2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

MSDNet: Efficient 4D Radar Super-Resolution via Multi-Stage Distillation

Minqing Huang*, Shouyi Lu*, Boyuan Zheng, Ziyao Li, Xiao Tang, Guirong Zhuo†

***Abstract*— 4D radar super-resolution, which aims to reconstruct sparse and noisy point clouds into dense and geometrically consistent representations, is a foundational problem in autonomous perception. However, existing methods often suffer from high training cost or rely on complex diffusion-based sampling, resulting in high inference latency and poor generalization, making it difficult to balance accuracy and efficiency. To address these limitations, we propose MSDNet, a multi-stage distillation framework that efficiently transfers dense LiDAR priors to 4D radar features to achieve both high reconstruction quality and computational efficiency. The first stage performs reconstruction-guided feature distillation (RGFD), aligning and densifying the student's features through feature reconstruction. In the second stage, we propose diffusion-guided feature distillation (DGFD), which treats the stage-one distilled features as a noisy version of the teacher's representations and refines them via a lightweight diffusion network. Furthermore, we introduce a noise adapter that adaptively aligns the noise level of the feature with a predefined diffusion timestep, enabling a more precise denoising. Extensive experiments on the VoD and in-house datasets demonstrate that MSDNet achieves both high-fidelity reconstruction and low-latency inference in the task of 4D radar point cloud super-resolution, and consistently improves performance on downstream tasks.**

I. INTRODUCTION

4D millimeter-wave (mmWave) radar, hereafter referred to as *4D radar*, provides 3D point clouds together with instantaneous velocity measurements. It has become a key sensor in autonomous driving, and has attracted substantial attention in both academia and industry [1]–[6]. Due to its longer operating wavelength, 4D radar is largely insensitive to fine particulates (e.g., fog, drizzle), enabling stable perception in adverse weather conditions. Such robustness makes it well suited for all-weather autonomous driving.

Despite these advantages, 4D radar data remain spatially sparse, noisy, and non-uniformly sampled compared to high-resolution LiDAR point clouds, severely limiting their effectiveness in fine-grained perception tasks such as object detection, scene understanding, and localization [7]. To address these issues, as shown in Fig.1(a), several studies process 4D radar signals directly, replacing traditional direction-of-arrival (DoA) estimation [8], [9] and constant false-alarm rate (CFAR) pipelines with deep neural networks to enhance point cloud quality [10], [11]; however, these approaches typically incur substantial data collection costs and exhibit limited transferability.

This work was supported by the National Natural Science Foundation of China under Grant 52325212. *The two authors contributed equally to this work. † indicates the corresponding author: Guirong Zhuo (zhuoguirong@tongji.edu.cn). The Authors are with School of Automotive Studies, Tongji University, Shanghai 201804, China. Code will be released at: https://github.com/potatochip1211/MSDNet-4D-Radar.

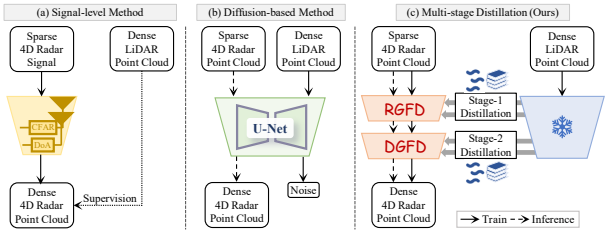

Fig. 1. Comparison of 4D Radar Super-Resolution Paradigms. (a) Signal-level methods replace classical signal-processing pipelines with deep neural networks. (b) Diffusion-based methods are trained on LiDAR point clouds and, during inference, iteratively denoise sparse 4D radar point clouds to produce dense point clouds. (c) Our MSDNet introduces a multi-stage distillation strategy that transfers LiDAR geometric priors to 4D radar and generates dense point clouds.

In recent years, diffusion models have achieved notable success in generative AI tasks [12], [13]. Building on these advances, recent work has further explored diffusion-based super-resolution for 4D radar point clouds. As illustrated in Fig.1(b), these methods [14], [15] typically employ a U-Net backbone within the diffusion process. During training, these methods apply forward diffusion to LiDAR point clouds to train a noise prediction model. At inference, dense point clouds are generated from sparse 4D radar inputs via iterative reverse denoising. However, the significant distribution and density gap between 4D radar and LiDAR often requires complex diffusion architectures and multi-step sampling schedules to achieve competitive reconstruction quality, resulting in high inference latency.

In this paper, we propose *Multi-Stage Distillation Net* (MSDNet), a knowledge distillation framework for 4D radar super-resolution. As illustrated in Fig.1(c), MSDNet transfers knowledge from high-resolution LiDAR to 4D radar representations in two progressive stages. **Stage 1** employs reconstruction-guided feature distillation to transfer dense LiDAR features through feature reconstruction. **Stage 2** treats the Stage-1 distilled features as a "noisy version" of the dense LiDAR features and applies a lightweight diffusion model equipped with prior modeling and iterative denoising to refine student representations. To properly initialize the diffusion process, we introduce a noise adapter module that dynamically estimates the noise level for each student feature and injects the corresponding Gaussian noise to match the appropriate diffusion timestep. Benefiting from the knowledge transferred in Stage 1, Stage 2 requires only a lightweight diffusion network, thereby significantly improving super-resolution quality while maintaining inference efficiency.

Overall, the contributions of our paper are as follows:

- We introduce *Multi-Stage Distillation Net* (MSDNet), a knowledge-distillation framework for 4D radar super-

resolution that efficiently transfers knowledge from dense LiDAR features to sparse 4D radar representations. To our knowledge, this is the first application of knowledge distillation to 4D radar super-resolution.
- We design reconstruction-guided feature distillation for the initial transfer of dense LiDAR features, and introduce diffusion-guided feature distillation to refine the first-stage distilled features. In addition, we incorporate a noise adapter to precisely align the student features' noise level with the initial timestep.
- We conduct a comprehensive evaluation of MSDNet on the VoD [16] and in-house datasets. Results show that MSDNet outperforms existing methods across reconstruction metrics while maintaining efficient inference. Moreover, MSDNet-reconstructed point clouds deliver significant gains on downstream tasks.

II. RELATED WORK

A. Knowledge Distillation for 4D Radar

Knowledge distillation (KD) [17] provides a unified paradigm for cross-modal and cross-cost sensor transfer, enabling dense geometric and structural priors from LiDAR and cameras to be injected into 4D radar. Recent LiDAR-to-4D-Radar KD methods [18]–[21] have achieved notable gains in object detection, strengthening the discriminative power of sparse 4D radar point clouds. However, these frameworks are typically tightly coupled to detection heads and task-specific losses, directing optimization toward detection accuracy rather than point cloud resolution. Consequently, they are ill-suited for direct application to 4D point cloud super-resolution. Motivated by this gap, we adopt a reconstruction-centric perspective and develop a multi-stage distillation framework that progressively aligns and constrains representations across feature hierarchies to systematically improve 4D radar point cloud quality.

B. 4D Radar Point Cloud Super-Resolution

Research on 4D radar point cloud super-resolution generally falls into two categories: (i) *signal-chain* super-resolution, which replaces or augments traditional array processing (e.g., DoA, CFAR) during signal formation; and (ii) *post-processing* super-resolution, which directly enhances the resolution and spatial regularity of 4D radar point clouds. On the signal-chain side, Zhao et al. [9] perform end-to-end angular-occupancy regression from the upper-triangular portion of the array-output covariance matrix. Jiang et al. [8] achieve single-frame super-resolution DoA by combining a dual pulse-repetition-frequency waveform with a complex-valued convolutional network within a time-division– and Doppler-division–multiplexed MIMO framework. Denser-radar trains directly on raw 4D radar tensors using dense occupancies synthesized from multi-frame LiDAR for supervision, while Franceschi et al. [10] replace the conventional CFAR front end with a CNN-based alternative. Despite improving 4D radar imaging, these approaches generally suffer from data scarcity and high training costs. On the post-processing side, building on advances in generative modeling, diffusion methods [22], [23] have also been adopted for 4D radar super-resolution. Luan et al. [14] project 4D radar point clouds into BEV images and enhance them via BEV-based reconstruction. R2LDM [15] introduces a voxel-feature diffusion model that further improves point cloud detail. However, diffusion-based methods typically rely on U-Net backbones [24] and multi-step sampling, resulting in high inference latency. To overcome these bottlenecks, we propose a knowledge-distillation-based post-processing super-resolution framework for 4D radar. Operating under a teacher–student paradigm, it performs multi-stage feature distillation and employs a lightweight diffusion network to streamline the inference pipeline, substantially reducing latency and computational overhead while preserving reconstruction fidelity.

III. PROPOSED METHOD

This section details MSDNet, with its overall architecture illustrated in Fig. 2. Section III-A provides a brief overview of the feature extraction and point cloud reconstruction networks employed as both teacher and student models. Section III-B describes the construction of the teacher network. Section III-C introduces the reconstruction-guided feature distillation (RGFD), which carries out the first-stage knowledge transfer through feature reconstruction. Finally, Section III-D presents the diffusion-guided feature distillation (DGFD), which leverages a lightweight diffusion denoiser to further refine the student's features.

A. Preliminary

Feature Encoding. We employ VoxelNet [25] as a common feature encoder for both the LiDAR and 4D radar point clouds. Initially, the two types of point clouds are voxelized into a set of 3D voxels, $\mathbf{V}^{\mathrm{3D}}_{\mathrm{mod}}$. Subsequently, each branch is processed by a dedicated Voxel Feature Encoding and a sparse 3D convolutional backbone (collectively, SparseEnc$(\cdot)$). The resulting features are then aggregated along the height dimension to generate BEV representations, yielding two BEV features: $\mathbf{F}^{\mathrm{S}}_{\mathrm{l}}$ and $\mathbf{F}^{\mathrm{S}}_{\mathrm{r}}$:

$$\mathbf{F}^{\mathrm{S}}_{\mathrm{mod}} = \Pi_z\left(\mathrm{SparseEnc}\left(\mathbf{V}^{\mathrm{3D}}_{\mathrm{mod}}\right)\right), \mathrm{mod} \in \{\mathrm{l}, \mathrm{r}\}, \quad (1)$$

where $\Pi_z(\cdot)$ denotes the aggregation along the height dimension, the subscripts l and r refer to the LiDAR and 4D radar branches. $\mathbf{F}^{\mathrm{D}}_{\mathrm{l}}$ and $\mathbf{F}^{\mathrm{D}}_{\mathrm{r}}$ are then obtained via feature enhancement and multi-stage feature distillation, respectively.

Point Cloud Reconstruction. We adopt a progressive point cloud reconstruction module [15] to map dense BEV features to a 3D point cloud. Given the modal features $\mathbf{F}^{\mathrm{D}}_{\mathrm{mod}}$, we apply 3D upsampling layers to generate multi-scale voxel features at scales $s \in \{1/4, 1/2, 1\}$:

$$\mathbf{G}^{(s)}_{\mathrm{mod}} = \mathrm{Up3D}_s(\mathbf{F}^{\mathrm{D}}_{\mathrm{mod}}). \quad (2)$$

At each scale, a dual-branch prediction head is employed to output the voxel occupancy and point offset, respectively:

$$\mathbf{M}^{(s)} = \sigma\left(\phi^{(s)}_{\mathrm{mask}}\left(\mathbf{G}^{(s)}_{\mathrm{mod}}\right)\right) \in [0, 1]. \quad (3)$$

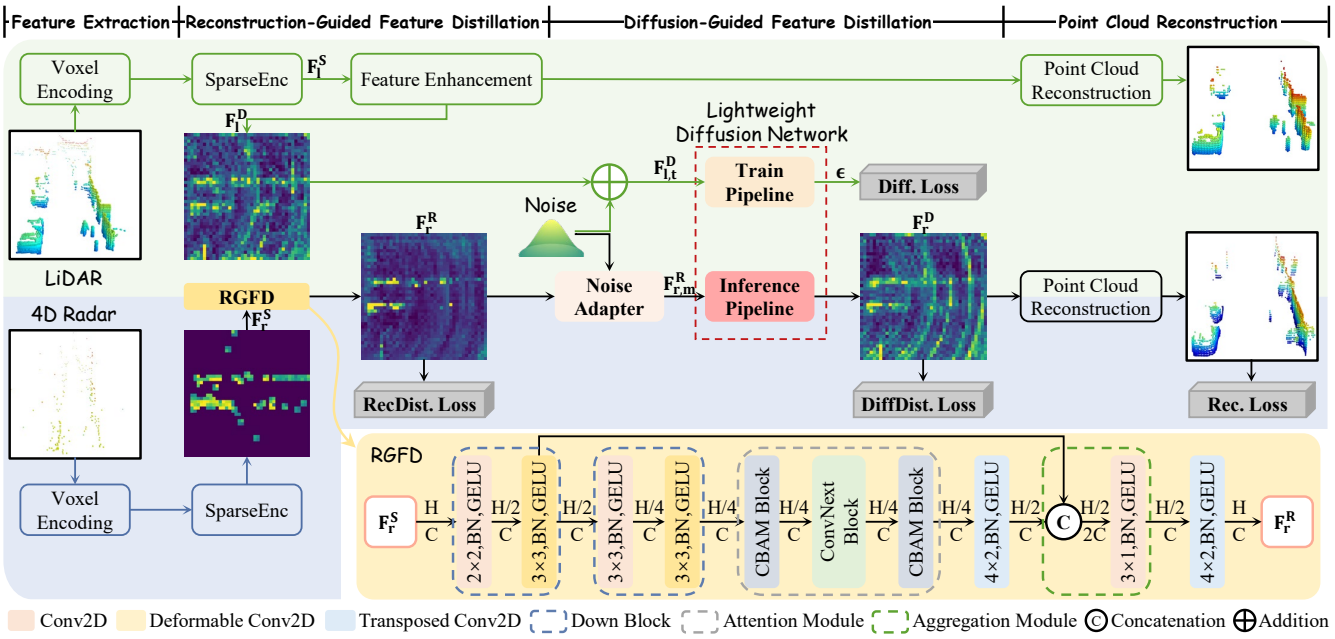

Fig. 2. Overall architecture of MSDNet. LiDAR and 4D radar point clouds are encoded by Voxel Encoding and SparseEnc to obtain BEV features, $\mathbf{F}_l^S$ and $\mathbf{F}_r^S$. The teacher branch applies feature enhancement to produce dense LiDAR features $\mathbf{F}_l^D$ and to reconstruct the LiDAR point cloud. The student branch first performs feature reconstruction on $\mathbf{F}_r^S$ within the RGFD module and conducts reconstruction distillation. Then, the DGFD module uses a lightweight diffusion model to refine the stage-one distilled features, producing the denoised representation $\mathbf{F}_r^D$. This representation is used for diffusion distillation and to reconstruct a dense 4D radar point cloud. The bottom right corner details the architecture of the RGFD module.

The offset branch predicts the point displacement relative to the center of its corresponding voxel:

$$\Delta\mathbf{P}^{(s)} = \tanh\left(\phi_{\text{off}}^{(s)}\left(\mathbf{G}_{\text{mod}}^{(s)}\right)\right)\cdot\frac{\mathrm{L}^{(s)}}{2} \in \left(-\frac{\mathrm{L}^{(s)}}{2}, \frac{\mathrm{L}^{(s)}}{2}\right)^3, \quad (4)$$

where $\sigma(\cdot)$ and $\tanh(\cdot)$ are the Sigmoid and Tanh activation functions, respectively. $\phi_{\text{mask}}^{(s)}(\cdot)$ and $\phi_{\text{off}}^{(s)}(\cdot)$ are the 3D convolutional branches for the occupancy and offset predictions. $\mathrm{L}^{(s)}$ is the side length of voxels at scale s. During inference, the point cloud is generated only at the full resolution ($s = 1$). Activated voxels are identified by applying a threshold to the occupancy map $\mathbf{M}^{(1)}$. The final point coordinates $\mathbf{P}^{(1)}$ are then computed by adding the predicted offsets $\Delta\mathbf{P}^{(1)}$ to the centers of these activated voxels $\mathbf{C}^{(1)}$:$\mathbf{P}^{(1)} = \mathbf{C}^{(1)} + \Delta\mathbf{P}^{(1)}$.

Notably, in our KD framework, the LiDAR branch serves as the teacher model, while the 4D radar branch acts as the student. The two branches employ distinct weights for feature extraction but share a common set of weights within the point cloud reconstruction network.

B. Teacher Network

Feature Enhancement. Reconstructing point clouds directly from the sparse voxel features encoded by VoxelNet [25] struggles to recover fine-grained details. To address this, we introduce the S2D module from Sparse2Dense [26] on the teacher side, which acts as a LiDAR feature enhancer. This module refines the LiDAR features, $\mathbf{F}_{\text{l}}^{\text{S}}$, yielding dense teacher features, $\mathbf{F}_{\text{l}}^{\text{D}}$, that are richer in geometric and semantic information.

Teacher Loss. The teacher network is trained with the objective of high-fidelity point cloud reconstruction. Given LiDAR point clouds as input, the network performs feature encoding and enhancement, followed by point cloud reconstruction, to generate multi-scale point clouds. The supervision signal comprises two terms—voxel occupancy and offset.

Let N denote the total number of voxels. For the i-th voxel, let $\mathbf{M}_i \in [0,1]$ denote the predicted occupancy and $\hat{\mathbf{M}}_i \in \{0,1\}$ the ground-truth occupancy. The occupancy loss $\mathcal{L}_{\text{occ}}$ is defined as:

$$\mathcal{L}_{\text{occ}} = -\frac{1}{N}\sum_{i=1}^{N}\left(\hat{\mathbf{M}}_i \log(\mathbf{M}_i) + \left(1-\hat{\mathbf{M}}_i\right)\log(1-\mathbf{M}_i)\right). \quad (5)$$

The offset loss $\mathcal{L}_{\text{off}}$ is calculated only over the $N_{\text{o}} = \sum_i \hat{\mathbf{M}}_i$ occupied voxels. For each occupied voxel i with center $\mathbf{C}_i \in \mathbb{R}^3$, predicted displacement $\Delta\mathbf{P}_i \in \mathbb{R}^3$, and ground-truth point coordinate $\hat{\mathbf{P}}_i \in \mathbb{R}^3$, $\mathcal{L}_{\text{off}}$ is defined as:

$$\mathcal{L}_{\text{off}} = \frac{1}{N_{\text{o}}}\sum_{i=1}^{N_{\text{o}}}\left\|(\Delta\mathbf{P}_i + \mathbf{C}_i) - \hat{\mathbf{P}}_i\right\|_1, \quad (6)$$

where $\|\cdot\|_1$ denotes the L1 norm.

The total loss, $\mathcal{L}_{\text{teacher}}$, is calculated as a weighted sum over $S = 3$ scales:

$$\mathcal{L}_{\text{teacher}} = \sum_{s=1}^{S}\rho^{(s)}\mathcal{L}_{\text{occ}}^{(s)} + \sum_{s=1}^{S}\zeta^{(s)}\mathcal{L}_{\text{off}}^{(s)}, \quad (7)$$

where $\rho^{(s)}$ and $\zeta^{(s)}$ are the weights for the occupancy loss and the offset loss at scale s, respectively.

C. Reconstruction-Guided Feature Distillation

The objective of Reconstruction-Guided Feature Distillation (RGFD) is to effectively transfer geometric priors from LiDAR to 4D radar through feature reconstruction. We observe that after VoxelNet [25] encoding, the number of non-empty grid cells in the 4D radar's BEV features is only about 10% of that in the dense LiDAR features. This significant density disparity makes cross-modal alignment challenging. To this end, we propose a feature-reconstruction network that converts sparse 4D radar features into dense representations, thereby facilitating first-stage knowledge transfer.

Network Architecture. As shown in Fig. 2, the RGFD network consists of two Down Blocks, two Up Blocks, an Attention Module, and an Aggregation Module. The Down Block combines standard and deformable convolutions to perform downsampling. This design enlarges the receptive field and alleviates feature misalignment induced by local deformations, while preserving high-response regions and filling gaps in the BEV grid. The Up Block progressively restores feature resolution using 2D transposed convolutions. The Attention Module consists of two CBAM modules [27] that apply both channel and spatial attention to enhance salient features and suppress noise. A ConvNeXt residual block [28] is inserted between these two modules to further improve semantic aggregation and boundary refinement. The Aggregation Module concatenates two input features and applies a 3×1 convolution layer. After applying the Down Block, Attention Module, and Up Block, in addition to incorporating a side pathway through the Aggregation Module, RGFD generates the reconstructed 4D radar features, $\mathbf{F}_{\mathrm{r}}^{\mathrm{R}}$.

Reconstruction Distillation Loss. To inject the dense geometric priors from the LiDAR into the 4D radar, the RGFD module is trained by minimizing the feature discrepancy between the reconstructed 4D radar features, $\mathbf{F}_{\mathrm{r}}^{\mathrm{R}}$, and the dense LiDAR features, $\mathbf{F}_{\mathrm{l}}^{\mathrm{D}}$. Let Ω_{ne} and Ω_{e} denote the sets of non-empty and empty grid cells in $\mathbf{F}_{\mathrm{l}}^{\mathrm{D}}$, with N_{ne} and N_{e} being the corresponding number of cells. The reconstruction distillation loss, $\mathcal{L}_{\mathrm{distill}}^{\mathrm{rec}}$, is defined as:

$$\mathcal{L}_{\mathrm{distill}}^{\mathrm{rec}} = \alpha\, \mathbb{E}_{i\in\Omega_{\mathrm{ne}}}\big[\|\Delta_{\mathrm{R}}^{\mathbf{F}(i)}\|_2\big] + \gamma\, \mathbb{E}_{i\in\Omega_{\mathrm{e}}}\big[\|\Delta_{\mathrm{R}}^{\mathbf{F}(i)}\|_2\big], \quad (8)$$

where $\Delta_{\mathrm{R}}^{\mathbf{F}(i)} = \mathbf{F}_{\mathrm{l}}^{\mathrm{D}}(i) - \mathbf{F}_{\mathrm{r}}^{\mathrm{R}}(i)$, $\|\cdot\|_2$ denotes the L2 norm, α and γ are the weights for the non-empty and empty feature terms, respectively.

D. Diffusion-Guided Feature Distillation

Although first-stage distillation substantially reduces cross-modal discrepancies, residual remain between student and teacher representations because of capacity differences [29]. We treat these residuals as noise in distilled features and therefore treat the reconstructed features $\mathbf{F}_{\mathrm{r}}^{\mathrm{R}}$ as a "noisy version" of the dense LiDAR features $\mathbf{F}_{\mathrm{l}}^{\mathrm{D}}$. Building on this premise, we propose Diffusion-Guided Feature Distillation (DGFD), which learns a diffusion denoising prior from dense LiDAR features and applies it to the reconstructed features, producing a denoised representation $\mathbf{F}_{\mathrm{r}}^{\mathrm{D}}$ with finer-grained details.

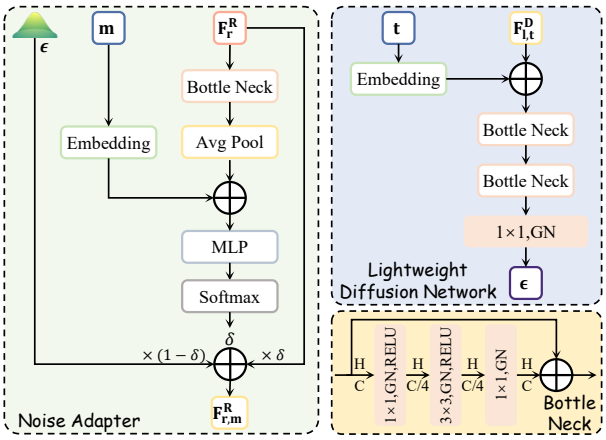

Fig. 3. Details of the Noise Adapter and Lightweight Diffusion Network.

Teacher Side: Learning the Denoising Prior. To learn the denoising prior, we subject the dense LiDAR features, $\mathbf{F}_{\mathrm{l}}^{\mathrm{D}}$, to a standard forward noising chain. The Markovian noising process is defined as:

$$q(\mathbf{F}_{\mathrm{l},t}^{\mathrm{D}}|\mathbf{F}_{\mathrm{l}}^{\mathrm{D}}) = \mathcal{N}\left(\mathbf{F}_{\mathrm{l},t}^{\mathrm{D}}|\sqrt{\bar{\alpha}_t}\mathbf{F}_{\mathrm{l}}^{\mathrm{D}}, (1-\bar{\alpha}_t)\,\mathbf{I}\right), \quad (9)$$

where $t \in \{0, 1, \ldots, T\}$, $\bar{\alpha}_t = \Pi_{s=1}^{t}(1-\beta_s)$, $\{\beta_s\}_{\mathrm{s=1}}^{T}$ is a noise variance schedule, and $\mathbf{I}$ denotes the identity matrix. The diffusion network, $\Phi_\theta(\cdot, t)$, takes the noisy features $\mathbf{F}_{\mathrm{l},t}^{D}$ and the timestep t as input to predict the noise term $\epsilon = \Phi_\theta(\mathbf{F}_{\mathrm{l},t}^{\mathrm{D}}, t)$. The denoising prior is learned by minimizing the following L2 loss:

$$\mathcal{L}_{\mathrm{diff}} = \mathbb{E}_{t,\hat{\epsilon}_t}\left\|\hat{\epsilon}_t - \Phi_\theta\left(\mathbf{F}_{\mathrm{l},t}^{\mathrm{D}}, t\right)\right\|_2^2, \quad (10)$$

where $\hat{\epsilon}_t \sim \mathcal{N}(0, \mathbf{I})$. This training process enables the diffusion network to accurately predict noise at each timestep, thereby establishing the foundation for the efficient denoising of the student's features.

Student Side: Noise Adaptation and Denoising. We treat the reconstructed features $\mathbf{F}_{\mathrm{r}}^{\mathrm{R}}$ as a "noisy version" of the dense LiDAR features. However, the noise level within $\mathbf{F}_{\mathrm{r}}^{\mathrm{R}}$ is unknown and varies across samples, making it impossible to determine the initial timestep for the diffusion process directly. To resolve this, we propose a noise adapter module that aligns the noise level of the reconstructed features with that of a predefined timestep m. As shown in Fig. 3, we first encode the timestep m into an embedding feature. Concurrently, the reconstructed features $\mathbf{F}_{\mathrm{r}}^{\mathrm{R}}$ are passed through a bottleneck block and a Global Average Pooling (GAP) layer to extract a global semantic feature, $\mathbf{F}_{\mathrm{r,g}}^{\mathrm{R}}$:

$$\mathbf{F}_{\mathrm{r,g}}^{\mathrm{R}} = \mathrm{GAP}\left(\mathrm{BottleNeck}\left(\mathbf{F}_{\mathrm{r}}^{\mathrm{R}}\right)\right). \quad (11)$$

Subsequently, the time embedding and the global semantic feature $\mathbf{F}_{\mathrm{r,g}}^{\mathrm{R}}$ are summed and fed into an MLP followed by a softmax to predict a gating coefficient, δ. This coefficient is then used to linearly mix the reconstructed features with Gaussian noise $\hat{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$, yielding a noisy initialization feature aligned with the predefined timestep m:

$$\mathbf{F}_{\mathrm{r},m}^{\mathrm{R}} = \delta \cdot \mathbf{F}_{\mathrm{r}}^{\mathrm{R}} + (1-\delta) \cdot \hat{\epsilon}. \quad (12)$$

This feature is then fed into the diffusion network to perform iterative denoising:

$$p_\theta\left(\mathbf{F}^{\mathrm{R}}_{\mathrm{r},t-n}|\mathbf{F}^{\mathrm{R}}_{\mathrm{r},t}\right)=\mathcal{N}\left(\mathbf{F}^{\mathrm{R}}_{\mathrm{r},t-n}|\Phi_\theta\left(\mathbf{F}^{\mathrm{R}}_{\mathrm{r},t},t\right),\sigma_t^2\mathbf{I}\right), \quad (13)$$

where $t\in\{m,m-n,\ldots,n\}$, n represents the sampling interval. After the denoising process is complete, we obtain the denoised reconstructed features, $\mathbf{F}^{\mathrm{D}}_{\mathrm{r}}$. These are then used along with the dense LiDAR features, $\mathbf{F}^{\mathrm{D}}_{\mathrm{l}}$, to calculate the diffusion distillation loss, $\mathcal{L}^{\mathrm{diff}}_{\mathrm{distill}}$:

$$\mathcal{L}^{\mathrm{diff}}_{\mathrm{distill}}=\alpha\,\mathbb{E}_{i\in\Omega_{\mathrm{ne}}}\left[\|\Delta_{\mathrm{D}}^{\mathbf{F}(i)}\|_2\right]+\gamma\,\mathbb{E}_{i\in\Omega_{\mathrm{e}}}\left[\|\Delta_{\mathrm{D}}^{\mathbf{F}(i)}\|_2\right], \quad (14)$$

where $\Delta_{\mathrm{D}}^{\mathbf{F}(i)}=\mathbf{F}^{\mathrm{D}}_{\mathrm{l}}(i)-\mathbf{F}^{\mathrm{D}}_{\mathrm{r}}(i)$.

Lightweight Diffusion Network. The first-stage distillation has injected the geometric and structural priors of the dense LiDAR representations into the 4D radar representations. Therefore, in the second-stage distillation, we replace a heavyweight U-Net [24] with a lightweight diffusion network to accelerate inference without compromising noise regression accuracy. As shown in Fig. 3, given a timestep t, we first encode it into a time embedding e_t and fuse it with the noisy features (either $\mathbf{F}^{\mathrm{D}}_{\mathrm{l},t}$ from the teacher side or $\mathbf{F}^{\mathrm{R}}_{\mathrm{r},t}$ from the student side). The fused features are then processed by two bottleneck blocks and an output head to predict the noise, ϵ. Each bottleneck block preserves spatial resolution and transforms only the channel dimension: a 1×1 convolution compresses channels, a 3×3 convolution introduces local context, and a final 1×1 convolution restores the channel count, with a residual connection to stabilize training. This design achieves accurate noise regression with minimal parameters and computational cost.

E. Loss Function

The denoised reconstructed features, $\mathbf{F}^{\mathrm{D}}_{\mathrm{r}}$, are fed into the point cloud reconstruction module to generate a dense 4D radar point cloud. Using the LiDAR point cloud as the ground truth, the student's reconstruction loss, $\mathcal{L}_{\mathrm{recon}}$, is then calculated following the same reconstruction paradigm as the teacher side.

Our KD framework jointly optimizes for both distillation and diffusion denoising. The diffusion branch learns a denoising prior via the diffusion loss, while the distillation branch learns a sparse-to-dense representation mapping through two distinct distillation losses. The total loss function for the student side is:

$$\mathcal{L}_{\mathrm{student}}=\lambda_1\mathcal{L}_{\mathrm{recon}}+\lambda_2\mathcal{L}^{\mathrm{rec}}_{\mathrm{distill}}+\lambda_3\mathcal{L}^{\mathrm{diff}}_{\mathrm{distill}}+\lambda_4\mathcal{L}_{\mathrm{diff}}. \quad (15)$$

where λ_1, λ_2, λ_3 and λ_4 are the weights for each loss term.

IV. Implementation

A. Dataset

We evaluate our method on the View-of-Delft (VoD) [16] and an in-house dataset.

VoD Dataset. We follow 4DRVO-Net's dataset partitioning scheme [30] and adopt a more challenging setting to increase scene disparity between the training and test sets: sequences 03, 04, and 22 are held out for testing, and the remaining sequences are used for training.

In-house Dataset. To further assess generalization across scenes and 4D radar sensor, we construct an in-house dataset spanning nine closed-campus scenarios. The dataset comprises 20,720 synchronized frames captured by a 128-beam LiDAR and a 4D radar. We hold out sequence 09 (4,012 frames; two driving laps) for testing and use the remaining sequences for training.

B. Implementation Details

For the BEV feature maps, the voxelization range is set to $[0,32]\times[-16,16]\times[-2,4]m$ along the (x,y,z) axes. The voxel size is set to $(0.1\times0.1\times0.15)m$. We set the weights in Eq. 7 to $\rho^{(1)}=\rho^{(2)}=\rho^{(3)}=1$ and $\zeta^{(1)}=\zeta^{(2)}=\zeta^{(3)}=10$. In Eq. 8 and Eq. 14, we use $\alpha=10$ and $\gamma=20$. In Eq. 15, the weights are $\lambda_1=1$, $\lambda_2=0.01$, $\lambda_3=5$, $\lambda_4=10$. We leverage the DDIM [23] with a total of $T=1000$ diffusion steps. The predefined starting timestep $m=500$, sampling steps $T_m=50$. The sampling interval $n=10$ for Eq. 13. All experiments are conducted on a single NVIDIA RTX 4090 GPU. We use the Adam optimizer with an initial learning rate of 10^{-3} and a OneCycleLR scheduler. The batch size is set to 4. The teacher and student networks are trained for 60 and 90 epochs, respectively. Our data preprocessing follows R2LDM [15], which includes removing ground points from the LiDAR data and cropping the LiDAR point cloud to match the Field of View (FOV) of the 4D radar.

C. Evaluation Metrics

We evaluate the model's performance from two aspects: 3D geometric accuracy and BEV spatial distribution consistency. For the former, we employ the Chamfer Distance (CD), Modified Hausdorff Distance (MHD) [15], [32], and F-score [33]. For the latter, we use the Jensen–Shannon Discrepancy (JSD) and Maximum Mean Discrepancy (MMD) [31], [34]. All evaluations use the preprocessed LiDAR point clouds as the reference ground truth.

D. Baselines

Given the limited availability of open-source methods for 4D radar point-cloud post-processing super-resolution, we select two representative baselines for comparison:

R2DM [31] is a diffusion-based generative method for LiDAR point cloud super-resolution. In our study, we adapt this method to the 4D radar domain by using the 4D radar's range image as input.

R2LDM [15] is the current SOTA work for 4D radar point cloud super-resolution. It performs diffusion in a latent space of voxelized features to enhance the details of the reconstructed point cloud.

V. Experimental Results

A. Performance Evaluation

As shown in Tab. I, MSDNet achieves SOTA results across all point cloud reconstruction metrics on both the VoD and in-house datasets. Both R2LDM and R2DM are

TABLE I

QUANTITATIVE RESULTS ON THE VOD DATASET [16] AND IN-HOUSE DATASET. THE BEST RESULT IS **BOLD**, AND THE SECOND BEST IS UNDERLINED.

Method	VoD Dataset					In-house Dataset				
	CD↓	MHD↓ ($\times 10^{-2}$)	F-score↑	JSD↓	MMD↓ ($\times 10^{-4}$)	CD↓	MHD↓ ($\times 10^{-2}$)	F-score↑	JSD↓	MMD↓ ($\times 10^{-4}$)
R2DM [31]	11.93	276.28	0.12	0.30	11.63	17.40	258.94	0.06	0.50	45.29
R2LDM [15]	6.07	172.28	0.27	0.35	21.88	9.67	288.53	0.13	0.45	19.49
MSDNet (Ours)	**5.16(↓15.0%)**	**58.98(↓65.8%)**	**0.39(↑44.4%)**	**0.21(↓30.0%)**	**5.51(↓52.6%)**	**6.59(↓31.9%)**	**156.19(↓39.7%)**	**0.17(↑30.8%)**	**0.32(↓28.9%)**	**7.49(↓61.6%)**

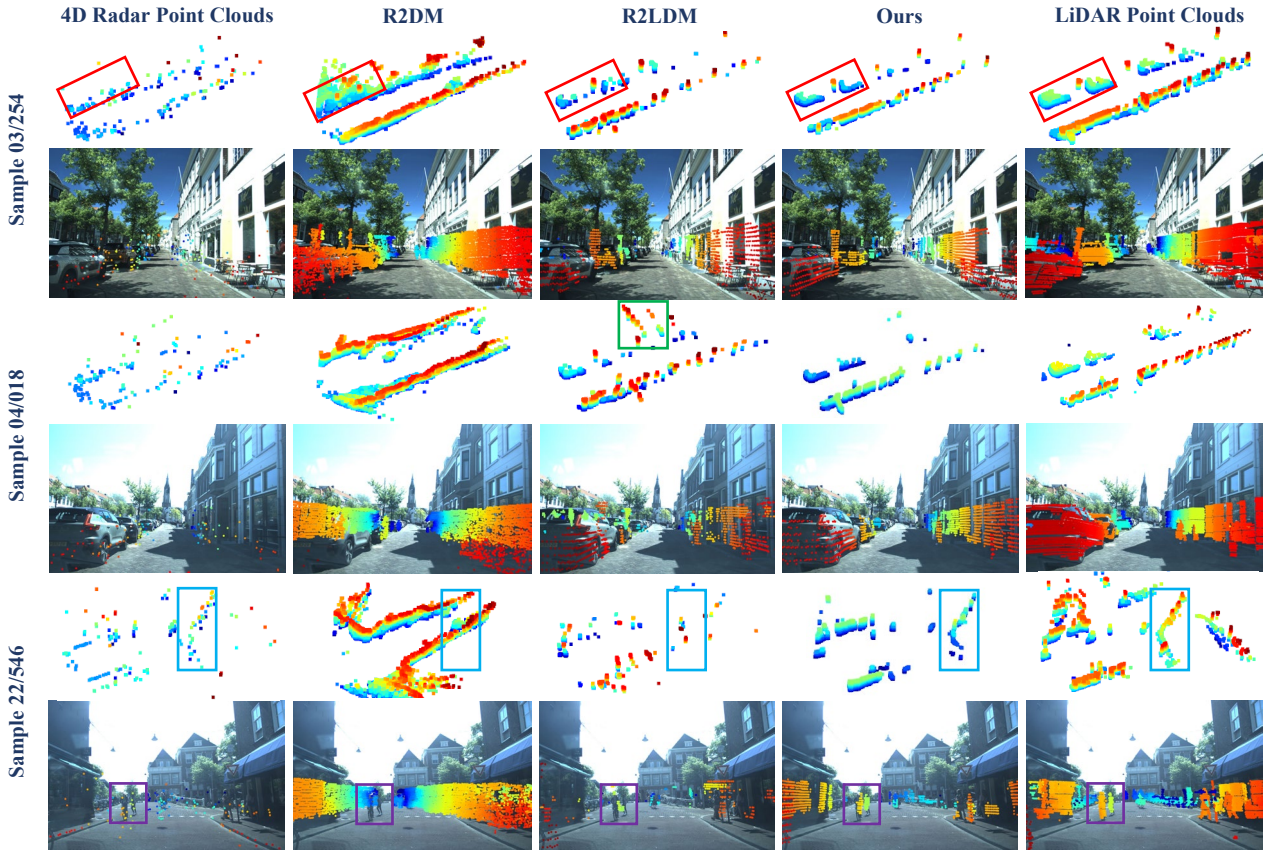

Fig. 4. Qualitative results on the VoD dataset [16].

diffusion-based super-resolution methods, but they are prone to generating hallucinatory points when under-constrained and overly reliant on training scene priors. Notably, R2DM, originally designed for LiDAR, is more sensitive to 4D radar noise, often resulting in denser point clouds but with inferior geometric fidelity.

In contrast, MSDNet effectively transfers the geometric and structural priors from dense LiDAR via a two-stage distillation process. It robustly regularizes the 4D radar features through the dual constraints of feature reconstruction and diffusion denoising. This leads to significant improvements in the generated point cloud's global similarity, local fidelity, and BEV spatial distribution consistency. Specifically, on the VoD dataset, MSDNet achieves improvements of 65.8% in MHD, 44.4% in F-score, and 52.6% in MMD. Our method also maintains leading performance on a more complex in-house dataset (with abundant trees and buildings), demonstrating strong cross-scene and cross-sensor generalization.

TABLE II

RESULTS OF THE ABLATION STUDY.

	Method	CD↓	MHD↓($\times 10^{-2}$)	JSD↓	MMD↓($\times 10^{-4}$)
(a)	w/o RGFD	5.88	66.43	0.22	5.94
	w/o DGFD	9.29	135.57	0.27	12.03
(b)	w/o Noise Adapter	5.53	61.74	0.22	6.21
	w/o Time Embedding	5.36	60.40	0.22	6.05
(c)	with U-Net	6.13	101.88	**0.20**	**5.48**
	MSDNet (Full)	**5.16**	**58.98**	0.21	5.51

B. Ablation Study

To analyze the effectiveness of each module in our proposed method, we conduct an ablation study on the VoD dataset by removing or replacing some key components. The results are presented in Tab. II and Tab. III.

Benefits of the Multi-Stage Distillation Module. As shown in Tab. II(a), RGFD transfers dense LiDAR feature priors to sparse 4D radar features via feature reconstruction, substantially enhancing their representational capacity.

TABLE III

ABLATION ON DIFFUSION SAMPLING STEPS.

T_m	CD↓	MHD↓($\times 10^{-2}$)	JSD↓	MMD↓($\times 10^{-4}$)	Runtime(s)
3	6.02	73.26	0.22	6.48	0.08s
10	5.34	65.94	0.21	5.63	0.09s
50	**5.16**	**58.98**	0.21	5.51	0.10s
100	**5.16**	64.07	**0.20**	**5.40**	0.13s

TABLE IV

AVERAGE RUNTIME AND PARAMETERS ON THE VOD DATASET.

Metrics	R2DM	R2LDM	MSDNet (Ours)
Runtime(s)↓	0.76	0.96	0.10
Parameters(M)↓	31.1	40.3	20.5

DGFD employs a diffusion model to remove residual noise from the distilled features, bringing them closer to dense LiDAR features. Consequently, all point cloud reconstruction metrics exhibit significant improvements.

Benefits of Noise Adapter and Time Embedding. As shown in Tab. II(b), with the introduction of noise adapter, the matching between the reconstructed features and the predefined noise level becomes more precise, leading to significant gains in reconstruction metrics. The time embedding module, by strengthening the correlation between features and their predefined timesteps, effectively boosts both noise matching accuracy and reconstruction fidelity.

Different Diffusion Network. As shown in Tab. II(c), we replaced our lightweight diffusion network with a U-Net [24] as the noise predictor. The average runtime reached 1.03s, which is significantly higher than the 0.10s achieved by our method; meanwhile, the performance on reconstruction metrics was comparable between the two. This result shows our lightweight design greatly cuts inference cost without losing reconstruction quality.

Effect of Sample Steps. As shown in Tab. III, we evaluate the impact of different numbers of sampling steps, T_m. $T_m = 50$ achieves the best or second-best performance with a relatively small time overhead. Therefore, we adopt it as our default setting.

C. Visualization

Fig. 4 and Fig. 5 present the qualitative comparisons on the VoD and in-house datasets, respectively. MSDNet is capable of reconstructing more complete and geometrically consistent structures from the sparse 4D radar point clouds, including road contours, side walls, vehicles, trees, pedestrians/cyclists, and distant small objects, with a more uniform point distribution. In contrast, the point clouds reconstructed by R2LDM suffer from an uneven distribution, an insufficient depiction of small objects, and a tendency to produce structural hallucinations (e.g., generating a non-existent turning lane). Although R2DM produces a higher point density, it lacks geometric accuracy and boundary fidelity, and is also prone to generating spurious structures.

D. Runtime and Params

As shown in Tab. IV, we compare the average inference time and parameters for each method on the VoD dataset.

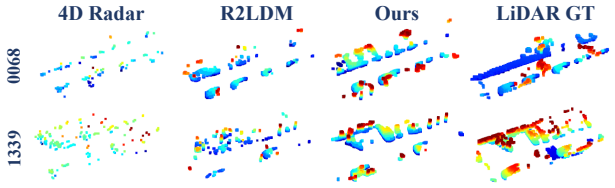

Fig. 5. Qualitative results on the in-house dataset.

TABLE V

4D RADAR ODOMETRY RESULTS ON THE VOD DATASET.

Point Cloud Type	03		04		22		Mean	
	t_{rel}↓	r_{rel}↓	t_{rel}↓	r_{rel}↓	t_{rel}↓	r_{rel}↓	t_{rel}↓	r_{rel}↓
4D radar	0.43	0.80	0.19	0.26	0.28	0.46	0.30	0.51
R2DM	0.79	1.74	0.84	0.73	0.65	1.49	0.76	1.32
R2LDM	0.56	1.05	0.55	0.41	0.57	0.52	0.56	0.66
MSDNet (Ours)	**0.35**	**0.73**	**0.13**	**0.23**	**0.25**	**0.41**	**0.24**	**0.46**

TABLE VI

PLACE RECOGNITION RESULTS ON THE IN-HOUSE DATASET.

Point Cloud Type	Recall@1↑	Recall@3↑	Recall@5↑
4D radar	47.06	64.14	71.16
R2DM	11.20	21.26	26.29
R2LDM	50.89	65.13	70.80
MSDNet (Ours)	**58.54**	**67.68**	**72.71**

Owing to our two-stage distillation and lightweight diffusion model, MSDNet is 89.6% faster than R2LDM and has only half the parameters. This indicates that our method not only improves point cloud reconstruction accuracy but also substantially reduces the model's runtime and parameters.

E. Performance on Downstream Task

4D Radar Odometry. Odometry accuracy is correlated with the clarity and geometric fidelity of the point cloud. We conduct an odometry evaluation on the VoD dataset using the GICP algorithm [35] on both the raw 4D radar point clouds and the point clouds generated by the methods. The evaluation metrics are adopted from 4DRO-Net [36]. As shown in Tab. V, the point clouds generated by MSDNet achieve the best results across all test sequences.

4D Radar Place Reconition. Place recognition performance is highly dependent on the richness and accuracy of the point cloud descriptors. We use ScanContext [37] for place recognition on our in-house dataset, and the evaluation protocol follows that of TDFANet [38]. As shown in Tab. VI, both MSDNet and R2LDM yield significant gains over the raw radar point cloud, with MSDNet comprehensively outperforming R2LDM, which indicates that the point clouds generated by MSDNet possess higher geometric fidelity and a more consistent BEV distribution.

VI. CONCLUSION

In this paper, we presented MSDNet, a multi-stage distillation framework for 4D radar super-resolution that progressively transfers dense LiDAR priors to the 4D radar representation through reconstruction-guided and diffusion-guided stages. This dual-stage approach significantly reduces

inference latency and parameter size while maintaining high reconstruction fidelity. Extensive experiments demonstrate the superiority and strong generalization of MSDNet, as well as its benefits on downstream tasks.

REFERENCES

[1] S. Lu, H. Zhou, and G. Zhuo, "Dnoi-4dro: Deep 4d radar odometry with differentiable neural-optimization iterations," *arXiv preprint arXiv:2505.12310*, 2025.

[2] M. Huang, S. Lu, and G. Zhuo, "Low-cost fusion odometry algorithm based on 4d radar and pseudo-lidar: Bridging the gap between 4d radar and images in 3d space," SAE Technical Paper, Tech. Rep., 2024.

[3] G. Peng, H. Li, Y. Zhao, J. Zhang, Z. Wu, P. Zheng, and D. Wang, "Transloc4d: Transformer-based 4d radar place recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 595–17 605.

[4] F. Ding, Z. Luo, P. Zhao, and C. X. Lu, "milliflow: Scene flow estimation on mmwave radar point cloud for human motion sensing," in *European Conference on Computer Vision*. Springer, 2024, pp. 202–221.

[5] F. Ding, X. Wen, Y. Zhu, Y. Li, and C. X. Lu, "Radarocc: Robust 3d occupancy prediction with 4d imaging radar," *Advances in Neural Information Processing Systems*, vol. 37, pp. 101 589–101 617, 2024.

[6] X. Zhang, L. Wang, J. Chen, C. Fang, G. Yang, Y. Wang, L. Yang, Z. Song, L. Liu, X. Zhang *et al.*, "Dual radar: A multi-modal dataset with dual 4d radar for autononous driving," *Scientific data*, vol. 12, no. 1, p. 439, 2025.

[7] L. Fan, J. Wang, Y. Chang, Y. Li, Y. Wang, and D. Cao, "4d mmwave radar for autonomous driving perception: A comprehensive survey," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 4, pp. 4606–4620, 2024.

[8] M. Jiang, G. Xu, H. Pei, Z. Feng, S. Ma, H. Zhang, and W. Hong, "4d high-resolution imagery of point clouds for automotive mmwave radar," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 1, pp. 998–1012, 2023.

[9] F. Zhao, G. Hu, C. Zhan, and Y. Zhang, "Doa estimation method based on improved deep convolutional neural network," *Sensors*, vol. 22, no. 4, p. 1305, 2022.

[10] R. Franceschi and D. Rachkov, "Deep learning-based radar detector for complex automotive scenarios," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 303–308.

[11] Z. Han, J. Jiang, X. Ding, J. Wang, Q. Meng, S. Xu, L. He, and J. Wang, "Denserradar: A 4d millimeter-wave radar point cloud detector based on dense lidar point clouds," in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2024, pp. 930–936.

[12] G. Cazenavette, A. Sud, T. Leung, and B. Usman, "Fakeinversion: Learning to detect images from unseen text-to-image models by inverting stable diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 759–10 769.

[13] X. Li, J. Lu, K. Han, and V. A. Prisacariu, "Sd4match: Learning to prompt stable diffusion model for semantic matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 27 558–27 568.

[14] K. Luan, C. Shi, N. Wang, Y. Cheng, H. Lu, and X. Chen, "Diffusion-based point cloud super-resolution for mmwave radar data," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 11 171–11 177.

[15] B. Zheng, S. Lu, R. Huang, M. Huang, F. Lu, W. Tian, G. Zhuo, and L. Xiong, "R2ldm: An efficient 4d radar super-resolution framework leveraging diffusion model," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025, pp. 769–776.

[16] A. Palffy, E. Pool, S. Baratam, J. F. Kooij, and D. M. Gavrila, "Multi-class road user detection with 3+1d radar in the view-of-delft dataset," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4961–4968, 2022.

[17] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[18] G. Bang, K. Choi, J. Kim, D. Kum, and J. W. Choi, "Radardistill: Boosting radar-based object detection performance via knowledge distillation from lidar features," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 491–15 500.

[19] R. Xu, Z. Xiang, C. Zhang, H. Zhong, X. Zhao, R. Dang, P. Xu, T. Pu, and E. Liu, "Sckd: Semi-supervised cross-modality knowledge distillation for 4d radar object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 9, 2025, pp. 8933–8941.

[20] S.-H. Song, D.-H. Paek, M.-Q. Dao, E. Malis, and S.-H. Kong, "A novel multi-teacher knowledge distillation for real-time object detection using 4d radar," *arXiv e-prints*, pp. arXiv–2502, 2025.

[21] W. Wang, B. Campbell, and S. Munir, "Self-supervised contrastive learning for camera-to-radar knowledge distillation," in *2024 20th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*. IEEE, 2024, pp. 154–161.

[22] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[23] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=St1giarCHLP

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[25] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4490–4499.

[26] T. Wang, X. Hu, Z. Liu, and C.-W. Fu, "Sparse2dense: Learning to densify 3d features for 3d object detection," *Advances in Neural Information Processing Systems*, vol. 35, pp. 38 533–38 545, 2022.

[27] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[28] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 976–11 986.

[29] T. Huang, S. You, F. Wang, C. Qian, and C. Xu, "Knowledge distillation from a stronger teacher," *Advances in Neural Information Processing Systems*, vol. 35, pp. 33 716–33 727, 2022.

[30] G. Zhuo, S. Lu, H. Zhou, L. Zheng, M. Zhou, and L. Xiong, "4drvo-net: Deep 4d radar–visual odometry using multi-modal and multi-scale adaptive fusion," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 6, pp. 5065–5079, 2023.

[31] K. Nakashima and R. Kurazume, "Lidar data synthesis with denoising diffusion probabilistic models," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 14 724–14 731.

[32] A. Prabhakara, T. Jin, A. Das, G. Bhatt, L. Kumari, E. Soltanaghai, J. Bilmes, S. Kumar, and A. Rowe, "High resolution point clouds from mmwave radar," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4135–4142.

[33] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, and T. Brox, "What do single-view 3d reconstruction networks learn?" in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3405–3414.

[34] V. Zyrianov, X. Zhu, and S. Wang, "Learning to generate realistic lidar point clouds," in *European Conference on Computer Vision*. Springer, 2022, pp. 17–35.

[35] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp." in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.

[36] S. Lu, G. Zhuo, L. Xiong, X. Zhu, L. Zheng, Z. He, M. Zhou, X. Lu, and J. Bai, "Efficient deep-learning 4d automotive radar odometry method," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 879–892, 2023.

[37] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4802–4809.

[38] S. Lu, G. Zhuo, H. Wang, Q. Zhou, H. Zhou, R. Huang, M. Huang, L. Zheng, and Q. Shu, "Tdfanet: Encoding sequential 4d radar point clouds using trajectory-guided deformable feature aggregation for place recognition," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 1–7.