MirageRoom: 3D Scene Segmentation with 2D Pre-trained Models by Mirage Projection

Haowen Sun, Yueqi Duan[*], Juncheng Yan, Yifan Liu, Jiwen Lu
Tsinghua University

Abstract

Nowadays, leveraging 2D images and pre-trained models to guide 3D point cloud feature representation has shown a remarkable potential to boost the performance of 3D fundamental models. While some works rely on additional data such as 2D real-world images and their corresponding camera poses, recent studies target at using point cloud exclusively by designing 3D-to-2D projection. However, in the indoor scene scenario, existing 3D-to-2D projection strategies suffer from severe occlusions and incoherence, which fail to contain sufficient information for fine-grained point cloud segmentation task. In this paper, we argue that the crux of the matter resides in the basic premise of existing projection strategies that the medium is homogeneous, thereby projection rays propagate along straight lines and behind objects are occluded by front ones. Inspired by the phenomenon of mirage where the occluded objects are exposed by distorted light rays due to heterogeneous medium refraction rate, we propose ***MirageRoom*** *by designing parametric mirage projection with heterogeneous medium to obtain series of projected images with various distorted degrees. We further develop a masked reprojection module across 2D and 3D latent space to bridge the gap between pre-trained 2D backbone and 3D point-wise features. Both quantitative and qualitative experimental results on S3DIS and ScanNet V2 demonstrate the effectiveness of our method.* [1]

1. Introduction

Understanding indoor scene-level point clouds has become a fundamental and crucial task for various applications, including robotics [30], augmented reality [1] and virtual reality [43]. Abundant methods [14, 20, 32, 33, 38, 45, 50] have investigated point cloud architectures to fully explore the potential of interactions between points. While these methods have achieved outstanding performance in various scene understanding tasks, a major bottleneck that hinders the researchers from moving forward is the lack of high-quality annotated 3D scene data. Compared to 2D image datasets [10], the scale of real-world datasets for 3D indoor point cloud understanding is much smaller [3, 9]. Thus, taking advantage from vast 2D image data and models to guide 3D tasks has become a direct and trending approach to compensate the lack of 3D data.

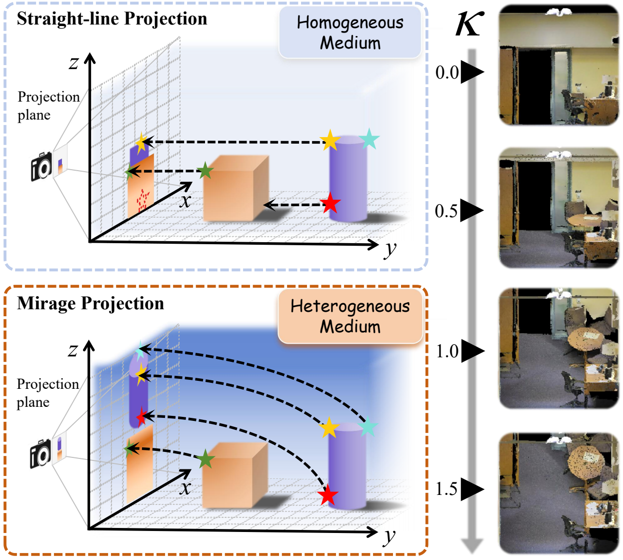

Figure 1. Comparison of straight-line projection under homogeneous medium and our mirage projection under heterogeneous medium. In straight-line projection, rays propagate along straight lines and objects behind are occluded (red and blue stars). In contrast, our mirage projection adopts a parameter κ to modify the distribution of medium where projection rays are distorted, thereby previously occluded points can be exposed. With series of κ, we can increase the occupancy of points in projection images. Best viewed in color.

Recently, many studies have explored methods for understanding 3D point cloud based on 2D images and models. Some prevailing approaches aim to combine both 2D and 3D architectures based on additional data, such as high-quality real-world 2D images and their accurate corresponding camera poses [8, 16, 17, 41], which pose high

[*] Corresponding author.
[1] Code will be available here.

requirement for point cloud datasets. More recently, another branch strives for investigating 3D-to-2D projection from 3D points to 2D plane to obtain 2D features [2, 29, 48, 49], and has been applied on object-level and LIDAR point clouds. However, as shown in Figure 1, the straight-line projection suffers from severe semantic occlusion and incoherence especially in indoor scenes where objects are densely distributed, leading to insufficient information in projection images for point-level feature learning. Despite changing view-points mitigates the problem to a certain degree on object-level point clouds [15], it is still hard to develop a unified algorithm which generates appropriate viewpoints for variable indoor scenes.

In this paper, we argue that the key factor leading to occlusion is the basic premise of homogeneous medium, thereby projection rays propagate along straight lines and behind objects are occluded by front ones. Inspired by mirage phenomenon where occluded objects are exposed by distorted rays due to heterogeneous medium, we propose mirage projection strategy so that the originally occluded points can be exposed through distorted rays. As is shown in Figure 1, we model the 3D space with heterogeneous medium, where we design a parameter κ to modify the distribution of medium to consequently change the curvature of projection rays. By selecting series of κ, projection images are able to cover more points, which increases the occupancy of points. Since straight-line projection is a degenerated case for mirage projection when $\kappa = 0$, our strategy provides a more general form of projection.

Based on the projection strategy, we further develop a point cloud architecture with 2D pre-trained models for segmentation task, named **MirageRoom**. Specifically, we first adopt group mirage projection over input point clouds, where we employ a group of κs to generate multi-view projections to provide comprehensive information for 2D models. Then, a frozen pre-trained 2D image backbone is followed to extract 2D features for transferring 2D knowledge to enhance the semantic representing ability of 3D point-wise features. In order to bridge the gap between 2D and 3D latent space in a precise way, we introduce a masked reprojection module to mix accurate 2D features for each point. We further construct a U-Net liked network for indoor segmentation task.

To test the effectiveness of our method, we conducted extensive experiments on two popular indoor point cloud benchmarks, *i.e*., S3DIS [3] and ScanNet V2 [9]. We find that our MirageRoom can truly benefit from 2D pre-trained models with the help of mirage projection, which achieves higher performance on S3DIS than state-of-the-art methods [33, 45] with much fewer learnable parameters. Meanwhile, our method outperforms other 2D-3D methods without any additional real-world images on ScanNet V2 dataset. Further experiments verify that our design is crucial for the improvement of performance.

2. Related Work

Point Cloud Segmentation. 3D scene segmentation task has become a core task for scene-level point cloud understanding. Point cloud segmentation problems are typically solved via two streams: voxel-based and point-based. Voxel-based methods [7, 12, 13, 21, 51] first divide 3D spaces into regular cubics to sparsely voxelize the point clouds, and then perform sparse operations over voxels. Although voxel-based methods are highly efficient, most of their performance are limited due to the positional geometry inaccuracy introduced by the sparsely voxelization process. As a result, voxel-based methods are mostly adopted in outdoor sparse point cloud segmentation. Meanwhile, Point-based methods [8, 11, 14, 18, 20, 22, 24, 32, 33, 38, 45, 50] take the dominant position in understanding indoor point cloud segmentation based on various structures, such as MLP-based [31–33], convolution-based [24, 38], graph-based [22, 39] and transformer-based [14, 20, 45, 50]. Though effective, the limited scale of 3D dataset constrains the further development of these methods.

2D Guided Point Cloud Understanding. Concurrent to point-based and voxel-based methods, some other works aim to understand point cloud with the help of plentiful 2D networks. Some of these works combine both 2D and 3D architectures based on additional data, such as real-world images and their corresponding camera poses [8, 16, 17, 41], which pose high requirement for dataset. Another branch strives in investigating 3D-to-2D projection from 3D points to 2D plane to obtain 2D features, *i.e*., projection-based methods [2, 15, 19, 29, 35, 42, 49]. MVCNN [35] and MVTN [15] employ multi-view projection to generate 2D maps of objects, with multi-view 2D networks followed to extract features. However, it is hard to develop a unified algorithm which generates appropriate multi-view points for scenes, which limits their applications in object-level tasks. The range projection achieves great success in LIDAR point cloud segmentation [2, 19, 29], while it still fails to process indoor point cloud due to severe occlusions caused by densely distributed points.

3. Proposed Method

In this section, we first introduce mirage projection to handle the occlusion and incoherence of conventional 3D-to-2D straight-line projection (Section 3.1). Then we develop a masked reprojection module to bridge the gap between 2D and 3D domain (Section 3.2). The whole network architecture, *i.e*., MirageRoom, is constructed based on our mirage projection (Section 3.3).

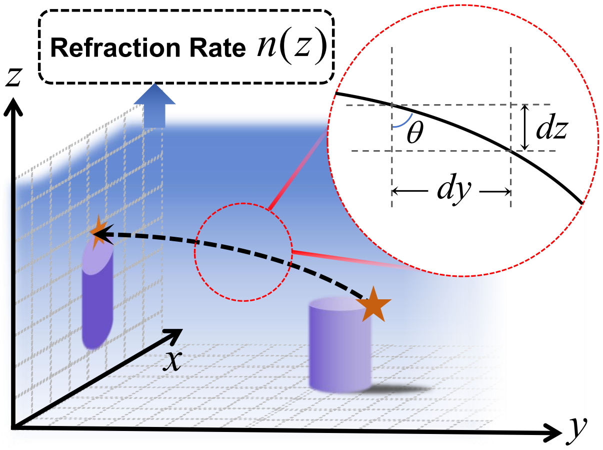

Figure 2. Physical model of our mirage projection. The heterogeneous medium refraction rate has a distribution of $n(z)$, which leads to the distorted projection ray. The curve function of the projection ray on $y - z$ plane can be formulated according to physical principles.

3.1. Mirage Projection

Review of Straight-line Projection. The conventional straight-line projection simply squeezes the points along axes [48, 49]. For each point $p = (x_p, y_p, z_p) \in \mathbb{R}^3$, the projection strategy omits one of the three axes and interpolates the projected position under image coordinates from the other two axes. Taking projection view which is parallel with $x - z$ plane as an example, the projected position of p can be formulated as:

$$H_p = \frac{z_p - z_{\text{min}}}{z_{\text{max}} - z_{\text{min}}} H,\ W_p = \frac{x_p - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} W, \tag{1}$$

where H and W denotes the size of projected 2D image. Simple as it is, there exists two major problems. On one hand, the behind objects are permanently occluded by front objects along the squeezed axis, which hinders the projected images from covering sufficient points to generate effective point-wise features. On the other hand, the simple scatter-style figures have clear empty girds due to the sparsity of point cloud, which makes it incoherent and different from the real-world images for 2D pre-training process.

Physical Modeling. To tackle the occlusion caused by straight-line projection, we refer to mirage phenomenon where occluded objects can be viewed through distorted rays. As is illustrated in Figure 2, this phenomenon arises with the heterogeneous distribution of medium refraction rate which can be formulated as [4, 5]:

$$n(z) = 1 + \rho_0 e^{-kz}, \tag{2}$$

where $n(z)$ denotes the refraction rate at height z above the ground, and ρ_0, k are constant coefficients, respectively. The refraction rate declines along with the increase of z, which causes the distortion. According to optical principles, the light ray will go through a distorted propagation, with the angle θ in Figure 2 following Snell's Law:

$$n(z) \sin\theta = C, \tag{3}$$

where C is a constant number. The curve of light ray in $y - z$ plane can be formulated as:

$$\frac{dz}{dy} = \frac{1}{\tan\theta} = \sqrt{\frac{(1+\rho_0 e^{-kz})^2}{C^2} - 1}. \tag{4}$$

Given the boundary condition where light rays propagate in parallel with ground near view plane, (4) can be further simplified as:

$$z = z_0 - \frac{k\rho_0}{2(1+\rho_0^2)} y^2. \tag{5}$$

In (5), z_0 denotes the boundary height where the light ray meets view plane, and y is the distance to view plane. The equation verifies that the light rays propagate along parabolic trajectories, and objects which are farther from the view plane will have a higher position in final views. Detailed derivation process of (5) is further provided in *supplementary pages*.

Projection Implementation. Following the guidance of physical modeling of mirage phenomenon, we implement mirage projection through a simple way. Given a point p that lies on the projection ray, (5) can be reorganized as:

$$z'_p = z_p + \frac{k\rho_0}{2(1+\rho_0^2)} y_p^2. \tag{6}$$

Here, z'_p represents the height of point p in projection image, which is consistent with z_0 in (5) where the projection ray meets view plane. We can observe that the projected height of point p is calculated by adding a certain offset to its original height. In order to regulate the offset in a convenient and effective way, we define $\kappa = k\rho_0 / 2(1+\rho_0^2)$ as the coefficient, which is the only parameter in our projection strategy and has a clear physical interpretation. Similar to (1), we can formulate the projected position of p under image coordinates as following:

$$H_p = \frac{z_p + \kappa y_p^2 - z_{\text{min}}}{z_{\text{max}} - z_{\text{min}}} H,\ W_p = \frac{x_p - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} W. \tag{7}$$

With different choices of parameter κ, we can modify the projection result flexibly. To be more specific, as we increase κ, more previously occluded objects can be presented on the projection image at the expense of narrowed view due to reduced height resolution H. It is noteworthy that mirage projection will degenerate to straight-line projection when $\kappa = 0$, indicating that our projection strategy acts as a superior substitute for straight-line projection.

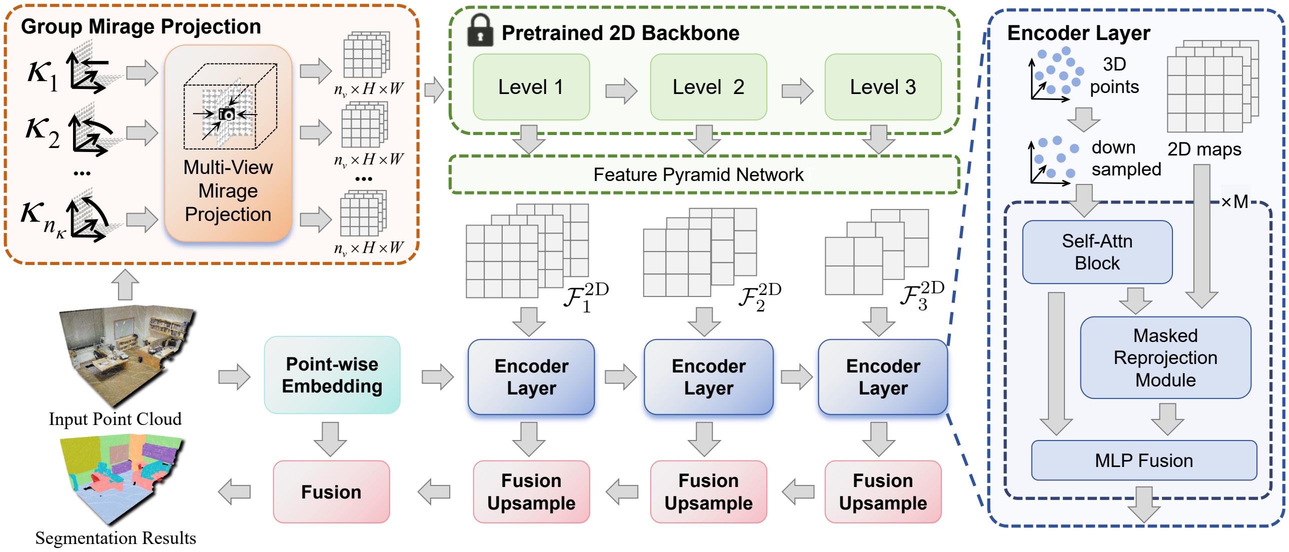

Figure 3. The pipeline of our MirageRoom for point cloud segmentation. We first generate multi-view projection images via group mirage projection module. Then a pre-trained 2D backbone is followed to extract 2D feature maps, with an FPN [26] to further generate multi-scale features. We design a U-Net liked architecture with encoder layers to aggregate point-wise 2D features and fuse with original 3D features.

Densify. Besides the aforementioned strategy to mitigate the occlusions, we also adopt a densify process over projection images via 2D neighbour pooling to handle the problem of empty grids caused by inherent sparsity of point cloud. Instead of establishing a one-to-one correspondence between points and pixels in projection images, we project each point onto pixels which are adjacent to its original projection position. Likewise, when multiple points are projected onto the same pixel, we preserve the value of the one closest to the projection plane. In this way, most vacant pixels can be filled without affecting local semantic features, and thus the projection images are more similar to real-world images.

3.2. Masked Reprojection Module

In this part, we mainly focus on generating precise and valid 3D point-wise features from 2D feature maps. Specifically, we develop a masked reprojection module to bridge the gap between 2D and 3D latent space. The module includes two parts, *i.e.*, mask generation and masked 2D-to-3D cross-attention, which will be further discussed in the following paragraphs.

Mask Generation. As a complement to 3D point-wise features, the final extracted features from 2D maps must exhibit a high degree of precision. Despite the mirage projection strategy can precisely calculate the corresponding position of each point, there still exists two types of mistakes. For one thing, it is natural that the projection plane lies inside indoor scene, which indicates that points can have a distribution on either side of the plane. However, a single projection image can only include points from one side for semantic coherence. For another, although the usage of mirage projection with series of parameter κ significantly mitigates occlusion issues, there are still points occluded in a single projection image. Thus, for each point p, we generate its corresponding mask $M(p) \in \{0,1\}^n$ of n projection images via reprojection. Specifically, we reproject p onto each projection plane according to its corresponding mirage projection setting. The i-th value of $M(p)$ is set to 0 if either of the aforementioned mistakes occurs in i-th projected map, otherwise the i-th value is set to 1. In this way, the mask $M(p)$ can help generate valid 2D features of point p from projection maps.

Masked 2D-to-3D Cross-Attention Based on the mask $M(\cdot)$, we further propose a masked 2D-to-3D cross-attention. Taking N points $\mathcal{P} = \{p_1, p_2, ..., p_N\} \in \mathbb{R}^{N\times 3}$ and their corresponding feature $\mathcal{F}^{3D} = \{f_1, f_2, ..., f_N\} \in \mathbb{R}^{N\times C}$ as input, we aim at fusing corresponding 2D features from n 2D projected maps $\mathcal{F}^{2D} \in \mathbb{R}^{n\times H\times W\times C}$. We take 3D point features as the queries of cross-attention, and combine their corresponding 2D features from each image as keys and values, respectively. Since local features surrounding the corresponding 2D pixel are closely related, we further take the features of k neighbours of corresponding pixel as keys and values besides combining their exact corresponding 2D features, which can be denoted as $\hat{\mathcal{F}}^{2D} \in \mathbb{N}^{N\times nk\times C}$. It is clear that each query vector is assigned with nk key vectors, and the masked 2D-to-3D cross-attention operator can be formulated as:

$$\boldsymbol{q} = \mathcal{F}^{\text{3D}}\mathbf{W}_q,\ \boldsymbol{k} = \hat{\mathcal{F}}^{\text{2D}}\mathbf{W}_k,\ \boldsymbol{v} = \hat{\mathcal{F}}^{\text{2D}}\mathbf{W}_v, \tag{8}$$

where $\mathbf{W}_q$, $\mathbf{W}_k$ and $\mathbf{W}_v$ are linear projections. For i-th point, the output of masked 2D-to-3D cross-attention operation can be formulated as:

$$\begin{aligned} w_{ij} &= \delta(\boldsymbol{q}_i, \boldsymbol{k}_{ij}), \\ \hat{\mathcal{F}}_i &= \sum_{j=1}^{nk} \text{SoftMax}(\boldsymbol{w}_i)_j M(p_i)_{[j/k]} \boldsymbol{v}_j. \end{aligned} \tag{9}$$

Here δ stands for a relation function between keys and queries. Inspired by [50], δ is set to be a subtraction operator. The subscripts under function SoftMax$(\cdot)$ and $M(\cdot)$ represents the index of the elements, and $[\cdot]$ denotes floor operation, respectively. Finally, we get the corresponding valid and precise 2D feature for each point.

3.3. Network Architecture

Based on our mirage projection strategy and masked reprojection module, we construct our network, *i.e.*, Mirage-Room, for indoor point cloud segmentation task. As is shown in Figure 3, our network architecture takes only point cloud as input, following with core modules: group mirage projection, pre-trained 2D backbone, encoder layer, and is finally combined with a U-Net liked structure.

Group Mirage Projection. For the purpose of generating sufficient projection images for point cloud, we adopt a group mirage projection module. For each raw input point cloud $\mathcal{P} \in \mathbb{R}^3$ with corresponding color $\mathcal{F}_c \in \mathbb{R}^{N\times 3}$, we first set the center of the bounding box of $\mathcal{P}$ as the center of view plane, where we generate projection images from n_v view planes. We choose $n_v = 4$ for a unified setting, where the normal directions of projection plane are front, back, left and right, respectively. For each projection planes, we set n_κ different parameter κs as a group for mirage projection, which can generate series of projection images to cover more points. Hence, we obtain $n = n_v n_\kappa$ 2D projection images with RGB information from raw point cloud, which can be denoted as $\mathcal{I}^{\text{2D}} \in \mathbb{R}^{n\times H\times W\times 3}$.

Pre-trained 2D Backbone. To encode 2D features with multi-scale semantic information, we then employ a pre-trained 2D image backbone to transfer rich 2D knowledge. We employ Swin Transformer [27] pre-trained on ImageNet-1K [10] as our 2D backbone. We collect multi-level 2D feature maps from different layers of 2D backbone. Then a Feature Pyramid Network [26] is utilized to further aggregate the multi-level features and provide meaningful 2D maps with different scales. In this way, the pre-trained 2D backbone takes $\mathcal{I}^{\text{2D}}$ as input, and the final output features can be formulated as $\mathcal{F}_1^{\text{2D}}, \mathcal{F}_2^{\text{2D}}, \mathcal{F}_3^{\text{2D}}$ with decreasing resolutions in height and width, respectively.

Encoder Layer. Although we have already obtained multi-scale 2D feature maps, it is crucial to generate corresponding 3D point-wise feature for segmentation task. Hence, we design the main encoder layer of our network, which takes both 3D point cloud features $\mathcal{F}_i^{\text{3D}}$ and 2D feature maps $\mathcal{F}_i^{\text{2D}}$ as input. As is shown in Figure 3, in each encoder layer, we first downsample the point cloud, then a self-attention block [45] is utilized over sampled point cloud to provide more comprehensive point-wise features. After that, we employ our masked reprojection module to aggregate precise and valid 3D point-wise features from 2D feature maps. Finally, the output features are concatenated with the previous 3D point-wise features and mixed through one MLP layer. Following the design of tranditional transformer backbones [20, 50], we repeat the whole process except 3D downsampling for M times to deeply extract and fuse features.

Whole Structure. As is illustrated in Figure 3, the whole U-Net liked network begins with a point-wise embedding layer, where points are embedded into 3D feature space. Then the encoder layers further aggregate 2D and 3D features with the parallel downsampling process of 3D points and 2D maps. In the decoder part, we simply employ an fusion and upsampling module, where 3D points are upsampled and fused with the short-cut point-wise features from corresponding encoder layer.

4. Experiments

4.1. Experimental Setting

Dataset. We use S3DIS [3] (Stanford Large-Scale 3D Indoor Spaces) and ScanNet V2 [9] for semantic segmentation experiments. The S3DIS dataset is a challenging benchmark which contains 6 large-scale indoor areas, 271 rooms and 13 semantic categories. Following the common split protocol [37], area 5 (68 rooms) is picked out for testing and others (203 rooms) are kept for training. ScanNet V2 provides a comprehensive collection of RGB-D scans of indoor environments, accompanied by semantic segmentation labels, camera poses, and other essential metadata. Totally 1,513 room scans in the dataset are divided into 1,201 scenes for training and 312 for validation.

Model Architecture. We adopt the same architecture on both datasets. For the group mirage projection, we choose $\kappa = 0, 1, 2$ as the group parameters of mirage projection. For each κ, we choose 4 projection planes where the normal directions of projection plane are front, back, left and

Table 1. Indoor semantic segmentation results on Area 5 of S3DIS [3] dataset. Overall accuracy (OA), class-average accuracy (mAcc) and classwise mean IoU (mIoU) are reported.

Method	OA	mAcc	mIoU
PointNet [31]	-	49.0	41.1
SegCloud [37]	-	57.4	48.9
TangentConv [36]	-	62.2	52.6
PointCNN [24]	85.9	63.9	57.3
SPGraph [22]	86.4	66.5	58.0
ParamConv [40]	-	67.0	58.3
PAT [47]	-	70.8	60.1
PCT [14]	-	67.6	61.3
HPEIN [18]	87.2	68.3	61.9
GACNet [39]	87.8	-	62.9
SegGCN [23]	88.2	70.4	63.6
MinkUNet [6]	-	71.7	65.4
KPConv [38]	-	72.8	67.1
PTv1 [50]	90.8	76.5	70.4
PointNeXt [33]	90.6	76.8	70.5
PMeta [25]	90.8	-	71.3
PTv2 [45]	91.1	77.9	71.6
MirageRoom w/o 2D model	90.7	76.2	70.4
MirageRoom	**91.3**	**78.2**	**72.0**

right, respectively. Hence, we have $n = n_\kappa n_v = 12$ projection images for each point cloud. For pre-trained 2D backbone, we choose the first 3 layers of Swin-Tiny [27] to obtain multi-level 2D features. For encoder layers, we downsample the points with grid sampling strategy [50]. We set the channel numbers of features in three encoder layers to be 96, 192, 384, respectively.

Implementation Details. We train our MirageRoom on 4 RTX A40 GPUs for all experiments. For S3DIS dataset, we use AdamW optimizer [28] with 0.05 weight decay and 0.006 learning rate to train for 100 epochs. The batch size is set to 12. For ScanNet V2 dataset, we use AdamW optimizer with a smaller weight decay 0.02 applied. The learning rate is scheduled by OneCycleLR [34], where where the learning raises from 0.0005 to 0.005 in the first 5 epochs and cosine annealing to 0 in the remaining 95 epochs. Other settings remain consistent with the training process on S3DIS.

4.2. Results

Segmentation Results on S3DIS. Table 1 demonstrates the results of recent state-of-the-art methods and the proposed MirageRoom on the Area 5 of S3DIS. Clearly, our MirageRoom achieves best performances on all three metrics including classwise mean IoU (%), class-average accuracy (%), and overall accuracy (%). Specifically, our method surpasses different kinds of methods including MLP-based [31, 33], convolution-based [24, 36], graph-based [22, 39] and transformer-based [14, 45, 50] approaches, especially the PTv2 [45] method where more point-wise transformer layers are adopted. It is worth noticing that our method achieves a considerable improvement over the baseline without the guidance from 2D features, which demonstrates the effectiveness of our method to benefit from 2D pre-trained models.

Table 2. Indoor semantic segmentation results on validation set of ScanNet V2 [9] dataset. The size of learnable parameters and classwise mean IoU (mIoU) are reported.

Method	Input Modality	Support 2D Guidance	Learnable Params	mIoU
PointNet++ [32]	3D	✗	1.0M	53.5
PointConv [44]	3D	✗	-	61.0
PointASNL [46]	3D	✗	-	63.5
KPConv [38]	3D	✗	15M	69.2
SparseCNN [13]	3D	✗	-	69.3
PTv1 [50]	3D	✗	7.8M	70.6
PointNext [33]	3D	✗	41.6M	71.5
PMeta [25]	3D	✗	19.7M	72.8
MinkUNet [6]	3D	✗	37.9M	72.2
StraFormer [20]	3D	✗	8.0M	74.3
PTv2 [45]	3D	✗	12.8M	75.4
MVPNet [17]	2D+3D	✓	0.98M	65.0
BPNet [16]	2D+3D	✓	-	73.9
MirageRoom	3D	✓	5.8M	74.9

Segmentation Results on ScanNet V2. The mIoU results on the validation set of ScanNet V2 is illustrated in Table 2. Apparently, our MirageRoom achieve a better performance than most of the recent state-of-the-art methods, including an improvement over StraFormer [20] with fewer learnable parameters. We also achieves a comparable results with PTv2 [45] based on about 60% fewer parameters. Besides, compared with methods which support the guidance from 2D models, our method takes only 3D points as input without additional images and camera poses, yet we still achieves better performance, *i.e.*, 74.9% vs 73.9% against BPNet [16].

Qualitative Results. Qualitative results of PTv2 [45] and our method are shown in Figure 4. It is clear that the results predicted by our model is highly close to groundtruth, especially objects which shares similar geometric structures and are likely to be recognized as other labels. Further detailed visual comparisons will be displayed in *supplementary pages* due to the space limitation.

4.3. Analysis of Mirage Projection

To verify the effectiveness of our mirage projection, we have a comprehensive comparison over the projection re-

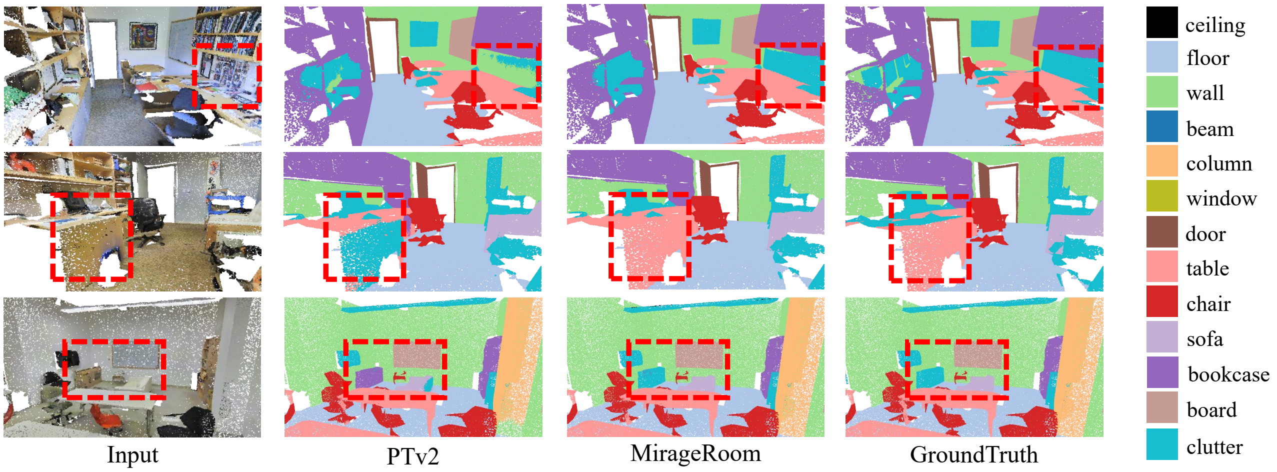

Figure 4. Visualization results of semantic segmentation results of PTv2 [45] and MirageRoom on S3DIS [3] dataset. Best viewed in color.

Table 3. The occupancy (%) of 3D points via different projection strategies in two datasets.

Strategy	S3DIS	ScanNet V2
Straight-line	20.0	47.9
Mirage	35.2	73.9
Relative ↑	+75.5%	+54.4%

sults of our strategy with the ones of conventional straight-line projection. Specifically, we evaluate our projection strategy from two aspects, *i.e*. the visual comparisons of 2D projection images and occupancy analysis of 3D points.

Visual Comparisons of 2D projection images. We generate the results of our projection images with different settings of κ compared with conventional straight-line projection in Figure 5. We choose the same group of κs in our segmentation tasks, *i.e*., 0, 1, 2. It is clear that with the help of our mirage projection, the originally occluded objects can be viewed without changing view-points. For example, in the first row of Figure 5, the red sofa is originally occluded by front tables and chairs, while with the help of our mirage projection, we can clearly see sofa without changing the position and direction of view plane. Meanwhile, as we increase κ, more occluded details can be further projected, such as the chair legs in the second row of Figure 5. Besides the visualization of originally occluded objects, our densify strategy also provide a more realistic projection image with less incoherence.

It is worth noting that although the visualization results of mirage projection are similar to the straight-line projection ones where view-points are changed, the basic physical modeling is different since varying view-points still follows straight-line projection under homogeneous medium. The new modeling brings two significant advantages to our projection strategy: 1) Convenience. Mirage projection produces various images with modifying only one parameter, while varying view-points requires elegant choice of many parameters to control the position and direction of view plane. 2) Generalization capability. Mirage projection is flexible to fit various scenes. In contrast, it is hard to develop a unified algorithm which generates appropriate view-points for variable indoor scenes.

Occupancy of 3D points. We further analyze the occupancy of 3D points by different projection strategies. The occupancy refers to the proportion of 3D points effectively covered by projection images (*i.e*., not occluded by other points) out of all points. We perform multi-view straight-line projection and mirage projection in rooms, respectively. The occupancy visualization is illustrated in Figure 6, and numeric results are shown in Table 3, respectively. We can clearly see from Figure 6 that massive points can not be mapped onto projection plane due to occlusion of straight-line projection, while our mirage projection covers most of the points without changing projection planes. The results in Table 3 further reveal the fact that our projection strategy is able to cover more points.

4.4. Ablation Study

In this section, we conduct extensive ablation studies to verify the effectiveness of each component in our method. We report the results on S3DIS dataset.

2D Pre-trained Model and Group Mirage Projection. We first evaluate the effectiveness of the guidance from 2D pre-trained model and group mirage projection. The results are illustrated in Table 4. In Experiment I, we do not use any 2D models, which is the baseline of our model. The model in Experiment II adopts 2D models to generate features from straight-line projection images. However, the perfor-

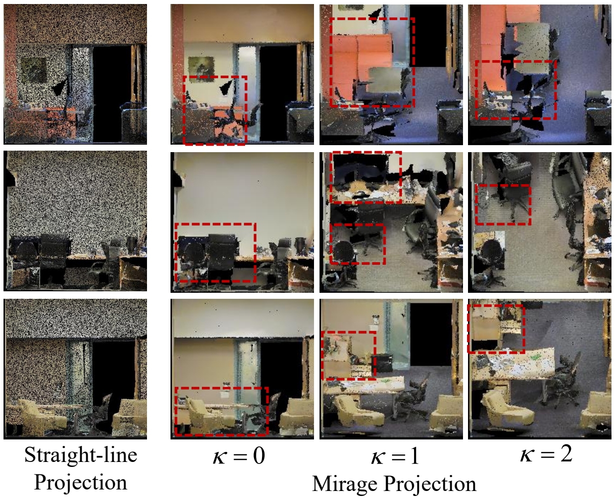

Figure 5. The projection visualization results of different projection strategies. With the help of mirage projection, we can generate realistic projections with more originally occluded details.

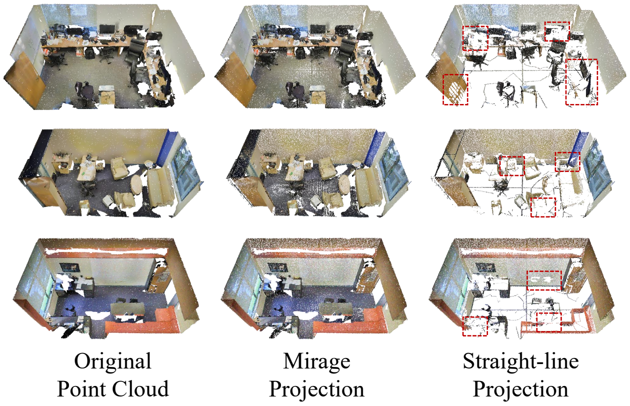

Figure 6. The occupancy visualization of 3D points by different projection strategies. **Left**: original point cloud. **Middle**: 3D points covered by mirage projection. **Right**: 3D points covered by straight-line projection.

mance drops due to the incoherence in vanilla straight-line projection. In Experiment III, we add densify process over straight-line projection. Due to the realistic projection images, the 2D model is able to extract features to promote the performance. Experiment II to VII reveal the promotion in employing group mirage projection with more points included, and an excessive κ value hurts the performance due to the overflow of edge pixels of projection images.

Effectiveness of Masked Reprojection Module. We then conduct the experiments over masked reprojection module to verify the importance of the precise mask and the number of 2D neighbour k which decides the length of key vectors for each point query vector. The results are illustrated in Table 5. We can clearly observe that the performance have a significant improvement with the help of our precise mask compared with experiment (1) to (2). We choose $k = 5$ as the number of candidate neighbour keys in our module due to the higher performance.

Table 4. Ablation study over 2D pre-trained model and group mirage projection on S3DIS. "S" and "M" in Projection Strategy represents straight-line projection and mirage projection, respectively.

ID	2D Model	Projection Strategy	κ	mIoU (%)
I	✗	-	-	70.4
II	✓	S	-	69.5
III	✓	M	0	70.9
IV	✓	M	0, 1	71.5
V	✓	M	0, 0.5, 1	71.8
VI	✓	M	0, 1, 2	72.0
VII	✓	M	0, 1, 3	71.6

Table 5. Ablation study over masked reprojection module on S3DIS dataset.

ID	Mask	k	mIoU (%)
(1)	w/o mask	1	70.7
(2)	w/ mask	1	71.8
(3)	w/ mask	5	72.0
(4)	w/ mask	9	71.9

5. Conclusion

In this paper, we propose MirageRoom, a point cloud architecture with 2D pre-trained models for segmentation task based on projection. Different from existing projection-based methods where straight-line projection causes many occlusions, we propose mirage projection based on the premise of heterogeneous medium, thereby the previously occluded objects can be exposed by distorted rays. In this way, the projection images can cover more points and therefore provide sufficient 2D features for fine-grained segmentation tasks. The network is further constructed based on our projection strategy, and we show that our proposed MirageRoom is effective in learning point-wise features.

Limitations and Future work. Despite MirageRoom successes in generating and combining comprehensive point-wise features from 2D projections, the complete solution of occlusion is still non-trivial and elusive. Meanwhile, the potential of mirage projection has not been fully discovered, and we believe it can further inspire future works.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China under Grant 62206147, Grant 62125603, Grant 62321005, Grant 62336004, and in part by DengFeng Fund.

References

[1] Evangelos Alexiou, Evgeniy Upenik, and Touradj Ebrahimi. Towards subjective quality assessment of point cloud imaging in augmented reality. In *MMSPW*, pages 1–6. IEEE, 2017. 1

[2] Angelika Ando, Spyros Gidaris, Andrei Bursuc, Gilles Puy, Alexandre Boulch, and Renaud Marlet. Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving. In *CVPR*, pages 5240–5250, 2023. 2

[3] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *CVPR*, pages 1534–1543, 2016. 1, 2, 5, 6, 7

[4] Bradford R Bean and Gordon D Thayer. Models of the atmospheric radio refractive index. *Proceedings of the IRE*, 47 (5):740–755, 1959. 3

[5] Lamont V Blake. Ray height computation for a continuous nonlinear atmospheric refractive-index profile. *Radio Science*, 3(1):85–92, 1968. 3

[6] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, pages 3075–3084, 2019. 6

[7] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, pages 3075–3084, 2019. 2

[8] Angela Dai and Matthias Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *ECCV*, pages 452–468, 2018. 1, 2

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017. 1, 2, 5, 6

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 1, 5

[11] Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. Dilated point convolutions: On the receptive field size of point convolutions on 3d point clouds. In *ICRA*, pages 9463–9469. IEEE, 2020. 2

[12] Benjamin Graham and Laurens Van der Maaten. Submanifold sparse convolutional networks. *arXiv preprint arXiv:1706.01307*, 2017. 2

[13] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, pages 9224–9232. IEEE, 2018. 2, 6

[14] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *CVM*, 7:187–199, 2021. 1, 2, 6

[15] Abdullah Hamdi, Silvio Giancola, and Bernard Ghanem. Mvtn: Multi-view transformation network for 3d shape recognition. In *ICCV*, pages 1–11, 2021. 2

[16] Wenbo Hu, Hengshuang Zhao, Li Jiang, Jiaya Jia, and Tien-Tsin Wong. Bidirectional projection network for cross dimension scene understanding. In *CVPR*, pages 14373–14382, 2021. 1, 2, 6

[17] Maximilian Jaritz, Jiayuan Gu, and Hao Su. Multi-view pointnet for 3d scene understanding. In *ICCVW*, pages 0–0, 2019. 1, 2, 6

[18] Li Jiang, Hengshuang Zhao, Shu Liu, Xiaoyong Shen, Chi-Wing Fu, and Jiaya Jia. Hierarchical point-edge interaction network for point cloud semantic segmentation. In *ICCV*, pages 10433–10441, 2019. 2, 6

[19] Lingdong Kong, Youquan Liu, Runnan Chen, Yuexin Ma, Xinge Zhu, Yikang Li, Yuenan Hou, Yu Qiao, and Ziwei Liu. Rethinking range view representation for lidar segmentation. In *ICCV*, pages 228–240, 2023. 2

[20] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *CVPR*, pages 8500–8509, 2022. 1, 2, 5, 6

[21] Xin Lai, Yukang Chen, Fanbin Lu, Jianhui Liu, and Jiaya Jia. Spherical transformer for lidar-based 3d recognition. In *CVPR*, pages 17545–17555, 2023. 2

[22] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, pages 4558–4567, 2018. 2, 6

[23] Huan Lei, Naveed Akhtar, and Ajmal Mian. Seggcn: Efficient 3d point cloud segmentation with fuzzy spherical kernel. In *CVPR*, pages 11611–11620, 2020. 6

[24] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *NIPS*, 31, 2018. 2, 6

[25] Haojia Lin, Xiawu Zheng, Lijiang Li, Fei Chao, Shanshan Wang, Yan Wang, Yonghong Tian, and Rongrong Ji. Meta architecture for point cloud analysis. In *CVPR*, pages 17682–17691, 2023. 6

[26] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 4, 5

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 5, 6

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[29] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *IROS*, pages 4213–4220. IEEE, 2019. 2

[30] François Pomerleau, Francis Colas, Roland Siegwart, et al. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends in Robotics*, 4(1):1–104, 2015. 1

[31] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017. 2, 6

[32] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NIPS*, 30, 2017. 1, 2, 6

[33] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved

training and scaling strategies. *NIPS*, 35:23192–23204, 2022. 1, 2, 6

[34] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. *arXiv e-prints*, 2017. 6

[35] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *ICCV*, pages 945–953, 2015. 2

[36] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *CVPR*, pages 3887–3896, 2018. 6

[37] Lyne Tchapmi, Christopher Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. Segcloud: Semantic segmentation of 3d point clouds. In *2017 international conference on 3D vision (3DV)*, pages 537–547. IEEE, 2017. 5, 6

[38] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, pages 6411–6420, 2019. 1, 2, 6

[39] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *CVPR*, pages 10296–10305, 2019. 2, 6

[40] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. Deep parametric continuous convolutional neural networks. In *CVPR*, pages 2589–2597, 2018. 6

[41] Ziyi Wang, Yongming Rao, Xumin Yu, Jie Zhou, and Jiwen Lu. Semaffinet: Semantic-affine transformation for point cloud segmentation. In *CVPR*, pages 11819–11829, 2022. 1, 2

[42] Ziyi Wang, Xumin Yu, Yongming Rao, Jie Zhou, and Jiwen Lu. P2p: Tuning pre-trained image models for point cloud analysis with point-to-pixel prompting. *NIPS*, 35:14388–14402, 2022. 2

[43] Florian Wirth, Jannik Quehl, Jeffrey Ota, and Christoph Stiller. Pointatme: efficient 3d point cloud labeling in virtual reality. In *IV*, pages 1693–1698. IEEE, 2019. 1

[44] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *CVPR*, pages 9621–9630, 2019. 6

[45] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *NIPS*, 35:33330–33342, 2022. 1, 2, 5, 6, 7

[46] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *CVPR*, pages 5589–5598, 2020. 6

[47] Jiancheng Yang, Qiang Zhang, Bingbing Ni, Linguo Li, Jinxian Liu, Mengdie Zhou, and Qi Tian. Modeling point clouds with self-attention and gumbel subset sampling. In *CVPR*, pages 3323–3332, 2019. 6

[48] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *CVPR*, pages 8552–8562, 2022. 2, 3

[49] Renrui Zhang, Liuhui Wang, Yu Qiao, Peng Gao, and Hongsheng Li. Learning 3d representations from 2d pre-trained models via image-to-point masked autoencoders. In *CVPR*, pages 21769–21780, 2023. 2, 3

[50] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *ICCV*, pages 16259–16268, 2021. 1, 2, 5, 6

[51] Hui Zhou, Xinge Zhu, Xiao Song, Yuexin Ma, Zhe Wang, Hongsheng Li, and Dahua Lin. Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation. *arXiv preprint arXiv:2008.01550*, 2020. 2