P2C: Self-Supervised Point Cloud Completion from Single Partial Clouds Supplementary Materials

Ruikai Cui[1] Shi Qiu[1] Saeed Anwar[2] Jiawei Liu[1] Chaoyue Xing[1] Jing Zhang[1] Nick Barnes[1]
[1]Australian National University [2]King Fahd University of Petroleum and Minerals

A. Technical Details on Implementation

Our network adopts an encoder-decoder structure, which we implemented using PyTorch [10]. To encode the input point cloud, we utilize the PCN architecture [19] which is a common choice for point cloud encoding [16, 3]. This encoder consists of two shared multi-layer perceptrons (MLPs) and two pointwise max-pooling layers, which are widely used in point cloud analysis [11, 13, 12, 15]. The input is represented as an $M \times 3$ matrix, where each row represents a point in 3D space with (x, y, z) coordinates. The number of points M can vary across inputs.

The encoder uses a shared MLP to transform each point p_i into a feature vector f_i. The MLP consists of two linear layers with ReLU activation [1]. Next, we apply a pointwise max-pooling to the features $\{f_i\}_{i=1}^{M}$, which produces a global feature vector $f'_g \in \mathbb{R}^{256}$. We then concatenate f'_g with each individual feature f_i and pass the resulting vectors through a second MLP and pooling layer. This process produces the final feature vector $f \in \mathbb{R}^{1024}$, which encodes the input point cloud. Our decoder is an MLP that consists of two hidden layers with 2048 hidden dimensions, followed by ReLU activation. The last linear layer outputs a vector $P_p \in \mathbb{R}^{6144}$, for which we then reshape as a 2048×3 matrix to represent a point cloud.

We train our network using the AdamW [8] optimizer. Moreover, we set the initial learning rate to $r = 10^{-3}$ and decay it by a factor of 0.9 every 20 epochs, with a minimum learning rate of $0.02 \times r$. We use a batch size of 32 and train our model for each category of the datasets, which takes an average of 8 hours on a Nvidia RTX 3090 GPU.

B. Dataset Details

We use the 3D-EPN [5] and PCN [19] datasets for synthetic evaluation, which are derived from ShapeNet [2]. We also utilize the ScanNet [4] dataset collected from the actual world for real-world evaluation. The details about each dataset are as below:

3D-EPN Dataset. The 3D-EPN dataset [5] is commonly used for evaluating unpaired point cloud completion methods. It includes 35,831 objects from eight diverse categories. Partial point clouds in the dataset contain 2048 points and were generated by transforming 2.5D depth maps into 3D coordinates given eight fixed camera poses. Ground truth complete point clouds were obtained by uniformly sampling 2048 points from the object surfaces.

PCN Dataset. The PCN dataset, introduced by Yuan *et al.* [19], is commonly used for evaluating supervised point cloud completion methods. It contains eight categories including 28,974 objects in the training set, 800 objects in the validation set, and 1200 objects in the test set. The complete point clouds are generated by uniformly sampling 16,384 points from CAD models, and the partial point clouds are obtained by back-projecting depth images from eight viewpoints into 3D space. Each partial sample originally contains an average of 1051 points. To ensure consistency across samples with varying point counts, we follow the common practice [18, 19, 3, 14] of padding the data with zeros to yield partial samples with 2048 points during training.

ScanNet Dataset. The ScanNet [4] dataset is a benchmark for large-scale 3D indoor scene understanding. It contains 2.5 million 3D scans of real-world indoor spaces captured by RGB-D cameras. Bounding-box annotations are used to identify chair and table objects with unknown incompleteness. We collect 4357 chair scans 1271 table scans for the training set, while the validation set contains 1368 chairs and 350 tables. Objects have 869 points on average. Unlike synthetic datasets where both object orientation and position are aligned, we only align the positions of the objects extracted from the ScanNet dataset, creating a more challenging scenario.

C. Evaluation Metric Details

We introduce detailed evaluation metrics including Chamfer Distance, Unidirectional Chamfer Distance, Unidirectional Hausdorff Distance, and Minimal Matching Distance.

Chamfer Distance (CD) [19] is a popular metric used to measure the similarity between two sets of points. The CD-ℓ_2 between two sets of points S_1 and S_2 can be expressed

Table 1. MMD ↓ ($\times 10^3$) improvement of P2C compared with the baseline.

Method	Chair	Table
Baseline	28.2	25.7
P2C(Ours)	14.1	8.1

as:

$$CD(S_1, S_2) = \frac{1}{|S_1|}\sum_{x\in S_1}\min_{y\in S_2}\|x-y\|_2 + \frac{1}{|S_2|}\sum_{y\in S_2}\min_{x\in S_1}\|x-y\|_2 \quad (1)$$

where $\|\cdot\|_2$ denotes ℓ_2-norm.

Unidirectional Chamfer Distance (UCD) [16] is a variant of Chamfer Distance that only considers the unidirectional distance from each point in one set to its nearest point in the other set. Therefore, it measures the similarity of one set of points to another, but not the other way around. Mathematically, the UCD from set S_1 to set S_2 can be expressed as:

$$UCD(S_1, S_2) = \frac{1}{|S_1|}\sum_{x\in S_1}\min_{y\in S_2}\|x-y\|_2 \quad (2)$$

Unidirectional Hausdorff Distance (UHD) [7] is another popular metric used to measure the similarity between two sets of points. It represents the maximum distance from each point in one set to its nearest point in the other set. The UHD from set S_1 to set S_2 can be expressed as:

$$UHD(S_1, S_2) = \max_{x\in S_1}\min_{y\in S_2}\|x-y\|_2 \quad (3)$$

Minimal Matching Distance (MMD) is a metric that is closely related to CD. It measures the minimum distance between one set S_1 and a set S_2 in a collection $\mathcal{C}$. Specifically, we follow the common practice [18] and uses the ShapeNet [2] as the collection $\mathcal{C}$, where the chair category and the table category both have 5750 complete samples. The MMD is defined as:

$$MMD(S_1, \mathcal{C}) = \min_{S_2\in\mathcal{C}} CD(S_1, S_2) \quad (4)$$

We employ CD as the evaluation metric for synthetic datasets because ground truth shapes are available, whereas for real-world datasets where ground truth completions are unavailable, we use UCD, UHD, MMD, and our proposed RCD. The UCD, UHD, and RCD measures the preservation of observed regions in the prediction, and MMD measures the completeness of the prediction.

D. More Real-World Results

We present additional samples from the ScanNet dataset in Fig. 1, where we also include the corresponding MMD scores for each prediction. Our proposed P2C method consistently generates superior visual results compared to models that are pre-trained on the ShapeNet dataset. Specifically, our approach yields shapes that better preserve the observed regions while simultaneously recovering the missing parts. For example, as illustrated in Fig. 1 (e), our method successfully recovers the legs of the chair, while in Fig. 1 (g), it fills the empty part on the top of the table.

It is worth noting that, although the visual result is impressive, the MMD score of our method is not leading due to the domain gap between real-world and synthetic data. Specifically, we test on real-world data and do not have training data other than being self-supervised on the ScanNet examples. Since MMD employs ShapeNet as the complete sample collection, our method has a domain gap, whereas, the other methods are trained on ShapeNet-derived datasets, and so they are better suited for predicting point clouds that resemble ShapeNet shapes, leading to smaller MMD scores. Consider Fig. 1 (a), our method produces a shape that is largely consistent with the input point cloud, while the PCN prediction has little distinction between possible a back-rest and arm-rests, yet has the best MMD score. This shows that MMD might be misleading when a domain gap exists and there is a trade-off between MMD score and prediction quality.

Overall, the MMD score of a prediction indicates completeness, but not necessarily correctness or validity of the shape. To establish a baseline, we provide an additional result in Tab 1, which shows the average MMD for ScanNet objects. Our method improved the baseline MMD by 14.1 and 17.6 for chair and table, respectively. Furthermore, our P2C approach significantly outperforms other methods in terms of fidelity metrics, as presented in Tab. **??** of the main paper, indicating that our method can complete partial objects while preserving observed regions.

E. More Qualitative Results on 3D-EPN

Fig. 2 and Fig. 3 show more qualitative results produced by P2C on 3D-EPN, further demonstrating the effectiveness of our method in completing semantically missing parts. For instance, our P2C approach is capable of recovering finer details in the tail of the airplane (Fig. 2 (b)) and preserving more observed structures in the sofa (Fig. 3 (l)) sample.

F. Hyperparameter Selection

We conduct an empirical study to investigate the impact of hyperparameters. In particular, we examine the effect of the number of patches on the model's performance, which is presented in Tab. 2. This parameter controls the granularity of the patches, and we evaluate three cases using 32, 64, and 128 patches, respectively. The experiments are based on

Figure 1. More qualitative results on the ScanNet dataset with MMD $\downarrow$ ($\times 10^3$) for each sample. The best MMD among predictions for each object is marked red.

the model variant using only RCD-based reconstruction and completion losses. The local patches is set to contain 64, 32, and 16 points for the respective divisions. Our results demonstrate that using 64 patches leads to the best overall performance.

Additionally, we investigated the effect of patch ratios ($r_1 : r_2$), where r_1 corresponds to G_{rec} and r_2 to G_{com}. For this study, we set the number of patches to 64 and

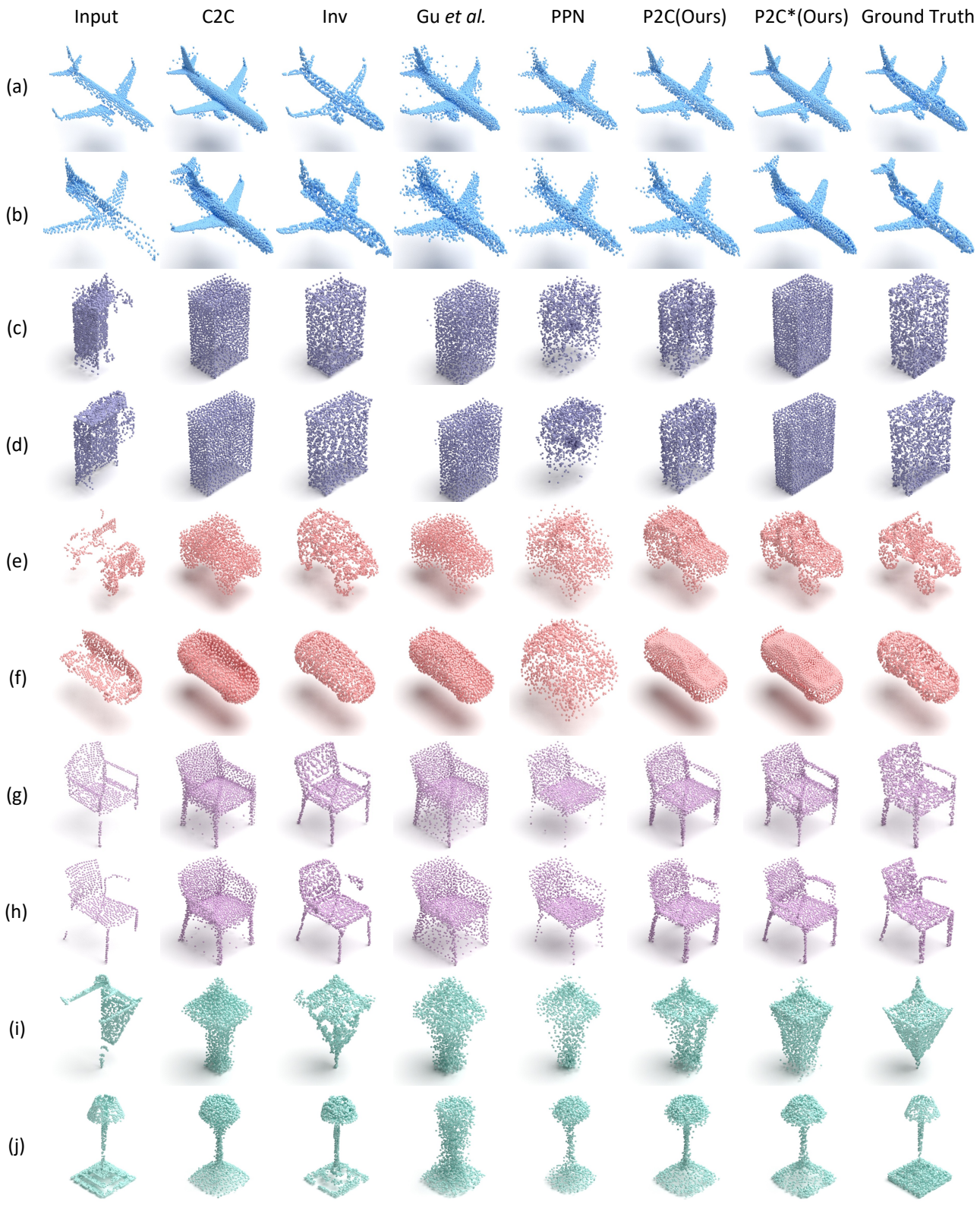

Figure 2. More qualitative results on the 3D-EPN dataset.

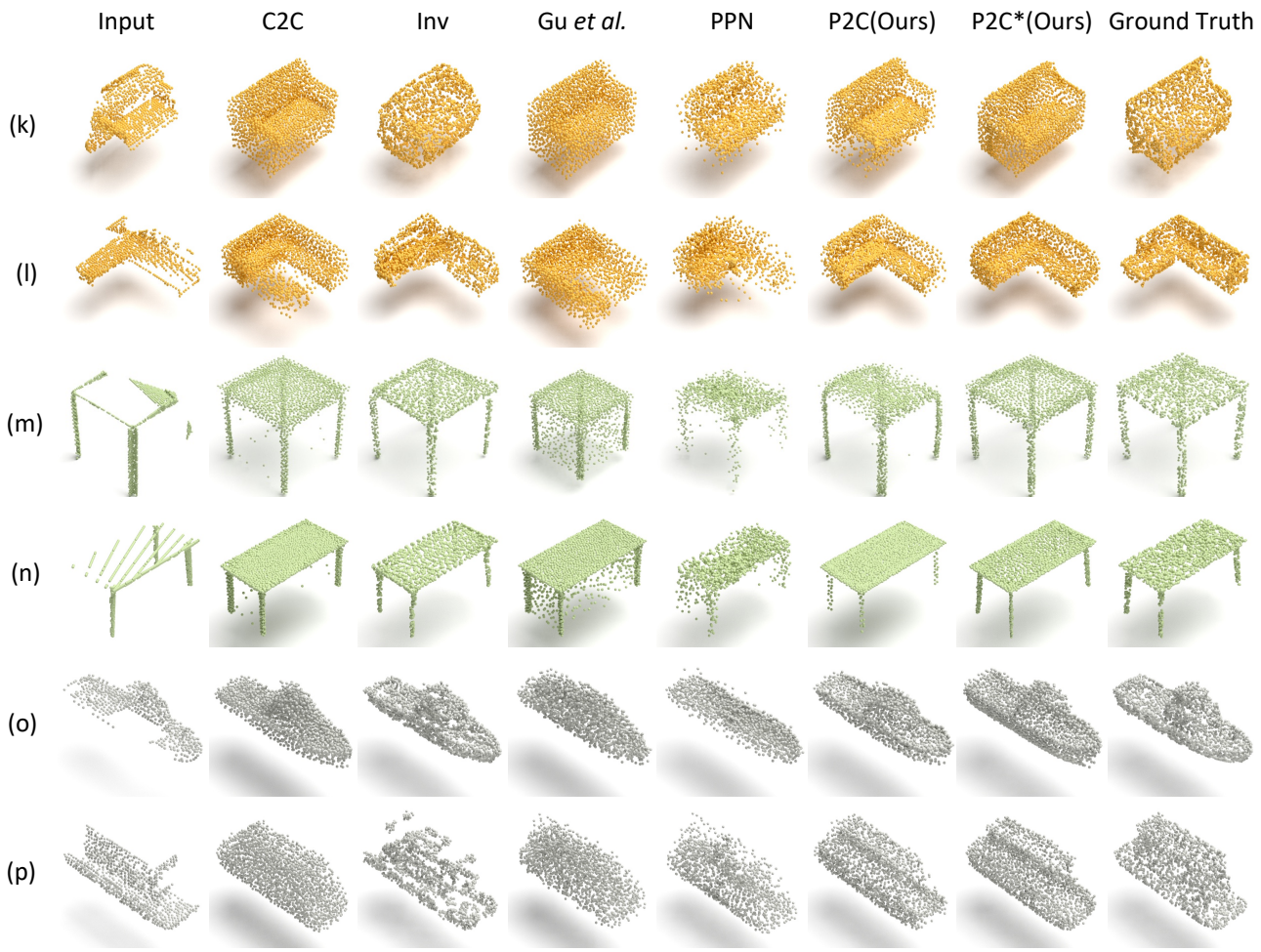

Figure 3. More qualitative results on the 3D-EPN dataset. (Continued)

train our P2C with different ratios. The results, presented in Tab. 3, illustrate a trade-off between reconstruction and completion patches. Selecting more patches for G_{rec} allows the model to observe more patches for shape representation. However, too many reconstruction patches leads to insufficient regions to learn what to complete, which restricts the completion capability of our method. This is demonstrated when the ratio is set as 40:20 or 50:10. On the other hand, if too many patches are selected for completion, i.e., unobservable regions, the model may not have sufficient regions to learn the underlying complete object, leading to a performance decrease. Our empirical findings indicate that setting the ratio as 20:40 yields the best performance.

G. Complexity and Efficiency Analysis

We provide a detailed complexity and efficiency analysis of our method in Tab. 4. Specifically, we report the number of parameters and frames per second (fps) for our method and six other methods, as well as the average CD-ℓ_2 as a reference metric. The fps is measured on a Nvidia RTX 3090 GPU with a batch size of 1. The results show that our model

Table 2. The effect of different number of patches for model performance in CD-$\ell_2 \downarrow$ scaled by 10^4.

# Patch	Average	Plane	Car	Chair	Table
32	15.3	4.8	15.7	16.9	23.6
64	**13.5**	4.7	**14.2**	**14.4**	**20.8**
128	14.4	**4.6**	16.3	15.7	21.0

Table 3. The effect of different ratio of patches for model performance in CD-$\ell_2 \downarrow$ scaled by 10^4.

Ratio	Average	Plane	Car	Chair	Table
10:50	14.1	**4.1**	14.9	15.7	21.6
20:40	**13.5**	4.7	14.2	**14.4**	20.8
30:30	13.9	4.7	**13.6**	16.9	**20.4**
40:20	15.1	4.8	14.3	19.1	22.1
50:10	16.2	5.1	14.5	20.5	24.5

Table 4. Complexity and efficiency analysis in terms of the number of parameters (Params) and frames per second (fps) with the average Chamfer Distance on the 3D-EPN dataset as references. The * indicates the result for P2C*

Method	Params ↓	fps ↑	Avg. CD-ℓ_2 ↓
Folding[17]	2.4M	21.0	6.8
PCN [19]	4.1M	20.6	7.4
C2C [16]	68.1M	2.0	14.3
Inv [20]	41.0M	0.01	23.6
Gu *et al.* [6]	9.2M	2.3	21.3
PPN [9]	2.4M	1.9	28.1
P2C(Ours)	23.9M	21.3	14.1/10.9*
- w/o Decoder	0.8M	N/A	N/A

Table 5. The effect of various input sparsity measured in CD-ℓ_2 scaled by 10^4.

#Points	Average	Plane	Car	Chair	Table
64	13.9	4.5	15.8	15.9	19.2
256	11.4	4.4	9.5	13.7	17.9
1024	11.2	4.3	8.6	13.6	18.3
2048	**11.1**	**4.3**	**8.6**	**13.5**	**18.1**

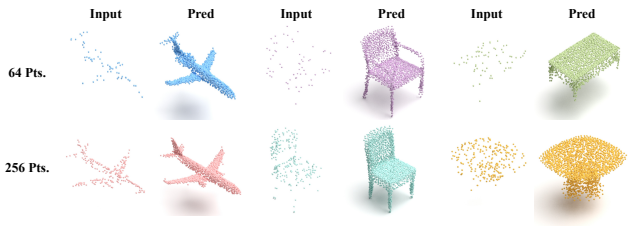

Figure 4. Completion results of our method under severely sparse input (64 or 256 points)

has more parameters than four out of six models, where the decoder consumes most of the parameters despite it being a simple MLP. However, our method still achieves a relatively high fps. Notably, our framework is flexible and can accommodate changes in the network architecture. Thus, we can implement our network using other lightweight or effective models to achieve a better overall balance between the efficiency and the effectiveness.

H. Robustness

We evaluate the robustness of the P2C framework by completing objects of varying input resolutions. We randomly downsample the original partial point cloud to different densities and evaluate the quantitative results, which are shown in Tab. 5. The results indicate that our method can work well even in severely sparse cases. Additionally, visual results presented in Fig. 4 demonstrate that our method can reliably generate complete shapes in various input resolutions, even in the cases of severe degradation.

I. Limitations

Although P2C has demonstrated promising results in completing point clouds with only single partial data needed for learning, several limitations still need to be addressed. For example, the effectiveness of the training process heavily relies on the representativeness of the training samples, as the model needs to learn an unbiased shape prior for the corresponding category: *e.g.*, if all training samples of chairs have no legs, the model will not be capable of completing legs for such biased chair observations.

References

[1] Abien Fred Agarap. Deep learning using rectified linear units (relu). *CoRR*, abs/1803.08375, 2018. 1

[2] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. 1, 2

[3] Ruikai Cui, Shi Qiu, Saeed Anwar, Jing Zhang, and Nick Barnes. Energy-based residual latent transport for unsupervised point cloud completion. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*, page 48. BMVA Press, 2022. 1

[4] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2432–2443. IEEE Computer Society, 2017. 1

[5] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6545–6554. IEEE Computer Society, 2017. 1

[6] Jiayuan Gu, Wei-Chiu Ma, Sivabalan Manivasagam, Wenyuan Zeng, Zihao Wang, Yuwen Xiong, Hao Su, and Raquel Urtasun. Weakly-supervised 3d shape completion in the wild. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V*, volume 12350, pages 283–299. Springer, 2020. 6

[7] Taifeng Li, Quanke Pan, Liang Gao, and Peigen Li. A novel simplification method of point cloud with directed hausdorff distance. In *2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 469–474, 2017. 2

[8] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[9] Himangi Mittal, Brian Okorn, Arpit Jangid, and David Held. Self-supervised point cloud completion via inpainting. *arXiv preprint arXiv:2111.10701*, 2021. 6

[10] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 1

[11] Shi Qiu, Saeed Anwar, and Nick Barnes. Dense-resolution network for point cloud classification and segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3813–3822, 2021. 1

[12] Shi Qiu, Saeed Anwar, and Nick Barnes. Geometric back-projection network for point cloud classification. *IEEE Transactions on Multimedia*, 24:1943–1955, 2021. 1

[13] Shi Qiu, Saeed Anwar, and Nick Barnes. Semantic segmentation for real point cloud scenes via bilateral augmentation and adaptive fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1757–1767, 2021. 1

[14] Shi Qiu, Saeed Anwar, and Nick Barnes. Pu-transformer: Point cloud upsampling transformer. In *Proceedings of the Asian Conference on Computer Vision*, pages 2475–2493, 2022. 1

[15] Shi Qiu, Saeed Anwar, and Nick Barnes. Pnp-3d: A plug-and-play for 3d point clouds. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(1):1312–1319, 2023. 1

[16] Xin Wen, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Cycle4completion: Unpaired point cloud completion using cycle transformation with missing region coding. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 13080–13089. Computer Vision Foundation / IEEE, 2021. 1, 2, 6

[17] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. 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*, pages 206–215. Computer Vision Foundation / IEEE Computer Society, 2018. 6

[18] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *ICCV*, 2021. 1, 2

[19] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision*, pages 728–737. IEEE Computer Society, 2018. 1, 6

[20] Junzhe Zhang, Xinyi Chen, Zhongang Cai, Liang Pan, Haiyu Zhao, Shuai Yi, Chai Kiat Yeo, Bo Dai, and Chen Change Loy. Unsupervised 3d shape completion through GAN inversion. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 1768–1777. Computer Vision Foundation / IEEE, 2021. 6