IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hangzhou, China, 2025.

DiffSSC: Semantic LiDAR Scan Completion using Denoising Diffusion Probabilistic Models

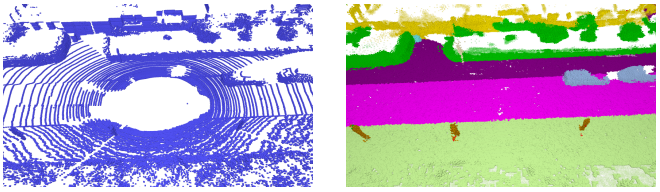
Helin Cao and Sven Behnke

***Abstract*— Perception systems play a crucial role in autonomous driving, incorporating multiple sensors and corresponding computer vision algorithms. 3D LiDAR sensors are widely used to capture sparse point clouds of the vehicle's surroundings. However, such systems struggle to perceive occluded areas and gaps in the scene due to the sparsity of these point clouds and their lack of semantics. To address these challenges, Semantic Scene Completion (SSC) jointly predicts unobserved geometry and semantics in the scene given raw LiDAR measurements, aiming for a more complete scene representation. Building on promising results of diffusion models in image generation and super-resolution tasks, we propose their extension to SSC by implementing the noising and denoising diffusion processes in the point and semantic spaces individually. To control the generation, we employ semantic LiDAR point clouds as conditional input and design local and global regularization losses to stabilize the denoising process. We evaluate our approach on autonomous driving datasets, and it achieves state-of-the-art performance for SSC, surpassing most existing methods.**

I. Introduction

Perception systems collect low-level attributes of the surrounding environment, such as depth, temperature, and color, through various sensor technologies. These systems leverage machine learning algorithms to achieve high-level understanding, such as object detection and semantic segmentation. 3D LiDAR is widely used in self-driving cars to collect 3D point clouds. However, 3D LiDAR has inherent limitations, such as unobservable occluded regions, gaps between sweeps, non-uniform sampling, noise, and outliers, which present significant challenges for high-level scene understanding.

To provide dense and semantic scene representations for downstream decision-making and action systems, Semantic Scene Completion (SSC) has been proposed, aimed at jointly predicting missing points and semantics from raw LiDAR point clouds. Given its potential to significantly improve scene representation quality, this task has garnered significant attention in the robotics and computer vision communities. Understanding 3D surroundings is an inherent human ability, developed from observing a vast number of complete scenes in daily life. When humans observe a scene from a single view, they can leverage prior knowledge to infer unseen geometry and semantics. Drawing inspiration from this capability, the SSC model learns prior knowledge of scenes, $P(\text{scene})$, by estimating the complete scene from partial inputs during training. During inference, new partial inputs captured from the scene serve as the likelihood, $P(\text{observation}|\text{scene})$, and the model finally estimates a reasonable posterior result. Notably, the final estimation is not a unique answer but rather a sample from the posterior distribution, $P(\text{scene}|\text{observation})$. This aligns with intuition, since humans also infer plausible results from partial inputs, while the unobserved parts can have multiple possible completions.



(a) Sparse LiDAR Input (b) Dense Semantic Estimation

Fig. 1: DiffSSC estimates unseen points with semantics (b) from raw LiDAR point clouds (a). The unknown areas, as defined by ground truth, are visualized at 20% opacity in (b).

However, most traditional SSC methods are limited to learning the prior distribution of data directly, i.e., training a model to estimate the target output directly from partial inputs. Another approach to learning prior distributions is to estimate residuals. Denoising Diffusion Probabilistic Models (DDPMs) gradually inject noise into the data in the forward diffusion process and employ a denoiser to learn how to remove these noise residuals. The denoiser iteratively predicts and removes noise, allowing the model to recover high-quality data from pure noise. This mechanism effectively learns the prior distribution of the data, which has the potential to be applied in SSC tasks.

In this work, we propose DiffSSC, a novel SSC approach leveraging DDPMs. As shown in Fig. 1, our method jointly estimates missing geometry and semantics from a scene using raw sparse LiDAR point clouds. During training, the model learns the prior distribution by predicting residuals at different noise intensity levels. These multi-level noisy data are generated from ground truth using data augmentation. In the inference stage, the sparse semantic logits serve as conditional input, and the model generates a dense semantic scene from pure Gaussian noise through a multi-step Markov process. We model both the point and semantic spaces, designing the forward diffusion and reverse denoising processes to enable the model to learn the scene prior to the semantic point cloud representation. In summary, our key contributions are:

This research has been supported by MBZIRC prize money. All authors are with the Autonomous Intelligent Systems group, Computer Science Institute VI – Intelligent Systems and Robotics – and the Center for Robotics and the Lamarr Institute for Machine Learning and Artificial Intelligence, University of Bonn, Germany; caoh@ais.uni-bonn.de

- We utilize DDPMs for the SSC task, introducing a residual-learning mechanism compared to traditional approaches that directly estimate the complete scene from partial input.
- We jointly model the noise injection process in both the spatial and semantic domains and design corresponding local and global regularization losses to enhance generation quality.
- Our approach operates directly on the point cloud, avoiding quantization errors and reducing memory usage, while making it a more efficient method for LiDAR point clouds.

II. Related Work

A. LiDAR Perception

LiDAR is widely used in various autonomous agents for collecting 3D point clouds from the environment. In the past, extensive research was dedicated to employing LiDAR for odometry [1] and mapping [2], [3]. Given the inherent challenges of LiDAR, including data sparsity, noise, and outliers, researchers concentrated on developing filtering algorithms [4] and robust point cloud registration [5] to achieve accurate and efficient LiDAR-SLAM systems. With the advent of deep learning, researchers began focusing on the semantic properties of LiDAR data, with notable applications in object detection [6] and semantic segmentation [7]. Additionally, unlike dense representations such as images, the sparse nature of LiDAR point clouds presents unique challenges for models. To address these challenges, some researchers focus on estimating the gaps between sweeps and occluded regions from sparse point clouds. This has led to the development of semantic scene completion, an emerging technique in LiDAR perception.

B. Semantic Scene Completion (SSC)

Semantic scene completion (SSC) aims to jointly infer complex geometric structures and diverse semantic categories of a scene from partial observations. Since its introduction, various input data modalities, such as occupancy grids [8], images [9], and LiDAR-camera fusion [10], have been explored. In parallel, a wide array of methodologies, including transformers [11], bird's-eye view (BEV) assistance [12], and object-centric modeling [13], have been employed to advance the state of the art in this domain. However, these approaches generally operate on voxelized grids, which poses specific challenges for LiDAR point clouds, as voxelization can introduce quantization errors, leading to resolution loss and increased memory usage. In this work, we operate directly on point clouds, offering a more efficient and resolution-preserving method for handling LiDAR data.

C. Denoising Diffusion Probabilistic Models

Although diffusion models were originally discovered and proposed in the field of physics, DDPMs [14] were the first to apply this method to generative models. In subsequent research, Rombach et al. [15] introduced latent diffusion models, where the diffusion process is performed in the latent space of the image. This significantly improved computational efficiency and reduced resource consumption, enabling the generation of high-quality and high-resolution images, marking a breakthrough in the field of artistic creation. Beyond artistic applications, diffusion models have been extended to spatiotemporal prediction tasks [16] and LiDAR perception [17], [18], where 3D data is often projected onto range images, allowing methods developed for image domains to be directly applied. Notably, due to the higher demands for accuracy in robotics, controlling the generative process to achieve realistic results remains a significant challenge when applying diffusion models in this field. The recent LiDiff [19] directly applies diffusion models to 3D point clouds for scene completion. However, it still lacks the capability to model and process semantics simultaneously. In this work, we apply DDPM to semantic scene completion, to generate dense and accurate semantic scenes.

III. Methodology

Given a raw LiDAR point cloud, our objective is to estimate a more complete semantic point cloud, including unobserved points with associated semantic labels within gaps and occluded regions. As illustrated in the Fig. 2, we build a diffusion model supported by a semantic segmentation module and a refinement module. First, the raw LiDAR point cloud is semantically segmented using a Cylinder3D [20] to generate initial semantic logits. Next, we upsample the semantic point cloud to increase point density for the diffusion process. The duplicated semantic points undergo a forward diffusion and a reverse denoising process to adjust their positions and semantics. Notably, the semantic point cloud also serves as a conditional input for the diffusion model, guiding the generation process. The generated scene includes semantic points located in gaps and occluded areas. To further enhance the quality of the generated scene, we designed a refinement model based on MinkUNet [21]–[24] to densify the point cloud.

A. Denoising Diffusion Probabilistic Models (DDPMs)

Ho et al. [14] introduced DDPMs to produce high-quality images through iterative denoising from Gaussian noise. This promising capability is driven by a residual learning mechanism that efficiently captures the data distribution. Specifically, the process begins with a forward diffusion step, during which noise is gradually injected into the target data over T steps. The model is then trained to estimate the noise injected at each step. By predicting and removing noise at time step t, the model generates results that closely approximate the raw data distribution.

1) Forward Diffusion Process: Assuming a sample $\boldsymbol{x}_0 \sim q(\boldsymbol{x})$ from a target data distribution, the diffusion process gradually adds noise to $\boldsymbol{x}_0$ over T steps, producing a sequence $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$. When T is large enough, $q(\boldsymbol{x}_T)$ is approximately equal to a normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$. The intensity of noise added at each step is determined by the

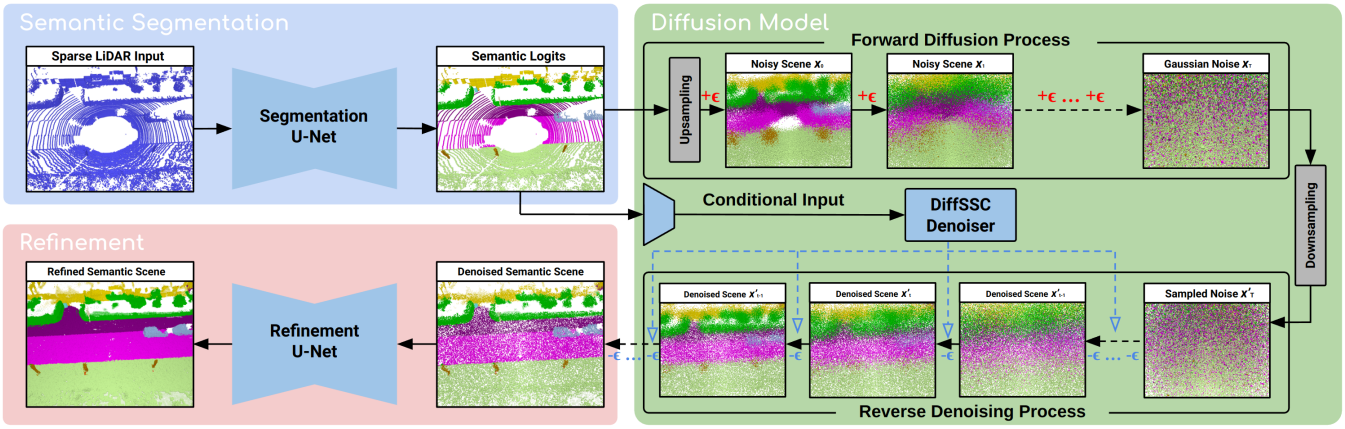


Fig. 2: The overall pipeline of DiffSSC. The raw LiDAR point cloud is semantically segmented using Cylinder3D [20] to generate initial semantic logits. The semantic point cloud is then upsampled. These duplicated points undergo forward diffusion and reverse denoising. The original semantic point cloud serves as a conditional input, guiding the scene generation. To further enhance the generated scene, we introduce a refinement model based on MinkUNet [21]–[24], which increases the density of the point cloud.

noise intensity factors $\beta_1, \ldots, \beta_T$, which significantly influences the performance of the diffusion model. Specifically, at step t, Gaussian noise amplified by β_t is sampled and added to $\boldsymbol{x}_{t-1}$. In [14], the noise parameter β_t is determined using a linear schedule, starting from an initial value β_0 and linearly increasing over T steps to a final value β_T. Subsequently, several improved noise schedules have been proposed, such as the cosine schedule [25] and the sigmoid schedule [26]. Due to the inefficiency of adding noise step by step, especially during batch loading, where the noise from different steps can be shuffled, one can simplify this process by sampling $\boldsymbol{x}_t$ from $\boldsymbol{x}_0$ without computing the intermediate steps $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{t-1}$. To achieve this, Ho et al. [14] define $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, allowing $\boldsymbol{x}_t$ to be sampled as:

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \tag{1}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. It is important to note that when T is large enough, $q(\boldsymbol{x}_T)$ approaches $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ because $\bar{\alpha}_T$ tends to zero.

2) Reverse Denoising Process: The denoising process reverses diffusion and aims to recover the original sample $\boldsymbol{x}_0$ from Gaussian noise. This is accomplished by a denoiser, which estimates and removes the noise at each step. The reverse diffusion step can be formulated as:

$$\boldsymbol{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right) + \sigma_t\boldsymbol{\epsilon},$$

$$\text{with } \sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{2}$$

where $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ is the noise estimated from $\boldsymbol{x}_t$ at step t. The process of generating the original data can be formulated as a Markov process that repeatedly calls the denoiser until $t = 0$. At this point, the model generates a result that approximates $\boldsymbol{x}_0$. Due to the denoiser effectively learning the high quality of the data distribution $q(\boldsymbol{x}_T)$, the generated samples are of similarly high quality.

While the denoising process generates samples with quality similar to the dataset, it only produces random samples. Hence, the denoising process cannot control the generation of specific desired data, which poses challenges for certain downstream applications. [25] addresses this issue by introducing conditional inputs to guide the generation process. This advancement allows us to apply diffusion models to tasks like SSC.

B. Diffusion Semantic Scene Completion

Regarding the principles of DDPMs, we introduce its application in SSC. To focus on the main components, we assume that primary semantic segmentation has been obtained using Cylinder3D. In the context of the diffusion model, the input is a partial semantic point cloud $\mathcal{X} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N\}$, where each semantic point $\boldsymbol{x}^n$ is a tuple of a point position and a semantic probability vector $(\boldsymbol{p}^n, \boldsymbol{s}^n)$. Here, $\boldsymbol{p}^n \in \mathbb{R}^3$ represents the 3D coordinates, and $\boldsymbol{s}^n \in \Delta^{C-1} = \{\boldsymbol{s} \in \mathbb{R}^C \mid \sum_{i=1}^{C} s^i = 1, s^i \geq 0\}$ lies in the standard $(C-1)$-dimensional simplex, assuming there are C classes in total. The output is the estimated complete point cloud $\hat{\mathcal{Y}} = \{\hat{\boldsymbol{y}}^1, \ldots, \hat{\boldsymbol{y}}^M\}$. We generate the reference $\mathcal{Y} = \{\boldsymbol{y}^1, \ldots, \boldsymbol{y}^M\}$ by fusing multiple frames with ground-truth semantic labels and then taking the corresponding region as the input scan $\mathcal{X}$. Our goal is to make the estimated $\hat{\mathcal{Y}}$ as close as possible to the ground truth $\mathcal{Y}$.

As mentioned in Sec. I, by learning scene priors, the model gains the ability to estimate a complete scene (posterior) from partial observations (likelihood). The diffusion model efficiently learns the distribution of the ground truth data, acquiring knowledge of the scene prior. To achieve this, we gradually inject noise into the ground truth $\mathcal{Y}$, resulting in $\mathcal{Y}_1, \ldots, \mathcal{Y}_T$, until $\mathcal{Y}_T$ approximates a Gaussian distribution. However, the noise injection process in Eq. 1 assumes that $\boldsymbol{x}_0$ is approximately isotropic, which is not suitable for LiDAR-scanned 3D scenes due to significant scale variations across the three spatial dimensions. While normalization can compress the scene into a more isotropic

form, it also leads to a significant loss of fine details. To adapt noise injection for LiDAR-scanned 3D scenes, [19] apply local noise offsets at each point. This ensures that the noise intensity remains consistent across all spatial locations. Similarly, semantic categories in autonomous driving follow a long-tailed distribution [27], indicating that their distribution exhibits anisotropy. Inspired by [19], we adopt a local noise offset strategy for semantic noise injection. Specifically, we first scale the one-hot semantic encoding of the ground truth $\mathcal{Y}$ into the logit domain, then add noise offsets independently to each category's semantic logit, and subsequently restore the probabilistic semantic distribution via softmax. Combining the spatial and semantic domains, we propose a local anisotropic noise injection mechanism.

$$\boldsymbol{y}_t^m = \boldsymbol{y}^m + \sqrt{1-\bar{\alpha}_t}\boldsymbol{W}\boldsymbol{\epsilon}, \boldsymbol{W} = \begin{bmatrix} \sigma_p \boldsymbol{I}_3 & \boldsymbol{0} \\ \boldsymbol{0} & \sigma_s \boldsymbol{I}_C \end{bmatrix} \quad (3)$$

Here, we employ an anisotropic scaling matrix $\boldsymbol{W}$, controlled by the scaling factors σ_p and σ_s, to modulate the standard Gaussian noise $\boldsymbol{\epsilon}$, ensuring that it appropriately adapts to the scale differences in both the spatial and semantic logits domains. The modulated noise is then injected into the local semantic point $\boldsymbol{y}^m \in \mathbb{R}^{3+C}$. Given a specific time step $t \in [0, T]$, applying Eq. 3 to all points allows for computing the entire noised scene $\mathcal{Y}_t$ in a single step, eliminating the need for intermediate computations. This significantly reduces both memory consumption and computational time.

To enable the model to generate a corresponding complete semantic scene based on the current partial input, we encode the partial semantic point cloud $\mathcal{X}$ as a conditional input, which is then fed into the model to guide the point cloud generation process. Thus, the denoiser integrates the noised scene $\mathcal{Y}_t$, the conditional input $\mathcal{X}$, and the time step t, which indicates the intensity of noise, to estimate the noise $\boldsymbol{\epsilon}_\theta(\mathcal{Y}_t, \mathcal{X}, t)$.

Based on our residual learning mechanism, we employ the L_2 loss to regularize the local discrepancy between the estimated noise and the real noise, rather than comparing the generated scene and the target scene.

$$L_2 = \left\| \sqrt{1-\bar{\alpha}_t}\boldsymbol{W}\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathcal{Y}_t, \mathcal{X}, t) \right\|^2 \quad (4)$$

Therefore, to generate the final scene, we additionally remove the estimated noise from the sample, followed by transforming the semantic logits into a semantic probability vector via the softmax function.

Besides the local loss L_2 commonly used in DDPM models, [19] propose a global regularization for the mean and variance of the estimated noise, enforcing it to follow the statistical properties of the injected noise, i.e., a Gaussian distribution. While first- and second-order moments effectively regularize noise in the spatial domain, the estimated noise in the semantic domain tends to exhibit a skewed distribution due to the long-tailed nature of semantic data in autonomous driving. Therefore, we introduce skewness regularization as the third-order constraint to improve the noise distribution in the semantic domain. Thus, the overall loss is formulated as

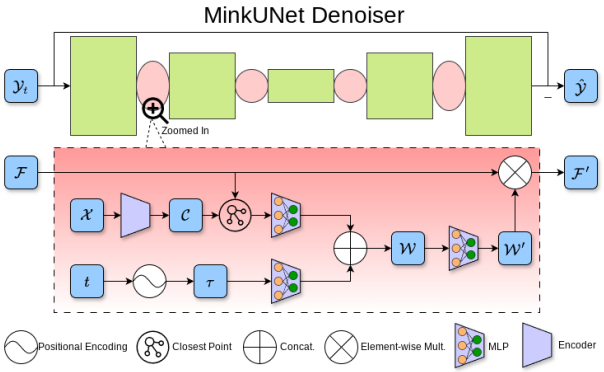


Fig. 3: Architecture of our MinkUNet Denoiser. As shown in the red area, we design information fusion layers and insert them between MinkUNet blocks to integrate the conditional input and step information, guiding the generation of the point cloud.

follows:

$$L = L_2 + \lambda_p(L_{\text{p,mean}} + L_{\text{p,var}}) + \lambda_s(L_{\text{s,mean}} + L_{\text{s,var}} + L_{\text{s,skew}})$$
$$L_{\text{p,mean}} = \bar{\boldsymbol{\epsilon}_p}^2, \quad L_{\text{p,var}} = (\hat{\boldsymbol{\epsilon}_p} - 1)^2$$
$$L_{\text{s,mean}} = \bar{\boldsymbol{\epsilon}_s}^2, \quad L_{\text{s,var}} = (\hat{\boldsymbol{\epsilon}_s} - 1)^2, \quad L_{\text{s,skew}} = \tilde{\boldsymbol{\epsilon}_s}^2 \quad (5)$$

where $\boldsymbol{\epsilon}_p$ and $\boldsymbol{\epsilon}_s$ correspond to the spatial and semantic domains, respectively, with their contributions weighted by λ_p and λ_s. Meanwhile, $\bar{\boldsymbol{\epsilon}_\theta}$, $\hat{\boldsymbol{\epsilon}_\theta}$, and $\tilde{\boldsymbol{\epsilon}_\theta}$ represent the mean, standard deviation, and skewness of the estimated noise $\boldsymbol{\epsilon}_\theta$, respectively. Compared to the local term L_2, the global term is used to regularize the statistical properties of the noise, ensuring that it approximates a Gaussian distribution.

C. Denoiser Architecture

As shown in Fig. 3, the denoiser is based on the MinkUNet architecture [21]–[24]. Given the feature $\mathcal{F}$ extracted from a layer of MinkUNet, we integrate the conditional input and step information between layers to obtain the fused feature $\mathcal{F}'$. The raw semantic point cloud $\mathcal{X}$ is encoded as a conditional input $\mathcal{C}$. To embed the most relevant conditional input into the feature space, a closest point algorithm is employed to effectively align the conditional input with the features. Simultaneously, the step t is encoded as τ using sinusoidal positional encodings. After passing through an MLP individually, the conditional input and step information are concatenated to form the weight $\mathcal{W}$. To align the dimensions with the feature $\mathcal{F}$, $\mathcal{W}$ is processed through an MLP to produce $\mathcal{W}'$. Finally, $\mathcal{W}'$ and $\mathcal{F}$ are element-wise multiplied to form the refined feature $\mathcal{F}'$, which is then passed to the next layer.

D. Refinement

Inspired by Lyu et al. [28], we design a refinement and upsampling scheme based on MinkUNet to further enhance the density of the diffusion model's output. This module predicts k bias $b_k \in \mathbb{R}^3$ for each point position in the completed scene, while the semantics are propagated to the biased points. The refinement module offers a marginal improvement in scene quality, but it functions as interpolating

points in the gaps, rather than learning to predict missing geometry and semantics. The main contribution is made by the diffusion model, as will be demonstrated in the ablation study.

IV. Experiments

A. Benchmark Result

To conduct a comprehensive comparison with other leading SSC methods, we first evaluate our model on the SemanticKITTI [27] Benchmark. SemanticKITTI is a widely used autonomous driving dataset that provides point-wise annotations on raw LiDAR point clouds, extending the original KITTI dataset for semantic understanding tasks. The SSC Benchmark is a subtask within SemanticKITTI, focusing on predicting both the semantic class and occupancy status of each voxel within a grid volume. To obtain the ground truth, the annotated sequential scans are first accumulated and then chopped based on a predefined range represented in the LiDAR sensor's coordinate system: $V_{\text{kitti}} = \{(x, y, z) \mid x \in [0, 51.2]\ \text{m}, y \in [-25.6, +25.6]\ \text{m}, z \in [-3.2, +3.2]\ \text{m}\}$. The extracted region is then voxelized into a $256 \times 256 \times 32$ grid volume, where each voxel represents a $0.2^3\,\text{m}^3$ cube in the real world. Although our method operates directly on point clouds, the discrete nature of point clouds makes it challenging to directly evaluate performance using traditional IoU metrics, which are designed for continuous spatial regions. To address this, we voxelized our results and employed IoU for scene completion and mIoU for semantic scene completion evaluation. While voxelization introduces quantization errors that may slightly degrade our model's performance, this approach ensures a fair and meaningful comparison.

TABLE I: Quantitative results on the SemanticKITTI benchmark.

Method	Reference	IoU(%)	mIoU(%)
LMSCNet [8]	3DV'20	55.3	17.0
S3CNet [12]	CoRL'21	45.0	29.5
SSA-SC [29]	IROS'21	58.8	23.5
JS3C-Net [30]	AAAI'21	56.6	23.8
LODE [31]	ICRA'23	51.2	23.4
SCPNet [32]	CVPR'23	56.1	<u>36.7</u>
TALoS [33]	NeurIPS'24	<u>60.2</u>	**37.9**
Ours	-	**63.4**	27.4

IoU is used to evaluate only the occupancy status, while mIoU assesses performance across all semantic classes. **Best** and <u>second best</u> results are highlighted.

We follow the official dataset split: sequences 00-07 and 09-10 are used for training, 08 for validation, and 11-21 for testing. Model performance is evaluated through the official online benchmark server. The model is trained on an NVIDIA A6000 GPU for 20 epochs. For the diffusion parameters, we employ a cosine schedule to modulate the intensity of noise at each step. Specifically, we set $\beta_0 = 3.5 \times 10^{-5}$ and $\beta_T = 0.007$, with the number of diffusion steps $T = 1000$, and define $\beta_1, \ldots, \beta_{T-1}$ using the following equation.

$$\beta_t = \beta_0 + \frac{1}{2}\left(1 + \cos\left(\frac{t}{T} \cdot \pi\right)\right) \cdot (\beta_T - \beta_0) \tag{6}$$

We set the ratio of global regularization to $\lambda_p = 5.0$ and $\lambda_s = 4.0$. Additionally, we define the scaling factors as $\sigma_p = 1.0$ and $\sigma_s = 0.2$. As shown in Tab. I, our method outperforms all state-of-the-art LiDAR-based approaches in scene completion, indicating that the model effectively captures geometric information. However, for semantic scene completion (measured by mIoU), a obvious gap remains compared to SCPNet and TALoS. It is important to note that both methods leverage additional information beyond single-frame input. Specifically, SCPNet utilizes multi-frame knowledge distillation to enhance the prediction of the current frame, while TALoS employs a test-time adaptation strategy, incorporating multi-frame observations and future data to refine its outputs. In contrast, DiffSSC relies solely on single-frame information and does not incorporate online model adjustments.

While the SemanticKITTI SSC benchmark provides a valuable framework for evaluating SSC models, its design is notably influenced by early indoor SSC tasks. These early methods mainly relied on RGB-D cameras, which inherently limited perception to front-facing scenes. To leverage the technical foundations of these methods and simplify their transition to outdoor environments, SemanticKITTI restricts the LiDAR field of view to the front half, covering only the range $[-90°, +90°]$ in the sensor coordinate system, while completely ignoring the rear part of the scene. Although this simplification facilitated early exploration of outdoor SSC, it fundamentally deviates from the natural properties of LiDAR data and the core requirements of autonomous driving tasks. Rear-view perception is equally critical for driving safety, especially for tasks like lane changes, reversing, and obstacle avoidance. Since LiDAR sensors capture data through continuous $360°$ rotations without directional bias, restricting input to the front half disrupts the spatial consistency of the data and limits the model's ability to fully understand the driving environment. To address this limitation, we extend the scene completion task to a full $360°$ panoramic view, preserving the intrinsic characteristics of LiDAR data and providing a more comprehensive representation of real-world autonomous driving scenarios.

B. Extended Experiment on Panoramic Scenarios

1) Panoramic Settings: In the raw SSC setting of SemanticKITTI, the scene is limited to a cuboid region V_{kitti}, covering the range $[-90°, +90°]$. To extend this to the full $[-180°, +180°]$ panoramic range while preserving spatial symmetry, the panoramic volume is defined as the combination of two SemanticKITTI volumes facing forward and backward. Specifically, in the LiDAR's local coordinate system, this is formulated as: $V_{\text{pano}} = \{(x, y, z) \mid x \in [-51.2, 51.2]\ \text{m}, y \in [-25.6, +25.6]\ \text{m}, z \in [-3.2, +3.2]\ \text{m}\}$. The front and rear halves of the panoramic

volume are designed to be identical to the original SemanticKITTI volume. Given the similar statistical characteristics of LiDAR data in the front and rear regions, retaining the original SemanticKITTI volume settings allows baseline methods to be seamlessly transferred to the panoramic setting without significant performance degradation.

We generate the ground truth following the guidelines of SemanticKITTI. First, using the pose information of each frame, we construct a global map by aggregating the semantic LiDAR sweeps within the sequence. Next, we extract the region within V_{pano}, with the LiDAR positioned at its center. Additionally, unknown areas defined by the raw dataset are mapped into V_{pano} using the known poses, and these regions are excluded from the evaluation.

Besides SemanticKITTI, we also conduct panoramic experiments on the SSCBench-KITTI360 dataset [34]. SSCBench-KITTI360 is another SSC benchmark derived from KITTI-360 [35], with semantic information aligned to the SemanticKITTI format. This consistency allows SSC methods evaluated on SemanticKITTI to be seamlessly transferred to the KITTI-360 scenario. However, both benchmarks only use the front half of the LiDAR scan as input. To address this, we apply the same panoramic processing approach to SSCBench-KITTI360 as we did for SemanticKITTI.

2) Baselines: We compare our approach against LMSCNet [8], JS3C-Net [30], and LODE [31]. Both LMSCNet and JS3C-Net take the front half of the quantized LiDAR sweep as input and are evaluated on the SSC benchmark of SemanticKITTI. LODE primarily focuses on geometry completion using implicit representations; however, to demonstrate its flexibility, the authors also report results with extended semantic parsing.

To enable these baselines to predict panoramic scenes, we split the 360° LiDAR point cloud into two halves and feed them separately into the baseline models. The front half follows the same settings as in SemanticKITTI, while the rear half is rotated by 180° before being passed to the models. After obtaining predictions for both halves, we concatenate them to form the complete panoramic scene. Although the baselines were trained solely on the front part of the scene, the statistical characteristics of LiDAR data in the front and rear regions are similar. This suggests that models trained on the front half remain effective when applied to the rear region. This approach minimizes performance degradation due to domain shift, ensuring a fair comparison. Additionally, we directly utilized the official code and pretrained checkpoints from the baselines to predict the panoramic scenes, further maintaining consistency in evaluation.

While these baselines have reported results on SemanticKITTI, they had not previously been tested on SSCBench-KITTI360 [34]. To supplement our evaluation, we ran these baselines on SSCBench-KITTI360 without fine-tuning. Since the semantic labels and the overall pipeline in SSCBench-KITTI360 are consistent with SemanticKITTI, the baselines could be seamlessly applied to this dataset.

3) Training and Inference: Since our method operates directly on point clouds rather than voxel-based volumes, the processing of training pairs differs from that of the baselines. Although we generate the ground truth by aggregating the semantic LiDAR sweeps and extracting point clouds within the V_{pano}, we do not further voxelize the data. To facilitate better diffusion and learning, we define the generated scene range during both training and inference as a spherical area centered on the LiDAR with a radius of 60 meters, i.e., $V_{\text{sphere}} = \{(x, y, z) \mid \sqrt{x^2 + y^2 + z^2} \leq 60\,\text{m}\}$. During evaluation, we voxelize the predicted point cloud scene and extract the portion within V_{pano}. Thus, while the generated scene range differs from V_{pano}, the evaluation region is restricted to V_{pano}, ensuring consistency with the baselines. Moreover, the model does not leverage any information outside V_{pano} but within V_{sphere} during the learning process, ensuring a fair comparison with the baselines. Our model is trained and validated purely on SemanticKITTI, using sequences 00-07 for training and sequences 09-10 for validation. We evaluate our model on the official validation sets of both datasets: sequence 08 of SemanticKITTI and sequence 07 of SSCBench-KITTI360. Since the baselines were not fine-tuned on SSCBench-KITTI360, we likewise did not perform any fine-tuning on SSCBench-KITTI360.

4) Experimental Results: Based on the experimental settings described above, we compare the performance of existing SSC methods with our approach in Tab. II. Although voxel-based evaluation introduces quantization errors, the output of our diffusion model surpasses all baselines, demonstrating the effectiveness of our method.

TABLE II: Quantitative results of the panoramic experiments on the SemanticKITTI and SSCBench-KITTI360 validation sets.

Method	SemanticKITTI		SSCBench-KITTI360	
	IoU(%)	mIoU(%)	IoU(%)	mIoU(%)
LMSCNet [8]	48.2	15.4	33.6	13.5
JS3C-Net [30]	51.3	21.4	35.6	17.0
LODE [31]	50.6	18.2	38.2	15.4
Ours	**60.3**	**26.7**	**47.3**	**20.4**

Best results are highlighted in bold.

Qualitative results are presented in Fig. 4. To highlight the advantages of our approach, which operates directly on point clouds, we visualize samples from both the SemanticKITTI and SSCBench-KITTI360 datasets in point cloud form. For voxel-based methods, point clouds are generated by sampling the center point of each occupied voxel. As shown in Fig. 4, our DiffSSC model predicts more accurate semantic segmentation of the background and offers a more precise representation of foreground shapes. Moreover, voxel-based baselines, which estimate the scene using two halves of a LiDAR sweep, exhibit discontinuities at the boundary between the front and rear parts. This further underscores the importance of learning from panoramic LiDAR data.

C. Ablation Studies

To systematically analyze the contribution of each component in our model, particularly the core diffusion model

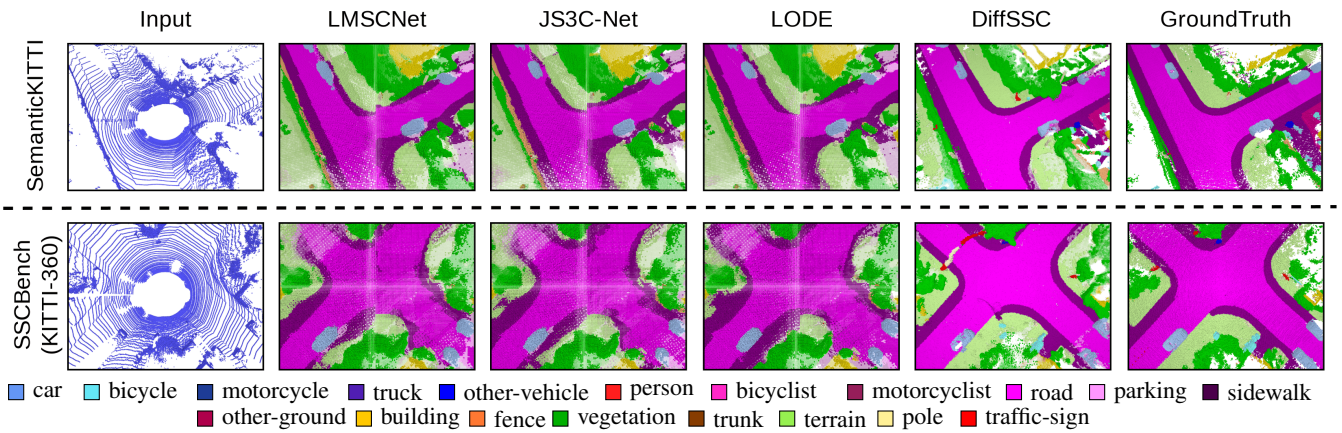


Fig. 4: Qualitative results of the panoramic experiments on the SemanticKITTI and SSCBench-KITTI360 validation sets. All 19 classes are displayed without empty spaces. Predicted points located in unknown regions are visualized with 20% opacity.

TABLE III: The results of ablation studies for the proposed DiffSSC. The performance is evaluated on SemanticKITTI and SSCBench-KITTI360 validation sets. **Best** results are highlighted.

	Method	Module			Noise Schedule			Regularization		SemanticKITTI		SSCBench-KITTI360	
		Segmentation	Diffusion	Refinement	Linear	Sigmoid	Cosine	Local	Global	IoU(%)	mIoU(%)	IoU(%)	mIoU(%)
Ours	A	✓	✓	✓	-	-	✓	✓	✓	**60.3**	**26.7**	**47.3**	**20.4**
Module-level	B	✓	-	✓	-	-	-	-	-	23.4	7.6	20.7	7.2
	C	✓	✓	-	-	-	✓	✓	✓	58.7	26.3	42.1	19.3
Policy-level	D	✓	✓	-	✓	-	-	✓	✓	52.6	20.7	37.1	16.3
	E	✓	✓	-	-	✓	-	✓	✓	56.9	25.8	40.9	18.5
	F	✓	✓	-	-	-	✓	✓	-	40.3	10.9	30.7	9.6

and key learning strategies, we ablate our method on SemanticKITTI and SSCBench-KITTI360, with results summarized in Tab. III. Our model architecture comprises three primary modules: semantic segmentation, diffusion, and refinement. We performed module-level ablations to evaluate the individual contributions of each submodule. Additionally, we conducted policy-level ablations to examine the influence of two critical factors on the diffusion model: the noise schedule function and global regularization.

1) Module-level: The semantic segmentation module is essential for providing initial semantic priors. Therefore, we focused on analyzing the individual contributions of the diffusion and refinement modules. In Method B, we removed the core diffusion module from our pipeline, directly refining the output of Cylinder3D. Without the diffusion step, the noise schedule and regularization become irrelevant. This resulted in poor performance, indicating that refinement alone cannot effectively predict unknown areas in the scene. In Method C, we used only the diffusion model without refinement. While performance slightly dropped compared to the full Method A, this suggests that refinement improves scene densification but relies on accurate predictions from the diffusion model.

2) Policy-level: As mentioned in Sec. III, the noise schedule determines the intensity of noise injected at each step, commonly including linear, cosine, and sigmoid schedules. We investigated the impact of different noise schedules on the diffusion process. To isolate this effect, we removed the refinement module, allowing a clearer view of how the noise schedule influences diffusion. Results are shown in Tab. III. Comparing Method C, D, and E, we observe that the linear schedule, being the simplest form, performs significantly worse than the other two schedules. The cosine schedule, with its S-shaped curve and precise control over noise introduction, balances faster convergence with high final generation quality, achieving the best results. The sigmoid schedule also shows competitive performance, slightly lagging behind the cosine schedule. Therefore, we adopted the cosine schedule in our main results.

We also investigated the impact of global regularization on model performance. By setting $\lambda_p = 5.0$ and $\lambda_s = 4.0$, we removed global regularization in Method F. Compared to Method C ($\lambda_p = \lambda_s = 0$), the model exhibited poorer performance, highlighting the advantages of incorporating global regularization.

V. Conclusions and Outlook

We proposed DiffSSC, a novel SSC approach based on DDPMs. It takes raw LiDAR point clouds as input and jointly predicts missing points along with their semantic labels, thereby extending the application boundaries of diffusion models. We evaluated our method on two autonomous driving datasets, achieving state-of-the-art performance. In future work, we plan to explore integrating cross-modal signals and prompt-guided learning to enhance scene understanding [36]–[38]. We will also explore strategies for improving

inference efficiency and reducing resource consumption, inspired by recent advances in lightweight modeling [39], [40].

REFERENCES

[1] J. Quenzel and S. Behnke, "Real-time multi-adaptive-resolution-surfel 6D lidar odometry using continuous-time trajectory optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 5499–5506.

[2] D. Droeschel and S. Behnke, "Efficient continuous-time slam for 3D lidar-based online mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5000–5007.

[3] X. Zhong, Y. Pan, J. Behley, and C. Stachniss, "SHINE-Mapping: Large-scale 3D mapping using sparse hierarchical implicit neural representations," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[4] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[5] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "KISS-ICP: In defense of point-to-point ICP – simple, accurate, and robust registration if done the right way," *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 2, pp. 1029–1036, 2023.

[6] M. Wang, D. Li, J. R. Casas, and J. Ruiz-Hidalgo, "Adaptive fusion of lidar features for 3d object detection in autonomous driving," *Sensors*, 2025.

[7] Y. Li, J. Dong, Z. Dong, C. Yang, Z. An, and Y. Xu, "SRKD: Towards efficient 3D point cloud segmentation via Structure- and Relation-aware knowledge distillation," *arXiv preprint arXiv:2506.17290*, 2025.

[8] L. Roldao, R. de Charette, and A. Verroust-Blondet, "LMSCNet: Lightweight multiscale 3D semantic completion," in *International Conference on 3D Vision (3DV)*, 2020, pp. 111–119.

[9] A.-Q. Cao and R. de Charette, "MonoScene: Monocular 3D semantic scene completion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 3991–4001.

[10] H. Cao and S. Behnke, "SLCF-Net: Sequential LiDAR-camera fusion for semantic scene completion using a 3D recurrent U-Net," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 2767–2773.

[11] H. Cao, R. Materla, and S. Behnke, "SWA-SOP: Spatially-aware window attention for semantic occupancy prediction in autonomous driving," *arXiv preprint arXiv:2506.18785*, 2025.

[12] R. Cheng, C. Agia, Y. Ren, X. Li, and B. Liu, "S3CNet: A sparse semantic scene completion network for LiDAR point clouds," in *Conference on Robot Learning (CoRL)*, 2020, pp. 2148–2161.

[13] H. Cao and S. Behnke, "OC-SOP: Enhancing Vision-Based 3d semantic occupancy prediction by Object-Centric awareness," *arXiv preprint arXiv:2506.18798*, 2025.

[14] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[15] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 684–10 695.

[16] H. Zeng, Y. Li, R. Niu, C. Yang, and S. Wen, "Enhancing spatiotemporal prediction through the integration of mamba state space models and diffusion transformers," *Knowledge-Based Systems (KBS)*, 2025.

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

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

[19] L. Nunes, R. Marcuzzi, B. Mersch, J. Behley, and C. Stachniss, "Scaling diffusion models to real-world 3D LiDAR scene completion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[20] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9939–9948.

[21] C. Choy, J. Gwak, and S. Savarese, "4D spatio-temporal convnets: Minkowski convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3075–3084.

[22] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 8958–8966.

[23] C. Choy, J. Lee, R. Ranftl, J. Park, and V. Koltun, "High-dimensional convolutional networks for geometric pattern recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[24] J. Gwak, C. B. Choy, and S. Savarese, "Generative sparse detection networks for 3D single-shot object detection," in *European Conference on Computer Vision (ECCV)*, 2020.

[25] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *IEEE International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 8162–8171.

[26] D. P. Kingma, T. Salimans, B. Poole, and J. Ho, "On density estimation with diffusion models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021, pp. 21 696–21 707.

[27] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[28] Z. Lyu, Z. Kong, X. Xu, L. Pan, and D. Lin, "A conditional point diffusion-refinement paradigm for 3D point cloud completion," in *International Conference on Learning Representations, (ICLR)*, 2022.

[29] X. Yang, H. Zou, X. Kong, T. Huang, Y. Liu, W. Li, F. Wen, and H. Zhang, "Semantic segmentation-assisted scene completion for LiDAR point clouds," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 3555–3562.

[30] X. Yan, J. Gao, J. Li, R. Zhang, Z. Li, R. Huang, and S. Cui, "Sparse single sweep LiDAR point cloud segmentation via learning contextual shape priors from scene completion," in *National Conference on Artificial Intelligence (AAAI)*, vol. 35, no. 4, 2021, pp. 3101–3109.

[31] P. Li, R. Zhao, Y. Shi, H. Zhao, J. Yuan, G. Zhou, and Y.-Q. Zhang, "LODE: Locally conditioned Eikonal implicit scene completion from sparse LiDAR," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 8269–8276.

[32] Z. Xia, Y. Liu, X. Li, X. Zhu, Y. Ma, Y. Li, Y. Hou, and Y. Qiao, "SCPNet: Semantic scene completion on point cloud," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 17 642–17 651.

[33] H. Jang, J. Kim, H. Kweon, and K. Yoon, "TALoS: Enhancing semantic scene completion via test-time adaptation on the line of sight," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[34] Y. Li, S. Li, X. Liu, M. Gong, K. Li, N. Chen, Z. Wang, Z. Li, T. Jiang, F. Yu *et al.*, "SSCBench: Monocular 3D semantic scene completion benchmark in street views," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[35] Y. Liao, J. Xie, and A. Geiger, "Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 45, no. 3, pp. 3292–3310, 2022.

[36] W. Wu, Z. Chen, X. Qiu, S. Song, X. Huang, F. Ma, and J. Xiao, "LLM-Enhanced multimodal fusion for Cross-Domain sequential recommendation," *arXiv preprint arXiv:2506.17966*, 2025.

[37] W. Wu, S. Song, X. Qiu, X. Huang, F. Ma, and J. Xiao, "Image fusion for cross-domain sequential recommendation," in *Companion Proceedings of the ACM Web Conference*, 2025.

[38] W. Wu, X. Qiu, S. Song, Z. Chen, X. Huang, F. Ma, and J. Xiao, "Prompt categories cluster for weakly supervised semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 3198–3207.

[39] Y. Li, Y. Lu, Z. Dong, C. Yang, Y. Chen, and J. Gou, "SGLP: A similarity guided fast layer partition pruning for compressing large deep models," *arXiv preprint arXiv:2410.14720*, 2024.

[40] Y. Li, C. Yang, H. Zeng, Z. Dong, Z. An, Y. Xu, Y. Tian, and H. Wu, "Frequency-Aligned knowledge distillation for lightweight spatiotemporal forecasting," *arXiv:2507.02939*, 2025.