DTU Electro
Department of Electrical and Photonics Engineering

Camera Radar Fusion For 3D Object Detection

Enhancing Sparsity Of Radar Pointclouds Using Diffusion Models

Master Thesis

Camera Radar Fusion For 3D Object Detection
Enhancing Sparsity Of Radar Pointclouds Using Diffusion Models

Master Thesis
March, 2026

By
Hassan Hotait

Cover photo: Vibeke Hempler, 2012
Published by: DTU, Department of Electrical and Photonics Engineering, Ørsteds Plads, Building 343, 2800 Kgs. Lyngby Denmark
https://electro.dtu.dk/

Approval

This thesis has been created in the context of a master thesis contract of the Infineon Technologies AG at its headquarters in Neubiberg, Germany.

Disclaimer

In the spirit of academic transparency, I hereby disclose the use of Generative Artificial Intelligence tools to support writing the thesis report. These tools were employed primarily support sentence phrasing, enhance textual clarity and assist in document structuring. All external sources and references are appropriately cited in accordance with academic citation standards. This declaration aims to maintain full intellectual honesty and transparency regarding the research and writing process.

Hassan Hotait - s203211

Hassan Hotait

Signature

07/04/2026

Date

Abstract

Accurate three-dimensional perception is a fundamental requirement for autonomous driving systems. While LiDAR sensors provide dense geometric information that enables reliable 3D object detection, they are expensive and power intensive. Radar sensors offer a cost-effective alternative with advantages such as long sensing range, robustness to adverse weather conditions, and direct velocity measurements. However, radar measurements are extremely sparse and noisy, which limits their effectiveness for radar-only 3D perception.

This thesis investigates the use of diffusion models to enhance sparse radar point clouds for radar-only 3D object detection. The proposed approach learns a mapping from radar observations to denser LiDAR-like bird's-eye-view (BEV) representations. To this end, radar and LiDAR point clouds from the nuScenes dataset are preprocessed into BEV occupancy and height maps, and a conditional diffusion model is trained to reconstruct LiDAR BEV targets from radar inputs. The generated representations are then evaluated using a downstream 3D object detection pipeline based on CenterPoint.

Experimental results show that the choice of intermediate representation and preprocessing pipeline is critical. In particular, preserving height information in the BEV representation is essential for downstream 3D detection, while ground removal and BEV resolution substantially influence the achievable upper bound. Among the evaluated diffusion variants, the occupancy-only denoiser achieved the best performance with an occupancy IoU of 46%/28% (train/validation) and 6% mAP in downstream detection. Joint occupancy-and-height prediction produced inferior results, largely due to loss-scaling issues, and end-to-end training of the denoiser and detector further reduced performance to 2% mAP.

Overall, the results show that diffusion models can improve the global spatial structure of sparse radar representations, but recovering the fine-grained geometry required for accurate 3D object detection remains challenging. The findings highlight both the potential and the current limitations of diffusion-based radar enhancement for autonomous driving perception.

Acknowledgements

I would like to express my sincere gratitude to my supervisors for their guidance and support throughout this thesis.

I am especially thankful to my supervisor Huawei Sun, for providing valuable insights, technical discussions, and continuous support during the course of this work. Her expertise and feedback were instrumental in shaping the direction of this project.

I would also like to thank my academic supervisor, Dimitrios Papadopoulos, for his guidance, constructive feedback, and support throughout the thesis process.

Contents

Preface . . . ii
Preface . . . ii
Abstract . . . iii
Acknowledgements . . . iv

1 Introduction 1
1.1 Current Approaches to 3D Detection . . . 1
1.2 Challenges in Radar-Based Perception . . . 2
1.3 Motivation and Need for Improved Radar Representations . . . 2
1.4 Thesis Overview . . . 3

2 Related Work 5
2.1 3D Detection . . . 5
2.2 SOTA Camera-Radar Fusion 3D Detection . . . 6
2.3 SOTA Radar only 3D Detection . . . 8
2.4 Diffusion Based Image Generators . . . 9
2.4.1 Sampling Stage Design Choices . . . 10
2.4.2 Training Stage Design Choices . . . 12
2.5 Radar-Diffusion Super Resolution Systems . . . 13

3 Research Gap & Thesis Scope 15

4 nuScenes: A multimodal dataset for autonomous driving 17
4.1 Preprocessing Pipeline . . . 18
4.1.1 Radar Multi-Sweep + Transformation to LiDAR Frame . . . 19
4.1.2 LiDAR Multi-Sweep + Ego & Object-Motion Compensation . . . 20
4.1.3 Ground Removal via Patchwork++ . . . 22
4.1.4 Point Cloud to BEV (Occupancy + Height Map) . . . 22

5 Methodology 25
5.1 Stage 1: Radar BEV Diffusion . . . 25
5.1.1 Intermediate Data Representation & Preprocessing Pipeline . . . 26
5.1.2 Loss Functions . . . 27
5.1.3 Cut & Mix Augmentation . . . 30
5.2 Stage 2: Centerpoint as benchmark for point-based 3D Detection . . . 31
5.2.1 Feature Extraction Pipeline . . . 31
5.2.2 Loss Functions . . . 31
5.2.3 Class-Balanced Grouping & Sampling . . . 32
5.3 End-To-End: Radar BEV Diffusion + 3D Detection . . . 34
5.4 Implementation Details . . . 36
5.5 Evaluation Metrics . . . 36

6 Experimental Results 39
6.1 Qualitative . . . 40
6.2 Occupancy Denoiser . . . 41
6.3 Cut & Mix Augmentation . . . 42
6.4 Occupancy & Height Map Denoiser . . . 42

6.5 Radar Height Map in Conditioning Signal 43
6.6 End-to-End Training 43

7 Future Work **45**
7.1 Upsampling Model 45
7.2 Diffusion In Enhanced Voxel Feature Space 47

8 Conclusion **49**

Bibliography **51**

1 Introduction

Autonomous driving systems rely on accurate perception of the surrounding environment in order to safely navigate complex traffic scenarios. A core component of the perception stack is **3D object detection**, which aims to identify and localize objects such as vehicles, pedestrians, cyclists, and traffic infrastructure in three-dimensional space. Reliable 3D perception enables autonomous vehicles to reason about scene geometry, estimate motion, and plan safe trajectories.

Modern perception systems typically combine multiple sensing modalities, including cameras, LiDAR, and radar. Cameras provide rich semantic information and high spatial resolution, making them well suited for object recognition and scene understanding. LiDAR sensors directly measure scene geometry through dense 3D point clouds and therefore form the basis of many state-of-the-art 3D detection systems. Radar sensors, on the other hand, offer several complementary advantages, including long sensing range, robustness to adverse weather conditions, and direct Doppler-based velocity measurements [1]. These properties make radar attractive for safety-critical perception, especially in conditions where camera and LiDAR performance may degrade.

At the same time, radar remains difficult to exploit effectively for 3D detection. Compared to LiDAR, radar point clouds are extremely sparse and noisy, and they provide much weaker spatial structure. This sparsity creates a substantial gap between the physical advantages of radar sensing and its practical usefulness in modern deep learning pipelines. As a result, improving radar representations has become an important research problem for both radar-only perception and multimodal fusion systems.

This thesis studies that problem through the lens of diffusion-based generative modeling. More specifically, it investigates whether diffusion models can reconstruct denser LiDAR-like bird's-eye-view (BEV) representations from sparse radar observations, and whether these enhanced representations improve downstream 3D object detection. The broader literature context is reviewed in Chapter 2, the research gap and thesis scope are defined in Chapter 3, the dataset and preprocessing pipeline are described in Chapter 4, and the proposed methodology and experiments are presented in Chapters 5 and 6.

1.1 Current Approaches to 3D Detection

Early deep learning approaches for 3D object detection relied primarily on LiDAR point clouds and often transformed the raw points into structured representations such as voxels or bird's-eye-view feature maps. These representations enable efficient processing with convolutional neural networks while preserving the spatial layout of the scene. Among them, BEV representations have become particularly popular because they simplify localization in the ground plane and align naturally with driving scenarios.

A major line of progress in 3D detection has been the transition from anchor-based to anchor-free center-based methods. Methods such as CenterNet formulate detection as keypoint prediction, representing objects by their centers instead of predefined anchor templates [2]. Building on this idea, CenterPoint extends the center-based formulation to 3D object detection and tracking by predicting object centers in BEV space together with associated box attributes such as size, orientation, and velocity [2]. Due to its strong performance and clean BEV formulation, CenterPoint has become an important baseline for autonomous driving datasets such as nuScenes [1].

In parallel, multimodal perception systems have become increasingly common. Camera–LiDAR and camera–radar fusion pipelines seek to combine the semantic richness of images with the geometric and motion cues provided by range sensors. In particular, many recent camera–radar fusion methods perform fusion directly in BEV space, since this avoids repeated projection losses and allows the use of strong BEV-based detection heads. These topics are reviewed in detail in Sections 2.1 and 2.2.

1.2 Challenges in Radar-Based Perception

Despite its attractive sensing properties, radar poses several challenges for 3D perception. First, radar point clouds are much sparser than LiDAR, often containing only a small number of returns per object. Second, radar measurements are noisy and may contain clutter, ghost detections, and ambiguities caused by multipath reflections and limited angular resolution. Third, the vertical structure of the scene is weakly captured compared to LiDAR, which makes it difficult to recover detailed 3D geometry directly from radar data alone.

Because of these limitations, many existing radar-based systems rely heavily on stronger modalities such as cameras or LiDAR to provide spatial structure. Even in fusion systems, radar often contributes secondary cues such as motion or range consistency, rather than serving as the dominant source of geometric information. At the same time, relying on LiDAR increases system cost and complexity, while reducing the appeal of radar as a lightweight and robust sensing modality.

Recent work has shown that improving the radar representation itself can substantially improve detection performance. In particular, RadarDistill demonstrated that transferring LiDAR knowledge to a radar student can boost radar-only 3D detection to 20.5% mAP on nuScenes, far above earlier radar-only baselines, and can also improve camera–radar fusion when integrated into a larger pipeline [3]. This result suggests that radar sparsity is not merely a nuisance, but rather a central bottleneck whose mitigation can benefit both radar-only and multimodal perception. A more detailed review of these methods is given in Sections 2.2 and 2.3.

1.3 Motivation and Need for Improved Radar Representations

The key motivation of this thesis is that radar sensing offers valuable physical advantages, yet its sparse and noisy output prevents current perception systems from fully exploiting them. If a model could reconstruct denser and more structured scene representations from radar, radar-only perception could become more competitive and fusion pipelines could benefit from stronger radar features.

Diffusion models provide a promising framework for this problem. Since the introduction of denoising diffusion probabilistic models, diffusion-based methods have demonstrated strong performance in image synthesis, restoration, and conditional generation tasks [4]. The EDM framework further showed that careful choices in parameterization, noise sampling, and preconditioning can substantially improve the stability and effectiveness of diffusion-based generation [5]. These properties are particularly appealing for radar enhancement, where the task can be interpreted as reconstructing a structured target from a sparse and noisy conditioning signal.

Motivated by this, the present thesis explores diffusion-based radar enhancement in BEV space. More specifically, it studies whether sparse radar point clouds can be transformed into LiDAR-like BEV occupancy and height representations, and whether these generated

representations are useful for downstream 3D detection. To support this, Chapter 4 introduces a dedicated preprocessing pipeline for radar and LiDAR multi-sweep aggregation, motion compensation, ground removal, and BEV conversion. Chapter 5 then presents the proposed diffusion-based methodology, including both a two-stage and an end-to-end formulation.

1.4 Thesis Overview

The proposed framework consists of two main stages. In the first stage, a conditional diffusion model is trained to reconstruct LiDAR BEV representations from radar observations. Since nuScenes does not provide radar azimuth heatmaps, the model is conditioned on radar BEV occupancy maps derived from radar point clouds. During development, it became clear that a pure occupancy representation is insufficient for 3D detection because it collapses important height information; therefore, the work also investigates joint occupancy-and-height representations. This design choice and its implications are discussed in Sections 5.1.1 and 6.4.

In the second stage, the generated BEV representations are evaluated through a downstream 3D object detection pipeline based on CenterPoint [2]. This allows the quality of the generated radar representations to be measured not only in terms of reconstruction metrics such as IoU and height MSE, but also in terms of their practical usefulness for object detection. The experimental setup and evaluation metrics are described in Sections 5.2 and 5.5, and the main results are presented in Chapter 6.

The remainder of this thesis is organized as follows. Chapter 2 reviews prior work on 3D detection, camera–radar fusion, radar-only detection, and diffusion-based generative models. Chapter 3 defines the research gap and explains the scope of the thesis. Chapter 4 presents the nuScenes dataset and the preprocessing pipeline used to construct the BEV representation. Chapter 5 describes the proposed methodology, including the diffusion model, the CenterPoint benchmark, and the end-to-end training formulation. Chapter 6 presents the experimental results and ablation studies. Chapter 7 discusses future work, including higher-resolution upsampling and diffusion in enhanced voxel feature space, and Chapter 8 concludes the thesis.

2 Related Work

Recent advances in autonomous driving have significantly improved 3D object detection through the use of multi-modal sensor systems. Cameras provide rich semantic information, while radar sensors offer robustness to adverse weather and reliable velocity measurements. However, radar data is inherently sparse and noisy, which limits its effectiveness when directly fused with camera features for accurate 3D perception. This section reviews existing work relevant to this thesis. We first provide an overview of 3D object detection approaches with a focus on perception systems used in autonomous driving. Next, we discuss camera–radar fusion methods and the strategies used to combine complementary sensor modalities. We then examine radar-only approaches that attempt to address the sparsity limitations of radar data. Finally, we review diffusion-based generative models and recent work applying diffusion models to radar data enhancement, which forms the foundation of the approach proposed in this thesis.

2.1 3D Detection

Early LiDAR-based 3D object detectors commonly relied on anchor-based detection heads, which enumerate predefined 3D bounding box templates with different orientations, sizes, and aspect ratios. Similar to 2D object detection, these approaches classify anchors and regress offsets to fit the ground-truth boxes. While effective, anchor-based designs introduce large numbers of candidate boxes and require careful anchor engineering to handle the wide variability in object orientation and scale present in 3D [6].

More recent methods adopt anchor-free, center-based detection heads that represent objects through their geometric centers rather than predefined bounding boxes. In these approaches, the detector predicts a dense heatmap over a bird's-eye-view (BEV) feature map, where peaks correspond to object centers. For each detected center, the network directly regresses object attributes such as 3D size, vertical position, orientation, and optionally velocity, enabling the reconstruction of the full 3D bounding box. This formulation simplifies the detection problem by removing the need for anchor generation and by avoiding explicit enumeration of object orientations [2].

Center-based frameworks further benefit from their compatibility with standard convolutional detection heads. After a backbone network produces BEV features from the point cloud, a lightweight head predicts the center heatmap along with additional regression branches for box parameters. Some approaches further introduce a second-stage refinement step, where points associated with each predicted object are aggregated to refine localization and box parameters. This two-stage formulation improves localization accuracy while maintaining the efficiency of the center-based detection paradigm [2].

Center-based BEV detectors dominate nuScenes [1] pipelines due to their efficient representation of 3D boxes as BEV center keypoints. CenterPoint [2] detects object centers via a keypoint head, and regresses size, orientation, and velocity, while a second stage refines estimates with additional point features. This ”center-as-point” formulation strongly influenced later fusion and camera-based works that reuse CenterPoint-style heads once a BEV representation is available.

2.2 SOTA Camera-Radar Fusion 3D Detection

Camera–radar fusion methods aim to exploit the complementary properties of cameras and radar sensors. Cameras provide high-resolution appearance information useful for object recognition and localization in the image plane, while radar in nuScenes [1] is sparse but provides direct Doppler/velocity cues motivating fusion approaches that improve mAVE and robustness to lighting/weather. Modern radar–camera fusion increasingly performs fusion in BEV to avoid repeated projection losses and to align with common detection heads. Despite promising results, many approaches struggle to fully utilize radar information due to the sparsity and noise characteristics of radar measurements. By reviewing the state-of-the art, we observe only very few methods target the sparse radar features which we believe is the current bottleneck in advancing camera-radar detection on nuScenes [1].

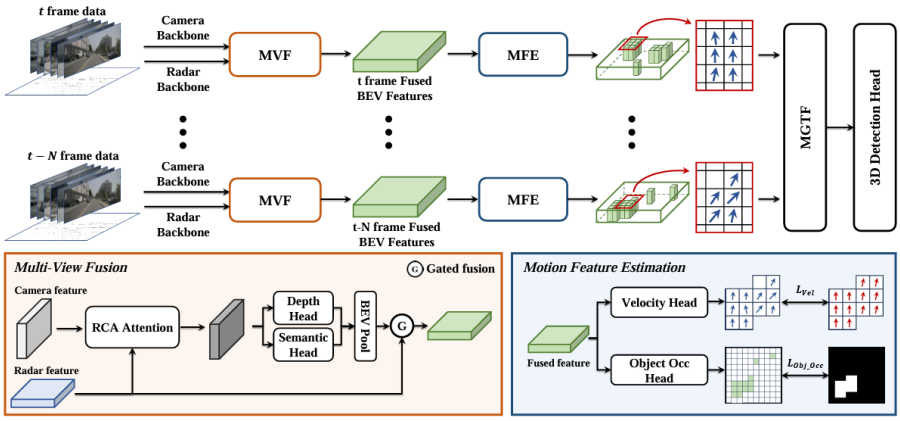


Figure 2.1: CRT-Fusion [7] model architecture.

CRT-Fusion [7] authors argue that existing methods don't effectively capture the motion of dynamic objects. The BEV features are simply concatenated with those from earlier frames, where data is merged from different time intervals. Consequently, the performance accuracy is compromised. While their design, uses motion compensation to rectify each feature map and fuse them in a temporally consistent manner.

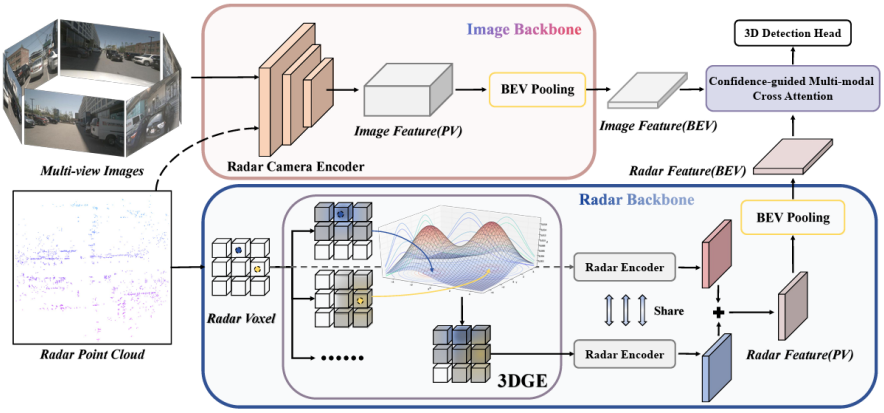


Figure 2.2: RobuRCDET [8] model architecture.

RobuRCDET [8] uses a query-based view approach, similarly sampling from camera and radar BEV features. However, they focus on designing a robust architecture that is resistant to noisy radar points and environmental camera disturbances such as illumination

and rain. The sparsity of the radar points is exploited to develop a spatial filter that enhances dense regions and reduces sparse areas. Additionally they enhance robustness by dynamically evaluating camera signal confidence to prioritize radar features when environmental conditions are harsh for the camera.

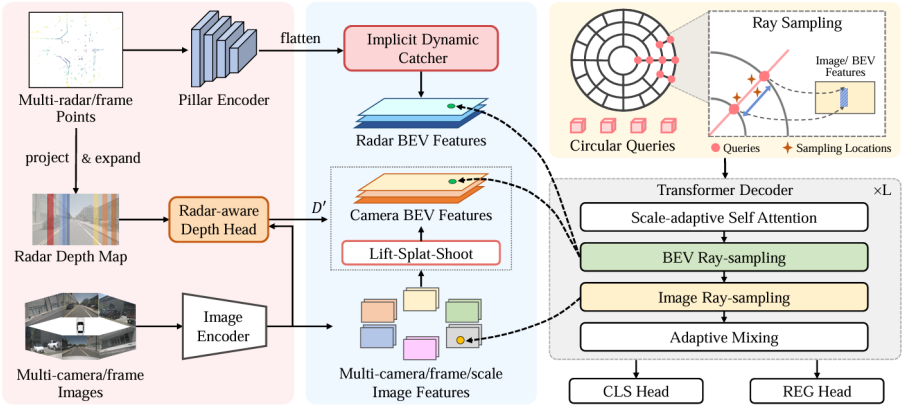


Figure 2.3: RacFormer [9] model architecture.

RacFormer [9] addresses the unaligned visual content due to depth errors when radar and camera BEV features are naively concatenated. Radar features are sparse and noisy, while camera BEV features are distorted due to view transformation. On the other hand, camera image view features are semantically rich and free from distortion. Therefore, they propose a query-based multi-view fusion architecture that samples simultaneously from radar and camera BEV features, but also camera image view features.

2.3 SOTA Radar only 3D Detection

To address the limitations of radar in multi-modal fusion systems, several works have focused on improving radar-only perception. These approaches attempt to extract richer spatial representations from radar measurements through learned feature encodings, radar point cloud processing, or temporal aggregation across multiple frames. While these methods can improve radar-based detection performance, they remain fundamentally constrained by the inherent sparsity of radar returns. This sparsity limits the amount of geometric structure that can be recovered from radar data alone, motivating research into methods that can enhance or densify radar representations.

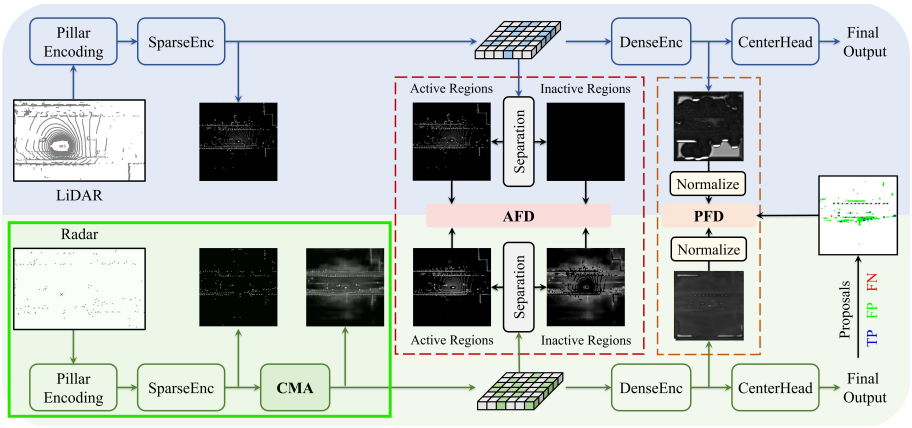


Figure 2.4: RadarDistill [3] model architecture.

RadarDistill [3] is a cross-modality knowledge distillation framework that substantially improves radar-only BEV 3D object detection by transferring LiDAR feature representations into a radar student during training, while requiring only radar at inference. Additionally, inserting RadarDistill into a camera-radar BEV fusion setup suggests a practical benefit even when radar is not used alone, especially if radar feature quality is a bottleneck in fusion.

The novelty is not simply BEV distillation, but rather distillation under extreme sparsity mismatch between radar and LiDAR pointclouds. Therefore, RadarDistill [3] first densifies radar BEV activations via a Cross-Modality Alignment module (CMA). Then it applies selective distillation at both low-level (activation-aware) and high-level (proposal-aware) feature stages. This encourages LiDAR-like representation in LiDAR-supported active areas while simultaneously penalizing radar-only activations in LiDAR-inactive areas acting as a learned false-positive suppression mechanism derived from the teacher.

On nuScenes [1] this method reports 20.5 mAP for radar-only detection, a large magin over prior radar-only baselines which report around 5 mAP on the nuScenes[1] leaderboards. Furthermore, when it is integrated into a camera-radar fusion pipeline based on BEVfusion [10]it yields an improvement oif +1.3% in mAP.

2.4 Diffusion Based Image Generators

Diffusion models have recently emerged as a powerful class of generative models capable of producing high-quality images and structured data. These models learn to generate data by progressively denoising a signal that has been corrupted with Gaussian noise over multiple diffusion steps. Through this iterative refinement process, diffusion models are able to capture complex data distributions and generate high-fidelity outputs. Their strong performance in image synthesis, super-resolution, and conditional generation tasks has led to growing interest in applying diffusion models to a wide range of computer vision problems.

A comprehensive analysis of diffusion model design choices is presented in Elucidating the Design Space of Diffusion-Based Generative Models (EDM [5]) framework by Karras et al. Rather than introducing a single new diffusion formulation, EDM studies the broader design space of diffusion based generative models and provides a unified view that connects multiple formulations including stochastic differential equation (SDE) based diffusion models, deterministic ordinary differential equation (ODE) samplers, and Markovian diffusion models such as Denoising Diffusion Probabilistic Models (DDPM [4]). By analyzing these formulations within a common framework, EDM identifies the key components that influence diffusion model performance and proposes optimal design guidelines for both the sampling and training stages for diffusion based image generation models as shown in Fig. 2.5.

		VP [49]	**VE [49]**	**iDDPM [37] + DDIM [47]**	**Ours ("EDM")**
Sampling (Section 3)					
ODE solver		Euler	Euler	Euler	2nd order Heun
Time steps	$t_{i<N}$	$1 + \frac{i}{N-1}(\epsilon_s - 1)$	$\sigma_{max}^2 \left(\sigma_{min}^2/\sigma_{max}^2\right)^{\frac{i}{N-1}}$	$u_{\lfloor j_0 + \frac{M-1-j_0}{N-1} i + \frac{1}{2} \rfloor}$, where $u_M = 0$, $u_{j-1} = \sqrt{\frac{u_j^2+1}{\max(\bar{\alpha}_{j-1}/\bar{\alpha}_j, C_1)} - 1}$	$\left(\sigma_{max}^{\frac{1}{\rho}} + \frac{i}{N-1}(\sigma_{min}^{\frac{1}{\rho}} - \sigma_{max}^{\frac{1}{\rho}})\right)^{\rho}$
Schedule	$\sigma(t)$	$\sqrt{e^{\frac{1}{2}\beta_d t^2 + \beta_{min} t} - 1}$	$\sqrt{t}$	t	t
Scaling	$s(t)$	$1/\sqrt{e^{\frac{1}{2}\beta_d t^2 + \beta_{min} t}}$	1	1	1
Network and preconditioning (Section 5)					
Architecture of F_θ		DDPM++	NCSN++	DDPM	(any)
Skip scaling	$c_{skip}(\sigma)$	1	1	1	$\sigma_{data}^2/\left(\sigma^2 + \sigma_{data}^2\right)$
Output scaling	$c_{out}(\sigma)$	$-\sigma$	σ	$-\sigma$	$\sigma \cdot \sigma_{data}/\sqrt{\sigma_{data}^2 + \sigma^2}$
Input scaling	$c_{in}(\sigma)$	$1/\sqrt{\sigma^2+1}$	1	$1/\sqrt{\sigma^2+1}$	$1/\sqrt{\sigma^2 + \sigma_{data}^2}$
Noise cond.	$c_{noise}(\sigma)$	$(M-1)\,\sigma^{-1}(\sigma)$	$\ln(\frac{1}{2}\sigma)$	$M - 1 - \arg\min_j \lvert u_j - \sigma \rvert$	$\frac{1}{4}\ln(\sigma)$
Training (Section 5)					
Noise distribution		$\sigma^{-1}(\sigma) \sim \mathcal{U}(\epsilon_t, 1)$	$\ln(\sigma) \sim \mathcal{U}(\ln(\sigma_{min}), \ln(\sigma_{max}))$	$\sigma = u_j,\ \ j \sim \mathcal{U}\{0, M-1\}$	$\ln(\sigma) \sim \mathcal{N}(P_{mean}, P_{std}^2)$
Loss weighting	$\lambda(\sigma)$	$1/\sigma^2$	$1/\sigma^2$	$1/\sigma^2$ (note: *)	$\left(\sigma^2 + \sigma_{data}^2\right)/(\sigma \cdot \sigma_{data})^2$
Parameters		$\beta_d = 19.9, \beta_{min} = 0.1$ $\epsilon_s = 10^{-3}, \epsilon_t = 10^{-5}$ $M = 1000$	$\sigma_{min} = 0.02$ $\sigma_{max} = 100$	$\bar{\alpha}_j = \sin^2(\frac{\pi}{2}\frac{j}{M(C_2+1)})$ $C_1 = 0.001, C_2 = 0.008$ $M = 1000, j_0 = 8^\dagger$	$\sigma_{min} = 0.002, \sigma_{max} = 80$ $\sigma_{data} = 0.5, \rho = 7$ $P_{mean} = -1.2, P_{std} = 1.2$

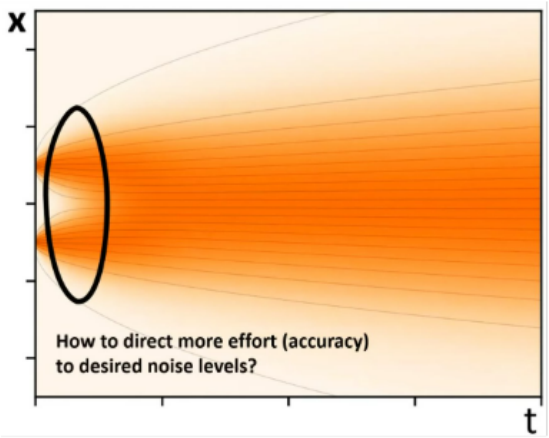
* iDDPM also employs a second loss term L_{vlb} † In our tests, $j_0 = 8$ yielded better FID than $j_0 = 0$ used by iDDPM

Figure 2.5: Specific design choices employed by different model families.

2.4.1 Sampling Stage Design Choices

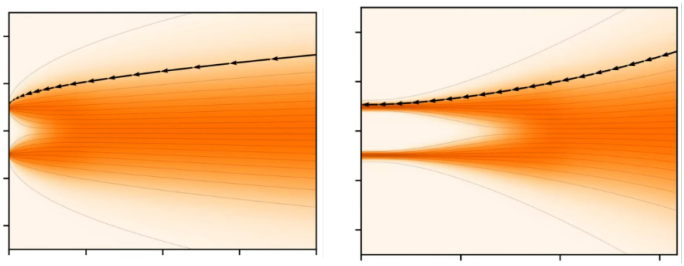
The sampling stage determines how images are generated from noise using the learned denoising network. EDM analyzes several factors that influence sample quality and computational efficiency, including the choice of time discretization, the noise schedule, the numerical solver used for integration, and the scaling of the signal during the diffusion trajectory. Importantly, the authors argue that sampling errors should be analyzed independently from training errors, since both stages introduce different sources of approximation.

Time Steps $t_{i<N}$ & Noise Scheduler $\sigma(t)$:



A central design decision concerns how noise levels are distributed across sampling steps. Image structure is primarily reconstructed at low noise levels, while high noise levels mainly correspond to global coarse structure. EDM therefore suggests allocating more denoising steps to lower noise levels where detailed image information is formed. This can be achieved either by using non-uniform step sizes—taking larger steps at high noise levels and smaller steps at lower as shown in levels Fig.2.6a or by warping the entire noise schedule to spend more iterations in the low noise region as shown in Fig. 2.6b.

The EDM analysis further argues that a linear noise schedule, similar to the one used in DDIM [11], provides favorable sampling behavior when solving the corresponding probability flow ODE. Under this schedule, the tangent of the diffusion trajectory aligns more closely with the desired denoising direction, resulting in lower curvature in the sampling path. Reduced curvature allows larger solver steps without significant degradation in sample quality, thereby reducing the number of required denoising iterations.



(a) Taking non-uniform step sizes.

(b) Warping the noise schedule.

Signal Scaling $s(t)$:

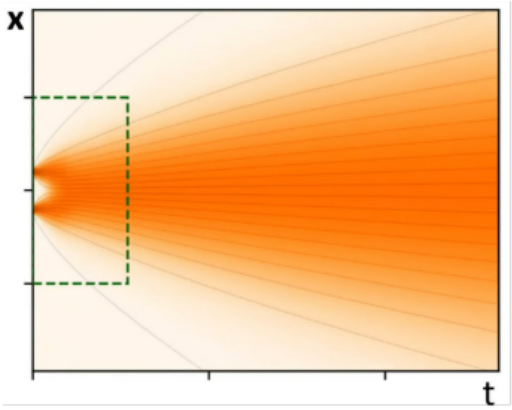
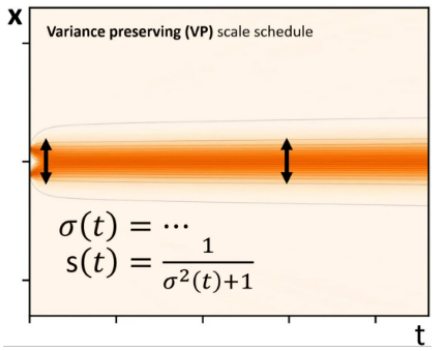


Figure 2.7: Noise levels growing unbounded.

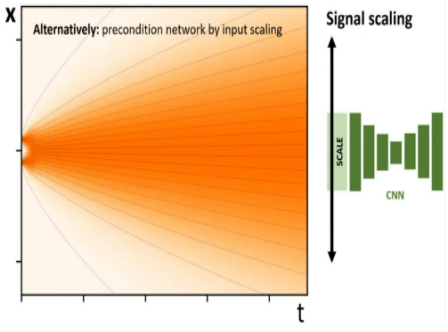
Another important aspect of the diffusion design space is the scaling of the signal as noise levels increase. Without appropriate normalization, the magnitude of the noisy signal can grow with the noise variance, which negatively affects neural network training dynamics as shown in Fig.2.7

Previous approaches such as Variance Preserving (VP [12]) diffusion models introduce explicit scaling schedules to keep the signal variance approximately constant during diffusion as shown in 2.9b.

EDM instead advocates leaving signal normalization largely to the neural network architecture itself rather than enforcing strict scaling through the diffusion process. According to their analysis, certain scale schedules can introduce unnecessary curvature in the diffusion trajectory, which makes numerical integration more difficult. Allowing the convolutional network to internally handle scaling can lead to simpler trajectories and more stable sampling as shwon in Fig.2.9b.

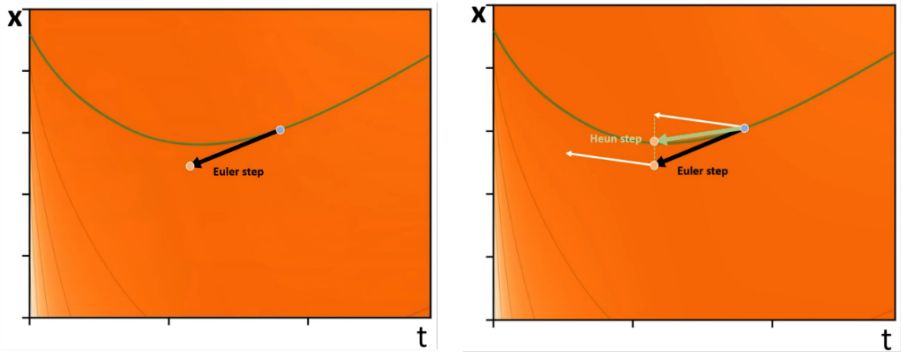


(a) Schedule scaling.



(b) Signal scaling.

ODE Solver $s(t)$:
Sampling in EDM is formulated as solving a reverse-time ODE. A simple approach is Euler integration, which uses a first-order update based on the current denoiser output but suffers from high discretization error when few steps are used. EDM instead employs Heun's method, a second-order predictor–corrector scheme that refines an initial Euler step by re-evaluating the denoiser and averaging the resulting slopes. This reduces numerical error and improves sample quality, allowing accurate generation with fewer sampling steps compared to Euler integration.



(a) Euler step.

(b) Heun step.

2.4.2 Training Stage Design Choices

Network Parameterization and Preconditioning
To simplify the learning problem for the neural network, EDM introduces a carefully designed preconditioning strategy. The input to the denoising network is normalized via the input scaling $c_{in}(\sigma)$ such that it has unit standard deviation regardless of the noise level. Similarly, training targets are scaled to maintain via the output scaling $c_{out}(\sigma)$ consistent magnitude across noise levels. This normalization stabilizes training and prevents the network from operating in regimes with extremely large or small activations.

EDM further introduces skipn connections $c_{skip}(\sigma)$ that allow the model to predict a mixture of the noisy input and the clean signal. At high noise levels the network primarily predicts the underlying clean signal to override the heavily corrupted input, while at low noise levels the skip connection encourages the network to predict residual noise corrections. This design prevents amplification of prediction errors when noise magnitudes are large and improves stability across the entire diffusion trajectory.

Loss Weighting $\lambda(\sigma)$ & Noise Sampling Distribution
The EDM framework also analyzes the training stage of diffusion models. Standard diffusion training uniformly samples noise levels during optimization, which can result in uneven gradient magnitudes across different noise scales. Some noise levels receive frequent small updates, while others receive rare but large updates, leading to unstable training dynamics. To address this issue, EDM proposes loss weighting strategies that equalize gradient magnitudes across noise levels. In addition, rather than sampling noise levels uniformly, the authors recommend sampling them according to a lognnormal distribution. This distribution allocates more training samples to intermediate noise levels where the denoising loss decreases most rapidly and where important image details are reconstructed. Together, these design choices significantly improve training efficiency and final image quality.

Overall, the EDM framework demonstrates that diffusion model performance depends not only on the network architecture but also on a range of design choices spanning the dif-

fusion formulation, noise schedule, numerical solver, preconditioning strategy, and training procedure. By analyzing these components in a unified framework, EDM provides practical guidelines that have influenced the design of many subsequent diffusionnbased generative models.

2.5 Radar-Diffusion Super Resolution Systems

More recently, diffusion models have been explored for generating and enhancing sensor data representations. In the context of radar perception, diffusion-based approaches aim to address the sparsity of radar measurements by generating denser radar representations that better capture the underlying scene structure. These models typically condition the diffusion process on sparse radar inputs or additional sensor modalities in order to produce improved radar maps or point clouds. Such approaches suggest that generative modeling may provide a promising direction for enhancing radar information prior to sensor fusion. Building on these developments, this thesis investigates the use of diffusion models to enhance radar data for improved camera–radar fusion in 3D object detection systems.

Recent radar-diffusion super resolution systems can be unified as conditional denoising in a chosen intermediate data representation. The main differentiators are (i) the intermediate radar/LiDAR representation (BEV occupancy map, BEV height map, high-dimensional voxel-feature latents), (ii) the diffusion parametrization (score, epsilon, clean data sample), and (iii) the solver/sampling strategy.

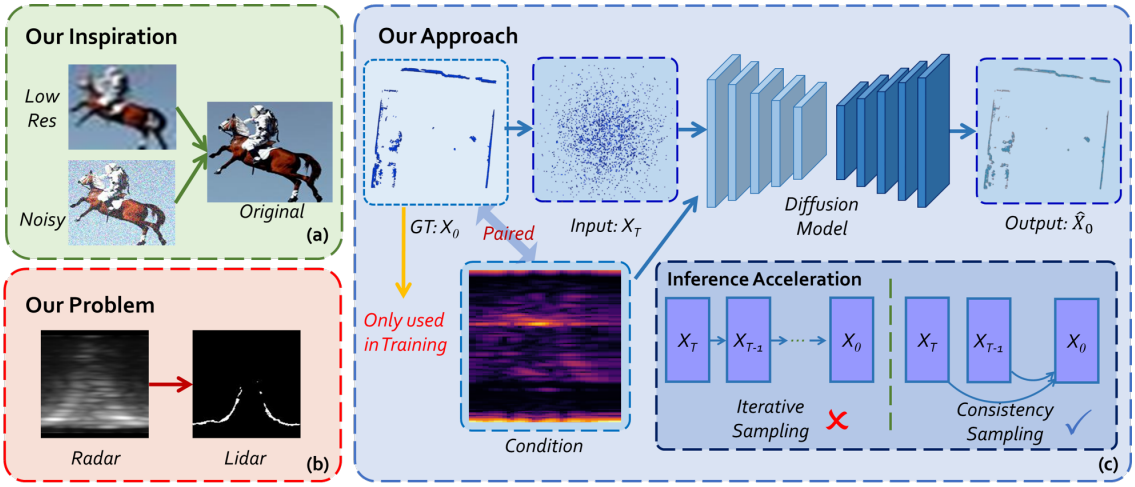


Figure 2.10: RadarDiffusion [13] model architecture.

An efficiency-oriented cross-modal approach [13] RadarDiffusion frames the radar super resolution problem as a conditional image restoration from RAH (Radar-Azimuth Heatmap) to LiDAR BEV occupancy map. By directly predicting x_0 (clean data sample) with a U-Net augmented by multi-head attention, optimizes a weighted MSE+LPIPS [14], and adopts EDM's [5] noise/timestamp schedules for iterative generation while also distilling a consistency model that enables one-step prediction crucial for embedded MAV constraints.

A mean reverting formulation [15] converts radar/LiDAR pointclouds into BEV height maps and defines LiDAR to radar degradation with an OU (Ornstein-Uhlenback) mean reverting SDE so the marginal radar distribution tends to the radar BEV mean (rather than zero). This yields a reverse-time SDE whose drift uses the score to learn the reverse of LiDAR to radar degradation. The method trains a time conditioned U-Net to predict epsilon (the

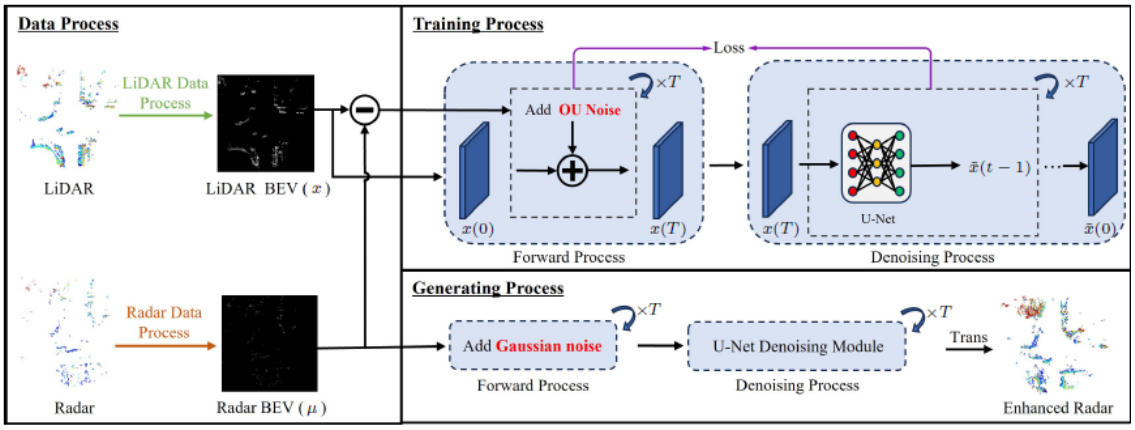


Figure 2.11: Diffusion based pointcloud super-resolution [15]

h!

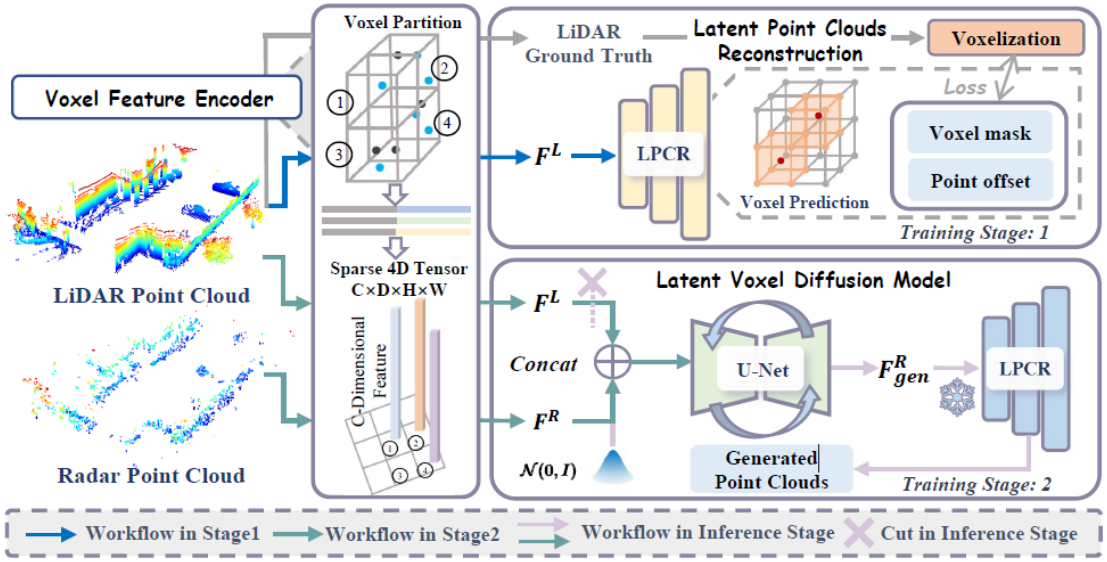


Figure 2.12: R2LDM [16] model architecture.

noise $_t$) and stabilizes denoising objective that seperately weights ”occupied” and ”blank” cells via different masks addressing the density imbalance in the BEV space.

R2LDM [16] argues that BEV/range images discard critical 3D structure and instead encodes both 4D radar and LiDAR point clouds into high-dimensional voxel-feature latents, runs a DDPM-style Gaussian Markov forward process in latent space, conditions denoising on radar latents (with concatenation empirically outperforming cross-attention), and reconstructs dense geometry using an explicit latent point-cloud reconstruction module trained with voxel-mask binary cross entropy (BCE) and offset L1; importantly, it reports that straightforward L2 denoising with noise-prediction and a long (128-step) sampling trajectory provides the best overall trade-off, while a two-stage curriculum (decoder first, then conditional diffusion reduces optimization difficulty under modality density gaps and improves compute efficiency for convergence.

3 Research Gap & Thesis Scope

As discussed in Sec.2.2, **SOTA Camera-Radar Fusion 3D Detection**, relatively few camera–radar fusion methods explicitly address the sparsity of radar features. Among the existing approaches, the pioneering work of RadarDistill proposes a cross-modality knowledge distillation framework that significantly improves radar-only BEV 3D object detection by transferring LiDAR feature representations to a radar-based student network during training, while requiring only radar inputs at inference time.

RadarDistill is currently the top-performing radar-only method on the nuScenes detection benchmark, achieving 20.5% mAP and outperforming prior radar-only approaches by a margin of +15.5% mAP. Furthermore, the authors demonstrate that their radar enhancement pipeline can be integrated into a camera–radar fusion architecture and trained end-to-end, resulting in a +1.5 mAP improvement over the camera–radar baseline. This finding suggests that improving radar feature representations can still positively impact the overall fusion pipeline, and that addressing radar sparsity remains an important factor even when strong camera-based components are present.

Motivated by the results of RadarDistill, the increasing adoption of diffusion models in computer vision and perception tasks, and the limited number of radar-only submissions (three) to the nuScenes 3D detection benchmark, the scope of the thesis is limited to enhancing the sparsity of radar point clouds using diffusion models for radar-only 3D object detection. Similar to the approach in RadarDistill, we had planned to integrate within camera-radar architecture but was not worth implementing given the performance issues observed.

4 nuScenes: A multimodal dataset for autonomous driving

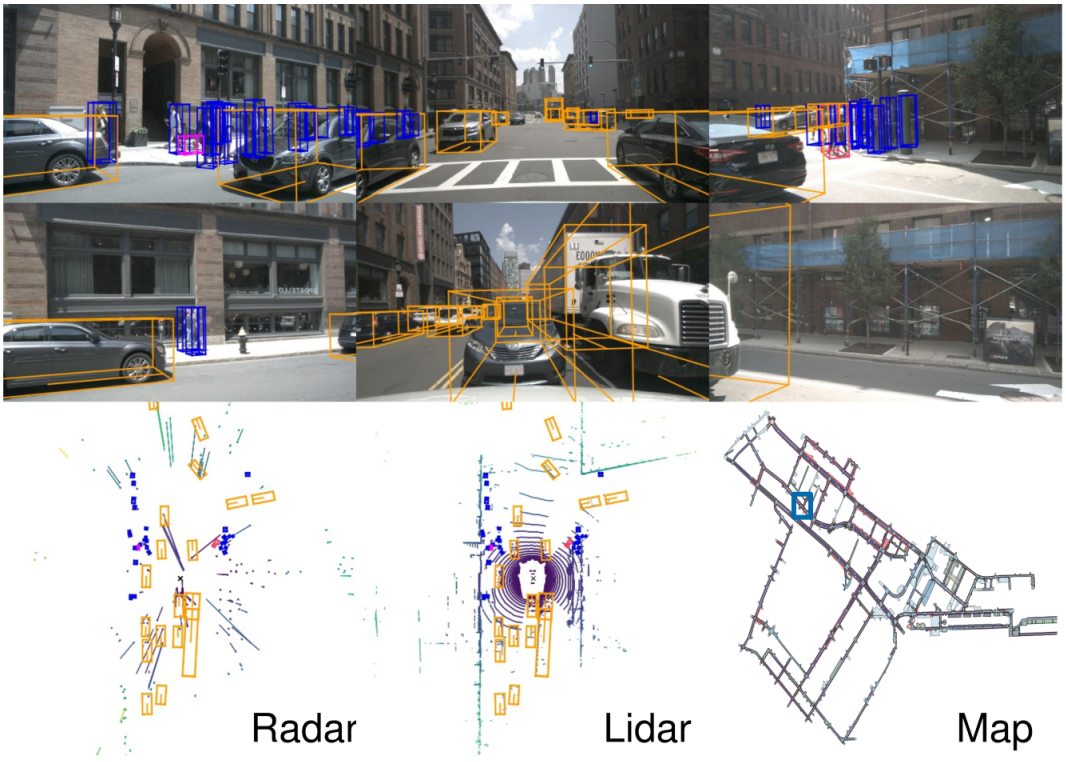


Figure 4.1: An example from the nuScenes dataset [1]. We see 6 different camera views, lidar and radar data, as well as the human annotated semantic map. At the bottom we show the human written scene description.

Sensor	Details
6x Camera	RGB, 12Hz capture frequency, $1/1.8$" CMOS sensor, 1600×900 resolution, auto exposur JPEG compressed
1x Lidar	Spinning, 32 beams, 20Hz capture fr quency, 360° horizontal FOV, -30° to 1(vertical FOV, $\leq 70m$ range, ± 2cm accurac up to $1.4M$ points per second.
5x Radar	$\leq 250m$ range, 77GHz, FMCW, 13Hz captu frequency, ± 0.1km/h vel. accuracy
GPS & IMU	GPS, IMU, AHRS. 0.2° heading, 0.1° rc l/pitch, 20mm RTK positioning, 1000Hz up- date rate

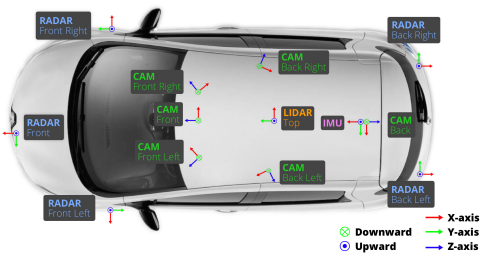


Figure 4.2: Sensor data and setup for nuScenes data collection vehicle [1].

The nuScenes dataset [1] is one of the most widely used benchmarks for autonomous driving perception. It provides a large-scale multimodal dataset collected from a vehicle equipped with multiple complementary sensors, including cameras, LiDAR, radar, and GPS/IMU. The dataset contains 1000 driving scenes recorded in urban environments

around Boston and Singapore, each lasting approximately 20 seconds. In total, nuScenes provides over 1.4 million camera images and corresponding LiDAR and radar measurements together with detailed 3D annotations.

The dataset has become a standard benchmark for evaluating 3D object detection methods in autonomous driving due to its diverse sensor setup shown in 4.2 and challenging environmental conditions. In addition to LiDAR point clouds, nuScenes provides radar measurements from five radar sensors distributed around the vehicle. Radar offers advantages such as long sensing range and robustness to adverse weather conditions, but the measurements are significantly sparser and noisier compared to LiDAR. As a result, extracting meaningful spatial structure from radar data remains an open challenge in autonomous driving perception.

For 3D detection tasks, nuScenes provides annotated bounding boxes for multiple object categories such as cars, pedestrians, bicycles, and traffic cones. These annotations allow algorithms to learn spatial representations of the environment and to detect objects directly in three-dimensional space. In practice, most modern 3D detection methods rely on bird's-eye-view (BEV) representations derived from LiDAR point clouds, since BEV provides a structured spatial representation that simplifies reasoning about object positions and orientations. Consequently, many perception pipelines transform raw point clouds into BEV representations before applying deep learning models.

The following section describes the preprocessing steps required to convert the raw sensor data provided by nuScenes into the intermediate representations used in this work.

4.1 Preprocessing Pipeline

Although nuScenes provides synchronized multimodal sensor data, the raw measurements cannot be used directly in most 3D perception pipelines. In particular, radar and LiDAR measurements differ significantly in density, coordinate frames, and temporal acquisition. Radar point clouds are extremely sparse and noisy, while LiDAR provides denser spatial measurements but is still captured over multiple sweeps at different timestamps.

To increase spatial coverage, it is common practice to aggregate multiple consecutive sensor sweeps. This process, known as multi-sweep accumulation, increases the number of observed points and improves the completeness of the scene representation. However, naïvely stacking sweeps introduces geometric distortions because both the ego vehicle and surrounding objects move between frames. Without proper alignment, this motion leads to blurred object shapes and duplicated structures in the aggregated point cloud.

Furthermore, different sensors operate in different coordinate frames. Radar sensors are mounted at multiple locations on the vehicle and therefore report measurements in their own local coordinate systems. In order to combine radar and LiDAR information, these measurements must first be transformed into a common reference frame.

Finally, many modern 3D detection methods operate on structured spatial representations such as bird's-eye-view (BEV) grids rather than raw point clouds. Converting point clouds into BEV representations simplifies spatial reasoning and is a suitable representation for the following tasks. For this reason, the aggregated point clouds are ultimately converted into BEV occupancy and height maps.

Various object clustering methods include ground segmentation as a prior step for object recognition. Because ground points are not the region of interest in object clustering methods, the elimination of ground points enhances both computational efficiency and accuracy. That is, approximately half of the points in a 3D LiDAR scan are given as

ground points in outdoor environments. This implies that the overall algorithm can be faster, as the size of the input point cloud will be half of the original via ground removal [17].

The preprocessing pipeline described in this section therefore performs four main steps: multi-sweep aggregation and coordinate transformation of radar measurements, motion compensation of LiDAR sweeps, removal of ground points, and conversion of the point clouds into BEV representations. These steps produce a geometrically consistent and structured representation of the scene that can be used for downstream perception tasks.

4.1.1 Radar Multi-Sweep + Transformation to LiDAR Frame

Let $k \in \mathcal{K}$ index radar sensors and $i \in \{0, \dots, M_k - 1\}$ index the selected sweeps ($M_k \leq M$).

Each sweep provides a radar point cloud:

$$\mathbf{P}_{k,i} \in \mathbb{R}^{N_{k,i} \times 18}, \tag{4.1}$$

where each point is

$$\mathbf{p} = \left[\mathbf{x}^{(s)}, \mathbf{a}, \mathbf{v}^{(s)}, \mathbf{v}^{(s)}_{\text{comp}}, \mathbf{b}\right], \tag{4.2}$$

with:

- $\mathbf{x}^{(s)} \in \mathbb{R}^3$: position in radar sensor frame,
- $\mathbf{v}^{(s)} \in \mathbb{R}^2$: raw Doppler velocity,
- $\mathbf{v}^{(s)}_{\text{comp}} \in \mathbb{R}^2$: ego-motion compensated velocity,
- $\mathbf{a}, \mathbf{b}$: additional radar attributes.

Spatial Transformation

Each sweep provides a rigid transformation from sensor frame to LiDAR frame:

$$\mathbf{R}_{k,i} \in SO(3), \qquad \mathbf{t}_{k,i} \in \mathbb{R}^3. \tag{4.3}$$

Positions are transformed as:

$$\mathbf{x}^{(\ell)} = \mathbf{R}_{k,i}\mathbf{x}^{(s)} + \mathbf{t}_{k,i}. \tag{4.4}$$

Radar velocities are lifted to 3D and rotated:

$$\tilde{\mathbf{v}}^{(s)} = \begin{bmatrix} \mathbf{v}^{(s)} \\ 0 \end{bmatrix}, \qquad \tilde{\mathbf{v}}^{(s)}_{\text{comp}} = \begin{bmatrix} \mathbf{v}^{(s)}_{\text{comp}} \\ 0 \end{bmatrix}. \tag{4.5}$$

$$\tilde{\mathbf{v}}^{(\ell)} = \mathbf{R}_{k,i}\tilde{\mathbf{v}}^{(s)}, \qquad \tilde{\mathbf{v}}^{(\ell)}_{\text{comp}} = \mathbf{R}_{k,i}\tilde{\mathbf{v}}^{(s)}_{\text{comp}}. \tag{4.6}$$

Only the planar components are retained:

$$\mathbf{v}^{(\ell)} = \left(\tilde{\mathbf{v}}^{(\ell)}\right)_{1:2}, \qquad \mathbf{v}^{(\ell)}_{\text{comp}} = \left(\tilde{\mathbf{v}}^{(\ell)}_{\text{comp}}\right)_{1:2}. \tag{4.7}$$

Temporal Encoding

Let $t_{k,i}$ denote the timestamp of sweep i, and $t_{k,0}$ the reference timestamp for sensor k.

The time lag is defined as:

$$\Delta t_{k,i} = t_{k,0} - t_{k,i}. \tag{4.8}$$

Per-Point Feature Vector
The resulting feature vector for each point is:

$$\mathbf{p}' = \left[\mathbf{x}^{(\ell)}, \mathbf{a}, \mathbf{v}^{(\ell)}, \mathbf{v}^{(\ell)}_{\text{comp}}, \mathbf{b}, \Delta t_{k,i}\right]. \tag{4.9}$$

Multi-Sweep Aggregation
All selected sweeps across all sensors are concatenated:

$$\mathbf{P}_{\text{agg}} = \text{concat}\left(\left\{\mathbf{P}'_{k,i}\right\}_{k,i}\right). \tag{4.10}$$

Optionally, a global rotation about the vertical axis is applied:

$$\mathbf{x} \leftarrow \mathbf{R}_z(-\tfrac{\pi}{2})\mathbf{x}. \tag{4.11}$$

4.1.2 LiDAR Multi-Sweep + Ego & Object-Motion Compensation

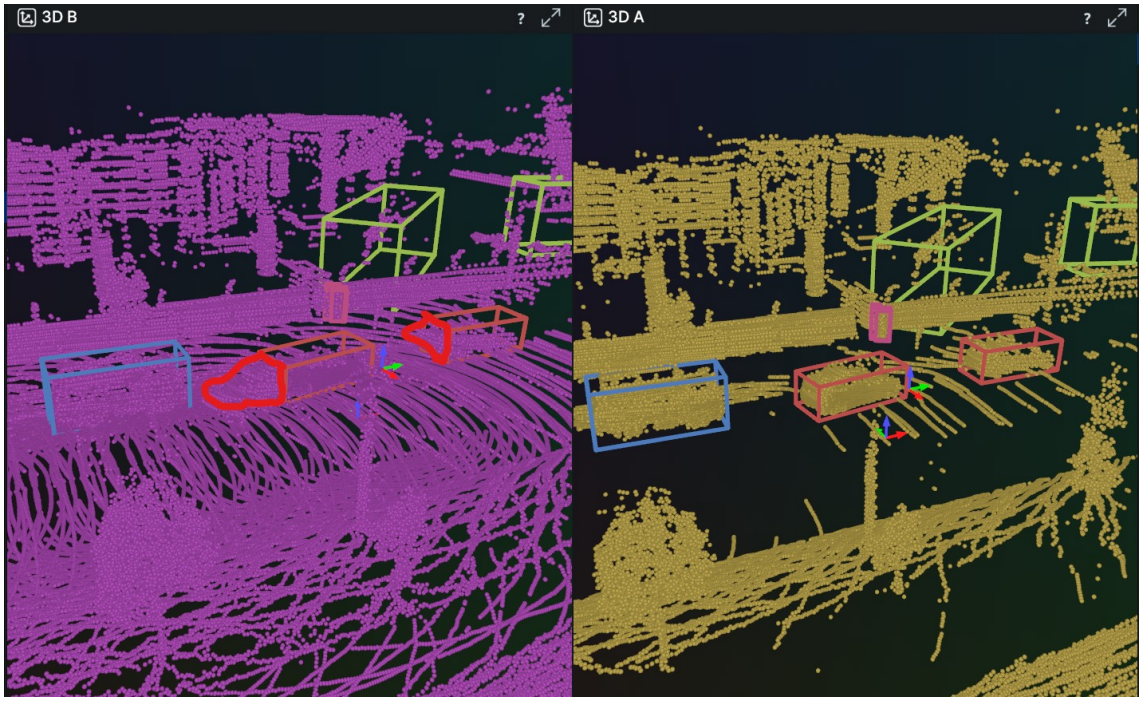


Figure 4.3: Before (left) & after (right) motion compensation:without compensation moving objects leave a trail of points (within red mask) behind them belonging to a previous time stamp.

Let $i \in \{0, \ldots, M-1\}$ index LiDAR sweeps, where $i = 0$ denotes the keyframe. The raw point cloud of sweep i is

$$\mathbf{P}_i = \{\mathbf{p}_{i,n}\}_{n=1}^{N_i}, \qquad \mathbf{p}_{i,n} \in \mathbb{R}^D, \tag{4.12}$$

and we use the 3D coordinates $\mathbf{x}_{i,n} \in \mathbb{R}^3$ (and optionally other channels).

Close-point removal. Define the keep set (axis-aligned box around the ego) with radius r:

$$\mathcal{K}_i(r) = \{n \mid \neg\left(|x_{i,n}| < r \wedge |y_{i,n}| < r \wedge |z_{i,n}| < r\right)\}, \tag{4.13}$$

and replace $\mathbf{P}_i \leftarrow \{\mathbf{p}_{i,n}\}_{n \in \mathcal{K}_i(r)}$.

Rigid pose transforms. Each sweep provides homogeneous transforms

$$\mathbf{T}^{g}_{e,i} \in SE(3) \ \ (\text{ego} \to \text{global}), \qquad \mathbf{T}^{e}_{\ell,i} \in SE(3) \ \ (\text{LiDAR} \to \text{ego}), \tag{4.14}$$

and similarly $\mathbf{T}^{g}_{e,0}$ and $\mathbf{T}^{e}_{\ell,0}$ for the keyframe. The ego motion from sweep i to keyframe 0 in the ego frame is

$$\mathbf{T}^{e_0}_{e_i} = \left(\mathbf{T}^{g}_{e,0}\right)^{-1} \mathbf{T}^{g}_{e,i}. \tag{4.15}$$

We also define $\mathbf{T}^{\ell_0}_{e_0} = \left(\mathbf{T}^{e}_{\ell,0}\right)^{-1}$.

Object sets and point-to-object assignment. Let the set of tracked objects common to sweep i and keyframe 0 be

$$\mathcal{O}_i = \mathcal{O}_0 \cap \mathcal{O}_i^{\text{sweep}}. \tag{4.16}$$

For each object $j \in \mathcal{O}_i$, let its 3D box parameters in LiDAR coordinates be

$$\mathbf{b}_j^{(0)} = (\mathbf{c}_j^{(0)}, \psi_j^{(0)}, \dots), \qquad \mathbf{b}_j^{(i)} = (\mathbf{c}_j^{(i)}, \psi_j^{(i)}, \dots), \tag{4.17}$$

where $\mathbf{c} = (c_x, c_y, c_z)$ is the box center and ψ is yaw. Define a point-to-object index map (via point-in-box test):

$$\alpha_i(n) \in \{-1\} \cup \mathcal{O}_i, \tag{4.18}$$

where $\alpha_i(n) = j$ means point n belongs to object j, and -1 denotes background.

Object-motion compensation (dynamic points). For a point assigned to object $j = \alpha_i(n) \neq -1$, define

$$\Delta\psi_{j,i} = \psi_j^{(0)} - \psi_j^{(i)}, \qquad \Delta\mathbf{t}_{j,i} = \begin{bmatrix} c_{j,x}^{(0)} - c_{j,x}^{(i)} \\ c_{j,y}^{(0)} - c_{j,y}^{(i)} \\ 0 \end{bmatrix}, \tag{4.19}$$

and the planar yaw rotation

$$\mathbf{R}_z(\Delta\psi_{j,i}) = \begin{bmatrix} \cos\Delta\psi_{j,i} & -\sin\Delta\psi_{j,i} & 0 \\ \sin\Delta\psi_{j,i} & \cos\Delta\psi_{j,i} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.20}$$

Then the object-compensated coordinate is

$$\mathbf{x}'_{i,n} = \mathbf{c}_j^{(i)} + \mathbf{R}_z(\Delta\psi_{j,i}) \left(\mathbf{x}_{i,n} - \mathbf{c}_j^{(i)}\right) + \Delta\mathbf{t}_{j,i}. \tag{4.21}$$

Ego-motion compensation (background points). For a background point ($\alpha_i(n) = -1$), we align it to the keyframe LiDAR coordinates by

$$\bar{\mathbf{x}}_{i,n} = \pi\left(\mathbf{T}^{\ell_0}_{e_0}\, \mathbf{T}^{e_0}_{e_i}\, \mathbf{T}^{e}_{\ell,i}\, \bar{\mathbf{x}}^{\ell}_{i,n}\right), \qquad \bar{\mathbf{x}}^{\ell}_{i,n} = \begin{bmatrix} \mathbf{x}_{i,n} \\ 1 \end{bmatrix}, \tag{4.22}$$

where $\pi([x\ y\ z\ w]^\top) = [x\ y\ z]^\top$ denotes dropping the homogeneous coordinate.

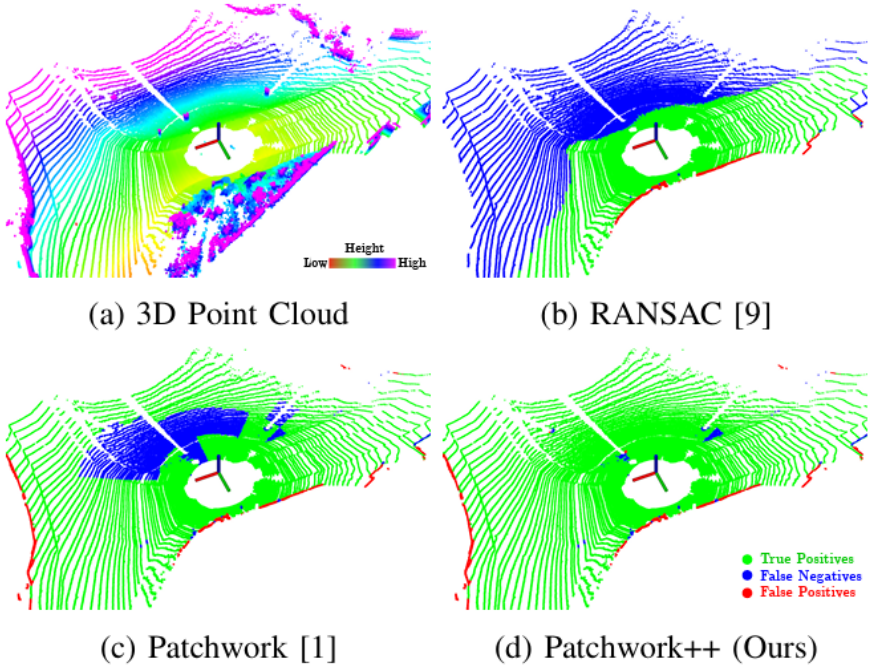
Combined compensation rule. For sweep i, the final motion-compensated point coordinates are

$$\tilde{\mathbf{x}}_{i,n} = \begin{cases} \mathbf{x}'_{i,n}, & \alpha_i(n) \neq -1, \\ \bar{\mathbf{x}}_{i,n}, & \alpha_i(n) = -1. \end{cases} \tag{4.23}$$

Multi-sweep aggregation. Let $\tilde{\mathbf{P}}_0 = \mathbf{P}_0$ (keyframe points, after close-point removal) and for $i \geq 1$ let $\tilde{\mathbf{P}}_i = \{\tilde{\mathbf{x}}_{i,n}\}_{n=1}^{N_i}$ be the compensated sweep points. The aggregated multi-sweep point set is

$$\mathbf{P}_{\text{agg}} = \bigcup_{i=0}^{M-1} \tilde{\mathbf{P}}_i. \tag{4.24}$$

4.1.3 Ground Removal via Patchwork++



(a) 3D Point Cloud (b) RANSAC [9]

(c) Patchwork [1] (d) Patchwork++ (Ours)

Figure 4.4: Enter Caption

Ground points are estimated via concentric region-wise PCA plane fitting with adaptive elevation and flatness thresholds, vertical plane rejection, and temporal consistency refinement as proposed in Patchwork++ [17]. We directly use the module provided without going into the details of implementation.

4.1.4 Point Cloud to BEV (Occupancy + Height Map)

Given a point set $\mathcal{P} = \{\mathbf{p}_n\}_{n=1}^{N}$, $\mathbf{p}_n = (x_n, y_n, z_n) \in \mathbb{R}^3$, we construct a BEV grid over $[x_{\min}, x_{\max}) \times [y_{\min}, y_{\max})$ with resolution (W, H) and cell sizes $\Delta x = \frac{x_{\max} - x_{\min}}{W}$, $\Delta y = \frac{y_{\max} - y_{\min}}{H}$.

Preprocessing. We apply a horizontal flip and range filtering:

$$(x'_n, y'_n, z'_n) = (x_n,\ -y_n,\ z_n), \qquad x'_n \in [x_{\min}, x_{\max}),\ y'_n \in [y_{\min}, y_{\max}). \tag{4.25}$$

Optionally, LiDAR-only height filtering keeps points with $z'_n \in [h_{\min}, h_{\max}]$, and/or downsampling keeps a quantile band $z'_n \in [q_\alpha, q_\beta]$.

Discretization. Each remaining point is mapped to a BEV cell:

$$u_n = \left\lfloor \frac{x'_n - x_{\min}}{\Delta x} \right\rfloor, \qquad v_n = \left\lfloor \frac{y'_n - y_{\min}}{\Delta y} \right\rfloor, \tag{4.26}$$

clipped to $u_n \in \{0, \ldots, W-1\},\ v_n \in \{0, \ldots, H-1\}$.

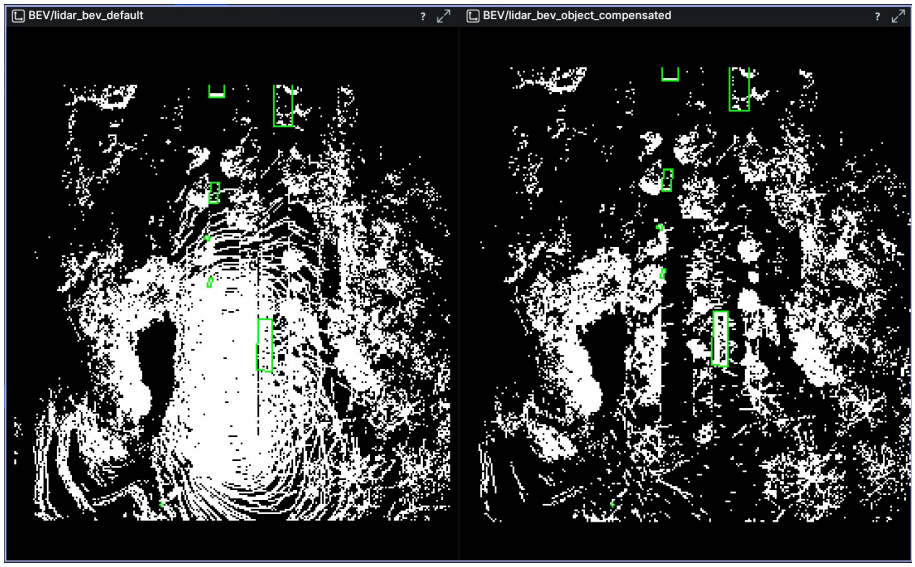


Figure 4.5: Point cloud to BEV with (left) & without (right) ground point filtering.

Occupancy map. Let $c(v,u)$ be the number of points falling in cell (v,u):

$$c(v,u) = \sum_{n=1}^{N} \mathbb{I}[u_n = u \wedge v_n = v]. \tag{4.27}$$

The BEV *occupancy* map is the binary image

$$O(v,u) = \mathbb{I}[c(v,u) > 0] \in \{0,1\}^{H \times W}. \tag{4.28}$$

Average height map. Let the accumulated height per cell be

$$s(v,u) = \sum_{n=1}^{N} z'_n \, \mathbb{I}[u_n = u \wedge v_n = v]. \tag{4.29}$$

The BEV *height* map is the per-cell average height:

$$Z(v,u) = \begin{cases} \frac{s(v,u)}{c(v,u)}, & c(v,u) > 0, \\ 0, & c(v,u) = 0, \end{cases} \qquad Z \in \mathbb{R}^{H \times W}. \tag{4.30}$$

Output tensor. The final BEV representation is a 2-channel tensor

$$\mathbf{B} = \text{stack}\big(Z,\ O\big) \in \mathbb{R}^{2 \times H \times W}, \tag{4.31}$$

where channel 1 is the **average height map** and channel 2 is the **occupancy map**.

5 Methodology

This chapter describes the methodology used to address the sparsity of radar measurements for radar-only 3D object detection. The proposed framework consists of two main stages. In the first stage, a conditional diffusion model is trained to enhance sparse radar observations by generating dense LiDAR-like bird's-eye-view (BEV) representations. The diffusion model learns to reconstruct clean LiDAR BEV occupancy and height maps from noise while being conditioned on radar BEV inputs. Several design choices are explored, including intermediate data representations, preprocessing strategies, loss formulations, and training augmentations aimed at improving robustness under weak radar conditioning.

In the second stage, the generated BEV representations are evaluated through a downstream 3D object detection pipeline using the CenterPoint detector. This allows the quality of the reconstructed radar representations to be assessed in terms of their impact on detection performance. Finally, the two components are combined into an end-to-end framework where the diffusion backbone and the detection head are jointly optimized. Implementation details and evaluation metrics used to measure both reconstruction fidelity and detection performance are also presented in this chapter.

5.1 Stage 1: Radar BEV Diffusion

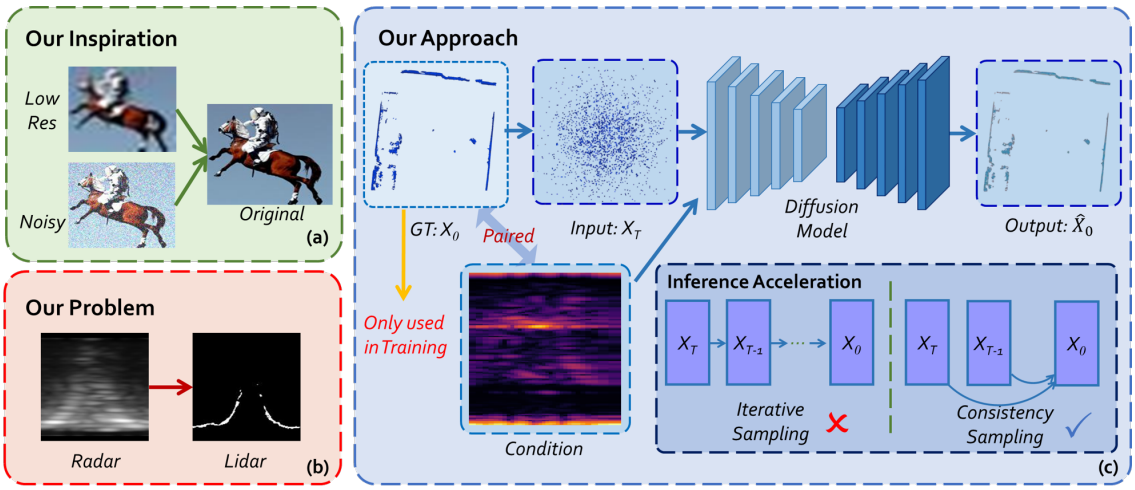


Figure 5.1: Radar–LiDAR translation can be formulated as an image restoration task, where noisy and low-resolution radar range–azimuth heatmaps (RAHs) are used to recover high-resolution LiDAR bird's-eye-view (BEV) representations. Due to the significantly lower angular resolution and higher noise levels of millimeter-wave radar compared to LiDAR, predicting LiDAR BEV images from paired radar RAHs can be treated as a restoration problem. The proposed method employs a diffusion-based framework where LiDAR point clouds are progressively corrupted during training, and a neural network learns to reconstruct the ground truth conditioned on radar observations. During inference, the model generates LiDAR BEV representations from Gaussian noise conditioned on radar inputs. To address the computational cost of iterative diffusion sampling, consistency models are incorporated to enable single-step generation.

5.1.1 Intermediate Data Representation & Preprocessing Pipeline

Our implementation considers RadarDiffusion [13] as a starting point given its use of optimal diffusion building blocks for image generation as argued by the EDM [5] paper and the availability of the source code. Given that nuScenes [1] only provides radar point clouds and not the RAHs, we condition the model instead on radar BEV occupancy maps and train the model using groundtruth LiDAR BEV occupancy maps. However, it quickly became clear that this representation omits critical height information for 3D detection. This is clearly shown in Table 5.1 where the mAP drops significantly from 56% to 13% when height information is lost and less significantly a reduction in pointcloud density. This is the case, even for the ideal LiDAR point cloud which acts as a theoretical upper bound for this representation. As this emphasizes the importance of the height information, the model is then modified to output a 2-channel tensor that includes both an occupancy map and a height map and is trained using the corresponding groundtruth. It is worth noting that the radar used in nuScenes only provides 2D point clouds and therefore the height information is only known for the LiDAR groundtruth and not the radar conditioning BEV occupancy signal.

Next, the ideal LiDAR point cloud is used to determine a suitable preprocessing pipeline and a BEV resolution for this choice of data representation. First, we observe the impact of filtering out the ground points. This becomes an issue as we convert points to BEV, where the height of points that fall within BEV cell are averaged and thus final value in the height map is pulled down in the presence of ground points. It is shown that this lowers 3D detection performance from 23% to 17% and is therefore considered when training our radar model using groundtruth LiDAR height maps. While for the BEV resolution, a significant performance gain is obtained as we increase the shape from 512×512 to 1024×1024 reaching 42% mAP.

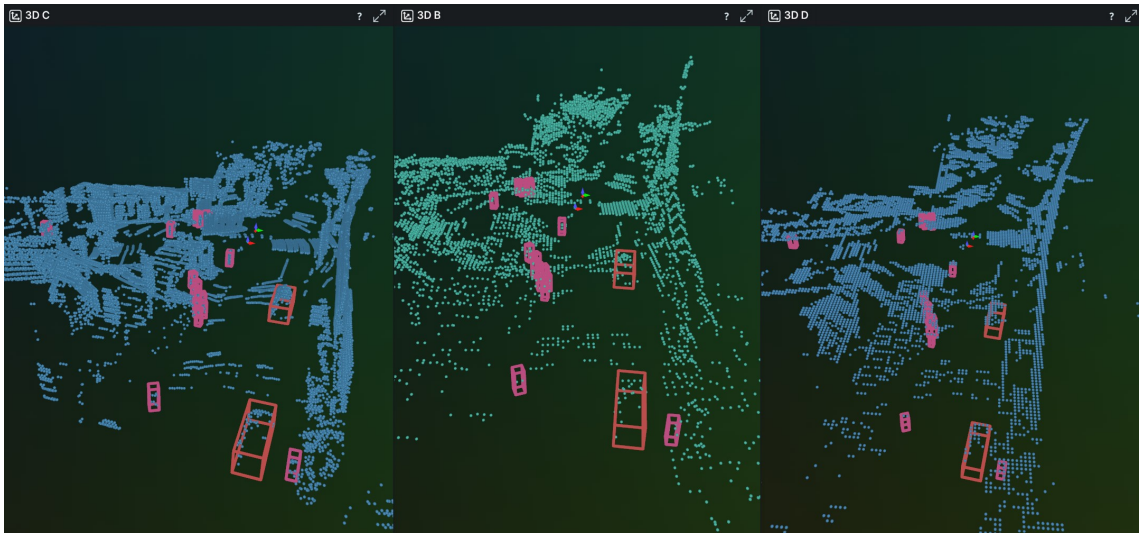


Figure 5.2: (Left 3DA): Normal PointCloud - (Middle 3DB): PointCloud from BEV Occupancy & Height Map - (Right 3DC): PointCloud from BEV Occupancy with constant height

Model	Preprocessing Pipeline	mAP
CenterPoint (Benchmark)	• Points From LiDAR MultiSweeps • No Motion Compensation • Temporal Encoding	56%
CenterPoint	• Points From LiDAR MultiSweeps • Motion Compensation • Ground Point Removal • Points to BEV – Occupancy BEV Map – Height BEV Map $\approx$ eq.(4.30) • BEV to Points	42% @ 1024$\times$1024 BEV 23% @ 512$\times$512
RadarDistill (Target)		20%
CenterPoint	• Points From LiDAR MultiSweeps • Motion Compensation • No Ground Point Removal • Points to BEV – Occupancy BEV Map – Height BEV Map $\approx$ eq.(4.30) • BEV to Points	17% @ 512$\times$512 BEV
CenterPoint	• Points From LiDAR MultiSweeps • Motion Compensation • Ground Point Removal • Points to BEV – Occupancy BEV Map – Constant Zero Height • BEV to Points (Flattened/Collapsed Height)	13%

Table 5.1: Comparison of preprocessing pipelines and resulting mAP.

5.1.2 Loss Functions

The radar BEV diffusion model is trained to reconstruct the clean LiDAR BEV representation from noisy inputs conditioned on radar observations. Let x_0 denote the ground-truth LiDAR BEV representation and x_t the noise-corrupted sample obtained during the diffusion process at timestep t. The network predicts the denoised output $\hat{x}_0 = x_\theta(x_t, t, c)$ conditioned on the radar BEV input c.

Mean Squared Error (MSE). The primary objective minimizes the pixel-wise ℓ_2 distance between the predicted BEV representation and the ground-truth LiDAR BEV:

$$\mathcal{L}_{\text{MSE}} = \|x_0 - x_\theta(x_t, t, c)\|_2^2. \tag{5.1}$$

This loss ensures accurate reconstruction of the underlying LiDAR BEV structure from noisy samples during training.

Perceptual Loss (LPIPS). To better preserve structural features and perceptual similarity, we incorporate the Learned Perceptual Image Patch Similarity (LPIPS) loss. Instead of comparing pixels directly, LPIPS compares deep features extracted from a pretrained network $f_p(\cdot)$:

$$\mathcal{L}_{\text{LPIPS}} = \|f_p(x_0) - f_p(x_\theta(x_t, t, c))\|_2^2. \tag{5.2}$$

This encourages the generated BEV representation to maintain high-level spatial structures such as object boundaries and scene layout. Similar perceptual losses have been shown to improve diffusion-based image restoration quality [13].

Binary Cross Entropy (BCE). Since the BEV representation includes an occupancy channel indicating the presence of points within each grid cell, we additionally optimize a binary cross-entropy loss for this channel:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right], \tag{5.3}$$

where y_i denotes the ground-truth occupancy label and $\hat{y}_i$ the predicted probability for cell i.

Training Objective. Following RadarDiffusion, the initial model predicting only an occupancy map is trained using a weighted combination of mean squared error (MSE) and perceptual loss:

$$\mathcal{L} = 0.8\,\mathcal{L}_{\text{MSE}} + 0.2\,\mathcal{L}_{\text{LPIPS}}. \tag{5.4}$$

When extending the representation to jointly predict an occupancy map and a height map, two additional training strategies are explored. First, both channels are optimized using MSE where The height regression loss is evaluated only for occupied BEV cells.

Let $\hat{o} = \text{sigmoid}(o_\theta)$ denote the predicted occupancy probability and define an occupancy mask

$$M_{\text{occ}} = \begin{cases} 1 & \text{if the predicted occupancy probability } \hat{o} > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

The height loss is then computed only over occupied cells as

$$\mathcal{L}_{\text{MSE}}^{\text{height}} = \frac{1}{\sum_i M_{\text{occ},i}} \sum_i M_{\text{occ},i}(h_i - \hat{h}_i)^2. \tag{5.6}$$

$$\mathcal{L} = \mathcal{L}_{\text{MSE}}^{\text{occ}} + \mathcal{L}_{\text{MSE}}^{\text{height}}. \tag{5.7}$$

Second, the occupancy channel is optimized using binary cross-entropy (BCE) while the height channel remains a regression target trained with masked MSE:

$$\mathcal{L} = \mathcal{L}_{\text{BCE}}^{\text{occ}} + \mathcal{L}_{\text{MSE}}^{\text{height}}. \tag{5.8}$$

While diffusion models are typically optimized using mean squared error derived from score matching, we employ binary cross-entropy for the occupancy channel to better model its Bernoulli distribution. This modifies the likelihood assumption but still results in valid gradients for learning the denoising mapping.

One critical consideration that was overlooked at the time of the experiments was the potential imbalance between the numerical scales of the loss terms. In particular, the height values were not normalized prior to training, which may cause the MSE loss for the height map to dominate the BCE loss used for occupancy prediction due to larger numerical magnitudes. As a result, the optimization process may implicitly prioritize height regression over accurate occupancy classification.

5.1.3 Cut & Mix Augmentation

Hallucination in diffusion models under weak conditioning signals refers to the generation of unrealistic, inconsistent, or non-existent objects/features when the input guidance (text, image, or latent constraint) is ambiguous, underspecified, or lacks necessary detail. This phenomenon occurs because the model, lacking strong guidance to stay within a specific semantic manifold, falls back on its internal, unconstrained data distribution, often leading to structural errors or the interpolation of data modes.

Mode Interpolation and Unconstrained Distribution
A core cause of hallucinations is ”mode interpolation”, where diffusion models create smooth approximations between disjoint data modes in the training set. When the model lacks strong guidance to stay within a specific semantic manifold, it generates samples that are ”completely outside the support of the original training distribution,” leading to non-existent objects or artifacts [18].

Ambiguous and Underspecified Guidance
Hallucinations are frequently triggered when prompts are ”vague, underspecified, or structurally misleading,” which forces the model into ”speculative generation”. In such cases, the model ”fills in gaps based on similar contexts from its training data” [19].

Structural and Semantic Errors
Weak semantic signals, particularly in specific layers of the model, lead to a phenomenon called ”Condition Isolation”. In these instances, the model detaches from the text guidance and over-relies on ”learned visual priors,” resulting in structural inconsistencies and the assembly of symbols in a nonsensical manner [20].

Falling Back on Internal Priors
Research on Dynamic Guidance and Temporal Alignment Guidance highlights that without continuous, strong alignment with the desired manifold, models accumulate errors and drift toward unrealistic regions of the data distribution [21].

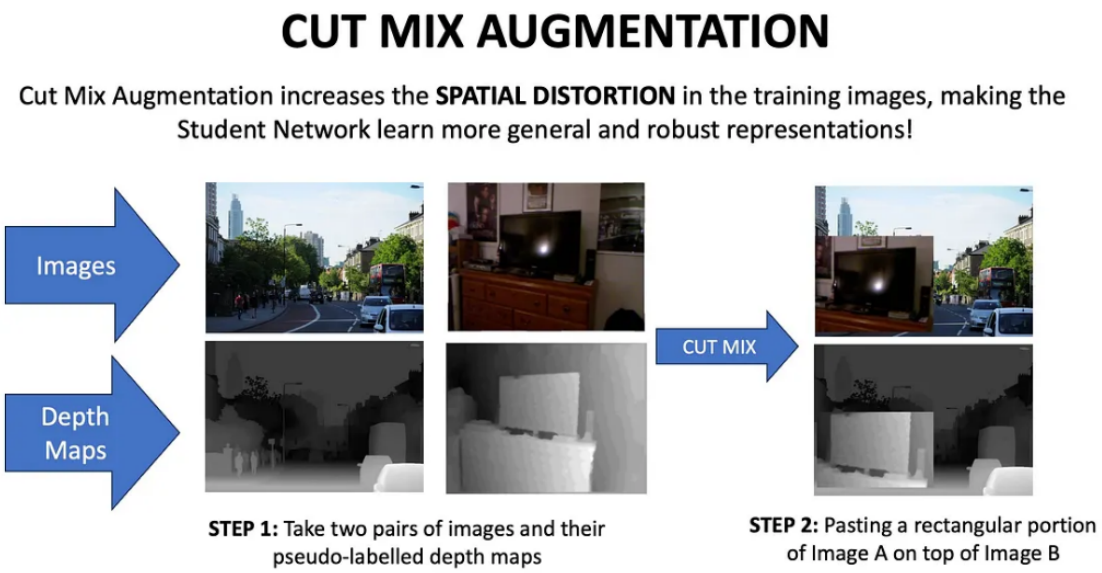
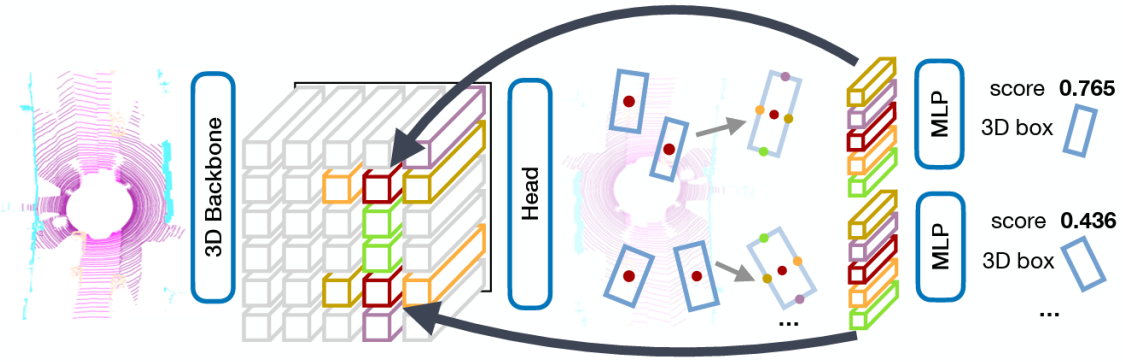


Figure 5.3: Cut & Mix Operation

To address the hallucination dilemma mentioned above, we propose to challenge the model by injecting strong perturbation during training specifically cut and mix augmentation shown in Fig. 5.3 inspired by the the work proposed in DepthAnything [22]. We hypothesize that this compels the model to learn more robust representations and strengthens the guidance by the conditioning signal.

5.2 Stage 2: Centerpoint as benchmark for point-based 3D Detection



(a) Point. cloud (b) Map-view features (c) First stage: Centers and 3D boxes (d) Second stage: Score and 3D boxes

Figure 5.4: Overview of CenterPoint [2] framework.We rely on a standard 3Dbackbone that extracts map-view feature representation from Lidarpoint-clouds.Then,a 2D CNN architecture detectionhead finds object centers and regress to full 3D bounding boxes using center features.This box prediction is used to extract point features at the 3D centers of each face of the estimated 3D bounding box, which are passed into MLP to predict an IoU-guided confidence score and box regression refinement.

5.2.1 Feature Extraction Pipeline

The detector is trained in the second stage, after the enhanced radar BEV maps are generated and saved offline. We can either use an image encoder on the generated radar BEV maps to extract features and feed forward to the detection head or we can convert the BEV maps back to points based on the grid occupancy, height and resolution and use a point encoder which is the common practice when working with pointclouds. Similar to Table 5.1 where the suitable preprocessing pipeline is determined experimentally using the ideal LiDAR point cloud, it is shown in Table 5.2 that the point encoder produces better results. Furthermore, by keeping the same encoder and the following feature extraction pipeline as the baseline the model performance can be direcly attributed to the quality generated BEV and pointclouds as points are the only variable.

5.2.2 Loss Functions

CenterPoint predicts a class-specific center heatmap $\hat{Y}$ and a set of regression targets at object centers:

$$o \in \mathbb{R}^2, \qquad h_g \in \mathbb{R}, \qquad s \in \mathbb{R}^3, \qquad (\sin\alpha, \cos\alpha) \in \mathbb{R}^2, \qquad v \in \mathbb{R}^2.$$

The center heatmap is trained using a Gaussian focal loss:

$$L_{\text{cls}} = L_{\text{focal}}(\hat{Y}, Y),$$

where Y is the target heatmap obtained by rendering a Gaussian at each ground-truth object center. CenterPoint enlarges the Gaussian radius as

$$\sigma = \max(f(wl), \tau), \qquad \tau = 2.$$

For the regression branches, an L1 loss is applied only at the ground-truth center locations:

$$L_{\text{reg}} = L_o + L_{h_g} + L_s + L_{\text{rot}} + L_v,$$

DetectorI	Preprocessing Pipeline	Feature Extraction Pipeline	mAP
CenterHead	• Points From LiDAR MultiSweeps • Motion Compensation • Ground Point Removal • Points to BEV – Occupancy BEV Map – Constant Zero Height • BEV to points	• Voxel Encoder: HardSimpleVFE • Sparse Encoder:SparseEncoder • Backbone SECOND • Neck / FPN SECONDFPN	13%
CenterHead	• Points From LiDAR MultiSweeps • Motion Compensation • Ground Point Removal • Points to BEV – Occupancy BEV Map – Constant Zero Height	• Image Encoder Backbone: HRNet • Neck (`HRFPN`) • FPN SECONDFPN	13%

Table 5.2: Comparison of preprocessing and feature extraction pipelines for the default CenterPoint model and a BEV image-encoder variant.

with

$$L_* = \sum_{p \in \mathcal{P}} \left\| \hat{t}_p - t_p \right\|_1 ,$$

where $\mathcal{P}$ denotes the set of ground-truth center locations and t_p is the corresponding target. For box size regression, CenterPoint uses logarithmic box dimensions. The overall first-stage objective is the sum of heatmap and regression losses:

$$L = L_{\text{cls}} + L_{\text{reg}}.$$

5.2.3 Class-Balanced Grouping & Sampling

The nuScenes dataset exhibits a strong long-tailed class distribution, where common categories such as *car* dominate the training data while rare classes (e.g., *bicycle* or *motorcycle*) appear significantly less frequently. This imbalance can bias the detector toward majority classes and degrade performance on tail classes. [23]

To address this issue, Class-Balanced Grouping and Sampling (CBGS) introduces two complementary strategies. First, a **dataset sampling** method is applied to balance the class distribution. Let $\mathcal{D}$ denote the original training set and $\mathcal{C}$ the set of object categories. For each category $c \in \mathcal{C}$, the number of samples containing that category is counted and resampled such that each category contributes approximately the same number of training samples. Formally, the sampled dataset $\mathcal{D}'$ is constructed as

$$\mathcal{D}' = \bigcup_{c \in \mathcal{C}} \text{Sample}(\mathcal{D}_c, N),$$

where $\mathcal{D}_c$ denotes the subset of samples containing class c and N is the target number of samples per category.

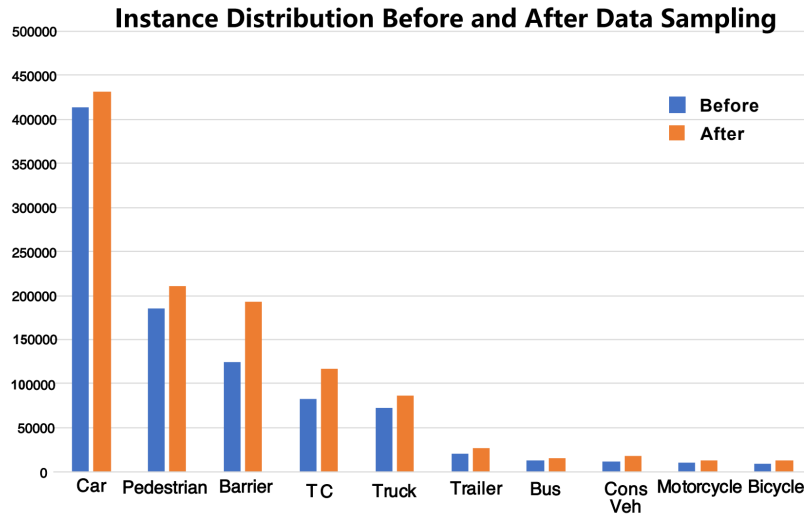


Figure 5.5: Caption

Second, CBGS introduces **class-balanced grouping** in the detection head. Instead of predicting all categories with a single shared head, classes are divided into groups according to two principles: (1) classes with similar shapes or sizes should be grouped together, and (2) instance numbers across groups should be balanced to prevent dominant classes from overwhelming the learning process. Following these principles, the nuScenes categories are divided into six groups:

- (Car)
- (Truck, Construction Vehicle)
- (Bus, Trailer)
- (Barrier)
- (Motorcycle, Bicycle)
- (Pedestrian, Traffic Cone)

Each group is handled by an independent detection head, allowing the model to learn shared geometric features within groups while reducing inter-class interference between unrelated categories [23].

Together, class-balanced sampling and grouping alleviate the class imbalance problem and significantly improve detection performance across both frequent and rare categories.

5.3 End-To-End: Radar BEV Diffusion + 3D Detection

Recent work has shown that diffusion models can also benefit from multi-task learning. For example, DiffusionMTL [24] jointly denoises predictions for multiple dense prediction tasks and achieves improved performance by modeling cross-task correlations during the diffusion process. Similarly, TaskDiffusion [25] performs joint diffusion across multiple dense prediction tasks, enabling shared reasoning during denoising. Another example is MTLSC-Diff [26], which jointly performs hyperspectral image super-resolution and classification using a diffusion framework, where the classification objective guides the reconstruction process and improves performance on both tasks. These results echo findings in models beyond diffusion such as DepthAnything [22], where auxiliary tasks like semantic segmentation improve depth estimation by providing complementary supervision.

Motivated by these findings we attempt to train Radar BEV Diffusion and 3D Detection (radar only) end-to-end. Knowing that RadarDiffusion uses an estimate of the clean data sample during training, this is a good first step. However, at high noise levels the corrupted sample x_t contains little information about the underlying scene structure as shown in 5.6, making it difficult for the denoiser to recover a meaningful BEV representation in a single forward pass.

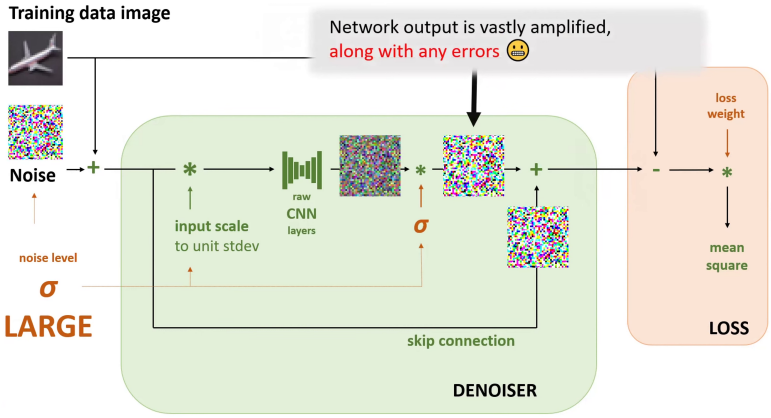


Figure 5.6: Higher noise levels amplifies training losses.

In classical diffusion models this issue is typically addressed through iterative sampling, where the denoiser is repeatedly applied while solving the reverse-time diffusion dynamics until the clean sample is obtained. Such iterative procedures shown in Fig. 5.7 are computationally expensive and make end-to-end training with a downstream detection head impractical.

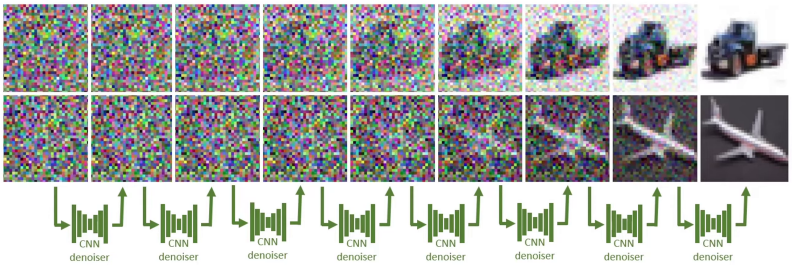


Figure 5.7: Iterative denoising in diffusion models.

The EDM parameterization adopted by RadarDiffusion provides a useful property in this context. The denoiser is written as

$$D_\theta(x_t, \sigma) = c_{\text{skip}}(\sigma)x_t + c_{\text{out}}(\sigma)F_\theta(c_{\text{in}}(\sigma)x_t, \sigma), \tag{5.9}$$

where F_θ denotes the raw neural network and

$$c_{\text{skip}}(\sigma) = \frac{\sigma_{\text{data}}^2}{\sigma^2 + \sigma_{\text{data}}^2}. \tag{5.10}$$

Unlike classical diffusion formulations, where the network predicts the additive noise and the clean sample must be reconstructed analytically, the EDM denoiser output $D_\theta(x_t, \sigma)$ is trained to directly approximate the clean sample x_0. As a result, a single forward pass already produces a noise-conditioned estimate of the clean BEV representation.

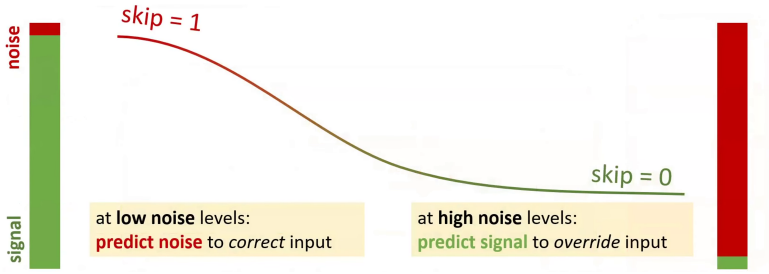


Figure 5.8: Skip scale as a function of noise level.

An important property of the EDM formulation is the introduction the skip connection shown in 5.9depends on the noise level. For small corruption levels, $c_{\text{skip}}(\sigma) \approx 1$, meaning that the denoiser behaves close to an identity mapping and preserves most of the spatial structure in the noisy input. In contrast, when the noise level becomes large, $c_{\text{skip}}(\sigma)$ approaches zero, reducing the influence of the corrupted input and forcing the model to rely more heavily on the learned residual branch to reconstruct the scene structure. This adaptive behavior is beneficial when integrating the diffusion backbone with a detection network. In the low-noise regime the detector receives BEV maps that are close to the original radar representation, while at higher noise levels the denoiser learns to infer missing structure from learned priors.

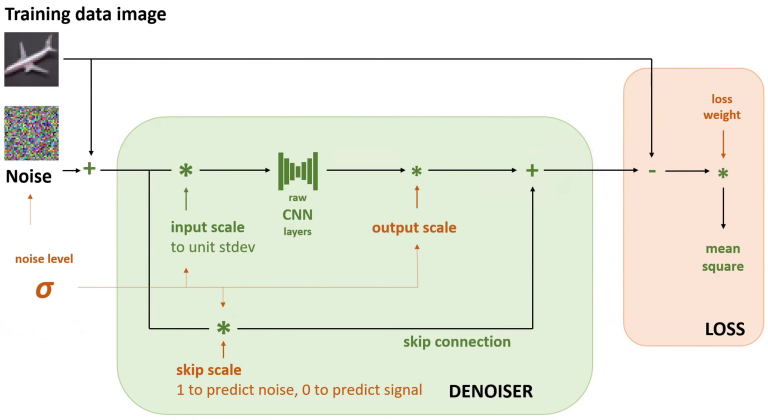


Figure 5.9: Skip connection proposed by EDM [16]

Another important factor is the distribution of noise levels used during training. Following EDM and RadarDiffusion, the noise level σ is sampled from a log-normal distribution,

which allocates higher probability mass to lower corruption levels. Consequently, the denoiser is optimized most frequently in the regime where the corrupted BEV still retains meaningful spatial information. This makes the single-step estimate $D_\theta(x_t, \sigma)$ particularly suitable as an intermediate representation for downstream tasks such as 3D object detection.

Based on these observations, we propose to jointly train the diffusion backbone and the detection head using the denoiser output as the input to the detector. During training, a noise level σ is sampled and the clean BEV map x_0 is corrupted according to

$$x_t = x_0 + \sigma\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{5.11}$$

The denoiser then produces a reconstruction

$$\hat{x}_0 = D_\theta(x_t, \sigma), \tag{5.12}$$

which is passed directly to the 3D detection network. The overall training objective combines the diffusion reconstruction loss with the detection loss,

$$\mathcal{L} = \mathcal{L}_{\text{diffusion}} + \lambda\mathcal{L}_{\text{detection}}. \tag{5.13}$$

In this formulation the diffusion backbone effectively acts as a noise-conditioned BEV refinement module. The adaptive skip connection ensures that the representation remains stable across different noise levels, while the log-normal sampling of σ focuses training on corruption levels that still preserve meaningful scene structure. As a result, the detector can learn to operate on partially denoised BEV representations without requiring the expensive iterative reverse diffusion process during training.

5.4 Implementation Details

For our radar bev diffusion model we adopt the entire implementation from RadarDiffusion [13] and follow their training procedure by using RAdam optimizer and a constant learning rate $1e-5$. They train for 280k steps with a global batch size of 32, for our nuScenes dataset which contains around 28100 samples. This corresponds to around 320 epochs according to:

$$\text{epochs} = \frac{280k \times 32}{28100} \approx 320 \tag{5.14}$$

For cut and mix augmentation, we only augment half the training set such that the probability of a training sample to be augmented is 0.5 similar to DepthAnything [22].

Similarly, for Centerpoint [2] we adopt their training procedure which uses AdamW optimizer with a one cycle learning rate between $1e-4$ and $1e-3$, trains for 20 epochs with a batch size of 4. This is also used for end-to-end training of the occupancy denoiser and the 3D detector.

5.5 Evaluation Metrics

To evaluate the effectiveness of the proposed radar sparsity enhancement approach, we consider metrics that capture both reconstruction quality and downstream detection performance. In particular, we measure the quality of the generated occupancy representation, the accuracy of the predicted height map, and the impact of the generated point clouds on 3D object detection.

The quality of the predicted occupancy map is evaluated using the Intersection over Union (IoU) metric. Let O_{pred} denote the predicted binary occupancy map and O_{gt} the ground truth occupancy map derived from LiDAR. The IoU is defined as

$$IoU = \frac{|O_{pred} \cap O_{gt}|}{|O_{pred} \cup O_{gt}|}, \tag{5.15}$$

where $|\cdot|$ denotes the number of occupied cells. This metric measures the spatial overlap between predicted and ground truth occupied regions and reflects how well the diffusion model reconstructs the LiDAR occupancy structure from sparse radar observations.

For models that additionally predict a height map, we evaluate the height reconstruction accuracy using the Mean Squared Error (MSE):

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (h_i^{pred} - h_i^{gt})^2, \tag{5.16}$$

where h_i^{pred} and h_i^{gt} denote the predicted and ground truth height values at cell i, and N is the total number of cells. This metric quantifies the geometric consistency of the reconstructed scene by measuring the deviation between predicted and ground truth heights.

To assess the usefulness of the generated radar point clouds for perception tasks, we evaluate the performance of a CenterPoint 3D object detector trained on the reconstructed data. Detection performance is measured using mean Average Precision (mAP). For each class c, precision and recall are defined as

$$Precision = \frac{TP}{TP + FP}, \qquad Recall = \frac{TP}{TP + FN}, \tag{5.17}$$

where TP, FP, and FN denote the number of true positives, false positives, and false negatives, respectively. The Average Precision for class c is computed as the area under the precision–recall curve:

$$AP_c = \int_0^1 P_c(R)\, dR, \tag{5.18}$$

where $P_c(R)$ denotes precision as a function of recall for class c. The mean Average Precision is then defined as

$$mAP = \frac{1}{C} \sum_{c=1}^{C} AP_c, \tag{5.19}$$

where C is the number of evaluated object classes.

Together, these metrics provide a comprehensive evaluation of the proposed approach: IoU measures the accuracy of the reconstructed occupancy structure, MSE evaluates geometric height consistency, and mAP quantifies the effectiveness of the generated radar points for downstream 3D detection tasks. This combination of metrics allows us to assess both the reconstruction fidelity of the diffusion model and its practical utility for autonomous driving perception systems.

6 Experimental Results

This chapter presents the experimental and qualitative evaluation of the proposed diffusion-based radar sparsity enhancement model and investigates the impact of different approaches. We conduct a series of ablation studies to analyze the contribution of individual design choices, including BEV representation, loss formulation, the conditioning signal, data augmentation, and training strategy. The experiments evaluate both the reconstruction quality of the diffusion model and the downstream 3D detection performance when training CenterPoint on the generated point clouds. Table 6.1 summarizes the results of the evaluated configurations.

Table 6.1: Evaluation of different training strategies for the diffusion-based radar 3D Detection. **Height Map** in the conditioning signal is artificially generated by using LiDAR values.

Approach	**Occupancy IoU (%) Train-Val**	**Height MSE Train-Val**	**mAP (%)**
• End-To-End: Train Occupancy & Height Map Denoise + CenterPoint – Loss: MSE + L_{cls} + L_{reg} – Conditioning Signal: Radar Occupancy + Height Map – Augmentation: None	19-12	15-11	2
• Stage 1: Train Occupancy & Height Denoiser – Loss: MSE + BCE – Conditioning Signal: Radar Occupancy + **Height Map** – Augmentation: Cut & Mix • Stage 2: Train CenterPoint on Generated Points	37-18	6-8	5
• Stage 1: Train Occupancy & Height Denoiser – Loss: MSE – Conditioning Signal: Radar Occupancy Map – Augmentation: Cut & Mix • Stage 2: Train CenterPoint on Generated Points	38-21	6-9	5
• Stage 1: Train Occupancy & Height Denoiser – Loss: MSE + BCE – Conditioning Signal: Radar Occupancy Map – Augmentation: Cut & Mix • Stage 2: Train CenterPoint on Generated Points	34–15	7–11	3
• Stage 1: Train Occupancy Denoiser – Loss: MSE + LPIPS – Conditioning Signal: Radar Occupancy Map – Augmentation: Cut & Mix • Stage 2: Train CenterPoint on Generated Points	46–28	-	6
• Stage 1: Train Occupancy Denoiser – Conditioning Signal: Radar Occupancy Map – Loss: MSE + LPIPS – Augmentation: None • Stage 2: Train CenterPoint on Generated Points	39–22	-	4

6.1 Qualitative

Here we present some examples of the generated BEV occupancy maps by our model RadarBEVDiffusion against the Lidar groundtruth BEVand the radar BEV conditioning signal. It is observed that instead the sparse radar BEV we obtain a much more globally aligned structure of the scene when compared with LiDAR. On a closer level, finer details are generated incorrectly (hallucination) or completely overlooked which affects the ability to detect the objects (green boxes). We believe this is due to the weak conditioning signal in some region of the occupancy map, where no structure is is shown but the model still has to generate an output with no guidance. This causes ambiguity and the model to hallucinate according to what we described earlier.

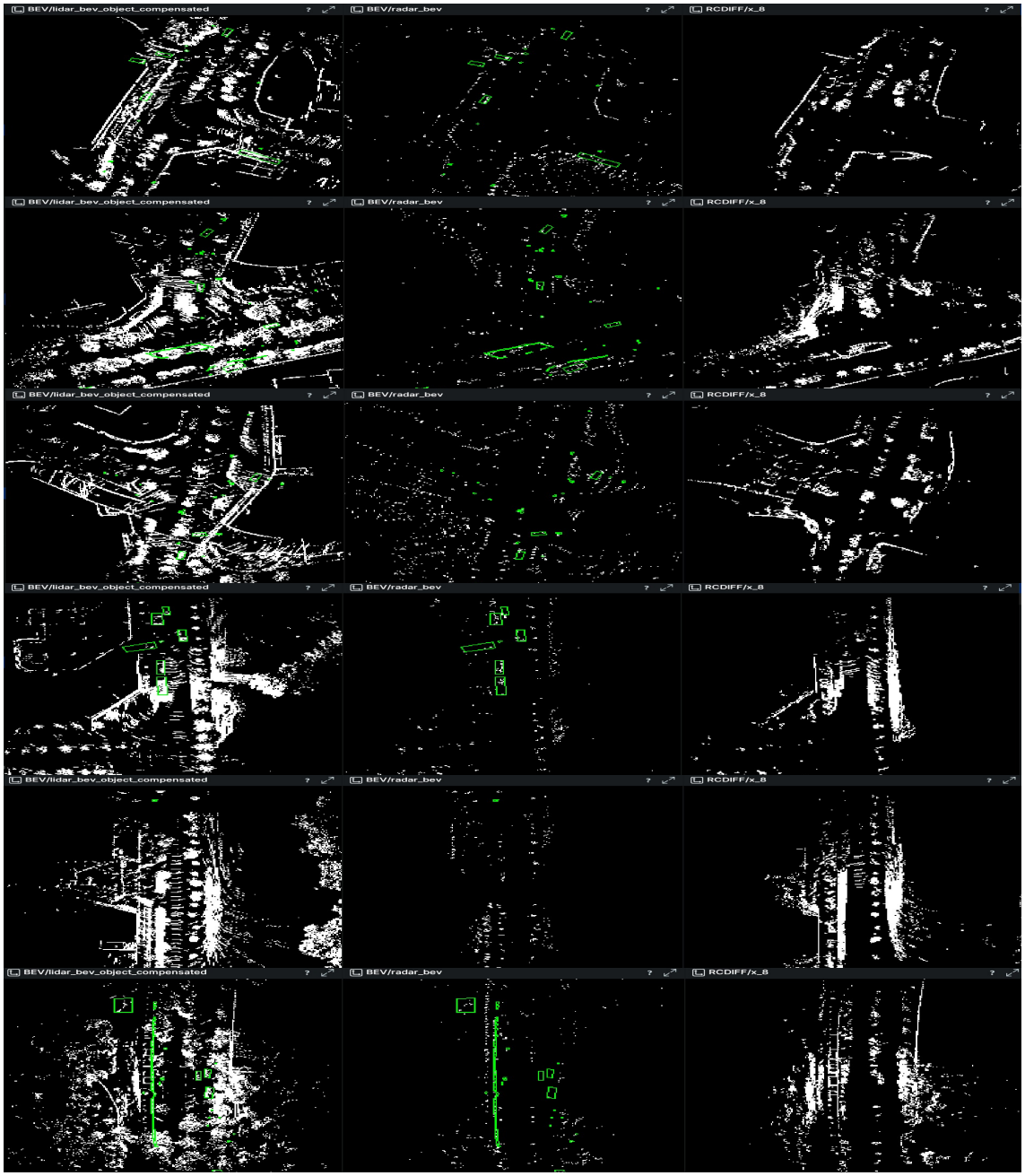


Figure 6.1: Examples of best performing RadarBEVDiffusion BEV occupancy maps. Column 1: LiDAR BEV - Column 2: Radar BEV - Column 3: Generated RADAR BEV

6.2 Occupancy Denoiser

We first analyze the performance of the diffusion-based occupancy denoiser, which forms the foundation of the proposed radar point cloud reconstruction pipeline. The model follows the architecture proposed in RadarDiff, but differs in the conditioning signal. While RadarDiff conditions the diffusion process on radar amplitude heatmaps (RAHs), our implementation instead conditions on a radar occupancy map due to its exclusive availability in nuScenes.

The occupancy denoiser achieves the strongest performance among all evaluated variants when trained solely to reconstruct occupancy. As shown in Table 6.1, the model trained with an MSE+LPIPS loss and Cut & Mix augmentation achieves the best performance with an occupancy IoU of 46% on the training set and 28% on the validation set. This configuration also yields the highest downstream detection performance with 6% mAP when training CenterPoint on the generated radar points.

Several factors influence the occupancy reconstruction performance. The dataset contains artifacts originating from imperfect ground point filtering shown in Fig. 6.2 and mismatches between the range and field of view (FOV) sensor specification between radar and LiDAR shown in Fig. 6.3. In fact, it is assumed that the specifications match mainly since the radar FOV specs are not available fully. Otherwise, only overlapping FOV regions should be considered. These artifacts introduce noisy supervision signals that can propagate into the diffusion model. Additionally, occupancy is obtained by thresholding predicted probabilities using a fixed threshold of 0.5. Exploring higher thresholds could further suppress spurious predictions by focusing only on higher-confidence occupancy estimates.

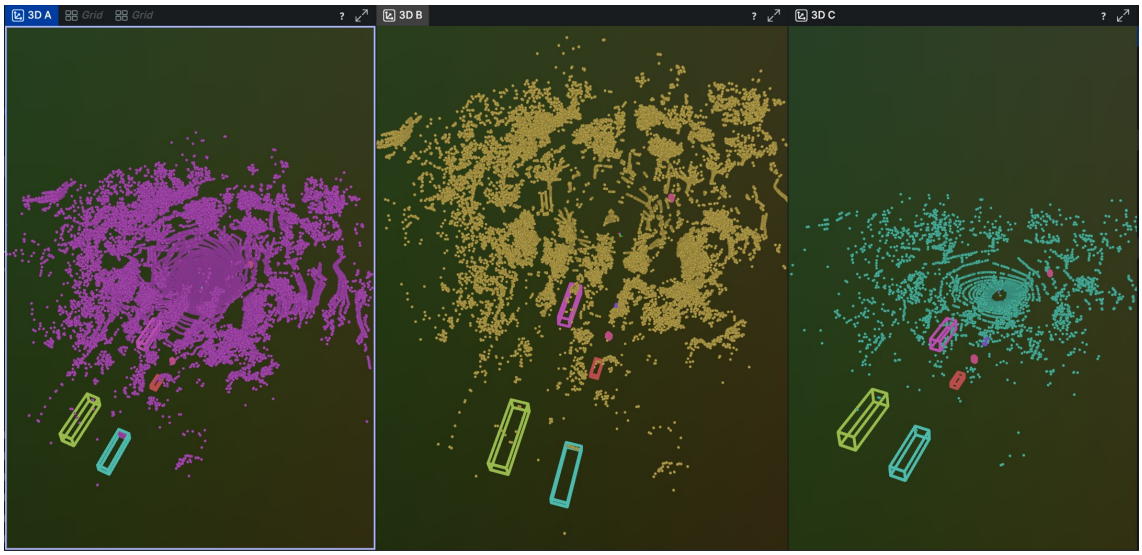


Figure 6.2: (Left 3DA): Original LiDAR PointCloud - (Middle 3DB): Filtered Ground Points LiDAR PointCloud - (Right 3D): Estimated Ground Plane PointCloud

A direct comparison to RadarDiffusion is challenging because the original work does not evaluate occupancy reconstruction using IoU metrics. Instead, in RadarDiffusion the evaluation is performed after converting the BEV representation into a point cloud. Reconstruction quality is therefore measured using point cloud similarity metrics such as Chamfer Distance, Hausdorff Distance, and F-score, rather than occupancy-based metrics like IoU.

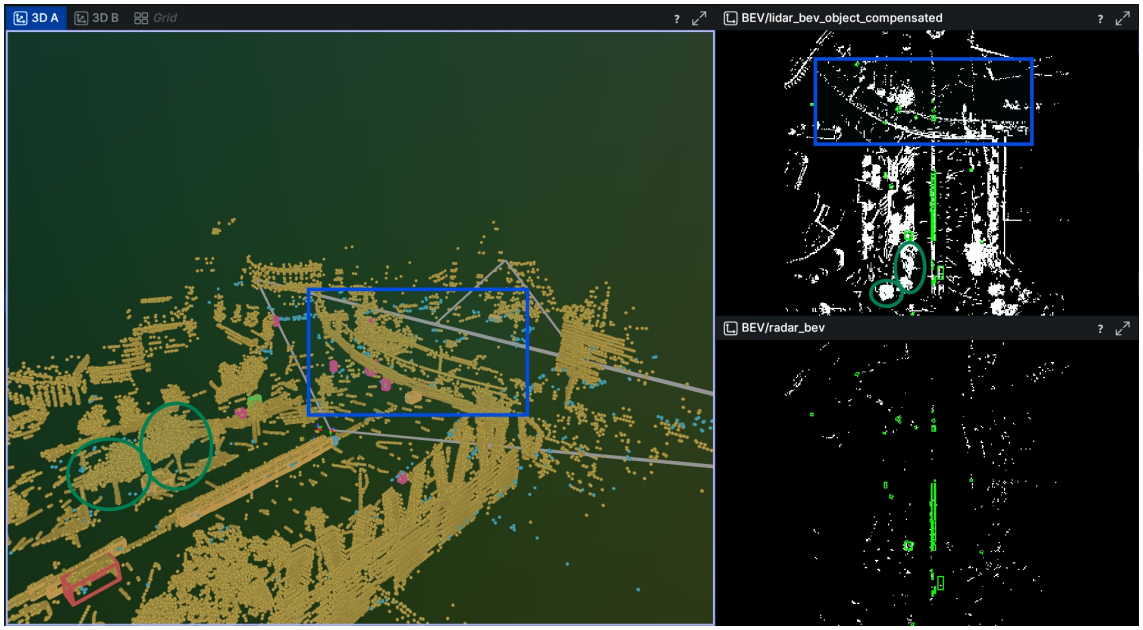


Figure 6.3: Examples of noisy signals in LiDAR BEV due to to vertical FOV mismatch between radar and LiDAR. The radar is unable detect the bridge (blue) and trees (green) as they require a much wider vertical FOV:

It would therefore be interesting to evaluate the RadarDiffusion model using IoU on their dataset and compare it to our results. Such an analysis would isolate the contribution of the conditioning signal and dataset quality. In particular, differences in performance could be attributed to the use of radar occupancy as conditioning input as well as variations in the preprocessing pipeline.

6.3 Cut & Mix Augmentation

We further analyze the effect of Cut & Mix augmentation on the diffusion training process. Cut & Mix operates by randomly swapping spatial regions between occupancy maps, increasing the diversity of spatial patterns observed during training.

The results in Table 6.1 indicate that Cut & Mix consistently provides a positive contribution to model performance. When training the occupancy denoiser with an MSE+LPIPS loss, the model trained without augmentation achieves an occupancy IoU of $39\% - 22\%$ (train-val), while enabling Cut & Mix improves the IoU to $46\% - 28\%$. This improvement also propagates to downstream detection performance, increasing the CenterPoint detection mAP from 4% to 6%.

These results suggest that Cut & Mix improves the robustness of the diffusion model by exposing it to more diverse spatial occupancy patterns. Since radar returns are sparse and noisy, such augmentation appears particularly beneficial for learning stable occupancy reconstruction.

6.4 Occupancy & Height Map Denoiser

We next investigate whether jointly predicting occupancy and a height map improves the quality of the generated radar point cloud. In this setup, the diffusion model predicts both occupancy and height values simultaneously.

However, the results indicate that adding height prediction degrades occupancy recon-

struction performance. As shown in Table 6.1, the joint occupancy-height models achieve lower occupancy IoU values compared to the occupancy-only model. For example, using an MSE+BCE loss with Cut & Mix yields an IoU of $37\% - 18\%$, which is substantially lower than the $46\% - 28\%$ achieved by the occupancy-only denoiser.

A key flaw of the current implementation is the lack of proper loss scaling and normalization for the height prediction task. Occupancy targets lie within the range $[0, 1]$, whereas height values have significantly larger numerical ranges. Without explicit normalization or loss weighting, the height prediction term dominates the optimization objective, which can negatively impact the learning of the occupancy reconstruction task.

We also experimented with different loss formulations, including MSE and a combination of MSE and BCE for occupancy prediction. However, the loss scaling issue remains present across these configurations, making it difficult to properly balance the contributions of the two objectives. Consequently, the current joint training setup should be considered a preliminary exploration. With proper height normalization and loss balancing, improved results could likely be achieved.

6.5 Radar Height Map in Conditioning Signal

Finally, we evaluate the effect of including a height map in the conditioning signal of the diffusion model. In this setting, the conditioning input consists of both the radar occupancy map and a height map derived from LiDAR ground truth.

Providing height information as part of the conditioning signal leads to moderate improvements in occupancy reconstruction. For example, the configuration using MSE+BCE loss with Cut & Mix achieves $37\% - 18\%$ IoU compared to $34\% - 15\%$ when conditioning on occupancy alone. This suggests that additional geometric context can help guide the diffusion process.

However, these improvements remain limited due to the previously mentioned loss scaling issues in the height prediction objective. Since the same height map is also used as a prediction target, errors in the height reconstruction task can propagate into the overall optimization process. As a result, the potential benefits of including height information in the conditioning signal are likely underestimated in the current experiments.

6.6 End-to-End Training

We also evaluate an end-to-end training strategy in which the diffusion-based occupancy and height denoiser is trained jointly with the CenterPoint detector. In this configuration, the model simultaneously optimizes the diffusion reconstruction objective together with the detection losses (L_{cls} and L_{reg}).

As shown in Table 6.1, this end-to-end configuration performs significantly worse than the two-stage training pipeline, achieving only 2% mAP. One reason for this degradation is that the previously discussed height prediction issues propagate into the joint optimization setup. In particular, the imbalance between occupancy and height losses negatively affects the quality of the reconstructed radar points, which in turn impacts the downstream detector.

Note that no data augmentation was applied in the end-to-end setup. Applying Cut & Mix in this scenario would require swapping ground-truth bounding boxes between paired training samples, and was not implemented in the current experiments. As demonstrated earlier, augmentation plays a significant role in improving occupancy denoiser quality. The

absence of augmentation therefore further contributes to the reduced performance of the end-to-end model.

7 Future Work

After covering the main failure modes in the Chapter 6, it is clear that the LiDAR groundtruth signal is far from perfect. First, the artifacts due to the vertical FOV mismatch can be addressed by height range filtering and better ground point filtering is still needed. Or as an alternative, switch to a data representation that is not weakened by ground points such as the one covered in Section 7.2 . While our approach to denoise both an occupancy and a height map, did not yet result in satisfying performance for 3D detection, the poor performance can be justified by the lack of proper loss scaling and/or height map normalization discussed in Section 6.4. Therefore, any future work should first evaluate the detection performance after using proper scaling. Furthermore, it is suggested to follow with an upsampling model to increase the resolution of the current outputs which will improve 3D performance.

7.1 Upsampling Model

The chosen BEV resolution for our model has a huge impact on the 3D detection performance. We demonstrate the effects of the chosen BEV resolution using the ideal LiDAR pointcloud in Table 7.1 where it is shown that mAP is reduced from 42% to 23% as a result of the reduction in pointcloud density.

Model	Preprocessing Pipeline	Pointcloud Density(Relative to Original PC)	mAP
CenterPoint	• Points From LiDAR MultiSweeps • Ego + Object Motion Compensation • Ground Point Removal • Points to BEV (Image Encoded Height) • BEV to Points	70% @ 1024×1024 32% @ 512×512	42% @ 1024×1024 23% @ 512×512

Table 7.1: Comparison of preprocessing pipelines and resulting mAP.

Generating high-resolution images directly with diffusion models is computationally expensive due to the quadratic scaling of spatial resolution with the number of processed pixels. To address this, many modern diffusion systems adopt hierarchical generation strategies. One common approach is *cascaded diffusion*, where an initial model generates a low-resolution image capturing the global structure, and subsequent diffusion models progressively upsample the result while adding higher-frequency details [27] as shown in Fig. 7.2. This design is used in several text-to-image systems such as GLIDE [28], DALL-E 2 [29], and Imagen [30], which employ dedicated diffusion upsamplers to increase image resolution.

An alternative strategy is to perform diffusion in a compressed latent space. In this approach, images are first encoded into a lower-dimensional latent representation using

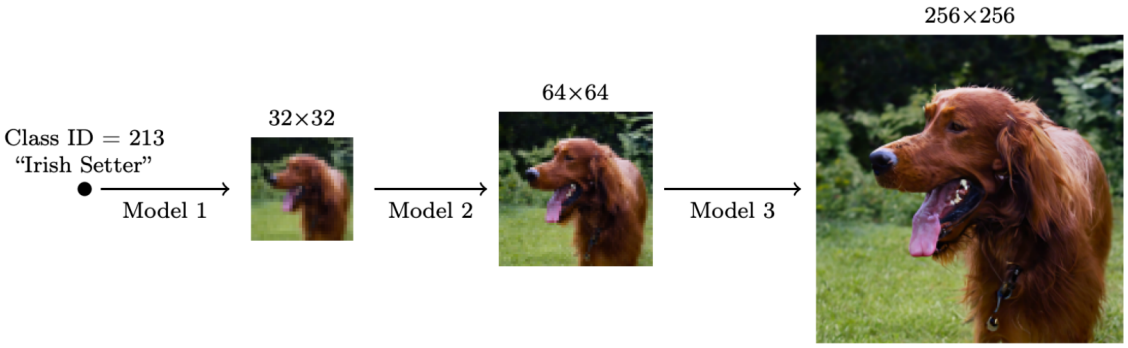


Figure 7.1: A cascaded diffusion model comprising a base model and two super-resolution models.

a variational autoencoder (VAE), diffusion is applied in this latent space, and the final high-resolution image is reconstructed using the VAE decoder. Architectures such as Stable Cascade combine hierarchical generation with latent representations to further improve efficiency [31]. These approaches have become standard for scalable high-quality diffusion-based image generation.

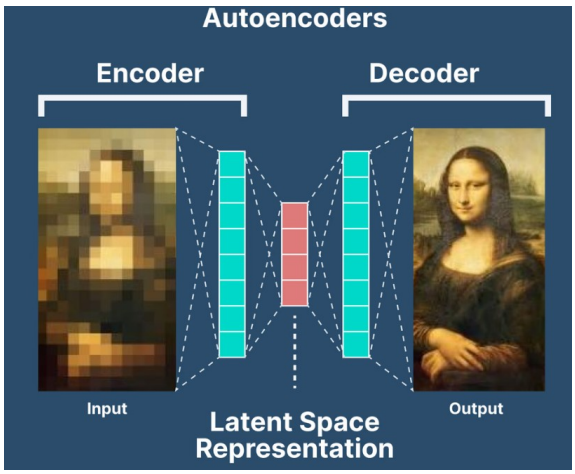


Figure 7.2: Upsampling via VAE decoder.

Given that our model is based on RadarDiffusion which is designed for autonomous navigation, the generated radar points provide only coarse spatial structure. While this maybe sufficient for navigation, 3D detection requires much richer details. Therefore, future work should consider a subsequent upsampling stage that can then refine the representation by adding finer spatial details and increasing point density which is particularly useful for 3D detection.

7.2 Diffusion In Enhanced Voxel Feature Space

In our approach the diffusion model is conditioned on the radar BEV maps, which are obtained directly from the radar pointcloud. Given how sparse and noisy the radar is, the following evidence suggests that performing diffusion in the voxel feature space is a more suitable data representation [16]. Furthermore, several relevant methods for enhancing radar feature extraction which can be considered a preprocessing step to further enhance diffusion on voxel feature space.. RadarDistill [3] propose cross-modality-alignment (CMA) to densify radar features. The authors then argue this step was critical in facilitating the transfer of knowledge from LiDAR features to radar features in their distillation framework [3]. RobuRCDET [8] introduce a 3D gaussian expanding module to filter radar points in the voxel space. The module leverages the semantic information in radar density, enhancing key points and suppressing false positives to handle extensive noise from radar corruption [8].

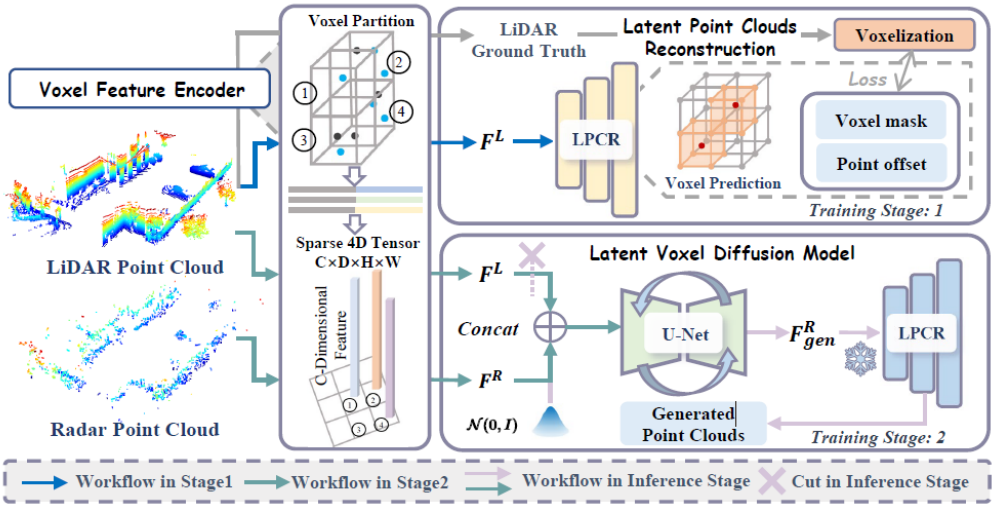
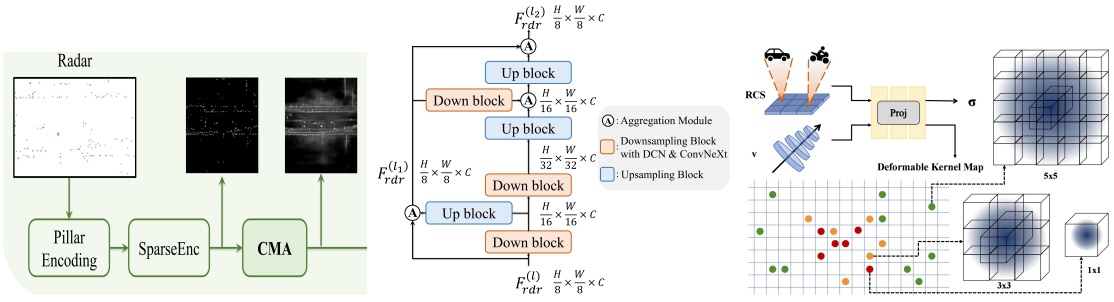


Figure 7.3: R2LDM [16]: Example of diffusion in voxel space .



(a) RadarDistill feature extraction branch.

(b) CMA Structure.

(c) 3D Gaussian Expanding

Figure 7.4: Examples of enhanced radar feature extraction.

8 Conclusion

This thesis investigated the use of diffusion models to enhance sparse radar point clouds for radar-only 3D object detection. The main motivation was to address one of the central limitations of radar perception: while radar offers long range, weather robustness, and direct velocity measurements, its sparse and noisy point clouds provide far less geometric structure than LiDAR. The goal of this work was therefore to determine whether a diffusion model could reconstruct denser LiDAR-like BEV representations from radar observations and whether such representations could improve downstream detection.

To this end, a complete pipeline was developed around the nuScenes dataset. Raw radar and LiDAR sweeps were aggregated and preprocessed into a common BEV representation consisting of occupancy and height maps. Based on this representation, a conditional diffusion model was trained to reconstruct LiDAR BEV targets from radar inputs. The generated outputs were then evaluated using both reconstruction metrics and a downstream CenterPoint detector in order to assess their practical value for 3D object detection.

The experiments lead to several important conclusions. First, the choice of intermediate representation is critical. The preprocessing analysis with ideal LiDAR point clouds showed that collapsing height information severely degrades 3D detection performance, reducing mAP from 56% for the original LiDAR benchmark to 13% when height is flattened, while preserving occupancy and image-encoded height allows a much stronger upper bound of 42% mAP at a BEV resolution of 1024×1024. This demonstrates that height information is not merely beneficial but essential for effective downstream 3D detection.

Second, the diffusion experiments showed that the occupancy-only denoiser was the most successful among the evaluated variants. The best configuration, trained with an MSE+LPIPS loss and Cut & Mix augmentation, achieved an occupancy IoU of 46% on the training set and 28% on the validation set, and yielded the strongest downstream detection result of 6% mAP. Cut & Mix augmentation was consistently beneficial, improving both reconstruction quality and detection performance. These results suggest that, under the sparse radar conditioning available in nuScenes, learning a robust occupancy prior is currently more tractable than jointly reconstructing more complete geometric structure.

Third, jointly predicting occupancy and height did not improve performance in the current setup. Although the motivation for introducing a height map was well founded from the preprocessing study, the occupancy-and-height denoisers underperformed the occupancy-only model, reaching 5% mAP in the best cases. The experimental analysis indicates that this was largely caused by loss-scaling issues: occupancy targets lie in the range $[0, 1]$, while height values are numerically much larger, allowing the height regression term to dominate optimization. As a result, the potential value of height-aware diffusion was likely underestimated in the current implementation rather than disproven.

Fourth, incorporating a radar height map in the conditioning signal provided only moderate improvements in occupancy reconstruction. For example, the MSE+BCE configuration with Cut & Mix improved from 34%/15% to 37%/18% IoU when height information was added to the conditioning input. However, these gains did not translate into a clear overall breakthrough because the same height-related optimization issues remained present in the prediction objective.

Finally, end-to-end training of the diffusion backbone together with the CenterPoint detector was not successful in its current form. The jointly trained model achieved only 2% mAP, substantially below the two-stage pipeline. This degradation can be attributed to the same instability in joint occupancy-height prediction, compounded by the absence of Cut & Mix augmentation in the end-to-end setup. In this sense, the experiments show that while end-to-end training is conceptually appealing, stable intermediate representation learning remains a prerequisite for making it effective.

Overall, this thesis shows that diffusion models can improve the global spatial structure of sparse radar representations, but that recovering the fine-grained geometry required for accurate 3D object detection remains difficult. The results support the view that radar sparsity is a genuine bottleneck and that generative modeling is a promising direction for addressing it, but they also show that the problem is highly sensitive to representation design, preprocessing, resolution, and loss balancing.

The future-work directions identified in this thesis follow naturally from these findings. Since BEV resolution was shown to have a major effect on downstream detection, an additional upsampling stage could refine the coarse structures generated by the current model and increase point density at a manageable computational cost. Likewise, diffusion in enhanced voxel feature space appears promising because it may preserve richer 3D structure than direct BEV diffusion and can be combined with radar feature enhancement techniques such as cross-modality alignment or Gaussian expanding. Together, these directions suggest that the most promising next step is not to abandon diffusion-based radar enhancement, but to move toward richer representations and higher-resolution generation.

In conclusion, the thesis provides an empirical study of diffusion-based radar enhancement for radar-only 3D object detection. While the achieved detection accuracy remains far below LiDAR-based performance and below the RadarDistill benchmark, the work clarifies several important design constraints and identifies the main failure modes of the current approach. These findings contribute to a better understanding of how diffusion models can be integrated into autonomous driving perception pipelines and establish a concrete foundation for future research on radar representation enhancement.

Bibliography

[1] Holger Caesar et al. *nuScenes: A multimodal dataset for autonomous driving*. 2020. arXiv: 1903.11027 [cs.LG]. URL: https://arxiv.org/abs/1903.11027.

[2] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. *Center-based 3D Object Detection and Tracking*. 2021. arXiv: 2006.11275 [cs.CV]. URL: https://arxiv.org/abs/2006.11275.

[3] Geonho Bang et al. *RadarDistill: Boosting Radar-based Object Detection Performance via Knowledge Distillation from LiDAR Features*. 2025. arXiv: 2403.05061 [cs.CV]. URL: https://arxiv.org/abs/2403.05061.

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. 2020. arXiv: 2006.11239 [cs.LG]. URL: https://arxiv.org/abs/2006.11239.

[5] Tero Karras et al. *Elucidating the Design Space of Diffusion-Based Generative Models*. 2022. arXiv: 2206.00364 [cs.CV]. URL: https://arxiv.org/abs/2206.00364.

[6] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. *Objects as Points*. 2019. arXiv: 1904.07850 [cs.CV]. URL: https://arxiv.org/abs/1904.07850.

[7] Jisong Kim, Minjae Seong, and Jun Won Choi. *CRT-Fusion: Camera, Radar, Temporal Fusion Using Motion Information for 3D Object Detection*. 2024. arXiv: 2411.03013 [cs.CV]. URL: https://arxiv.org/abs/2411.03013.

[8] Jingtong Yue et al. *RobuRCDet: Enhancing Robustness of Radar-Camera Fusion in Bird's Eye View for 3D Object Detection*. 2025. arXiv: 2502.13071 [cs.CV]. URL: https://arxiv.org/abs/2502.13071.

[9] Xiaomeng Chu et al. *RaCFormer: Towards High-Quality 3D Object Detection via Query-based Radar-Camera Fusion*. 2025. arXiv: 2412.12725 [cs.CV]. URL: https://arxiv.org/abs/2412.12725.

[10] Zhijian Liu et al. *BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation*. 2024. arXiv: 2205.13542 [cs.CV]. URL: https://arxiv.org/abs/2205.13542.

[11] Jiaming Song, Chenlin Meng, and Stefano Ermon. *Denoising Diffusion Implicit Models*. 2022. arXiv: 2010.02502 [cs.LG]. URL: https://arxiv.org/abs/2010.02502.

[12] Yang Song et al. *Score-Based Generative Modeling through Stochastic Differential Equations*. 2021. arXiv: 2011.13456 [cs.LG]. URL: https://arxiv.org/abs/2011.13456.

[13] Ruibin Zhang et al. *Towards Dense and Accurate Radar Perception Via Efficient Cross-Modal Diffusion Model*. 2024. arXiv: 2403.08460 [cs.CV]. URL: https://arxiv.org/abs/2403.08460.

[14] Richard Zhang et al. *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*. 2018. arXiv: 1801.03924 [cs.CV]. URL: https://arxiv.org/abs/1801.03924.

[15] Kai Luan et al. *Diffusion-Based Point Cloud Super-Resolution for mmWave Radar Data*. 2024. arXiv: 2404.06012 [cs.CV]. URL: https://arxiv.org/abs/2404.06012.

[16] Boyuan Zheng et al. *R2LDM: An Efficient 4D Radar Super-Resolution Framework Leveraging Diffusion Model*. 2025. arXiv: 2503.17097 [cs.CV]. URL: https://arxiv.org/abs/2503.17097.

[17] Seungjae Lee, Hyungtae Lim, and Hyun Myung. *Patchwork++: Fast and Robust Ground Segmentation Solving Partial Under-Segmentation Using 3D Point Cloud*. 2022. arXiv: 2207.11919 [cs.RO]. URL: https://arxiv.org/abs/2207.11919.

[18] Sumukh K Aithal et al. *Understanding Hallucinations in Diffusion Models through Mode Interpolation*. 2024. arXiv: 2406.09358 [cs.LG]. URL: https://arxiv.org/abs/2406.09358.
[19] Kostas Triaridis et al. *Mitigating Diffusion Model Hallucinations with Dynamic Guidance*. 2025. arXiv: 2510.05356 [cs.CV]. URL: https://arxiv.org/abs/2510.05356.
[20] Yuanyang Yin et al. *Focal Guidance: Unlocking Controllability from Semantic-Weak Layers in Video Diffusion Models*. 2026. arXiv: 2601.07287 [cs.CV]. URL: https://arxiv.org/abs/2601.07287.
[21] Inho Kong et al. *Error as Signal: Stiffness-Aware Diffusion Sampling via Embedded Runge-Kutta Guidance*. 2026. arXiv: 2603.03692 [cs.CV]. URL: https://arxiv.org/abs/2603.03692.
[22] Lihe Yang et al. *Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data*. 2024. arXiv: 2401.10891 [cs.CV]. URL: https://arxiv.org/abs/2401.10891.
[23] Benjin Zhu et al. *Class-balanced Grouping and Sampling for Point Cloud 3D Object Detection*. 2019. arXiv: 1908.09492 [cs.CV]. URL: https://arxiv.org/abs/1908.09492.
[24] Hanrong Ye and Dan Xu. *DiffusionMTL: Learning Multi-Task Denoising Diffusion Model from Partially Annotated Data*. 2024. arXiv: 2403.15389 [cs.CV]. URL: https://arxiv.org/abs/2403.15389.
[25] Yuqi Yang et al. "Multi-Task Dense Predictions via Unleashing the Power of Diffusion". In: *The Thirteenth International Conference on Learning Representations*.
[26] Jiahui Qu et al. "MTLSC-Diff: Multitask learning with diffusion models for hyperspectral image super-resolution and classification". In: *Knowledge-Based Systems* 303 (2024), p. 112415. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2024.112415. URL: https://www.sciencedirect.com/science/article/pii/S0950705124010499.
[27] Jonathan Ho, Chitwan Saharia, et al. "Cascaded Diffusion Models for High Fidelity Image Generation". In: *JMLR* (2022).
[28] Alex Nichol, Prafulla Dhariwal, et al. "GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models". In: (2022).
[29] Aditya Ramesh et al. "Hierarchical Text-Conditional Image Generation with CLIP Latents". In: (2022).
[30] Chitwan Saharia et al. "Imagen: Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding". In: (2022).
[31] Bilal Khalid et al. *Efficient and Robust Semantic Image Communication via Stable Cascade*. 2025. arXiv: 2507.17416 [eess.IV]. URL: https://arxiv.org/abs/2507.17416.

Technical
University of
Denmark

Ørsteds Plads, Building 343
2800 Kgs. Lyngby
Tlf. 4525 1700

https://electro.dtu.dk/