MVP-Net: Multi-View Depth Image Guided Cross-Modal Distillation Network for Point Cloud Upsampling

Supplementary Material

This document provides more implementation details and additional experimental results for comparative studies.

A Implementation Details

This section provides additional information about hyperparameter settings and detailed network architecture.

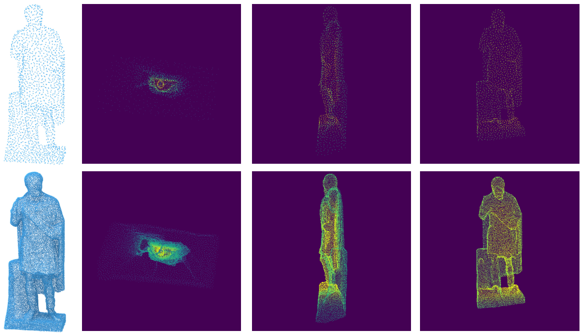


Figure 1: An example of the multi-view depth image rendering. The top row is a sparse point cloud and its multi-view depth images. The bottom row is the ground truth and its multi-view depth images. The depth images are colored only for visualization.

A.1 Multi-View Depth Image Rendering

We apply the basic renderer proposed in SimpleView [3] to get the depth images with the resolution of 128×128 from three orthogonal viewpoints, as shown in Figure 1. Each point cloud is normalized in a unit sphere. Therefore, the viewpoints for different point clouds are fixed, regardless of the scale of the point cloud. The renderer applies perspective projection to get the 2D coordinate ($\tilde{x} = x/z, \tilde{y} = y/z$) of a point p at depth z, and then uses ($\lceil\tilde{x}\rceil, \lceil\tilde{y}\rceil$) to be the final coordinate of p on the image plane since coordinates on image plane have to be discrete. Multiple points may be projected to the same discrete location on the image plane. We follow the approach described in SimpleView [3] to perform a weighted average of depth values, assigning higher weight ($\frac{1}{z}$) giver to closer points, which could reduce in noise due to the averaging of nearby pixels on the surface.

A.2 Cross-Modal Feature Extraction

Our cross-modal feature extraction module contains dual branches to extract point features and depth image features, respectively. As described in the main text, we utilize the ResNet [4] to extract multi-hierarchical feature map $\{F_D^i\}_{i=1}^L$ by hierarchically downsampling. Figure 2 shows the architecture of the ResNet we utilized. We employ dynamic graph construction and Edge Convolution introduced in DGCNN [17], and the intra-level and inter-level dense connections [6, 18] to organize point features at different hierarchies for extracting point features.

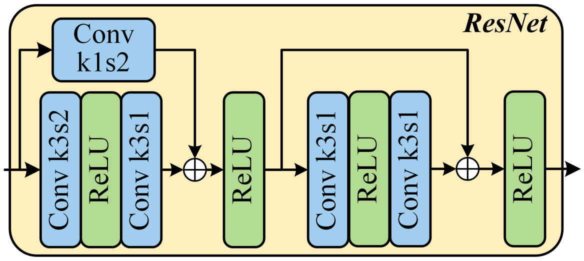


Figure 2: The structure of the ResNet utilized in our cross-modal feature extraction module. Conv kKsS denotes 2D convolution with a kernel size of $K \times K$ and a stride of S.

Our cross-modal feature extraction module contains 4 layers, i.e., $L = 4$. For the depth image branch, we first utilize a 2D Convolution with a kernel size of 1×1 to embed depth images into 16 channels and set the number of channels in the depth image branches to 16, 32, 64, and 128 for each layer, respectively. For the point cloud branch, we first utilize a 1D Convolution with a kernel size of 1 to embed point clouds into 24 channels and set the number of output channels of DenseNet at different layers in cross-modal feature extraction to 120, 168, 168 and 168, respectively. This is because the output channel of each edge convolution is set to 24, and each DenseNet contains 3 edge convolution layers. The input channels of each DenseNet are set to 48, except for the first DenseNet, which is set to 24. The input of each DenseNet is the concatenation of all the outputs from previous hierarchies, followed by a 1D convolution with a kernel size of 1 to reduce the number of channels. Finally, a 1D convolution with a kernel size of 1 is employed to fuse all the point features from these hierarchies.

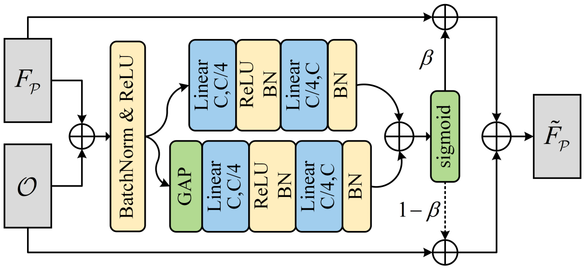


Figure 3: The architecture of the modified attentional feature fusion. GAP denotes global average pooling. BN denotes BatchNorm Layer [7].

Table 1: Quantitative comparisons of state-of-the-art methods on PUGeo-Net dataset. We highlight the best and the second-best results in bold and underlined, respectively.

Ratio	2× Upsampling					8× Upsampling				
Method	DCD	CD	HD	P2FM	P2FS	DCD	CD	HD	P2FM	P2FS
PU-Net [20]	0.324	0.241	3.328	1.660	2.055	0.461	0.303	6.577	2.949	2.863
MPU [18]	0.359	0.275	2.442	1.165	1.652	0.360	0.199	5.194	1.492	2.126
PU-GAN [8]	0.397	0.403	13.20	2.032	3.200	0.422	0.299	7.600	1.923	1.884
Dis-PU [9]	0.325	0.257	2.653	1.305	1.785	0.327	0.175	3.896	1.288	2.020
PU-GCN [12]	0.384	0.325	3.489	1.580	2.011	0.378	0.180	5.192	1.711	2.167
PUGeo-Net [13]	0.373	0.295	3.914	1.305	2.122	0.354	0.177	3.955	1.334	1.892
MAFU [14]	0.406	0.362	3.651	1.086	1.637	0.332	0.151	4.026	1.258	1.844
PUFA-GAN [10]	0.311	0.272	9.108	1.736	2.920	0.325	0.177	11.281	1.556	3.094
PU-Flow [11]	0.258	0.174	1.400	0.860	1.271	0.307	0.104	1.635	1.163	1.387
Grad-PU [5]	0.240	0.165	**1.168**	0.698	1.207	0.263	0.073	**1.036**	0.803	**1.108**
SSPU-Net [16]	0.248	0.168	1.473	**0.620**	**0.957**	0.250	0.075	1.969	0.754	1.238
Ours (Teacher)	0.191	0.127	1.219	0.690	0.958	0.214	0.055	1.559	0.525	0.692
Ours (Student)	**0.193**	**0.133**	1.332	0.687	0.983	**0.232**	**0.067**	1.898	**0.667**	1.179

A.3 Corss-Modal Feature Fusion

Inspired by [1], we introduce the modified attentional feature fusion (AFF), i.e., sub-network $\mathcal{A}(\cdot)$, as mentioned in the main text. As shown in Figure 3, the AFF block begins with a BatchNorm Layer [7] to mitigate differences in the data distribution between cross-modal features. Then, the two branches in the AFF block are utilized to capture the local context (via the top branch) and the global context (via the bottom branch), respectively. Finally, the Sigmoid activation function is employed to calculate the parameter β.

A.4 Upsampling Tail

Our upsampling tail contains two modules: dense point generator and spatial refinement. We directly employ the I^2-Feature Aggregation [16], the Folding strategy [19] and the Structure-Sensitive Transformer (SSTr) [16] as our dense point generator. Additionally, we utilize the Multi-Scale Spatial Refinement introduced in SSPU-Net [16]. For more details about the upsampling tail, please refer to SSPU-Net [16].

A.5 Detail Estimation and Distillation

In this section, we discuss how to accurately align the depth image of the sparse point cloud with the depth image of ground truth, during the training process of the teacher network. We hypothesize two conditions that are extremely easy to achieve: (i) The sparse point cloud is precisely aligned with the spatial pose of the ground truth; (ii) The viewpoints of the basic renderer are fixed when generating the depth images of a sparse point cloud and the depth images of its ground truth. We randomly downsample the dense point clouds (ground truth) for the first condition to generate their corresponding sparse point clouds. Therefore, the spatial pose of the ground truth and its sparse counterpart are well-aligned. We use three fixed viewpoints on the 3D coordinate axis for the second condition.

B Experimental Details

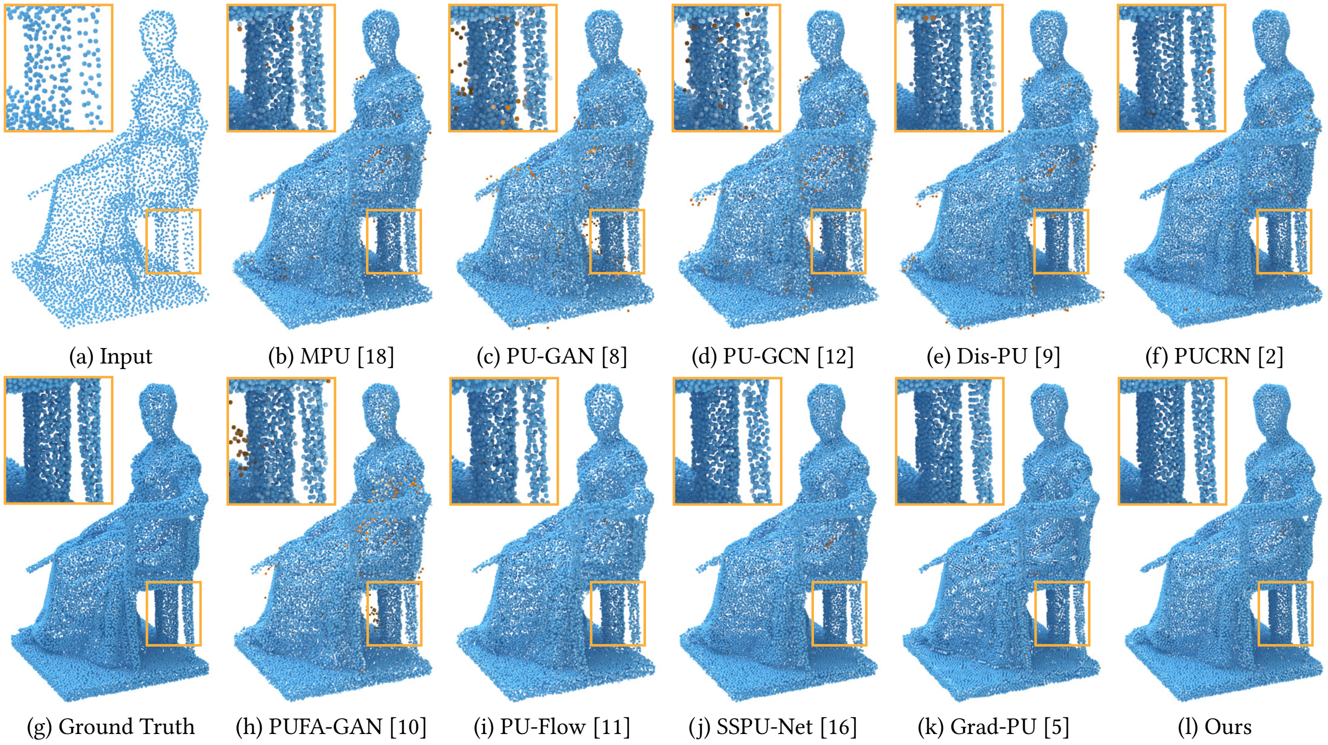
B.1 Inference Phase of Teacher Network

In this section, we describe our approach to conducting inference with the teacher network on the test dataset. Since we train the network on patch point clouds, we first utilize the farthest point sampling algorithm to select a series of seed points in each test point cloud, then employ the KNN algorithm to crop the test point cloud into sparse patches. Then, we employ the KNN algorithm on the ground truth of each test point cloud to crop dense patches as the approximated ground truth of sparse patches with the same seed points. Please note that we perform quantitative comparisons for the teacher network only to assess the performance gap between it and its student network, and all the visual comparisons of our method are the results of the student network that does not take ground truths as inputs.

C Additional Experimental Results

C.1 Quantitative Results on PUGeo-Net Dataset

In this section, we provide more quantitative results on the PUGeo-Net 2× and 8× dataset [13], as shown in Table 1. Our method achieves the best performance according to the DCD and CD metrics. For the HD and P2F metrics, our method demonstrates competitive performance alongside recent state-of-the-art methods, Grad-PU [5] and SSPU-Net [16]. It is evident that our method achieves the best overall upsampling results in terms of the average point-to-point error (CD) and the density similarity (DCD) with ground truth, resulting in a more uniform distribution. The performance


(a) Input (b) MPU [18] (c) PU-GAN [8] (d) PU-GCN [12] (e) Dis-PU [9] (f) PUCRN [2]

(g) Ground Truth (h) PUFA-GAN [10] (i) PU-Flow [11] (j) SSPU-Net [16] (k) Grad-PU [5] (l) Ours

Figure 4: Visual comparison for 4× upsampling on PUGeo-Net dataset.

on the P2F metric demonstrates that the upsampling results generated by our method are faithfully distributed on the surface of the underlying object.

Table 2: Ablation studies of the setting of viewpoints conducted on the PU-GAN 4× dataset without the guidance of the teacher network.

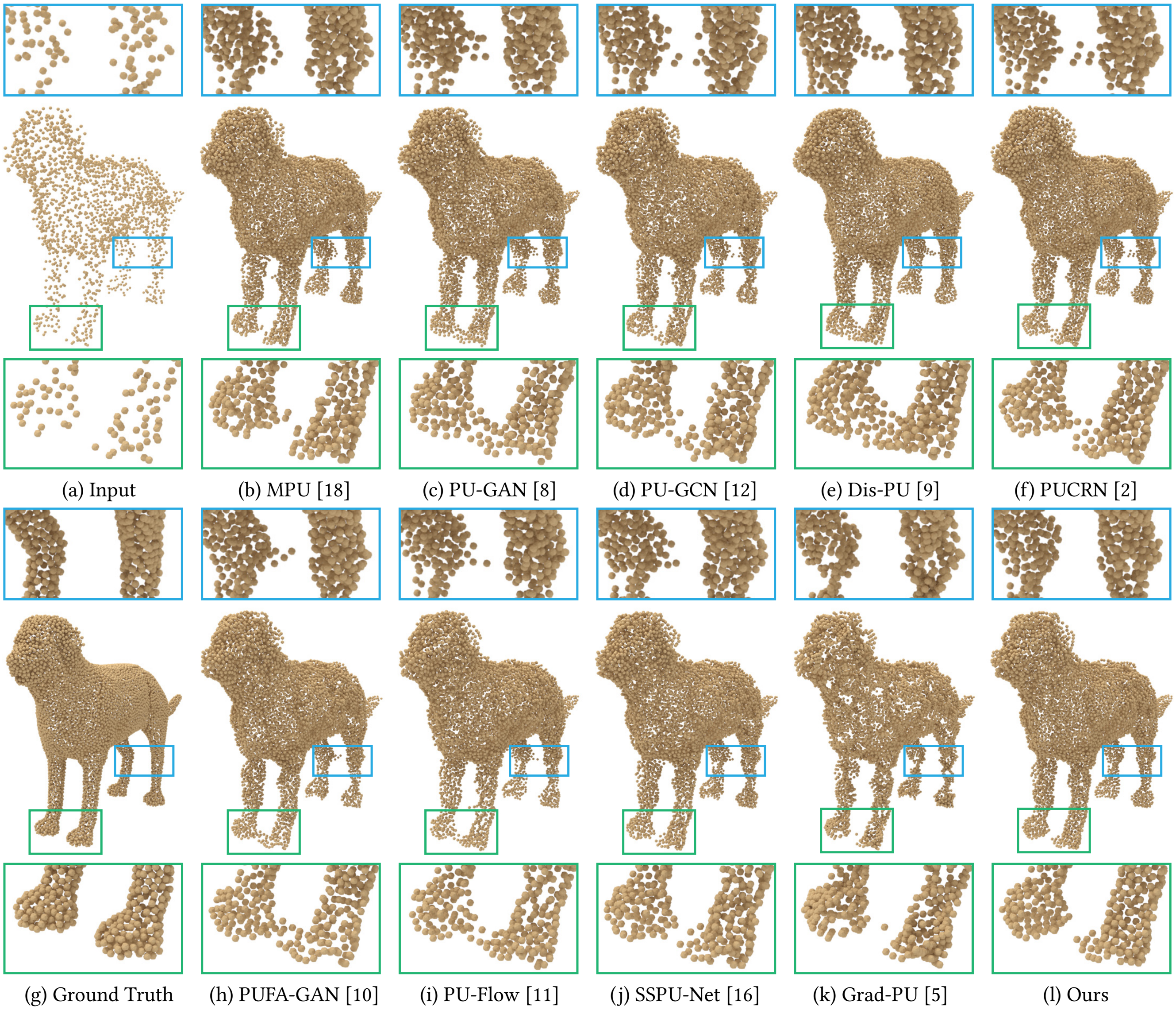
Model	Variants	CD (10^{-3})	HD (10^{-3})
C1	1 viewpoints	0.182	2.101
C2	2 viewpoints	0.179	2.084
C3	3 viewpoints	0.177	2.095
C4	6 viewpoints	0.176	2.086

C.2 Ablation Studies

In this section, we provide more ablation studies of the viewpoints of the multi-view depth image, as shown in Table 2. As mentioned in the main text, we utilize three orthogonal viewpoints to generate multi-view depth images for each point cloud. The results of the three orthogonal viewpoints are described in model C3 of Table 2. The results of only one viewpoint and two viewpoints in shown in models C1 and C2, respectively. As we can see, reducing the number of viewpoints leads to performance degradation. Then, we attempt to increase the number of viewpoints to six adopting the same setting of SimpleView [3]. Its results in depicted in model C4 of Table 2. It is evident that the number of views is doubled compared to model C3, but performance is only slightly improved. Therefore, to balance the trade-off between effectiveness and computational consumption, we chose the three orthogonal viewpoints as the default setting to generate multi-view depth images.

C.3 Qualitative Results

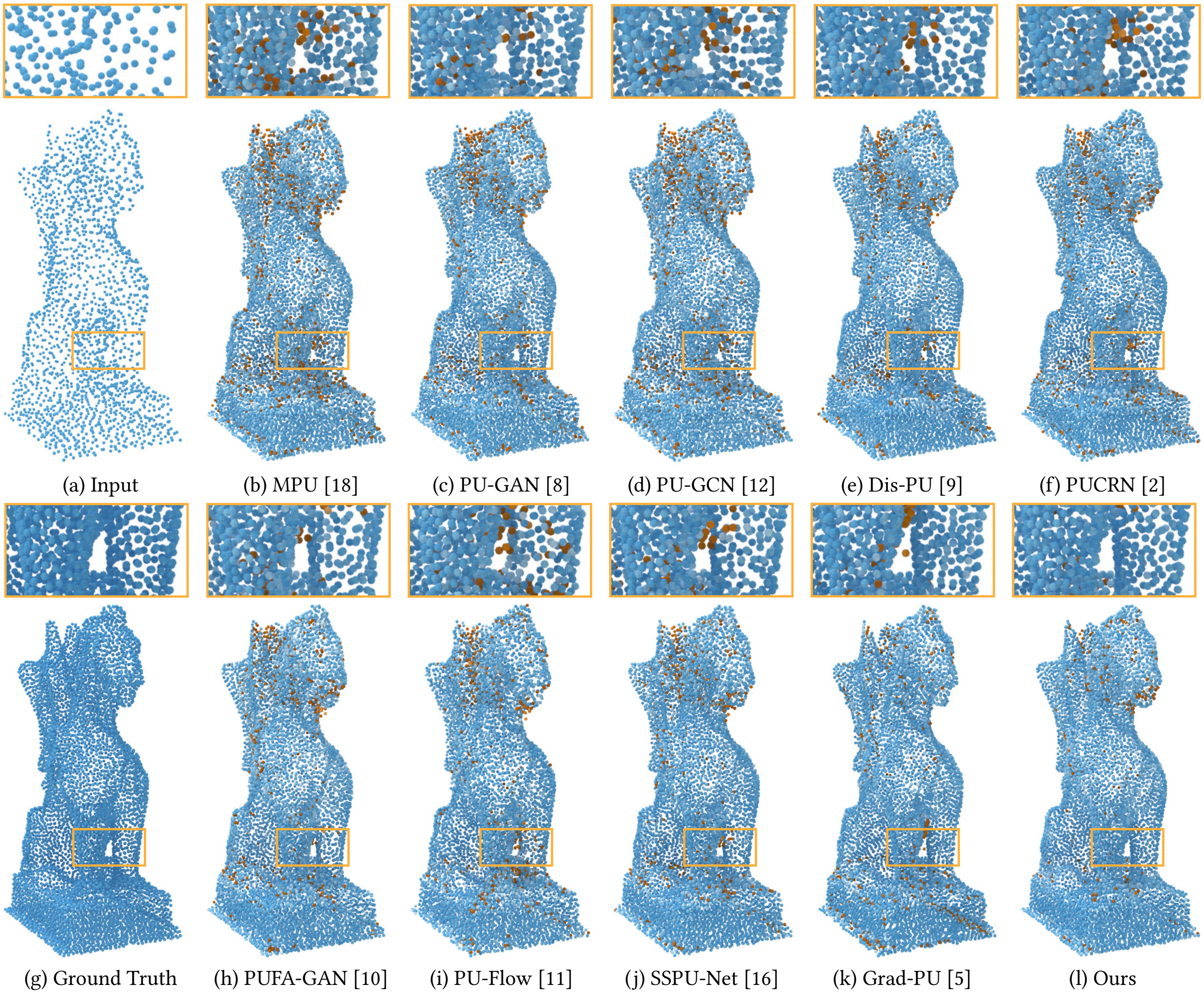
In this section, we show some more qualitative results on the PU-GAN [8], PUGeo-Net [13] and ScanObjectNN [15] dataset. Specifically, in Figure 5, a visual comparison of noisy input with Gaussian noise at 1.0% level is presented. Figure 6 and Figure 4 provide further visual comparisons on the PU-GAN and PUGeo-Net datasets, under 4× settings. Figure 7 shows additional visual comparisons on the PU-GAN dataset, under 8× settings. Figure 8 presents further visual comparisons on the PUGeo-Net dataset, under 8× settings. Finally, Figure 9 demonstrates a visual comparison of the ScanObjectNN dataset with non-uniform inputs, under 16× settings.



(a) Input (b) MPU [18] (c) PU-GAN [8] (d) PU-GCN [12] (e) Dis-PU [9] (f) PUCRN [2]

(g) Ground Truth (h) PUFA-GAN [10] (i) PU-Flow [11] (j) SSPU-Net [16] (k) Grad-PU [5] (l) Ours

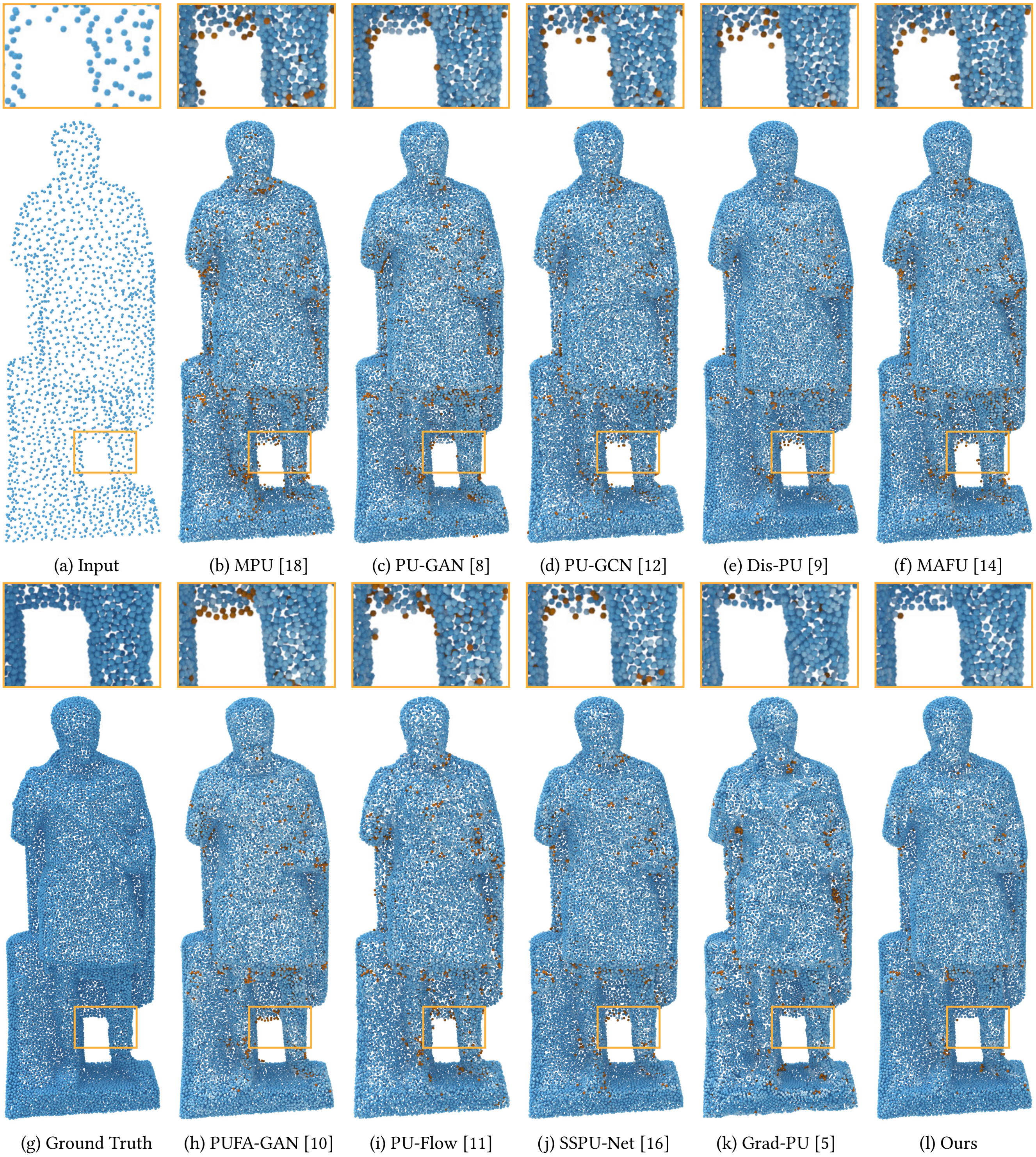
Figure 5: 4× **qualitative results on the PU-GAN dataset with Gaussian noise level** 1.0%.



(a) Input (b) MPU [18] (c) PU-GAN [8] (d) PU-GCN [12] (e) Dis-PU [9] (f) PUCRN [2]

(g) Ground Truth (h) PUFA-GAN [10] (i) PU-Flow [11] (j) SSPU-Net [16] (k) Grad-PU [5] (l) Ours

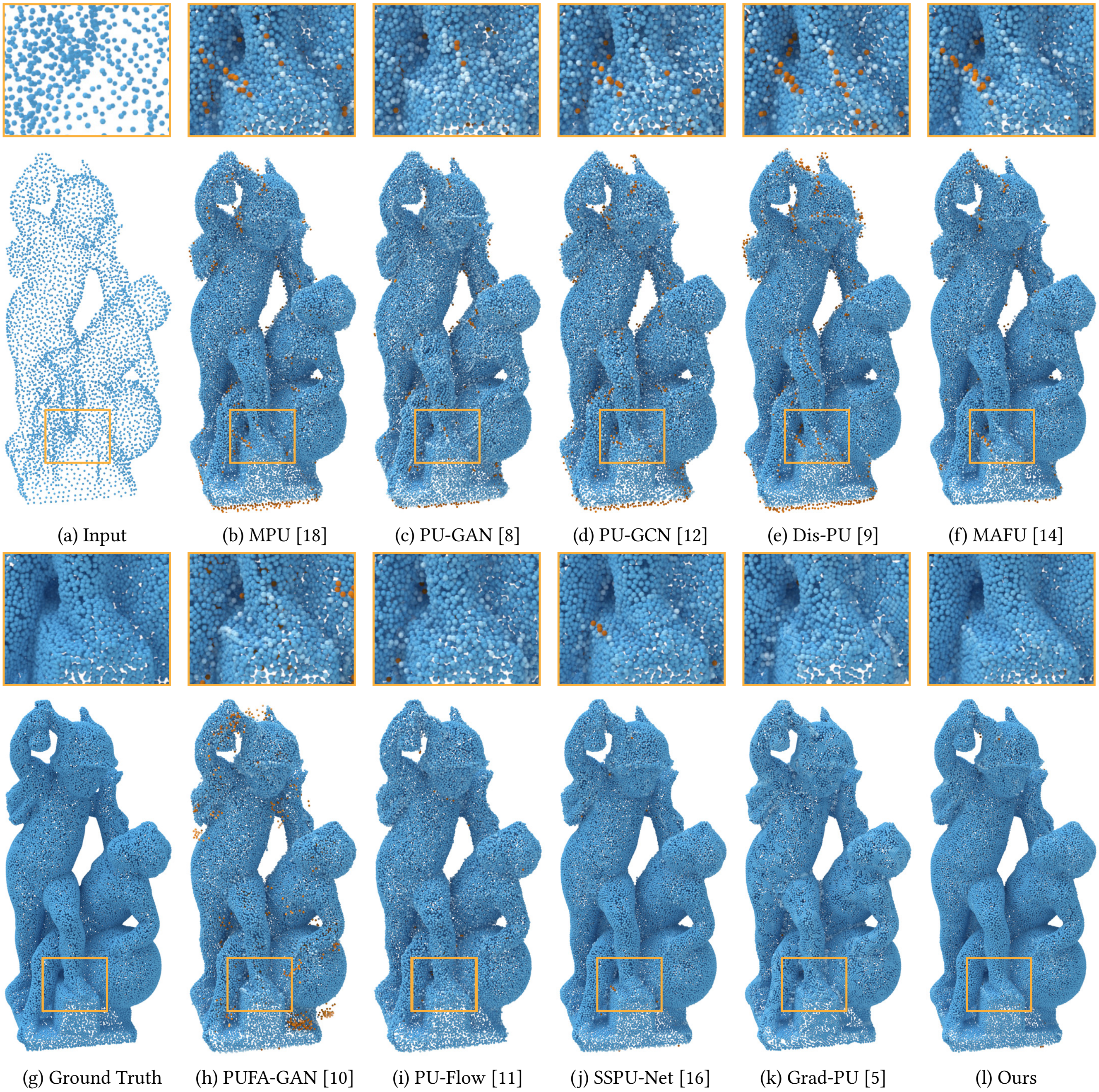
Figure 6: Visual comparison for 4× upsampling on PU-GAN dataset.



(a) Input (b) MPU [18] (c) PU-GAN [8] (d) PU-GCN [12] (e) Dis-PU [9] (f) MAFU [14]

(g) Ground Truth (h) PUFA-GAN [10] (i) PU-Flow [11] (j) SSPU-Net [16] (k) Grad-PU [5] (l) Ours

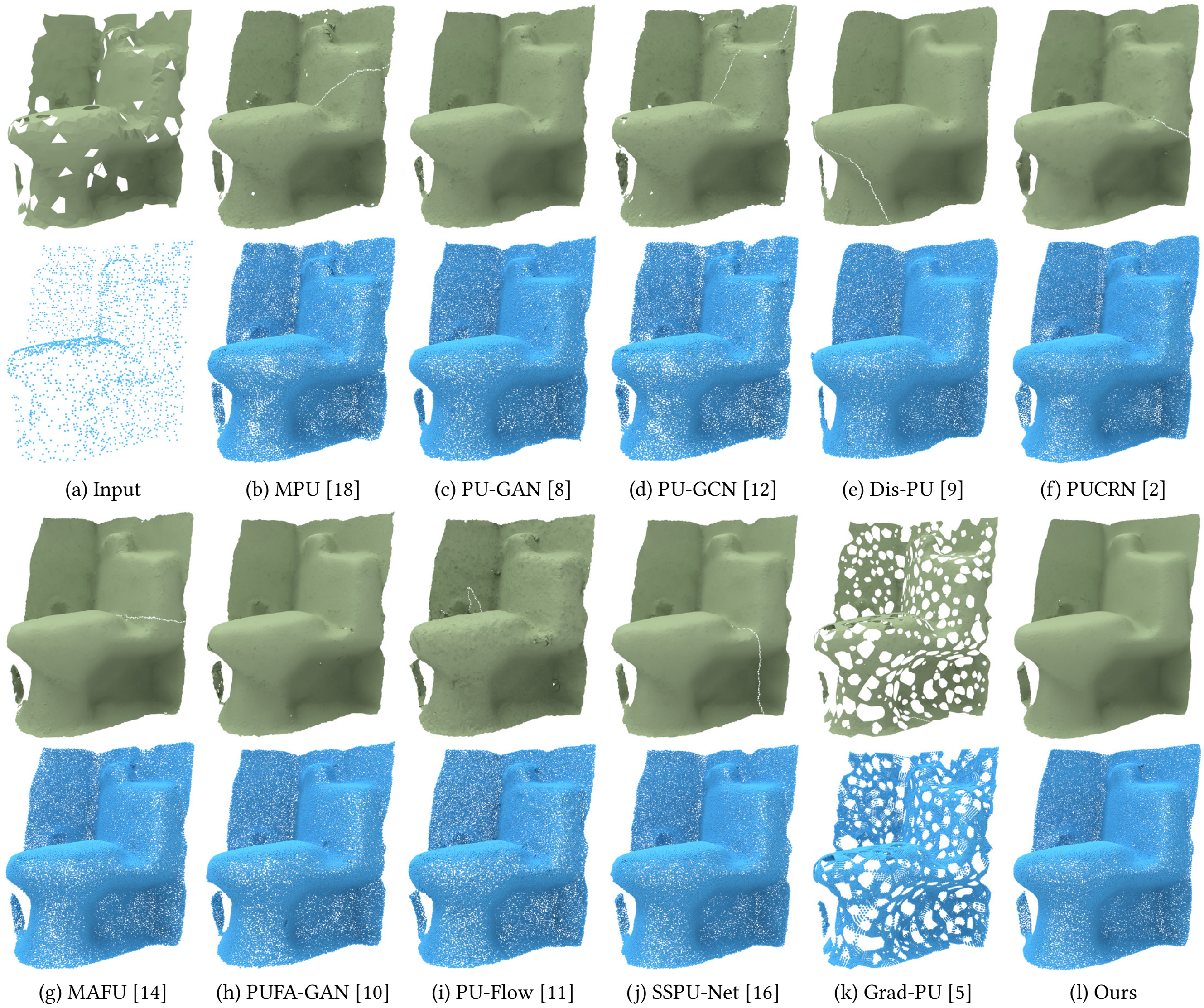
Figure 7: Visual comparison for $8\times$ upsampling on PU-GAN dataset.



(a) Input (b) MPU [18] (c) PU-GAN [8] (d) PU-GCN [12] (e) Dis-PU [9] (f) MAFU [14]

(g) Ground Truth (h) PUFA-GAN [10] (i) PU-Flow [11] (j) SSPU-Net [16] (k) Grad-PU [5] (l) Ours

Figure 8: Visual comparison for 8× upsampling on PUGeo-Net dataset.



(a) Input (b) MPU [18] (c) PU-GAN [8] (d) PU-GCN [12] (e) Dis-PU [9] (f) PUCRN [2]

(g) MAFU [14] (h) PUFA-GAN [10] (i) PU-Flow [11] (j) SSPU-Net [16] (k) Grad-PU [5] (l) Ours

Figure 9: Visual comparison for 16× upsampling on ObjectScanNN dataset.

References

[1] Yimian Dai, Fabian Gieseke, Stefan Oehmcke, Yiquan Wu, and Kobus Barnard. 2021. Attentional Feature Fusion. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*. IEEE, 3559–3568. https://doi.org/10.1109/WACV48630.2021.00360

[2] Hang Du, Xuejun Yan, Jingjing Wang, Di Xie, and Shiliang Pu. 2022. Point Cloud Upsampling via Cascaded Refinement Network. In *Computer Vision - ACCV 2022 - 16th Asian Conference on Computer Vision, Macao, China, December 4-8, 2022, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 13841)*, Lei Wang, Juergen Gall, Tat-Jun Chin, Imari Sato, and Rama Chellappa (Eds.). Springer, 106–122. https://doi.org/10.1007/978-3-031-26319-4_7

[3] Ankit Goyal, Hei Law, Bowei Liu, Alejandro Newell, and Jia Deng. 2021. Revisiting Point Cloud Shape Classification with a Simple and Effective Baseline. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 3809–3820. http://proceedings.mlr.press/v139/goyal21a.html

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 770–778. https://doi.org/10.1109/CVPR.2016.90

[5] Yun He, Danhang Tang, Yinda Zhang, Xiangyang Xue, and Yanwei Fu. 2023. Grad-PU: Arbitrary-Scale Point Cloud Upsampling via Gradient Descent with Learned Distance Functions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 5354–5363. https://doi.org/10.1109/CVPR52729.2023.00518

[6] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2017. Densely Connected Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2261–2269. https://doi.org/10.1109/CVPR.2017.243

[7] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015 (JMLR Workshop and Conference Proceedings, Vol. 37)*, Francis R. Bach and David M. Blei (Eds.). JMLR.org, 448–456. http://proceedings.mlr.press/v37/ioffe15.html

[8] Ruihui Li, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. 2019. PU-GAN: A Point Cloud Upsampling Adversarial Network. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 7202–7211. https://doi.org/10.1109/ICCV.2019.00730

[9] Ruihui Li, Xianzhi Li, Pheng-Ann Heng, and Chi-Wing Fu. 2021. Point Cloud Upsampling via Disentangled Refinement. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 344–353. https://doi.org/10.1109/CVPR46437.2021.00041

[10] Hao Liu, Hui Yuan, Junhui Hou, Raouf Hamzaoui, and Wei Gao. 2022. PUFA-GAN: A Frequency-Aware Generative Adversarial Network for 3D Point Cloud Upsampling. *IEEE Trans. Image Process.* 31 (2022), 7389–7402. https://doi.org/10.1109/TIP.2022.3222918

[11] Aihua Mao, Zihui Du, Junhui Hou, Yaqi Duan, Yong-jin Liu, and Ying He. 2022. PU-Flow: a Point Cloud Upsampling Network with Normalizing Flows. *IEEE Transactions on Visualization and Computer Graphics* (2022), 1–14. https://doi.org/10.1109/TVCG.2022.3196334

[12] Guocheng Qian, Abdulellah Abualshour, Guohao Li, Ali K. Thabet, and Bernard Ghanem. 2021. PU-GCN: Point Cloud Upsampling Using Graph Convolutional Networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 11683–11692. https://doi.org/10.1109/CVPR46437.2021.01151

[13] Yue Qian, Junhui Hou, Sam Kwong, and Ying He. 2020. PUGeo-Net: A Geometry-Centric Network for 3D Point Cloud Upsampling. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XIX (Lecture Notes in Computer Science, Vol. 12364)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 752–769. https://doi.org/10.1007/978-3-030-58529-7_44

[14] Yue Qian, Junhui Hou, Sam Kwong, and Ying He. 2021. Deep Magnification-Flexible Upsampling Over 3D Point Clouds. *IEEE Trans. Image Process.* 30 (2021), 8354–8367. https://doi.org/10.1109/TIP.2021.3115385

[15] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. 2019. Revisiting Point Cloud Classification: A New Benchmark Dataset and Classification Model on Real-World Data. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 1588–1597. https://doi.org/10.1109/ICCV.2019.00167

[16] Jin Wang, Jiade Chen, Yunhui Shi, Nam Ling, and Baocai Yin. 2023. SSPU-Net: A Structure Sensitive Point Cloud Upsampling Network with Multi-Scale Spatial Refinement. In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, ON, Canada, 29 October 2023- 3 November 2023*, Abdulmotaleb El-Saddik, Tao Mei, Rita Cucchiara, Marco Bertini, Diana Patricia Tobon Vallejo, Pradeep K. Atrey, and M. Shamim Hossain (Eds.). ACM, 1546–1555. https://doi.org/10.1145/3581783.3613807

[17] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. 2019. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.* 38, 5 (2019), 146:1–146:12. https://doi.org/10.1145/3326362

[18] Yifan Wang, Shihao Wu, Hui Huang, Daniel Cohen-Or, and Olga Sorkine-Hornung. 2019. Patch-Based Progressive 3D Point Set Upsampling. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 5958–5967. https://doi.org/10.1109/CVPR.2019.00611

[19] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. 2018. FoldingNet: Point Cloud Auto-Encoder via Deep Grid Deformation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 206–215. https://doi.org/10.1109/CVPR.2018.00029

[20] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. 2018. PU-Net: Point Cloud Upsampling Network. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 2790–2799. https://doi.org/10.1109/CVPR.2018.00295