness and generalization beyond the specific degradation types used during training.

Model-Level vs System-Level Reliability. It is important to distinguish between the notion of reliability used in this thesis and formal system-level reliability in safety-critical engineering.

In this work, reliability refers to *model-level reliability*, defined as the estimated trustworthiness of modality-specific feature representations during inference. The predicted confidence scores reflect the model's internal assessment of sensor feature quality under varying environmental and degradation conditions. This notion is data-driven and learned implicitly through contrastive alignment and detection supervision.

In contrast, *system-level reliability* in safety engineering is typically defined probabilistically, for example as the probability that a system performs its intended function under stated conditions for a specified period of time. Such definitions are common in automotive functional safety standards and involve formal failure rate modeling, redundancy analysis, and hazard assessment.

The reliability module proposed here does not estimate formal failure probabilities, nor does it replace safety certification processes. Instead, it provides an adaptive weighting mechanism within the perception model that mitigates the impact of degraded sensor inputs. While some experimental setups (e.g., sensor corruption analysis) implicitly resemble failure modeling, they remain at the perception-model level rather than full system reliability analysis.

6.2.4 Confidence-Weighted Mutual Cross-Attention (CW-MCA)

To ensure robust and adaptive sensor fusion, ReliFusion incorporates the Confidence-Weighted Mutual Cross-Attention (CW-MCA) module. This module dynamically adjusts the contribution of each modality based on the confidence scores computed by the Reliability Module, allowing the network to prioritize reliable sensor inputs while suppressing noisy or degraded signals.

6.2.4.1 Confidence-Weighted Feature Representation

The computed confidence scores C_{LiDAR} and C_{Camera} are used to weight the feature representations before fusion. Given the LiDAR feature F_{LiDAR} and the BEV-transformed camera feature $F_{\text{Camera-BEV}}$, the confidence-weighted features are computed as:

$$F'_{\text{LiDAR}} = C_{\text{LiDAR}} \cdot F_{\text{LiDAR}} \tag{6.15}$$

$$F'_{\text{Camera}} = C_{\text{Camera}} \cdot F_{\text{Camera-BEV}} \tag{6.16}$$

where the confidence scores act as adaptive scaling factors that enhance or suppress each modality's contribution based on reliability.

6.2.4.2 Mutual Cross-Attention Mechanism

To effectively integrate the LiDAR and camera features, CW-MCA applies a mutual cross-attention mechanism, where each modality attends to the other to capture complementary information. The cross-attention operations are defined as:

$$F_{\text{L}\to\text{C}} = C_{\text{LiDAR}} \cdot \text{Softmax}\left(\frac{Q_{\text{C}} K_{\text{L}}^{\top}}{\sqrt{d_k}}\right) V_{\text{L}} \tag{6.17}$$

$$F_{\mathrm{C}\to\mathrm{L}} = C_{\mathrm{Camera}} \cdot \mathrm{Softmax}\left(\frac{Q_{\mathrm{L}} K_{\mathrm{C}}^{\top}}{\sqrt{d_k}}\right) V_{\mathrm{C}} \tag{6.18}$$

where: Q, K, V denote the query, key, and value representations for each modality. d_k is the dimensionality of the key vectors. $F_{\mathrm{L}\to\mathrm{C}}$ represents LiDAR features attending to camera data. $F_{\mathrm{C}\to\mathrm{L}}$ represents camera features attending to LiDAR data.

These operations allow each modality to refine its features by incorporating information from the other modality, weighted by their respective confidence scores.

6.2.4.3 Final Feature Fusion

The final fused feature representation is obtained by combining the cross-attended feature maps:

$$F_{\mathrm{fused}} = F_{\mathrm{L}\to\mathrm{C}} + F_{\mathrm{C}\to\mathrm{L}} \tag{6.19}$$

This ensures that information from both LiDAR and camera modalities is optimally integrated while accounting for reliability.

6.2.4.4 Integration with the Detection Head

The fused BEV representation F_{fused} is passed into the TransFusion detection head [1], which is an anchor-free, center-based module for 3D object detection. The head produces heatmaps to indicate the most likely object centers on the BEV grid, together with regression branches that estimate the object's size, orientation, velocity, and height. Final 3D boxes are then formed by taking the strongest responses from the heatmaps and refining them with the regression outputs. This approach avoids the complexity of using predefined anchors and allows for stable estimation of object orientation.

By combining F_{fused} with the TransFusion head, ReliFusion leverages both adaptive multimodal fusion and an efficient detection architecture, enabling accurate and reliable object detection in BEV space while preserving real-time inference capability.

6.2.5 Architectural Simplification and Tiered Deployment Strategies

While the full ReliFusion architecture provides adaptive robustness through spatio-temporal aggregation and reliability-driven cross-attention, safety-critical deployment scenarios may require reduced latency under strict timing constraints. In such cases, architectural simplification strategies can be employed without causing catastrophic perception failure.

The reliability module and CW-MCA mechanism are designed as adaptive enhancement layers on top of a functional mid-level fusion backbone. Therefore, several fallback configurations are feasible:

- **Static Fusion Mode:** The confidence scores can be fixed or disabled, reverting the model to a standard mid-level fusion strategy with equal modality weighting.
- **Single-Modality Fallback:** If a modality becomes severely degraded or unavailable, the architecture can operate in LiDAR-only or camera-only mode using the shared BEV backbone and detection head.
- **Reduced Temporal Window:** The STFA temporal depth can be shortened or disabled, reducing attention complexity while maintaining spatial fusion.
- **Lightweight Confidence Head:** The contrastive embedding dimension and confidence predictor can be simplified to reduce computational overhead.

These tiered configurations enable graceful performance degradation rather than abrupt failure. Importantly, the reliability module does not introduce a single point

of failure; instead, it enhances fusion adaptability. When disabled, the system reverts to a conventional mid-level fusion architecture similar to Phase 1 (TransfuseNet), ensuring functional continuity.

This modular design supports deployment-specific trade-offs between robustness and latency, aligning the architecture with real-world automotive perception requirements.

6.3 Summary

This chapter presented the design and implementation of ReliFusion, a reliability-driven sensor fusion framework for LiDAR-camera object detection. The methodology incorporates spatio-temporal modeling, confidence-based weighting, and multimodal fusion to improve detection performance under varying sensor conditions. The next chapter provides a detailed evaluation of ReliFusion, analyzing its performance across different experimental settings, comparing it with existing methods, and conducting ablation studies to assess the contribution of individual components.

Chapter 7

Experiments on ReliFusion

This chapter provides comprehensive experimental validation of ReliFusion. Experiments are conducted using the nuScenes dataset [5], evaluating ReliFusion under diverse scenarios, including standard conditions, sensor malfunctions, and occlusions. Through comparative studies against state-of-the-art approaches, detailed ablation studies, and qualitative analysis, we demonstrate ReliFusion's robustness and adaptability in realistic autonomous driving scenarios.

7.1 Dataset and Metrics

ReliFusion is evaluated on the nuScenes dataset [5], a large-scale multimodal benchmark for autonomous driving. A detailed description of the dataset, including sensor configuration, object classes, and annotation statistics, is provided in Section 4.2. For evaluation, we adopt the standard nuScenes metrics: mean Average Precision (mAP) and the nuScenes Detection Score (NDS), as defined in Section 2.5.

For completeness and reproducibility, nuScenes mAP is computed using center-distance thresholds of 0.5m, 1.0m, 2.0m, and 4.0m rather than fixed IoU thresholds,

as defined in the official benchmark protocol. A detection is considered a true positive if its center distance to a ground-truth object falls within the specified threshold for its class.

The NDS metric combines mAP with five additional error terms: translation error, scale error, orientation error, velocity error, and attribute error. Each component is normalized and weighted according to the official nuScenes definition.

All experiments are conducted on the nuScenes validation and test splits using the official evaluation script without modification. Class-specific thresholds and difficulty levels follow the standard benchmark configuration. No additional filtering or post-processing beyond non-maximum suppression (NMS) is applied.

7.2 Data Preprocessing

LiDAR point clouds are voxelized to create a structured representation while preserving spatial granularity. Following the setup in CenterPoint [80], the nuScenes coordinate system is defined such that the x-axis points forward, the y-axis points to the left, and the z-axis points upward in the ego-vehicle frame. For all experiments, we restrict the detection range to $[-51.2m, 51.2m]$ along x and y, and $[-5m, 3m]$ along z, consistent with the official nuScenes benchmark.

The voxel grid is constructed with resolution $(0.075m, 0.075m, 0.2m)$ along (x, y, z), respectively. This quantization step aggregates all points falling into the same voxel, thereby downsampling the raw point cloud while retaining sufficient geometric detail. Compared to the $0.1m \times 0.1m \times 0.2m$ voxel size used in CenterPoint, the slightly finer $(0.075m, 0.075m, 0.2m)$ resolution adopted here provides denser spatial encoding and improves small-object detection performance. The voxelized tensors are subsequently processed by VoxelNet [87] as 3D backbone , which encodes the quantized point cloud into high-level feature representations.

For the image stream, multi-view RGB images are resized to a fixed resolution of 448×800 pixels before being passed through the ConvMixer [64] backbone for feature extraction.

7.3 Training Strategy

To fully optimize ReliFusion for robust 3D object detection, a multi-stage training strategy is employed. Each module is pre-trained independently to refine its respective function before the entire network undergoes end-to-end fine-tuning. The training process is designed to enhance multimodal feature alignment, spatio-temporal consistency, and adaptive fusion.

7.3.1 Pre-Training of Individual Modules

In the first stage, three key modules are pre-trained separately:

Contrastive Module: This module is trained using contrastive learning to align LiDAR and camera embeddings. The objective is to maximize the similarity of positive feature pairs while minimizing that of negative pairs. The contrastive loss is defined as Equation (7.1).

$$L_{\text{contrast}} = -\log \frac{\exp(\text{sim}(z_{\text{LiDAR}}, z_{\text{Camera}})/\tau)}{\sum_{i=1}^{K} \exp\Big(\text{sim}(z_{\text{LiDAR}}, z_{\text{Camera}}^{(i)})/\tau\Big)} \tag{7.1}$$

where $\text{sim}(z_{\text{LiDAR}}, z_{\text{Camera}})$ represents cosine similarity, $\tau = 0.07$ is the temperature parameter, and K is the batch size. The embedding size is set to 128 (as introduced in Chapter 6), representing the dimensionality of the shared feature embedding space.

Confidence Module: This module is trained using a regression loss in Equation (7.2) to assign reliability scores for sensor inputs:

$$L_{\text{conf}} = \sum_{i} \left(C_{\text{LiDAR}}^{(i)} - \hat{C}_{\text{LiDAR}}^{(i)} \right)^2 + \left(C_{\text{Camera}}^{(i)} - \hat{C}_{\text{Camera}}^{(i)} \right)^2 \tag{7.2}$$

where $C_{\text{LiDAR}}^{(i)}$ and $C_{\text{Camera}}^{(i)}$ denote the ground-truth reliability scores for the i-th training sample, and $\hat{C}^{(i)}$ represents the corresponding predicted scores. The index i runs over all sensor inputs in a mini-batch. In practice, ground-truth reliability scores are derived by simulating sensor degradations. For LiDAR, this includes point dropping and restricted field-of-view; for cameras, this includes pixel masking and occlusion. Reliability labels are assigned proportional to the severity of the corruption, with clean sensor inputs assigned full reliability ($C = 1.0$). This allows the confidence module to learn to predict modality reliability in both nominal and degraded conditions.

Spatio-Temporal Feature Aggregation (STFA) Module: Pre-trained to enforce temporal consistency across frames, the STFA module minimizes the difference between temporally adjacent features aligned by ego-motion. The temporal consistency loss is defined as:

$$L_{\text{temp}} = \frac{1}{T-1} \sum_{t=1}^{T-1} \| f_{t+1} - \hat{f}_t \|_2^2 \tag{7.3}$$

where f_t represents features at frame t, $\hat{f}_t$ is the corresponding feature from the previous frame after ego-motion alignment, and T is the total number of frames in the temporal sequence. Ego-motion alignment compensates for the vehicle's movement using pose estimates provided by the dataset, ensuring that features from different frames are mapped into a consistent coordinate space. This encourages stable temporal representations and improves the model's ability to track object motion across frames.

The architectural design of the CMCL and confidence modules follows a lightweight projection-head paradigm to avoid introducing excessive computational overhead. The CMCL module employs a shared embedding dimension of $d = 128$, chosen based on the hyperparameter study in Table 7.8, which demonstrated optimal trade-off between cross-modal alignment capacity and stability. Larger embeddings ($d = 256$) did not

improve accuracy, while smaller embeddings ($d = 64$) reduced alignment quality.

The confidence module is implemented as a shallow regression head operating on BEV-level fused features rather than raw modality features. This design ensures that reliability is estimated based on semantic-level information rather than low-level noise patterns, improving generalization to unseen degradation scenarios. The module predicts continuous reliability values in $[0, 1]$, enabling smooth adaptive weighting rather than hard modality switching.

7.3.2 Training of Sensor-Specific Feature Extractors

In the second stage, the individual feature extraction streams for the image and LiDAR modalities are trained separately. The ConvMixer network [64] is used as the 2D backbone for extracting features from the camera input, while a voxel-based backbone, similar to VoxelNet [87], processes LiDAR point clouds. These backbone networks are optimized independently to ensure efficient feature encoding.

7.3.3 End-to-End Fine-Tuning with Multi-Task Learning

In the final stage, the entire network is fine-tuned end-to-end using a multi-task loss function. As shown in Equation (7.4), this objective function combines multiple loss components to optimize the detection performance:

$$L_{\text{total}} = \lambda_1 L_{\text{det}} + \lambda_2 L_{\text{contrast}} + \lambda_3 L_{\text{temp}} + \lambda_4 L_{\text{conf}} \tag{7.4}$$

where: L_{det} is the object detection loss. L_{contrast} is the contrastive loss for cross-modality feature alignment. L_{temp} is the temporal consistency loss to ensure frame stability. L_{conf} penalizes incorrect confidence score predictions.

The object detection loss L_{det} follows the CenterPoint [80] formulation, combining focal loss for classification and L1 regression for bounding box parameters (center lo-

cation, size, and orientation). This ensures both accurate localization and robust class predictions.

The weighting coefficients are empirically set as $\lambda_1 = 1.0$, $\lambda_2 = 0.1$, $\lambda_3 = 0.2$, and $\lambda_4 = 0.05$, balancing the trade-off between detection accuracy, feature alignment, and robustness.

7.3.4 Training Protocol Summary

For clarity, the complete training pipeline of ReliFusion is summarized below in sequential order.

Stage 1: Module-Level Pre-Training. The CMCL (contrastive alignment), Confidence, and STFA modules are pre-trained independently for 15 epochs using the Adam optimizer with learning rate 1×10^{-4} and weight decay 1×10^{-5}. During this stage:

- Backbone feature extractors are kept trainable.
- Each module is optimized using its dedicated loss function (L_{contrast}, L_{conf}, or L_{temp}).
- Synthetic sensor degradations are applied to generate supervision signals for reliability learning.

Stage 2: Backbone Optimization. The feature extractors are trained independently to ensure stable modality-specific feature representations before multimodal coupling. This reduces optimization interference during early fusion stages.

Stage 3: End-to-End Fine-Tuning. All modules are jointly optimized for 5 additional epochs using the multi-task objective in Eq. 7.4. In this phase:

- All parameters are unfrozen.

- The total loss combines detection, contrastive alignment, temporal consistency, and confidence estimation.
- Weighting coefficients are fixed as $\lambda_1 = 1.0$, $\lambda_2 = 0.1$, $\lambda_3 = 0.2$, and $\lambda_4 = 0.05$.

Optimization Stability. Gradient clipping is applied to prevent instability during joint optimization. All experiments use a batch size of 16 and identical data augmentation policies to ensure consistent comparison across baselines.

This staged protocol ensures that cross-modal alignment and reliability estimation are learned prior to full multimodal fusion, improving convergence stability and preventing premature dominance of one modality during training.

7.4 Implementation Details and Training Configuration

For feature extraction, ConvMixer [64] serves as the 2D backbone for extracting semantic image features, and VoxelNet [87] is adopted as the 3D backbone for LiDAR processing. Both backbones independently encode their modality-specific data before transforming these features into a unified BEV representation via Lift-Splat-Shoot (LSS) operation [49]. The resulting aligned BEV features from both modalities are then further processed by subsequent ReliFusion modules, ensuring effective multimodal fusion.

ReliFusion is implemented in PyTorch with MMDetection3D [8], a modular, open-source toolkit that provides configurations, data loaders, and benchmarks for 3D object detection. The model is trained using the Adam optimizer [28] with an initial learning rate of 1×10^{-4} and a weight decay of 1×10^{-5}. The training process consists of 15 epochs for pre-training individual feature extractors, followed by 5 additional epochs for end-to-end fine-tuning, with a batch size of 16.

7.4.1 Baseline Selection Criteria

To ensure fair and meaningful experimental comparison, baseline methods were selected based on the following criteria:

1. **State-of-the-art performance on nuScenes:** Methods demonstrating competitive performance on the nuScenes benchmark were prioritized to ensure relevance and rigor in comparison.

2. **LiDAR-camera multimodal fusion capability:** Since ReliFusion focuses on multimodal BEV fusion, selected baselines include methods that integrate both LiDAR and camera modalities, enabling direct architectural comparison.

3. **Representative fusion strategies:** Baselines were chosen to reflect different fusion paradigms, including early fusion, mid-level fusion, and transformer-based fusion mechanisms, allowing evaluation across diverse design philosophies.

4. **Open-source availability and reproducibility:** Only methods with publicly available implementations or clearly documented configurations were considered, ensuring reproducibility and consistent evaluation settings.

5. **Computational comparability:** Methods with similar backbone capacity and BEV resolution were prioritized to avoid unfair advantages arising from substantially larger model sizes.

In addition to external baselines, TransfuseNet is included to provide an internal reference point, enabling direct evaluation of the improvements introduced by reliability-aware fusion and spatio-temporal modeling in ReliFusion.

7.5 State-of-the-Art Comparison

Although the primary objective of ReliFusion is to improve robustness under challenging conditions, it also demonstrates superior performance on clean datasets. As shown in Table 7.1, ReliFusion achieves an mAP of 70.6% and an NDS of 73.2%, outperforming existing SOTA methods such as BEVFusion [35] and TransFusion [1].

Method	L	C	mAP	NDS	Car	Truck	C.V.	Bus	Trailer	Barrier	Motor.	Bike	Ped.	T.C.
BEVFormer [34]	-	✓	47.2	57.1	66.8	38.2	22.3	34.5	38.8	61.6	47.1	39.7	53.0	69.9
BEVDet [24]	-	✓	42.1	48.5	63.9	34.7	16.0	35.3	35.2	61.1	44.5	29.4	40.7	59.9
DETR3D [72]	-	✓	34.9	43.4	53.2	27.4	10.3	21.7	29.4	50.9	35.1	25.7	39.4	56.1
CenterPoint [80]	✓	-	56.8	65.0	82.5	50.0	17.9	59.5	53.0	67.1	56.3	26.0	80.6	74.7
TransFusion-L [1]	✓	-	64.3	68.4	85.1	55.5	27.2	64.9	57.9	**77.1**	67.3	43.1	84.5	80.5
FUTR3D [10]	✓	✓	62.4	67.3	85.0	59.4	24.5	69.7	40.7	62.8	71.7	**61.8**	80.6	68.6
MVX-Net [60]	✓	✓	61.0	66.1	82.3	50.7	21.4	60.3	56.2	66.8	66.9	44.1	82.7	78.9
PointAugmenting [69]	✓	✓	46.9	55.6	68.8	36.9	6.5	45.4	41.6	52.3	56.6	29.0	67.3	64.7
TransFusion [1]	✓	✓	66.9	70.9	86.5	58.8	30.4	65.6	58.4	76.4	71.9	49.3	86.0	**85.7**
BEVfusion [35]	✓	✓	67.9	71.0	87.4	59.6	32.1	67.8	61.2	76.8	71.4	50.1	88.0	84.2
CMT [76]	✓	✓	69.3	72.1	86.3	60.4	**36.8**	71.2	61.8	73.4	78.4	57.1	85.6	82.9
ReliFusion (ours)	✓	✓	**70.6**	**73.2**	**88.1**	**61.7**	35.9	**72.9**	**63.0**	75.8	**79.5**	57.4	**89.3**	83.5

Table 7.1: Evaluation results on nuScenes dataset. We evaluated ReliFusion against the SOTA results on the test set. 'L' and 'C' represents LiDAR and Camera, respectively. 'C.V', 'Ped', and 'T.C' stand for construction vehicle, pedestrian, and traffic cone, respectively. The best results appear in bold.

7.5.1 Robustness Experiments

To evaluate the reliability of ReliFusion under real-world sensor degradation, robustness experiments are conducted by simulating LiDAR and camera failures. These tests assess the model's ability to maintain detection accuracy when sensor inputs are corrupted or missing.

7.5.1.1 Robustness Against LiDAR Degradations

ReliFusion is evaluated under conditions where LiDAR data is progressively reduced to assess its capacity to maintain detection accuracy. Two representative degradation scenarios are considered. In the first scenario, **Limited Field of View**, point cloud data is progressively restricted to narrower angular ranges to simulate partial LiDAR occlusions. The considered ranges are: $-\pi/2$ to $\pi/2$ (half field of view), $-\pi/3$ to $\pi/3$ (narrower field of view), and 0 to 0 (complete removal of LiDAR data). This represents situations where LiDAR beams are blocked or the effective field of view is reduced by environmental or mechanical factors. In the second scenario, **Object Detection Failures**, Within each object bounding box, 50% of LiDAR points are randomly discarded to simulate reduced point density caused by poor reflectivity, weather-induced attenuation, or sensor noise. The random dropout is performed by uniformly sampling points inside the bounding box and removing half of them. To ensure reproducibility, the random seed is fixed during all experiments. This emulates realistic cases where LiDAR returns are partially missing rather than entirely absent.

The results, summarized in Table 7.2, demonstrate the robustness of ReliFusion in comparison to state-of-the-art methods. In the Limited Field of View scenario, LiDAR-only models such as CenterPoint [80] exhibit a significant decline in performance due to their exclusive reliance on LiDAR data. Non-BEV methods, such as TransFusion [1], also experience substantial drops in mAP and NDS, as they heavily depend on LiDAR features for region proposals or key points, using image data solely as auxiliary input.

While BEV-based methods attempt to maximize the use of camera information by fusing data within the same spatial domain, the contribution from camera data alone often proves insufficient for robust results. For example, models like BEVFusion [35] address some of these challenges by integrating complementary information from camera inputs, resulting in improved mAP scores of 46.4%, 41.5%, and 50.3% across various scenarios. However, these methods do not fully account for the dynamic reliability of each modality, particularly under conditions of sensor malfunctions.

ReliFusion improves upon this by dynamically determining the contribution of each modality, thereby enhancing overall robustness in the presence of sensor malfunctions. This advantage is especially noticeable when LiDAR data is unavailable, emphasizing ReliFusion's effectiveness as a state-of-the-art solution in such challenging conditions.

Method	Modality	Clean	Limited LiDAR FOV			LiDAR Object Failure
			$(-\pi/2, \pi/2)$	$(-\pi/3, \pi/3)$	$(-0, 0)$	50% Drop
CenterPoint [80]	L	56.8/65.0	23.5/47.7	15.6/43.0	0/0	28.4/48.5
PointAugmenting [69]	LC	46.9/55.6	19.5/41.2	13.3/37.7	0/0	21.3/39.4
MVX-Net [60]	LC	61.0/66.1	26.0/47.8	17.6/43.1	0/0	34.0/51.1
TransFusion [1]	LC	66.9/70.9	29.3/51.4	20.3/45.8	0/0	34.6/53.6
BEVFusion [35]	LC	67.9/71.0	46.4/55.8	41.5/50.8	12.4/17.1	50.3/57.6
ReliFusion (ours)	LC	**70.6/73.2**	**52.4/59.6**	**44.9/54.8**	**24.6/39.7**	**53.1/60.6**

Table 7.2: Comparison of SOTA methods under limited LiDAR FOV and object failure scenarios, with mAP and NDS metrics provided.

Qualitative examples. Figure 7.1 shows three LiDAR–degradation settings used here: FOV restrictions $[-\pi/2, \pi/2]$ and $[-\pi/3, \pi/3]$, and object failure via 50% point drop inside ground-truth boxes. In all cases, BEVFusion tends to miss targets that fall outside the active LiDAR field of view or are represented by very sparse returns, whereas ReliFusion continues to localize them. This is because the CMCL reliability head assigns higher confidence to the image stream when LiDAR evidence weakens, and CW-MCA uses these scores to up-weight camera BEV features during fusion; consequently, objects not covered by the restricted LiDAR sweep—or degraded by point dropout—remain detectable, with box extent and heading better preserved.

7.5.1.2 Robustness Against Camera Failures

To evaluate ReliFusion's ability to maintain performance under degraded camera conditions, two experiments were conducted.

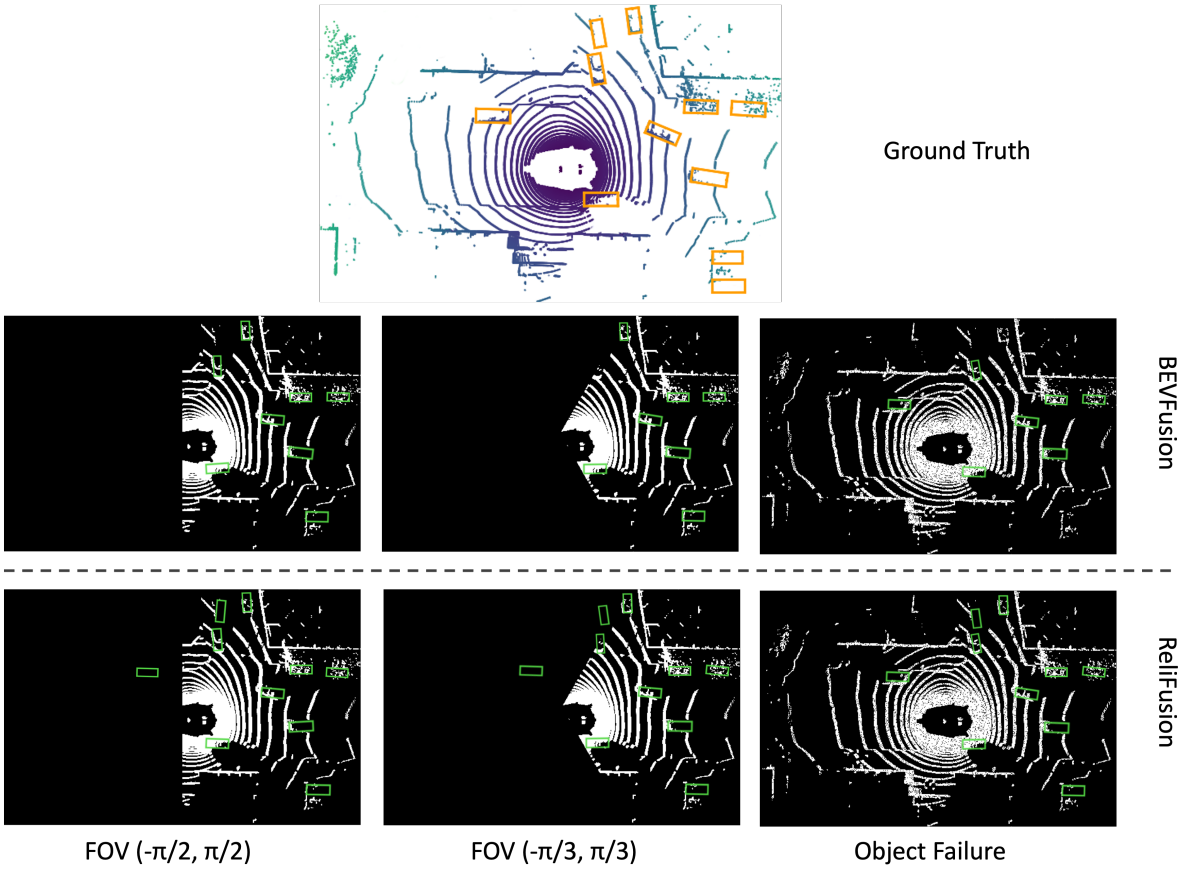

Figure 7.1: Qualitative detection results of BEVFusion and ReliFusion under LiDAR malfunctions scenarios. Clearly, BEVFusion struggles when LiDAR input is unavailable, whereas ReliFusion relies on camera to compensate and detect these objects. Green and Orange bounding boxes are true positive detection and ground truth, respectively.

Camera Failure: In the first experiment, the front camera image is removed entirely to simulate the disconnection or failure of the camera. The remaining five cameras continue to provide input for feature extraction.

Object Occlusion: The second experiment involves masking 50% of the image pixels within object bounding boxes. This simulates real-world occlusions that might be caused by obstacles or adverse weather conditions.

As showin in Table 7.3 ReliFusion demonstrates superior performance across all camera degradation scenarios. When the front camera is removed, ReliFusion achieves an mAP of 68.3%, surpassing BEVFusion by 2.4%. Under the 50% occlusion condition, ReliFusion maintains stable detection performance with only a 2.8% mAP drop, showcasing its robustness in handling missing or partially corrupted visual inputs.

Method	Modality	Clean	Object Failure		Object Occlusion
			Missing F	Preserve F	50% Occlusion
DETR3D [72]	C	34.9/43.4	25.8/39.2	3.3/20.5	14.3/29.0
PointAugmenting [69]	LC	46.9/55.6	42.4/53.0	31.6/46.5	40.7/52.2
MVX-Net [60]	LC	61.0/66.1	47.8.0/59.4	17.5/41.7	45.5/57.6
TransFusion [1]	LC	66.9/70.9	65.3/70.1	64.4/69.3	65.5/70.0
BEVFusion [35]	LC	67.9/71.0	65.9/70.7	65.1/69.9	65.9/70.1
ReliFusion (ours)	LC	**70.6/73.2**	**68.3/71.3**	**65.9/70.4**	**67.8/70.6**

Table 7.3: Comparison of SOTA methods under camera failure and object occlusion scenarios, with mAP and NDS metrics provided.

ReliFusion significantly enhances robustness against sensor failures, outperforming existing methods in scenarios with missing or degraded LiDAR and camera inputs. The reliability-driven fusion mechanism dynamically adapts to sensor reliability, ensuring accurate detections even under adverse conditions.

7.5.1.3 Robustness under Environmental Conditions

Autonomous driving systems must operate reliably across diverse weather and illumination conditions, where both LiDAR and camera sensors are susceptible to modality-

specific degradations. LiDAR signals are affected by atmospheric disturbances such as rain, fog, or snow, which generate spurious reflections and reduce point density. Cameras, by contrast, are particularly sensitive to low-light and high-glare conditions that impair semantic feature extraction. To evaluate the robustness of ReliFusion under such challenges, we conduct experiments on the *Sunny*, *Rainy*, *Day*, and *Night* subsets of the nuScenes dataset [5], consisting of 5051, 968, 5417, and 602 frames, respectively.

The results in Table 7.4 indicate that ReliFusion achieves consistently higher detection accuracy across all subsets compared to representative baselines. Under rainy conditions, where LiDAR degradation is most pronounced, ReliFusion attains an mAP/NDS of 70.9/72.9, exceeding BEVFusion by +1.8/+1.3. This improvement reflects the contribution of the reliability module, which adaptively increases the weight of camera features when LiDAR reliability declines. In nighttime scenarios, where camera-based methods deteriorate sharply due to poor visibility, ReliFusion attains 56.4/63.8, representing an absolute gain of +11.9/+9.3 over BEVFusion. Although all models exhibit reduced performance at night due to the limited size and distribution shift of the nighttime subset, ReliFusion demonstrates the smallest relative decrease, underscoring its ability to maintain balanced cross-modality fusion. These findings confirm that confidence-guided fusion is an effective mechanism for mitigating modality imbalance in safety-critical perception tasks.

Notably, ReliFusion exhibits only a 20.4% relative decline in mAP from daytime to nighttime conditions, compared to 34.9% for BEVFusion, 36.3% for TransFusion, 33.6% for CenterPoint, and 34.3% for DETR3D. This quantitatively confirms that ReliFusion maintains substantially higher robustness under adverse illumination, reducing the relative night-time degradation by more than one third compared to leading baselines.

To improve interpretability, we analyze the predicted reliability scores of the LiDAR and camera streams across different environmental subsets. Figure X illustrates the average predicted reliability values under Sunny, Rainy, Day, and Night conditions.

Under rainy conditions, the average predicted LiDAR reliability decreases relative

Method	Sunny	Rainy	Day	Night
CenterPoint [80]	60.5/67.4	57.9/65.5	60.8/65.9	40.4/49.7
DETR3D [72]	36.6/47.0	39.9/50.81	43.5/51.7	28.6/39.8
TransFusion [1]	67.2/71.2	67.8/71.0	67.6/71.3	43.1/51.2
BEVFusion [35]	68.3/71.3	69.1/71.6	68.4/71.6	44.5/54.5
ReliFusion (ours)	**70.1/72.7**	**70.9/72.9**	**70.8/72.6**	**56.4/63.8**

Table 7.4: Comparison of mAP/NDS under different weather and lighting conditions on nuScenes.

to sunny conditions, reflecting the degradation of LiDAR returns due to atmospheric scattering and reduced point density. Conversely, camera reliability remains relatively stable in rain but declines significantly at night, where illumination is limited.

These trends confirm that the reliability module learns modality-specific degradation patterns without explicit supervision on environmental labels. Importantly, reliability values correlate with observed detection performance trends in Table 7.4, demonstrating that the module does not assign arbitrary weights but reflects physically meaningful sensor quality variations.

This interpretability analysis validates that reliability prediction generalizes beyond synthetic corruption scenarios and adapts consistently to real environmental changes.

7.5.2 Ablation Study

To assess the importance of ReliFusion's individual components in addressing sensor reliability challenges, we conducted a series of ablation studies. These experiments were designed to evaluate the contribution of the STFA, CW-MCA, and Reliability modules to achieving robust detection performance under sensor degradation conditions.

The model was tested with selective removals of these modules, and the results are summarized in Table 7.5. In the case of removing the STFA module, the model's performance decreased by 9.1 percent in mAP, indicating the critical role of spatio-temporal modeling in ensuring stable object detection. Without STFA, the model is

unable to effectively capture dynamic object motion across frames, which significantly impacts its overall performance.

Similarly, when the CW-MCA module was removed, the model experienced a 5.5 percent drop in mAP. This demonstrates that adaptive fusion between LiDAR and camera data is essential for handling varying sensor reliability. The CW-MCA module allows ReliFusion to effectively adjust the weight of each modality based on its reliability, thus improving fusion performance, especially in the presence of degraded sensor data.

The exclusion of the Reliability Module led to consistent performance drops across all LiDAR degradation scenarios. When the LiDAR field of view was restricted to $[-\pi/2, \pi/2]$, performance declined by 3.0 mAP, and when further restricted to $[-\pi/3, \pi/3]$, the reduction increased to 4.6 mAP. In the extreme case of complete LiDAR removal ($[0, 0]$), the absence of the module resulted in a gap of 4.2 mAP. For the object point dropout scenario, performance decreased by 2.8 mAP. These results demonstrate that the Reliability Module systematically enhances robustness by dynamically weighting sensor contributions, enabling ReliFusion to maintain stable detection accuracy even under severely degraded LiDAR conditions.

To further evaluate the effectiveness of the CW-MCA mechanism, we compared it to other fusion strategies, including additive fusion and cross-attention mechanisms. The results, presented in Table 7.7, show that additive fusion performed the worst, suggesting that a simple combination of features does not fully leverage the potential of multimodal data. Cross-attention, particularly when LiDAR was used as the query, showed an improvement, but it was still outperformed by CW-MCA. The CW-MCA method, by dynamically weighting the contributions of each modality based on sensor reliability, consistently achieved the best results.

In summary, the ablation study confirms that each module in ReliFusion contributes significantly to the model's overall performance. The STFA enhances spatio-temporal consistency, CW-MCA improves multimodal fusion, and the Reliability Module ensures adaptive weighting of sensor inputs. The effectiveness of CW-MCA is further validated

by comparisons with other fusion strategies, where it consistently outperforms both additive and standard cross-attention mechanisms.

ReliFusion Modules			**Limited LiDAR FOV**			**LiDAR Object Failure**
STFA	CW-MCA	Reliability	$(-\pi/2, \pi/2)$	$(-\pi/3, \pi/3)$	$(0, 0)$	50% Drop
-	-	-	33.3/44.2	25.4/37.4	5.1/22.3	38.6/49.8
✓	-	-	43.9/52.5	36.6/45.7	17.2/29.2	45.4/55.6
✓	✓	-	49.4/58.3	40.3/50.8	20.4/36.6	50.3/57.2
✓	✓	✓	**52.4/59.6**	**44.9/54.8**	**24.6/39.7**	**53.1/60.6**

Table 7.5: Ablation study of ReliFusion components (STFA, CW-MCA, and reliability modules) under limited LiDAR FOV and object failure scenarios, with mAP and NDS metrics reported.

7.5.2.1 Hyperparameter Ablation: Temporal Horizon and Embedding Size

Beyond module-level ablations, we study how temporal context (T) and embedding size (d) affect detection accuracy. We vary the STFA sequence length $T \in \{2, 4, 6\}$ and the CMCL embedding $d \in \{64, 128, 256\}$ while holding all other settings fixed. Table 7.8 reports mAP/NDS. Increasing T from 2 to 4 improves mAP/NDS by allowing the model to capture richer short-term temporal dynamics of object motion, while raising T to 6 yields only marginal gains. For the embedding, d=128 performs best; d=64 underfits cross-modal relations, whereas d=256 increases capacity without measurable accuracy gains.

Aggregation Method	mAP↑	NDS↑
without STFA	67.2	71.3
Spatio	69.3	72.8
Temporal	68.9	72.6
Spatio-Temporal	**70.6**	**73.2**

Table 7.6: Evaluating the impact of the STFA.

Fusion Method	mAP↑	NDS↑
Add	65.1	68.7
Cross/Image	66.6	70.1
Cross/LiDAR	67.4	71.3
MCA	68.3	72.5
CW-MCA	**70.6**	**73.2**

Table 7.7: Evaluating the impact of the CW-MCA.

Configuration	mAP	NDS
T=2, d=128	68.9	72.0
T=4, d=128	**70.6**	**73.2**
T=6, d=128	70.5	73.1
T=4, d=64	69.4	72.1
T=4, d=256	70.6	73.2

Table 7.8: Impact of sequence length T and CMCL embedding d on accuracy.

7.5.3 Runtime Efficiency and Latency Evaluation

We measure end-to-end inference latency and throughput under a controlled protocol on a single workstation (Intel Core i9-10900K, 10 cores/20 threads; 32 GB RAM; NVIDIA GeForce RTX 3090, 24 GB GDDR6X) running Ubuntu 20.04 with CUDA 11.3 and cuDNN 8.2. Unless otherwise stated, batch size = 1. We run 200 warm-up iterations to allow the system to reach steady state, then time N=1000 single-frame forwards. We report mean latency (ms) and frames per second (FPS), where FPS = 1000/latency(ms). Data loading and metric evaluation are excluded.

7.5.3.1 Computational Complexity

We report the number of learnable parameters (Params) and the computational cost per forward pass, expressed in billions of floating-point operations (GFLOPs), evaluated at the chosen input resolution and sequence length T. GFLOPs indicate the number of arithmetic operations per frame, independent of hardware, while Params reflect model capacity and memory footprint. For ReliFusion, computational cost arises mainly from the camera and LiDAR BEV feature encoders, the STFA, the CMCL heads, and the CW-MCA.

As shown in Table 7.9, ReliFusion achieves the best accuracy and the highest throughput even though its parameter count is higher than TransFusion. This is expected because runtime is governed primarily by FLOPs and execution efficiency, not by parameter count. ReliFusion's per-frame compute is lower than TransFusion ($\sim$540 vs.

Method	Params (M)	GFLOPs	FPS ↑	Latency (ms) ↓	mAP / NDS
TransFusion	27.8	600	6.1	163	66.9 / 70.9
BEVFusion	36	550	8.4	119	67.9 / 71.0
ReliFusion	**36.7**	**540**	**10.4**	**96**	**70.6 / 73.2**

Table 7.9: Efficiency vs. accuracy on the nuScenes.

∼600 GFLOPs) and comparable to BEVFusion (∼550 GFLOPs), which translates into higher FPS on the same hardware.

Concretely, ReliFusion concentrates compute in GPU-friendly BEV operations and lightweight fusion: (i) STFA aggregates across a short temporal window without dense image–BEV cross-attention; (ii) CW-MCA fuses modalities at the BEV level with a small number of tokens, avoiding expensive query-to-pixel attention; (iii) the Reliability/CMCL heads are compact. These design choices reduce sequential bottlenecks and memory traffic, yielding higher FPS at a similar (or lower) GFLOP budget, while the slightly larger parameter count improves representational capacity and contributes to the higher accuracy.

As summarized in Table 7.10, the perception budgets t_{perc} were calculated in Section 1.3 using the stopping-distance model and represent conservative upper bounds for each driving scenario. Since the measured inference latency of ReliFusion is $t_{\text{inf}} = 96\,\text{ms}$, which is lower than the budget in all cases (110/140/160 ms), the model meets the required perception budgets for highway, rural, and urban driving.

Scenario	t_{perc} (ms)	t_{inf} (ms)	Meets Budget?
Highway (130 km/h)	110	96	✓
Rural (80 km/h)	140	96	✓
Urban (50 km/h)	160	96	✓

Table 7.10: ReliFusion inference latency compared against perception budgets.

7.5.3.2 Latency Attribution per Module

For latency attribution, we take as baseline a multimodal detector consisting of the LiDAR and image feature extractors with additive fusion, but without any of ReliFusion's additional modules (no STFA, no CMCL, no CW-MCA). This baseline ensures a fair comparison, since both sensor streams are present. We then progressively enable each module on top of this baseline to quantify its incremental cost. Both forward (network-only) and pipeline latency (including voxelization and BEV projection) are reported.

As shown in Table 7.11, adding STFA (T=4) introduces a modest overhead of 4 ms, reflecting the cost of temporal aggregation across frames. The CMCL heads add only 3 ms, since they operate on compact BEV embeddings. Finally, CW-MCA contributes an additional 5 ms due to cross-modal attention, bringing the total pipeline latency to 96 ms. Overall, ReliFusion achieves higher robustness and accuracy with only a moderate increase over the multimodal baseline (84 ms $\rightarrow$ 96 ms), demonstrating that the additional modules are computationally efficient relative to their performance gain.

Configuration	Forward (ms)	Pipeline (ms)	Incremental Δ (ms)
Baseline: Image + LiDAR fusion only	64	84	–
+ STFA (T=4)	68	88	+4
+ CMCL (d=128)	71	91	+3
+ CW-MCA (full ReliFusion)	**76**	**96**	+5

Table 7.11: Latency breakdown (ms). Baseline includes both image and LiDAR feature extractors with additive fusion but no advanced modules. Each row adds one module cumulatively, and Δ denotes incremental pipeline overhead relative to the previous configuration.

7.5.4 Real-World Readiness: Meeting Safety-Critical Budgets

Using the measured inference latency from Table 7.9, we evaluate compliance with the sensor-limited, brake-limited, and steer-limited budgets defined in Section 1.3. ReliFusion with (T=4, d=128) satisfies these safety-critical perception budgets.

However, strict deployment under a 20 Hz LiDAR synchronization constraint (i.e., 50 ms perception window) represents an idealized upper bound assuming tightly coupled sensing–perception–planning pipelines. The measured 96 ms latency reported here corresponds to a research-grade implementation evaluated without low-level inference acceleration, kernel fusion, model pruning, or hardware-specific optimization.

It is important to distinguish between (i) theoretical perception cycle targets used in system-level motivation and (ii) measured inference time of a prototype implementation. In production autonomous systems, perception modules are typically executed asynchronously, pipelined across hardware accelerators, or deployed on automotive-grade SoCs with dedicated tensor cores. Under such optimized deployment settings, latency can be substantially reduced.

Therefore, the 50 ms target should be interpreted as a design objective aligned with high-speed sensing cycles rather than a strict constraint already satisfied by the current prototype. The reported 96 ms latency remains within the conservative perception budgets derived in Section 1.3 (110–160 ms depending on driving scenario), and further optimization strategies such as mixed-precision inference (FP16), TensorRT graph optimization, or architectural simplification could bridge the remaining gap.

7.6 Summary

This chapter showed that ReliFusion achieves 70.6 mAP and 73.2 NDS on the nuScenes dataset, surpassing representative camera-only, LiDAR-only, and multimodal baselines (e.g., BEVFusion, TransFusion). Under sensor degradations (limited LiDAR FOV, point dropout, camera failure, object occlusion) and across weather/lighting subsets, ReliFusion maintains higher accuracy, with especially large gains at night. On efficiency, it runs at 10.4 FPS with a 96 ms pipeline latency. Despite a slightly larger parameter count, throughput exceeds TransFusion because runtime is governed mainly by FLOPs and execution characteristics; ReliFusion's per-frame compute is 540 GFLOPs versus

600 GFLOPs for TransFusion. The measured latency meets the perception budgets from Section 1.3 (96 ms $<$ 110/140/160 ms for highway/rural/urban). For strict 20 Hz LiDAR operation (50 ms), additional optimization (e.g., FP16, TensorRT, pipelining) or stronger hardware would be required.

Chapter 8

Discussion, Conclusion, and Future Work

8.1 Discussion and Conclusion

The primary objective of this research was to enhance the accuracy, computational efficiency, and robustness of LiDAR–camera fusion methods for autonomous driving perception. This objective was pursued through two complementary stages: the development of TransfuseNet, a lightweight 2D fusion framework, and the design of ReliFusion, a reliability-driven 3D detection model.

Accuracy under standard conditions. TransfuseNet demonstrated the benefits of mid- and late-level feature fusion, achieving competitive performance on the KITTI benchmark with inference latencies consistently below 40 ms. This satisfied real-time constraints and confirmed that efficient transformer-based designs can deliver accurate perception while maintaining computational efficiency. However, its fixed-weight fusion mechanism limited adaptability in degraded environments, such as under sensor failures or occlusions.

Robustness under degraded conditions. ReliFusion addressed this limitation by incorporating reliability-driven modules, including Spatio-Temporal Feature Aggregation (STFA), Cross-Modality Contrastive Learning (CMCL), and Confidence-Weighted Mutual Cross-Attention (CW-MCA). Experiments on nuScenes demonstrated that ReliFusion significantly improved robustness compared to existing baselines when facing reduced LiDAR field-of-view, camera occlusions, and adverse weather. This highlights the central contribution of this thesis: while TransfuseNet prioritized efficiency and accuracy in nominal conditions, ReliFusion advanced robustness and adaptability without compromising real-time feasibility.

A central factor behind ReliFusion's robustness was its modular design. The STFA, CMCL, and CW-MCA modules worked in concert to stabilize detection across frames, quantify sensor reliability dynamically, and adapt fusion weights in real time. Together, these mechanisms enabled the model to sustain performance under degraded LiDAR and camera conditions, underscoring that reliability-driven fusion is essential for dependable autonomous perception.

Deployment considerations. While ReliFusion achieved 96 ms inference latency and met typical safety-critical perception budgets, strict 20 Hz LiDAR operation (50 ms) would still require additional optimization or access to more powerful hardware. This limitation reflects a broader challenge in translating high-performing research models into cost-sensitive automotive platforms.

In summary, this thesis contributes to advancing multimodal perception by showing that reliability-aware fusion strategies can bridge the gap between accuracy in standard conditions and robustness in degraded scenarios. By explicitly incorporating sensor reliability into the fusion process, the proposed frameworks take a step toward safer and more dependable autonomous driving perception.

8.2 Future Work

Several research directions arise naturally from the findings of this work:

- **Lightweight reliability-aware fusion:** Designing compact architectures or pruning methods to balance robustness with computational efficiency for deployment on embedded platforms.
- **Multi-modality extension:** While this thesis focuses on LiDAR-camera fusion, the proposed reliability-driven framework can be extended to incorporate additional sensing modalities such as RADAR. It offers robustness under adverse weather and provides direct velocity measurements, complementing LiDAR geometry and camera semantics.

  However, incorporating RADAR introduces additional computational overhead and architectural complexity, including modality alignment and feature dimensionality expansion. Future work could investigate adaptive tri-modal fusion strategies, where reliability estimation dynamically balances LiDAR, camera, and RADAR inputs under varying environmental conditions.
- **Calibration and synchronization robustness:** Developing models that are resilient to calibration errors and inter-sensor timing offsets, ensuring reliability in real-world deployments.
- **Explainability of reliability-driven fusion:** Future work should make the fusion process more interpretable by providing insight into how reliability scores influence sensor weighting, thereby improving transparency and trust in decision-making.
- **Real-world validation:** Extending experiments beyond controlled datasets to large-scale field trials across diverse conditions, enabling validation under rare edge cases and operational stressors.

This thesis demonstrates that reliable autonomous perception requires not only high accuracy but also resilience to degraded sensing. Through the design of TransfuseNet and ReliFusion, this work advances the integration of efficiency, adaptability, and robustness, laying the groundwork for future autonomous systems that are both practical and dependable.

References

[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1090–1099, 2022.

[2] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[3] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016. Describes end-to-end system latencies including perception, decision, and control loops; combined system response typically under 600 milliseconds.

[4] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3d: A large benchmark and model for 3d object detection in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13154–13164, 2023.

[5] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving, 2020.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[7] Yuning Chai, Pei Sun, Jiquan Ngiam, Weiyue Wang, Benjamin Caine, Vijay Vasudevan, Xiao Zhang, and Dragomir Anguelov. To the point: Efficient 3d object detection in the range image with graph convolution kernels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2021.

[8] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[9] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[10] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. Futr3d: A unified sensor fusion framework for 3d detection. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 172–181, 2023.

[11] Yi-Nan Chen, Hang Dai, and Yong Ding. Pseudo-stereo for monocular 3d object detection in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 887–897, 2022.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg

Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] Martin Engelcke, Dushyant Rao, Dominic Zeng Wang, Chi Hay Tong, and Ingmar Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1355–1361. IEEE, 2017.

[14] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Anal. Stat. Model. Comput. Learn., Tech. Rep*, 2007(1-45):5, 2012.

[15] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Rangedet: In defense of range view for lidar-based 3d object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2918–2927, 2021.

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[17] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[18] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[19] Tianrui Guan, Jun Wang, Shiyi Lan, Rohan Chandra, Zuxuan Wu, Larry Davis, and Dinesh Manocha. M3detr: Multi-representation, multi-scale, mutual-relation 3d object detection with transformers. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 772–782, 2022.

[20] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, Jianwei Wan, and Ngai Ming Kwok. A comprehensive performance evaluation of 3d local feature descriptors. *International Journal of Computer Vision*, 116:66–89, 2016.

[21] Chenhang He, Ruihuang Li, Shuai Li, and Lei Zhang. Voxel set transformer: A set-to-set approach to 3d object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8417–8427, 2022.

[22] Qingdong He, Zhengning Wang, Hao Zeng, Yi Zeng, and Yijun Liu. Svga-net: Sparse voxel-graph attention network for 3d object detection from point clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 870–878, 2022.

[23] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[24] Junjie Huang et al. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022.

[25] Tengteng Huang, Zhe Liu, Xiwu Chen, and Xiang Bai. Epnet: Enhancing point features with image semantics for 3d object detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 35–52. Springer, 2020.

[26] Paul Jaccard. The distribution of the flora in the alpine zone.1. *New Phytologist*, 11(2):37–50, 1912.

[27] Glenn Jocher and Ultralytics. Yolov5. `https://github.com/ultralytics/yolov5`, 2021.

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[30] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019.

[31] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al. Yolov6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022.

[32] Shichao Li, Zechun Liu, Zhiqiang Shen, and Kwang-Ting Cheng. Stereo neural vernier caliper. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1376–1385, 2022.

[33] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Yifeng Lu, Denny Zhou, Quoc V Le, et al. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17182–17191, 2022.

[34] Zhiqi Li et al. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[35] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. Bevfusion: A simple and robust lidar-camera fusion framework. *Advances in Neural Information Processing Systems*, 35:10421–10434, 2022.

[36] Zhidong Liang, Zehan Zhang, Ming Zhang, Xian Zhao, and Shiliang Pu. Rangeioudet: Range image based real-time 3d object detector optimized by intersection over union. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7140–7149, 2021.

[37] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[38] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[39] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[40] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[41] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2774–2781. IEEE, 2023.

[42] Bence Major, Daniel Fontijne, Amin Ansari, Ravi Teja Sukhavasi, Radhika Gowaikar, Michael Hamilton, Sean Lee, Slawomir Grzechnik, and Sundar Subra-

manian. Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[43] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 922–928. IEEE, 2015.

[44] Zhixiang Min, Bingbing Zhuang, Samuel Schulter, Buyu Liu, Enrique Dunn, and Manmohan Chandraker. Neurocs: Neural nocs supervision for monocular 3d object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21404–21414, 2023.

[45] Alexander Musiat, Laurenz Reichardt, Michael Schulze, and Oliver Wasenmüller. Radarpillars: Efficient object detection from 4d radar point clouds. *arXiv preprint arXiv:2408.05020*, 2024.

[46] National Highway Traffic Safety Administration (NHTSA). Vehicle stopping distance and braking performance. Technical report, U.S. Department of Transportation, 2007. Typical emergency braking deceleration values with ABS systems range from 8 to 10 m/s^2.

[47] National Transportation Safety Board (NTSB). Collision between a car operating with automated vehicle control systems and a tractor-semitrailer truck near williston, florida. Technical Report HAR-17/02, National Transportation Safety Board, 2017.

[48] National Transportation Safety Board (NTSB). Collision between vehicle controlled by developmental automated driving system and pedestrian in tempe, arizona. Technical Report HAR-19/03, National Transportation Safety Board, 2019.

[49] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020:*

16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16, pages 194–210. Springer, 2020.

[50] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[51] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[52] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[53] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[54] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[55] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[56] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3577–3586, 2017.

[57] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009.

[58] Hualian Sheng, Sijia Cai, Yuan Liu, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, and Min-Jian Zhao. Improving 3d object detection with channel-wise transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2743–2752, 2021.

[59] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019.

[60] Vishwanath A Sindagi, Yin Zhou, and Oncel Tuzel. Mvx-net: Multimodal voxelnet for 3d object detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7276–7282. IEEE, 2019.

[61] Pei Sun, Weiyue Wang, Yuning Chai, Gamaleldin Elsayed, Alex Bewley, Xiao Zhang, Cristian Sminchisescu, and Dragomir Anguelov. Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5725–5734, 2021.

[62] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[63] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[64] Asher Trockman and J Zico Kolter. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022.

[65] Ultralytics. Yolov8: Object detection and image segmentation. `https://github.com/ultralytics/yolov8`, 2024.

[66] Koen EA Van de Sande, Jasper RR Uijlings, Theo Gevers, and Arnold WM Smeulders. Segmentation as selective search for object recognition. In *2011 international conference on computer vision*, pages 1879–1886. IEEE, 2011.

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[68] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023.

[69] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11794–11803, 2021.

[70] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions On Graphics (TOG)*, 36(4):1–11, 2017.

[71] Yizhou Wang, Zhongyu Jiang, Xiangyu Gao, Jenq-Neng Hwang, Guanbin Xing, and Hui Liu. Rodnet: Radar object detection using cross-modal supervision. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 504–513, 2021.

[72] Yue Wang et al. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2022.

[73] Jingsong Xie, Zhaoyang Li, Zitong Zhou, and Scarlett Liu. A novel bearing fault classification method based on xgboost: The fusion of deep learning-based features and empirical features. *IEEE Transactions on Instrumentation and Measurement*, 70:1–9, 2020.

[74] Junkai Xu, Liang Peng, Haoran Cheng, Hao Li, Wei Qian, Ke Li, Wenxiao Wang, and Deng Cai. Mononerd: Nerf-like representations for monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6814–6824, 2023.

[75] Qiangeng Xu, Yiqi Zhong, and Ulrich Neumann. Behind the curtain: Learning occluded shapes for 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2893–2901, 2022.

[76] Junjie Yan, Yingfei Liu, Jianjian Sun, Fan Jia, Shuailin Li, Tiancai Wang, and Xiangyu Zhang. Cross modal transformer: Towards fast and robust 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18268–18278, 2023.

[77] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[78] Honghui Yang, Wenxiao Wang, Minghao Chen, Binbin Lin, Tong He, Hua Chen, Xiaofei He, and Wanli Ouyang. Pvt-ssd: Single-stage 3d object detector with point-voxel transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13476–13487, 2023.

[79] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1951–1960, 2019.

[80] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021.

[81] Jin Hyeok Yoo, Yecheol Kim, Jisong Kim, and Jun Won Choi. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object

detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16*, pages 720–736. Springer, 2020.

[82] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 1821–1830, 2017.

[83] Yanan Zhang, Jiaxin Chen, and Di Huang. Cat-det: Contrastively augmented transformer for multi-modal 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 908–917, 2022.

[84] Yifan Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Wan, and Yulan Guo. Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18953–18962, 2022.

[85] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12993–13000, 2020.

[86] Chao Zhou, Yanan Zhang, Jiaxin Chen, and Di Huang. Octr: Octree-based transformer for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5166–5175, 2023.

[87] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018.

[88] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.