Distilling Diffusion Models to Efficient 3D LiDAR Scene Completion
Supplementary Material

Contents

1. Experiment protocol

1.1. Dataset setup

SemanticKITTI [1] dataset is a large-scale benchmark for 3D semantic segmentation in autonomous driving, extending the KITTI Odometry dataset with dense semantic annotations for over 43,000 LiDAR scans. It provides labels for 25 classes, such as “car,” “road,” and “building,” capturing diverse urban and rural scenes. The SemanticKITTI dataset consists of 22 sequences, where sequences 00-10 are densely annotated for each scan, enabling tasks such as semantic segmentation and semantic scene completion using sequential scans. Sequences 11-21 serve as the test set, showcasing diverse and challenging traffic situations and environment types to evaluate model performance in real-world autonomous driving scenarios. SemanticKITTI is widely used in research and serves as a critical resource for advancing LiDAR-based perception systems.

KITTI-360 [12] dataset is a comprehensive benchmark for 3D scene understanding in autonomous driving, capturing 360-degree panoramic imagery and 3D point clouds across diverse urban environments. It includes over 73 km of driving data with dense semantic annotations for both 2D (images) and 3D (point clouds), covering categories like “vehicles,” “buildings,” and “vegetation.” KITTI-360 provides high-resolution sensor data, including LiDAR, GPS/IMU, and stereo camera recordings, making it ideal for tasks such as 3D semantic segmentation, panoptic segmentation, and mapping in real-world driving scenarios.

1.2. Evaluation metrics

Chamfer Distance (CD) [2] is a metric used to measure the similarity between two sets of points, often employed for evaluating the quality of generated point clouds or geometric shapes. For two point sets P and Q, the Chamfer Distance is defined as:

$$CD(P,Q) = \frac{1}{|P|}\sum_{p\in P}\min_{q\in Q}\|p-q\|^2 + \frac{1}{|Q|}\sum_{q\in Q}\min_{p\in P}\|q-p\|^2 \tag{1}$$

The first term calculates the average squared distance from each point in P to its nearest neighbour in Q. The second term calculates the average squared distance from each point in Q to its nearest neighbour in P. Chamfer Distance evaluates how well two point sets approximate each other by considering their nearest neighbour distances in both directions. CD effectively captures local geometric features and exhibits strong robustness in local shape matching, which is commonly used in evaluating the matching and reconstruction of 3D point clouds.

Jensen-Shannon Divergence (JSD) [14] is a symmetric measure of similarity between two probability distributions. It is a variation of the Kullback-Leibler (KL) divergence and is widely used in information theory, statistics, and machine learning. Given two probability distributions P and Q over the same domain, JSD is defined as:

$$JSD(P\|Q) = \frac{1}{2}KL(P\|M) + \frac{1}{2}KL(Q\|M) \quad (2)$$

Here $M = \frac{1}{2}(P + Q)$ is the average distribution, and $KL(P||M)$ is the Kullback-Leibler divergence.

JSD measures how much P and Q diverge from their average distribution M. It is symmetric $(JSD(P||Q) = JSD(Q||P))$ and always produces a finite value in the range [0, 1] when using base-2 logarithms. Unlike KL divergence, JSD avoids issues with undefined values when probabilities are zero in one of the distributions. JSD is an efficient metric to evaluate the similarity between two distributions. The calculation of JSD in this paper is followed by Xiong *et al.* [25].

1.3. Implementation details

We choose the pre-trained LiDiff [16] model as the teacher model $\boldsymbol{\epsilon_\theta}$, the student model G_{stu} and the auxiliary diffusion model $\boldsymbol{\epsilon}_\phi$ shares the same network architecture as the teacher model and are initialized by the teacher model. The ScoreLiDAR is trained on SemanticKITTI dataset. The pre-trained diffusion model is provided by the official release of LiDiff [16]. For fair comparison, we follow LiDiff [16]'s training strategy. ScoreLiDAR is trained on sequences 00–07 and 09–10 of SemanticKITTI (does not train on KITTI-360), and evaluated on sequence 08 of SemanticKITTI and sequence 00 of KITTI-360.

For optimization, we use the Stochastic Gradient Descent (SGD) optimizer with the default parameters. The learning rate is set to $3e-5$ and the batch size is set to 1. The training ratio between the student model and the auxiliary diffusion model is maintained at $1:1$. To reduce computational costs, when calculating the point-wise loss, we first randomly select $\frac{1}{10}$ of the points from the ground truth scene. Then, following the proposed method, we select the top $\frac{1}{3}$ points with the highest curvature from these points as the key points to calculate the distance matrix. That is, the final number of key points is $\frac{1}{30}$ of the total number of points in the ground truth scene. When calculating the K-nearest neighbours, we set $K = 180$. The weights of scene-wise loss λ_{scene} and the point-wise loss λ_{point} are set to 0.5 and 0.01, respectively. ScoreLiDAR requires only 50 iterations to achieve convergence, taking approximately 10 minutes on a single A40 GPU, which is highly efficient. Our model and code are publicly available on https://github.com/happyw1nd/ScoreLiDAR.

2. Discussion

2.1. The effectiveness of the distillation on improving the completion efficiency

In this part, we provide a detailed discussion about the efficiency of the proposed distillation method.

Why is it reasonable to initialize the student model and auxiliary diffusion model using the teacher model? Firstly, such an initialization method is commonly used in existing methods [13, 22, 26–28]. Secondly, the pre-trained teacher model $\boldsymbol{\epsilon}_\theta$ contains the information about the training distribution, initializing the student model with the teacher model to perform distillation is essentially a fine-tuning process for the teacher model, which can accelerate the efficiency of the distillation. Third, although the student model has the same parameters and the network structure as the teacher model, its sampling distribution is different from that of the teacher model due to the different sampling steps. The teacher model $\boldsymbol{\epsilon}_\theta$ (LiDiff [16] in this case) conducts the multi-step sampling.

$$\mathcal{G}^{t-1} = \frac{1}{\sqrt{\alpha^t}}\left(\mathcal{G}^t - \frac{1-\alpha^t}{\sqrt{1-\bar{\alpha}^t}}\boldsymbol{\epsilon}_\theta\left(\mathcal{G}^t, \mathcal{P}, t\right)\right) + \sigma^t \boldsymbol{z} \quad (3)$$

The teacher model $\boldsymbol{\epsilon}_\theta$ from LiDiff [16] conducts 50-steps sampling by repeating Eq. (3). The student model G_{stu} conducts the single-step sampling. After G_{stu} predicts the noise, a single-step denoising in Eq. (4) is performed to directly obtain the completed scene $\mathcal{G}^0$.

$$\mathcal{G}^0 = \frac{1}{\sqrt{\alpha^t}}\left(\mathcal{G}^t - \frac{1-\alpha^t}{\sqrt{1-\bar{\alpha}^t}}\boldsymbol{\epsilon}_\theta\left(\mathcal{G}^t, \mathcal{P}, t\right)\right) \quad (4)$$

Thus, the single-step sampling scene of the student model is different from the multi-step sampling scene of the teacher model.

Why is the distillation loss effective? Why does the student model get optimized? Firstly, our distillation loss is different from the standard loss of the diffusion models such as DDPM [4]. The loss of DDPM is to directly predict the noise added to the training sample. Differently, in proposed ScoreLiDAR, the distillation loss utilizes the noise predicted by the student model G_{stu} to perform a one-step sampling, resulting in a completed scene $\mathcal{G}^0$ different from the multi-step sampling of the teacher model. The completed scene $\mathcal{G}^0$ is then perturbed noise on a random timestep t to obtain the noisy scene $\mathcal{G}^t$, and the difference between two score functions according to $\mathcal{G}^t$ is calculated to serve as the gradient for optimizing the student model, as shown in Eq. (5)

$$\mathcal{L}_{KL} \approx \mathbb{E}_{t,\epsilon}\left[\left\|\boldsymbol{\epsilon}_\theta\left(\mathcal{G}^t, \mathcal{P}, t\right) - \boldsymbol{\epsilon}_\phi\left(\mathcal{G}^t, \mathcal{P}, t\right)\right\|_2^2\right] \tag{5}$$

Secondly, although the auxiliary diffusion model $\boldsymbol{\epsilon}_\phi$ is initilized from the teacher model $\boldsymbol{\epsilon}_\theta$, the gradient in Eq. (5) is non-zero and efficient. Recall that the auxiliary diffusion model$\boldsymbol{\epsilon}_\phi$ and the student model G_{stu} are trained alternately. The auxiliary diffusion model $\boldsymbol{\epsilon}_\phi$ is first trained on $\mathcal{G}^0$ to fit the one-step sampling distribution, which is different from the pre-trained distribution of the teacher model. Although the auxiliary diffusion model $\boldsymbol{\epsilon}_\phi$ is also initilized from the teacher model, its parameter will be updated and become different from the teacher model's after one optimization. Then, when G_{stu} is optimized using the gradient in Eq. (5), the output of $\boldsymbol{\epsilon}_\theta\left(\mathcal{G}^t, \mathcal{P}, t\right)$ is naturally different from the output of $\boldsymbol{\epsilon}_\phi\left(\mathcal{G}^t, \mathcal{P}, t\right)$ due to the optimization of $\boldsymbol{\epsilon}_\phi$. Thus, as mentioned in Sec. 4.1 of the main paper, the non-zero gradient in Eq. (5) will optimize the distribution of G_{stu} moving towards the pre-trained distribution of the teacher model. Then, $\boldsymbol{\epsilon}_\phi$ and G_{stu} are optimized in turn to convergence.

Why is one-step sampling used during training, while few-step sampling is used during inference? Firstly, in traditional diffusion models like DDPM [4] and DDIM [19], they directly predict the whole noise added in the diffusion process during the training, whereas a multi-step sampling process is performed during sampling. Meanwhile, different sampling methods allow for sampling with different numbers of steps. Intuitively, this indicates that although the training objective remains the same, the sampling can be performed with different numbers of steps. A more profound explanation is from solving the stochastic differential equation as in the diffusion model. Moreover, Consistency Model [22] also has a similar setting; its student model performs one-step generation during training but conducts multi-step generation during sampling. We adopted a similar principle as in the Consistency Model. The one-step sampling in our training is to increase the fidelity of the resulting sample given different noisy samples, while in inference multi-step sampling would gradually refine the final result. Therefore, although the student model performs one-step sampling during training, the quality of scene completion can be improved by increasing the number of sampling steps during the inference. More visually, the one-step generation procedure of the Consistency Model is "noise→image", and the multi-step generation procedure is "noise→noisy image→$\cdots$→image". In this paper, the one-step sampling of student model G_{stu} during training is "noise→predicted noise→completed scene" and the few-step sampling during inference is "noise→predicted noise→noisy scene→predicted noise→noisy scene→predicted noise→$\cdots$→completed scene". In summary, it is reasonable to use one-step sampling during training and multi-step sampling during inference.

2.2. The differents of scene-wise loss and Chamfer Distance

The scene-wise loss has the following form

$$\mathcal{L}_{scene} = \frac{1}{|\mathcal{G}^0|}\sum_{\boldsymbol{p}_i^0 \in \mathcal{G}^0} \min_{\boldsymbol{p} \in \mathcal{G}} \|\boldsymbol{p}_i^0 - \boldsymbol{p}\|^2 \tag{6}$$

The scene-wise loss is part of the Chamfer Distance. The reason for using only part of the Chamfer Distance (CD) is that the optimization objective of scene-wise loss is to ensure that the points in the generated scene $\mathcal{G}^0$ are as close as possible to the corresponding points in the ground truth. The unused term in CD matches each point in ground truth with its nearest point in generated scene $\mathcal{G}^0$, which may lead to points in $\mathcal{G}^0$ being pushed toward the average position of non-existent matching points in the ground truth. This effect is detrimental to scene completion.

2.3. Discussion on the significance of this study

Firstly, we discuss the significance of this study. For autonomous vehicles, accurately recognizing and perceiving their surrounding environment during operation is critical [10, 11]. This is particularly important for identifying objects that may affect the vehicle's movement, such as other vehicles, pedestrians, traffic cones, and signposts [6]. The accurate and efficient recognition of these objects is essential for the safe operation of autonomous vehicles. However, the scan data obtained by onboard LiDAR is sparse [9, 23], and it is difficult to identify key objects such as vehicles from the magnified regions of the sparse scan. Autonomous vehicles cannot obtain sufficient information about the driving environment from these sparse LiDAR scans [7, 17]. Therefore, it is necessary to use appropriate methods to complete the sparse LiDAR scans.

LiDiff [16] uses DDPM [4] models to complete 3D LiDAR scenes, achieving impressive results. However, due to the inherent characteristics of diffusion models, LiDiff [16] requires approximately 30 seconds to complete a single scene, limiting its applicability in autonomous vehicles. In contrast, the proposed ScoreLiDAR can complete a scene in almost 5 seconds, more than 5 times faster than LiDiff [16], while achieving higher completion quality. Thus, with the scenes completed by the proposed ScoreLiDAR, autonomous vehicles can more easily recognize critical objects in their driving environment, enabling safer and more effective navigation.

3. Additional completed scenes

Fig. 2 and Fig. 3 show additional completed scenes by the proposed ScoreLiDAR and compare them with the scenes

Model	SemanticKITTI			KITTI360		
	CD ↓	JSD ↓	EMD ↓	CD ↓	JSD ↓	EMD ↓
ScoreLiDAR	0.342	0.399	23.26	0.452	0.437	23.02
w/o Point-wise loss	0.351	0.414	23.37	0.485	0.486	23.29
w/o Scene-wise loss	0.367	0.422	24.89	0.477	0.451	24.69
w/o Structural Loss	0.419	0.430	24.13	0.549	0.445	24.56

Table 1. Ablation study of the scene-wise and point-wise loss. The metrics refer to the performance with refinement. Colors denote the 1st, 2nd, and 3rd best-performing model.

ScoreLiDAR	SemanticKITTI			KITTI360		
	CD ↓	JSD ↓	EMD ↓	CD ↓	JSD ↓	EMD ↓
$\lambda_{scene} = 0.5, \lambda_{point} = 0.01$	0.342	0.399	23.26	0.452	0.437	23.02
$\lambda_{scene} = 0.05, \lambda_{point} = 0.01$	0.354	0.409	23.40	0.494	0.457	23.21
$\lambda_{scene} = 0.5, \lambda_{point} = 0.1$	0.358	0.419	23.27	0.539	0.474	23.09

Table 2. Ablation study of the different weights of the scene-wise and point-wise loss. The first row refers to the default configuration of the ScoreLiDAR. The metrics refer to the performance with refinement.

Model	SemanticKITTI		KITTI360	
	CD ↓	JSD ↓	CD ↓	JSD ↓
LiDiff (Refined)	0.375	0.416	0.517	0.446
w/ Structural loss	0.399	0.426	0.535	0.450

Table 3. Ablation study of training LiDiff [16] with structural loss.

completed by LiDiff [16].

4. Additional experiment results

4.1. More ablation study of structural loss

To further validate the effectiveness of the structural loss, we evaluated the performance of variants trained with only point-wise loss or scene-wise loss and compared them with default ScoreLiDAR. As shown in Tab. 1, compared to the default ScoreLiDAR, the performance of variants trained with only scene-wise loss or point-wise loss decreased. However, compared to the variants without structural loss, the variants using only one type of loss still showed improved completion performance. These results confirm the effectiveness of the structural loss in the distillation process.

Additionally, we investigated the impact of different weights of scene-wise and point-wise loss on the completion quality. The results are shown in Tab. 2. It can be observed that reducing λ_{scene} or increasing λ_{point} leads to a decline in the performance of ScoreLiDAR but still achieves a comparable performance. This verifies the effectiveness of the proposed structural loss in improving the completion performance of the student model.

Finally, we trained LiDiff using structural loss to investigate whether structural loss can enhance the performance of LiDiff. The results are shown in Tab. 3. Training LiDiff [16] with structural loss does not result in a performance improvement. This may be because structural loss is not suitable for direct addition to the training loss of LiDiff [16], *i.e.* the traditional diffusion model training loss.

ScoreLiDAR	SemanticKITTI			KITTI360		
	CD ↓	JSD ↓	EMD ↓	CD ↓	JSD ↓	EMD ↓
$n = 1/20$	0.329	0.392	24.35	0.528	0.475	24.01
$n = 1/30$	0.342	0.399	23.26	0.452	0.437	23.02
$n = 1/60$	0.346	0.409	25.19	0.452	0.471	25.05
$n = 1/70$	0.428	0.454	25.60	0.466	0.479	25.44

Table 4. Ablation study of different key points number. The result of $n = 1/30$ refers to the default configuration of the ScoreLiDAR. The metrics refer to the performance with refinement.

Model	CD ↓	JSD ↓	EMD ↓
Random selection	0.384	0.433	23.23
farthest selection	0.442	0.459	23.63
ScoreLiDAR	0.342	0.399	23.26

Table 5. Comparison of different selection methods on SemanticKITTI dataset. The performance of the proposed selection method is better than others.

4.2. Ablation study of different key point number

As mentioned in Sec. 1.3, the optimal number of the key point is set to the $\frac{1}{30}$ of the total number of points in the ground truth. To investigate the impact of different numbers of key points on the completion performance of ScoreLiDAR, we decreased the number of key points for model training and evaluated the completion performance. As shown in Tab. 4, the final performance of ScoreLiDAR is positively correlated with the number of key points. When the number of key points decreases, the performance of ScoreLiDAR declines. This is because an insufficient number of key points causes the point-wise loss to fail in effectively capturing the relative positional information between key points, preventing the student model from learning the local geometric structure, and thereby reducing the completion quality. However, when the number of key points is too large, it can easily cause an out-of-memory issue and reduce training efficiency. Therefore, this paper sets $n = \frac{1}{30}$.

4.3. Ablation study of different key point selection method

We compare different key point selection methods including random selection and farthest selection with the proposed selection method based on curvature. The results in Tab. 5 show that using the proposed selection method based on curvature achieves the optimal performance than other selection methods.

Model	CD ↓	JSD ↓	EMD ↓	Time (s) ↓
LiDiff (50 steps) [16]	0.564	0.549	21.98	29.18
LiDiff (50 steps Refined) [16]	0.517	0.446	22.96	29.43
LiDiff (8 steps) [16]	0.619	0.471	24.85	5.46
LiDiff (8 steps Refined) [16]	0.550	0.462	25.49	5.77
ScoreLiDAR (8 Steps)	0.452	0.437	23.02	5.14
ScoreLiDAR (4 Steps)	0.482	0.461	23.76	3.16
ScoreLiDAR (2 Steps)	0.525	0.457	-	1.69
ScoreLiDAR (1 Steps)	0.750	0.478	-	1.03

Table 6. Ablation study of different sampling steps on the KITTI-360 dataset. The metrics of ScoreLiDAR refer to the performance with refinement.

Model	User preference ↑
LiDiff [16]	35%
ScoreLiDAR	65%

Table 7. Results of user study. Our ScoreLiDAR outperforms the existing SOTA model.

4.4. Ablation study of different sampling steps on KITTI-360

We also conduct the ablation study of different sampling steps on the KITTI-360 dataset. The results are shown in Tab. 6. Similar to the results on the SemanticKITTI dataset, as the number of sampling steps decreases, the time required for ScoreLiDAR to complete a scene is reduced. Although the completion performance declines slightly, it remains comparable to that of existing SOTA models.

4.5. User study

The user study is conducted to verify the completion performance of ScoreLiDAR further. We first used ScoreLiDAR and the current SOTA method LiDiff [16] to complete the same 30 input LiDAR scans, resulting in 30 pairs of completed scenes. We then randomly recruited seven volunteers and guided each to evaluate the detail and fidelity of these 30 pairs of scene images, selecting the one they believed to be closer to the ground truth. The seven volunteers included five men and two women, aged 24–30, with five participants having research backgrounds related to autonomous driving or LiDAR scene completion and the remaining two participants having backgrounds related to artificial intelligence. They were given unlimited time for the evaluation, but the average completion time for all volunteers was 30 minutes.

The result of the user study is shown in Tab. 7. Compared to LiDiff, ScoreLiDAR received a 65% user preference, surpassing the majority threshold. This indicates that, in the eyes of most users, the detail and fidelity of the scenes completed by ScoreLiDAR more closely resemble the ground truth. The results of the user study further demonstrate the effectiveness of ScoreLiDAR in LiDAR scene completion.

Model	SemanticKITTI (IoU) % ↑		
	0.5m	0.2m	0.1m
LMSCNet [18]	32.23	23.05	3.48
LODE [9]	43.56	47.88	6.06
MID [23]	45.02	41.01	16.98
PVD [29]	21.20	7.96	1.44
LiDiff [16]	42.49	33.12	11.02
LiDiff (Refined) [16]	40.71	38.92	24.75
ScoreLiDAR	38.43	25.75	8.34
ScoreLiDAR (Refined)	37.33	29.57	15.63

Table 8. The IoU evaluation results on the SemanticKITTI dataset.

Model	KITTI-360 (IoU) % ↑		
	0.5m	0.2m	0.1m
LMSCNet [18]	25.46	16.35	2.99
LODE [9]	42.08	42.63	5.85
MID [23]	44.11	36.38	15.84
LiDiff [16]	42.22	32.25	10.80
LiDiff (Refined) [16]	40.82	36.08	21.34
ScoreLiDAR	36.82	25.49	9.70
ScoreLiDAR (Refined)	33.29	28.60	15.95

Table 9. The IoU evaluation results on the KITTI-360 dataset.

4.6. Visualization of key points

To validate the feasibility of our proposed key point selection method, we visualized the selected key points in the ground truth scene. As shown in Fig. 1, the red key points are mostly distributed on walls, traffic cones, cars, and corners, while smooth areas such as the road surface have no key points. These key points are crucial for expressing the details of 3D LiDAR scenes. Selecting these points to compute the point-wise loss allows the student model to more easily capture the relative configuration information between key points, thereby better completing key objects in the scene.

4.7. Experiments on semantic scene completion

The objective of this paper is to propose a foundational method for distillation acceleration applicable to various LiDAR scene completion diffusion models. Semantic scene completion is not within the scope of the fundamental experiments considered in this paper. In spite of this, to validate the generalizability of the proposed method, we use SemCity [8] as the teacher model to verify the effectiveness of ScoreLiDAR on semantic scene completion tasks. Because the pre-trained models and code for the metric computation of SemCity are not publicly available, we retrained SemCity based on the official implementation and repro-

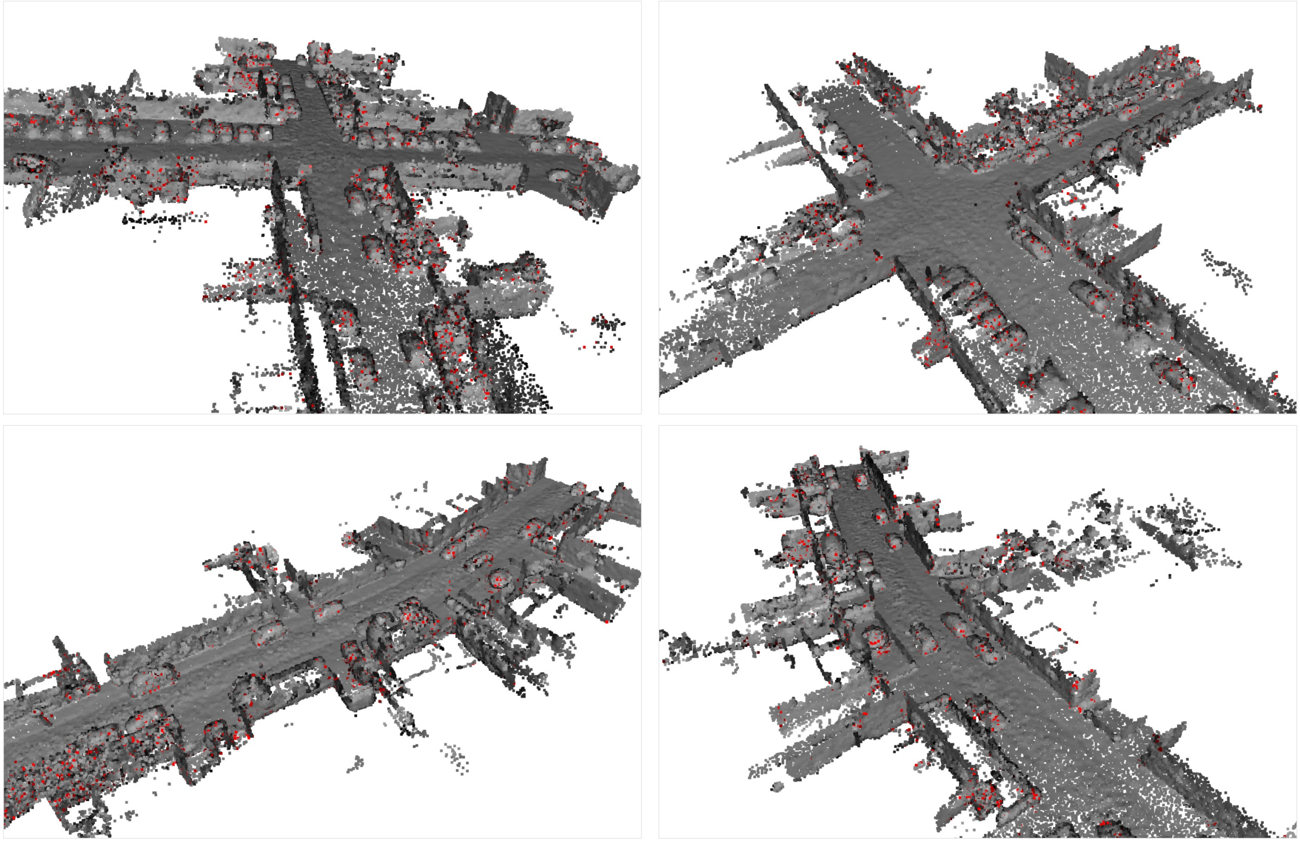

Figure 1. The visualization of the selected key points. Red points refer to the key points selected by the proposed method.

Model	FID ↓	KID ↓
SemCity [8]	88.52	0.11
ScoreLiDAR	81.76	0.09

Table 10. Results of semantic scene completion. Our ScoreLiDAR shows better performance.

duced the metric computation ourselves. Tab. 10 shows the results. In semantic scene completion tasks, the proposed ScoreLiDAR still shows better completion quality than that of the teacher model.

4.8. Experiments on scene occupancy

We calculate the Intersection-Over-Union (IoU) [20] to evaluate the occupancy of the completed scene compared with the ground truth scene. IoU represents the degree of overlap between the voxels in the completed scene and those in the ground truth scene. A higher IoU value indicates a higher completeness of the completed scene. During the evaluation, we considered three different voxel resolutions: $0.5m$, $0.2m$, and $0.1m$. The smaller the voxel resolution, the more fine-grained details are considered in the evaluation metrics, and vice versa. However, IoU is a voxel-based metric, and in some voxel-based LiDAR scene completion methods, it can serve as an accurate measure of completion quality. In contrast, ScoreLiDAR is a point cloud-based completion method, which differs from traditional voxel-based approaches. As a result, IoU may introduce bias when evaluating the completion quality. Therefore, here we provide IoU results solely as a relative reference.

Tab. 8 and Tab. 9 show the IoU of ScoreLiDAR and existing models. Under low voxel resolutions, ScoreLiDAR achieves comparable IoU values, meaning ScoreLiDAR generates dense and accurate point clouds. When the voxel resolutions become higher, the performance of ScoreLiDAR declines. As mentioned above, the existing method is mainly based on signed distance fields, which implement the scene completion using a voxel representation. ScoreLiDAR is point-level scene completion with the input of point clouds obtained from LiDAR scans, which works better at smaller voxel resolutions.

5. Theoretical demonstration

The gradient of the student model is Eq. (7).

$$\begin{aligned}&\nabla_\eta D_{\mathrm{KL}}\left(p_G^t\left(\mathcal{G}^t\right)\|q^t\left(\mathcal{G}^t\right)\right)\\&=\mathbb{E}_{t,\epsilon}\left[\nabla_{\mathcal{G}^t}\log p_G^t\left(\mathcal{G}^t\right)-\nabla_{\mathcal{G}^t}\log q^t\left(\mathcal{G}^t\right)\right]\frac{\partial\mathcal{G}^t}{\partial\eta}\end{aligned}\tag{7}$$

As proposed in ScoreSDE [21], the log likelihood $\nabla_{\mathcal{G}^t}\log q^t\left(\mathcal{G}^t\right)$ can be approximated by the predicted noise $\hat{\epsilon}$ with $\nabla_{\mathcal{G}^t}\log q^t\left(\mathcal{G}^t\right)\approx-\frac{\hat{\boldsymbol{\epsilon}}}{\sqrt{1-\bar{\alpha}^t}}$. Thus, the gradient in Eq. (7) can be written as

$$\begin{aligned}&\nabla_\eta D_{\mathrm{KL}}\left(p_G^t\left(\mathcal{G}^t\right)\|q^t\left(\mathcal{G}^t\right)\right)\\&=\mathbb{E}_{t,\epsilon}\left[\nabla_{\mathcal{G}^t}\log p_G^t\left(\mathcal{G}^t\right)-\nabla_{\mathcal{G}^t}\log q^t\left(\mathcal{G}^t\right)\right]\frac{\partial\mathcal{G}^t}{\partial\eta}\\&\approx\mathbb{E}_{t,\epsilon}\left[-\frac{\boldsymbol{\epsilon}_\phi\left(\mathcal{G}^t,\mathcal{P},t\right)}{\sqrt{1-\bar{\alpha}^t}}-\left(-\frac{\boldsymbol{\epsilon}_\theta\left(\mathcal{G}^t,\mathcal{P},t\right)}{\sqrt{1-\bar{\alpha}^t}}\right)\right]\frac{\partial\mathcal{G}^t}{\partial\eta}\\&=\mathbb{E}_{t,\epsilon}\left[\frac{\boldsymbol{\epsilon}_\theta\left(\mathcal{G}^t,\mathcal{P},t\right)}{\sqrt{1-\bar{\alpha}^t}}-\frac{\boldsymbol{\epsilon}_\phi\left(\mathcal{G}^t,\mathcal{P},t\right)}{\sqrt{1-\bar{\alpha}^t}}\right]\frac{\partial\mathcal{G}^t}{\partial\eta}\end{aligned}\tag{8}$$

Here $\sqrt{1-\bar{\alpha}^t}$ can be ignored. Thus, the gradient of G_{stu} can be approximated by

$$\begin{aligned}&\nabla_\eta D_{\mathrm{KL}}\left(p_G^t\left(\mathcal{G}^t\right)\|q^t\left(\mathcal{G}^t\right)\right)\\&\approx\mathbb{E}_{t,\epsilon}\left[\boldsymbol{\epsilon}_\theta\left(\mathcal{G}^t,\mathcal{P},t\right)-\boldsymbol{\epsilon}_\phi\left(\mathcal{G}^t,\mathcal{P},t\right)\right]\frac{\partial\mathcal{G}^t}{\partial\eta}\end{aligned}\tag{9}$$

6. Introduction on utilized methods

6.1. Variational score distillation

Variational Score Distillation (VSD), proposed by ProlificDreamer [24], is designed to leverage a pre-trained diffusion model to train a NeRF [15], enabling the rendering of high-quality 3D objects.

Given a text prompt y, the probabilistic distribution of all possible 3D representations can be modeled as a probabilistic density $\mu(\theta\|y)$ by a NeRF model parameterized by θ. Let $q_0^\mu(\boldsymbol{x}_0\|c,y)$ as the distribution of the rendered image $\boldsymbol{x}_0$ of NeRF given the camera c, and $p_0(\boldsymbol{x}_0\|y)$ as the distribution of the pre-trained text-to-image diffusion model at $t=0$. To generate high-quality 3D objects, ProlificDreamer [24] optimizes the distribution of μ by minimizing the following KL divergence

$$\min_\mu D_{\mathrm{KL}}\left(q_0^\mu\left(\boldsymbol{x}_0\mid y\right)\|p_0\left(\boldsymbol{x}_0\mid y\right)\right)\tag{10}$$

However, directly solving this variational inference problem is challenging because p_0 is complex, and its high-density regions may be extremely sparse in high-dimensional spaces. Therefore, ProlificDreamer reformulates it as an optimization problem at different time steps t, referring to these problems as Variational Score Distillation (VSD),

$$\min_\mu\mathbb{E}_{t,c}\left[\left(\sigma_t/\alpha_t\right)\omega(t)D_{\mathrm{KL}}\left(q_t^\mu\left(\boldsymbol{x}_t\mid c,y\right)\|p_t\left(\boldsymbol{x}_t\mid y\right)\right)\right]\tag{11}$$

Theorem 1 in [24] proves that introducing the additional t does not affect the global optimum of Eq. (10). Theorem 2 in [24] provides the method for optimizing the problem in Eq. (11).

$$\begin{aligned}\frac{\mathrm{d}\theta_\tau}{\mathrm{d}\tau}=&-\mathbb{E}_{t,\epsilon,c}[\omega(t)(\underbrace{-\sigma_t\nabla_{\boldsymbol{x}_t}\log p_t\left(\boldsymbol{x}_t\mid y\right)}_{\text{score of noisy real images}}\\&-\underbrace{\left(-\sigma_t\nabla_{\boldsymbol{x}_t}\log q_t^{\mu_\tau}\left(\boldsymbol{x}_t\mid c,y\right)\right)}_{\text{score of noisy rendered images}})\frac{\partial\boldsymbol{g}\left(\theta_\tau,c\right)}{\partial\theta_\tau}]\end{aligned}\tag{12}$$

Here the score of noisy real images is approximated by the pre-trained diffusion model $\boldsymbol{\epsilon}_{pretrain}(\boldsymbol{x}_t,t,y)$ and the score of noisy rendered images is approximated by another diffusion model $\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t,t,c,y)$, which is trained on the rendered images with the standard diffusion objective.

$$\min_\phi\sum_{i=1}^n\mathbb{E}_{t,\epsilon,c}\left[\left\|\boldsymbol{\epsilon}_\phi\left(\alpha_t\boldsymbol{g}\left(\theta^{(i)},c\right)+\sigma_t\boldsymbol{\epsilon},t,c,y\right)-\boldsymbol{\epsilon}\right\|_2^2\right]\tag{13}$$

In practice, $\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t,t,c,y)$ is parameterized by a small UNet or the Low-rank adaptation (LoRA) [5] of the teacher model. With the alternating training of NeRF and $\boldsymbol{\epsilon}_\phi(\boldsymbol{x}_t,t,c,y)$, ProlificDreamer [24] is ultimately able to generate high-quality 3D objects.

6.2. MinkowskiEngine

Sparse tensor computation plays a critical role in fields such as 3D point cloud processing, computer vision, and physical simulations. Unlike dense tensors, sparse tensors contain a high proportion of zero values and directly applying traditional tensor operations can lead to inefficient use of computational resources. Minkowski Engine [3] addresses these challenges by providing a high-performance framework tailored for sparse tensor computation, enabling efficient operations on high-dimensional sparse data. In this paper, we used the Minkowski Engine to process sparse point cloud data.

Minkowski Engine introduces several innovative approaches to sparse tensor processing.

- Efficient Sparse Tensor Representation. Sparse tensors are represented using coordinate-value pairs, eliminating the need to store zeros. This representation reduces both memory usage and computational overhead.
- Sparse Convolution Operations The framework supports high-dimensional sparse convolutions, with kernels designed to adapt to varying sparsity patterns. Optimized memory access patterns and parallel computation strategies ensure high efficiency.

- Fast Coordinate Mapping Minkowski Engine employs hash tables for rapid coordinate mapping, which accelerates tensor indexing and sparse pattern matching.
- Automatic Differentiation Support The framework includes built-in support for automatic differentiation, facilitating the training of machine learning models based on sparse tensors.
- Multi-Dimensional Capability Minkowski Engine can handle sparse tensors of arbitrary dimensions, making it suitable for a wide range of applications, from 2D image processing to 5D simulations.

Minkowski Engine has been widely adopted in various domains including 3D point cloud processing, physical simulations and medical imaging. By significantly improving computational efficiency and scalability, the Minkowski Engine has become a preferred choice for handling sparse tensor computations in both research and industrial applications.

7. Ethical statement

The potential ethical impact of our work is about fairness. As "human" is included as a kind of object in the LiDAR scene, when performing scene completion, it may be necessary to complete human figures. Human-related objects may have data bias related to fairness issues, such as the bias to gender or skin colour. Such bias can be captured by the student model in the training.

7.1. Notification to human subjects

In our user study, we present the notification to subjects to inform the collection and use of data before the experiments.

> Dear volunteers, we would like to thank you for supporting our study. We propose ScoreLiDAR, a novel distillation method tailored for 3D LiDAR scene completion, which introduces a structural loss to help the student model capture the geometric structure information. All information about your participation in the study will appear in the study record. All information will be processed and stored according to the local law and policy on privacy. Your name will not appear in the final report. Only an individual number assigned to you is mentioned when referring to the data you provided.
>
> We respect your decision whether you want to be a volunteer for the study. If you decide to participate in the study, you can sign this informed consent form.

The Institutional Review Board approved the use of users' data of the main authors' affiliation.

8. Failure examples

Fig. 4 presents some failure cases of ScoreLiDAR. From these examples, it can be observed that ScoreLiDAR exhibits over-completion to some extent, where regions that do not exist are completed. Before the completion, as mentioned in Sec.3 in the main paper, the number of points of the input sparse scan $\mathcal{P}$ is increased by concatenating its points K times and the dense input $\mathcal{P}^*$ is obtained. If the number of points of $\mathcal{P}^*$ exceeds the actual number of points in the ground truth, it can lead to redundant points in the completed scene. These redundant points may be distributed in areas that do not require completion, resulting in the situations observed in the failure cases.

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A Dataset For Semantic Scene Understanding Of Lidar Sequences. In *ICCV*, pages 9297–9307, 2019. 1

[2] M Akmal Butt and Petros Maragos. Optimum Design of Chamfer Distance Transforms. *TIP*, 7(10):1477–1484, 1998. 1

[3] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *CCPR*, pages 3075–3084, 2019. 7

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *NIPS*, 33:6840–6851, 2020. 2, 3

[5] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-Rank Adaptation Of Large Language Models. In *ICLR*, 2021. 7

[6] Jaehoon Jung, Michael J Olsen, David S Hurwitz, Alireza G Kashani, and Kamilah Buker. 3D Virtual Intersection Sight Distance Analysis Using LiDAR Data. *Transportation research part C: emerging technologies*, 86:563–579, 2018. 3

[7] Jumin Lee, Woobin Im, Sebin Lee, and Sung-Eui Yoon. Diffusion Probabilistic Models For Scene-Scale 3d Categorical Data. *arXiv preprint arXiv:2301.00527*, 2023. 3

[8] Jumin Lee, Sebin Lee, Changho Jo, Woobin Im, Juhyeong Seon, and Sung-Eui Yoon. Semcity: Semantic Scene Generation With Triplane Diffusion. In *CVPR*, pages 28337–28347, 2024. 5, 6

[9] Pengfei Li, Ruowen Zhao, Yongliang Shi, Hao Zhao, Jirui Yuan, Guyue Zhou, and Ya-Qin Zhang. Lode: Locally Conditioned Eikonal Implicit Scene Completion From Sparse Lidar. In *ICRA*, pages 8269–8276, 2023. 3, 5

[10] You Li, Julien Moreau, and Javier Ibanez-Guzman. Emergent Visual Sensors For Autonomous Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 24(5):4716–4737, 2023. 3

[11] Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao, Jose M Alvarez, Sanja Fidler, Chen Feng, and Anima Anandkumar. Voxformer: Sparse Voxel Transformer For Camera-based 3d Semantic Scene Completion. In *CVPR*, pages 9087–9098, 2023. 3

[12] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A Novel Dataset And Benchmarks For Urban Scene understanding in 2d And 3d. *PAMI*, 45(3):3292–3310, 2022. 1

[13] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-Instruct: A Universal Approach for Transferring Knowledge From Pre-trained Diffusion Models. *NIPS*, 36:76525–76546, 2023. 2

[14] María Luisa Menéndez, JA Pardo, L Pardo, and MC Pardo. The Jensen-Shannon Divergence. *Journal of the Franklin Institute*, 334(2):307–318, 1997. 2

[15] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 7

[16] Lucas Nunes, Rodrigo Marcuzzi, Benedikt Mersch, Jens Behley, and Cyrill Stachniss. Scaling Diffusion Models To Real-World 3D LiDAR Scene Completion. In *CVPR*, pages 14770–14780, 2024. 2, 3, 4, 5

[17] Haoxi Ran, Vitor Guizilini, and Yue Wang. Towards Realistic Scene Generation With LiDAR Diffusion Models. In *CVPR*, pages 14738–14748, 2024. 3

[18] Luis Roldao, Raoul de Charette, and Anne Verroust-Blondet. Lmscnet: Lightweight Multiscale 3d Semantic Completion. In *Proceedings of International Conference on 3D Vision*, pages 111–119, 2020. 5

[19] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. *arXiv preprint arXiv:2010.02502*, 2020. 3

[20] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic Scene Completion From A Single Depth Image. In *CVPR*, pages 1746–1754, 2017. 6

[21] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *ICLR*, 2021. 7

[22] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models. In *ICML*, pages 32211–32252, 2023. 2, 3

[23] Ignacio Vizzo, Benedikt Mersch, Rodrigo Marcuzzi, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. Make It Dense: Self-Supervised Geometric Scan Completion of Sparse 3d Lidar Scans In Large Outdoor Environments. *IRAL*, 7(3):8534–8541, 2022. 3, 5

[24] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. ProlificDreamer: High-Fidelity And Diverse Text-to-3D Generation With Variational Score Distillation. *NIPS*, 36:8406–8441, 2023. 7

[25] Yuwen Xiong, Wei-Chiu Ma, Jingkang Wang, and Raquel Urtasun. Learning Compact Representations for Lidar Completion and Generation. In *CVPR*, pages 1074–1083, 2023. 2

[26] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved Distribution Matching Distillation For Fast Image Synthesis. *NIPS*, 37:47455–47487, 2024. 2

[27] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step Diffusion With Distribution Matching Distillation. In *CVPR*, pages 6613–6623, 2024.

[28] Shengyuan Zhang, Ling Yang, Zejian Li, An Zhao, Chenye Meng, Changyuan Yang, Guang Yang, Zhiyuan Yang, and Lingyun Sun. Distribution Backtracking Builds A Faster Convergence Trajectory for One-step Diffusion Distillation. In *ICLR*, 2025. 2

[29] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d Shape Generation And Completion Through Point-voxel Diffusion. In *ICCV*, pages 5826–5835, 2021. 5

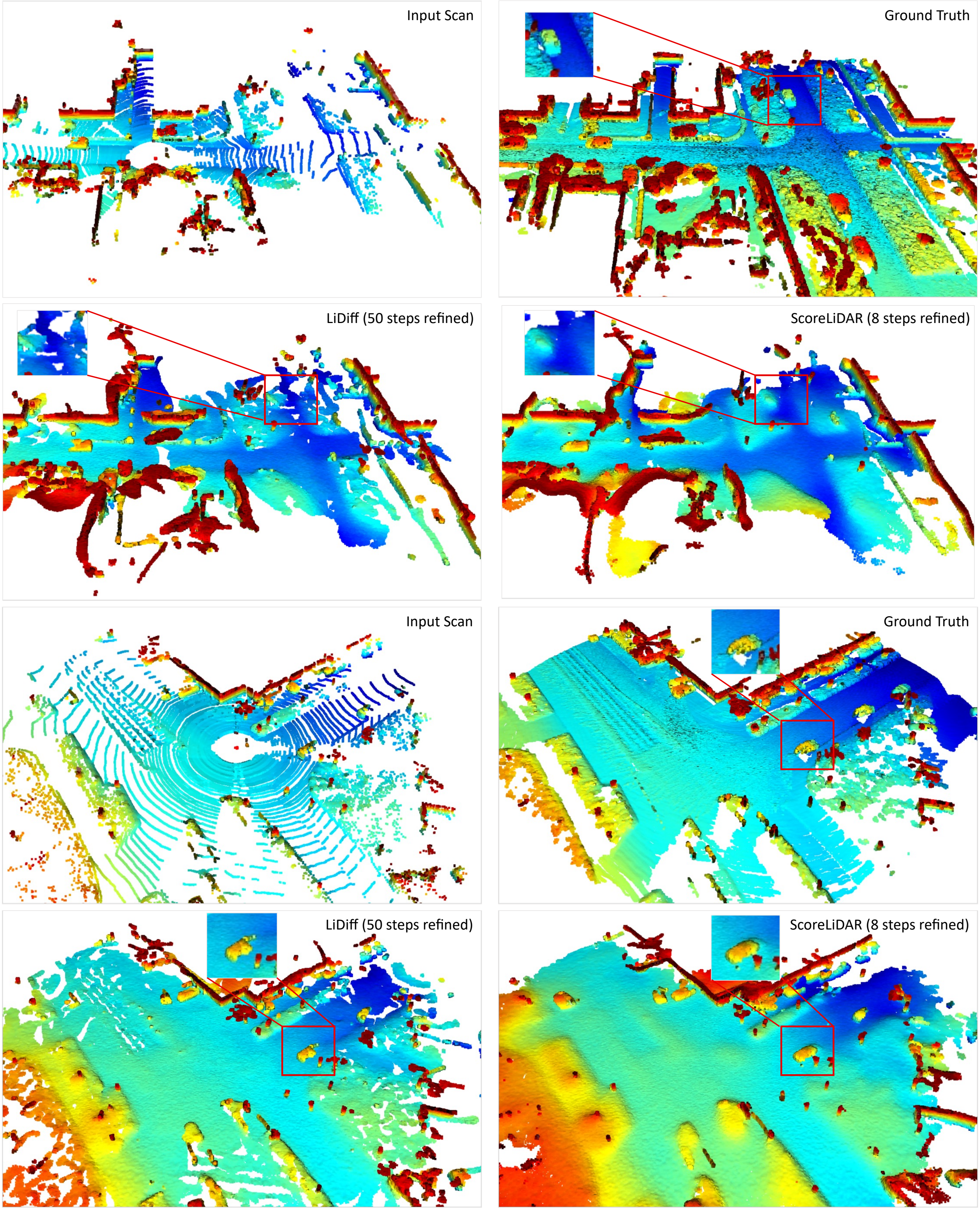
Input Scan
Ground Truth
LiDiff (50 steps refined)
ScoreLiDAR (8 steps refined)
Input Scan
Ground Truth
LiDiff (50 steps refined)
ScoreLiDAR (8 steps refined)

Figure 2. Completed samples of ScoreLiDAR from KITTI-360 dataset.

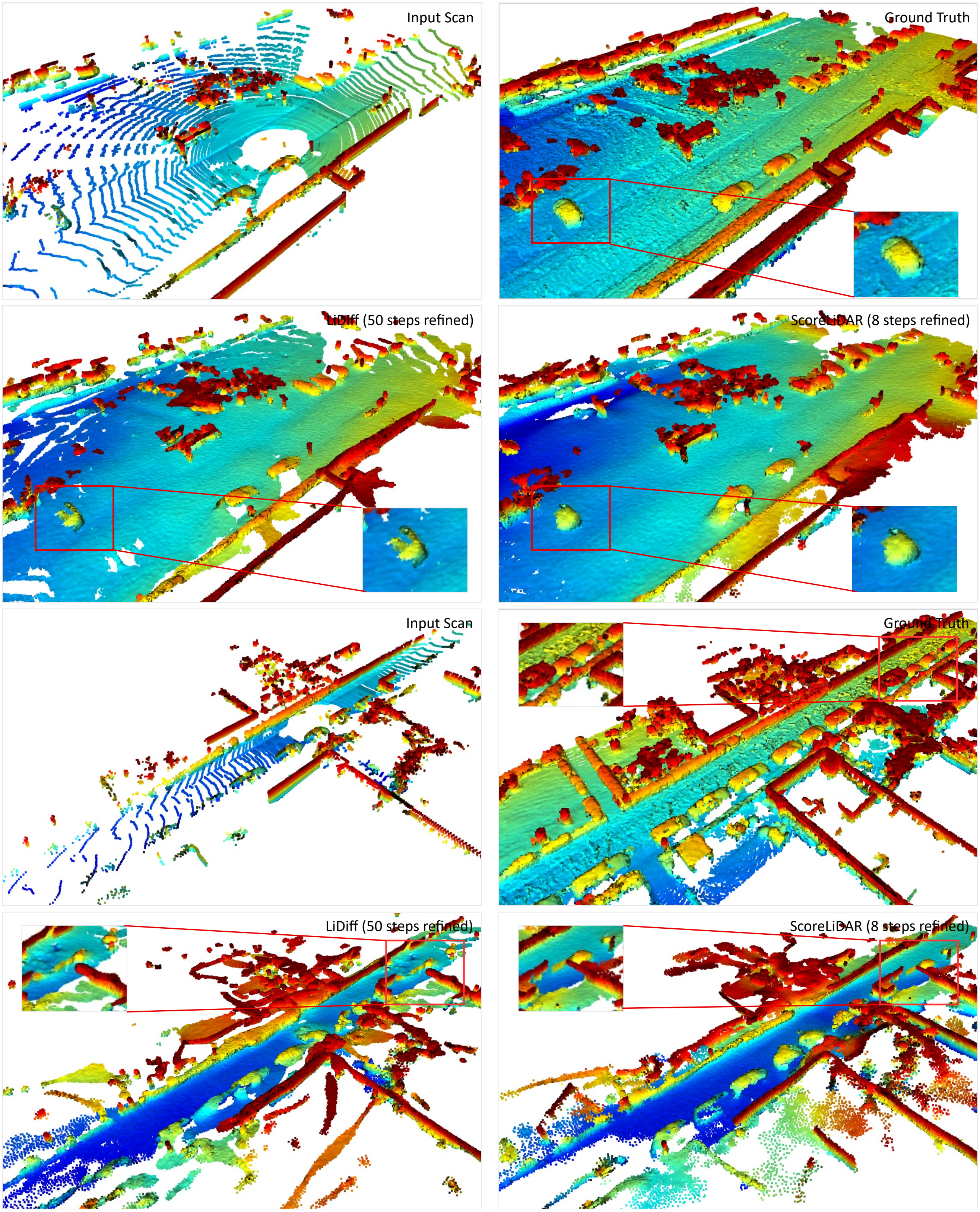
Input Scan
Ground Truth
LiDiff (50 steps refined)
ScoreLiDAR (8 steps refined)
Input Scan
Ground Truth
LiDiff (50 steps refined)
ScoreLiDAR (8 steps refined)

Figure 3. Completed samples of ScoreLiDAR from SemanticKITTI dataset.

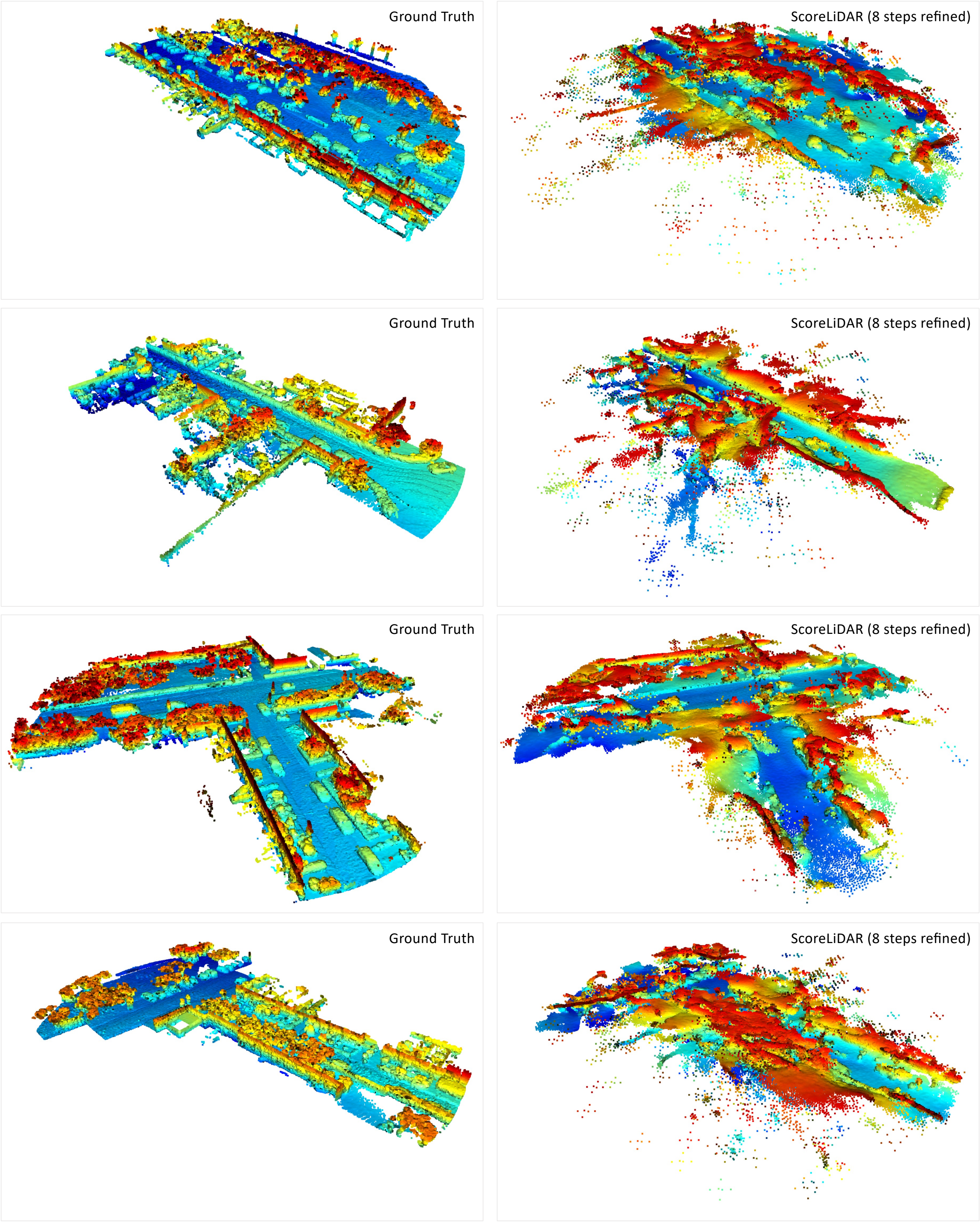
Ground Truth
ScoreLiDAR (8 steps refined)
Ground Truth
ScoreLiDAR (8 steps refined)
Ground Truth
ScoreLiDAR (8 steps refined)
Ground Truth
ScoreLiDAR (8 steps refined)

Figure 4. Failure examples of ScoreLiDAR.