arXiv:2404.04629v2 [cs.CV] 24 Sep 2024

Diffusion Model for Robust Multi-Sensor Fusion in 3D Object Detection and BEV Segmentation

Duy-Tho Le[1], Hengcan Shi[2†], Jianfei Cai[1], and Hamid Rezatofighi[1]
[1]Monash University, [2] Hunan University
tho.le1@monash.edu hengcan.shi@gmail.com
https://ldtho.github.io/DifFUSER/

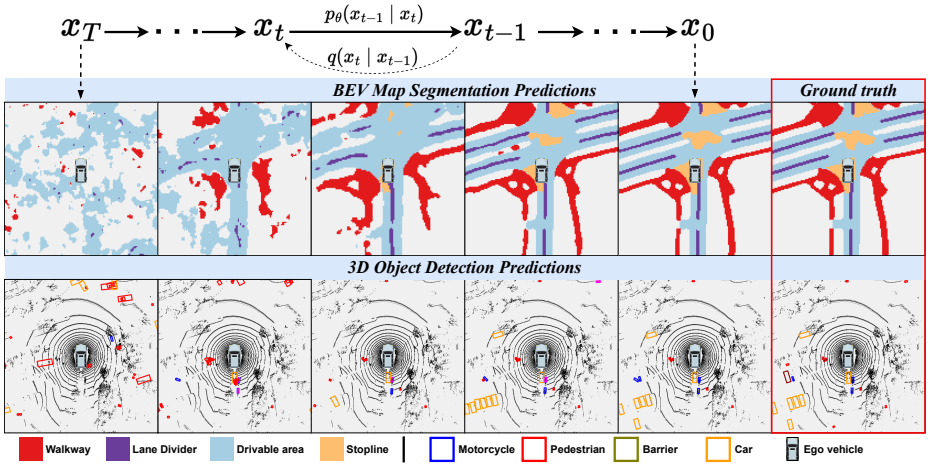


Fig. 1: We propose to use denoising diffusion process for multi-modal BEV features fusion, which are then used for 3D object detection and BEV map segmentation.

Abstract. Diffusion models have recently gained prominence as powerful deep generative models, demonstrating unmatched performance across various domains. However, their potential in multi-sensor fusion remains largely unexplored. In this work, we introduce "DifFUSER", a novel approach that leverages diffusion models for multi-modal fusion in 3D object detection and BEV map segmentation. Benefiting from the inherent denoising property of diffusion, DifFUSER is able to refine or even synthesize sensor features in case of sensor malfunction, thereby improving the quality of the fused output. In terms of architecture, our DifFUSER blocks are chained together in a hierarchical BiFPN fashion, termed cMini-BiFPN, offering an alternative architecture for latent diffusion. We further introduce a Gated Self-conditioned Modulated (GSM) latent diffusion module together with a Progressive Sensor Dropout Training (PSDT) paradigm, designed to add stronger conditioning to the diffusion process and robustness to sensor failures. Our extensive evaluations on the Nuscenes dataset reveal that DifFUSER not only achieves state-of-the-art performance with a 70.04% mIOU in BEV map segmentation tasks but also competes effectively with leading transformer-based fusion techniques in 3D object detection.

† Corresponding author

Keywords: 3D Object Detection · BEV Map Segmentation · Diffusion

1 Introduction

3D object detection and Bird's Eye View (BEV) map segmentation are fundamental tasks in autonomous driving. The former involves detecting objects in 3D space, while the latter segments BEV maps into semantic categories. A growing trend in this field is the fusion of features from multiple sensors, leveraging their complementary strengths. For example, 3D point clouds provide geometric data for 3D object detection, but lack the rich color information crucial for BEV semantic segmentation —a gap effectively filled by 2D images.

Recent transformer-based fusion methods [1,47,49] have set new benchmarks in 3D object detection by learning feature mappings from both sensors through cross-attention mechanisms. Nevertheless, these methods are less adaptable for extra tasks like BEV map segmentation. An alternative is to create unified BEV representations from both sensors, as seen in LSS-based [31] approaches. However, these often yield sub-optimal results due to: 1) inadequately designed fusion module architectures that fail to capture the intricate relationship between the sensors, and 2) the inherent noise between different modalities, causing the fused features to be noisy and inaccurate.

To address these challenges, we explore the use of generative models, particularly Diffusion Probabilistic Models (DPMs), for multi-modal fusion and denoising. DPMs have shown promise in various applications, including image generation [9, 37] and object detection [3]. Yet, the potential of DPMs in multi-sensor fusion, especially between 3D point clouds and 2D multi-view images, remains unexplored. To this end, we propose *DifFUSER*, a conditional diffusion-based generative model with enhanced fusion architecture designed for multi-modal learning in 3D perception. It processes features from both sensors to output a refined BEV representation, which is subsequently directed to task-specific heads for 3D object detection and BEV map segmentation. We also show DifFUSER's robustness in synthesizing new features to compensate for missing modality, minimising performance loss and ensuring reliability even with compromised sensor.

In terms of architecture, our DifFUSER blocks are chained together in a hierarchical BiFPN (Bidirectional Feature Pyramid Network) fashion [14, 39], termed cMini-BiFPN, offering an alternative architecture for latent diffusion, particularly good in handling detailed and multi-scales features from different sensors. This architectural change is complemented by a Gated Self-conditioned Modulated (GSM) latent diffusion module, designed to strengthen the conditioning modulation during diffusion, leading to more precise denoising and feature enhancement. We train DifFUSER end-to-end using our proposed Progressive Sensor Dropout Training (PSDT) paradigm, together with task-specific losses, which enhance the model's robustness and capability in denoising corrupted BEV features and generating more accurate outputs for downstream tasks.

Fig. 2 presents the output activation maps from the BEV encoder of the baseline BEVFusion [23] and our DifFUSER module. The activation map from our BEVFusion module is blurry, which might hinder the downstream task performance. In contrast, our fusion module, with better architecture design and the denoising property of the diffusion model, can effectively mitigate the noises and intrinsic differences between different sensors, notably improving the predictions and strong robustness against sensor malfunctioning. We conduct extensive experiments on the large-scale Nuscenes dataset [2] to evaluate the performance and design choice of DifFUSER. The results show that DifFUSER achieves SOTA performance (70.04% mIOU) in the BEV map segmentation task and performs on par with SOTA transformer-based fusion methods in the 3D object detection task. We also conduct ablation studies to show the effectiveness of each component in DifFUSER.

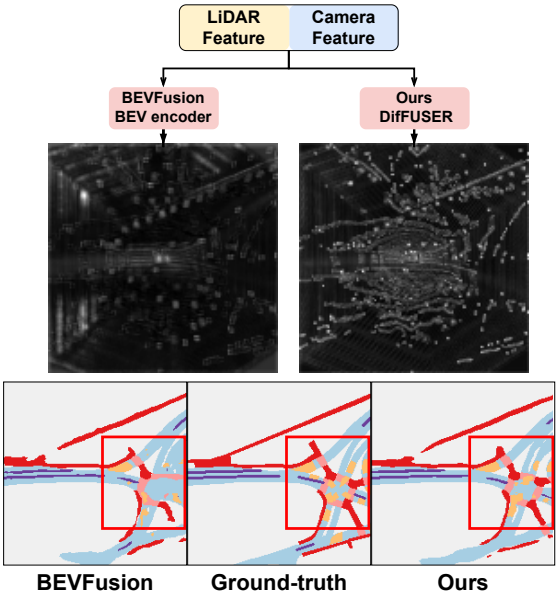


Fig. 2: Comparison of our DifFUSER fusion module with BEV encoder of the baseline. Our fusion module's output activation map is much more expressive than the baseline's, resulting in much better performance in downstream tasks.

In summary, our contributions are listed as follows.

- We are first to introduce a new DPM-based approach for multi-modal fusion in 3D perception. With a well-designed diffusion architecture (cMini-BiFPN) and training paradigm (PSDT), our method showcases enhanced multi-modal feature fusion and denoising capabilities, as well as robustness to sensor failure by generatively synthesizing missing sensor data.
- We propose a Gated Self-Conditioned Modulated (GSM) latent diffusion mechanism within the DifFUSER blocks, reinforcing the fusion process with stronger conditioning and shaping the diffusion trajectory towards high-quality features.
- We conduct extensive experiments on the Nuscenes dataset [2] to evaluate the performance of DifFUSER. The results show that DifFUSER surpasses the current SOTA by 7.34% in BEV map segmentation tasks and performs on par with SOTA transformer-based fusion methods in 3D object detection.

2 Related Work

Single-modality 3D perception. Single-modality 3D perception has been widely studied in the past few years and mainly comprises two categories: 1) Lidar-based and 2) camera-based methods. Camera-based detection evolved from early dense prediction pipelines [42, 43] to sophisticated methods like DETR3D [44], BEVFormer [17] and BEVDepth [16], which integrate Bird's Eye View (BEV) representations with transformer attention for multi-view fusion. In contrast, LiDAR-based detection relies on point cloud data, with pioneering architectures like PointNet [34] processing raw point clouds and subsequent developments like VoxelNet [55] transforming into structured formats like voxels [6,7,48], range images [8,38], and pillars [13,14,22]. These approaches reflect the ongoing evolution in object detection, leveraging the unique strengths of each modality.
Multi-modality fusion approaches in 3D perception. Fusion-based approaches have been widely adopted in 3D perception tasks and can be categorised into 3 main groups: 1) early fusion, 2) late fusion, and 3) feature fusion. ***Early fusion*** methods directly fuse the raw data from both sensors, either by augmenting the raw point cloud with image features [35,40,46] or leveraging the image features to guide the point cloud feature extraction [28,33]. The fusion is performed sequentially, introducing additional latency to the system and prone to error propagation. ***Late fusion*** methods [29,30] operate on the predictions from both sensors, either by fusing the predictions [30] or by fusing the features extracted from both sensors [29]. The fusion is performed at the end of the pipeline, which is more efficient than early fusion methods but fails to capture the feature-to-feature relationships between both sensors. Thus the fusion results are often noisy and sub-optimal. ***Feature fusion*** methods [1,23,47,49] have deeper feature fusion, where the features from both sensors are fused at multiple levels of the network. Although the fused representations are more expressive and yield better performance, LiDAR and camera features are intrinsically different and aligning their features is still a challenging task. Most current SOTA methods fall into this category. Particularly, *transformer-based fusion* methods [1,47,49] have been proposed to fuse the features from both sensors, leveraging feature-to-feature relationships captured by the transformer encoder-decoder architecture and achieving SOTA performance in 3D object detection. However, transformer-based fusion is usually computationally demanding and is not easy to adapt to other perception tasks such as BEV map segmentation. Another way of fusing multi-modalities is to construct an *unified BEV representation* [18,23] from both sensors using LSS-based [31] backbone to lift the 2D image features to 3D space, and then apply multiple task-specific heads to the fused representation. This approach is more efficient and can be easily extended to multiple perception tasks. However, the fusion result is often sub-optimal due to the inherent noise and misalignment between different sensors.

Our DifFUSER belongs to the feature-fusion sub-category, we aim to improve the fusion architecture and leverage the denoising property of generative models to mitigate the noises and intrinsical differences between different sensors, thus improving downstream tasks' performance.

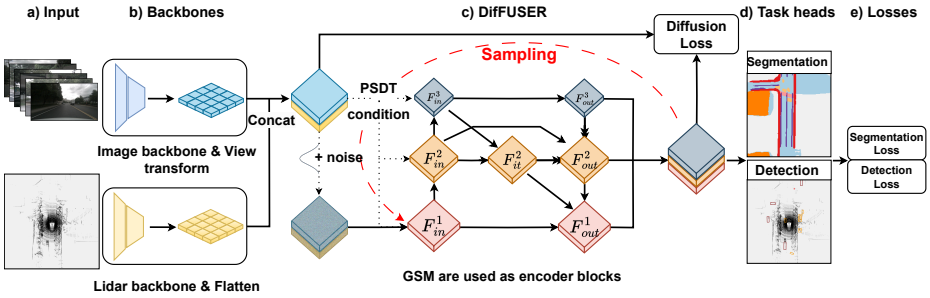


Fig. 3: Our DifFUSER framework is structured to first process input data—comprising both point clouds and images—through respective backbones to create initial latent features. These features are then concatenated and fed into the DifFUSER blocks. Within these blocks, the concatenated feature is used as condition (partially masked out) to iteratively denoise the corrupted features, enhancing its quality at each step. The output feature is then used for downstream tasks.

Generative models for 3D perception. Generative models have been recently adopted in 3D perception tasks. However, most of them are for text-to-3D [19, 32], image-to-3D [20], and 3D trajectory prediction [11]. We notice that there are concurrent works [54, 56] that utilise diffusion models for 3D object detection, but they only operate on single-modality (either camera or LiDAR) and do not utilise diffusion for multi-modal fusion.

In contrast, we aim to leverage the denoising property of generative models to mitigate the noises and intrinsic differences between different sensors, tackling some of the most fundamental tasks in autonomous driving, such as 3D object detection and BEV map segmentation. Our DifFUSER is the first to adapt diffusion models for multi-modal fusion for 3D perception tasks. The generated BEV feature can be shared and optimized end-to-end with downstream tasks.

3 Methodology

In this section, we introduce DifFUSER, a conditional generative model with enhanced fusion architecture, designed for multi-modal and multi-task learning within the scope of 3D object detection and BEV map segmentation. Our DifFUSER features take the extracted features from lidar and camera sensors and process them in three ways: one used as the diffusion target, one infused with Gaussian noise (x_t^F) and one is partially masked by PSDT ($\tilde{x}_0^F$). $\tilde{x}_0^F$ and x_t^F are then input into the cMini-BiFPN (with GSM as encoder blocks) module, where they are denoised and refined. This enhanced representation is finally utilised by task-specific heads for precise 3D object detection and BEV map segmentation.

3.1 Fuser Architecture Design (cMini-BiFPN).

In autonomous driving, where fast processing is the top priority, the need to ensure both the performance and efficiency of the diffusion module is impor-

tant. Thus, we introduce a conditional BiFPN-like diffusion architecture, which we term as the conditional-Mini-BiFPN (cMini-BiFPN). Note that we use the term cMini-BiFPN to indicate the bi-directional feature fusion alone, without the GSM module. As depicted in Fig. 3, the initial three blocks $F^1_{in}, F^2_{in}, F^3_{in}$—the encoder blocks—process both partially masked original $\tilde{x}^F_0$ and perturbed BEV features x^F_t, where $\tilde{x}^F_0$ serves as a guide for the diffusion process, aiding in the progressive refinement of the corrupted BEV feature. Subsequent blocks $F^1_{out}, F^2_{out}, F^3_{out}$—the decoder blocks—accept the conditioned BEV representation and apply iterative top-down and bottom-up bi-directional fusion to generate the final BEV feature. Particularly,

$$F^2_{it} = \text{Conv}\left(\delta\left(\frac{\left(w'_1 \cdot F^2_{in} + w'_2 \cdot Resize\left(F^3_{in}\right)\right)}{w'_1 + w'_2 + \varepsilon}\right)\right) \tag{1}$$

$$F^2_{out} = \text{Conv}\left(\delta\left(\frac{\left(w''_1 \cdot F^2_{in} + w''_2 \cdot F^2_{it} + w''_3 \cdot Resize\left(F^1_{out}\right)\right)}{w''_1 + w''_2 + w''_3 + \varepsilon}\right)\right) \tag{2}$$

where δ denotes the Swish activation function [39], F^2_{it} represents the intermediate fusion result of F^2_{in} and F^3_{in}, and F^2_{out} is computed by blending F^2_{it}, F^1_{out}, and F^2_{in}. The weights $w'_{1,2,3}$ and $w''_{1,2,3}$ are learnable and optimised during training. F^1_{out} and F^3_{out} are derived similarly to F^2_{out}. The final feature map is the concatenation of F^1_{out}, F^2_{out}, and F^3_{out}, which is then processed through a Transfusion-L [1] detection head and a segmentation head [23], facilitating both 3D object detection and BEV map segmentation. We believe that this streamlined and efficient cMini-BiFPN architecture is well-suited for the demanding diffusion processes in autonomous driving applications, balancing computational efficiency with the necessity for expeditious sampling.

3.2 Progressive Sensor Dropout Training (PSDT)

Sensor reliability in autonomous driving is crucial. Cameras and LiDAR systems, the eyes of robotic agents, can occasionally be occluded or fail, potentially compromising safety and operational efficiency. Recognising this, we propose the Progressive Sensor Dropout Training (PSDT) method, which enhances our DifFUSER model's robustness and adaptability, particularly in scenarios where sensor inputs may be occluded or failed. It enables the model to generate or reconstruct missing features from camera or LiDAR inputs by leveraging the learned distribution across both modalities. This approach ensures consistently high performance, even in cases of sensor malfunction or when sensor data is partially missing, closely simulating real-world operational challenges.

Specifically in PSDT, x^F_0, the input for the whole fusion process, containing features from both camera and LiDAR, is utilised in three key ways: as a training target, as a noisy input for the diffusion module, and as a diffusion condition with randomly dropped camera or LiDAR features. To simulate the conditions of sensor dropout or malfunction, we progressively increase the dropout rate for either the camera or LiDAR inputs from 0% to a predefined maximum $\alpha = 25$

during training. This is represented as:

$$F^m_{\text{masked}} = F^m \odot \text{Bernoulli}\left(\frac{\alpha}{100} \cdot \frac{e}{E}\right) \tag{3}$$

where e represents the current epoch out of the total E epochs, defining the dropout probability $p_{\text{dropout}}(e)$ that dictates the chance of each feature in modality F^m being dropped. This strategic process, achieved by applying a binary mask generated from a Bernoulli distribution based on $p_{\text{dropout}}(e)$, not only trains the model to effectively denoise and generate more expressive features but also minimises its reliance on any single sensor, thereby enhance its capability to handle incomplete sensor data with greater resilience.

3.3 Gated Self-Conditioned Modulated (GSM) Diffusion Module

Departing from traditional diffusion models [9,37], which primarily use just t as the condition, the GSM module leverages both the temporal (time step t) and the sensor information contained within the partially masked latent sample ($\tilde{x}_0^F$) to guide the diffusion process. This requires stronger conditioning on the noisy latent to shape the diffusion trajectory towards generating high-quality features. This dual-conditioning approach, augmented by the PSDT paradigm and task-specific losses, ensures the diffusion process focuses on feature enhancement and not mere replication, encourage the synthesis of more expressive features.

Specifically, in DifFUSER, the GSM module is used in encoder blocks $F^1_{in}, F^2_{in}, F^3_{in}$ (see Fig. 3). The GSM diffusion process is encapsulated by a FiLM-like (Feature-wise Linear Modulation) operation. This modulation computes parameters that dynamically adjust the diffusion trajectory based on the conditions $\tilde{x}_0^F$. The modulation formula is defined as:

$$\begin{aligned}
\tilde{x}_0^F &= \tilde{x}_0^F + time_emb(t), \\
\gamma_t &= \sigma(W_\gamma \cdot \tilde{x}_0^F + b_\gamma), \\
\alpha_t &= W_\alpha \cdot \tilde{x}_0^F + b_\alpha, \\
\beta_t &= W_\beta \cdot \tilde{x}_0^F + b_\beta, \\
\hat{x}_0^F &= \gamma_t \cdot (x_t^F \cdot (1 + \alpha_t) + \beta_t).
\end{aligned} \tag{4}$$

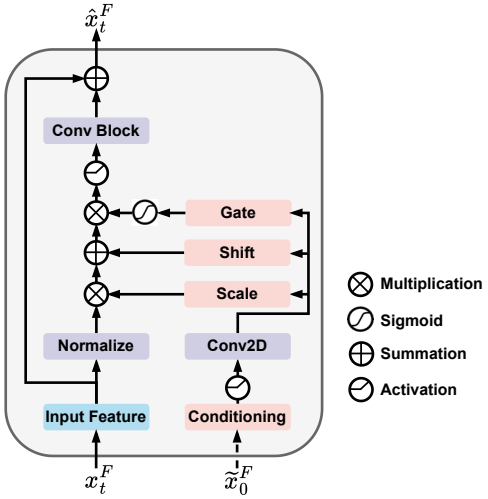


Fig. 4: GSM diffusion block.

Here, $\tilde{x}_0^F$ is the version of x_0^F that may be partially masked due to PSDT. The parameters α_t, β_t, and γ_t modulate the noisy input feature x_t^F via scaling and shifting, with γ_t acting as a sigmoid-gated self-conditioning element to recalibrate

layer activation based on the feature itself. These parameters are derived from the condition $\tilde{x}_0^F$ using a simple convolution block, we deliberately limit gradient flow when introducing noise to x_0^F to prevent model memorisation. The architecture of our gated self-conditioned modulated diffusion mechanism is detailed in Fig. 4.

3.4 DifFUSER training and loss

DifFUSER is trained end-to-end in either single-task or multi-task fashion, together with task-specific losses, which encourage the module to not just denoising but also generating more expressive BEV features serving the downstream tasks. In the forward noising process, Gaussian noise ε is gradually added to the extracted features x_0^F via a discrete *stochastic differential equation* (SDE):

$$q(x_t^F, x_0^F) = \sqrt{\bar{\alpha}_t} x_0^F + \varepsilon\sqrt{1-\bar{\alpha}_t} \tag{5}$$

where $\bar{\alpha}_t$ is a hyperparameter and $\varepsilon \sim \mathcal{N}(0, I)$. If the time step t is large, x_t^F would become Gaussian noise (see Appendix for more discussion).

The reverse denoising process is a discrete SDE that gradually maps a Gaussian noise into a sample. At each time step, given x_t^F, t, and $\tilde{x}_0^F$, it predicts the next reverse step:

$$p^{(t)}(x_{t-1}^F | x_t^F, \tilde{x}_0^F, t) = \begin{cases} \mathcal{N}\left(p_\theta^{(1)}(x_1^F, \tilde{x}_0^F, t), \sigma^2 I\right) & \text{if } t = 1 \\ q(x_{t-1}^F | x_t^F, p_\theta^{(t)}(x_t^F, \tilde{x}_0^F, t)) & \text{otherwise.} \end{cases} \tag{6}$$

DDPM sampler [9] is not considered a practical model due to its slow sampling speed. Later, we explore and compare the potential of using training-free faster diffusion samplers, such as the first-order sampler DDIM [37], or higher-order samplers like DPM-Solver++ [26] and DEIS [52].

Diffusion loss. Instead of using ε-prediction from DDPM [9], we train our model to predict the denoised and unmasked sample x_0^F from the noisy x_t^F at each time step, and use a simple mean-squared error loss on the prediction, thanks to the parameterisation of μ_θ as a denoising network p_θ :

$$\mathcal{L}_{diffusion} = \mathbb{E}_{t\sim[1,T], \widetilde{x}_t^F \sim q_t}\left[\left\| p_\theta^{(t)}(x_t^F, t, \tilde{x}_0^F) - x_0^F \right\|^2\right] \tag{7}$$

where $\widetilde{x}_0^F$ is the original partially masked feature from both sensors, x_t^F is the noisy feature at time step t, q_t is the distribution of x_t^F at time step t, and $p_\theta^{(t)}(x_t^F, t, x_0^F))$ is the predicted $\hat{x}_0^F$ at time step t.

Segmentation loss. We use focal loss $\mathcal{L}_{seg}$ for the segmentation head, as in [23].

Detection loss. Similar to [1,23], the Hungarian set prediction loss is applied to the detection head, which is a combination of the classification focal loss and the smooth L1 regression loss. Top K predictions are selected and matched with the ground truth boxes by Hungarian matching. The detection loss is then calculated as $\mathcal{L}_{detection} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{reg}\mathcal{L}_{reg}$.

Total loss. The total loss is the weighted sum of the diffusion loss, the detection loss, and the segmentation loss, depending on the training mode:

$$\mathcal{L}_{total} = \mathcal{L}_{diffusion} + \lambda_{seg}\mathcal{L}_{seg} + \lambda_{det}\mathcal{L}_{detection} \tag{8}$$

where $\lambda_{diff}, \lambda_{det}, \lambda_{seg}$ are the weights for each loss.

4 Experiment

Our experiments focus on two critical tasks in autonomous driving: 3D object detection and Bird-Eye-View (BEV) map segmentation. 3D object detection is essential for accurately identifying and locating objects within a 3D space, such as vehicles, pedestrians, and road obstacles. BEV map segmentation, on the other hand, involves partitioning the bird's-eye view of the driving environment into different semantic categories, aiding in path planning and navigation. Both tasks benefit significantly from the fusion of LiDAR and camera data, offering comprehensive spatial and contextual understanding.

Dataset. We report DifFUSER's performance on the nuScenes [2] dataset, a challenging large-scale benchmark with multiple multi-modal data annotations for autonomous driving (3D bounding box and map segmentation), which is suitable for our method. This dataset comprises roughly 1000 scenes, split into 700 for training, 150 for validation, and another 150 for testing. Each scene in the dataset represents a 20-second duration, recorded using six cameras and a LiDAR, along with additional data from five radars. The cameras, calibrated for precision, capture RGB images at a rate of 12 FPS, encompassing a full 360-degree horizontal field of view. Simultaneously, the 32-beam LiDAR scans the environment at 20 FPS. We do not use Radar data in our experiments.

BEV map segmentation. BEV map segmentation, an integral part of autonomous driving, involves segmenting the bird's-eye view of the road into distinct semantic categories. This task aids in path planning and navigation by providing a comprehensive overview of the road layout and relevant features. We adopt the training approach of the baseline BEVFusion [23], assessing IOUs of *drivable areas, pedestrian crossings, walkways, stop lines, carparks, dividers*, and class-averaged mean IoU (mIOU). Our evaluation is conducted in a 100m x 100m area centered around the ego vehicle and reports mIOU.

3D object detection. Next, we evaluate DifFUSER on 3D object detection. NuScenes [2] dataset provides 3D bounding boxes for 10 object classes, namely *car, truck, bus, trailer, construction vehicle, pedestrian, motorcycle, bicycle, traffic cone, barrier*. Adhering to the dataset's established protocols, we transform the data from the preceding nine frames into the current point cloud frame for both training and evaluation. We use the official evaluation metrics for 3D object detection, namely *NDS* and *mAP*. And mean Average Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error (mAOE), mean Average Velocity Error (mAVE), and mean Average Attribute Error (mAAE) are reported. Higher *NDS* and *mAP* indicate better performance.

Table 1: DifFUSER outperforms state-of-the-art multi-sensor fusion methods on BEV map segmentation on nuScenes *val* with a 6.4% improvement across categories. Abbreviations: Mod. = Modality, Driv. = Drivable, Ped. Cr. = Pedestrian Crossing, Wlk. = Walkway, Stp. Ln. = Stop Line, Cpk. = Carpark, Div. = Divider. Relative improvements over the baseline are shown below. Bold, blue, and underlined values indicate the best, second-best, and baseline performance, respectively.

Method	**Mod.↑**	**Driv.↑**	**Ped. Cr.↑**	**Wlk.↑**	**Stp. Ln.↑**	**Cpk.↑**	**Div.↑**	**Mean↑**
LSS [31]	C	75.4	38.8	46.3	30.3	39.1	36.5	44.4
CVT [53]	C	74.3	36.8	39.9	25.8	35.0	29.4	40.2
M^2BEV [45]	C	77.2	-	-	-	-	40.5	-
BEVFusion [23]	C	81.7	54.8	58.4	47.4	50.7	46.4	56.6
PointPillars [13]	L	72.0	43.1	53.1	29.7	27.7	37.5	43.8
CenterPoint [50]	L	75.6	48.4	57.5	36.5	31.7	41.9	48.6
PointPainting [40]	LC	75.9	48.5	57.1	36.9	34.5	41.9	49.1
MVP	LC	76.1	48.7	57.0	36.9	33.0	42.2	49.0
BEVFusion [23]	LC	85.5	60.5	67.6	52.0	57.0	53.7	62.7
DifFUSER (Ours)	LC	**89.7**	**69.0**	**74.3**	**61.9**	**63.0**	**62.3**	**70.0**
Improvement		↑4.2	↑8.5	↑6.7	↑9.9	↑6.0	↑8.6	↑7.3

Training Details. Unless specified otherwise, all models are trained using PSDT, with a batch size of 3 on two NVIDIA 4090 GPUs. Following [23], we keep Swin-T [21] and Voxelnet [55] as image and LiDAR backbones, respectively. We reduce the resolution of camera images to 448×800 pixels and convert the LiDAR point cloud into voxels, using a size of 0.075 meters for detection and 0.1 meters for segmentation. To accommodate different spatial requirements of detection and segmentation, we use grid sampling with bilinear interpolation to adjust BEV feature maps before each specific task. For 3D object detection, we first pretrain LiDAR only backbone for 15 epochs, then finetune the fusion module for 10 epochs with AdamW [25] optimizer with a learning rate of 3.75e-5 and a Cosine Annealing scheduler [24]. For BEV Map Segmentation, we train the model for 20 epochs with AdamW optimizer with a learning rate of 2e-5 and a OneCycle scheduler [36]. We use the same training strategy in our ablation study for fair comparison. We also study joint 3D detection and segmentation training in our ablation study using a shorter training schedule for both tasks.

5 Dicussion and Comparison with State-of-the-arts

BEV Map Segmentation. Tab. 1 shows the comparison of BEV map segmentation performance on the NuScenes *val* set. DifFUSER exhibits a remarkable performance leap over the baseline BEVFusion [23]. While BEVFusion [23] achieves an mIOU score of 62.7, DifFUSER significantly elevates this to 70.04, marking a substantial 7.3% improvement. This pronounced advancement is consistent across various categories, as in Tab. 1 all outperforming their respective scores in BEVFusion [23]. These results underscore DifFUSER's capability to enhance the quality of fused features, which is crucial in the segmentation task.

Table 2: Comparison of 3D object detection performance on the Nuscenes *test* dataset. Mod. abbreviates Modality, where L and C denote LiDAR and Camera, respectively. Relative improvement over the baseline is shown in a separate row. Bold, blue, and underlined values indicate the best, second-best, and baseline performance, respectively.

Methods	Mod.↑	NDS↑	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
PointPainting [40]	LC	61.0	54.1	0.380	0.260	0.541	0.293	0.131
PointAugmenting [41]	LC	71.1	66.8	0.253	0.235	0.354	0.266	0.123
MVP [51]	LC	70.5	66.4	0.263	0.238	0.321	0.313	0.134
FusionPainting [46]	LC	71.6	68.1	0.256	0.236	0.346	0.274	0.132
UVTR [15]	LC	71.1	67.1	0.306	0.245	0.351	0.225	0.124
TransFusion [1]	LC	71.7	68.9	0.259	0.243	0.359	0.288	0.127
BEVFusion [18]	LC	71.8	69.2	0.260	0.250	0.370	0.270	0.130
BEVFusion [23]	LC	72.9	70.2	0.261	0.239	0.329	0.260	0.134
DeepInteraction [49]	LC	73.4	70.8	0.257	0.240	0.325	0.245	0.128
MSMDFusion [12]	LC	73.0	71.0	0.250	0.240	0.330	0.260	0.140
LinK [27]	LC	73.4	69.8	0.258	0.229	0.312	0.234	0.136
FocalFormer3D [5]	LC	73.9	71.6	0.250	0.240	0.330	**0.220**	0.130
LargeKernel-F [7]	LC	**74.1**	71.1	**0.236**	**0.228**	**0.298**	0.241	0.131
CMT [47]	LC	**74.1**	**72.0**	0.279	0.235	0.308	0.259	**0.112**
DifFUSER (ours)	LC	73.8	71.3	0.250	0.241	0.317	0.244	0.130
Improvement		↑1.9	↑1.0	↓0.011	↑0.002	↓0.012	↓0.016	↓0.004

Table 3: Results for 3D object detection on NuScenes (val). Bold, blue, and underlined values indicate the best, second-best, and baseline performance, respectively. Mod. stands for Modality.

Methods	Mod.	NDS↑	mAP↑
TransFusion [1]	L	70.1	65.1
FUTR3D [4]	LC	68.3	64.5
UVTR [15]	LC	70.2	65.4
TransFusion [1]	LC	71.3	67.5
BEVFusion [23]	LC	71.4	68.5
DeepInteration [49]	LC	72.6	69.9
CMT [47]	LC	**72.9**	**70.3**
DifFUSER	LC	72.6	70.0
Improvement		↑1.2	↑1.5

Fig. 5: Performance Comparison of DifFUSER with and without Sensor Drop (either Lidar or Camera)

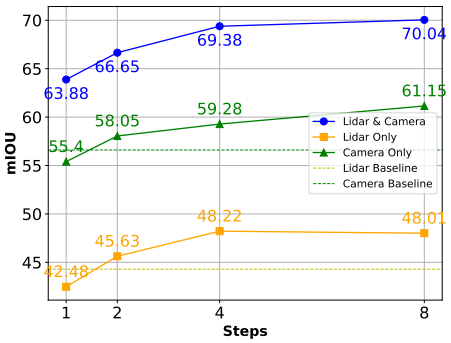


Fig. 6 shows the qualitative results of DifFUSER's BEV map segmentation on the NuScenes *val* set. BEVFusion tends to produce noisy segmentation results, especially at further distances, where the LiDAR point cloud is sparse and sensor misalignment is more pronounced. In contrast, DifFUSER produces more accurate segmentation results, with finer details and less noise.

3D Object Detection. Tab. 2 shows the comparison of 3D object detection performance on the NuScenes *test* set. DifFUSER demonstrates notable improvement over the baseline BEVFusion [23]. While BEVFusion [23] already showed

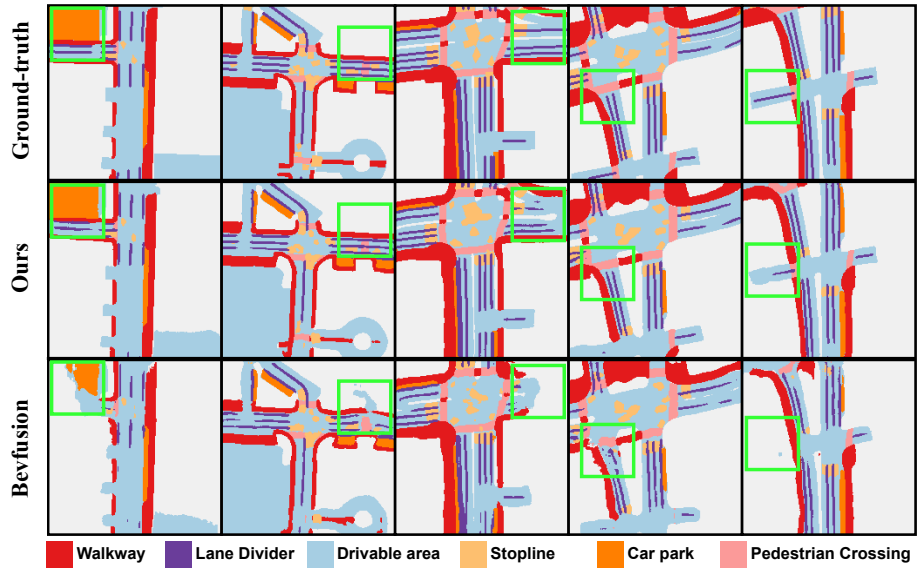


Fig. 6: Qualitative result on BEV map segmentation task on the Nuscenes *val* set, compared to BEVFusion [23]. Green bounding boxes highlight the areas where the baseline method fails to segment, while our method can segment correctly.

impressive results with an NDS of 72.9 and an mAP of 70.2, DifFUSER further enhances these metrics to 73.8 (+0.9%) and 71.2 (+1%), competing closely with CMT [47], which achieves an NDS of 74.1 and an mAP of 72.0. This improvement is attributed to the more effective multi-modal diffusion-based fusion. Specifically, DifFUSER's denoising capability contributes to finer detail preservation and noise reduction in fused features, leading to better object detection accuracy. The improvement is also evident in other metrics, with DifFUSER achieving lower error rates (mATE, mASE, mAOE, mAVE, mAAE) compared to BEVFusion [23]. This suggests that DifFUSER not only enhances the detection accuracy but also improves the overall reliability and precision of the object detection task. The same trend is observed in the validation set, as in Tab. 3.

Robustness to sensor failure. To highlight the robustness of DifFUSER under sensor failures, we present the segmentation performance comparison in Fig. 5. The findings reveal that with a sufficient number of sampling steps, the model is capable of generating features that effectively compensate for and act as replacements for the missing sensor data. This capability is a significant advancement over existing models like CMT [47], which, under similar missing sensor conditions, can only match the performance of single sensor baselines. Our DifFUSER model distinguishes itself by not just matching but exceeding the baseline performance achieved with single sensor inputs thanks to the synthetic features it generates. The ability to generate and utilise synthetic features effectively mitigates the reliance on any single sensor modality, ensuring the model's operational resilience in diverse and challenging environments.

6 Ablation Study

GSM diffusion block design. We conducted experiments about the components of the GSM module, where all models are trained with a shorter training schedule (6 epochs). As shown in Tab. 7, by incrementally incorporating Scale, Shift, and Gate operations, we observed a marked improvement in mean Intersection over Union (mIOU), with the Gate operation significantly boosting the mIOU to 62.17. This underscores the effectiveness of GSM's dynamic feature modulation, especially its capacity for strong conditioning, which selectively emphasises relevant features, enabling the model to generate meaningful outputs.

Contribution of each component. We investigate the contribution of each component in DifFUSER to understand whether the improvement comes from the enhanced cMini-BiFPN fusion architecture or the gated self-conditioned latent diffusion module. We conduct experiments on the NuScenes *val* set with the same settings as in Sec. 4, using single-task models. It can be seen that by adding BiFPN-like feature fusion module, the performance of BEV map segmentation and 3D object detection are improved by 3.8% in mIOU and 0.4% in NDS, respectively. Further integrating the GSM module with a 4-step sampling improves mIOU by an additional 2.3% and NDS by an additional 0.3%. Extending the diffusion sampling to 8 steps further raises mIOU to 70.04% and NDS to 72.6%, which is the final performance of DifFUSER.

Diffusion ODE Solvers. We perform a small experiment to compare the performance of other second-order ODE solvers besides the de-facto standard first-order DDIM solver. We choose BEV map segmentation for this experiment and report the segmentation mIOU results. We use the same settings as those in Sec. 4, and compare the performance of DDIM [37], DPM-Solver++ [26] and DEIS [52] at different sampling steps. As shown in Tab. 4, the performance of DPM-Solver++ and DEIS are comparable to DDIM, and the performance of all three methods are improved as the number of sampling steps increases. It shows that all solvers can lead to good convergence of the fused features, and the performance is not sensitive to the choice of solvers. For best performance, We choose DDIM with 8 sampling steps as the default solver in our experiments due to its simplicity and efficiency. However, in terms of efficiency, we believe 4-step sampling is already sufficient.

Inference Time. Here we measure the inference time of DifFUSER on a single NVIDIA 4090 GPU. At 1-step sampling, Diffuser can reach an average speed of 8 FPS (123.1 ms latency), the full break-down is shown in Tab. 8

Multi-task joint training. Finally, we investigate the performance of DifFUSER in multi-task joint training. We use a shorter training schedule and re-scale the losses of each task to balance the training. As shown in Tab. 5, the performance of both tasks decreases compared to single-task training, an issue commonly known as negative transfer, which we leave for future investigation.

[1] Including BEV-Pooling, a non-DL operation without extra parameters.

Table 4: Performance comparison of different sampling methods at varying steps

Sampling step	1	2	4	8
DDIM [37]	63.88	66.65	68.82	**70.04**
DPM-Solver [26]	64.02	66.20	68.90	**70.02**
DeIS [52]	63.80	66.30	68.81	**69.95**

Table 5: Multi-task joint training performance

Task	NDS	mIOU
Detection	69.59	-
Segmentation	-	65.8
Joint	68.23	61.36

Table 6: Comparison of Detection and Segmentation Performance

Component	mAP	NDS	mIOU
Baseline**	68.9	71.4	62.7
+cMiniBiFPN**	69.4	71.8	66.5
+GSM (4-step) + PSDT*	69.7	72.1	68.82
+GSM (8-step) + PSDT*	**70.0**	**72.6**	**70.04**

[*] With $\mathcal{L}_{diffusion}$, [**] Without $\mathcal{L}_{diffusion}$

Table 7: GSM diffusion block design.

Scale	Shift	Gate	mIOU
✓			59.92
✓	✓		60.73
✓	✓	✓	**62.17**

Table 8: Detailed breakdown of inference speeds, excluding the speed of point cloud voxelisation module, following our baseline.

Module	Latency	Parameters	GFLOPs	MACs
Camera	63ms[1]	27.5M	127.7	63.8
LiDAR	25ms	7.1M	32.3	16.2
Fuser	24ms	37.5M	294.0	147.0
Task heads	11ms	4.7M	189.0	94.5
Total	123ms	76.8M	643.0	321.5

7 Conclusion

In this work, we propose DifFUSER, a diffusion-based generative model for multi-modal fusion in 3D perception. DifFUSER is designed to enhance the quality of fused features by improving the fusion architecture and leveraging the denoising property of diffusion models. Central to our approach is the introduction of a Gated Self-Conditioned Modulated (GSM) latent diffusion module and PSDT paradigm, specifically engineered to improve the fusion output features and robustness against sensor noise and failure. Additionally, cMiniBiFPN fusion architecture emerged as a promising multi-scale latent diffusion module. Extensive experiments on the NuScenes dataset show that DifFUSER achieves SOTA performance in BEV map segmentation tasks and can fairly compete with SOTA transformer-based fusion methods in 3D object detection. This shows the potential of generative models to significantly enhance the capability of autonomous driving systems. Our approach opens a direction of using diffusion models for better accuracy and reliability in 3D perception tasks, promising further enhancements and broader applicability of diffusion-based models in complex, real-world environments.

Acknowledgments. This work has been partially funded by The Australian Research Council Discovery Project (ARC DP2020102427). We acknowledge the partial sponsorship of our research by DARPA Assured Neuro Symbolic Learning and Reasoning (ANSR) program, under award number FA8750-23-2-1016.

References

1. Bai, X., Hu, Z., Zhu, X., Huang, Q., Chen, Y., Fu, H., Tai, C.L.: Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In: CVPR. pp. 1090–1099 (2022)
2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR. pp. 11621–11631 (2020)
3. Chen, S., Sun, P., Song, Y., Luo, P.: Diffusiondet: Diffusion model for object detection. In: ICCV. pp. 19830–19843 (2023)
4. Chen, X., Zhang, T., Wang, Y., Wang, Y., Zhao, H.: Futr3d: A unified sensor fusion framework for 3d detection. In: CVPR. pp. 172–181 (2023)
5. Chen, Y., Yu, Z., Chen, Y., Lan, S., Anandkumar, A., Jia, J., Alvarez, J.M.: Focalformer3d: focusing on hard instance for 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8394–8405 (2023)
6. Chen, Y., Li, Y., Zhang, X., Sun, J., Jia, J.: Focal sparse convolutional networks for 3d object detection. In: CVPR. pp. 5428–5437 (2022)
7. Chen, Y., Liu, J., Zhang, X., Qi, X., Jia, J.: Largekernel3d: Scaling up kernels in 3d sparse cnns. In: CVPR. pp. 13488–13498 (2023)
8. Fan, L., Xiong, X., Wang, F., Wang, N., Zhang, Z.: Rangedet: In defense of range view for lidar-based 3d object detection. In: ICCV. pp. 2918–2927 (2021)
9. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS **33**, 6840–6851 (2020)
10. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
11. Jiang, C., Cornman, A., Park, C., Sapp, B., Zhou, Y., Anguelov, D., et al.: Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In: CVPR. pp. 9644–9653 (2023)
12. Jiao, Y., Jie, Z., Chen, S., Chen, J., Ma, L., Jiang, Y.G.: Msmdfusion: Fusing lidar and camera at multiple scales with multi-depth seeds for 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21643–21652 (2023)
13. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: CVPR. pp. 12697–12705 (2019)
14. Le, D.T., Shi, H., Rezatofighi, H., Cai, J.: Accurate and real-time 3d pedestrian detection using an efficient attentive pillar network. RA-L **8**(2), 1159–1166 (2022)
15. Li, Y., Chen, Y., Qi, X., Li, Z., Sun, J., Jia, J.: Unifying voxel-based representation with transformer for 3d object detection. NeurIPS **35**, 18442–18455 (2022)
16. Li, Y., Ge, Z., Yu, G., Yang, J., Wang, Z., Shi, Y., Sun, J., Li, Z.: Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In: AAAI. vol. 37, pp. 1477–1485 (2023)

17. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: European conference on computer vision. pp. 1–18. Springer (2022)
18. Liang, T., Xie, H., Yu, K., Xia, Z., Lin, Z., Wang, Y., Tang, T., Wang, B., Tang, Z.: Bevfusion: A simple and robust lidar-camera fusion framework. NeurIPS **35**, 10421–10434 (2022)
19. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3d: High-resolution text-to-3d content creation. In: CVPR. pp. 300–309 (2023)
20. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: ICCV. pp. 9298–9309 (2023)
21. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV. pp. 10012–10022 (2021)
22. Liu, Z., Zhao, X., Huang, T., Hu, R., Zhou, Y., Bai, X.: Tanet: Robust 3d object detection from point clouds with triple attention. In: AAAI. vol. 34, pp. 11677–11684 (2020)
23. Liu, Z., Tang, H., Amini, A., Yang, X., Mao, H., Rus, D.L., Han, S.: Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 2774–2781. IEEE (2023)
24. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
25. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
26. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. arXiv preprint arXiv:2211.01095 (2022)
27. Lu, T., Ding, X., Liu, H., Wu, G., Wang, L.: Link: Linear kernel for lidar-based 3d perception. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1105–1115 (2023)
28. Paigwar, A., Sierra-Gonzalez, D., Erkent, Ö., Laugier, C.: Frustum-pointpillars: A multi-stage approach for 3d object detection using rgb camera and lidar. In: ICCV. pp. 2926–2933 (2021)
29. Pang, S., Morris, D., Radha, H.: Clocs: Camera-lidar object candidates fusion for 3d object detection. In: IROS. pp. 10386–10393. IEEE (2020)
30. Pang, S., Morris, D., Radha, H.: Fast-clocs: Fast camera-lidar object candidates fusion for 3d object detection. In: WACV. pp. 187–196 (2022)
31. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16. pp. 194–210. Springer (2020)
32. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022)
33. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from rgb-d data. In: CVPR. pp. 918–927 (2018)
34. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR. pp. 652–660 (2017)

35. Simon, M., Amende, K., Kraus, A., Honer, J., Samann, T., Kaulbersch, H., Milz, S., Michael Gross, H.: Complexer-yolo: Real-time 3d object detection and tracking on semantic point clouds. In: CVPR Workshops. pp. 0–0 (2019)
36. Smith, L.N., Topin, N.: Super-convergence: Very fast training of neural networks using large learning rates. In: Artificial intelligence and machine learning for multi-domain operations applications. vol. 11006, pp. 369–386. SPIE (2019)
37. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
38. Sun, P., Wang, W., Chai, Y., Elsayed, G., Bewley, A., Zhang, X., Sminchisescu, C., Anguelov, D.: Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In: CVPR. pp. 5725–5734 (2021)
39. Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: CVPR. pp. 10781–10790 (2020)
40. Vora, S., Lang, A.H., Helou, B., Beijbom, O.: Pointpainting: Sequential fusion for 3d object detection. In: CVPR. pp. 4604–4612 (2020)
41. Wang, C., Ma, C., Zhu, M., Yang, X.: Pointaugmenting: Cross-modal augmentation for 3d object detection. In: CVPR. pp. 11794–11803 (2021)
42. Wang, T., Xinge, Z., Pang, J., Lin, D.: Probabilistic and geometric depth: Detecting objects in perspective. In: Conference on Robot Learning. pp. 1475–1485. PMLR (2022)
43. Wang, T., Zhu, X., Pang, J., Lin, D.: Fcos3d: Fully convolutional one-stage monocular 3d object detection. In: ICCV. pp. 913–922 (2021)
44. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on Robot Learning. pp. 180–191. PMLR (2022)
45. Xie, E., Yu, Z., Zhou, D., Philion, J., Anandkumar, A., Fidler, S., Luo, P., Alvarez, J.M.: M^2bev: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation. arXiv preprint arXiv:2204.05088 (2022)
46. Xu, S., Zhou, D., Fang, J., Yin, J., Bin, Z., Zhang, L.: Fusionpainting: Multimodal fusion with adaptive attention for 3d object detection. In: ITSC. pp. 3047–3054. IEEE (2021)
47. Yan, J., Liu, Y., Sun, J., Jia, F., Li, S., Wang, T., Zhang, X.: Cross modal transformer: Towards fast and robust 3d object detection. In: ICCV. pp. 18268–18278 (2023)
48. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
49. Yang, Z., Chen, J., Miao, Z., Li, W., Zhu, X., Zhang, L.: Deepinteraction: 3d object detection via modality interaction. NeurIPS **35**, 1992–2005 (2022)
50. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: CVPR. pp. 11784–11793 (2021)
51. Yin, T., Zhou, X., Krähenbühl, P.: Multimodal virtual point 3d detection. NeurIPS **34**, 16494–16507 (2021)
52. Zhang, Q., Chen, Y.: Fast sampling of diffusion models with exponential integrator. arXiv preprint arXiv:2204.13902 (2022)
53. Zhou, B., Krähenbühl, P.: Cross-view transformers for real-time map-view semantic segmentation. In: CVPR. pp. 13760–13769 (2022)
54. Zhou, X., Hou, J., Yao, T., Liang, D., Liu, Z., Zou, Z., Ye, X., Cheng, J., Bai, X.: Diffusion-based 3d object detection with random boxes. arXiv preprint arXiv:2309.02049 (2023)
55. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: CVPR. pp. 4490–4499 (2018)

56. Zou, J., Zhu, Z., Ye, Y., Wang, X.: Diffbev: Conditional diffusion model for bird's eye view perception. arXiv preprint arXiv:2303.08333 (2023)

Diffusion Model for Robust Multi-Sensor Fusion in 3D Object Detection and BEV Segmentation

Supplementary Material

8 Additional implementation details

8.1 Formulation of Diffusion Models

Here, we review the formulation of diffusion models used in DiFFUSER, which based on the frameworks established in [9, 10, 37]. Beginning with an original latent features $\boldsymbol{x}_0^F \sim q(\boldsymbol{x}_0^F)$, we apply a sequential noising mechanism q, which incrementally infuses Gaussian noise into latent features $\boldsymbol{x}_1^F$, $\boldsymbol{x}_2^F$, ..., $\boldsymbol{x}_T^F$ at each time t. Specifically, the noise infusion at each stage is determined by a variance schedule $\beta_t \in (0, 1)$:

$$q(\boldsymbol{x}_{1:T}^F|\boldsymbol{x}_0^F) = \prod_{t=1}^{T} q(\boldsymbol{x}_t^F|\boldsymbol{x}_{t-1}^F) \tag{9}$$

$$q(\boldsymbol{x}_t^F|\boldsymbol{x}_{t-1}^F) = \mathcal{N}(\boldsymbol{x}_t^F; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}^F, \beta_t \boldsymbol{I}) \tag{10}$$

Following DDPM [9], instead of iteratively apply q, it is feasible to sample $\boldsymbol{x}_t^F$ at any chosen t directly:

$$q(\boldsymbol{x}_t^F|\boldsymbol{x}_0^F) = \mathcal{N}(\boldsymbol{x}_t^F; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0^F, (1-\bar{\alpha}_t)\boldsymbol{I}) \tag{11}$$

$$= \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0^F + \epsilon\sqrt{1-\bar{\alpha}_t}, \epsilon \sim \mathcal{N}(0, \boldsymbol{I}) \tag{12}$$

where $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$ and $\alpha_t = 1 - \beta_t$. In this context, $\bar{\alpha}_t$ is employed for scheduling the noise. According to Bayes' theorem, the posterior $q(\boldsymbol{x}_{t-1}^F|\boldsymbol{x}_t^F, \boldsymbol{x}_0^F)$ is also Gaussian:

$$q(\boldsymbol{x}_{t-1}^F|\boldsymbol{x}_t^F, \boldsymbol{x}_0^F) = \mathcal{N}(\boldsymbol{x}_{t-1}^F; \mu_t'(\boldsymbol{x}_t^F, \boldsymbol{x}_0^F), \beta_t'\mathbf{I}) \tag{13}$$

where the mean and variance, given by

$$\mu_t'(\boldsymbol{x}_t^F, \boldsymbol{x}_0^F) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_0^F + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t^F \tag{14}$$

and

$$\beta_t' = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{15}$$

respectively, characterize this Gaussian distribution.

To generate a sample from $q(\boldsymbol{x}_0^F)$, we may sample from $q(\boldsymbol{x}_T^F)$ and iteratively apply the reverse transitions $q(\boldsymbol{x}_{t-1}^F|\boldsymbol{x}_t^F)$ back to $\boldsymbol{x}_0^F$. Given a sufficiently extensive T and a suitably chosen β_t series ($\beta_t \rightarrow 0$), the distribution $q(\boldsymbol{x}_T^F)$ converges closely to an isotropic Gaussian. Specifically, our conditional diffusion model is parametrized through $p_\theta(x_{t-1}^F|x_t^F, t, \tilde{x}_0^F)$. Because $p_\theta(x_{t-1}^F|x_t^F, t, \tilde{x}_0^F)$ is not be easily estimated but could be approximated using a neural network (*i.e.* DifFUSER), where the aim is to predict μ_θ and Σ_θ:

$$p_\theta(\boldsymbol{x}_{t-1}^F|x_t^F, t, \tilde{x}_0^F) \coloneqq \mathcal{N}(\boldsymbol{x}_{t-1}^F; \mu_\theta(x_t^F, t, \tilde{x}_0^F), \Sigma_\theta(x_t^F, t, \tilde{x}_0^F)) \tag{16}$$

Here, a neural network can also be optimized to predict either ϵ or $\boldsymbol{x}_0^F$ [10, 37]. We opt for predicting $\boldsymbol{x}_0^F$ in this paper.

8.2 Pseudo-code of DifFUSER training and sampling

Algorithm 1 DifFUSER Training

```
function TRAIN_LOSS(images, pointcloud, gt_boxes, gt_map)
   """
   Input:
      images: [B, V, H, W, 3] # V: number of multi-view images
      pointcloud: [B, N, 3] # N: number of points in pointcloud
      gt_boxes: [B, *, D] # D: number of box dimensions
      gt_map: [B, C, 200, 200] # C: number of segmentation classes
   Output: total loss which is a measure of the prediction quality
   """
   # Encode the image and LiDAR pointcloud features
   cam_feats ← image_encoder(images)
   lidar_feats ← lidar_encoder(pointcloud)

   # Concatenate extracted features forming initial latent
   # representation and set as diffusion target
   x_0^F ← concatenate(cam_feats, lidar_feats)
   diffusion_target ← x_0^F

   # Perturb the initial representation using PDST
   x̃_0^F ← PDST(x_0^F)
   # Randomly select a time step in the diffusion process
   t ← randint(0, T)
   noise ← normal(mean=0, std=1) # Generate noise
   # Apply the noise to the latent representation at time step t
   x_t^F ← sqrt(alpha_cumprod(t) * x_0^F + sqrt(1-alpha_cumprod(t)) * noise
   # Predict the denoised features at time t using the DifFUSER model
   x̂_0^F ← DifFUSER(x̃_0^F, t)
   # Pass the enhanced features through the task-specific head
   pred ← task_head(x̂_0^F)
   # Compute the total loss
   loss ← diffusion_loss(diffusion_target, x̂_0^F)
          + task_loss(pred, gt_boxes, gt_map)
   return loss
end function
```

Algorithm 2 DifFUSER Sampling

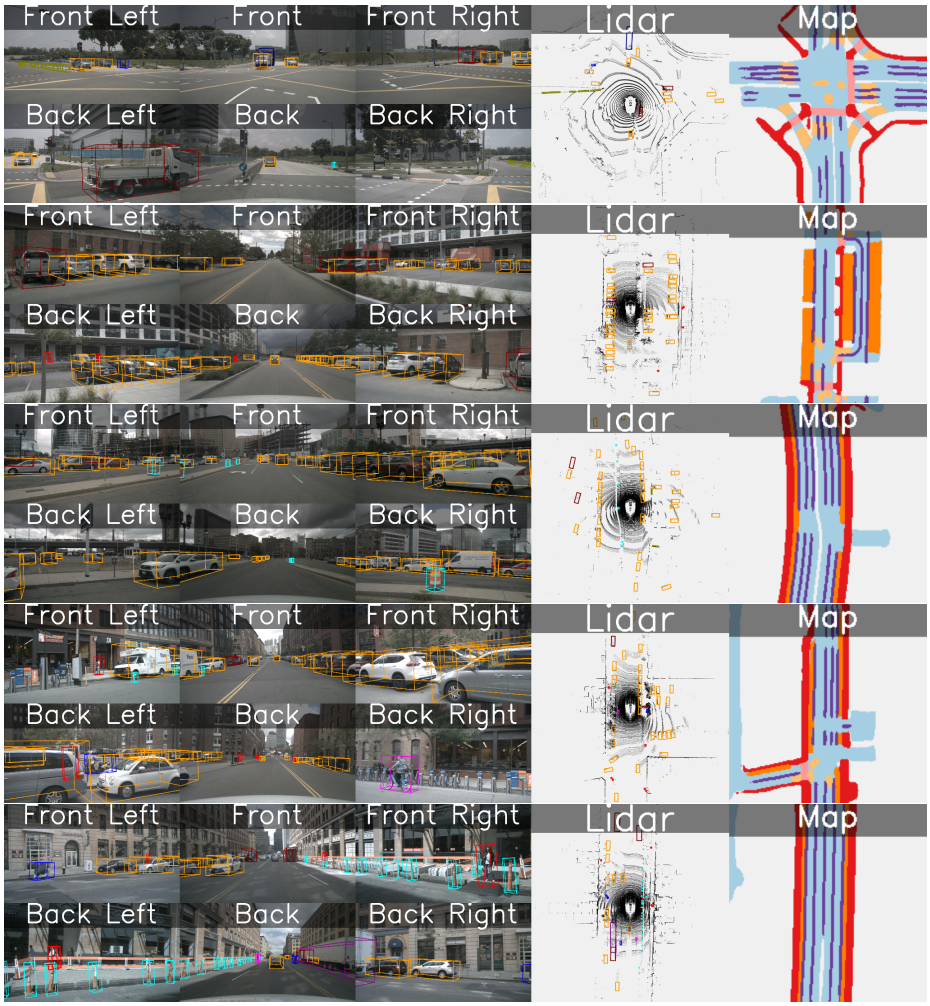
```
function TRAIN_LOSS(images, pointcloud, steps, T)
   """
   Input:
      images: [B, V, H, W, 3] # V: number of multi-view images
      pointcloud: [B, N, 3] # N: number of points in pointcloud
      steps: number of sampling steps
      T: number of time steps
   Output: task-specific predictions
   """
   # Encode the image and LiDAR pointcloud features
   cam_feats ← image_encoder(images)
   lidar_feats ← lidar_encoder(pointcloud)

   # Concatenate extracted features forming initial latent
   # representation and use as diffusion condition
   x_0^F ← concatenate(cam_feats, lidar_feats)

   # Sampling step sizes [(T-1, T-2), (T-2, T-3), ..., (1, 0)]
   times ← reverse(linspace(-1, T - 1, steps + 1))
   time_pairs ← list(zip(times[:-1], times[1:]))

   x_t^F ← normal(mean=0, std=1) # Sample x_t^F from noise
   for (t_now, t_next) in time_pairs:
      x̂_0^F ← DifFUSER(x_0^F, x_t^F, t_now)
   # estimate x_t^F at t_next, sampler can be [37], [52], or [26]
      x_t^F ← sampler_step(x̂_0^F, t_now, t_next)
   # Pass the predicted enhanced features through the task-specific
head
   pred ← task_head(x̂_0^F)
   return pred
end function
```

9 Additional Qualitative Results



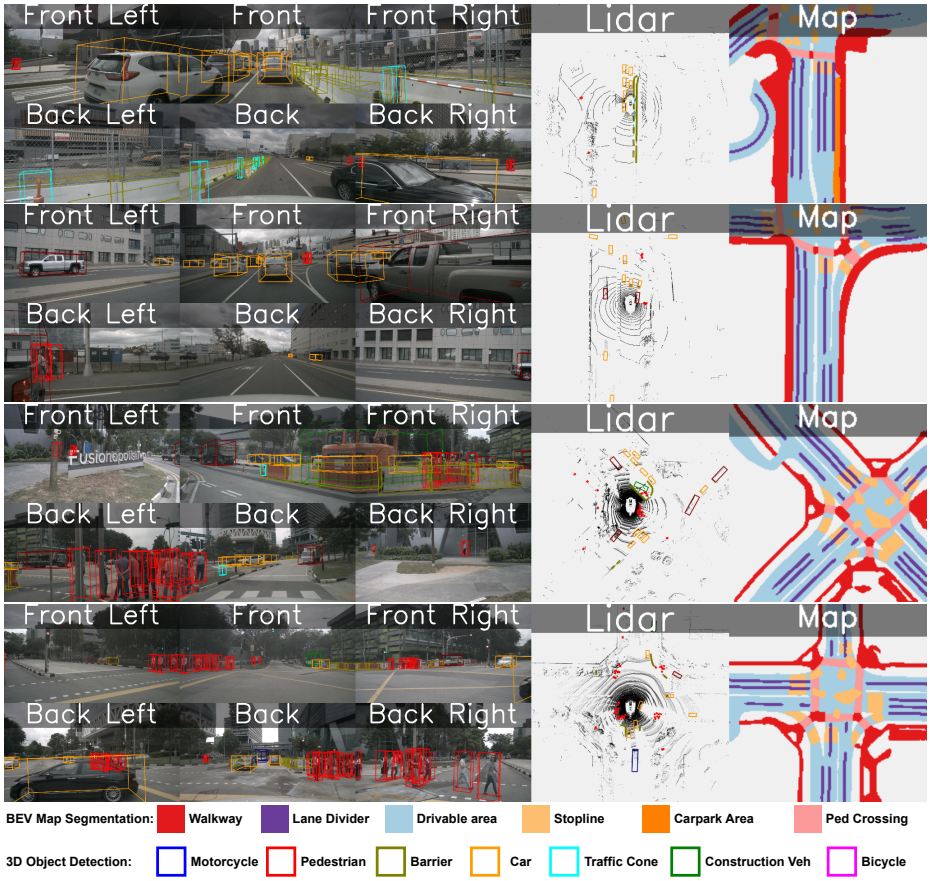


Fig. 7: We provide additional qualitative results for challenging scenarios from the NuScenes *val* set. On the left, we show the projections of predicted 3D bounding boxes on the camera images. On the right, we show the 3D prediction (in BEV) and BEV map segmentation results. Best viewed in color and zoomed in.