SAMFusion3D: Self-Adaptive Multi-modality Fusion for 3D Object Detection in Autonomous Driving

Tianyi Yu, MRes

SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER BY RESEARCH IN COMPUTING SCIENCE

SCHOOL OF COMPUTING SCIENCE

08 2024

Abstract

Autonomous vehicles rely on a diverse array of sensors to achieve comprehensive visual perception of their surroundings. Consequently, the integration of multimodal data, aimed at harnessing the complete spectrum of features from each sensor's Bird's Eye View (BEV) information, has emerged as a pivotal area of interest for numerous researchers. Currently, the research community is dedicated to enhancing the accuracy of detection models. However, given that the visual perception systems of autonomous vehicles are typically compact to medium-sized mobile platforms, computational complexity and efficiency are paramount. As the surrounding environment of an autonomous vehicle can fluctuate rapidly at times, maintaining a static sampling rate in such varied contexts results in suboptimal computational efficiency. Furthermore, as each modality's features are processed through Vision Transformers, particularly in the self-attention mechanism where the attention values for features are computed, it has been observed that adhering to the conventional pipeline approach results in elevated computational complexity and diminished efficiency. For the self-adaptive sampling mechanism, we adeptly extract depth information from camera features by utilizing point cloud data. Then, the fusion rate, which functions as a regulatory factor, dynamically adjusts the size of the effective sampling intervals, significantly impacting the computational load of the feature integration process. We also adopted the structure of the iTransformer that masterfully inverts the dimensions of the embedding. Our experiments conducted on the nuScenes dataset prove that our model can perform with reduced computational complexity while maintaining results comparable to those of the baseline model.

Contents

Acknowledgements

As my Master's program draws to a close, I find myself reflecting on the journey and the many individuals who have played a pivotal role in my academic and research growth.

At the forefront, I extend my profound thanks to my advisor, Dr. Hang Dai, for his steadfast support, incisive guidance, and relentless encouragement. His profound knowledge and astute feedback have been instrumental in refining this research endeavor. I am equally grateful to my committee members, Dr. Gerardo Aragon Camarasa and Dr. Paul Henderson, for their insightful critiques that have substantially elevated the caliber of my work.

My sincere gratitude goes to my colleagues and peers at the CVAS group, School of Computing Science, whose collaborative spirit and enriching dialogues have been integral to this study. Special mentions to Mr. Yingdong Ru, Mr. Zhuo He, Miss Lipeng Zhuang, and Mr. Shiyu Fan for their invaluable support and constructive critique of my master's project. Among the CVAS group, I single out Mr. Zeyu Dong and Miss Qianying Liu for their technical prowess and engineering assistance, which were crucial in integrating various modules with my work.

I am deeply thankful to my family and friends for their unwavering support and understanding throughout this academic voyage. Their constant encouragement has been a beacon of strength and motivation, propelling me forward in my research and personal endeavors. I am particularly touched by their timely presence and words of inspiration during moments of self-doubt, enabling me to swiftly regain my footing.

In conclusion, I wish to convey my heartfelt appreciation to everyone who has contributed to the success of this project. Your collective support has been the cornerstone of this endeavor, and I am eternally grateful for the role each of you has played in bringing this project to fruition.

Declaration

I declare that, except where explicit reference is made to the contribution of others, that this dissertation is the result of my own work and has not been submitted for any other degree at the University of Glasgow or any other institution

Tianyi Yu

Chapter 1

Introduction

Autonomous driving technology is transforming transportation by offering improved safety, efficiency, and convenience. At the heart of this advancement lies 3D Object Detection, a crucial innovation that maps objects in three-dimensional space, determining their positions, orientations, and dimensions. Unlike traditional 2D detection, 3D Object Detection is essential for autonomous vehicles to accurately identify and track pedestrians, vehicles, and obstacles. This capability empowers vehicles to make precise, informed decisions about speed, direction, and collision avoidance, ensuring the safety of passengers and other road users.

Modern 3D Object Detectors, driven by the growing demand for precise spatial perception in applications such as autonomous driving, primarily leverage deep learning methodologies. The two predominant approaches are Point Cloud-based and Voxel-based methods. Point Cloud-based detectors, like those introduced by Ding et al. 2021 in 2021, directly process raw LiDAR point cloud data, capturing fine-grained geometric details through a combination of CNNs and GNNs, effectively managing the irregular and sparse nature of the data. In contrast, Voxel-based detectors, such as those proposed by Zhou and Tuzel 2018 in 2018, transform point clouds into structured voxel grids, enabling 3D CNNs to extract spatial features with remarkable efficiency and success.

The escalating demand for autonomous vehicles has outpaced the capabilities of traditional single-sensor systems, such as cameras or LiDAR, which often falter in providing consistent and reliable data across varied conditions. This challenge has spurred the innovation of multimodal visual fusion techniques, which amalgamate data from an array of sensors to bolster the vehicle's perceptive prowess . In the realm of multimodal visual data fusion, each sensor brings its suite of strengths: cameras offer high-resolution color imagery, LiDAR provides accurate depth measurements, radar maintains effectiveness in inclement weather, and thermal cameras detect heat signatures in low-light conditions, proving invaluable for night-time navigation. By integrating these diverse datasets, as demonstrated in works such as Liu et al. 2023b, Li et al. 2022, and Kurniawan and Trilaksono 2023, autonomous vehicles can gain a more holistic and precise understanding of their surroundings. The Liu et al. 2022 and Liu et al. 2023a introduced a 3D Positional Encoding module that enhances multi-task detection capabilities. Cai et al. 2024, on the other hand, employs monocular depth estimation to transpose LiDAR features into image features, and BEVDepth introduces a DepthNet within RGB features, utilizing ground truth from points to supervise the RGB's sampling task, thus mitigating the impact of depth information scarcity.

The research on autonomous vehicles based on multimodal sensors is indeed promising, offering a comprehensive view of the environment and enhancing the safety and reliability of self-driving systems. However, several challenges remain to be addressed: To begin with, the integration of various sensor modalities, such as LiDAR, radar, and cameras, is complex due to the heterogeneity of data they provide. This includes differences in physical units, sampling resolutions, and spatio-temporal alignment, which significantly increase the complexity of preprocessing and alignment, thus affecting the efficiency of fusion Bokade et al. 2021.

Following up, the computational intensity of deep learning models, which are often central to multimodal fusion systems, is another critical challenge. While these models offer superior performance, they require substantial computational resources, potentially compromising real-time processing capabilities. This is especially problematic in autonomous driving scenarios, where timely and accurate perception is crucial for safety and efficiency Bokade et al. 2021. Moreover, the literature has presented innovative solutions to address the complex issue of multimodal sensor fusion efficiency, particularly within the context of autonomous driving scenarios. These contributions are instrumental in advancing the field by providing robust methods to handle the intricate challenges associated with integrating diverse sensor data for autonomous vehicle systems. For efficiency problem in multimodal fusion, Zhang et al. 2025 is an efficient multimodal data fusion framework that has been developed for 3D object detection and tracking. It introduces innovative fusion strategies such as High-Level Semantic Guidance (HLSG) and Multi-Priority Matching (MPM), which not only optimize the utilization of multimodal data but also enhance the complementary integration of different data types, significantly improving performance in adverse weather conditions.

Meanwhile, deploying autonomous driving vision systems in real-world scenarios presents a multifaceted set of challenges that significantly impact system performance and reliability. One of the primary issues is sensor noise, which arises from the inherent limitations of cameras and other visual sensors. These sensors are susceptible to variations in lighting conditions, weather, and occlusions, leading to degraded image quality and reduced accuracy in object detection and recognition tasks Geiger et al. 2012, Yan et al. 2024. Additionally, hardware limitations pose a significant constraint. The computational resources required for real-time processing of high-resolution visual data are substantial, necessitating efficient algorithms and specialized hardware to ensure low latency and high throughput Huang and Chen 2020. Environmental factors, such as adverse weather conditions (e.g., rain, fog, snow) and complex urban environments with dynamic objects and varying illumination, further complicate the task. These conditions can introduce additional noise and occlusions, making it difficult for vision systems to maintain consistent

performance Bijelic et al. 2020a. Addressing these challenges requires a combination of robust sensor fusion techniques, advanced machine learning algorithms, and adaptive computational frameworks that can handle the variability and unpredictability of real-world driving conditions.

Furthermore, the development of sensor-adaptive multimodal fusion methods, such as Palladin et al. 2024, has been crucial for 3D perception. These methods are designed to learn from a combination of RGB and LiDAR sensors, and can also incorporate data from Near-Infrared (NIR) gated cameras and radar to handle low-light and adverse weather conditions. By using attention-based deep fusion schemes and refinements in the Bird's Eye View (BEV) plane, these methods effectively integrate image and range features, enhancing the overall performance of autonomous vehicle systems. However, these articles do not address a key issue in autonomous driving: the integration of multimodal sensor systems in autonomous vehicles, while enhancing performance, also increases computational complexity in real-time, thereby reducing computational efficiency. This decrease in efficiency can lead to accidents when vehicles need to make rapid judgments. Therefore, this paper aims to propose a model that can solve the aforementioned problems.

Among all the challenges mentioned, the most critical and pressing issue that needs to be addressed is the trade-offs between the computational efficiency issue and the performance of the model. This complexity is a double-edged sword; it sharpens the vehicle's ability to perceive and interpret its surroundings but simultaneously dulls the blade of computational efficiency. The consequence of this is a potential lag in the vehicle's response time, which is particularly dangerous in scenarios requiring swift and decisive actions. A moment's delay could mean the difference between a safe maneuver and a catastrophic collision. Thus, the imperative for a model that can adeptly balance the scales of performance and efficiency becomes evident.

It is interesting to note that in the field of computer vision and autonomous driving perception, the computational demands have been a subject of considerable research. For example, papers such as Zhang et al. 2025 have already tackled the problem of improving the detection performance for cars by applying a better fusion strategy, Liu et al. 2018 proposed a method that reduces computational complexity by employing a low-rank multimodal fusion strategy in human motion detection, and Tan et al. 2022 resolved the problem by optimizing the data fusion strategy, efficient multimodal data processing has been achieved, improving the efficiency of occupancy detection in residential buildings. These scholarly contributions have been pivotal in propelling the field forward by offering robust methodologies designed to navigate the complex challenges associated with the integration of diverse sensor data within autonomous vehicle systems. However, autonomous driving systems need to process vast amounts of data in real-time and respond swiftly. Excessive computational complexity can slow down the system's processing speed, failing to meet the demands of real-time performance, which in turn affects the safety and efficiency of autonomous driving. Moreover, the computational resources of autonomous vehicles are limited, including the processing power of the processors, memory capacity, and power supply. Reducing computational complexity can decrease the consumption of hardware resources, enabling the system to operate more stably within these constraints. This work, which is the pursuit of efficient 3D multimodal autonomous driving perception is ongoing, with researchers actively developing new methods to improve data fusion strategies and computational efficiency. The ultimate goal is to create autonomous vehicle systems that can reliably interpret and react to their environment in real-time, ensuring safety and performance.

This paper, therefore, sets out to introduce an innovative model that not only maintains the high standards of performance offered by multimodal sensor integration but also ensures that this performance does not come at the cost of computational efficiency. By doing so, it seeks to mitigate the risk of accidents caused by computational delays, thereby

contributing to a safer autonomous driving experience. The proposed model is designed to be a robust solution that can operate efficiently under the stringent demands of real-time processing, ensuring that autonomous vehicles can make time-sensitive decisions without compromise.

To solve the problem mentioned above, we proposed two innovative modules. The first module is formed upon that the environment surrounding an autonomous vehicle can change rapidly at times, and at other moments, it evolves more gradually. Employing a static sampling rate in the face of such diverse scenarios leads to less than optimal computational efficiency. Moreover, when processing the features of each modality through Vision Transformers, especially within the self-attention mechanism where the attention values for features are determined, it has become apparent that sticking to the conventional processing pipeline increases computational complexity and reduces efficiency. Drawing inspiration from Li et al. 2023a, we propose a depth-aware adaptive multimodal visual fusion strategy. This strategy is predicated on the notion that an autonomous vehicle's perception system should intelligently adapt its data fusion tactics in response to the prevailing traffic conditions and driving environment. For example, the reliance on visual information from cameras is heightened in crowded areas. In contrast, in less dense or clearer settings, the reliance on camera-derived visual data is diminished. This adaptive approach ensures that the autonomous vehicle can sustain peak perceptual performance, regardless of the environmental challenges it encounters. In this work, we introduce a depth-aware self-adaptive sampler that extracts a depth map from camera features within a point cloud. Subsequently, it dynamically calculates the fusion rate—the proportion of camera features to be integrated—based on this depth map. This innovative method allows for the efficient and real-time determination of the average depth information for any given scene.

In our innovative approach, we've taken inspiration from the observation that traditional methods of processing multimodal features through Vision Transformers, especially within the self-attention mechanism, can be computationally intensive. As highlighted in the seminal work of Woo et al. 2018, inferring attention weights from mid-level feature maps by considering both channel and spatial dimensions, and then refining the original features accordingly, has been shown to enhance the model's ability to capture a wider array of object parts, thereby improving detection accuracy. This insight has led us to believe that the attention mechanism is pivotal in steering the model's focus toward the most relevant information. Building on this concept, we propose a novel hypothesis: detection accuracy during the feature preparation phase of our model could be significantly boosted by concatenating dimensional values of all patches into a single token. Borrowing from the innovative ideas of Liu et al. 2024, we emphasize the calculation of inter-signal correlations across all patches to create a comprehensive global representation. This method aligns with a signal-centric strategy, which is particularly advantageous for the emerging attention mechanisms aimed at harmonizing multiple signals. To implement this, we've introduced an architectural innovation: inverting the feature dimensions before they are fed into both the self-attention and feed-forward layers. This inversion allows for attention to be computed based on the intrinsic dimensional correlations within each patch at specific locations. A significant computational benefit arises from the fact that, in our model, the feature dimension for each patch is less than the number of patches. Consequently, as detailed in Woo et al. 2018, we achieve a substantial reduction in complexity, scaling down from $O(900^2)$ to $O(256^2)$.

This paper delves into the cutting-edge techniques of multimodal visual fusion, pivotal for the advancement of autonomous driving. Our key contributions in this domain are as follows:

1). We introduce a depth-aware fusion sampler that dynamically adjusts the fusion rate between the auxiliary and primary detectors, marking a novel approach in optimizing the multimodal visual fusion process for autonomous vehicles. This intelligent adaptation ensures that the auxiliary sensor operates only when necessary, with its activation gauged by depth calculations. Specifically, when the depth surpasses the threshold of 52 meters— the maximum measurable range in our project— the auxiliary detector is deactivated, relying solely on the primary detector for perception.

2). We incorporate the attention mechanism from the iTransformer into our model, augmenting the self-attention module for each feature dimension. This refined focus enables the model to concentrate more intently on the unique signals of each modality, thereby boosting the model's performance and its capacity to generalize across diverse scenarios. Particularly beneficial for detecting rare targets that infrequently appear within scenes, our experimental evaluations demonstrate a notable improvement in model performance with the integration of the iTransformer's attention mechanism.

Chapter 2

Related Work and Background

2.1 Related work

In recent years, remarkable strides have been made in the realm of 3D object detection and the fusion of multi-modality sensor data. Several distinguished studies have tackled the complex challenges within this field, proposing groundbreaking solutions aimed at enhancing both the precision and efficiency of detection systems. The following overview synthesizes discussions on the pivotal contributions made in this domain.

3D Object Detection. Within the realm of computer vision and sensory perception, a multitude of researchers have been intently focused on the pursuit of advancements in 3D Object Detection. As delineated in Section 1, the field of 3D Object Detection is primarily categorized into three approaches: Voxel-based, Point-based, and Bird's Eye View (BEV)-based. In the realm of voxel-based 3D Object Detection, the pioneering work of Ding et al. 2021 pioneered the direct extraction of voxel features for advanced processing. Meanwhile, the grid-based methodology, exemplified by the success of Zhou and Tuzel 2018, involves the transformation of raw point cloud data into a structured voxel grid, thereby facilitating more intricate analysis. Shifting focus to BEV-based detection, Wang et al. 2021b has made significant strides by converting multi-tiered RGB features

into object-centric BEV features. This approach provides a more holistic understanding of the spatial interrelations among objects within the environment. Similarly, the works of Li et al. 2022, Liu et al. 2023b, and Li et al. 2023a have each contributed to the extraction of multi-tiered RGB features into dense, grid-based BEV features. This methodology has proven to be more versatile in processing and more adept at identifying objects in scenarios rife with uncertainty.

Single-modality. The landscape of contemporary 3D Object Detection is neatly segmented into three categories, reflecting the diversity of detectors used: Single-modality, Dual-modality, and Multi-modality Object Detection. Single-modality approaches have achieved notable milestones, especially in environments where data is procured from a solitary sensor type, such as RGB cameras or LiDAR.

In the case of RGB cameras, Yan et al. 2023 noted that early efforts concentrated on dense prediction pipelines, which often fell short of capturing a holistic view essential for real-world autonomous driving scenarios. Furthermore, these detectors struggled to retain information about previously detected objects, leading to subpar detection outcomes. To surmount these challenges, Wang et al. 2021b introduced a groundbreaking transformer-based 3D object detector. This innovation sparked a new trend: transforming the vehicle's visual data into Bird's Eye View (BEV) features. The success of this paradigm was further substantiated by the works of Li et al. 2022 and Philion and Fidler 2020. Specifically, BEVFormer aggregates camera features from all six perspectives, demonstrating the efficacy of this approach. On the LiDAR front, Ding et al. 2021 pioneered an architecture that processes point clouds in an end-to-end manner. Subsequently, Zhou and Tuzel 2018 advanced the field by converting raw point clouds into structured voxel grids, streamlining the detection process.

Dual-modalities. However, in the intricate tapestry of driving scenarios, the reliance on a single modality is often insufficient. Addressing this challenge, dual-modality 3D Object Detection has emerged as a powerful solution. For instance, Li et al. 2023a ingests both raw RGB images and raw LiDAR point clouds, harnessing the depth information from the latter to enhance the precision of the semantic data derived from the former. Liu et al. 2023b, on the other hand, takes in raw data from cameras and LiDAR, and innovates by introducing two Task-Specific Heads: one dedicated to BEV Map Segmentation and the other to 3D Object Detection. This dual-focus approach allows for a more nuanced understanding and processing of the data. Yan et al. 2023 has introduced a Coordinates Encoding Module, which plays a pivotal role in minimizing the discrepancies in coordinates between different modalities, thereby improving the accuracy of the detection process. Cai et al. 2024 made a pivotal observation: before the fusion of BEV features, there is an inherent fusion error stemming from the varied representational formats of different modal features. To counteract this, it proposes a method that fuses the BEV features before feature processing, streamlining the integration and enhancing overall accuracy. In the realm of Radar detectors, Stäcker et al. 2023 represents a novel approach, presenting a radar-camera fusion architecture that operates on the BEV plane. This innovation is significant as it opens up new possibilities for how radar and camera data can be integrated to yield more robust detection results.

Multiple-modalities. As dual-modality fusion technology matures, there's a growing interest in leveraging raw data from three or more modalities to enhance real-time responsiveness to dynamic road traffic conditions. In this context, multi-modal 3D Object Detection is increasingly being explored. For instance, Li et al. 2023a ingests both raw RGB images and LiDAR point clouds, extracting depth information to enrich the semantic insights from cameras. Liu et al. 2023b processes inputs from cameras and LiDAR, introducing Task-Specific Heads that cater to BEV Map Segmentation and 3D Object Detection. Yan et al. 2023 introduces a Coordinates Encoding Module designed to minimize discrepancies in coordinates between different modalities. Cai et al. 2024 addresses the fusion error that arises from differences in feature representation of various modalities

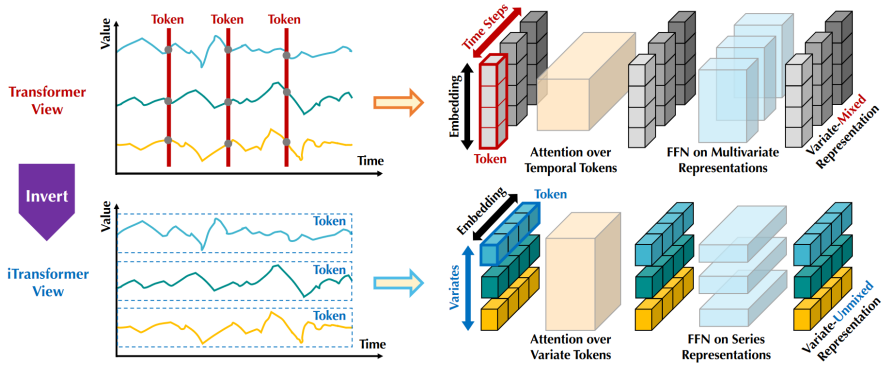

Figure 2.1: The overall pipeline of iTransformer. Liu et al. 2024

and proposes a method to fuse BEV features before feature processing. In the domain of Radar detectors, Stäcker et al. 2023 presents a novel radar-camera fusion architecture that operates on the BEV plane. Notably, Chen et al. 2023, as proposed by Chen et al., stands out as a pioneering unified end-to-end sensor fusion framework for 3D detection. It is designed to be versatile across all sensor configurations, integrating a query-based Modality-Agnostic Feature Sampler (MAFS) with a transformer decoder, complemented by a set-to-set loss function for 3D detection. This approach elegantly sidesteps the reliance on late fusion heuristics and post-processing techniques. FUTR3D can process raw inputs from RGB cameras, LiDAR, and radar, defining the 3D reference point from the projection plane coordinates, and applying a set-to-set loss in the decoding stage. Remarkably, FUTR3D, even when equipped with just a 4-beam LiDAR and cameras, achieves a 58.0 mAP score, surpassing the state-of-the-art 3D detection model, which scores a 56.6 mAP with a 32-beam LiDAR. This achievement highlights FUTR3D's potential to be a game-changer by offering high-performance detection capabilities with a fraction of the hardware resources typically required. In our project, we aim to delve into how modality fusion is executed across all Camera, LiDAR, and Radar features, which is why we have chosen FUTR3D as our baseline model.

Fusion Strategy. These years, researchers have dived into research on multimodal fusion strategies. Fusion strategies are divided into three categories: Early-fusion, Middle-fusion, and Late-fusion. For Early-fusion, Bijelic et al. 2020b proposed an entropy-based method for multimodal feature fusion across LiDAR, RGB Front View (FV) data, and LiDAR data into a specific type of BEV embedding called "entropy". The entropy itself is the variety of modalities. The richer the entropy is, the richer the information contained in the fused feature is. This paper tackled the problem of scenarios where annotated data is scarce and challenging to acquire due to the inherent bias of natural weather conditions. Meanwhile, conducting early-level fusion can easily deal with the problem stated above. Middle fusion is referred to by most of the current papers, which focus on multimodal fusion methods. For example, Li et al. 2022, Liu et al. 2023b, and Wang et al. 2021b uses a backbone to extract features from the raw data. Currently, late fusion is less commonly employed as the sole fusion technique in multimodal models due to its heightened requirements for sophisticated loss functions and nuanced attention mechanisms. The prevailing models in the field often leverage Cross-Attention within the Transformer architecture to calculate attention and manage the fusion process. Consequently, implementing late fusion without the integration of Cross-Attention presents a significant challenge, as it necessitates the development of advanced techniques to effectively amalgamate the diverse modalities at the decision-making stage.

Summary. Research in multimodal fusion, particularly using the Vision Transformer (ViT), is currently focused on improving the accuracy of predictive models. However, there appears to be a gap in the literature for reducing computational complexity to improve the efficiency of multimodal 3D object fusion tasks without sacrificing the detection accuracy of individual object classes. This research gap suggests that new research directions are needed to optimize algorithms and model structure to make the feature extraction, sampling, secondary sampling, and fusion process more memory-saving.

2.2 Background

Vision Transformer. Transformer Vaswani et al. 2017 was proposed for machine language tasks, and has been cited in many Natural Language Processing (NLP) tasks. Still, it can only deal with time-series-related works. As stated in Dosovitskiy et al. 2021, convolutional neural networks (CNNs) continue to hold a dominant position in computer vision, as evidenced by seminal works such as LeCun et al. 1989. Despite their theoretical advantages, these models have yet to achieve widespread adoption in large-scale image recognition tasks, primarily due to the challenges associated with scaling on modern hardware accelerators, which are often limited by the specialized nature of the attention patterns they employ. Consequently, in the context of large-scale image recognition, the classic ResNet-like architectures continue to set the benchmark for performance.

As depicted in Figure 2.2, different from the traditional Transformer model, which takes features across all feature dimensions at one timestamp as one token, Vision Transformer divides the feature at one timestamp into blocks, which are called patches. Each patch contains information on the features of the image in one position. After obtaining the information of one patch, attention is calculated to measure the correlation between each patch. For the rest of the model, it follows the pipeline proposed by Vaswani et al. 2017.

iTransformer. The recent surge in linear forecasting models has sparked a reevaluation of the fervor surrounding architectural enhancements to Transformer-based forecasters. These forecasters utilize Transformers to capture global dependencies across temporal tokens of time series data, with each token comprising multiple variates from the same timestamp. However, Transformers encounter challenges in forecasting tasks with extensive lookback windows, often leading to performance deterioration and computational strain. Moreover, the embedding of each temporal token, which amalgamates multiple variates representing potential lagged events and diverse physical measurements, may struggle to learn variate-specific representations, thereby resulting in ineffective attention

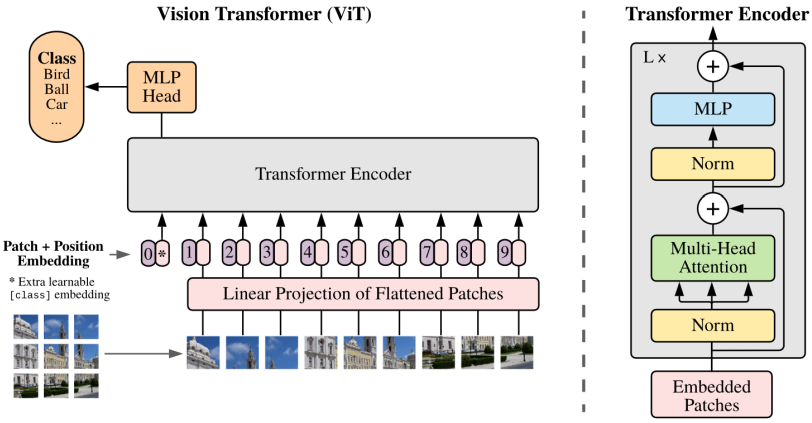

Figure 2.2: The overview of the Vision Transformer Model. Feature information is divided into several patches, each patch contains a batch of feature dimensions. In the vanilla Vision Transformer model, each token is obtained along the dimension of number of patches. Dosovitskiy et al. 2021

maps. iTransformer is a model that employs the attention and feed-forward mechanisms on inverted dimensions. Specifically, the time points of individual series are transformed into variate tokens, which are then leveraged by the attention mechanism to discern multivariate correlations. Concurrently, the feed-forward network is tasked with learning nonlinear representations for each variate token. The iTransformer model has achieved state-of-the-art performance on complex real-world datasets, thereby augmenting the Transformer family with enhanced performance, improved generalization across various variates, and optimized use of arbitrary lookback windows.

Chapter 3

Methods

3.1 Problem statement

The primary objective of this research is to optimize the computational efficiency of multimodal fusion while ensuring that the accuracy of the system remains uncompromised. To accomplish this goal, two innovative modules have been specifically designed and integrated into the framework. The first module is a dynamic adaptive sampling mechanism, which leverages the inherent sparsity of real-world scenes to selectively sample the most informative data points. By doing so, it effectively reduces the amount of redundant information processed, thereby minimizing unnecessary computational overhead. The second module is the iTransformer, a novel architecture designed to enhance the efficiency of feature extraction. It achieves this by optimizing the way features are extracted and processed, ensuring that only the most relevant and discriminative features are utilized in subsequent stages. The synergistic interaction between these two modules allows for a substantial reduction in computational complexity during the fusion process. This not only accelerates the overall system performance but also ensures that the fusion of multimodal data remains both efficient and effective, paving the way for more scalable and practical applications in complex, real-world scenarios.

3.1.1 Input Description.

1.Multi-view RGB images (F1). These images capture the visual spectrum and provide rich texture and color information. They are initially processed to extract features that are crucial for object recognition and classification.

2 LiDAR Data (F2). LiDAR sensors provide high-resolution 3D information about the environment. The data is used to generate a detailed point cloud, which is then processed to create a BEV feature set that is particularly robust in challenging conditions such as low light or adverse weather.

3. Radar Data (F3). Radar sensors contribute with their ability to penetrate through various weather conditions and provide velocity information. The radar data is processed to produce a BEV feature set that complements the LiDAR data, especially in dynamic scenarios.

The SAMFusion3D architecture, as introduced in our research, is designed to address the critical issue of real-time computational efficiency in autonomous driving systems equipped with multimodal sensors. The primary inputs to our model are multi-view RGB images, LiDAR, and Radar data, which are processed through specialized backbones to generate three distinct Bird's Eye View (BEV) feature sets, labeled as F1, F2, and F3.

Our proposed SAMFusion3D architecture, as depicted in Figure 3.1, synergizes multi-view RGB images, LiDAR, and Radar data through specialized backbones to generate three distinct Bird's Eye View (BEV) feature sets (F1, F2, and F3). We employ multi-head attention, feedforward layers, and layer normalization during feature extraction. Inspired by iTransformer, we invert the feature dimensions within the self-attention and feedforward layers to enhance processing efficiency. The extracted features are then subjected

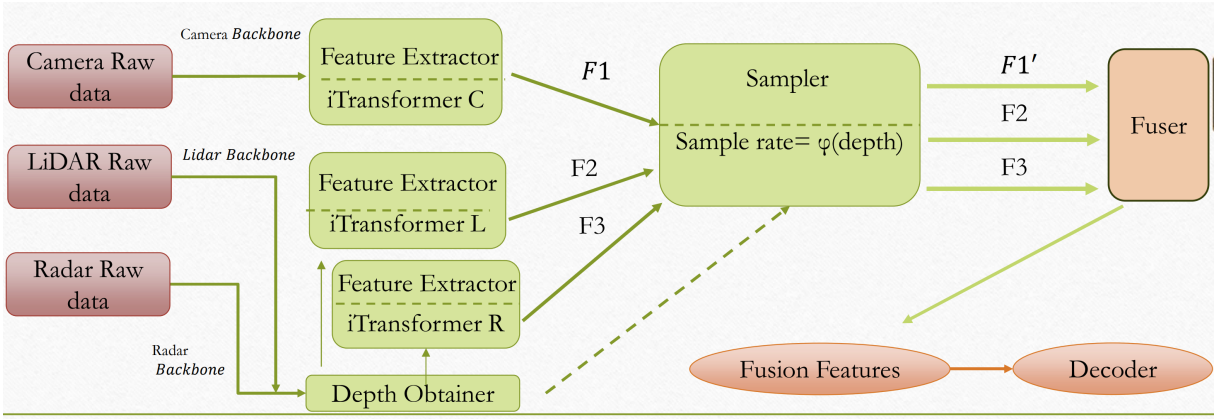

Figure 3.1: In the SAMFusion3D framework, we meticulously process multi-view imagery and point cloud data through their respective backbones. Features extractors, Depth Obtainer, and Sampler are employed to make the pipeline function more efficiently. The re-sampled camera feature and LiDAR feature are fused through the Cross-attention layers of the Transformer. After calculating attention across modalities, features are concatenated through linear layers (Fusion Features). Last but not least, feature is fed to the Decoder for loss calculation. For Camera and Radar fusion, we conduct experiments in the same way.

to a Sampler, which adjusts the feature set based on a real-time fusion sampling rate derived from depth cues. Depth is ascertained by processing detected object data through a Depth Obtainer, generating a depth map that aligns with the camera's coordinate system. This depth information is converted into a sampling rate by the Sampler module. While the Camera's BEV feature (F1) is re-sampled for fusion to enrich the detail, the LiDAR and Radar features (F2 and F3) are retained in their original form, capitalizing on their reliability in challenging conditions. The resulting re-sampled features (F1', F2, and F3) are merged in a final fusion step, as illustrated in Figure 3.3, culminating in an integrated representation that optimizes perception for autonomous driving applications.

3.2 Feature Extractor

The SAMFusion3D system adeptly handles a comprehensive array of raw data, including images captured by the six cameras mounted on the autonomous vehicle and raw point cloud data sourced from LiDAR and Radar sensors. This diverse dataset is funneled into the backbone network for initial processing. We utilize ResNet50 as the backbone network for image data, with dedicated networks addressing the point cloud data. The features extracted by these backbones are then conveyed to a Vision Transformer, which employs attention mechanisms to bolster feature integration. For the camera images, considering we have six distinct images, we adhere to the methodology established in previous work Li et al. 2022. This involves performing self-attention calculations on each image while also executing cross-attention across the entire suite of images. Once the features from the three distinct modalities are harvested, they are subjected to further refinement through self-attention layers that are tailored to each modality. During this refinement phase, we introduce an inversion process that targets both the self-attention and feed-forward layers, thereby optimizing the transformation of features. Prior to the Layer Normalization stage, the feature dimensions are meticulously restored to their original state. This carefully orchestrated series of operations ensures a thorough and sophisticated fusion of features drawn from multiple data sources, establishing a solid foundation for robust and dependable 3D object detection.

The structure of iTransformer is shown in Figure 3.2. Following is a detailed explanation of each block of iTransformer.

- **Embedding:** The process begins with the crucial step of transforming raw time series data into an embedding format. In this initial phase, each embedding is encapsulated as a query, adopting the shape of (batch size,number of queries, embedding dimensions). In a conventional transformer, each token is embedded across the dimension of the number of queries. However, in the innovative iTransformer, each token is embedded through the dimension of the embedding dimensions itself.

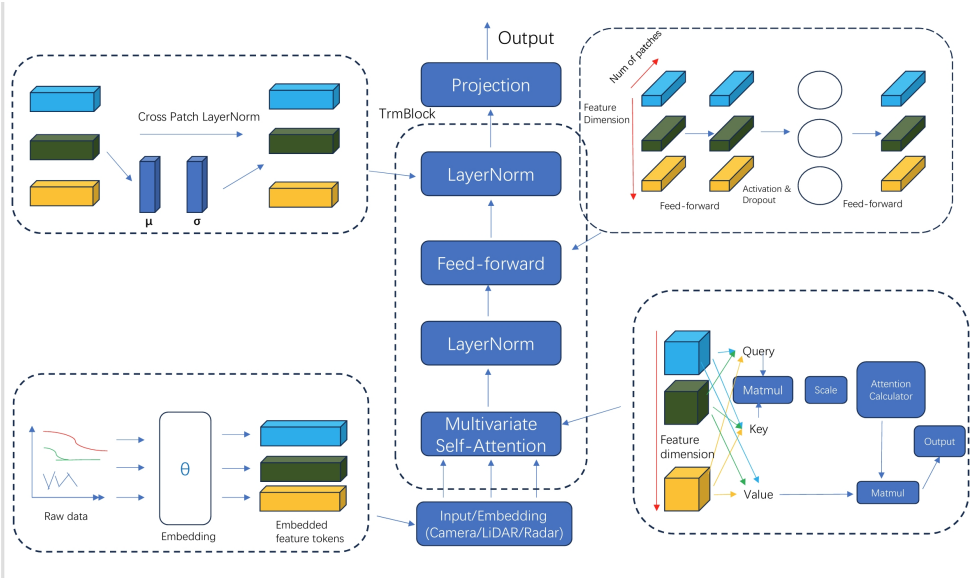

Figure 3.2: The iTransformer model, akin to the classic Transformer architecture, is meticulously structured into several key components: an embedding input, multi-head self-attention mechanisms, feed-forward networks, and layer normalization processes. Within the multi-head self-attention layers, the model ingeniously inverts the relationship between the embedding dimension and the feature dimensions of each patch. This inversion is then reversed in preparation for the layer normalization phase. Post the feed-forward layers, the tokens, which are derived from the feature dimensions of the individual patches, are subjected to another round of layer normalization. Ultimately, the meticulously processed features are projected and presented as the final output.

In our project, the dimensionality transformation process is ingeniously implemented within the Multivariate Attention module. To detail, Let's assume we have an input $X = \{x_1, \ldots, x_T\} \in \mathbb{R}^{T \times N}$, where T represents the number of patches and N represents the number of pixels for each patch. Note that $x_i = \{x_{i1}, \ldots, x_{iN}\}$ and here x_i is a vector used to store all the pixel features within each patch. x_i also represents a token that is input to the Transformer encoder. As depicted in Figure 1, we assume there are 4 pixels and 9 patches. Therefore, it has $T = 9$ and $N = 4$. Embarking on a novel approach, we adjust the input to $\hat{X} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_N\}$, and $\hat{x}_i = \{x_{1i}, x_{2i}, \ldots, x_{Ti}\}$. As shown in Figure 1, the novel number of tokens becomes 4. As we know, the computational complexity of the Transformer is about $O(\text{Tokens}^2)$, therefore, a substantial reduction in complexity scaling down from $O(9^2)$ to $O(4^2)$ with our proposed Feature Extractor in our example.

- **Multivariate Attention:** This section elaborates on the attention mechanism employed within our model. Similar to all Vision Transformers, it encompasses a multivariate attention process where queries, keys, and values undergo transformation and scaling before they are amalgamated into a multivariate correlation map. Leveraging the strengths of the iTransformer architecture, we introduce a novel inversion of the second and third dimensions of our embeddings at this stage. This inversion allows the transformer to calculate the attention correlations such that each token is represented from the perspective of the number of modalities. In contrast, traditional transformers typically derive tokens from the viewpoint of multiple queries, which can limit the nuanced representation of the data.
- **Feed-forward:** This part illustrates a sequence of dense layers, where the variates pass through dense transformations, activations, and dropout processes before moving to the next dense layer. In this part, we first invert the dimension of queries fed to the FFN (Feed-Forward networks) in each layer and revert it back to the original dimension for further processing.
- **LayerNorm:** In the realm of layer normalization, the iTransformer standardizes features based on their mean (μ) and standard deviation (σ) before their progression to the subsequent dense layer. In our project, we adopt the transformer base layers from Chen et al. 2019 to carry out the layer normalization. Consequently, the mean and standard deviation of the query samples are determined in accordance with the original configuration, ensuring a faithful and effective normalization process.

3.3 Depth Obtainer

The input features comprise Camera Bird's Eye View (BEV), LiDAR BEV, and Radar BEV features. All these BEV features are meticulously extracted utilizing the backbone architecture of Chen et al. 2023. Each feature set extracted—denoted as F1, F2, and F3—is pivotal in capturing distinct aspects of the surroundings, thereby enhancing the robustness and reliability of the fusion process. These trio of BEV features (F1, F2, and F3) are then ingested into the Sampler module.

Depth Obtainer. The Depth Obtainer will get the depth of objects detected by the LiDAR detector and create a corresponding depth map. Given the detected objects' information, the depth d is calculated. The transformation from LiDAR's depth map to the camera's coordinate system is performed using a method similar to Li et al. 2023a and Li et al. 2023b. This transformation involves the following steps:

1). **Depth Map Collection and Transformation.** This process starts with collecting raw 3D points from the LiDAR sensor. Each point is represented by coordinates (xi, yi, zi), where xi, yi, and zi represent the spatial coordinates of the i-th point.

2). **Transformation to Camera Coordinates.** To integrate the LiDAR data with camera data, we need to transform the LiDAR points into the camera's coordinate system. This involves using a transformation matrix T, which compresses rotation R and translation t. In this process, the py quaternion library in Python is used to conduct rotations and orientations. The transformation is performed as follows:

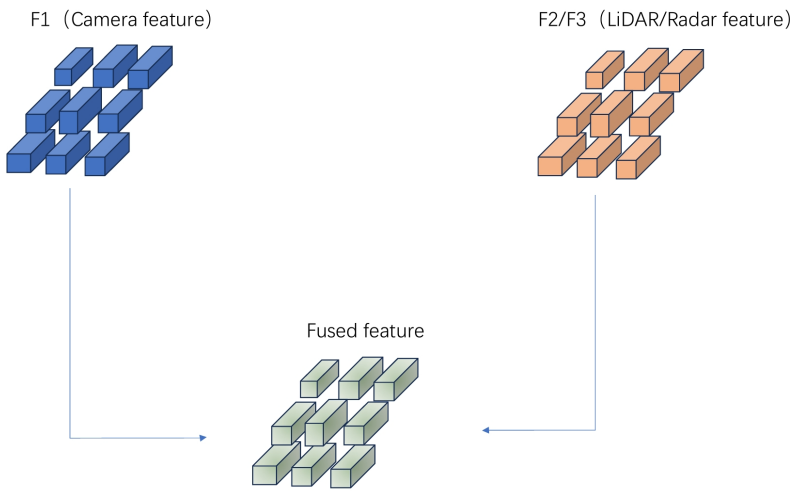

Figure 3.3: In the original scenario of feature fusion, a concatenation process is employed to integrate the distinct features derived from Camera and LiDAR/Radar sensors. This approach ensures that the complementary information from both sources is effectively combined, enhancing the overall performance of the system.

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = R \begin{pmatrix} x \\ y \\ z \end{pmatrix} + t \tag{3.1}$$

In this transformation equation 3.1, $R \in \mathbb{R}^{3\times3}$ represents **Rotation Matrix**. This matrix aligns the LiDAR coordinate system with the camera coordinate system by rotating the LiDAR points. $t \in \mathbb{R}^{3\times1}$ means **Translation Vector**. This vector shifts the LiDAR points to align them with the camera's position. In this paper, the z' after transformation represents the real-time depth information of the cameras.

3). **Average Depth Calculation.** The average depth $\bar{d}$ Is calculated by utilizing the mean of the depth values from the cameras.

$$\bar{d} = \frac{1}{N}\sum_{i=1}^{N} d_i \tag{3.2}$$

In equation 3.2, N represents the number of depth values from the cameras, and di represents the depth at the i-th point.

3.4 Sampler

Sample Rate Calculation and Resampling Process. In the Sampler module, a function is defined to project the calculated average depth value into the fusion sampling rate. The equation is illustrated as follows:

$$\phi(d) = \alpha \cdot d \cdot e^{-\beta} \tag{3.3}$$

In Equation 3.3, parametersα and β dictate the sampling rate, which is depth-dependent. Utilizing this rate, the Camera's Bird's Eye View (BEV) feature (F1) is resampled to become F1'. In contrast, the Radar and LiDAR BEV features (F2 and F3) remain unaltered, capitalizing on their superior reliability and confidence even under challenging conditions. For this study, we have assigned α and β the values of 1.

For further details of our proposed sampler, we extract a depth map from the LiDAR point cloud data, then employ a conversion matrix to align this depth map with the Camera's BEV feature, as delineated by Equation 3.1. Subsequently, we determine the sampling rate using the camera feature's BEV map. This rate guides our selection of which feature areas will contribute to the fusion process. We implement a zero-padding strategy for areas not selected for fusion, effectively padding them with zeros.

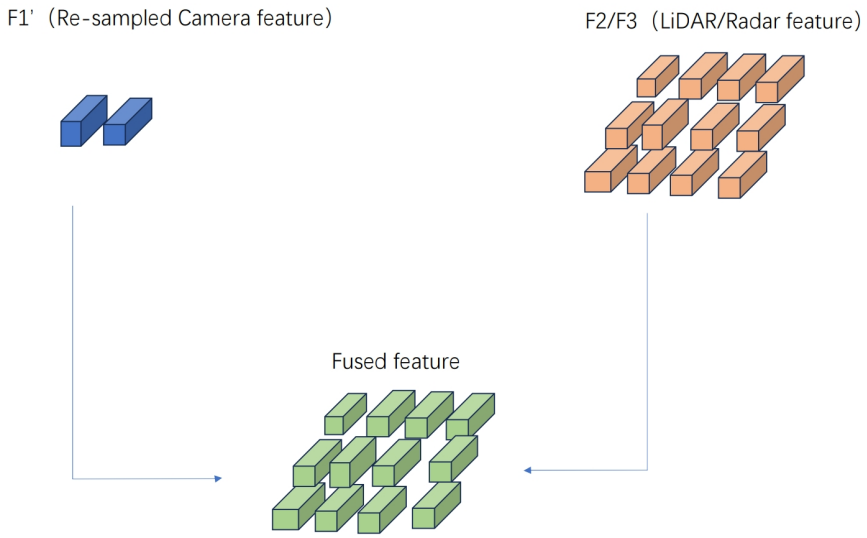

Figure 3.4: Feature fusion is enhanced with our innovative sampler, where F1' denotes the Bird's Eye View (BEV) feature from the camera, and F3 signifies the BEV feature from the radar (alternatively denoted as F2 when fused with the LiDAR BEV feature). In our approach, the feature region utilized for fusion is dynamically modulated based on the depth value, allowing for an adaptive adjustment of the fusion rate. The blocks with blue gradient color indicate effectively fused patches, where objects are detected. For patches without gradient colors,

The figure illustrates arrows connecting F1 with F3/F2, indicating that padding is not required for these features, as they serve as the primary detectors in their respective detection tasks. To encapsulate, this fusion strategy is prevalent in architectures aimed at tasks that require multi-scale feature integration, such as image segmentation, object detection, and complex scene understanding. By effectively merging features of varying resolutions and types, this approach not only bolsters the model's performance but also enhances computational efficiency by reducing the processing time for a set number of frames.

3.5 Loss Function

We adhere to the methodologies outlined in Wang et al. 2021b, Wang and Solomon 2021, and Zhu et al. 2021 employing a set-to-set loss calculation between the predictions and the ground truths, facilitated through one-to-one correspondence. For the classification task, we utilize focal loss, while for the 3D bounding box regression, we adopt the L1 loss, aligning with the approaches of DETR3D and FUTR3D. In detail, The typical scenario of this kind of problem involves a smaller count of ground-truth bounding boxes, in comparison to the number of predicted bounding boxes. To facilitate the computational process, we augment the ground-truth set with placeholder values (indicating no object) until it matches the count of predictions. This alignment is achieved by addressing the task as a bipartite matching problem, thereby creating a clear correspondence between each ground-truth box and its predicted counterpart. In detail, the equation of the loss function is listed as follows:

L1 Loss. The L1 loss, also known as the Mean Absolute Error (MAE), is given by:

$$L1 = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{3.4}$$

where N is the number of samples, y_i is the actual value for the i-th sample, and $\hat{y}_i$ is the predicted value for the i-th sample.

Focal Loss. The Focal loss is defined as:

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{3.5}$$

where p_t is the model's predicted probability for the true class, α_t is the weight for the class, and γ is the focusing parameter.

The total Focal loss for a batch of samples is the average of the Focal losses for each sample:

$$FL_{\text{total}} = \frac{1}{N}\sum_{i=1}^{N} FL(p_{ti}) \tag{3.6}$$

where p_{ti} is the predicted probability for the true class of the i-th sample in the batch.

3.6 Algorithms

To elucidate the training process of our model, we have meticulously crafted a visual representation of the algorithm in Algorithm 1. This diagram provides a clear and intuitive overview of the model's training regimen. It is important to note that Algorithm 1 specifically illustrates the fusion of Camera and LiDAR features. The feature extraction and other steps for Camera and Radar fusion are analogous to those depicted. As shown in Algorithm 1, the model begins by ingesting raw data from the Camera, LiDAR, and Radar sensors within the nuScenes dataset. Prior to training, we initialize the weights for the feature extractors of each modality and the decoder, denoted as θ_m^f and θ_d, respectively. During each epoch, if LiDAR is selected as the primary detector to integrate with the Camera feature, the feature extractor is employed to extract features from the LiDAR feature. Concurrently, our Depth Obtainer calculates the average depth of the live scene. This depth measurement, guided by the LiDAR, is then assigned to variable D. Once D is determined, the Sampler is utilized to re-sample the Camera feature, using both the extracted Camera feature and depth D as inputs. Subsequently, the feature from the primary detector, F_L, and the re-sampled Camera feature are combined and fed into the Fuser for feature fusion. The output of the Fuser is the fused feature F, which, in conjunction with the decoder training parameter θ_d, is passed to the Decoder module.

This process culminates in the computation of the loss, for which we have adopted the Focal Loss method as detailed in Section 3.5. Upon calculating the loss value, we proceed to update the weights utilized throughout the training process following the loss function outlined in Chen et al. 2023.

Algorithm 1: Algorithm for the training process of SAMFusion3D

input : nuScenes raw data from Camera C, LiDAR raw point cloud data L
Initialize weights of feature extractor θ_m^f for modality m;
Initialize weights of decoder θ_d ;
for $i = 1$ *to max_iter* **do**
  $F_m = \text{FeatureExtractor}(m, \theta_m^f), m \in \{C, L\}$;
  if $m = L$ **then**
    $D \leftarrow \text{DepthObtainer}(L, C)$;
  $F_C^{'} = \text{Sampler}(F_C, D)$;
  $F = \text{Fuser}(F_C^{'}, F_L)$;
  $\text{Loss}(\theta_m^f, \theta_d) = \text{Decoder}(F, \theta_d)$;
  Update θ_m^f, θ_d using $\text{Loss}(\theta_m^f, \theta_d)$ [Chen et al. 2023];

3.7 Summary

In summary, our model has been meticulously crafted, building upon an existing model that adeptly handles the fusion of three-modal raw data as our baseline. Within the Feature Extractor module, we have integrated the iTransformer to reverse the dimensions processed into Transformer tokens, thereby enhancing the computational efficiency of the feature extraction process. Furthermore, we have introduced a novel Sampler that re-samples the extracted Camera feature, leveraging the guidance from the LiDAR feature and its real-time average depth, which is derived by our Depth Obtainer. Once the re-sampled Camera feature and the LiDAR feature—or Radar feature, should we opt for Radar as the primary detector—are obtained, they are channeled into the Fuser and Fusion Features module. This step concatenates the features from the different modalities.

Ultimately, the fused feature is then passed to the Decoder component to compute the loss value. This comprehensive approach ensures that our model not only maintains the performance standards but also optimizes the computational efficiency, making it a robust solution for multi-modal data fusion tasks.

Chapter 4

Experiments Evaluation

4.1 Dataset and Model

We evaluate our proposed method on **nuScenes** benchmark and dataset. The nuScenes dataset, proposed by Caesar et al. 2019, is a large-scale and multi-task dataset designed for tasks such as 3D object detection, bird's eye view (BEV) segmentation/detection, and 3D object tracking. It is divided into training, validation, and testing sets, containing 700, 150, and 150 scenes, respectively.

For both 3D object detection, each scene contains 20 seconds of video sequences and is annotated with around 40 keyframes. Detailed ground-truth data is provided for specific frames in the video, like the positions and attributes of objects in each scene. These annotations allow the system to evaluate how well the model detects objects over time.

4.2 Metrics

To assess the performance of these detection approaches, several key metrics are used:

1). nuScenes Detection Score (NDS): This is a composite score that evaluates the overall performance of the 3D object detection system. It takes into account multiple aspects of detection performance, providing a holistic view of how well the model performs. A higher NDS score represents a better performance across all the key metrics. This metric is higher the better. The formula of calculating NDS is:

$$\text{NDS} = \frac{\text{mAP} + \text{mATE} + \text{mASE} + \text{mAOE} + \text{mAVE} + \text{mAAE}}{6} \tag{4.1}$$

The nuScenes Detection score is obtained by averaging all six metrics, to get a comprehensive value reflecting the model's overall performance.

2). Mean Average Precision (mAP): This metric measures the precision of the model in detecting objects at various levels of confidence. It evaluates how accurately the model detects objects without producing a great amount of false positives. Mean Average Precision is calculated by averaging the precision scores across different detection thresholds and object categories. This metric is higher the better as well. The formula of mAP is:

$$mAP = \frac{\sum_{\text{all classes}} AP_{\text{class}}}{\text{number of classes}} \tag{4.2}$$

where AP represents the average precision of each category. For each category, precision is calculated according to the recall values. Each recall will correspond to a value of precision. Each recall will correspond to a value of precision. The correspondence between these values and the recall can be reacted as an R-P curve. Our AP value is obtained by calculating the area below the R-P curve.

3). Five True Positive(TP) Metrics: These metrics specifically evaluate different aspects of true positive detections, or correctly detected objects. They are critical for understanding the strengths and weaknesses of the detection system.

- **Mean Average Translation Error (mATE) :** This measures the average **positional error** in the detected objects. It tells how far the predicted object's position is from its ground-truth location in 3D space.

  The formula of mATE is:

$$mATE = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(x_i - x_i^*)^2 + (y_i - y_i^*)^2} \tag{4.3}$$

  where N represents the total number of objects detected, xi and yi represent the coordinates of objects detected to the center point, while $xi*$ and $yi*$ represent the coordinates of ground-truth objects to the center point.
- **Mean Average Scale Error (mASE) :** This captures the error in estimating the **scale** between the detected objects and their ground-truths. mASE is calculated in the formula of:

$$mASE = \frac{1}{N}\sum_{i=1}^{N}(1 - \mathrm{IoU}_i) \tag{4.4}$$

  where $IoUi$ represents the Intersection over Union between the i-th predicted bounding boxes and the i-th ground truth bounding boxes.
- **Mean Average Orientation Error (mAOE):** This evaluates how well the model predicts the orientation or direction of detected objects. It is important for tasks to know the objects' facing direction, e.g. Predicting where a vehicle is going. The formula of mAOE is:

$$mAOE = \frac{1}{N}\sum_{i=1}^{N}|\theta_i - \theta_i^*| \tag{4.5}$$

- **Mean Average Velocity Error (mAVE):** This assesses the accuracy in estimating the velocity of moving vehicles. For example, it checks how many accuracies between the predicted velocity, direction, and ground-truth velocity. The formula of mAVE is:

$$mAVE = \frac{1}{N}\sum_{i=1}^{N}|\mathbf{v}_i - \mathbf{v}_i^*| \tag{4.6}$$

  where N represents the total number of objects, vi represents the predicted velocity vector, and $vi*$ represents the ground-truth velocity vector.

- **Mean Average Attribute Error (mAAE):** This metric measures the model's accuracy in estimating the velocity of moving objects. For example, when the model successfully detects and recognizes a car, this doesn't mean the model can accurately recognize whether the car is parked or moving. This metric can tell whether a car is parked or moving, or whether it has its lights on. Different objects may have different relevant attributes, more attributes being recognized means the model's performance is better. The formula for calculating mAAE is listed as follows:

$$mAVE = \frac{1}{N}\sum_{i=1}^{N}|\theta_i - \theta_i^*| \tag{4.7}$$

  where θ represents the angle between which the object's velocity is heading and directly in front.

4.3 Implementation details

Cameras. The nuScenes dataset gives images from six cameras around the ego-car for each frame. They are *front-left, front, front-right, back-left, back and back-right ;* The resolution of frames is 1600 * 900.

LiDARs. Low-resolution LiDARs are frequently utilized in many cost-effective applications. We view these low-resolution LiDARs as complementary to high-resolution counterparts due to their scalability for deployment in production-ready platforms. To simulate low-resolution LiDAR outputs, we downsample data from a 32-beam LiDAR. This process includes coordinate transformation, defined by range (γ), inclination (θ), and azimuth (ϕ). LiDARs are separated according to the number of beams. For 4-beam LiDAR, we select beams whose inclination angles (θ) fall within the intervals$[-7.1^\circ, -5.8^\circ] \cup [-4.5^\circ, -3.2^\circ] \cup [-1.9^\circ, -0.6^\circ] \cup [0.7^\circ, 2.0^\circ]$.

Radars. For Radar information, we aggregate the points collected from all five radar detectors into a single point cloud, with each point cloud containing between 200 and 300 points per frame. For each radar point, we utilize its coordinates, velocity measurements, and intensity values. To refine the radar data, we apply the official filtering tool provided by the nuScenes dataset.

Experimental setting-up. In this segment, we adhere to the configuration of our baseline model. For each of the three feature sets, we establish a feature dimension of 256. A total of 900 object queries are deployed, with each feature map dimensioned at 30×30 pixels. For the extraction of features from LiDAR and camera data, we utilize M=4 layers of multi-scale features, which are encoded through a Feature Pyramid Network (FPN). During the application of Deformable Attention, we implement K=4 sampling offsets. The transformer's detection head is configured with nHead=8, and within each head, it encompasses a total of L=6 blocks in its decoder. The choice of nHead=8 in the transformer's detection head is crucial for enhancing the model's ability to capture diverse and complex patterns within the input data. Specifically, the multi-head attention mechanism allows the model to split the input into multiple subspaces, each processed by an individual attention head. With nHead=8, the model can effectively attend to eight different aspects of the input simultaneously. This parallel processing enables the model to capture various relationships, such as syntactic and semantic dependencies in natural language processing tasks or spatial and contextual relationships in computer vision tasks. Each

attention head computes its own set of query (Q), key (K), and value (V) matrices, allowing the model to focus on different parts of the input sequence. The outputs from these heads are then concatenated and transformed through a linear layer, which integrates the information from all heads. This design not only increases the model's capacity to capture nuanced patterns but also reduces the risk of missing important details that might be overlooked by a single attention mechanism. For our multi-modality fusion sampler, the parameters α and β represents he weights for each modality in the fusion process. Representing the level of trust in a particular modality, an example would be adjusting the value of α when the camera detector's performance is suboptimal. By lowering the α value, even if the fusion process is currently integrating all features from the camera, our trust in the camera modality is diminished due to the reduced alpha value, meaning it won't be fully integrated. For β, the principle is the same. However, this approach should only be used when fusing different detectors to avoid introducing human-induced errors into the experimental results. In that case, we assign the values of each in equation 3.3 to be 1. Regarding the iTransformer's configuration, we retain the settings equivalent to the iTransformer's baseline model Liu et al. 2024, setting the embedding dimension for the attention layer to 256 and designating 900 queries per feature map. In light of the necessity to switch between various sensor modalities within our model, we have meticulously designed a sophisticated fusion mechanism. This device is pivotal in the final stage of our processing pipeline, where a critical decision regarding adopting a fusion strategy is made. Should we opt not to employ a fusion strategy, the LiDAR features, which have undergone dimension inversion processing, will be utilized directly as the model's output. Conversely, suppose a fusion strategy is deemed beneficial. In that case, it is instantiated by incorporating a cross-attention mechanism at the culmination of the fusion device, effectively materializing the fusion function.

Throughout the training phase of this study, we have explored three distinct modal configurations to leverage the strengths of each sensor type and their combinations:

LiDAR Only Configuration. In this setup, the model is trained exclusively with LiDAR data, bypassing the fusion strategy. This allows us to assess the performance of LiDAR as a standalone modality and serves as a baseline for comparison.

Camera-LiDAR Fusion Strategy. Here, we integrate data from both the Camera and LiDAR sensors. The fusion strategy is applied to harmoniously combine the rich semantic information from the Camera with the precise spatial data from the LiDAR, enhancing the model's environmental perception capabilities.

Camera-Radar Fusion Strategy. This configuration focuses on the synergy between Camera and Radar data. By fusing these two modalities, we aim to capitalize on the Radar's ability to penetrate through adverse weather conditions and provide velocity information, complemented by the Camera's detailed visual cues.

The implementation of these modal configurations and the strategic decision to employ or forgo fusion strategies are integral to our research. They provide a comprehensive evaluation of the model's adaptability and performance across different sensory inputs, offering insights into the most effective ways to harness multimodal data for autonomous driving applications.

Training details. In our experimental phase, we trained our model for a duration of 20 epochs under the aforementioned settings and carried out testing on both the nuScenes test set and validation set, utilizing the Camera and LiDAR configuration. When it came to the Camera and Radar configuration, we conducted a training period of 6 epochs, followed by testing and validation on the nuScenes datasets, all in accordance with the parameters of our baseline model.

Significantly, for LiDAR-based detectors, we adopted the AdamW optimizer Loshchilov and Hutter 2019 for training, initializing the learning rate at 1.0×10^{-4} and implementing a cyclic learning rate policy. During the training period, the learning rate follows an initial upward trend, subsequently declining. This pattern is often employed to facilitate a more robust optimization process, allowing the model to explore the parameter space more effectively at the outset and then fine-tuning its weights with smaller steps as it approaches the optimal solution. The cyclic learning rate policy involves varying the learning rate during training, which can help the model escape local minima and achieve better generalization. Other configurations adhere to the original settings established in FUTR3D. In line with established practices Li et al. 2023a, Huang and Huang 2022, we employed ResNet50 He et al. 2016a with the native image dimensions of 256 x 704. Here, the number 256 represents the width of the image, and 704 represents its height. Consequently, for each image-derived point cloud map, we obtain a total of 256 * 704 values, which correspond to the raw features of the current image (features that have not been processed). These raw features serve as the initial data points that our model will use to learn and make predictions, providing a foundational representation of the visual information captured by the camera. Throughout the training process, we executed a data augmentation strategy encompassing the camera, LiDAR, and radar modalities. These augmentations comprised random scaling, flipping, and rotation, mirroring the methodology employed in BEVDet Huang et al. 2021. In the initial training phase for the fusion sampler and the iTransformer, we trained the model with both the sampler and the iTransformer, applying dimension inversion in both the Camera and LiDAR configuration and the Camera-Radar configuration. Subsequently, in the ablation study, we eliminated the sampler and refrained from conducting dimension inversion to assess the model's performance. Regarding the Camera's real-time depth information, we utilized the real-time LiDAR point data, leveraging the LiDAR raw points' coordinate information and transforming it into the Camera's feature map for depth supervision via the matrix that converts the Camera coordinates to LiDAR coordinates, which facilitates the

transformation between different modal features. For the LiDAR and Radar's real-time depth information, to emulate a full-time operational state of the detectors, we designated the real-time depth information for both LiDAR and Radar to a maximum of 50 meters, correlating to the maximum observable depth range of the detectors.

Table 4.1: The table provides a comprehensive comparison of various methodologies utilized for 3D object detection tasks, as evaluated by several performance metrics on the nuScenes test dataset.

Method	**Modality**	**NDS**	**mAP**	**mATE**	**mASE**	**mAOE**	**mAVE**	**mAAE**
FCOS3D [Wang et al. 2021a]	C	40.2	32.6	74.3	25.9	44.1	134.1	16.3
PETR [Liu et al. 2022]	C	48.1	43.4	64.1	24.8	43.7	89.4	14.3
FUTR3D-vovNet [Chen et al. 2023]	C	47.9	41.2	64.1	25.5	39.4	84.5	13.3
MMDetection3D [Chen et al. 2019]	L	65.3	57.5	31.6	25.6	40.9	23.6	12.4
FUTR3D [Chen et al. 2023]	L	69.9	65.3	28.1	24.7	36.8	25.3	12.4
3D-CVF [Yoo et al. 2020]	L+C	62.3	52.7	30.0	24.5	45.8	27.9	12.2
FUTR3D-ResNet50 [Chen et al. 2023]	L+C	67.0	**61.9**	31.6	26.6	31.0	32.8	17.8
SAMF3D-ResNet50	L+C	66.3	60.0	32.7	**25.9**	**29.0**	**28.2**	19.0

Fusion Rate. To evaluate the real-time memory computational complexity, we employ the term *Fusion Rate* as a metric to quantify the workload during the actual fusion process. The concept of *Fusion Rate* is defined as follows:

$$\text{Fusion rate} = \frac{\text{Pu}}{\text{Pa}} \cdot 100\% \tag{4.8}$$

In this context, *Pu* denotes the subset of patches that are actively engaged in the actual fusion process, while *Pa* refers to the complete set of patches encompassed within the Camera feature. These patches are essentially the blocks segmented during the vision transformer stage. In the baseline scenario, the Camera feature incorporates 900 patches for fusion, representing the total count of patches available. However, upon re-sampling the feature, the actual number of patches utilized is expected to decrease below 900. This reduction occurs because the real-time depth measurement is invariably greater than zero, which impacts the patch selection. Consequently, the computational complexity can be articulated as follows:

$$\text{Complexity} \propto \sum_{n=1}^{\text{n\_heads}} O(\Phi \text{c}^2) + O(\Phi \text{l}^2) \tag{4.9}$$

where ϕ_c represents the input dimension of the Camera modality and ϕ_l represents the input dimension of the LiDAR modality. The specific formulas for these two parameters are as follows:

$$\phi_c = \phi_{co} \times fusion\_rate\_cam \tag{4.10}$$

where ϕ_{co} means the original input dimension of Camera modality, and ϕ_{lo} indicates the original input dimension of LiDAR modality. As detailed in equation 4.10 and 4.11.

$$\phi_l = \phi_{lo} \times fusion\_rate\_lidar \tag{4.11}$$

As detailed in Woo et al. 2018, the computational complexity of a Transformer model is $O(n^2)$. Accordingly, this complexity is proportional to the square of the fusion rate. Therefore, the overall computational complexity is determined by the sum of the complexities associated with each modality, followed by a simultaneous summation across the dimension of the number of detected heads. In our project, we adhere to the original configuration of FUTR3D by setting the value of ***n_heads*** to 8. The formula that delineates the relationship between the fusion rates of LiDAR and Camera, as well as the number of detection heads, is presented in Equation 4.9.

4.4 Results

Camera and LiDAR fusion. We have conducted experiments focusing on the fusion of LiDAR and Camera data. To simulate a realistic scenario where LiDAR serves as the primary detector and the Camera acts as the secondary detector, we have chosen to utilize raw image data from the Camera and raw point cloud data from the LiDAR as our input data. Table 4.5 presents a comparative analysis of 3D multi-modality detection results on the nuScenes test set, featuring our baseline FUTR3D [Chen et al. 2023], FB-BEV [Li et al. 2023b], and other leading-edge methods. We trained the model on the nuScenes training dataset for 20 epochs, following the original setting of FUTR3D. After training, we tested the results on the nuScenes test dataset, to get the results. To verify the effect of conducting experiments on different servers to the results, we conducted experiments on FUTR3D on ResNet50 backbone and LiDAR-only configurations. Experiments show that this impact is low enough that can be neglected.

As shown in Figure 4.5, the methodologies, such as FCOS3D Wang et al. 2021a, PETR Liu et al. 2022, and FUTR3D, incorporate diverse network architectures and leverage different input data modalities, encompassing Camera (C), LiDAR (L), and their combined use (L+C). The performance metrics encompass the nuScenes Detection Score (NDS), mean Average Precision (mAP), mean Average Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error (mAOE), mean Average Velocity Error (mAVE), and mean Average Attribute Error (mAAE). The table underscores the comparative strengths and weaknesses of each methodology across these metrics. Notably, the CenterPoint-Ensemble achieves the highest NDS score of 67.5%, demonstrating its overall effectiveness in the detection task. FUTR3D-ResNet50 stands out in terms of mAVE, with the lowest error rate of 28.2, indicating its precision in estimating the velocity of detected

objects. Furthermore, SAMF3D-ResNet50 (an abbreviation for SAMFusion3D-ResNet50) excels in mASE, mAOE, and mAVE, achieving the lowest error rates of 25.9%, 29.0%, and 28.2% respectively, which highlights its robustness in accurately determining the scale, orientation, and velocity attributes of detected objects.

When FUTR3D employs only LiDAR data (L), it demonstrates a notable enhancement over preceding methods like MMDetection3D Chen et al. 2019, with an NDS of 69.9% and an mAP of 65.3%. In the realm of multi-modality fusion, our initial experiment combined LiDAR and Camera features. For this, we selected ResNet50 He et al. 2016a as the backbone for the Camera. Given that the FUTR3D baseline was originally executed with VovNet, we also re-implemented FUTR3D on ResNet-50 for 20 epochs to establish a control group for our experimental outcomes. As indicated in Table 4.5, SAMF3D-ResNet50 outperforms in mAVE, with the lowest error rate of 28.2, suggesting that despite similar mAP and NDS values compared to FUTR3D, its capability to estimate the velocity of moving objects is superior. This affirms that SAMF3D-ResNet50 matches the performance metrics of the FUTR3D model.

Regarding SAMF3D, as illustrated in Figure 4.2, the original fusion rate is consistently maintained at 1.00 throughout all iterations, indicating that the fusion process is uniformly applied using the complete set of camera and LiDAR features. Conversely, the self-adaptive fusion rate exhibits considerable variation throughout the iterations. This variation suggests that the self-adaptive approach incurs reduced computational demands when processing the same frame count, specifically over 4000 epochs with 50 frames per epoch. Our findings indicate that while our model may not excel in metrics such as mAP and NDS, it shows a marked advantage in achieving a lower fusion rate. This suggests that our model can attain comparable performance levels with a diminished computational load. The capacity to sustain performance efficiency while reducing computational requirements underscores the effectiveness of our approach, rendering it a more resource-efficient solution.

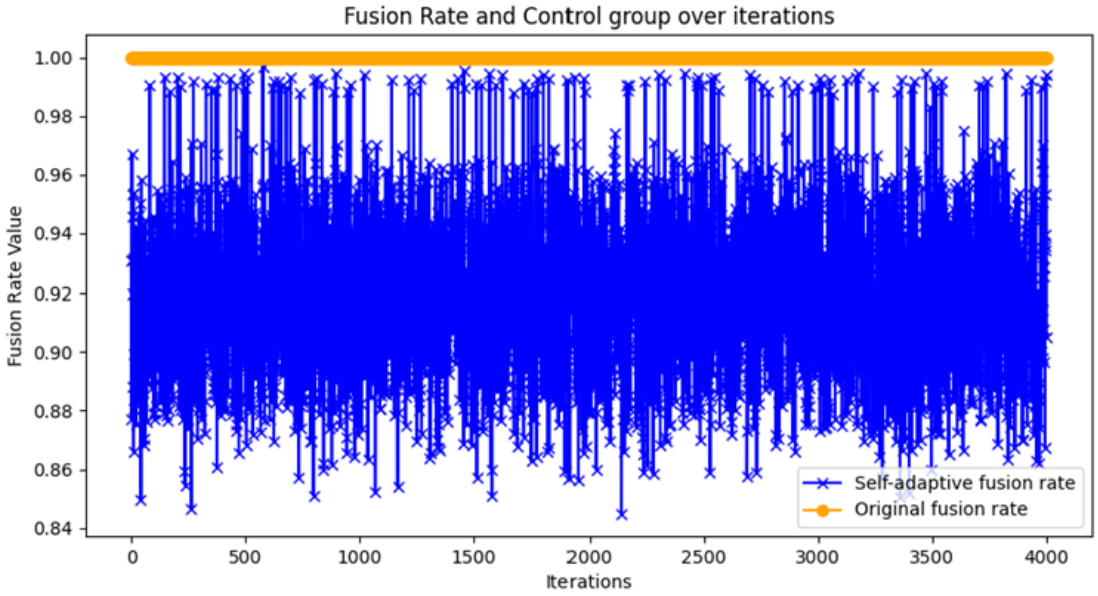

Figure 4.1: Changes in self-adaptive fusion rate and Original fusion rate at different numbers of iterations.

Table 4.2: Methods comparison in 3D object detection Camera and Radar fusion tasks on the nuScenes test dataset. Here, R50 represents ResNet-50 backbone.

Method	**Modality**	**NDS**	**mAP**	**mATE**	**mASE**	**mAOE**	**mAVE**	**mAAE**
TransCAR [Pang et al. 2023]	C+R	52.2	42.2	63.0	26.0	38.3	49.5	12.1
CRAFT [Kim et al. 2023]	C+R	48.7	34.1	58.7	25.7	42.4	46.1	10.8
FUTR3D-R50 [Chen et al. 2023]	C+R	50.3	38.9	65.4	27.9	38.8	41.7	17.9
SAMF3D-R50	C+R	48.5	38.9	65.5	27.9	38.9	38.8	17.9

Camera and Radar fusion. In our inaugural Radar and Camera fusion experiment, we embarked on an integrative approach by melding radar and camera features. Paralleling our strategy in the LiDAR and Camera fusion setup, we adopted ResNet50 as the foundational architecture for the camera features, while maintaining the Radar backbone's consistency with our previous experiments for comparative integrity. To ensure a controlled comparison, we meticulously re-implemented the baseline model, FUTR3D, in a Camera and Radar fusion context, dedicating 20 epochs to its training to establish a benchmark for our analyses. To obtain accurate ground truth radar point cloud data, we need to first capture, extract, and convert the real-time depth information from radar sensors into the corresponding average depth values. Following the established approach used in LiDAR-Camera fusion, we apply the method proposed by Li et al. 2023b to extract the depth information and compute the average depth accordingly.

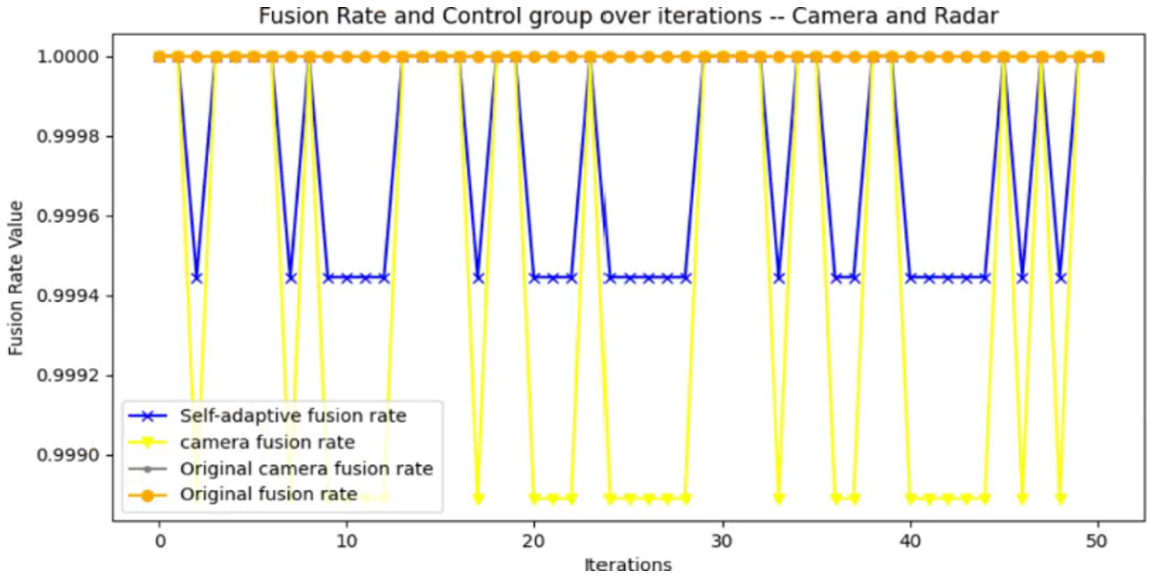

Figure 4.2: Dynamic visualization of the fusion rate's evolution in tandem with the progression of iterations.

In this experimental series, we positioned our results against the current state-of-the-art (SOTA) methods that support the fusion of Camera and Radar data. As depicted in Table 4.2, TransCAR [Pang et al. 2023] secured the leading position with an impressive NDS score of 52.2 and an mAP of 42.2. In contrast, our FUTR3D-ResNet50 exhibited a commendable performance, with NDS and mAP values consistently hovering around 38.9.

Furthermore, we conducted a comparative analysis between FUTR3D-ResNet50 and SAMF3D-ResNet50 with the existing SOTA works. This comparison revealed that while our NDS and mAP figures may not have scaled the peaks of SOTA, the sampling rate, a critical indicator of the efficiency in targeting sampling, demonstrated a marginal superiority for SAMF3D-ResNet50 in the Camera and Radar fusion domain. This observation suggests that our SAMF3D-ResNet50 model, when processing an equivalent number of video frames, can achieve a lower computational expense while sustaining a performance closely approximating the original, whether in LiDAR or Radar-Camera fusion scenarios. This underscores the efficiency and potential of the model for real-world applications where computational resources are measured.

Complexity Comparison. As shown in figure 4.1, it can be easily seen that the original fusion rate, which is shown in the orange line consists of orange dots, remains 1.00 because, in the original setting, all patches of the processed feature are considered to be fused. Therefore, the effective fusion rate which, also known as the fusion rate in the table is always the same as the actual fusion rate. In that case, the computation complexity will not reduce, which is always O(900) because every time the detector detects the object, the model will consider all the patches for each feature, i.e. each scene. However, when the self-adaptive sampler is adopted, the patches used for fusion will be selected according to the depth value. In that case, the effective fusion rate for the Camera feature will be reduced. For the LiDAR feature, the feature will not be re-sampled because, in our project, the LiDAR detector is the main detector, while the Camera detector is the minor detector. For minor detectors, it is not much use in good road conditions, so we conduct secondary sampling on the Camera's feature. It can be seen that the fusion rate rises and falls during the detection process and, maintains an average value of 0.84. This proves that the computation complexity of our model will be reduced to some extent, therefore reducing the burden of the detector in the real scene, which is mounted on the car. For Radar & Camera fusion, as shown in Figure 4.2, the original fusion rate is marked by yellow spots, while the yellow solid line delineates the specific fusion rate attributed to the Camera, reflecting the computational process dedicated to the Camera's features. The fusion rates for the Camera and the overall system are respectively denoted by the yellow and blue data points, each adorned with an 'x' marker. The proximity of the fusion rate to the value of one can be attributed to the depth of information provided by the Radar. We can also see that the fusion rate of Radar and Camera decreases from 1.0000 to 0.9994 in average. This shows that the fusion efficiency has been increased to some extents. Note that in the context of both LiDAR-Camera and Radar-Camera fusion strategies, our sampling of the Camera feature, which serves as our auxiliary sampler, is based on features extracted from either Radar or LiDAR data. Following the extraction of depth information, we proceed with the sampling process. It is important to note that the depth we refer to is actually the average depth value derived from the point cloud features. Consequently, the accuracy of this approach is somewhat lower. This explains the experimental results where the fusion rate only reaches 90% of the original fusion rate.

Additionally, When compared to the baseline performance of the model, which is at 67%, our model achieves a performance of 66.3%. Although there is a minor difference of 0.7 percentage points, this result is effectively equivalent to the baseline performance. This indicates that our model maintains a strong level of performance, closely matching the established benchmark.

Categorie. During actual driving, different types of targets such as cars, buses, and motorcycles have varying dimensions, a fact also reflected in the nuScenes dataset. Consequently, detection models respond differently to these targets. To investigate how the sampler and the use of an iTransformer to reverse input dimensions affect the detection outcomes for various targets, we conducted experiments as detailed in Table 4.3. We experimented with both LiDAR-only and LiDAR-Camera fusion setups on the FUTR3D dataset. The parameter configurations and training iterations were consistent with our previous experiments. It's important to note that in the experimental setup for L-SAMF3D, we incorporated the iTransformer solely during the feature processing stage, as there was feature information available for only one modality. This approach allows us to assess the impact of the iTransformer on object categories without the confounding influence of the sampler on the model's detection performance. To detail the experiment, we utilize the ResNet-50 architecture on the nuScenes validation dataset. Our evaluation focused on the Average Precision, i.e. AP values for each detected object category, aiming to demonstrate the positive impact on detecting targets that may appear irregularly or vary significantly in size within a scene. As detailed in Table 4.3, we examined the iTransformer's contribution to the model's detection capabilities by integrating it alone, without the influence of the adaptive sampler, and with LiDAR as the primary detection sensor.

Table 4.3: Detection performance analysis, we've highlighted the category-specific results, ensuring a fair comparison by including outcomes from systems using only Camera input. Values shown in the table are Average Precision (AP)

Modalities	Car	Truck	Bus	Ped	Motor
L-FUTR3D	89.4	63.3	97.9	90.3	23.8
L-SAMF3D	89.8	71.0	98.2	89.3	40.6
L+C-FUTR3D	95.0	69.2	99.0	92.7	78.0
L+C-SAMF3D	94.5	65.8	99.0	88.7	81.1

The experimental outcomes indicate a comprehensive improvement in performance across all object categories. Notably, the detection score for motorcycles saw the most significant boost, with a remarkable increase of 16.8 % over the baseline model. This was followed by the category of trunks, which exhibited a 7.7 % improvement. The enhancement in AP for these categories can be attributed to the relatively sparse representation of motorcycles and trunks in the dataset. By reorienting the embedded tokens, our approach facilitates the aggregation of global representations that are more attuned to variability, thereby optimizing the utility of the emerging attention mechanisms tailored for multivariate correlation. Simultaneously, the feed-forward network excels at acquiring generalizable representations, proficient in encoding a spectrum of variables from diverse historical series and decoding them to anticipate future patterns. Consequently, this strategy propels the model to achieve more robust outcomes.

Object distance. Our experiments were designed to assess the impact of our model on the detection performance of a specific category of objects, namely buses, across various distance intervals. The results, as depicted in Table 4.4, present the NDS (Normalized Detection Score) and AP (Average Precision) scores for the bus category about distance. The data reveals that the AP value tends to rise as the distance increases, and the application of the iTransformer further enhances the overall AP value compared to scenarios without it.

It is particularly noteworthy that as the distance extends, the gap in AP values between our baseline model equipped with the self-adaptive sampler (FU3D-S) and the model with both the Sampler and iTransformer becomes more pronounced. At closer distances, the inclusion of the iTransformer results in a 4% improvement in AP. However, this improvement diminishes as the distance grows; at 30 meters, the AP increase is a modest 2%, and beyond 30 meters, the AP only rises by 1%.

These findings underscore the effectiveness of the iTransformer in reducing computational load and enhancing detection precision, especially for objects that are infrequently encountered and near the ego-car. The diminishing returns in AP improvement at greater distances suggest a nuanced relationship between detection accuracy and the application of advanced computational techniques.

Table 4.4: Detection Performance for Buses at Different Distances

	NDS	**AP(0-10m)**	**AP(10-20m)**	**AP(20-30m)**	**AP(30m- inf)**
Bus (**SA3D-S**)	0.73	0.49	0.77	0.908	0.93
Bus **(SA3D-S+i)**	0.72	**0.53**	**0.79**	**0.92**	**0.94**

4.5 Ablation Study

In this part, as shown in Table 4.5, we explore the effects of incorporating the Sampler (S) and iTransformer (i) on detection performance. Consistent with the approach outlined in Chen et al. 2023, we have chosen ResNet50 He et al. 2016b, which has the advantage of effectively tackling the residual between outputs instead of learning the output itself also alleviates the problem of gradient explosion as our backbones of interest. To rigorously assess the impact of our self-adaptive Sampler and iTransformer, we conducted a series of model training sessions, each with a distinct configuration. Before each training iteration, we made a deliberate choice regarding the application of the Sampler by modifying the fusion strategy within the Cross-Attention component of our framework. Specifically, if the fusion strategy opted for non-fusion, we merged all camera-derived feature patches. On the other hand, if fusion was selected, we anchored our fusion around the central patch, expanding the fusion area following a predefined rate. For the dimension inversion process, we made a binary decision on whether to incorporate the third dimension into the attention calculation module. In the subsequent paragraph, we will delve into a comprehensive analysis of our experimental outcomes. This analysis will shed light on the determinants behind the observed performance metrics and explore the broader implications of our findings within the context of our research.

Table 4.5: Detection performance of models utilizing ResNet50 backbones, under different configurations of Self-adaptive Sampler (S) and iTransformer (i).

Backbone	S	i	mAP	NDS	mATE	mASE	mAOE	mAVE	mAAE
ResNet50	×	×	0.595	0.662	0.326	0.259	0.290	0.282	0.189
ResNet50	×	✓	0.613	0.670	0.321	0.260	0.282	0.323	0.225
ResNet50	✓	✓	0.595	0.663	0.327	0.259	0.290	0.282	0.190

For ResNet50, it can be seen that without the Sampler and the iTransformer, the model achieves an mAP of 59.5 % and an NDS of 66.2 %. Also when only the iTransformer is added, it is clear to see that both the value of mAP and NDS is slightly increased. This proves that our iTransformer can perform the target detection task more accurately while utilizing resources more efficiently. This enhancement stems from the optimization of the model's time complexity through dimensional inversion, a technique that allows the model to concentrate more intently on the core task of target detection, thereby enhancing its overall performance. It's important to highlight that the integration of both the iTransformer and the self-adaptive Sampler in our detection framework while maintaining error metrics comparable to the control group that lacks these enhancements, can sometimes result in a performance level that mirrors the control group. This observation is particularly pertinent when considering the impact of the self-adaptive Sampler on the detection process. The self-adaptive Sampler is designed to refine the detection process by adjusting the sampling strategy based on depth information. However, this additional layer of complexity can introduce potential pitfalls. Specifically, when the Sampler is engaged, it performs secondary sampling, which can lead to a drop in performance if not executed accurately. The crux of the issue lies in the depth calculation: any inaccuracies in determining the depth can result in the re-sampling process inadvertently excluding regions that contain key objects.

4.6 Summary

We have conducted extensive experiments using the nuScenes dataset, adhering strictly to the settings and configurations outlined in Chen et al. 2023. In terms of dataset partitioning, we have also maintained the same configuration as detailed in ibid. For our experiments, we designate either LiDAR or Radar as the primary detector, with the Camera serving as the secondary detector. Depending on the chosen primary detector, we select various data categories from the nuScenes dataset to simulate real-world driving scenarios. Our experimental results indicate that our proposed model significantly reduces computational complexity when either LiDAR or Radar is used as the main detector, as compared to the approach in ibid. Moreover, our model achieves performance on par with that reported in ibid. This comparison suggests that our proposed model offers superior efficiency over the baseline model. Furthermore, our ablation study reveals that the application of the iTransformer leads to a noticeable improvement in the model's overall performance, indicating that the iTransformer indeed enhances the model's performance.

Chapter 5

Limitations

We have identified several limitations within our current model. Initially, our depth-aware self-adaptive sampler relies on the average depth rather than the specific depth information of the nuScenes target associated with each segment of the feature map. This approach can result in an inaccurate fusion rate, particularly when the local traffic environment is intricate and the movement of traffic participants is highly unpredictable. Secondly, our fusion configuration is constrained by computational resources and hardware capabilities, which currently only support the integration of two modalities at a time. This limitation precludes us from exploring the potential of fusing Lidar, Camera, and Radar data simultaneously. However, we are optimistic that advancements in technology will address this challenge in the future. Thirdly, our model employs a two-stage approach, which, while enhancing accuracy and adaptability, introduces certain complexities. The training process can be more intricate, and there is an inherent risk of overfitting. Additionally, the model requires meticulous parameter tuning due to the interplay between the two stages. Furthermore, the direct saving of detection results may hinder the application of our model to other downstream tasks, such as 3D object tracking or trajectory prediction. This limitation poses a barrier to integrating our model with other state-of-the-art tracking systems.

5.1 Key Limitations

Apart from the abovementioned limitations, some key limitations should be acknowledged. First, the study used only a single dataset, which may not adequately capture the diversity and complexity of real-world scenarios. This limitation could potentially affect the generalizability of the findings. Additionally, the backbone architecture employed was limited to ResNet50, and the compatibility of the proposed method with other backbone architectures has not been examined. This raises concerns about the adaptability of the approach across different network structures. Lastly, the experimental results are only compared to one baseline model. More baseline models should be compared to ensure the model is adaptable enough across all mainstream 3D Object Detection models in autonomous driving perception systems.

Chapter 6

Conclusion and Future work

In conclusion, as autonomous vehicles rely on sensor fusion to achieve higher precision in detection, researchers are continually exploring ways to enhance the detection capabilities of models. In this context, we have identified a method to reduce computational complexity during both the feature processing and modality fusion stages. Our approach introduces a self-adaptive Sampler that dynamically integrates features from one detector with the average depth information extracted from another modality, providing a more accurate representation of real-time driving scenarios. Additionally, we have developed the iTransformer to reorient the dimensions of the embeddings before they are fed into the multi-head self-attention and feed-forward layers of the Transformer encoder, thereby enhancing feature extraction and processing.

Our experiments on the nuScenes test and validation datasets demonstrate that our model can attain baseline detection performance. Notably, our Sampler is capable of moderately reducing the fusion rate, which in turn decreases computational complexity. Furthermore, we have observed that the application of the iTransformer notably improves the detection of targets that are less frequently encountered during actual driving, particularly when they are near the vehicle.

Looking ahead, we plan to train our model on a broader range of datasets to assess its generalizability across more diverse scenarios. We also intend to fine-tune VovNet to better accommodate our proposed Sampler. Additionally, we have recognized the need to address the model's current challenges in detecting small and static objects, a problem we are committed to solving in future research. Furthermore, we envision our model being extended to a broader range of tasks in the future, such as enhancing the efficiency of multimodal object tracking and improving the efficiency of object segmentation. We aspire for this paper to serve as a source of inspiration for self-adaptive fusion techniques in multi-modal fusion and scene perception.

Bibliography

Bijelic, Mario et al. (2020a). 'Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather'. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, pp. 11679–11689. DOI: 10.1109/CVPR42600.2020.01170. URL: https://openaccess.thecvf.com/content_CVPR_2020/html/Bijelic_Seeing_Through_Fog_Without_Seeing_Fog_Deep_Multimodal_Sensor_Fusion_CVPR_2020_paper.html.

– (2020b). 'Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather'. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, pp. 11679–11689. DOI: 10.1109/CVPR42600.2020.01170. URL: https://openaccess.thecvf.com/content_CVPR_2020/html/Bijelic_Seeing_Through_Fog_Without_Seeing_Fog_Deep_Multimodal_Sensor_Fusion_CVPR_2020_paper.html.

Bokade, Rohit et al. (2021). 'A cross-disciplinary comparison of multimodal data fusion approaches and applications: Accelerating learning through trans-disciplinary information sharing'. In: *Expert Systems with Applications* 165, p. 113885. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2020.113885. URL: https://www.sciencedirect.com/science/article/pii/S0957417420306886.

Caesar, Holger et al. (2019). 'nuScenes: A multimodal dataset for autonomous driving'. In: *CoRR* abs/1903.11027. arXiv: 1903.11027. URL: http://arxiv.org/abs/1903.11027.

Cai, Yanlu et al. (2024). 'FusionFormer: A Concise Unified Feature Fusion Transformer for 3D Pose Estimation'. In: *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*. Ed. by Michael J. Wooldridge, Jennifer G. Dy and Sriraam Natarajan. AAAI Press, pp. 900–908. DOI: 10.1609/AAAI.V38I2.27849. URL: https://doi.org/10.1609/aaai.v38i2.27849.

Chen, Kai et al. (2019). 'MMDetection: Open MMLab Detection Toolbox and Benchmark'. In: *CoRR* abs/1906.07155. arXiv: 1906.07155. URL: http://arxiv.org/abs/1906.07155.

Chen, Xuanyao et al. (2023). 'FUTR3D: A Unified Sensor Fusion Framework for 3D Detection'. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023 - Workshops, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, pp. 172–181. DOI: 10.1109/CVPRW59228.2023.00022. URL: https://doi.org/10.1109/CVPRW59228.2023.00022.

Ding, Longyuan et al. (2021). 'Pointnet: Learning Point Representation for High-Resolution Remote Sensing Imagery Land-Cover Classification'. In: *IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2021, Brussels, Belgium, July 11-16, 2021*. IEEE, pp. 4956–4959. DOI: 10.1109/IGARSS47720.2021.9554009. URL: https://doi.org/10.1109/IGARSS47720.2021.9554009.

Dosovitskiy, Alexey et al. (2021). 'An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale'. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. URL: https://openreview.net/forum?id=YicbFdNTTy.

Geiger, Andreas, Philip Lenz and Raquel Urtasun (2012). 'Are we ready for autonomous driving? The KITTI vision benchmark suite'. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361. DOI: 10.1109/CVPR.2012.6248074.

He, Kaiming et al. (2016a). 'Deep Residual Learning for Image Recognition'. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, pp. 770–778. DOI: 10.1109/CVPR.2016.90. URL: https://doi.org/10.1109/CVPR.2016.90.

– (2016b). 'Deep Residual Learning for Image Recognition'. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, pp. 770–778. DOI: 10.1109/CVPR.2016.90. URL: https://doi.org/10.1109/CVPR.2016.90.

Huang, Junjie and Guan Huang (2022). 'BEVDet4D: Exploit Temporal Cues in Multi-camera 3D Object Detection'. In: *CoRR* abs/2203.17054. DOI: 10.48550/ARXIV.2203.17054. arXiv: 2203.17054. URL: https://doi.org/10.48550/arXiv.2203.17054.

Huang, Junjie et al. (2021). 'BEVDet: High-performance Multi-camera 3D Object Detection in Bird-Eye-View'. In: *CoRR* abs/2112.11790. arXiv: 2112.11790. URL: https://arxiv.org/abs/2112.11790.

Huang, Yu and Yue Chen (2020). *Autonomous Driving with Deep Learning: A Survey of State-of-Art Technologies*. arXiv: 2006.06091 [cs.CV]. URL: https://arxiv.org/abs/2006.06091.

Kim, Youngseok et al. (2023). 'CRAFT: Camera-Radar 3D Object Detection with Spatio-Contextual Fusion Transformer'. In: *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*. Ed. by Brian Williams, Yiling Chen and Jennifer Neville. AAAI Press, pp. 1160–1168. DOI: 10.1609/AAAI.V37I1.25198. URL: https://doi.org/10.1609/aaai.v37i1.25198.

Kurniawan, Irfan Tito and Bambang Riyanto Trilaksono (2023). 'ClusterFusion: Leveraging Radar Spatial Features for Radar-Camera 3D Object Detection in Autonomous Vehicles'. In: *IEEE Access* 11, pp. 121511–121528. DOI: 10.1109/ACCESS.2023.3328953. URL: https://doi.org/10.1109/ACCESS.2023.3328953.

LeCun, Yann et al. (1989). 'Backpropagation Applied to Handwritten Zip Code Recognition'. In: *Neural Comput.* 1.4, pp. 541–551. DOI: 10.1162/NECO.1989.1.4.541. URL: https://doi.org/10.1162/neco.1989.1.4.541.

Li, Yinhao et al. (2023a). 'BEVDepth: Acquisition of Reliable Depth for Multi-View 3D Object Detection'. In: *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*. Ed. by Brian Williams, Yiling Chen and Jennifer Neville. AAAI Press, pp. 1477–1485. DOI: 10.1609/AAAI.V37I2.25233. URL: https://doi.org/10.1609/aaai.v37i2.25233.

Li, Zhiqi et al. (2022). 'BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers'. In: *CoRR* abs/2203.17270. DOI: 10.48550/ARXIV.2203.17270. arXiv: 2203.17270. URL: https://doi.org/10.48550/arXiv.2203.17270.

Li, Zhiqi et al. (2023b). 'FB-BEV: BEV Representation from Forward-Backward View Transformations'. In: *CoRR* abs/2308.02236. DOI: 10.48550/ARXIV.2308.02236. arXiv: 2308.02236. URL: https://doi.org/10.48550/arXiv.2308.02236.

Liu, Yingfei et al. (2022). 'PETR: Position Embedding Transformation for Multi-view 3D Object Detection'. In: *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVII*. Ed. by Shai Avidan et al. Vol. 13687. Lecture Notes in Computer Science. Springer, pp. 531–548. DOI: 10.1007/978-3-031-19812-0_31. URL: https://doi.org/10.1007/978-3-031-19812-0_31.

Liu, Yingfei et al. (2023a). 'PETRv2: A Unified Framework for 3D Perception from Multi-Camera Images'. In: *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, pp. 3239–3249. DOI: 10.1109/ICCV51070.2023.00302. URL: https://doi.org/10.1109/ICCV51070.2023.00302.

Liu, Yong et al. (2024). 'iTransformer: Inverted Transformers Are Effective for Time Series Forecasting'. In: *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. URL: https://openreview.net/forum?id=JePfAI8fah.

Liu, Zhijian et al. (2023b). 'BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation'. In: *IEEE International Conference on Robotics and Automation, ICRA 2023, London, UK, May 29 - June 2, 2023*. IEEE, pp. 2774–2781. DOI: 10.1109/ICRA48891.2023.10160968. URL: https://doi.org/10.1109/ICRA48891.2023.10160968.

Liu, Zhun et al. (2018). 'Efficient low-rank multimodal fusion with modality-specific factors'. In: *arXiv preprint arXiv:1806.00064*.

Loshchilov, Ilya and Frank Hutter (2019). 'Decoupled Weight Decay Regularization'. In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

Palladin, Edoardo et al. (2024). 'SAMFusion: Sensor-Adaptive Multimodal Fusion for 3D Object Detection in Adverse Weather'. In: *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXI*. Milan, Italy: Springer-Verlag, pp. 484–503. ISBN: 978-3-031-73029-0. DOI: 10.1007/978-3-031-73030-6_27. URL: https://doi.org/10.1007/978-3-031-73030-6_27.

Pang, Su, Daniel D. Morris and Hayder Radha (2023). 'TransCAR: Transformer-Based Camera-and-Radar Fusion for 3D Object Detection'. In: *IROS*, pp. 10902–10909. DOI: 10.1109/IROS55552.2023.10341793. URL: https://doi.org/10.1109/IROS55552.2023.10341793.

Philion, Jonah and Sanja Fidler (2020). '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*. Ed. by Andrea Vedaldi et al. Vol. 12359. Lecture Notes in Computer Science. Springer, pp. 194–210. DOI: 10.1007/978-3-030-58568-6_12. URL: https://doi.org/10.1007/978-3-030-58568-6_12.

Stäcker, Lukas et al. (2023). 'RC-BEVFusion: A Plug-In Module for Radar-Camera Bird's Eye View Feature Fusion'. In: *Pattern Recognition - 45th DAGM German Conference, DAGM GCPR 2023, Heidelberg, Germany, September 19-22, 2023, Proceedings*. Ed. by Ullrich Köthe and Carsten Rother. Vol. 14264. Lecture Notes in Computer Science. Springer, pp. 178–194. DOI: 10.1007/978-3-031-54605-1_12. URL: https://doi.org/10.1007/978-3-031-54605-1_12.

Tan, Sin Yong et al. (2022). 'Multimodal sensor fusion framework for residential building occupancy detection'. In: *Energy and Buildings* 258, p. 111828. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2021.111828. URL: https://www.sciencedirect.com/science/article/pii/S0378778821011129.

Vaswani, Ashish et al. (2017). 'Attention is All you Need'. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al., pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Wang, Tai et al. (2021a). 'FCOS3D: Fully Convolutional One-Stage Monocular 3D Object Detection'. In: *IEEE/CVF International Conference on Computer Vision Workshops, ICCVW 2021, Montreal, BC, Canada, October 11-17, 2021*. IEEE, pp. 913–922. DOI: 10.1109/ICCVW54120.2021.00107. URL: https://doi.org/10.1109/ICCVW54120.2021.00107.

Wang, Yue and Justin M. Solomon (2021). 'Object DGCNN: 3D Object Detection using Dynamic Graphs'. In: *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. Ed. by Marc'Aurelio Ranzato et al., pp. 20745–20758. URL: https://proceedings.neurips.cc/paper/2021/hash/ade1d98c5ab2997e867b1151a5c5028d-Abstract.html.

Wang, Yue et al. (2021b). 'DETR3D: 3D Object Detection from Multi-view Images via 3D-to-2D Queries'. In: *Conference on Robot Learning, 8-11 November 2021, London, UK*. Ed. by Aleksandra Faust, David Hsu and Gerhard Neumann. Vol. 164. Proceedings of Machine Learning Research. PMLR, pp. 180–191. URL: https://proceedings.mlr.press/v164/wang22b.html.

Woo, Sanghyun et al. (2018). 'CBAM: Convolutional Block Attention Module'. In: *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*. Ed. by Vittorio Ferrari et al. Vol. 11211. Lecture Notes in Computer Science. Springer, pp. 3–19. DOI: 10.1007/978-3-030-01234-2_1. URL: https://doi.org/10.1007/978-3-030-01234-2_1.

Yan, Junjie et al. (2023). 'Cross Modal Transformer: Towards Fast and Robust 3D Object Detection'. In: *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, pp. 18222–18232. DOI: 10.1109/ICCV51070.2023.01675. URL: https://doi.org/10.1109/ICCV51070.2023.01675.

Yan, Xu et al. (2024). *Forging Vision Foundation Models for Autonomous Driving: Challenges, Methodologies, and Opportunities*. arXiv: 2401.08045 [cs.CV]. URL: https://arxiv.org/abs/2401.08045.

Yoo, Jin Hyeok et al. (2020). '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*. Ed. by Andrea Vedaldi et al. Vol. 12372. Lecture Notes in Computer Science. Springer, pp. 720–736. DOI: 10.1007/978-3-030-58583-9_43. URL: https://doi.org/10.1007/978-3-030-58583-9_43.

Zhang, Cheng et al. (2025). 'MixedFusion: An Efficient Multimodal Data Fusion Framework for 3-D Object Detection and Tracking'. In: *IEEE Transactions on Neural Networks and Learning Systems* 36.1, pp. 1842–1856. DOI: 10.1109/TNNLS.2023.3325527.

Zhou, Yin and Oncel Tuzel (2018). 'VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection'. In: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, pp. 4490–4499. DOI: 10.1109/CVPR.2018.00472. URL: http://openaccess.thecvf.com/content_cvpr_2018/html/Zhou_VoxelNet_End-to-End_Learning_CVPR_2018_paper.html.

Zhu, Xizhou et al. (2021). 'Deformable DETR: Deformable Transformers for End-to-End Object Detection'. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. URL: https://openreview.net/forum?id=gZ9hCDWe6ke.