SuperPC: A Single Diffusion Model for Point Cloud Completion, Upsampling, Denoising, and Colorization

Supplementary Material

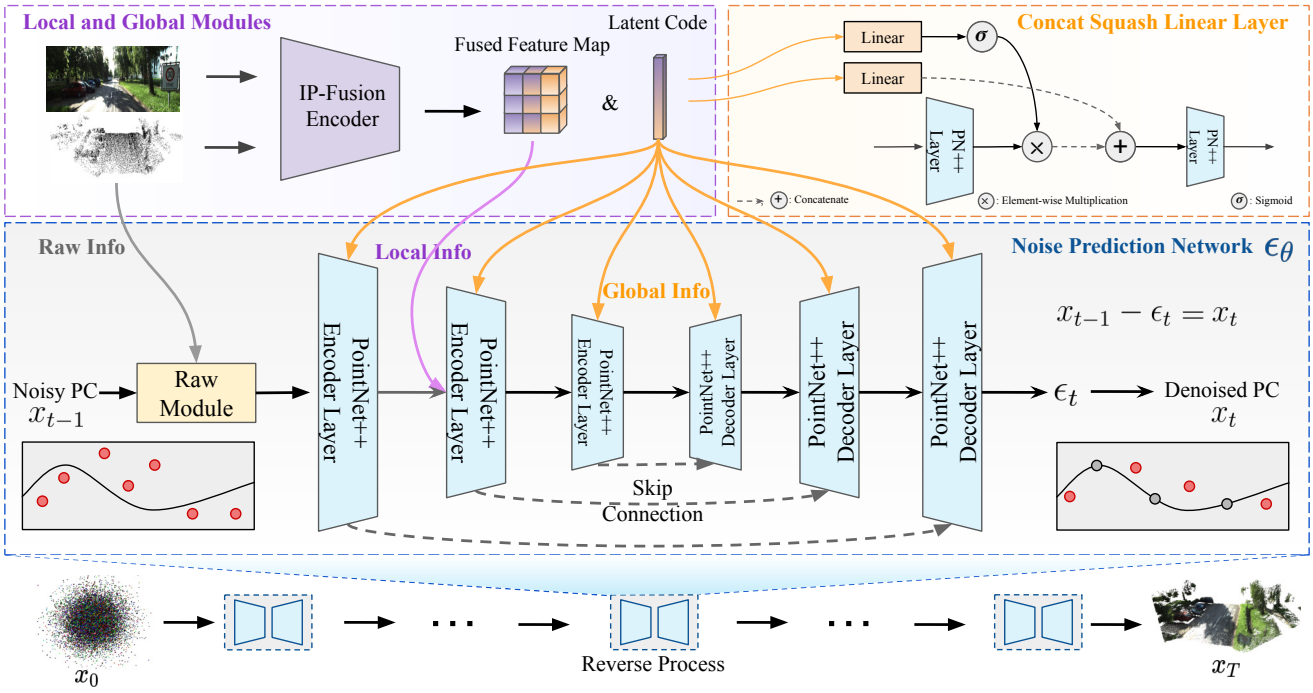

Figure 6. The detailed structure of the SuperPC network.

A. Model Structure Details

The SuperPC Model's detailed structure, including the core diffusion noise prediction network, is shown in Figure 6. Moreover, it shows how information from raw, local, and global modules is integrated into the main network. We delve into the detailed explanation of them in this section.

For the core noise prediction network, we use the PointNet++ [45] architecture as the backbone. However, we deviate from its original configuration, opting instead for the modified version introduced in PDR [36] and DDPMPU [47]. This adaptation is due to the original PointNet++ network's inability to effectively process point features from clouds resembling Gaussian noise. The network comprises a three-level PointNet++ encoder and decoder structure. In the encoding stages, we set the number of neighbors, K, as 16 for set abstraction purposes. In the decoding stages, K is set to 8 to facilitate feature propagation.

Since we have already detailed how raw information is incorporated into the core network in Section 4.2, we will next explain how information from local and global levels is integrated into the core network. After passing the input image and point cloud into the local and global modules, we get the local feature map and the global latent code. The local feature map $M(N_l, 3+C_l)$ can essentially be viewed as N_l points each possessing $3+C_l$ features, which shares the same format as the output $P_1(N_1, 3+C_1)$ of the core network's first layer. We employ the point spatial interpolation method mentioned in Section 4.2 to align the local information from the local feature map with the output of the first layer of the core network. The global latent code $\boldsymbol{z}(1, 1024)$ is added to the core network via the concatsquash layer [17] suggested by Luo [34], which is defined as:

$$P_1' = CS(P_1, t, \boldsymbol{z}) = P_1 \odot \sigma(\boldsymbol{W_1 c} + \boldsymbol{b_1}) + \boldsymbol{W_2 c}, \quad (7)$$

where P_1 represents the input to the layer, while P_1' signifies the output. Here, $\boldsymbol{c} = [t, \sin(t), \cos(t), \boldsymbol{z}]$ constitutes the context vector including the embedding of time t and the global latent code $\boldsymbol{z}$, with σ indicating the sigmoid function. The parameters $\boldsymbol{W_1}, \boldsymbol{W_2}$ and $\boldsymbol{b_1}$ are all subject to optimization during training. The code with detailed implementation will be released upon the paper's acceptance.

B. Benchmark Details

B.1. ShapeNet

For the object-level benchmark, we render the images of the 3D objects in the ShapeNet [7] following the setting and

method provided by [11, 68] since SuperPC integrates the image information. In addition to the rendered images, we sample the 3D models from the ShapeNet Core Dataset to create ground truth point clouds that are paired with the images. We appreciate the perspective of the ShapeNet55/34 dataset [72] on the necessity of assessing model generalization performance across unseen categories. We chose thirteen categories used by [11, 68] from the ShapeNet Core dataset. Ten of these categories are designated as 'seen' for training and validation, with the remaining three categories earmarked as 'unseen' for testing, to evaluate the models' ability to generalize. Each ground truth point cloud is standardized to contain 8,192 points, following the specifications stated in the ShapeNet55/34 dataset [72].

B.2. TartanAir

Existing point cloud processing datasets typically concentrate on simple, synthetic objects, which are insufficient to verify the performance of models in complex scenarios. To bridge this gap and demonstrate the effectiveness of our work, we introduce a scene-level benchmark utilizing the TartanAir dataset [63], aimed at evaluating model performance in complex environments. It features fifteen diverse indoor and outdoor environments, covering different seasons and lighting conditions, and is derived from 176 sequences totaling over 600,000 frames. This results in a comprehensive dataset of 85,618 point clouds paired with images, providing a robust benchmark for assessing point cloud processing tasks. Most existing point cloud processing datasets focus solely on simple, virtual objects. However, in real-world applications, the point clouds we often need to process consist of complex scenes with many objects. Therefore, we propose constructing a scene-level benchmark based on the TartanAir [63] dataset to evaluate the effectiveness of models and methods in performing point cloud processing tasks within complex scenarios. TartanAir provides the RGB and Depth images in eighteen photo-realistic simulation environments. We generate the raw point clouds based on the RGBD images with a depth-limit truncation to remove those points with huge depth values like the points representing the sky. These raw point clouds are downsampled to 46080 points to serve as the ground truth, accommodating the memory constraints of training baseline models [35, 72] and meeting the requirements for Earth Mover's Distance (EMD) calculations.

B.3. KITTI-360

Although the TartanAir Benchmark provides data for evaluating scene-level performance, it is based on simulations. To better validate the effectiveness of our method in real-world scenarios, we also include a real-world, scene-level point cloud processing benchmark based on KITTI-360 [31]. It provides high-quality images and accurate accumulated point clouds. We stitch together the accumulated point clouds from each sequence to create dense global maps and, based on the pose information, crop out dense local point clouds from these maps. Each local point cloud is then downsampled to 46,080 points to serve as ground truth and matched with the corresponding frame's image to form the KITTI-360 benchmark dataset used in this work.

C. Metrics Details

C.1. Density-aware Chamfer Distance

DCD [65] improves the evaluation of visual quality for 3D shape generation tasks by considering the density of points in a point cloud, unlike the traditional Chamfer Distance. DCD's formulation takes into account both the point-to-point distances and the point densities, providing a more discriminative measure. The DCD between two point clouds S_1 and S_2 is given by the following equation:

$$d_{DCD}(S_1, S_2) = \frac{1}{2}\left(\frac{1}{|S_1|}\sum_{x\in S_1}\min_{y\in S_2}\left(1-e^{-\alpha||x-y||^2}\right) + \frac{1}{|S_2|}\sum_{y\in S_2}\min_{x\in S_1}\left(1-e^{-\alpha||x-y||^2}\right)\right) \tag{8}$$

In this formula, S_1 and S_2 are the two sets of points that represent point clouds. The cardinalities $|S_1|$ and $|S_2|$ indicate the number of points in each set. The variables x and y correspond to the points in S_1 and S_2, respectively. The term $||x-y||^2$ is the squared Euclidean distance between the points x and y. The exponential term $e^{-\alpha||x-y||^2}$ is used to calculate a distance that is sensitive to the point density, with α acting as a temperature scalar that influences the sensitivity of the distance to point density variations. The minimum function $\min$ finds the nearest neighbor distance, ensuring that each point in one set is compared to its closest point in the other set. This formulation indicates that the DCD is not just the average nearest neighbor distance but also incorporates a normalization based on the local density of points, which helps to prevent the measure from being too sensitive to outliers and provides a better representation of the actual shape and structure of the point clouds.

D. Additional Experiments

In the main text, we have already discussed the performance and generalization experiments. To provide a more comprehensive evaluation, we will provide (1) the ablation study in Appendix D.1, (2) the complexity analysis in Appendix D.2, (3) the comparison between the single unified model and different combinations of multiple models for individual tasks with the same SuperPC framework in Appendix D.3, (4) an evaluation of different integration orders when combining SOTA methods in Appendix D.4, and (5) the colorization task results in Appendix D.5.

Table 3. Ablation Study of the Image-Point Fusion and Three-Level-Info conditions.

Fusion Stage		Condition Module			ShapeNet [7]			TartanAir [63]			KITTI-360 [31]		
Early	Deep	Raw	Local	Global	DCD(↓)	EMD(↓)	F1(↑)	DCD(↓)	EMD(↓)	F1(↑)	DCD(↓)	EMD(↓)	F1(↑)
×	✓	✓	✓	✓	0.661	8.59	0.254	0.822	11.73	0.183	0.935	20.89	0.177
✓	×	✓	✓	✓	0.623	8.24	0.295	0.796	11.45	0.239	0.896	20.56	0.203
✓	✓	×	✓	✓	0.648	8.31	0.263	0.811	12.68	0.201	0.925	23.79	0.181
✓	✓	✓	×	✓	0.594	7.79	0.375	0.658	9.68	0.319	0.794	17.28	0.227
✓	✓	✓	✓	×	0.693	8.96	0.248	0.697	10.43	0.298	0.852	19.71	0.205
✓	✓	✓	✓	✓	**0.476**	**2.21**	**0.409**	**0.558**	**3.527**	**0.384**	**0.681**	**9.58**	**0.365**

Table 4. Comparison between the single unified model and different combinations of multiple models for individual tasks (all the models use the same SuperPC framework for fairness).

Different Combinations	**ShapeNet** [7]			**TartanAir** [63]			**KITTI-360** [31]		
	DCD (↓)	EMD (↓)	F1 (↑)	DCD (↓)	EMD (↓)	F1 (↑)	DCD (↓)	EMD (↓)	F1 (↑)
SPC(U) + SPC(C) + SPC(D)	0.506	2.46	0.374	0.574	3.67	0.354	0.715	11.92	0.335
SPC(C) + SPC(D+U)	0.492	2.37	0.388	0.561	3.59	0.373	0.692	10.17	0.352
SPC(D) + SPC(C+U)	0.495	2.41	0.383	0.571	3.63	0.364	0.707	10.84	0.343
SPC(U) + SPC(C+D)	0.489	2.32	0.391	0.564	3.61	0.368	0.698	10.53	0.347
SPC(C+U+D)	**0.476**	**2.21**	**0.409**	**0.558**	**3.53**	**0.384**	**0.681**	**9.58**	**0.365**

D.1. Ablation Study

The ablation studies are performed to evaluate the effectiveness of the five critical components in our model: the dual-spatial early fusion, the attention-based deep fusion, the raw module, the local module, and the global module.

Early Fusion and Deep Fusion To demonstrate the importance of our spatial-mixed-fusion strategy, we conducted an ablation study by removing the image modality at two critical fusion stages: the early-fusion stage (image feature projection) and the deep-fusion stage (image encoder with the cross-attention module), as described in Section 4.2 and Section 4.3. As shown in Table 3, excluding either fusion stage results in a significant decline in overall performance across all three benchmarks, underscoring the importance of incorporating both the early fusion and the deep fusion.

Raw, local, and global module The evaluation of the three-level modules involves removing each of these components individually. Excluding any of these elements disrupts the integrity of the three-level-conditioned framework, leading to a marked deterioration in overall performance as shown in Table 3. This effect is most pronounced with the raw module, as its exclusion leads to a notable decline in performance. Generally speaking, every module plays a significant role in building the TLC framework and GMF strategy.

D.2. Complexity Analysis

To fulfill the goal of the combination task, previous single-task models [21, 35, 47, 72] could only be sequentially interpreted together to accomplish point cloud upsampling, completion, and denoising step by step. In contrast, our SuperPC is capable of completing the entire combination task within one single model. Therefore, theoretically, not only can it achieve higher performance as proven in Section 3.1, but it also requires less computational consumption and shorter inference time. As shown in Table 5, SuperPC demonstrates higher performance across all three metrics, along with lower FLOPs and inference time (t_{inf}) compared with the combination of the SOTAs [35, 47, 72] of the three single tasks, whether setting the reverse steps of the diffusion model to 50 or 100. Due to the principles of diffusion models, more reverse steps can improve the quality of inference but also require more computation and inference time. In practical applications, using 100 steps allows the model to generate high-quality point clouds within a relatively short inference time. Moving forward, we aim to further enhance the model's efficiency by either refining the point diffusion mechanism or replacing the current complex point cloud learning backbone [45] with the sparse-tensor-based backbone like Minkowski Engine [10].

Table 5. Complexity of SOTAs combination and SuperPC with different reverse steps. PU, PC, and PD are the SOTAs of denosing [35] , completion [72], and upsampling [47] on ShapeNet. All the results were tested on an NVIDIA GeForce RTX 3090 GPU.

Method	Params	FLOPs	t_{inf}	DCD(↓)
PD+PC+PU	**33.36 M**	593.6 G	3.92 s	0.462
SuperPC (50 steps)	36.78 M	**93.5 G**	**0.76 s**	0.441
SuperPC (100 steps)	36.78 M	183.4 G	1.38 s	0.412
SuperPC (1000 steps)	36.78 M	1809.6 G	14.69 s	**0.387**

Table 6. Results of different SOTAs integration methods on the combination task.

Task	Methods	ShapeNet [7]			TartanAir [63]			KITTI-360 [31]		
		DCD (↓)	EMD (↓)	F1 (↑)	DCD (↓)	EMD (↓)	F1 (↑)	DCD (↓)	EMD (↓)	F1 (↑)
Combination	PD→PC→PU	0.489	2.64	0.391	0.612	3.93	0.125	0.749	10.18	0.254
	PU→PD→PC	0.497	2.36	0.375	0.609	3.82	0.130	0.763	10.29	0.248
	PU→PC→PD	0.521	2.93	0.362	0.583	3.64	0.139	0.725	10.06	0.266
	SuperPC (ours)	**0.476**	**2.21**	**0.409**	**0.558**	**3.527**	**0.154**	**0.681**	**9.58**	**0.287**

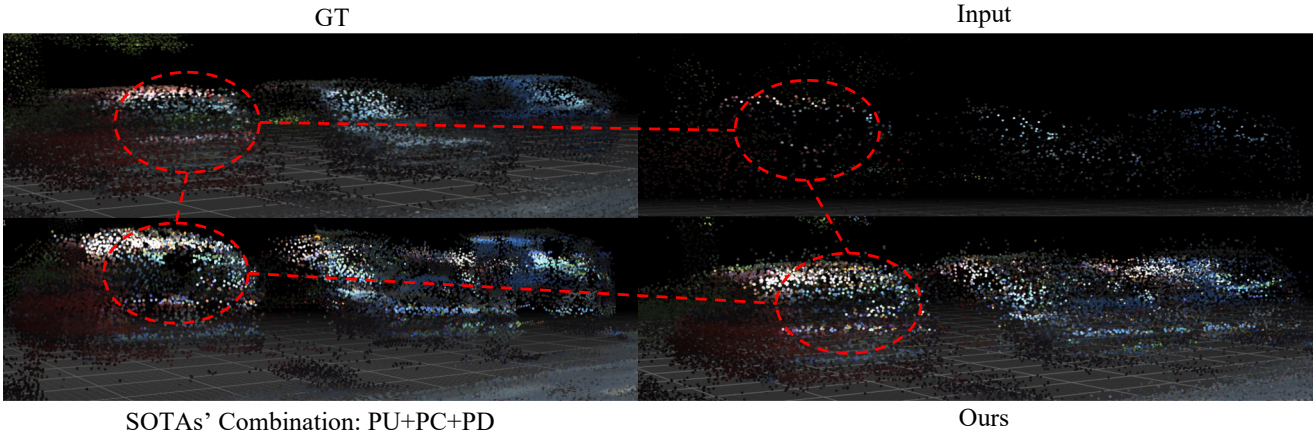

Figure 7. The quality results of SuperPC and SOTAs on the combination task.

D.3. Single model vs multiple-models-combination

Despite significant advancements, prior approaches [21, 32, 35, 72] predominantly tackle each of these tasks—completion, upsampling, denoising, and colorization—independently. However, such isolated strategies overlook the inherent interdependence among defects including incompleteness, low resolution, noise, and lack of color, which frequently coexist and influence one another.

Currently, there is no single model capable of addressing all four tasks simultaneously. A unified model offers not only computational efficiency but also the ability to prevent error accumulation across tasks while leveraging their interconnectivity to mutually enhance performance. For instance, as illustrated in Figure 7, errors from a completion model often propagate to subsequent upsampling. In addition to qualitative observations highlighting the limitations of combining multiple specialized models, we conducted extensive quantitative experiments to substantiate this claim. Specifically, we compared our single unified model with various combinations of multiple models for individual tasks, ensuring a fair comparison by implementing all models within the same SuperPC framework. As demonstrated in Table 4, the single unified model consistently outperforms all combinations across three benchmarks. These findings underscore the necessity of a single, integrated model capable of simultaneously addressing all four tasks.

D.4. Different SOTAs integration methods

In the combination task, we integrate the SOTA models [21, 35, 72] for each individual task in various reasonable sequences, as shown in Table 6. The sequence starting with upsampling (PU) [21], followed by completion (PC) [42], and ending with denoising (PD) [35] yields relatively better outcomes compared to other combinations in the scene-scale datasets - TartanAir and KITTI-360. However, the sequence of "PD→PC→PU" shows better performance on the object-level dataset - ShapeNet. Obviously, SuperPC surpasses all the integration methods across the three datasets.

Table 7. Colorization Experiment. MSE is used as the metric.

Methods	ShapeNet	TartanAir	KITTI-360
Learning-based [32]	0.0316	0.0429	0.0536
Geometry-based	0.0276	0.0131	0.0142
SuperPC	**0.0102**	**0.0117**	**0.0129**

D.5. Colorization Task

In the colorization task, we evaluate the qualitative performance of the SuperPC compared to the baseline model [32] and the SOTA - geometry-based method. The learning-based baseline model sometimes generates weird unreal colors as shown at the left bottom of Figure 9. The projection-based method exhibits limitations in rendering colors for obscured scenes. A specific instance highlighted in Figure 8 reveals its failure to accurately colorize grass hidden by a tree. In contrast, SuperPC effectively predicts the colors for occluded areas, producing the point cloud that closely aligns with the ground truth texture and colors. Additionally, as shown in Table 7, SuperPC outperforms both the learning-based and the geometry-based methods.

D.6. Experiment on Observation Incompleteness

We generate three distinct levels of observation incompleteness by stitching point clouds from one, three, and five adjacent frames, followed by cropping them to maintain consistency in camera pose and field of view. The PC completion performance of SuperPC is compared against the current SOTA method using the average results across these three levels of incompleteness, evaluated on two scene-level datasets: TartanAir [63] and KITTI-360 [31].

As shown in Table 8, SuperPC consistently outperforms the SOTA method across both evaluated datasets, demonstrating superior robustness and effectiveness in handling varying degrees of observation incompleteness.

Datasets	Methods	CD	DCD	EMD	F1
KITTI-360 [31]	LiDiff [42]	9.41	0.693	9.82	0.247
	SuperPC	**8.63**	**0.667**	**9.24**	**0.298**
TartanAir [63]	LiDiff [42]	7.91	0.631	4.52	0.296
	SuperPC	**7.04**	**0.597**	**4.15**	**0.327**

Table 8. Incomplete observations evaluation.

D.7. Combination Experiment on PCN Dataset

We present a brief performance comparison between SuperPC and SOTA methods combination on the PCN dataset [73]. As shown in Table 9 below, the results demonstrate that SuperPC significantly outperforms SOTA methods on the combination task, which is our main contribution.

Methods	CD	DCD	EMD	F1
[21]→[42]→[35]	11.03	0.495	3.44	0.592
SuperPC	**10.12**	**0.432**	**2.13**	**0.675**

Table 9. Combination task performance on PCN dataset.

E. More Qualitative Samples

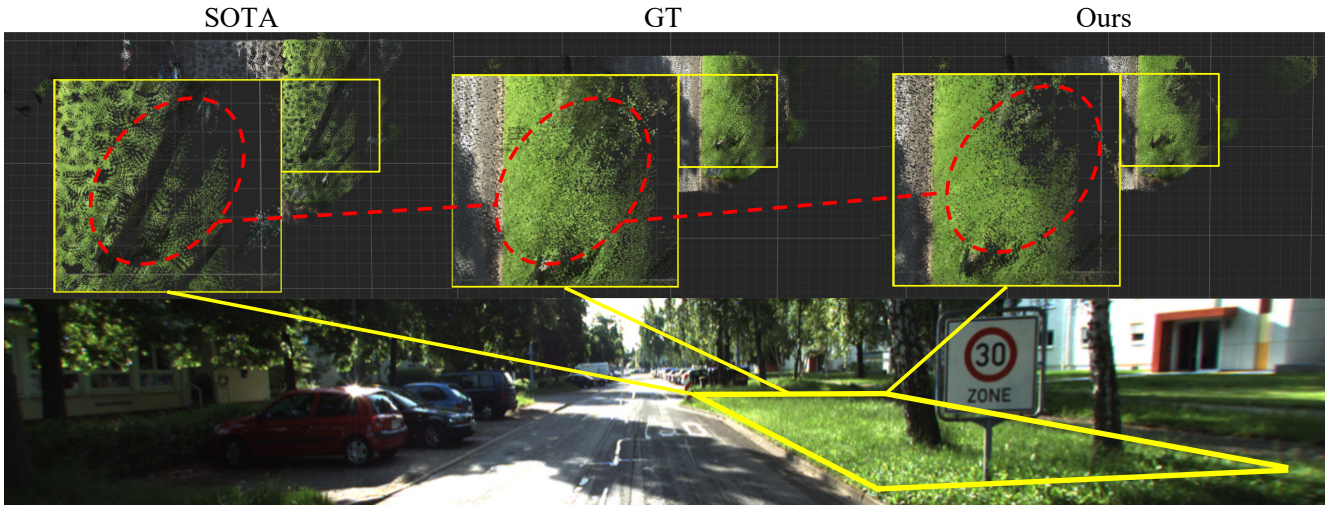

Figure 8. The quality results of SuperPC and SOTA (projection) on the point cloud colorization task with zoom in details on the generated green color of the grass field.

Figure 9. The quality results of SuperPC method and baseline learning method (PCCN [32]).

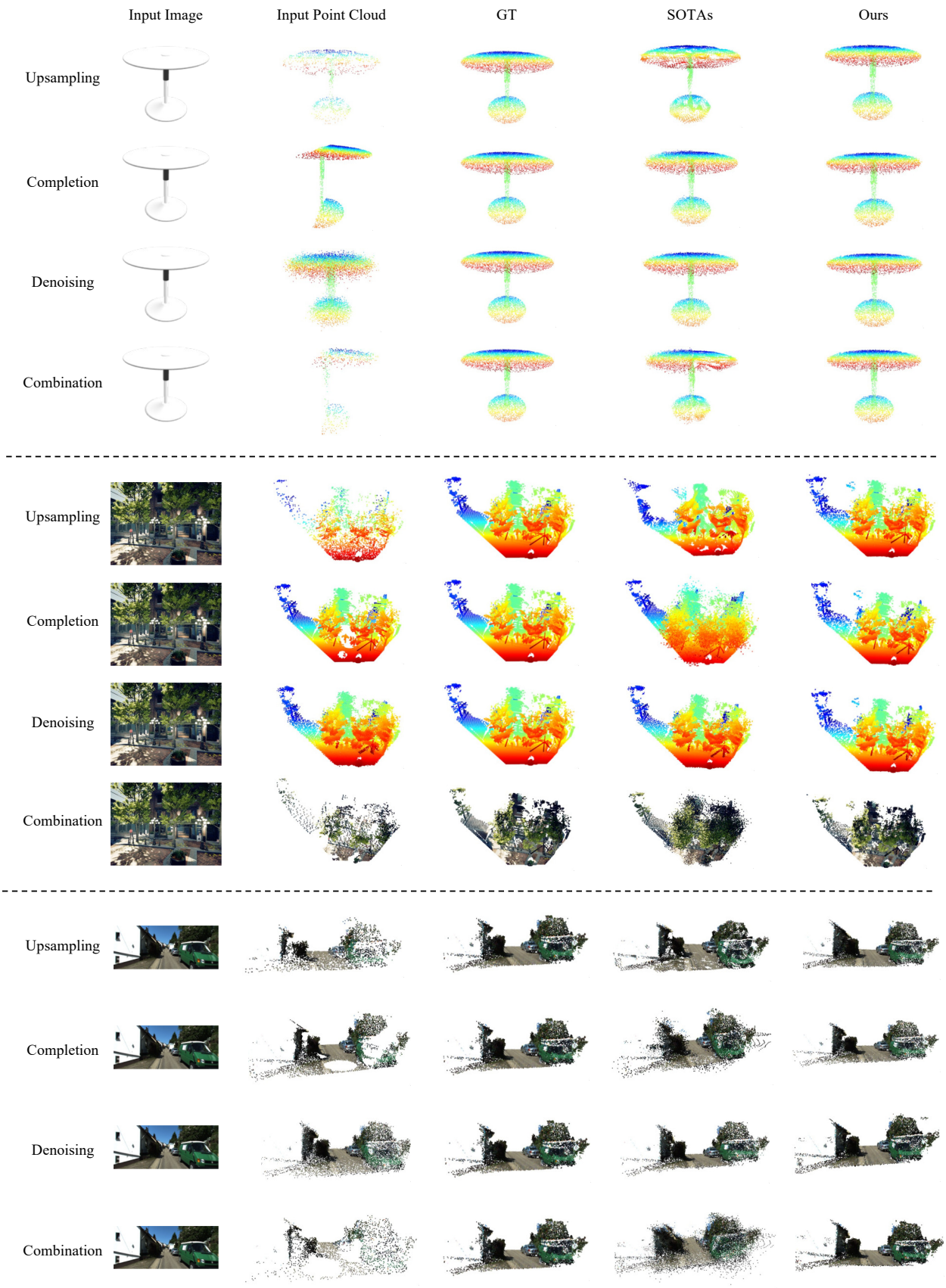

Figure 10. More qualitative results on the ShapeNet, TartanAir, and KITTI-360 dataset.

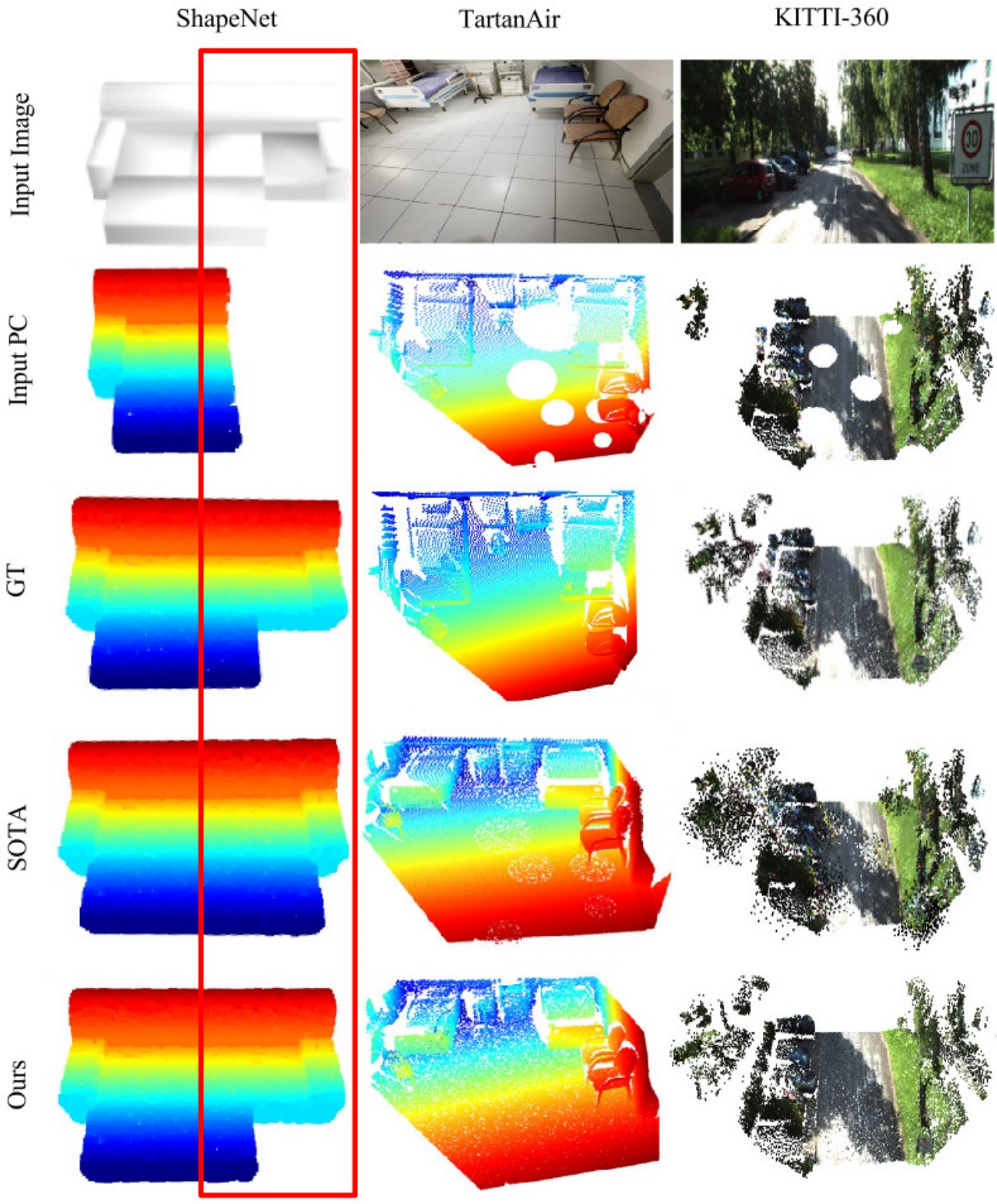

Figure 11. Zoom-in figure of the completion task qualitative results.

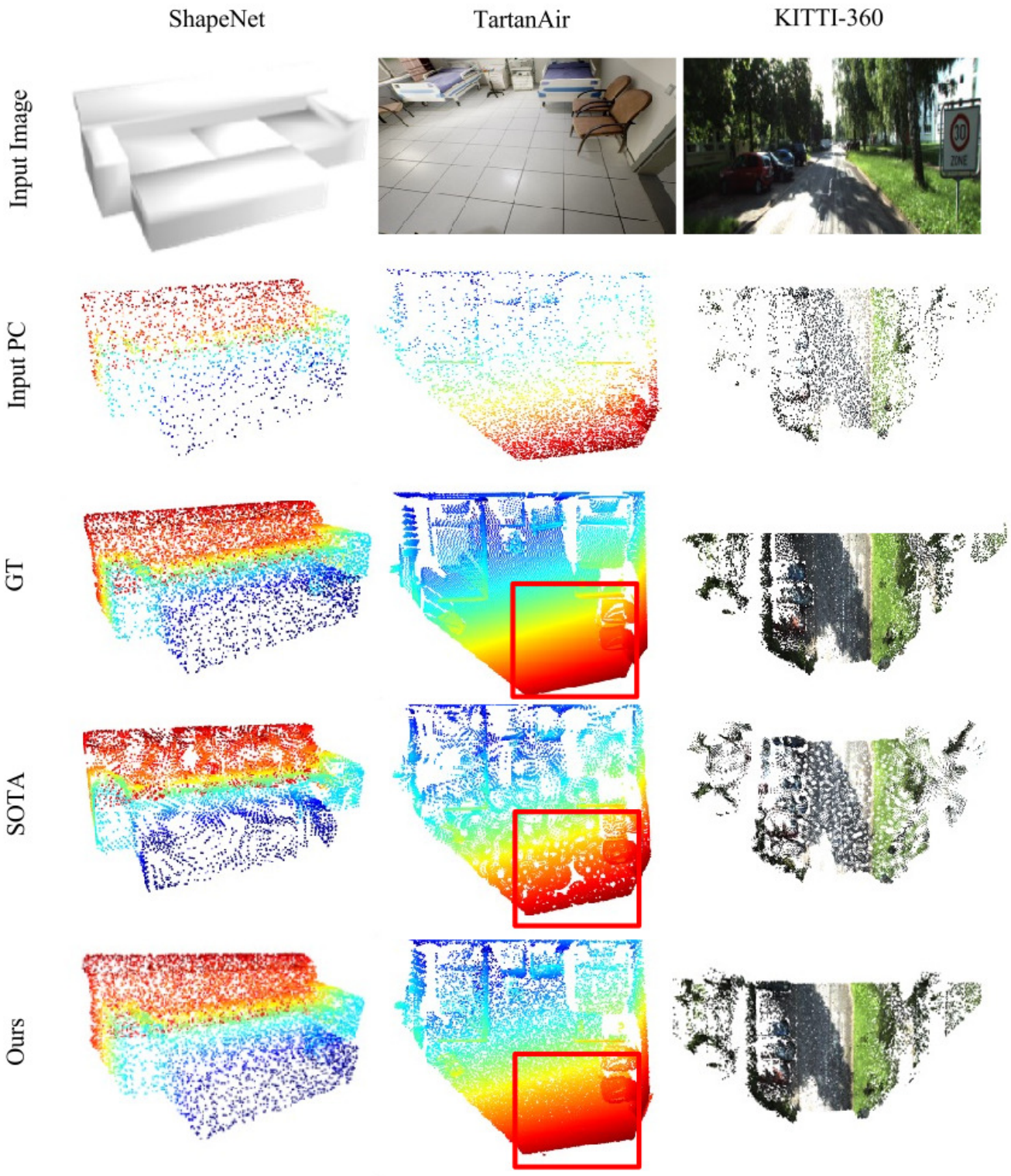

Figure 12. Zoom-in figure of the upsampling task qualitative results.

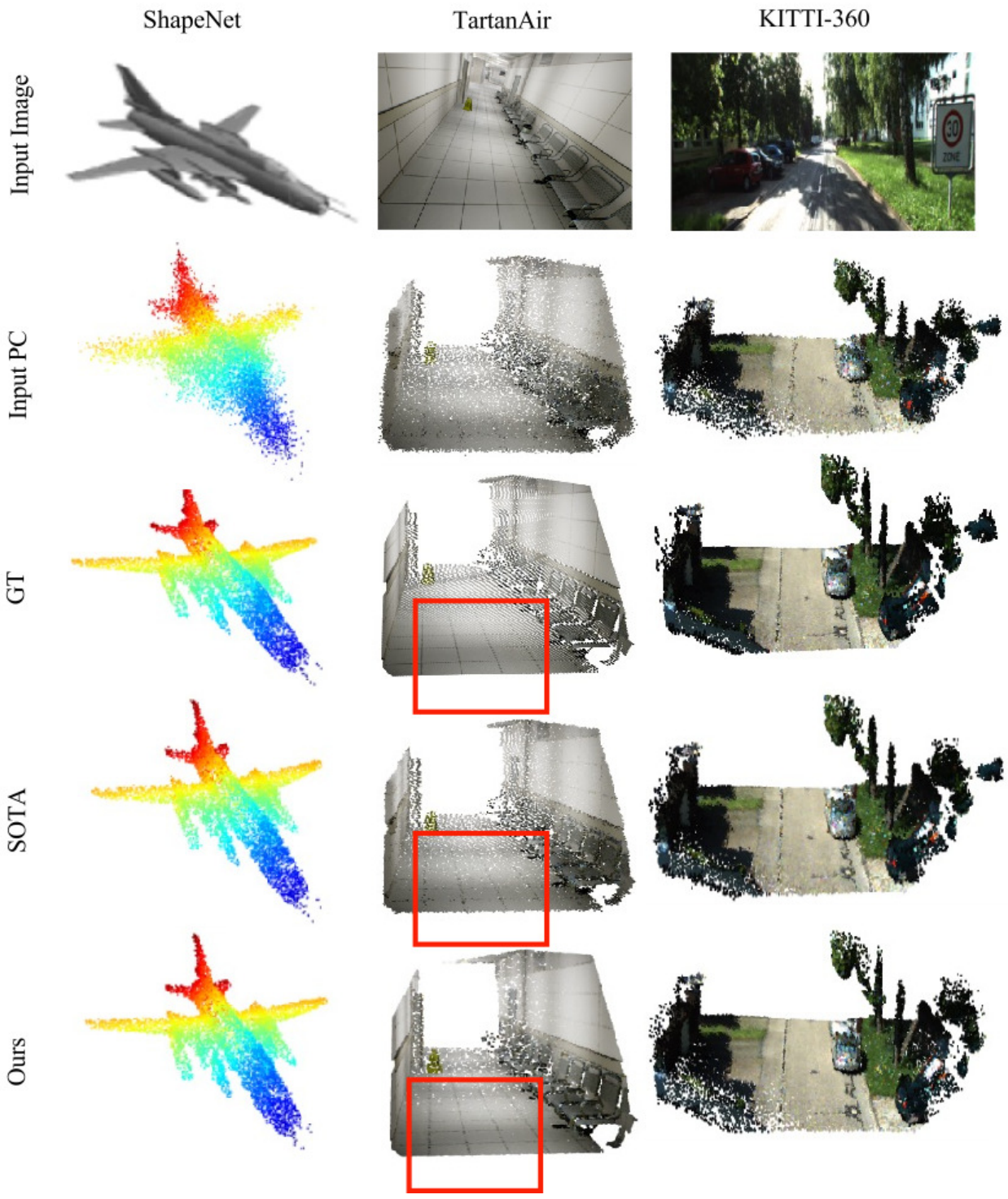

Figure 13. Zoom-in figure of the denoising task qualitative results.