SDNet: LiDAR Semantic Scene Completion with Sparse-Dense Fusion and Input-Aware Label Refinement[†]

Tingming Bai[1], Zhiyu Xiang[12*], Peng Xu[1], Tianyu Pu[1], Kai Wang[1], Eryun Liu[1]

[1]Zhejiang University, China
[2]Innovation Center of Yangtze River Delta of Zhejiang University, China
{incredibai, xiangzy, xxxupeng, 3190105835, kai-wang, eryunliu}@zju.edu.cn

Abstract

LiDAR Semantic Scene Completion (SSC) in autonomous driving requires predicting both dense occupancy and semantic labels from sparse input point cloud. Existing methods typically adopt cascaded architecture for feature dilation and semantic abstraction, which blurs distinctive geometric patterns and reduces feature discriminability. Moreover, given an input, conventional processing of the ground truth labels overlooks voxel predictability in the target, resulting in ill-posed supervision and discards informative voxels. To address these limitations, we propose Sparse-Dense Net (SDNet), a dual-branch architecture that processes the input points through parallel sparse and dense encoders. The complementary features are aligned and fused using a Sparse Dense Feature Fusion (SDFF) module and further refined by a Feature Propagation (FP) module. Additionally, we introduce an input-aware label refinement strategy, including Sparse-Guided Filtering (SGF) to filter unpredictable targets and Ignored Voxel Recycling (IVR) to leverage informative ignored voxels for auxiliary supervision. These innovations enhance both feature learning and label quality. Extensive experiments on the SemanticKITTI and nuScenes OpenOccupancy datasets validate the effectiveness of our approach, with SDNet achieving state-of-the-art performance on both datasets and ranking 1st on the official SemanticKITTI benchmark with 42.1 mIoU, outperforming the previous best by 4.2 (+11.1%).

Code — https://github.com/Incredibai/SDNet

Introduction

LiDAR Semantic Scene Completion (SSC) aims to jointly predict dense 3D volume of occupancy and semantics from input sparse point cloud, providing holistic understanding of the scene. As shown in Fig.1(a) and (b), representative models (Yan et al. 2021; Wang et al. 2023; Xia et al. 2023) adopt cascaded architectures for occupancy dilation and semantic abstraction. However, this design can result in suboptimal feature extraction. For object classes, applying dilation before abstraction may blur distinctive geometric patterns in the original sparse input. For background classes, semantic abstraction on isolated sparse points tends to generate weak and noisy features.

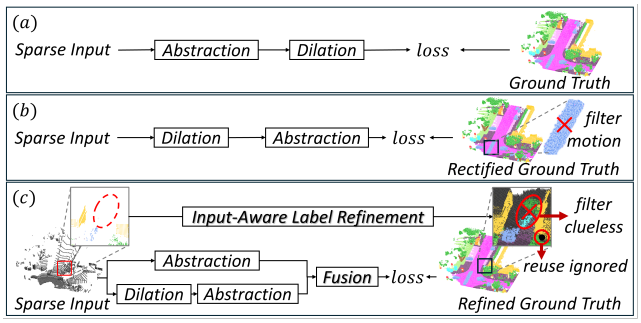

Figure 1: Comparison between the current mainstream SSC network architecture (a-b) and ours (c).

On the other hand, ground truth labels in the SSC task are generated by first voxelizing the stacked and aligned point cloud frames into a label volume. Voxels are then marked as either labeled or ignored based on occlusion relationships within this volume (Behley et al. 2019). To improve the label quality, as shown in Fig.1(b), a label rectification method is proposed (Xia et al. 2023) to erase motion blurs in the ground truth labels. However, existing works overlook the predictability of the completion target with respect to the sparse input. Some of the labeled voxels have no corresponding points in the sparse input, supervising with these target voxels means forcing the network to make predictions out of nothing, which forms an ill-posed situation.

Moreover, voxels marked as ignored are typically discarded in training and contribute nothing. However, being ignored does not imply that a voxel is useless. While the exact class of an ignored voxel may be uncertain, it can be reasonably assumed that it does not belong to any class absent from its local neighborhood. Otherwise, it represents an ill-posed case, lacking sufficient contextual support. This suggests that the potential of the ignored voxels remains underexplored.

Based on the analysis above, in this paper we design a parallel architecture as shown in Fig. 1(c). The sparse input is processed through parallel abstraction and dilation branches. The resulting sparse discriminative features and dense di-

*Corresponding author. †Supported by the Key Project of NSF of Zhejiang Province with number LZ26F010003.

lated features are then fused, yielding enhanced representations that combine the strengths of both paths. Specifically, we propose our Sparse-Dense network (SDNet) to process the input sparse point cloud with sparse and dense encoders in parallel, yielding two feature volumes emphasizing different aspects. Then the volumes are aligned and fused into one with the Sparse-Dense Feature Fusion (SDFF) module and further propagated by the Feature Propagation (FP) module to output the final result. Meanwhile, we propose a input-aware label refinement strategy for LiDAR SSC task including Sparse-Guided Filtering (SGF) and Ignored Voxel Recycling (IVR), which filters the unpredictable voxels in the completion target and provides additional supervision with ignored voxels, enhancing the training process of the network, thereby leading to improved final performance. Extensive experimental results on SemanticKITTI and nuScenes OpenOccupancy datasets show the effectiveness and generality of our designs. Our SDNet achieves state-of-the-art performance on both datsets and ranks 1st on SemanticKITTI SSC benchmark with 42.1 mIoU, outperforming the previous state-of-the-art method (Jang et al. 2024) by 4.2 mIoU.

In summary, our contributions are:

- We propose SDNet, a two-branch network with parallel sparse and dense encoders for enhancing feature diversity and discriminability.
- We design a Sparse-Dense Feature Fusion (SDFF) module and a Feature Propagation (FP) module to fuse and propagate the sparse and dense features, enabling complementary information integration and improved output.
- We identify limitations in conventional SSC label usage and propose an input-aware refinement strategy to address them. This includes Sparse-Guided Filtering (SGF), which removes unreliable supervision based on input visibility, and Ignored Labels Recycling (IVR), which reuses ignored voxels by leveraging neighborhood context for additional supervision.
- Extensive experiments on SemanticKITTI and nuScenes OpenOccupancy are carried out for evaluation. Our method achieved state-of-the-art results on both datasets, and ranks 1st on the SemanticKITTI SSC benchmark with 42.1 mIoU, surpassing the previous best by 4.2 (+11.1%).

Related Works

Semantic Scene Completion

Semantic Scene Completion (SSC) requires to predict a dense 3D volume of the scene with a semantic class prediction in each voxel. Compared with indoor scene where dense depth images are used as input (Song et al. 2017; Wang et al. 2024a), SSC in large-scale outdoor scene has been drawing increasing attentions. In contrast to the image-based methods (Cao and De Charette 2022; Li et al. 2023; Wang and Tong 2024), LiDAR-based methods exhibit superior performance due to the accurate depth measurement from the LiDAR sensor. LMSCNet (Roldao, De Charette, and Verroust-Blondet 2020) built a multi-scale 2D convolution network to process 3D input to maintain efficiency. JS3CNet (Yan et al. 2021) employed light weight dense blocks to process LiDAR segmentation results for SSC output. A coarse-to-fine sampling strategy is proposed in (Wang et al. 2023) to balance performance with resolution. SCPNet (Xia et al. 2023) redesigned the completion network and introduced knowledge distillation to enhance model performance. TALoS (Jang et al. 2024) addresses SSC from a test-time augmentation perspective by using confident predictions from one frame as pseudo labels to supervise another during inference. In addition, a trend of multi-modal (Pan, Wang, and Wang 2024; Li et al. 2025; Xue et al. 2025) and new representation (Huang et al. 2023, 2025; Zuo et al. 2023) methods is observed recently. Different from existing works (Yan et al. 2021; Xia et al. 2023; Wang et al. 2023) that adopt a cascaded architecture, our method processes the input through parallel branches and adaptively fuses sparse and dense features to preserve the strengths of both, resulting in significant performance gains.

Sparse Point Cloud Processing

Contemporary works in sparse 3D point cloud processing span from early point-based architectures such as (Qi et al. 2017), which introduced shared MLPs with symmetric functions for permutation invariance and hierarchical sampling for local feature extraction, to convolution-based models like (Wu, Qi, and Fuxin 2019; Thomas et al. 2019; Choy, Gwak, and Savarese 2019). These methods redefine convolution for irregular point sets by leveraging continuous kernels, kernel point interpolation, or sparse voxelized grids. More recently, transformer-based approaches including (Wu et al. 2024; Lai et al. 2023) have demonstrated strong performance by incorporating attention mechanisms that model long-range dependencies while preserving local geometric context. For LiDAR-specific data, (Zhu et al. 2021) employs cylindrical voxelization to alleviate the uneven point distribution, capturing structural details and contextual cues more effectively, leading to improved feature extraction performance.

Label Processing in SSC

Recent advances in SSC highlight the critical role of high-quality label processing in improving model performance. (Xia et al. 2023) focuses on cleaning noisy label traces caused by motion artifacts in dynamic scenes, keeping target voxels of static instances only. (Li et al. 2020) and (Wang et al. 2024b) address label imbalance by identifying and reweighting voxels that are semantically ambiguous and near boundaries, using proposed hardness metrics. (Kälble et al. 2025) introduces an evidential labeling pipeline that converts multi-frame LiDAR data into semantic occupancy maps using belief theory, explicitly modeling uncertainty and occlusion to regenerate ground truth labels with higher fidelity.

We propose our label refinement strategy from a novel perspective that the dense completion target should be predictable with respect to the sparse input points, filtering the ill-posed targets without any supportive information in the input which hinder the network learning, as well as recycling the ignored voxels to provide additional supervision to the network.

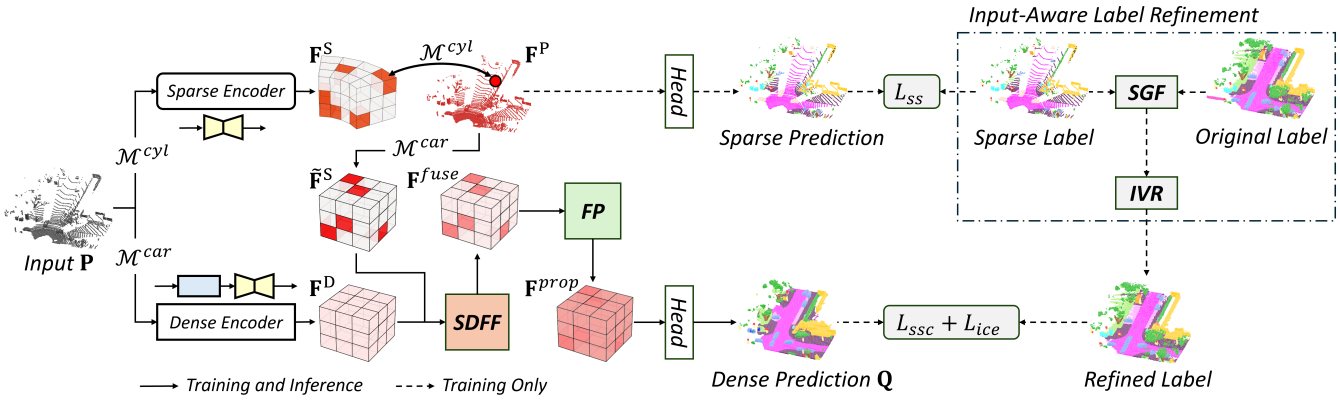

Figure 2: The architecture of our proposed method. The input $\mathbf{P}$ is inputted through two parallel encoders, with cylindrical and cartesian voxelization $\mathcal{M}^{cyl}$ and $\mathcal{M}^{car}$, respectively, yielding the sparse feature volume $\mathbf{F}^S$ and the dense feature volume $\mathbf{F}^D$. The cylindrical sparse volume is then aligned via an intermediate point feature map $\mathbf{F}^P$, obtaining the aligned sparse feature volume $\tilde{\mathbf{F}}^S$. Two complementary feature volumes $\mathbf{F}^D$ and $\tilde{\mathbf{F}}^S$ are fused and further propagated with the Sparse-Dense Feature Fusion (SDFF) module and Feature Propagation (FP) module, respectively. To supervise the training better, the original label is processed by our input-aware label refinement strategy, including Sparse-Guided Filtering (SGF) to eliminate unpredictable targets and Ignored Voxel Recycling (IVR) to leverage ignored voxels for auxiliary supervision.

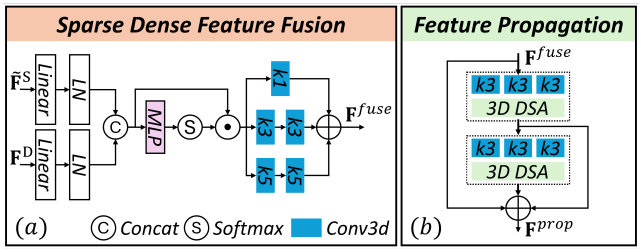

Figure 3: The architecture of the SDFF and FP module in our SDNet.

Method

The overall architecture of our method is shown in Fig. 2, it includes the Sparse-Dense network (SDNet) and the input-aware label refinement strategy. The SDNet separately processes the sparse input points with a dense and a sparse encoder first, then fuses and propagates the sparse and dense feature volumes with the Sparse-Dense Feature Fusion (SDFF) module and the Feature Propagation (FP) module, respectively, to obtain the dense prediction output. The input-aware label refinement strategy includes Sparse-Guided Filtering (SGF) to eliminate unpredictable targets and Ignored Voxel Recycling (IVR) to leverage ignored voxels for auxiliary supervision. We will elaborate our method in this chapter.

Sparse-Dense Net

Parallel feature extraction. Unlike existing works which deploy the feature dilation and abstraction modules in a cascaded manner as shown in Fig.1(a-b), we propose to process the input with parallel sparse and dense encoders, accounting for the abstraction of the sparse and the densified features, respectively, yielding two feature volumes emphasizing different aspects.

For the input point cloud $\mathbf{P}$, each point $\mathbf{p}_i$ in $\mathbf{P}$ is voxelized to its corresponding voxel indices $\mathbf{v}_i$ using the voxelization mapping $\mathcal{M}$ as:

$$\mathbf{v}_i = \mathcal{M}(\mathbf{p}_i) = \lfloor (\mathcal{T}(\mathbf{p}_i) - \mathbf{r}_{min})/\delta \rfloor, \tag{1}$$

where the $\mathcal{T}$, $\mathbf{r}_{min}$ and δ denote the coordinates transform, the minimum range of volume and the voxel resolution, respectively. Cylindrical voxelization $\mathcal{M}^{cyl}$ and cartesian voxelization $\mathcal{M}^{car}$ can be obtained with different parameter configurations accordingly.

For the object classes with distinct shape patterns, abstracting on the dilated feature volume impairs the feature discriminability. Besides, as verified in (Zhu et al. 2021), cylindrical voxelization is more effective for sparse LiDAR point cloud processing. Therefore, we voxelize the input points $\mathbf{P}$ using cylindrical voxelization $\mathcal{M}^{cyl}$ and directly pass the input volume through the sparse encoder without dilation to preserve the shape information. We employ a SpConv-based sparse U-Net (Hou et al. 2022) as the sparse encoder to extract the sparse feature volume $\mathbf{F}^S \in \mathbb{R}^{|\mathbf{P}| \times D}$, given its proven effectiveness in collecting multi-scale local and contextual information.

For background classes like building and terrain, isolated points are noisy and difficult to distinguish. Dilating them into regular local regions improves discriminability. Thus, we voxelize $\mathbf{P}$ using cartesian voxelization $\mathcal{M}^{car}$ and pass it through the dense encoder where the sparse inputs are dilated first then abstracted. We adopt the completion network proposed in (Xia et al. 2023) as the dense encoder to extract the densified feature volume $\mathbf{F}^D \in \mathbb{R}^{H \times W \times L \times D}$, given its multi-scale feature dilation ability without decreasing the voxel resolution, where H, W, L and D denote the height,

width, length and feature channel number of the feature volume, respectively.

Feature volume alignment. The sparse and dense feature volume $\mathbf{F}^S$ and $\mathbf{F}^D$ are in different coordinate systems and with different shapes. Alignment is required before fusing them. To align $\mathbf{F}^S$ with $\mathbf{F}^D$, we use the original points as intermediate. First we index the sparse feature volume $\mathbf{F}^S$ using the mapping $\mathcal{M}^{cyl}$ to acquire the corresponding feature $\mathbf{f}_i^P$ for each point $\mathbf{p}_i$ as:

$$\mathbf{f}_i^P = \mathbf{F}^S(\mathcal{M}^{cyl}(\mathbf{p}_i)). \tag{2}$$

Then we can fill all the point features $\mathbf{F}^P = \{\mathbf{f}_i^P\}$ into a new cartesian feature volume $\tilde{\mathbf{F}}^S$ using the cartesian voxelization mapping $\mathcal{M}^{car}$:

$$\tilde{\mathbf{F}}^S(\mathbf{v}_i^{car}) = \tilde{\mathbf{F}}^S(\mathcal{M}^{car}(\mathbf{p}_i)) = \mathbf{F}^P(\mathbf{p}_i). \tag{3}$$

Sparse-Dense Feature Fusion. Given two spatially aligned feature volumes of the same shape, $\tilde{\mathbf{F}}^S$ emphasizes sparse shape abstraction, while $\mathbf{F}^D$ captures dense structural information. To adaptively balance their contributions, i.e., favoring $\tilde{\mathbf{F}}^S$ for object voxels and $\mathbf{F}^D$ for background voxels, we design the Sparse-Dense Feature Fusion (SDFF) module. As shown in Fig. 3(a), the two volumes are projected via linear layer and normalized using layer normalization first. They are then concatenated and passed through a channel-wise attention mechanism that selectively focus on informative channels for each voxel. Then we use 3D convolution blocks of different kernel sizes to spatially diffuse the fused feature, increasing the influence range of the fused strong feature in each voxel. The fused feature volume $\mathbf{F}^{fuse}$ is computed as:

$$\mathbf{F}^{fuse} = \mathcal{C}_1(\tilde{\mathbf{F}}^C) + \mathcal{C}_3(\mathcal{C}_3(\tilde{\mathbf{F}}^C)) + \mathcal{C}_5(\mathcal{C}_5(\tilde{\mathbf{F}}^C)), \tag{4}$$

with

$$\tilde{\mathbf{F}}^C = Softmax(MLP(\mathbf{F}^C)) \cdot \mathbf{F}^C, \tag{5}$$

and

$$\mathbf{F}^C = Cat[LN(\mathcal{L}^D(\mathbf{F}^D)), LN(\mathcal{L}^S(\tilde{\mathbf{F}}^S))], \tag{6}$$

where $\mathcal{C}_k$, $Cat[:]$, LN and $\mathcal{L}$ denote 3D convolution block with kernel size k, concatenation, layer normalization and linear layer, respectively.

Feature Propagation. Due to the limited perceptive field of the conv-based operators, the feature volume $\mathbf{F}^{fuse}$ is locally discriminative but not completed enough. As shown in Fig.3(b), we adopt our FP module which is composed of two layers of multi-scale 3D convolution and deformable self attention to further propagate the features in the whole feature volume. The propagated feature $\mathbf{F}^{prop}$ is computed as:

$$\mathbf{F}^{prop} = \mathbf{F}^{fuse} + \mathcal{P}(\mathbf{F}^{fuse}) + \mathcal{P}(\mathcal{P}(\mathbf{F}^{fuse})), \tag{7}$$

with

$$\mathcal{P}(\mathbf{f}_i) = DSA(\mathcal{C}_3(\mathbf{f}_i) + \mathcal{C}_5(\mathbf{f}_i) + \mathcal{C}_7(\mathbf{f}_i)), \tag{8}$$

and

$$DSA(\mathbf{f}_i) = \sum_{j=0}^{K-1} a_j w_j \mathbf{F}(\mathbf{v}_i + \delta_j), \tag{9}$$

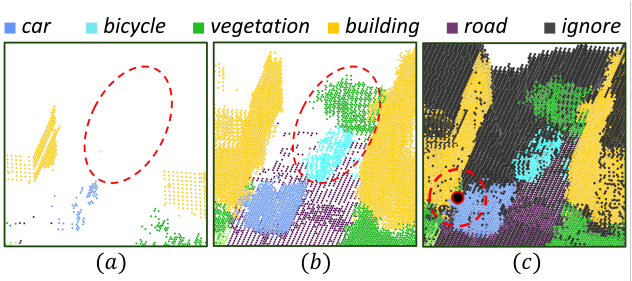

Figure 4: An example to illustrate the problems existing in current SSC labels. (a) The input sparse points and labels. (b) The labeled dense voxels serving as the completion target, with ignored voxels visualized in (c).

where the K, a_j, w_j, $\mathbf{v}_i$ and δ_j denote the sampling location number, attention weight, feature projection weight, center point location and sampling location offset in the DSA mechanism, respectively. With the FP module, the fused strong features are further propagated, yielding a better-completed discriminative output feature volume $\mathbf{F}^{prop}$. By passing $\mathbf{F}^{prop}$ through a classification head, the final prediction output $\mathbf{Q}$ can be obtained.

Input-Aware Label Refinement

In SSC, ground truth labels are generated by voxelizing stacked point cloud frames into a label volume $\mathbf{Y}^{gt} \in \mathbb{N}^{*H\times W\times L\times 1}$ and applying ray tracing to partition it into empty ($\mathbf{Y}^{emp}$), labeled ($\mathbf{Y}^{lab}$), and ignored ($\mathbf{Y}^{ign}$) voxels (Behley et al. 2019). While empty voxels are reliably identified, the treatment of $\mathbf{Y}^{lab}$ and $\mathbf{Y}^{ign}$ may introduce issues. Voxels observable by the LiDAR in current or future frames are marked as labeled and used for the main completion target, whereas permanently unobservable regions (e.g., behind walls) are marked as ignored and excluded during training.

The SSC network takes sparse point cloud as input and predicts a dense semantic label volume, where each prediction inherently depends on the sparse input. However, the original ground truth voxels are marked as labeled $\mathbf{Y}^{lab}$ or ignored $\mathbf{Y}^{ign}$ solely based on occlusion analysis in the target volume, omitting to account for this input-output dependency. Take the data shown in Fig. 4(a) as an example, input points from classes such as car, building and road justify corresponding dense labels in (b). In contrast, labeled voxels from classes like bicycle and vegetation, which lack supporting input points, impose ill-posed supervision by requiring predictions from no observable evidence. On the other hand, as shown in Fig. 4(c), an ignored voxel is surrounded by voxels of car, building and vegetation. It is illogical to predict a non-neighboring class for this voxel, since there is no observable evidence in the input for absent classes. However, such information is omitted in existing methods. To solve these problems, we propose our input-aware label refinement strategy, including Sparse-Guided Filtering (SGF) and Ignored Voxel Recycling (IVR), which will be elaborated below.

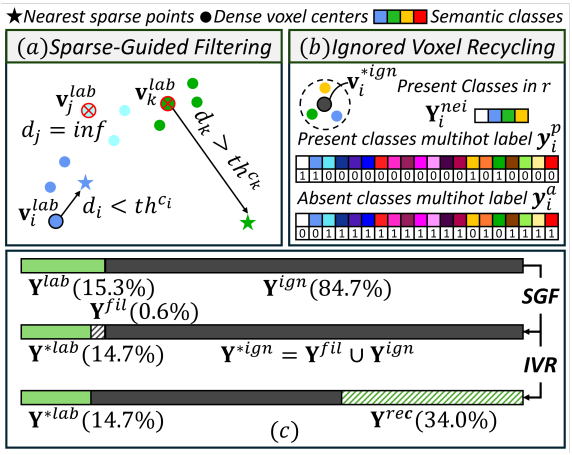

Figure 5: The illustration of our SGF and IVR methods.

Sparse-Guided Filtering. The SGF method filters unpredictable labeled voxels that lack supporting information from input to prevent them from imposing ill-posed supervision on the network, enabling more effective and reliable learning.

We take the voxel center coordinates $\mathbf{V}^{lab}$ of the labeled voxels $\mathbf{Y}^{lab}$. For each point $\mathbf{v}_i^{lab}$ in $\mathbf{V}^{lab}$ with its label $y_i^{lab} = c_i$, it should have at least one point of the same class c_i in the sparse input $\mathbf{P}$ within a class-specific range threshold th^{c_i}. In other words, the nearest distance d_i between $\mathbf{v}_i^{lab}$ and the sparse input of the same class $\mathbf{P}^{c_i}$ should be lower than th^{c_i}, otherwise this label voxel becomes an ill-posed target, since the network can not predict out of nothing. As illustrated in Fig.5(a), we filter the ill-posed target voxels by setting them to $ignore$:

$$y_i = \begin{cases} c_i & d_i \le th^{c_i} \\ ignore & else \end{cases} \tag{10}$$

with

$$d_i = dist(NN(\mathbf{v}_i^{lab}, \mathbf{P}^{c_i})), \tag{11}$$

where NN and $dist$ denote nearest neighbor querying and distance computing, respectively.

The class-specific threshold **th** is computed over the training set, using the average instance size for object classes (e.g., car, bicycle) and the 90th percentile of nearest distances d_i for background classes (e.g., road, terrain). It defines the predictable spatial extent of a class based on its presence in the input. Target points exceed this threshold lack visibility evidence from input, resulting in ill-posed supervision and should be filtered. The average ratios of the original labeled voxels $\mathbf{Y}^{lab}$, filtered voxels $\mathbf{Y}^{fil}$ and refined voxels $\mathbf{Y}^{*lab} = \{y^{*lab}\} = \{y_i \mid y_i \neq ignore\}$ after the SGF on the training set are shown in Fig.5(c).

Ignored Voxel Recycling. For an ignored voxel, although its true class is uncertain, it can be constrained not to belong to classes absent from its local neighborhood. Otherwise, it represents ill-posed targets lacking contextual support. Leveraging this insight, we propose the IVR method, which selectively reuses ignored voxels instead of discarding them totally.

As illustrated in Fig.5(b), for a point $\mathbf{v}_i^{*ign}$ in the ignored voxel centers $\mathbf{V}^{*ign}$, we collect the present classes of its neighboring points within r in $\mathbf{V}^{*lab}$:

$$\mathbf{Y}_i^{nei} = unique(\{y_j^{*lab} \mid dist(\mathbf{v}_i^{*ign}, \mathbf{v}_j^{*lab}) \le r\}), \tag{12}$$

where $unique$ denotes the unique operation to remove duplicates of the same class. Note that voxel i is passed if no class other than *ignore* exists within r. Then we calculate the multi-hot label present in the neighborhood of the point $\mathbf{v}_i^{*ign}$:

$$\mathbf{y}_i^p = \sum_{y_j \in \mathbf{Y}_i^{nei}} onehot(y_j). \tag{13}$$

The multi-hot label for absent classes in the neighborhood can be obtained simply by:

$$\mathbf{y}_i^a = \mathbf{1} - \mathbf{y}_i^p. \tag{14}$$

The average ratio of the recycled ignore voxels $\mathbf{Y}^{rec} = \{\mathbf{y}_i^a\}$ on the training set is shown in Fig.5(c). With theses recycled voxel, we can compute an inverse cross-entropy loss L_{ice} as auxiliary supervision. The original cross-entropy loss pushes the predicted distribution to a certain class. On the contrary, the true class of an ignored voxel is uncertain, we can only confidently suppress the output probabilities of the absent classes by pushing them towards zero, i.e., maximizing $\mathbf{1} - \mathbf{q}_i$. Therefore, the L_{ice} is computed as:

$$L_{ice} = -\sum_{i \in |\mathbf{Y}^{rec}|} \boldsymbol{w} \cdot \mathbf{y}_i^a log(\mathbf{1} - \mathbf{q}_i), \tag{15}$$

with $\mathbf{q}_i$ and $\boldsymbol{w}$ denoting the predicted $softmax$ logit of voxel i and the class-wise weights, respectively.

Losses

The overall loss is consisted of the conventional SSC loss L_{ssc}, our inverse CE loss L_{ice} and the point semantic segmentation loss L_{ss}:

$$L = L_{ssc} + L_{ice} + L_{ss}, \tag{16}$$

with

$$L_{ssc} = L_{wce} + L_{lov} + L_{scale}, \tag{17}$$

and

$$L_{ss} = L_{wce} + L_{lov}, \tag{18}$$

where L_{wce}, L_{lov} and L_{scale} denote the weighted CE loss, the lovasz-softmax loss and the scale loss (Cao and De Charette 2022) respectively. Note that the L_{ss} supervises the training of the sparse encoder, and the sparse label is a subset of the original dense label, requiring no extra annotation.

Experiments

Dataset and Metrics

We conduct experiments on the popular SSC datasets SemanticKITTI and nuScenes OpenOccupancy for evaluation. SemanticKITTI has 22 sequences including 8550

Methods	Input	mIoU	IoU	car	bicycle	motor.	truck	oth. veh.	person	bicyclist	mot. cycl.	road	parking	sidewalk	oth. gnd.	building	fence	vege.	trunk	terrain	pole	traf. sign
CGFormer (Yu et al. 2024)	C	16.6	44.4	26.1	3.7	1.3	4.3	2.7	1.7	3.6	0.4	64.3	34.1	34.2	12.1	25.8	18.7	24.5	11.2	29.3	8.7	9.3
Co-Occ (Pan, Wang, and Wang 2024)	C+L	24.4	-	40.0	4.4	3.3	6.4	8.8	1.6	3.3	0.4	72.0	42.5	43.5	10.2	35.1	32.7	41.2	30.8	40.8	26.6	20.7
SSA-SC (Yang et al. 2021)	L	23.5	58.8	36.5	13.9	4.6	5.7	7.4	4.4	2.6	0.7	72.2	37.4	43.7	10.9	43.6	30.7	43.5	25.6	41.8	14.5	6.9
JS3CNet (Yan et al. 2021)	L	23.8	56.6	33.3	14.4	8.8	7.2	12.7	8.0	5.1	0.4	64.7	34.9	39.9	14.1	39.4	30.4	43.1	19.6	40.5	18.9	15.9
SCPNet* (Xia et al. 2023)	L	36.7	56.1	46.4	33.2	34.9	13.8	29.1	28.2	24.7	1.8	68.5	51.3	49.8	30.7	38.8	44.7	46.4	40.1	48.7	40.4	25.1
TALOS* (Jang et al. 2024)	L	37.9	60.2	46.4	34.4	36.9	14.0	30.0	30.5	27.3	2.2	73.0	51.3	53.6	28.4	40.8	45.1	50.6	38.8	51.0	**40.7**	24.4
SDNet* (Ours)	L	**42.1**	**66.7**	**51.8**	**37.5**	**38.9**	**22.8**	**36.8**	**31.8**	**32.8**	**3.0**	**78.8**	**56.6**	**56.9**	**33.5**	**48.8**	**45.8**	**53.3**	**40.2**	**54.3**	40.4	**35.8**

Table 1: The quantitative result on SemanticKITTI test set. * with pretrained backbone. C and L denote camera and LiDAR, respectively. The best and second-best are in bold and underlined, respectively.

Methods	Input	mIoU	IoU	barrier	bicycle	bus	car	cons. veh.	motor.	pede.	traf. cone	trailer	truck	driveable	other flat	sidewalk	terrain	manmade	vege.
TPVFormer (Huang et al. 2023)	C	7.8	15.3	9.3	4.1	11.3	10.1	2.5	4.3	5.9	5.3	6.8	6.5	13.6	9.0	8.3	8.0	9.2	8.2
OccMamba (Li et al. 2025)	C+L	26.2	35.7	30.2	20.5	26.5	29.5	18.8	26.0	23.7	19.9	20.6	25.4	38.4	26.5	27.0	26.6	28.9	30.5
PVP (Xue et al. 2025)	C+L	28.0	36.3	32.2	24.5	26.7	31.8	16.4	29.9	34.9	21.9	21.5	26.5	40.3	27.5	28.6	26.9	28.5	30.4
JS3CNet (Yan et al. 2021)	L	12.5	30.2	14.2	3.4	13.6	12.0	7.2	4.3	7.3	6.8	9.2	9.1	27.9	15.3	14.9	16.2	14.0	24.9
PointOcc (Zuo et al. 2023)	L	23.9	34.1	24.9	**19.0**	20.9	25.7	13.4	**25.6**	**30.6**	**17.9**	16.7	21.2	36.5	25.6	25.7	24.9	24.8	29.0
OccMamba (Li et al. 2025)	L	22.7	**36.4**	26.8	11.3	20.8	26.1	14.6	20.3	14.0	14.0	17.5	20.3	**39.5**	24.9	25.9	25.3	28.3	30.6
SDNet (Ours)	L	**25.8**	36.0	**28.3**	18.7	**25.6**	**29.6**	**14.6**	24.9	29.9	16.1	**20.8**	**24.6**	39.3	**27.6**	**27.5**	**26.4**	**28.9**	**30.7**

Table 2: The quantitative result on nuScenes OpenOccupancy validation set. C and L denote camera and LiDAR, respectively. The best and second-best are in bold and underlined, respectively.

frames. Among them, 3834 frames in sequence 00 to 10 except for 08 are used for training, 815 frames in sequence 08 and 3901 frames in sequence 11 to 21 are use for validation and testing, respectively. The classification target includes 19 classes. The minimum, maximum range and the target shape of the scene is $[0.0m, -25.6m, -2.0m]$, $[51.2m, 25.6m, 4.4m]$ and $[256, 256, 32]$, respectively. nuScenes OpenOccupancy has 850 sequences including 28130 frames for training and 6019 frames for validation. The classification target includes 16 classes. The minimum, maximum range and the target shape of the scene is $[-51.2m, -51.2m, -5.0m]$, $[51.2m, 51.2m, 3.0m]$ and $[512, 512, 40]$, respectively. We report the IoU and the mIoU as the evaluation metrics, note that the mIoU is the major metric for the SSC task.

Implementation Details

The dense feature volume shape for SemanticKITTI and OpenOccupancy is set as $[256, 256, 32]$ and $[128, 128, 10]$, respectively, and the corresponding voxel sizes δ^{car} and δ^{cyl} for voxelization are set as $0.2m, [0.11m, 1.0°, 0.20m]$ and $0.8m, [0.11m, 1.0°, 0.25m]$, respectively. The sampling location number in DSA layer is set as $K = 8$. For the SGF, the class-specific threshold **th** is determined as described in SGF section. For the IVR, the neighborhood radius r is set to 0.8m.

To train the model for the SemanticKITTI benchmark, we first pretrain the backbone encoders as illustrated in (Xia et al. 2023), then we use AdamW optimizer with learning rate 2e-4, weight decay 1e-2 and cosine annealing scheduler to train the network with overall loss L on the training set for 20 epochs, with 4 Nvidia RTX 4090 GPUs and the total batch size set to 4. For the training of the rest experiments, we directly train the whole network.

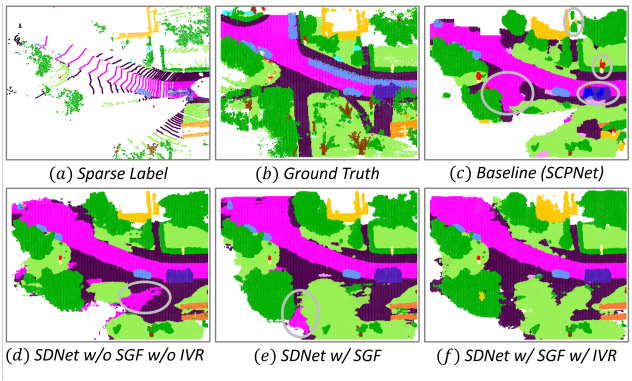

Figure 6: Qualitative results. (a) The sparse labels. (b) The dense ground truth labels. (c-f) The improved prediction results with our designs added progressively. The erroneous areas in the predictions are highlighted with gray circles, from which we can observe the improvements more clearly.

Main Results

The quantitative results on SemanticKITTI and nuScenes OpenOccupancy datasets are shown in Table 1 and Table 2, respectively. Compared with the recent methods (Jang et al. 2024; Xia et al. 2023; Yan et al. 2021; Zuo et al. 2023; Li et al. 2025), our method achieved the state-of-the-art performance on both datasets. Notably, in contrast to the previous SOTA method (Jang et al. 2024) on SemanticKITTI, the mIoU and IoU metrics are significantly increased by 4.2 (+11.1%) and 6.5 (+10.8%) with our method, respectively. For classes such as truck, other-vehicle, building and traffic-sign, the performances are improved by 62.9%, 22.7%, 19.6% and 46.7%, respectively. This verified the superiority, effectiveness and generality of our proposed method.

#	$\mathbf{F}^D$	$\mathbf{F}^S$	SDFF	FP	SGF	IVR	mIoU	IoU
(a)	o	x	x	x	x	x	22.7	57.6
(b)	x	o	x	x	x	x	17.3	32.1
(c)	o	o	o	x	x	x	27.8	58.6
(d)	o	o	o	o	x	x	28.5	59.3
(e)	o	o	o	o	o	x	29.0	60.0
(f)	o	o	o	o	x	o	28.7	59.3
(g)	o	o	o	o	o	o	29.4	60.0
(h)	o*	o	o	o	x	x	40.7	61.0
(i)	o*	o	o	o	o	o	42.5	61.5

Table 3: Ablation on effect of each design. The experiments are conducted on the SemanticKITTI validation set. $\mathbf{F}^D$: Dense feature volume. $\mathbf{F}^S$: Sparse featrue volume. * with pretrained backbone.

			(a)					(b)	
#	Cas	Para	Car	Cyl	mIoU	IoU	Models	mIoU	IoU
(i)	o	x	o	x	22.7	57.6	SSA-SC	24.3	58.1
(ii)	x	o	o	x	23.6	58.2	w/ Ours	25.4	57.9
(iii)	o	x	x	o	26.4	55.2	JS3CNet	23.2	55.5
(iv)	x	o	x	o	27.8	58.6	w/ Ours	24.6	55.7

Table 4: Ablation on (a) parallel architecture and cylindrical voxelization, (b) generality of our input-aware label refinement strategy. Cas: cascaded. Para: parallel. Car: cartesian voxelization. Cyl: cylindrical voxelization.

Fig.6 shows the qualitative results of our method on SemanticKITTI. Comparing (c) and (d), we can observe a significant part of erroneous areas such as the truck, the trafficsign and the road are corrected, thanks to the discriminative sparse features adaptively fused by our SDFF module. Comparing (a), (d) and (e), we can observe that the misclassified area in (d) corresponds to the empty area in the sparse input. With our SGF to filter such ill-posed voxels during the training, the network was trained better to predict the correct result (e). Comparing (e) and (f), with our IVR to suppress the emergence of the absent classes, false positive prediction of road class was further corrected (f), yielding the final output of our full method.

Ablation Studies

All the ablation experiments are conducted on the validation set of the SemanticKITTI dataset if not noted particularly.

Effect of each design. Table 3 shows the effect of each of our designs. Comparing (a), (b) and (c), we verified that the baseline (Xia et al. 2023) (a) can be effectively improved by fusing the sparse features in parallel with our SDFF module. In (d) we can observe the FP further completed the feature volume and improved the results by 0.7 mIoU and IoU. Comparing (e), (f) and (g) we can observe the effectiveness of our SGF and IVR methods, improving the mIoU metric by 0.5 and 0.2 when used alone and by 0.9 when jointly used. The performance of the model for the benchmark is also showed in (h) and (i), we can observe a higher improvement (+1.8 mIoU) brought by our label refinement strategy when applied on a stronger backbone. Note that the pretrained backbone (SCPNet) inputs future frames to form KD teacher and trains current-only student. Since SSC is largely a future predict task, this helps a lot. As in Table 1, we can still improve the pretrained model from 36.7 to 42.1, verifying our effectiveness.

			(a)				(b)		
Methods	add	cat	CA	CWA	Methods	GFLOPs	Param	mIoU
mIoU	27.4	27.4	27.5	27.8	SCPNet	5395	52M	36.7
IoU	57.0	58.3	57.9	58.6	Ours	6764	69M	42.1

Table 5: Ablation on (a) alternative fusion methods in SDFF, (b) computational efficiency. add: addition. cat: concatenation. CA: cross attention. CWA: channel-wise attention.

The effectiveness of our parallel architecture and the use of cylindrical voxelization for sparse feature extraction is demonstrated in Table 4(a). A comparison between (i-ii) and (iii-iv) confirms the superiority of cylindrical voxelization in enhancing semantic abstraction by capturing distinct sparse geometries. Additionally, comparisons between (i) and (ii), as well as (iii) and (iv), validate the improvement achieved through the parallel structure.

Generality of the label refinement strategy. We applied our label refinement strategy to other existing SSC models, i.e., SSA-SC (Yang et al. 2021) and JS3CNet (Yan et al. 2021), to verify the generality of it. The results are shown in Table 4(b). With our input-aware label refinement strategy, the mIoU metric was improved by 1.1 and 1.4 for two models, respectively, verifying the generality of our method.

Alternative fusion methods in SDFF We explored alternative designs to fuse the features in the SDFF other than channel-wise attention, and the results are show in Table 5(a). Due to the inherent disparity between sparse and dense features, direct addition or concatenation tends to obscure their distinctions. Moreover, similarity-based fusion like cross-attention struggle to associate weak or empty voxels with strong features, resulting in suboptimal performance. Compared with others, CWA enables to adaptively emphasize the most informative feature channels while suppressing less relevant ones for each fused voxel, yielding a better result.

Computational efficiency. The computational efficiency result is shown in Table 5(b), we made an effective balance between precision and efficiency.

Conclusion

In this paper, we proposed Sparse-Dense Net (SDNet) for LiDAR-based Semantic Scene Completion task that processes input point cloud through parallel sparse and dense encoders, then fuses and refines the complementary intermediate features via Sparse-Dense Feature Fusion (SDFF) module and Feature Propagation (FP) module. To address limitations in label supervision, we introduced a input-aware label refinement strategy, including Sparse-Guided Filtering (SGF) to remove ill-posed targets and Ignored Voxel Recycling (IVR) to leverage ignored voxels as auxiliary supervision. Extensive experiments on two popular datasets show that SDNet achieves state-of-the-art performance, demonstrating the effectiveness and generality of our designs.

Acknowledgments

This work was supported by the Joint R&D Program of the Yangtze River Delta Community of Sci-Tech Innovation with grant number 2024CSJGG01000, and the Key Research & Development Plan of Zhejiang Province under Grant No.2024C01010, 2024C01017, 2025C01039.

References

Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; and Gall, J. 2019. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9297–9307.

Cao, A.-Q.; and De Charette, R. 2022. Monoscene: Monocular 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3991–4001.

Choy, C.; Gwak, J.; and Savarese, S. 2019. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3075–3084.

Hou, Y.; Zhu, X.; Ma, Y.; Loy, C. C.; and Li, Y. 2022. Point-to-voxel knowledge distillation for lidar semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8479–8488.

Huang, Y.; Thammatadatrakoon, A.; Zheng, W.; Zhang, Y.; Du, D.; and Lu, J. 2025. Gaussianformer-2: Probabilistic gaussian superposition for efficient 3d occupancy prediction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 27477–27486.

Huang, Y.; Zheng, W.; Zhang, Y.; Zhou, J.; and Lu, J. 2023. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9223–9232.

Jang, H.-K.; Kim, J.; Kweon, H.; and Yoon, K.-J. 2024. TALoS: Enhancing semantic scene completion via test-time adaptation on the line of sight. *Advances in Neural Information Processing Systems*, 37: 74211–74232.

Kälble, J.; Wirges, S.; Tatarchenko, M.; and Ilg, E. 2025. EvOcc: Accurate Semantic Occupancy for Automated Driving Using Evidence Theory. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 27467–27476.

Lai, X.; Chen, Y.; Lu, F.; Liu, J.; and Jia, J. 2023. Spherical transformer for lidar-based 3d recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 17545–17555.

Li, H.; Hou, Y.; Xing, X.; Ma, Y.; Sun, X.; and Zhang, Y. 2025. Occmamba: Semantic occupancy prediction with state space models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 11949–11959.

Li, J.; Liu, Y.; Yuan, X.; Zhao, C.; Siegwart, R.; Reid, I.; and Cadena, C. 2020. Depth Based Semantic Scene Completion With Position Importance Aware Loss. *IEEE Robotics and Automation Letters*, 5(1): 219–226.

Li, Y.; Yu, Z.; Choy, C.; Xiao, C.; Alvarez, J. M.; Fidler, S.; Feng, C.; and Anandkumar, A. 2023. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9087–9098.

Pan, J.; Wang, Z.; and Wang, L. 2024. Co-occ: Coupling explicit feature fusion with volume rendering regularization for multi-modal 3d semantic occupancy prediction. *IEEE Robotics and Automation Letters*, 9(6): 5687–5694.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Roldao, L.; De Charette, R.; and Verroust-Blondet, A. 2020. Lmscnet: Lightweight multiscale 3d semantic completion. In *2020 International Conference on 3D Vision (3DV)*, 111–119. IEEE.

Song, S.; Yu, F.; Zeng, A.; Chang, A. X.; Savva, M.; and Funkhouser, T. 2017. Semantic scene completion from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1746–1754.

Thomas, H.; Qi, C. R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; and Guibas, L. J. 2019. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6411–6420.

Wang, F.; Sun, Q.; Zhang, D.; and Tang, J. 2024a. Unleashing network potentials for semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10314–10323.

Wang, S.; Yu, J.; Li, W.; Liu, W.; Liu, X.; Chen, J.; and Zhu, J. 2024b. Not all voxels are equal: Hardness-aware semantic scene completion with self-distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14792–14801.

Wang, X.; Zhu, Z.; Xu, W.; Zhang, Y.; Wei, Y.; Chi, X.; Ye, Y.; Du, D.; Lu, J.; and Wang, X. 2023. Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 17850–17859.

Wang, Y.; and Tong, C. 2024. H2gformer: Horizontal-to-global voxel transformer for 3d semantic scene completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 5722–5730.

Wu, W.; Qi, Z.; and Fuxin, L. 2019. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 9621–9630.

Wu, X.; Jiang, L.; Wang, P.-S.; Liu, Z.; Liu, X.; Qiao, Y.; Ouyang, W.; He, T.; and Zhao, H. 2024. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4840–4851.

Xia, Z.; Liu, Y.; Li, X.; Zhu, X.; Ma, Y.; Li, Y.; Hou, Y.; and Qiao, Y. 2023. Scpnet: Semantic scene completion on point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 17642–17651.

Xue, Y.; Liu, J.; Du, J.; and Zhou, J. T. 2025. PVP: Polar Representation Boost for 3D Semantic Occupancy Prediction. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2746–2755. IEEE.

Yan, X.; Gao, J.; Li, J.; Zhang, R.; Li, Z.; Huang, R.; and Cui, S. 2021. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 3101–3109.

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

Yu, Z.; Zhang, R.; Ying, J.; Yu, J.; Hu, X.; Luo, L.; Cao, S.-Y.; and Shen, H.-L. 2024. Context and geometry aware voxel transformer for semantic scene completion. *Advances in Neural Information Processing Systems*, 37: 1531–1555.

Zhu, X.; Zhou, H.; Wang, T.; Hong, F.; Ma, Y.; Li, W.; Li, H.; and Lin, D. 2021. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9939–9948.

Zuo, S.; Zheng, W.; Huang, Y.; Zhou, J.; and Lu, J. 2023. Pointocc: Cylindrical tri-perspective view for point-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2308.16896*.