Supplementary Material for IPoD: Implicit Field Learning with Point Diffusion for Generalizable 3D Object Reconstruction from Single RGB-D Images

Yushuang Wu[1,2] Luyue Shi[1,2] Junhao Cai[4] Weihao Yuan[3] Lingteng Qiu[1,2]
Zilong Dong[3] Liefeng Bo[3] Shuguang Cui[2,1] Xiaoguang Han[2,1†]

[1]SSE, CUHKSZ [2]FNii, CUHKSZ [3]Alibaba Group [4]HKUST

1. More Details for the Method

Model Architectures For the Transformer-based implementation (Ours2), we use the same encoder for the image and partial point cloud input as in NU-MCC [1]. E_I is a Vision Transformer (ViT) with a 16×16 patch embedding layer based on 2D convolution of hidden dimension 768 and 12 Transformer layers, of which each layer consists of a 768-dimensional self-attention operator with 12 heads and a 3072-dimensional 2-layer MLP. So E_I extracts an image feature map of shape 196×768 for each input image of shape 224×224×3. We use an anchor prediction module following NU-MCC. The input point cloud P is first fed into a positional embedding layer to be embedded by 16 × 16 patch, which down-samples and serializes it from 224×224×3 into 196×768, and 12 Transformer layers are used as in E_I to extract point cloud features of shape 196×768. The two features are concatenated into 196×1536 for further decoding. For the query points, we use 2048 points to form one X_t in each training iteration. The encoder E_X first uses a positional embedding to raise the dimension of X_t from 3 to 512. Then another frequency-based embedding layer is used to embed the time step t into a vector of length 20, with a linear layer mapping it into a vector with two values as the scale and shift factor, respectively. The two factors are then used in the affine transformation of X_t's embedding. The embedding of X_t finally goes through a linear layer to get the feature shape of 2048×512. In the decoding stage, we use the same anchor prediction module to speed up the decoding, where 200 anchor points with 512-dimensional features are decoded using the input features at first. Then, the anchor features and query point features are used in a feature aggregation module to compute ν' first, and then concatenated with the feature of X_t to compute ϵ', both of which rely on a Transformer-based decoder plus an MLP for output. The feature aggregation module is implemented with two cross-attention layers of 512 hidden dimensions and five 512-dimensional linear layers with residual shortcuts. The architecture details of the anchor decoder and the feature aggregation module are introduced in the NU-MCC paper [1]. The prediction head for ν is implemented with a linear layer and the noise prediction head is constructed based on an MLP with hidden dimension 128, and the MLP is implemented with two 1d convolution layers and a group normalization layer of 8 groups.

For the PVCNN-based implementation (Ours1), we adopt a PVCNN [2] as the decoders, and the detailed architecture of PVCNN follows the implementation in PC^2 [3] and PVD [8], where the point features are extracted by a point-based CNN (PointNet++ [4]), and another voxel-based branch simultaneously aggregate the nearby features for each point in a more hierarchical manner. At the encoding stage, we employ the same implementation of E_X as in the Transformer-based one. We use a ViT (vit_small_patch16_224_msn[1]) as the image encoder E_I that output 384-dimensional features. The partial point cloud encoder E_P uses a 60-dimensional positional embedding layer plus a 1d convolutional layer to get 512-dimensional point-wise features for PVCNN decoding. Besides, we use the same projection conditioning method proposed in PC^2 [3]. It works by projecting the image features onto each point in P and X_t according to the geometry relation between each pixel and its corresponding 3D position. We randomly sample 16384 points in P and 4096 points in forming X_t. Therefore, the concatenated features for decoding are in the shape of 20480×896.

More Training and Inferring Details We sample 4096 query points as a noisy sample for both implicit field learning and diffusion learning in each iteration for training. At the inferring stage, we denoise 12 noisy samples in a parallel manner and combine all points to form the final output point cloud. In the Transformer-based implementation, we use an anchor prediction loss as in NU-MCC [1]. So the final loss function is the summation of a UDF value loss, an RGB prediction loss, an anchor prediction loss, and a noise prediction loss. The weighting factors are 1.0, 0.01,

[1] https://dl.fbaipublicfiles.com

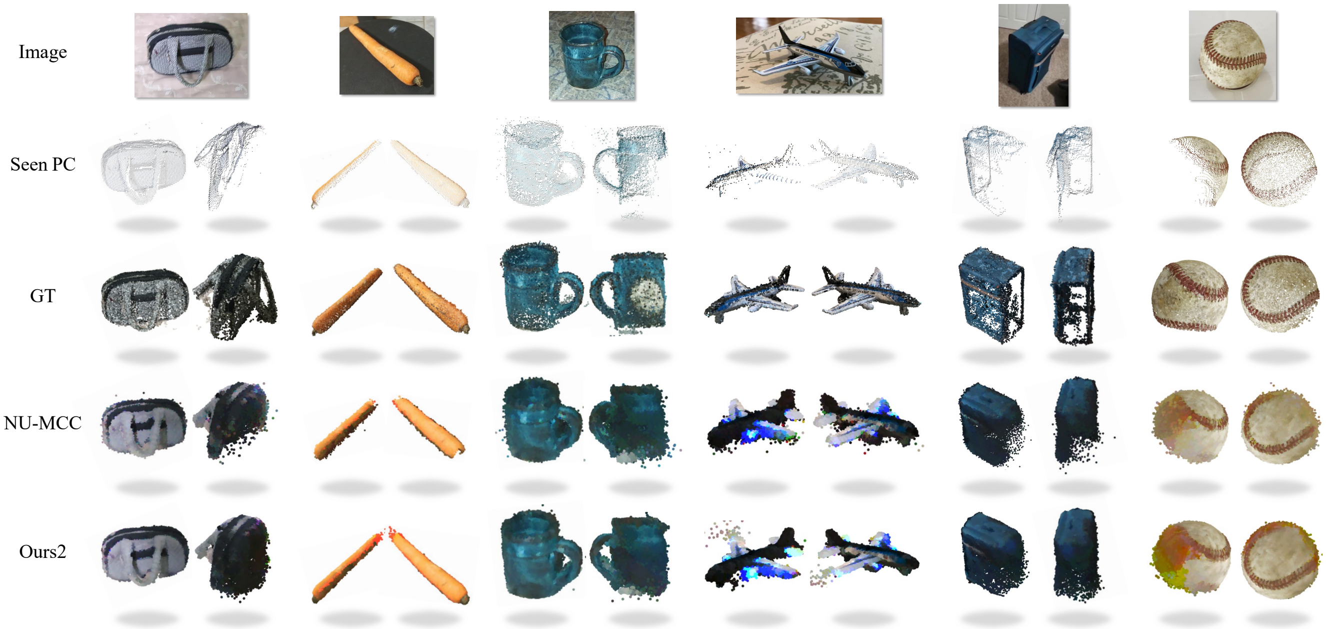

Figure S1. More qualitative comparison of reconstructions by Ours2 and NU-MCC, evaluated on CO3D-v2 held-out categories. We visualize two views for each seen, GT, and predicted point cloud by NU-MCC and Ours2.

0.03, and 1.0, following the practice in NU-MCC. In the UDF learning, we set the UDF value supervision clamping to 0.5 as in NU-MCC, which can take less care about the positions that are too far away from the object surface. At the inferring stage, we also use a post-processing manner as in NU-MCC, which takes four steps, first filtering out those points with UDF values larger than 0.23, secondly taking 10 times forwarding to shift the points along their gradient directions, thirdly computing the repulsive shift to adjust the point positions, and at last computing the corresponding RGB values at all of the final point positions. Such post-processing operations are used for a more fair comparison with NU-MCC, but note that they can only bring a 2% gain to the F-score according to our experiments and our proposed method does not rely on such operations.

2. More Details for the Experiments

Held-out CO3D categories In our experiments, we hold out 10 categories of CO3D-v2 [5] as ones in MCC [6] as the test set to follow MCC's setting. The 10 selected categories are: {*apple, ball, baseball-glove, book, bowl, carrot, cup, handbag, suitcase, toyplane*}. We randomly sample three views for all testing samples and average the scores on all views to get the average results.

Evaluation metrics Denote the predicted and the GT point cloud as $A \in \mathbb{R}^{N\times 3}$ and $B \in \mathbb{R}^{N'\times 3}$, and any point in them is denoted as a and b, respectively. We first define a thresholding function $\mu(\cdot)$:

$$\mu(x) = \begin{cases} 0, & \text{if } x \le \rho \\ 1, & \text{if } x > \rho \end{cases}$$

where $\rho = 0.1$ is a pre-defined threshold as in our paper. It is used to judge whether a point-to-point distance is within the threshold ρ.

The used evaluation metrics include:

Acc: $L1$ distance from the predicted to GT point cloud:

$$\mathrm{Acc}(A,B) = \sum_{a\in A} \min_{b\in B} |a-b|.$$

Comp: $L1$ distance from GT to the predicted point cloud:

$$\mathrm{Comp}(A,B) = \sum_{b\in B} \min_{a\in A} |a-b|.$$

Chamfer distance (CD): the summation of the above two:

$$\mathrm{CD}(A,B) = \mathrm{Acc}(A,B) + \mathrm{Comp}(A,B).$$

Prec: the ratio of generated points with correct predictions:

$$\mathrm{Prec}(A,B) = \frac{1}{N'} \sum_{a\in A} \mu(\min_{b\in B} |a-b|).$$

Recall: the ratio of recalled points in the GT point cloud:

$$\mathrm{Recall}(A,B) = \frac{1}{N} \sum_{b\in B} \mu(\min_{a\in A} |a-b|).$$

F-score: the harmonic mean of precision and recall:

$$\mathrm{Recall}(A,B) = \frac{2\times \mathrm{Prec}(A,B)\times \mathrm{Recall}(A,B)}{\mathrm{Prec}(A,B)+\mathrm{Recall}(A,B)}.$$

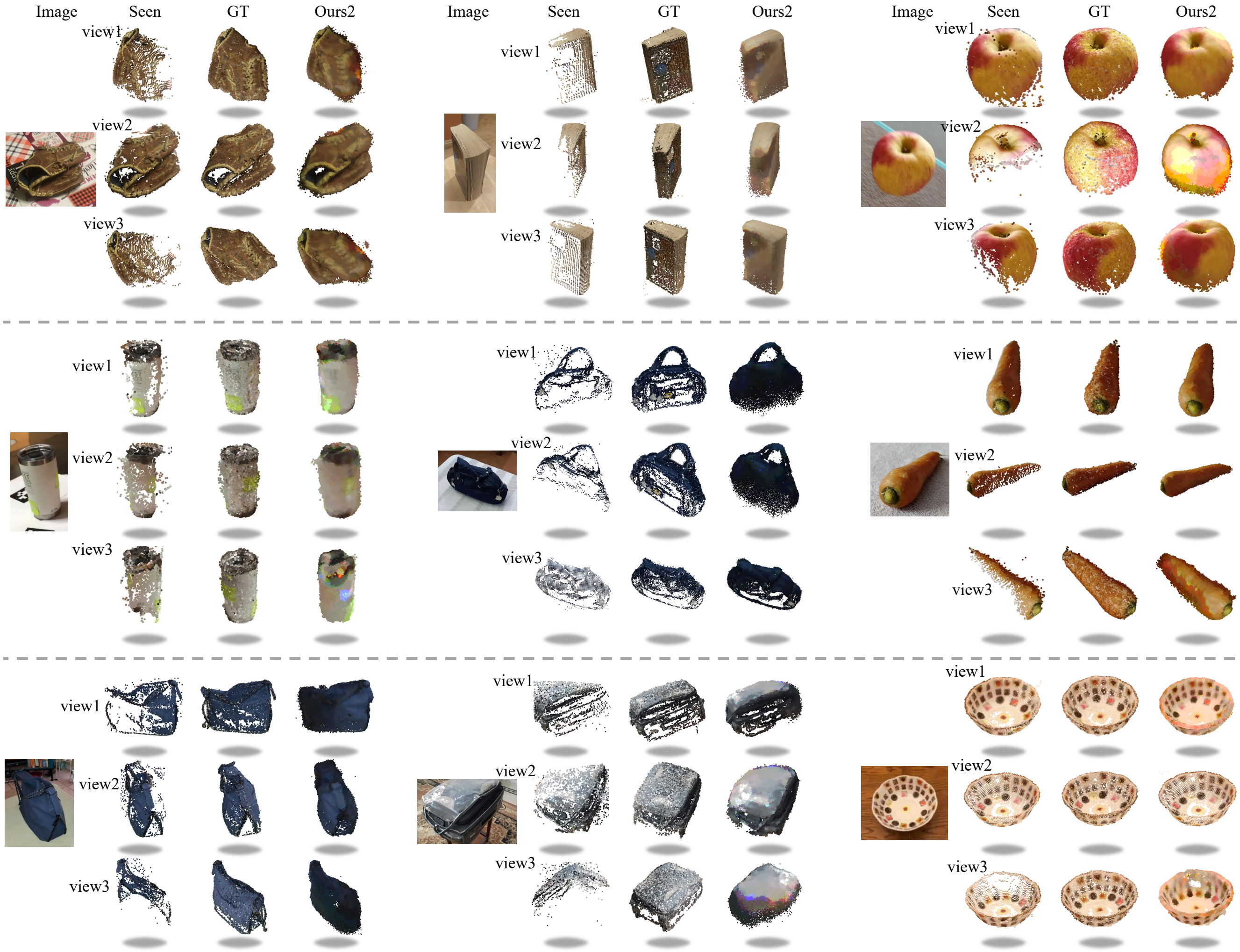

Figure S2. Visualization of reconstructions by Ours2, evaluated on CO3D-v2 held-out categories. We visualize three views for each seen point cloud, GT point cloud, and the predicted point cloud of Ours2.

Point Cloud Cleaning in MVImgNet We cleaned the reconstructed point clouds in MVImgNet [7] by removing samples with low completeness, which often consist of only a few points after multi-view reconstruction, usually caused by the surface material, color, lighting, video blur, etc. For some point clouds that exhibit significant deformation to the true object shape, we also remove them. Such deformations are usually caused by the camera pose estimation error. Besides, we also manually remove the significant noisy points in each point cloud. These noisy points are usually background points, preserved due to the error in object mask estimation. The cleaning is completed by a group of (more than 10) human annotators.

3. More Results

Visualization on CO3D-v2 Held-out Categories We provide more reconstruction results by our method implemented based on Transformer (Ours2) in Fig. S1, as an extension of Fig. 6 in the main paper. To highlight the comparison, we only choose the previous best baseline method NU-MCC and show the comparison between its reconstruction results and the ones produced by Ours2. As shown, our method can produce better reconstructions than NU-MCC on both completeness and precision. Besides, we also provide more visualization results of Ours2 in Fig. S2 to demonstrate the effectiveness of our method.

Visualization on CO3D-v2 Held-in Categories Except for evaluation on CO3D-v2 held-out categories, we also visualize the reconstruction results on testing samples from categories that are seen in training (held-in). As shown in Fig. S3, our method also gets better results than NU-MCC on held-in categories.

More results on MVImgNet We provide qualitative results of our method implemented with Transformer (Ours2) on the unseen categories from the MVImgNet dataset [7].

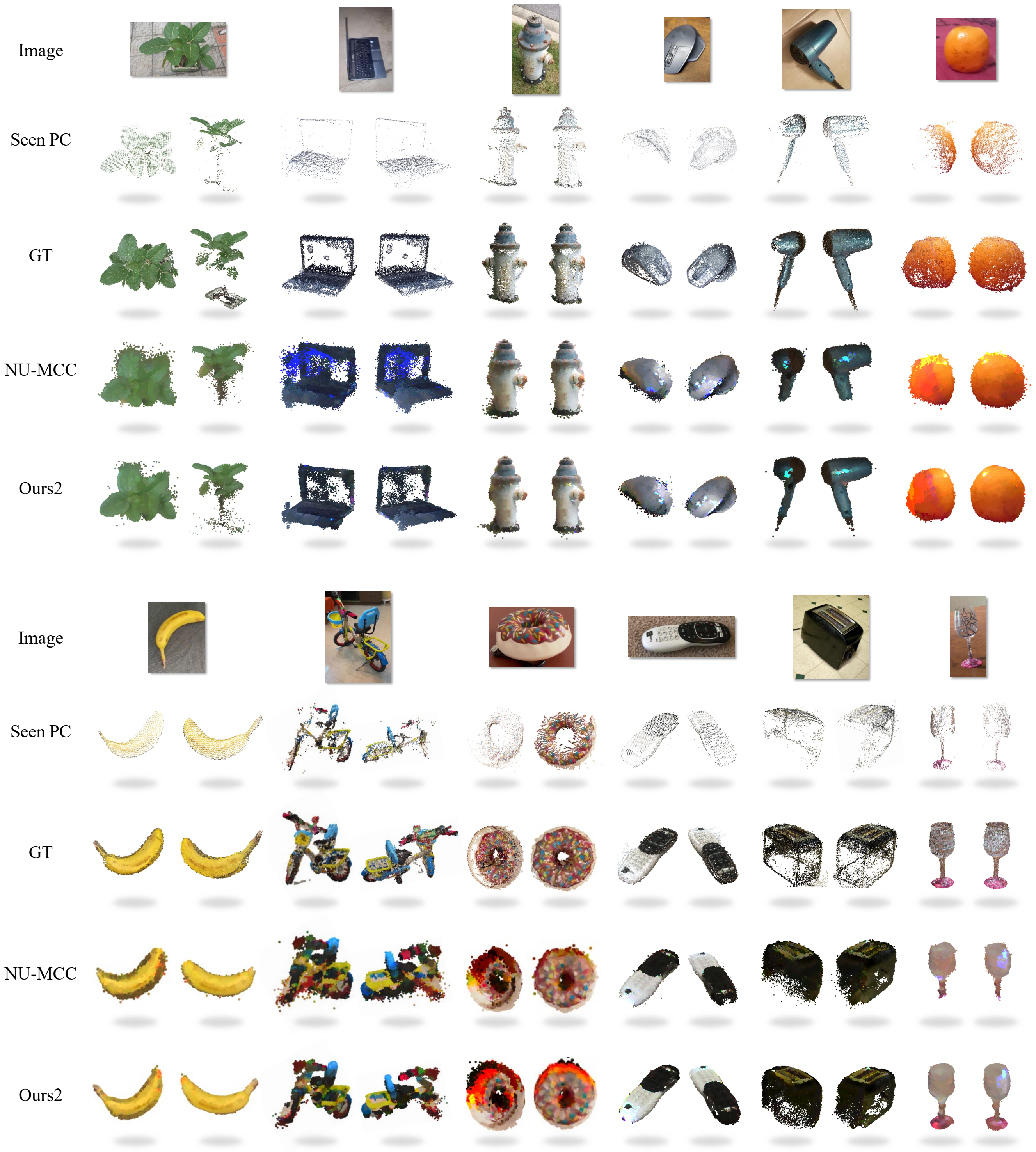

Figure S3. More qualitative comparison of reconstructions by Ours2 and NU-MCC, evaluated on CO3D-v2 testing samples from held-in categories. We visualize two views for each seen, GT, and predicted point cloud by NU-MCC and Ours2.

We visualize several generalization results by NU-MCC and Ours2 and also provide the quantitative evaluations for comparison in Fig. S4. As shown, Ours2 produces more accurate details and more complete shapes. Besides, we sampled 1k data from the cleaned MVImgNet point clouds as the test set for a larger-scale quantitative evaluation. As shown in Tab. S1, our method achieves 0.158 on CD and 0.903 F1-score, which significantly outperforms NU-MCC

Table S1. Quantitative comparison of reconstructions by NU-MCC and Ours2 on 1k samples from cleaned MVImgNet data.

Methods	Acc↓	Comp↓	CD↓	Prec↑	Recall↑	F1↑
NU-MCC	0.130	0.066	0.196	0.732	0.948	0.824
Ours2	**0.098**	**0.061**	**0.158**	**0.858**	**0.955**	**0.903**

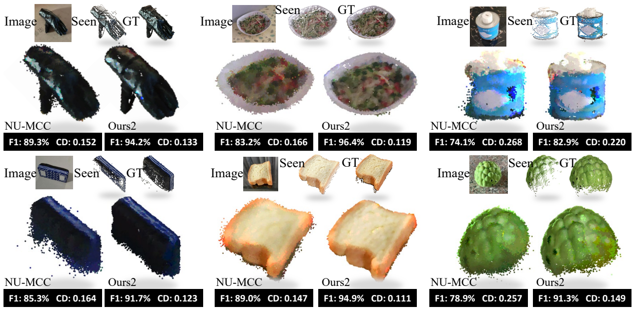

Figure S4. Visualization comparison between the reconstructions of NU-MCC and Ours2 on samples from MVImgNet. We also compute the F1-score and CD for each reconstructed sample.

on both reconstruction accuracy and completeness (0.196 on CD and 0.824 on F1-score). Results on MVImgNet further justify the superiority of the proposed method.

Visualization for ablation study on self-conditioning In the paper, we provide the quantitative results on the ablation of using the proposed self-conditioning mechanism or not. We additionally provide some qualitative comparison to better visualize the effect of the self-conditioning mechanism, as shown in Fig. S5.

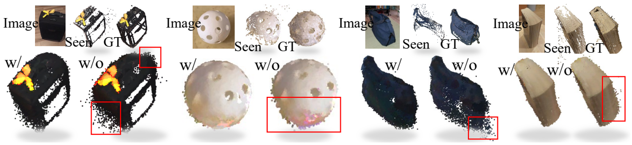

Figure S5. Qualitative results for ablation study on using (w/) self-conditioning or not (w/o). Red frames highlight the differences.

Video Visualization We provide videos for (i) the dynamic process (in inferring) visualization of implicit field learning with point diffusion for several samples, see "diffusion_process.mp4" in the compressed file; (ii) a 360° rotational display for several reconstructed point clouds by Ours2, see "rotational_display.mp4" in the compressed file.

References

[1] Stefan Lionar, Xiangyu Xu, Min Lin, and Gim Hee Lee. Nu-mcc: Multiview compressive coding with neighborhood decoder and repulsive udf. *arXiv preprint arXiv:2307.09112*, 2023. 1

[2] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019. 1

[3] Luke Melas-Kyriazi, Christian Rupprecht, and Andrea Vedaldi. Pc2: Projection-conditioned point cloud diffusion for single-image 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12923–12932, 2023. 1

[4] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017. 1

[5] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10901–10911, 2021. 2

[6] Chao-Yuan Wu, Justin Johnson, Jitendra Malik, Christoph Feichtenhofer, and Georgia Gkioxari. Multiview compressive coding for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9065–9075, 2023. 2

[7] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9150–9161, 2023. 3

[8] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5826–5835, 2021. 1