Supplemental material:

PU-GAN: a Point Cloud Upsampling Adversarial Network

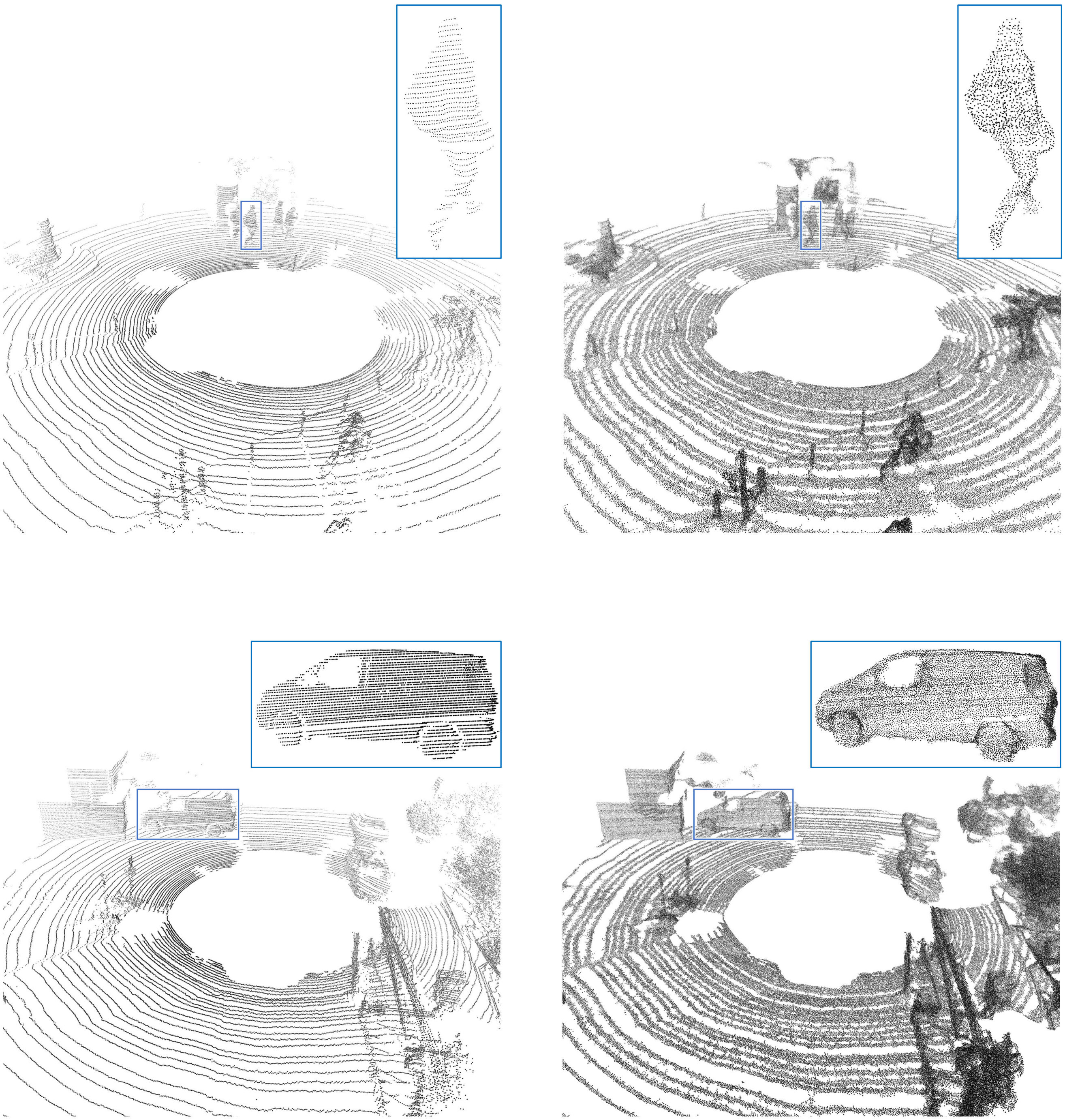
Overview

This supplemental material consists of the following sections:

- In Section A, we provide more PU-GAN results on real-scanned LiDAR point clouds.
- In Section B, we provide more visual comparisons with the state-of-the-art methods.
- In Section C, we present more PU-GAN results with varying noise levels of inputs.
- In Section D, we show more PU-GAN results with varying sizes of point sets.
- In Section E, we show a typical failure case for PU-GAN.
- In Section F, we present the detailed architecture of the baseline network.
- In Section G, we provide the derivation of the expected point-to-neighbor distance $\hat{d}$; see Section 3.5.2 in the main paper for detailed definition.
- In Section H, we show how our uniform loss enhances a non-GAN method such as PU-Net.
- In Section I, we show all of our collected training and testing models.

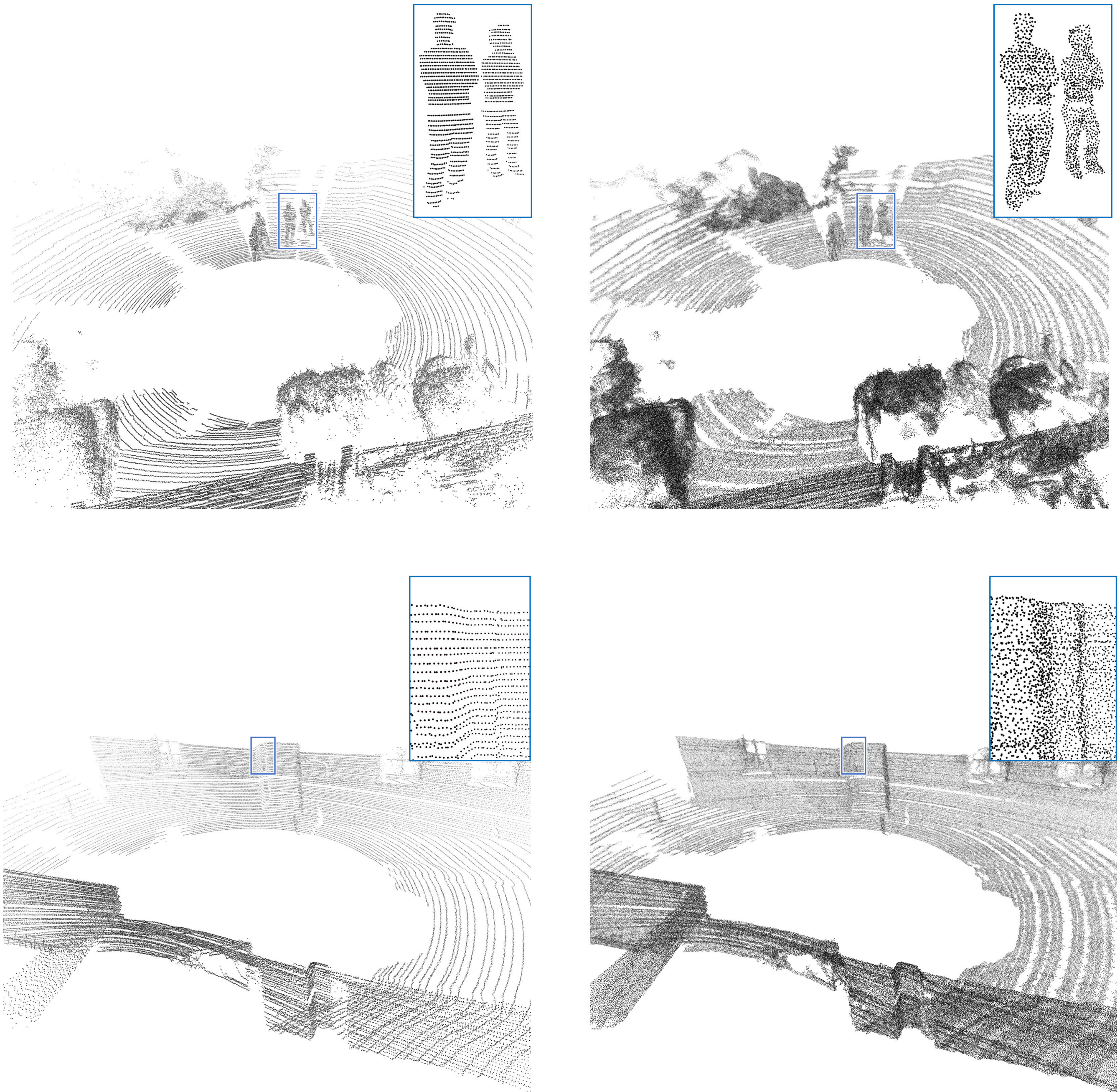
A. More Real-scanned LiDAR Results

We provide more PU-GAN results on real-scanned LiDAR point clouds; see Figures 1-4. Results show that even input LiDAR points have line-like distribution with many tiny holes, our PU-GAN can still generate promising results with uniformly-distributed points.



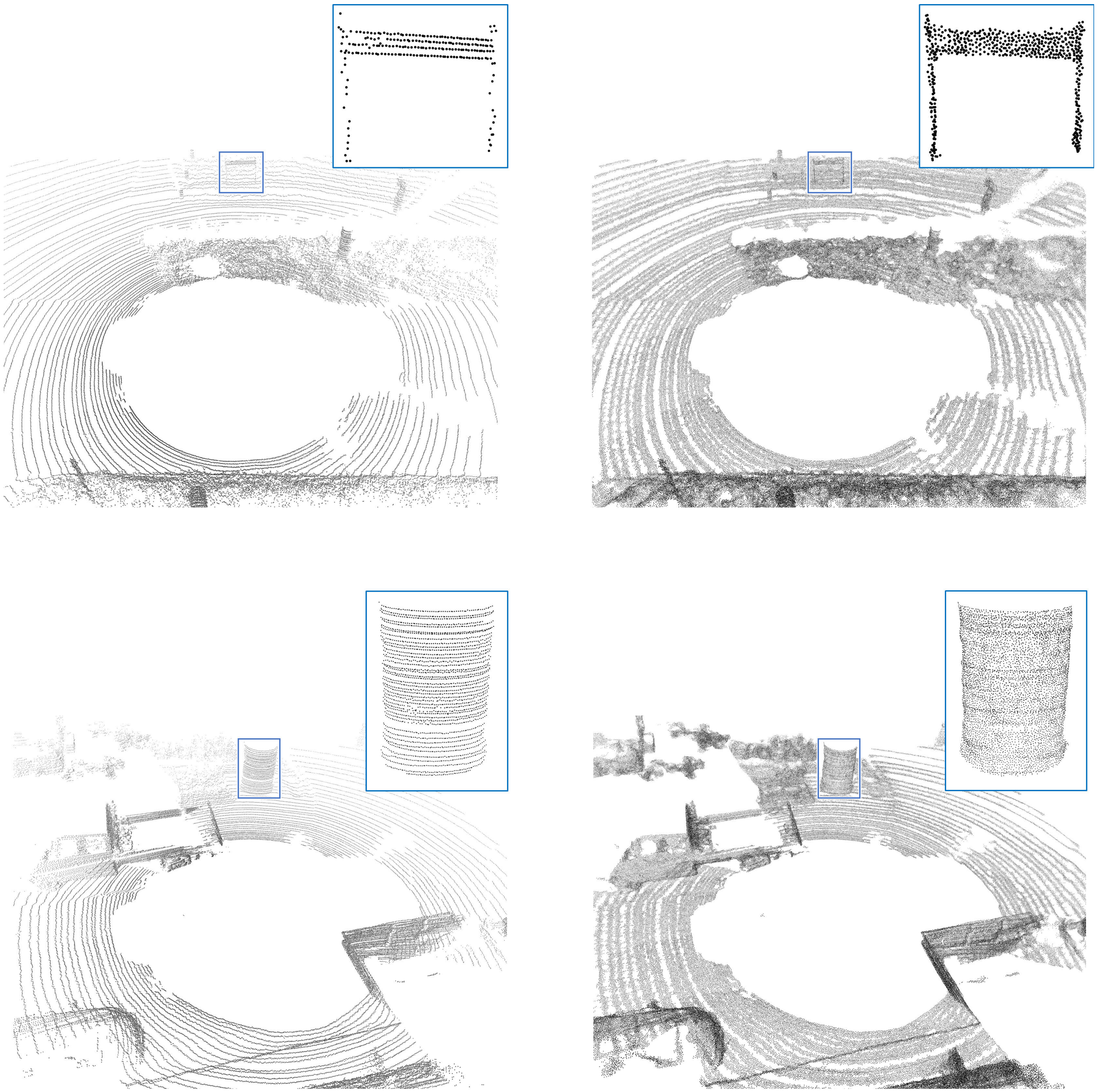
(a) LiDAR inputs (b) PU-GAN

Figure 1. More real-scanned results (1/4).



(a) LiDAR inputs (b) PU-GAN

Figure 2. More real-scanned results (2/4).



(a) LiDAR inputs (b) PU-GAN

Figure 3. More real-scanned results (3/4).

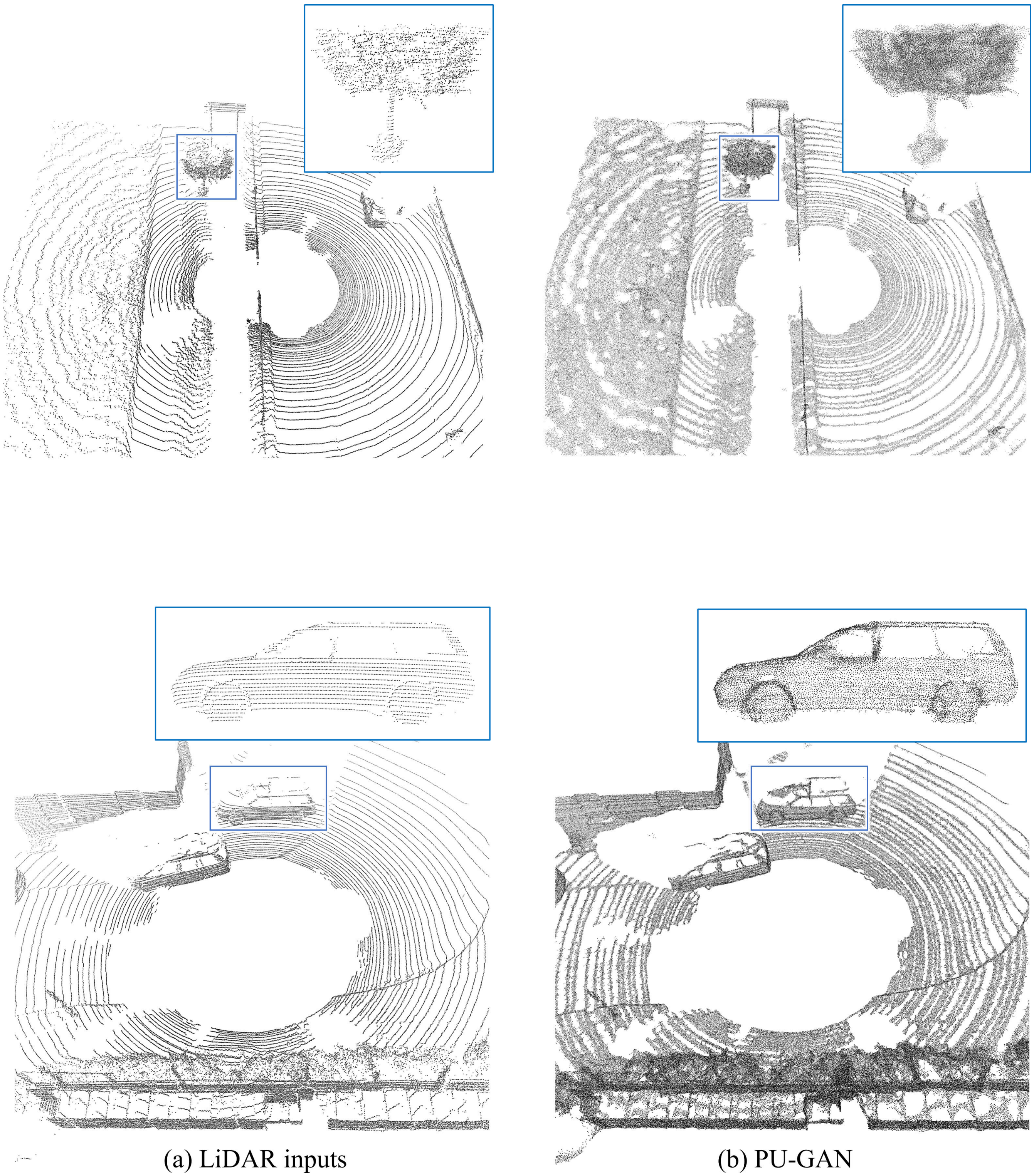


Figure 4. More real-scanned results (4/4).

B. More Visual Comparisons

Figures 5-8 demonstrate more visual comparisons by applying our PU-GAN and three state-of-the-art methods, *i.e.*, EAR [1], PU-Net [3], and MPU [2], to point cloud upsampling, as well as the corresponding surface reconstruction results. From these results we can see that, our PU-GAN can generate a more uniform point distribution and better reconstruct local structures.

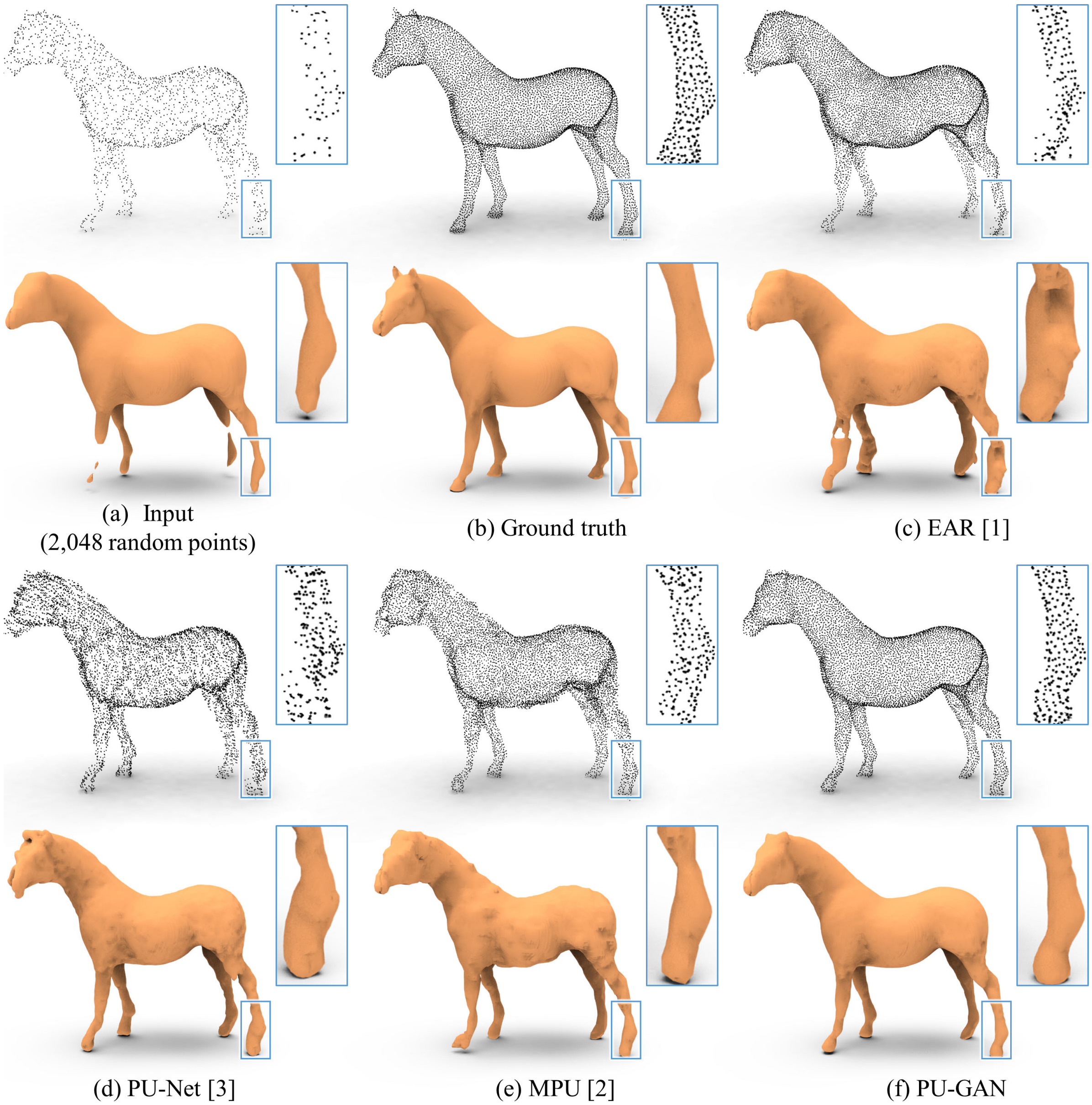
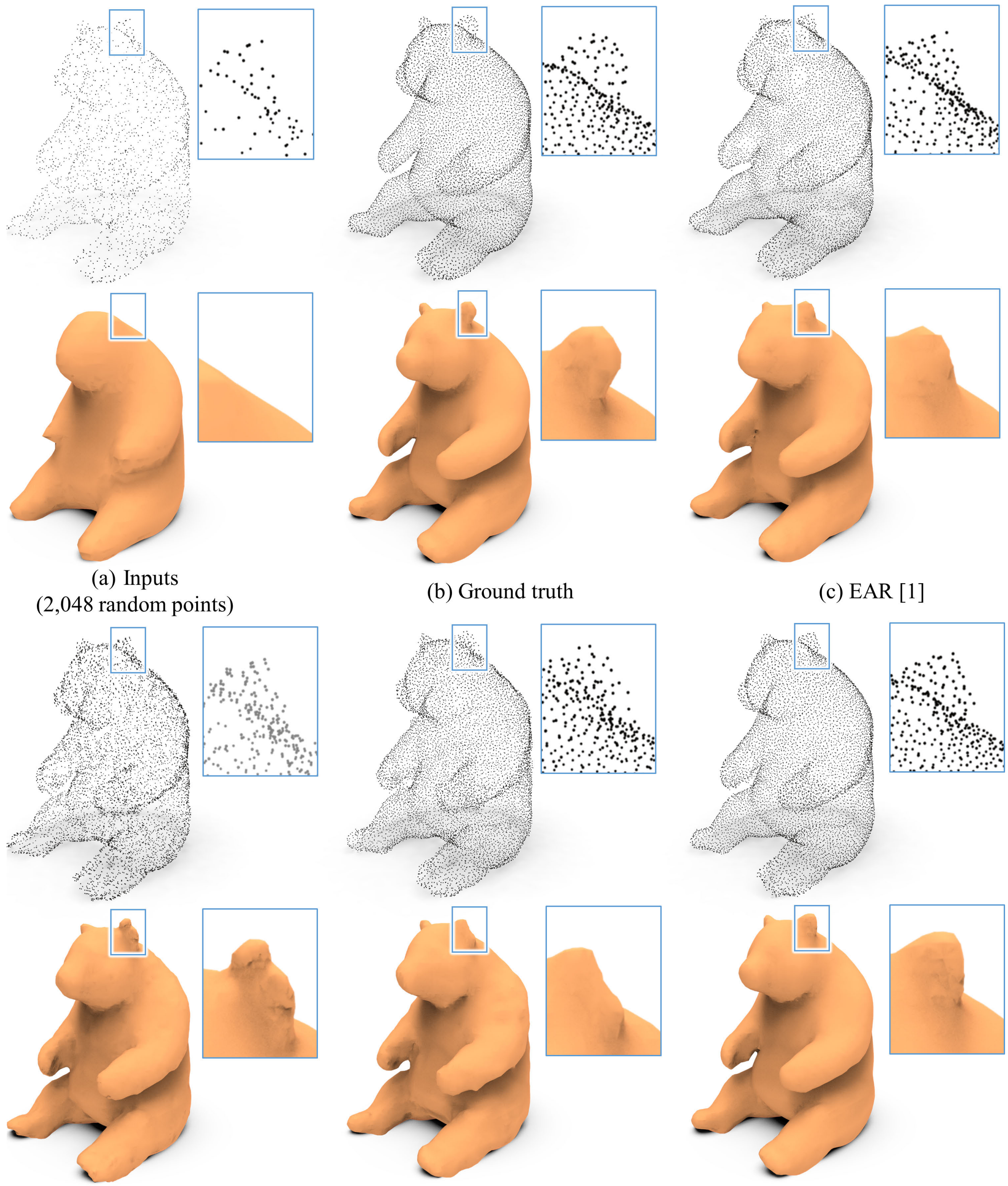


Figure 5. Comparing point set upsampling and surface reconstruction results produced by different methods (c-f) from inputs (a) (1/4).



(a) Inputs
(2,048 random points)

(b) Ground truth

(c) EAR [1]

(d) PU-Net [3]

(e) MPU [2]

(f) PU-GAN

Figure 6. Comparing point set upsampling and surface reconstruction results produced by different methods (c-f) from inputs (a) (2/4).

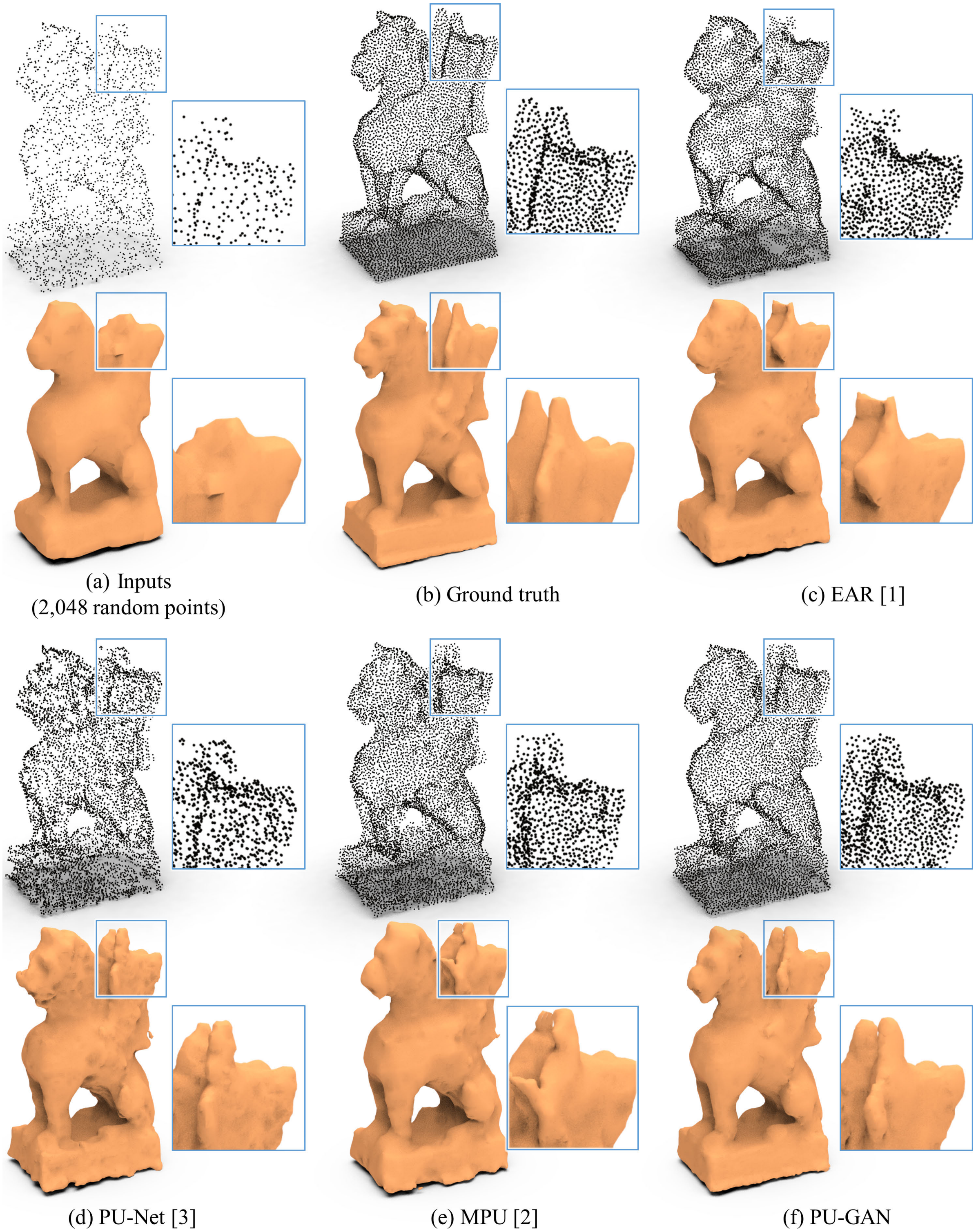


Figure 7. Comparing point set upsampling and surface reconstruction results produced by different methods (c-f) from inputs (a) (3/4).

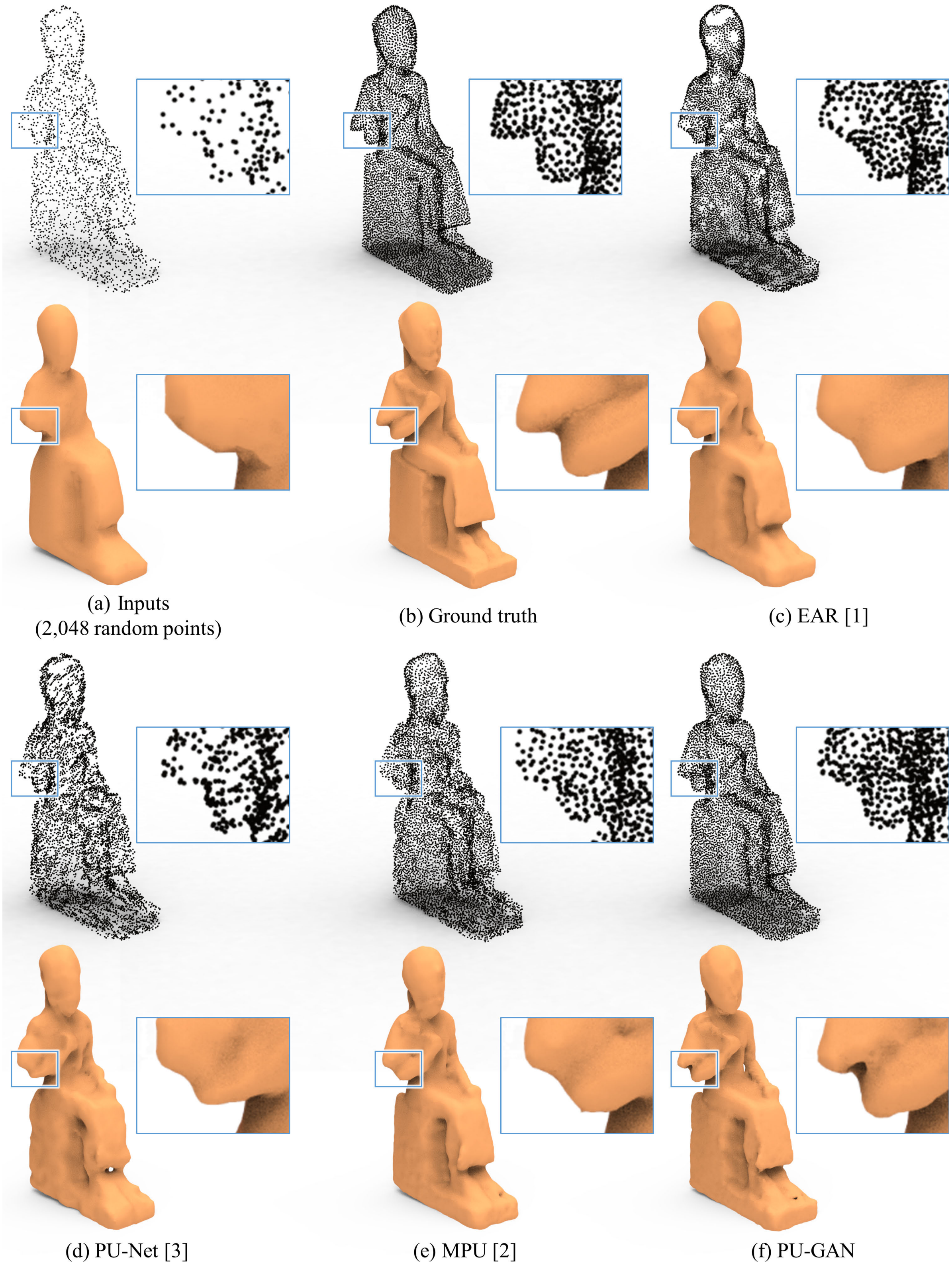


Figure 8. Comparing point set upsampling and surface reconstruction results produced by different methods (c-f) from inputs (a) (4/4).

C. More Upsampling Results of Varying Noise Levels

Figure 9 shows more upsampling results by applying MPU [2] and PU-GAN to inputs of increasing noise levels, indicating that PU-GAN can consistently generate plausible results even under large noise corruption.

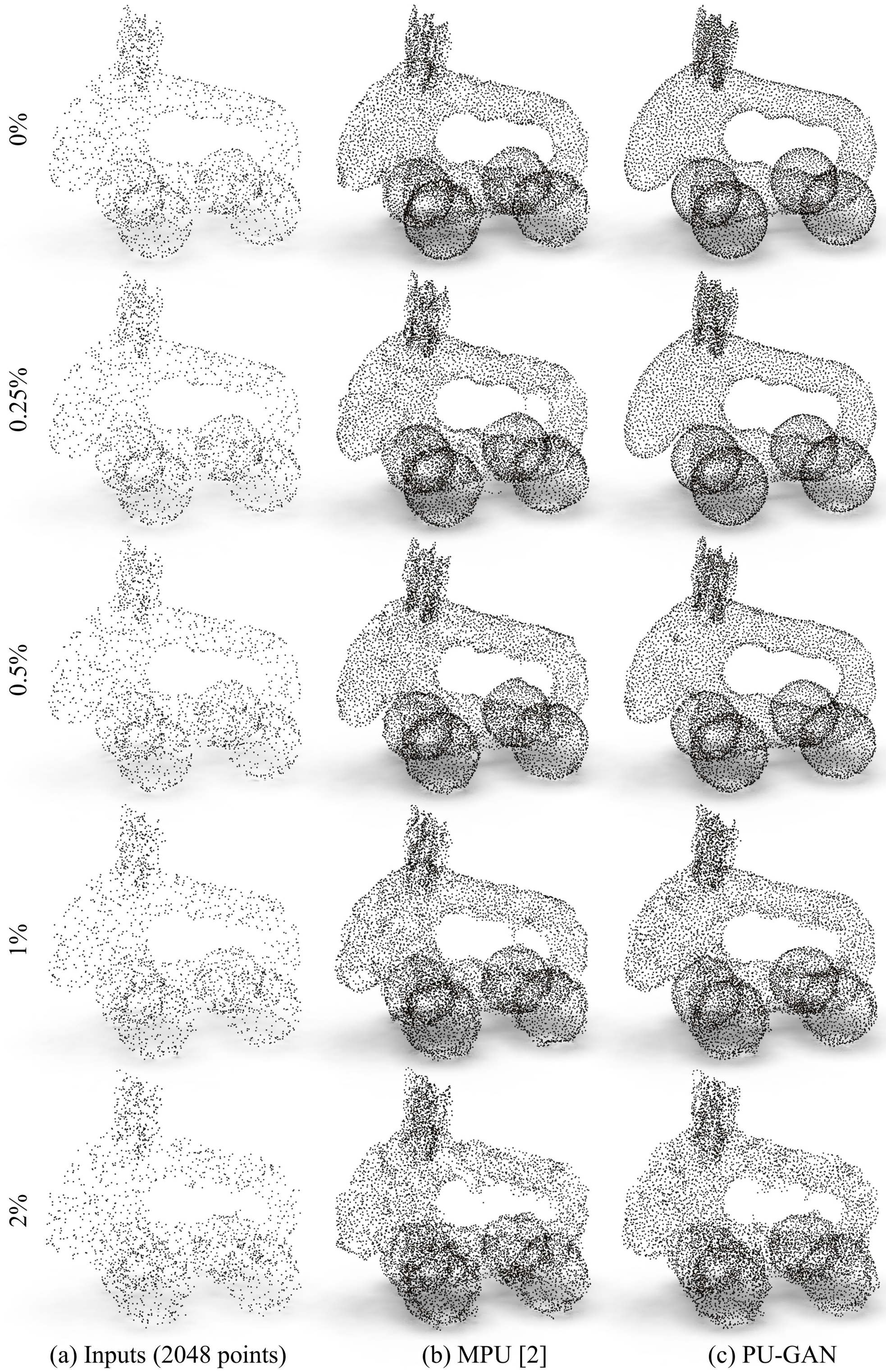


Figure 9. Upsampling results by applying MPU (b) and PU-GAN (c) to inputs with noise level of 0, 0.25%, 0.5%, 1%, and 2% (from top to bottom).

D. More Upsampling Results of Varying Sizes

Figure 10 shows two more upsampling examples by applying PU-GAN to inputs of decreasing numbers of points.

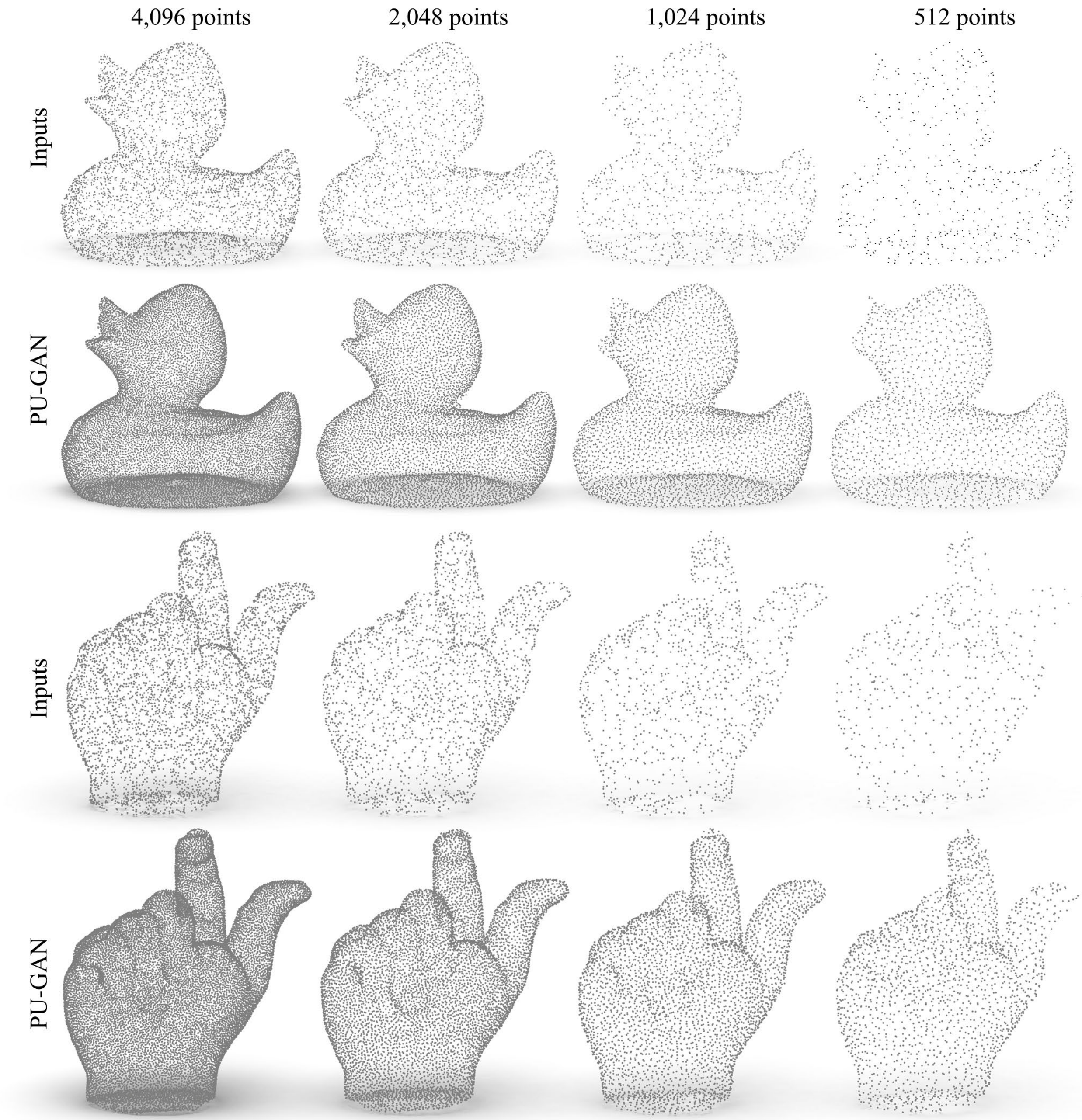


Figure 10. Upsampling results of varying input sizes.

E. Failure Case

As shown in Figure 7 of the main paper, our method is able to handle fine-grained parts such as the elephant's nose and tiger's tail. However, for very challenging cases such as the camel's feet shown in Figure 11, the input noise and sparsity make it too challenging to amend the point cloud, where the most recent state-of-the-art MPU [2] also suffers from the same problem; see the blown-up views below.

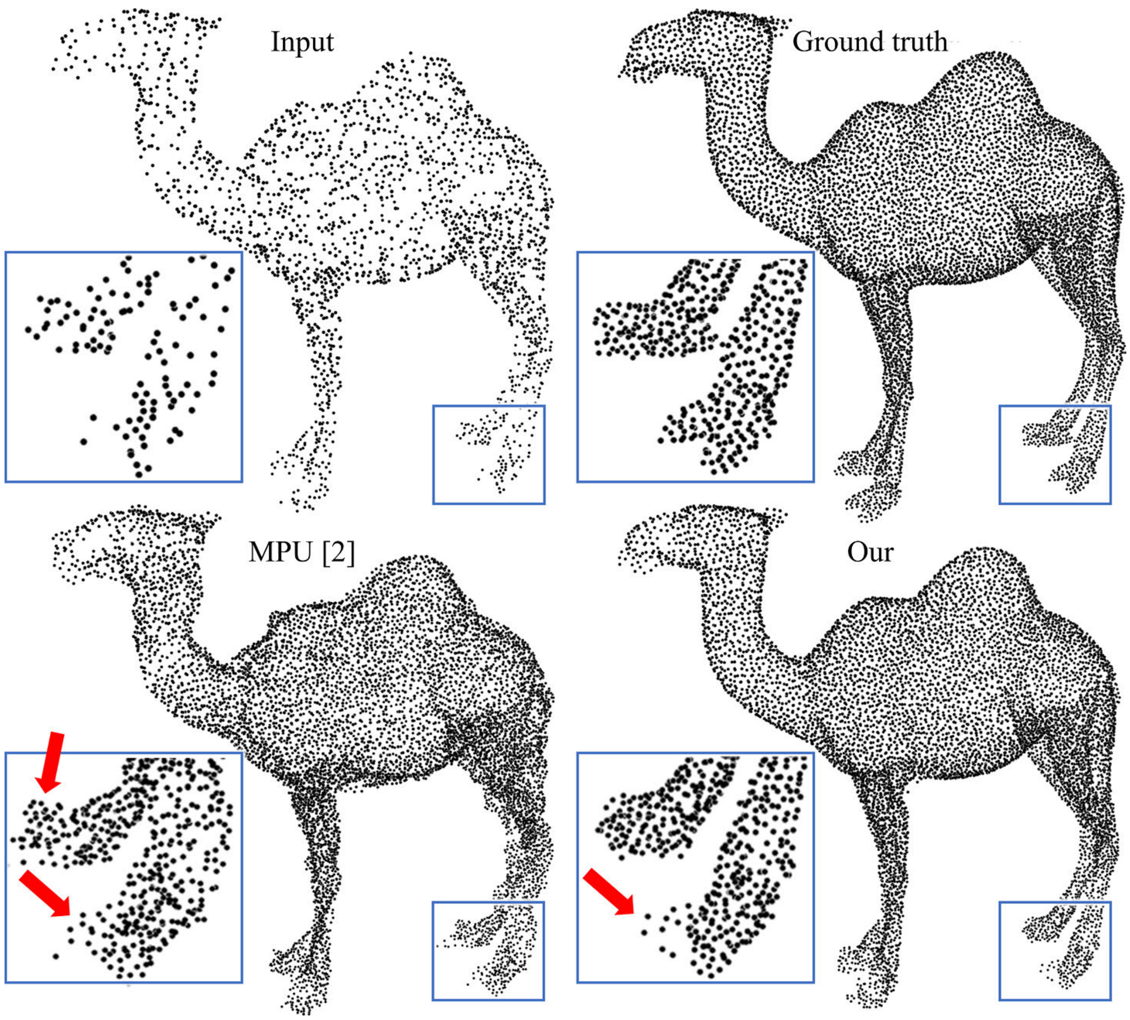


Figure 11. A typical failure case for PU-GAN.

F. Baseline Network Architecture (See Section 4.6 in main paper)

The detailed architecture of the baseline network is illustrated in Figure 12 below. Here, we remove the uniform loss $\mathcal{L}_{\text{uni}}$ (see Eq.(6) in our main paper), the self-attention unit, the farthest sampling altogether and replace the up-down-up expansion unit with a single up-feature operator (see Figure 3 in our main paper) without self-attention unit.

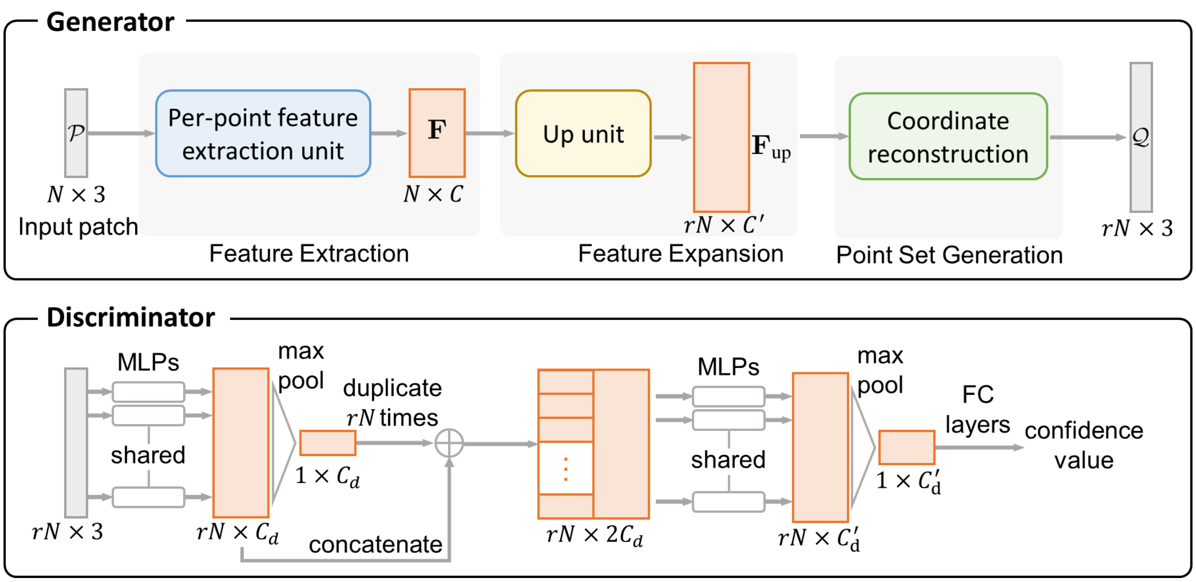
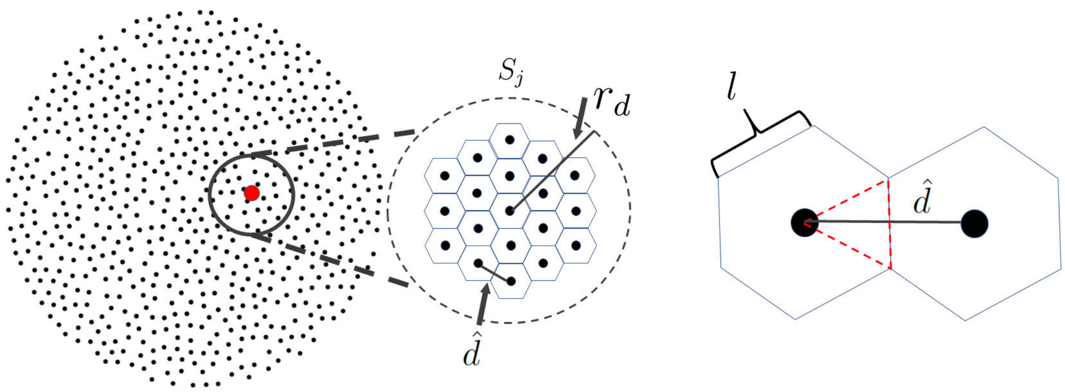


Figure 12. The architecture of the baseline network for point cloud upsampling.

G. Derivation of the Expected Point-to-neighbor Distance $\hat{d}$



(a) Q (b)

Figure 13. Illustration of the derivation of $\hat{d}$.

As we mentioned in Section 3.5.2 of the main paper, during the training, we use the ball query of radius r_d to crop a point subset (denote as S_j) in patch Q at each seed; see Figure 13 (a). Since r_d is typically very small and S_j roughly lies on a small local disk, the area of S_j could thus be approximated by πr_d^2. If S_j has a uniform distribution, each point inside S_j should roughly occupy a hexagonal region, as shown in Figure 13 (middle), where the area of each hexagon is $\frac{\pi r_d^2}{|S_j|}$.

On the other hand, each hexagon could be divided into six equilateral triangles with the side length l; see the red triangle in Figure 13 (b). The area of each hexagon can be calculated as an area sum of six triangles, so we can obtain the following equality:

$$\frac{\pi r_d^2}{|S_j|} = 6 \times \frac{1}{2} \times l \times \frac{\sqrt{3}}{2} l.$$

Hence, the side length l of each triangle can be solved as,

$$l = \sqrt{\frac{2\pi r_d^2}{3\sqrt{3}|S_j|}}. \tag{1}$$

Since $\hat{d} = \sqrt{3}l$, we can also solve $\hat{d}$ as,

$$\hat{d} = \sqrt{\frac{2\pi r_d^2}{|S_j|\sqrt{3}}}.\blacksquare \tag{2}$$

H. How our uniform loss enhances PU-Net

To explore the applicability of our uniform loss, we adopted it in the PU-Net architecture, which is a non-GAN patch-based method for point upsampling, and performed the same experiment as presented in Table 1 of the main paper on this PU-Net architecture with our uniform loss (denoted as PU-Net + L_{uni}). Table 1 below shows quantitative results in terms of the point to surface (P2F), CD, and HD metrics. From the results, we can see that with the uniform loss, the performance of PU-Net can be improved for all the three metrics.

Table 1. Quantitative comparison for PU-Net with or without uniform loss.

Methods	P2F(10^{-3})	CD(10^{-3})	HD(10^{-3})
PU-Net+L_{uni}	**6.17**	**0.59**	**7.86**
PU-Net	6.84	0.72	8.94

I. Illustration of our Collected Dataset

We collected a rich variety of 3D models, where 120 models are for training and 27 models are for testing; see Figures 14-18 for details. As we can see, our collected models have a large variation in geometry shapes, ranging from 3D models with smooth surface regions to high-detailed structures.

Figure 14. Collected 3D models in our training dataset (1/4).

Figure 15. Collected 3D models in our training dataset (2/4).

Figure 16. Collected 3D models in our training dataset (3/4).

Figure 17. Collected 3D models in our training dataset (4/4).

Figure 18. Collected 3D models in our testing dataset.

References

[1] Hui Huang, Shihao Wu, Minglun Gong, Daniel Cohen-Or, Uri Ascher, and Hao Zhang. Edge-aware point set resampling. *ACM Trans. Gr.*, 32(1):9:1–12, 2013. 6

[2] Wang Yifan, Shihao Wu, Hui Huang, Daniel Cohen-Or, and Olga Sorkine-Hornung. Patch-based progressive 3D point set upsampling. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 5958–5967, 2019. 6, 10, 12

[3] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. PU-Net: Point cloud upsampling network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2790–2799, 2018. 6

The End