Supplementary Material – Diverse Shape Completion via Style Modulated Generative Adversarial Networks

Wesley Khademi
Oregon State University
khademiw@oregonstate.edu

Li Fuxin
Oregon State University
lif@oregonstate.edu

1 Overview

- Architecture (Section 2): we provide a detailed description of our architecture
- Datasets (Section 3): we provide a detailed description of datasets used in our experiments
- Metrics (Section 4): we formally define our evaluation metrics
- Baseline diversity penalty (Section 5): we discuss an alternative diversity penalty which we treat as a baseline in our ablations
- More results (Section 6): we share more qualitative results of our method
- Limitations (Section 7): we discuss some of the limitations of our method

2 Architecture

2.1 Partial encoder

Our partial encoder takes in a partial point cloud $X_P \in \mathbb{R}^{1024\times 3}$ and first produces a set of point-wise features $F_0 \in \mathbb{R}^{1024\times 16}$ via a 3-layer MLP with dims $[16, 16, 16]$. To extract local features, $L = 4$ PointConv [1] downsampling blocks are used, where the number of points are halved and the feature dimension is doubled in each block, producing a set of downsampled points $X_L \in \mathbb{R}^{128\times 3}$ with local features $F_L \in \mathbb{R}^{128\times 256}$. We use a neighborhood size of 16 for PointConv layers in our downsampling blocks. Additionally, a global partial shape vector $f_P \in \mathbb{R}^{512}$ is produced from concatenated $[X_L, F_L]$ via a 2-layer MLP with dims $[512, 512]$ followed by a max-pooling.

2.2 Style encoder

We represent our style encoder E_S as a learned Gaussian distribution $E_S(z|X) = \mathcal{N}(z|\mu(E(X)), \sigma(E(X)))$ where E is an encoder, μ and σ are linear layers, and X is a complete point cloud.

Encoder E follows a PointNet [2] architecture. In particular, encoder E takes in a complete point cloud $X \in \mathbb{R}^{2048\times 3}$ and passes it through a 4-layer MLP with dims [64, 128, 256, 512] followed by a max-pooling to aggregate the point-wise features into a single feature vector $f_S \in \mathbb{R}^{512}$. The global shape vector f_S is then passed through two separate linear layers to produce our style code distribution with parameters $\mu = \mu(f_S) \in \mathbb{R}^8$ and $\sigma = \sigma(f_S) \in \mathbb{R}^8$. During training, we sample style code z using the reparameterization trick:

$$z = \mu + \sigma \cdot \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, I) \tag{1}$$

We train our style encoder with the losses from our completion network. To enable sampling during inference, we also minimize the KL-divergence between $E_S(z|X)$ and a standard normal distribution during training:

$$\mathcal{L}_{KL} = \lambda_{KL}\, D_{KL}(E_S(z|X)\,||\,\mathcal{N}(0, I)) \tag{2}$$

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

where λ_{KL} is a weighting term (we set $\lambda_{KL} = 1e-2$ in our experiments).

2.3 Style modulator

Our style modulator network M takes in a partial latent vector $f_P \in \mathbb{R}^{512}$ and a style code $z \in \mathbb{R}^8$ as input and produces a newly styled partial shape latent vector $f_C \in \mathbb{R}^{512}$. The style modulator network is a 4-layer network consisting of style-modulated convolutions at every layer. The partial latent vector remains the same dimension (i.e., 512-dim) throughout the entire network and the style code z is injected at every layer through the style-modulated convolution. Note our style code only modulates the partial latent vector f_P and leaves local features F_L from our partial encoder untouched. We make this choice as F_L carries critical information about local geometric structure in the partially observed regions that we want to preserve.

2.4 Style-based seed generator

Our style-based seed generator takes in as input the downsampled partials points $X_L \in \mathbb{R}^{128\times3}$ with local features $F_L \in \mathbb{R}^{128\times256}$, global partial shape vector $f_P \in \mathbb{R}^{512}$, and sampled style code $z \in \mathbb{R}^8$ and produces Patch Seeds $(\mathcal{S}, \mathcal{F})$ as output.

To produce diverse Patch Seeds, we inject sampled style code z into f_P using our style modulator network to produce a styled partial shape vector $f_C = M(f_P, z) \in \mathbb{R}^{512}$. A set of upsampled features $F_{up} \in \mathbb{R}^{N_S\times C_S}$ are computed via an Upsample Transformer [3] using partial points X_L and features F_L. Upsampled features F_{up} are concatenated with styled partial shape vector f_C and passed through an MLP to produce Patch Seed features $\mathcal{F} \in \mathbb{R}^{N_S\times C_S}$. Finally, another MLP regresses Patch Seed coordinates $\mathcal{S} \in \mathbb{R}^{N_S\times3}$ from seed features $\mathcal{F}$ concatenated with styled partial shape vector f_C. Note we set $N_S = 256$ and $C_S = 128$ and a neighborhood size of 20 is used in the Upsample Transformer for computing local self-attention. We refer readers to the original SeedFormer [3] work for a full description of the Upsample Transformer.

2.5 Coarse-to-fine decoder

Note that our decoder starts from generated Patch Seeds $(\mathcal{S}, \mathcal{F})$, where we set our coarsest completion $\mathcal{G}_0 = \mathcal{S} \in \mathbb{R}^{256\times3}$. During this stage, the completion is upsampled by a factor r and refined through a series of upsampling layers to produce denser completions. We use 3 upsampling layers and set the upsampling rate $r = 2$. The output of our decoder is point clouds $\mathcal{G}_i$ for $i = 0, ..., 3$ with 256, 512, 1024, and 2048 points, respectively. Interpolated seed features and point features used in the Upsample Transformer at each upsampling layer share the same feature dimension size, which we set to 128. Seed features are interpolated using a PointConv layer with a neighborhood of size 8. The Upsample Transformer uses a neighborhood size of 20 for computing local self-attention.

2.6 Discriminator

We have a discriminator D_i for each output level $i = 0, ..., 3$ of our completion network. Each discriminator D_i shares the same architecture; however, they do not share parameters. In particular, each discriminator uses a PointNet-Mix architecture [4]. The discriminator D_i takes either a ground truth point cloud or completion $X \in \mathbb{R}^{N_i\times3}$, where N_i is the point cloud resolution at output level i of our decoder, and produces a prediction of whether the point cloud is real or fake. The point cloud X is first passed through a 4-layer MLP with dims $[128, 256, 512, 1024]$ producing a set of point-wise features $F \in \mathbb{R}^{N_i\times1024}$. The features F are then both max-pooled and average-pooled to produce two global latent features $f_{max} \in \mathbb{R}^{1024}$ and $f_{avg} \in \mathbb{R}^{1024}$, respectively. These features are concatenated to produce our mix-pooled feature $f_{mix} = [f_{max}, f_{avg}] \in \mathbb{R}^{2048}$ and passed through another 4-layer MLP with dims $[512, 256, 64, 1]$ to produce our final prediction.

3 Datasets

We conduct experiments on data from the ShapeNet [5], PartNet [6], 3D-EPN [7], Google Scanned Objects [8], and ScanNet [9] datasets, which are all publicly available. All datasets were obtained directly from their websites and permission to use the data was received for those that required it.

3.1 3D-EPN

For the 3D-EPN dataset [7], we evaluate on the Chair, Table, and Airplane categories and follow the train/test splits used in [10]. In particular the train/test splits are 4068/1171, 4137/1208, 2832/808 for the Chair, Table, and Airplane categories, respectively. The 3D-EPN dataset is derived from a subset of the ShapeNet dataset [5]. Ground truth complete point clouds are produced by sampling 2048 points from the complete shape's mesh uniformly. Partial point clouds are generated by virtually scanning ground truth meshes from different viewpoints to simulate partial scans from a LiDAR or depth camera.

3.2 PartNet

For the PartNet dataset [6], we evaluate on the Chair, Table, and Lamp categories and once again follow the train/test splits used in [10]. In particular the train/test splits are 4489/1217, 5707/1668, 1545/416 for the Chair, Table, and Lamp categories, respectively. Ground truth point clouds are generated by sampling 2048 points from the complete point cloud. To model part-level incompleteness, the semantic segmentation information provided by PartNet is used to produce partial point clouds. In particular, we randomly sample semantic part labels for each shape and remove all points corresponding to those part labels from the ground truth point cloud.

3.3 Google Scanned Objects

For the Google Scanned Objects dataset [8], we evaluate on the Shoe, Toys, and Consumer Goods categories. We choose these categories as they are the three largest categories in the dataset containing 254, 147, and 248 meshes, respectively. Meshes of the objects in each category were acquired via a high-quality 3D scanning pipeline and we generate ground truth point clouds by uniformly sampling 2048 points from the mesh surface. To generate partial point clouds, we virtually scan each mesh from 8 random viewpoints to simulate partial scans from a sensor. We use 7 of the partial views for training and holdout 1 unseen view per object for testing.

3.4 ScanNet

For the ScanNet dataset [9], we use the preprocessed data provided by [11]. In particular, chair object instances are extracted from ScanNet scenes and manually aligned to ShapeNet data. Since there are no ground truth completions for these objects, we use our model pre-trained on the Chair category from the 3D-EPN dataset and provide some qualitative results on real scanned chairs from ScanNet.

4 Metrics

We define the quantitative metrics used to evaluate our method against other baselines on the task of multimodal shape completion. We first define the Chamfer Distance between two point clouds, which is used by several of our evaluation metrics. In particular, the Chamfer Distance between point clouds $P \in \mathbb{R}^{N \times 3}$ and $Q \in \mathbb{R}^{M \times 3}$ can be defined as:

$$d^{CD}(P, Q) = \frac{1}{|P|} \sum_{x \in P} \min_{y \in Q} \|x - y\|_2^2 + \frac{1}{|Q|} \sum_{y \in Q} \min_{x \in P} \|x - y\|_2^2 \tag{3}$$

For our evaluation metrics, we let $\mathcal{T}$ represent the test set of ground truth complete point clouds and $\mathcal{P}$ be the test set of partial point clouds. For each $p_i \in \mathcal{P}$, we produce K completions c_{ij} for $j = 1, ..., K$ to construct a completion set $\mathcal{C} = \{c_{ij}\}$.

4.1 Minimal Matching Distance (MMD)

Minimal matching distance measures how well the test set of complete point clouds $\mathcal{T}$ is covered by the completion set $\mathcal{C}$. In particular, for each ground truth complete shape $t \in \mathcal{T}$, it finds its most similar point cloud in the completion set $\mathcal{C}$ and computes the Chamfer Distance between them:

$$MMD = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \left(\min_{c \in \mathcal{C}} d^{CD}(t, c) \right) \tag{4}$$

4.2 Total Mutual Difference (TMD)

Total mutual difference is a measure of how diverse generated completions are. For each partial shape $p_i \in \mathcal{P}$, each of the K completions c_{ij} for $j = 1, ..., K$ computes the average Chamfer Distance between itself and the other $K-1$ completions. The K average Chamfer Distances are then summed to produce a single value per $p_i \in \mathcal{P}$. TMD is then defined as the average of these values over partial input shapes $\mathcal{P}$:

$$TMD = \frac{1}{|\mathcal{P}|} \sum_{i=1}^{|\mathcal{P}|} \left(\sum_{j=1}^{K} \frac{1}{K-1} \sum_{1 \leq l \leq K, l \neq j} d^{CD}(c_{ij}, c_{il}) \right) \tag{5}$$

4.3 Unidirectional Hausdorff Distance (UHD)

To measure how well the completions respect their partial inputs, we use unidirectional Hausdorff distance. We define the unidirectional Hausdorff distance d^{UHD} between point clouds $P \in \mathbb{R}^{N \times 3}$ and $Q \in \mathbb{R}^{M \times 3}$ as:

$$d^{UHD}(P, Q) = \max_{x \in P} \min_{y \in Q} \|x - y\|_2 \tag{6}$$

Then the metric we report in our evaluations is simply the average unidirectional Hausdorff distance from a partial point cloud $p_i \in \mathcal{P}$ to its K completions c_{ij} for $j = 1, ..., K$:

$$UHD = \frac{1}{|\mathcal{P}|} \sum_{p_i \in \mathcal{P}} \left(\frac{1}{K} \sum_{j=1}^{K} d^{UHD}(p_i, c_{ij}) \right) \tag{7}$$

5 Baseline diversity penalty

We discuss an alternative diversity penalty which we treat as a baseline in our ablation in Table 1. Instead of computing our diversity penalty in the discriminator's feature space, our baseline computes such a penalty directly on the output space of our completion network using Earth Mover's Distance (EMD).

Inspired by [12, 13], we construct a diversity penalty in the output space of our completion network. In the image space, one way in which this can be done is by maximizing the L1 norm of the per pixel difference between two images. However, the image space is a 2D-structured grid that enables direct one-to-one matching of pixels between images, while point clouds are unstructured and a one-to-one correspondence does not directly exist. To overcome this, we make use of the Earth Mover's Distance, which produces a one-to-one matching and computes the distance between these matched points. In particular, the EMD between two point clouds $P \in \mathbb{R}^{N \times 3}$ and $Q \in \mathbb{R}^{N \times 3}$ can be defined as:

$$d^{EMD}(P, Q) = \min_{\phi : P \to Q} \frac{1}{|P|} \sum_{x \in P} \|x - \phi(x)\|_2 \tag{8}$$

where $\phi : P \to Q$ is a bijection.

Now let X_P be a partial point cloud. We sample two style codes $z_1 \sim E_S(z|X_1)$ and $z_2 \sim E_S(z|X_2)$ from random complete shapes X_1 and X_2 to condition the completion of X_P on. Our completion network takes in partial input X_P and style code z and produces a completion $\mathcal{G}_i(X_P, z)$ at each output level i. Then an EMD-based diversity penalty can be defined as:

$$\mathcal{L}_{div} = \sum_{i=0}^{3} \frac{1}{d^{EMD}(\mathcal{G}_i(X_P, z_1),\ \mathcal{G}_i(X_P, z_2))} \tag{9}$$

Note, by minimizing Equation 9 we try to encourage our network to produce completions whose points do not have a high amount of overlap in 3D space for different style codes.

6 More results

In this section, we share more qualitative results from our multi-modal point cloud completion algorithm and conduct further ablations on our method.

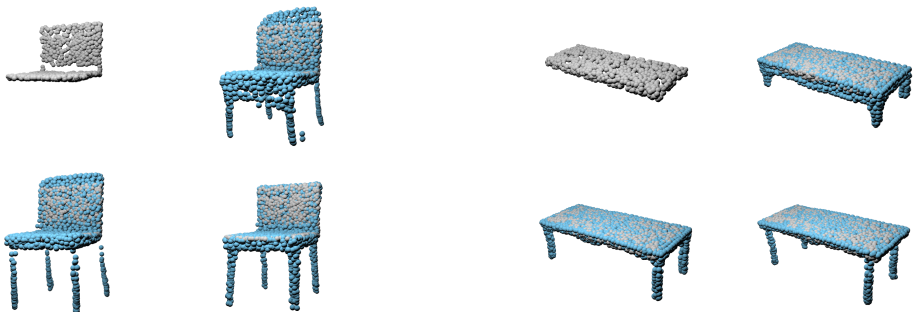

Figure 1: Visualization of partial shapes (gray) overlaid on completions from our method (blue).

6.1 Partial reconstructions

To see how well our method respects the partial input, we visualize the partially observed point cloud overlaid onto our completions. We show some of these results in Figure 1. It can be seen that the completions produced by our method well respect the partial input, which aligns with the low UHD values we observe in our quantitative results.

6.2 More completions

We share more multi-modal completions produced by our method in Figure 2. Our method is able to produce high-quality completions where we observe higher levels of diversity with increasing ambiguity in partial scans.

Additionally, in Figure 3, we share some example completions of real scanned chairs from ScanNet using our model pre-trained on the 3D-EPN dataset. Our model produces diverse completions with fairly clean geometry, suggesting we can even generalize well to real scans when trained on synthetic data.

6.3 Visualizing style codes

In Figure 4, we plot our learned style codes extracted from shapes in the training set by projecting them into 2D using principal component analysis (PCA). To better understand whether our style encoder is learning to extract style from the shapes, we visualize the corresponding shapes in random neighborhoods/clusters of our projected data. We find that the shapes contained in a neighborhood have a shared style or characteristic. For example, the chairs in the brown cluster all have backs whose top is curved while the black cluster has chairs that all have thin slanted legs.

6.4 Nearest neighbors of completions

In Figure 5, we share several completions (in blue) of a partial input and each completions nearest neighbor (in yellow) to a ground truth complete shape in the training set. Our method produces a different nearest neighbor for each completion of a partial input, demonstrating our methods ability to overcome conditional mode collapse. Additionally, each nearest neighbor is similar to the partially observed region and varies more in the missing regions, suggesting that our method is capturing plausible diversity in our completions that matches with variance in the ground truth shape distribution.

6.5 Ablations

In this section, we present another ablation on our method as well as share a qualitative comparison on some of our ablations.

In particular, we also explored training with an alternative diversity penalty, where the penalty is computed directly in the generator's output space by maximizing the Earth Mover's Distance (EMD) between two completions. In Table 1, we see that our proposed feature space penalty obtains better MMD and UHD compared to regularizing in the output space using EMD, suggesting our penalty

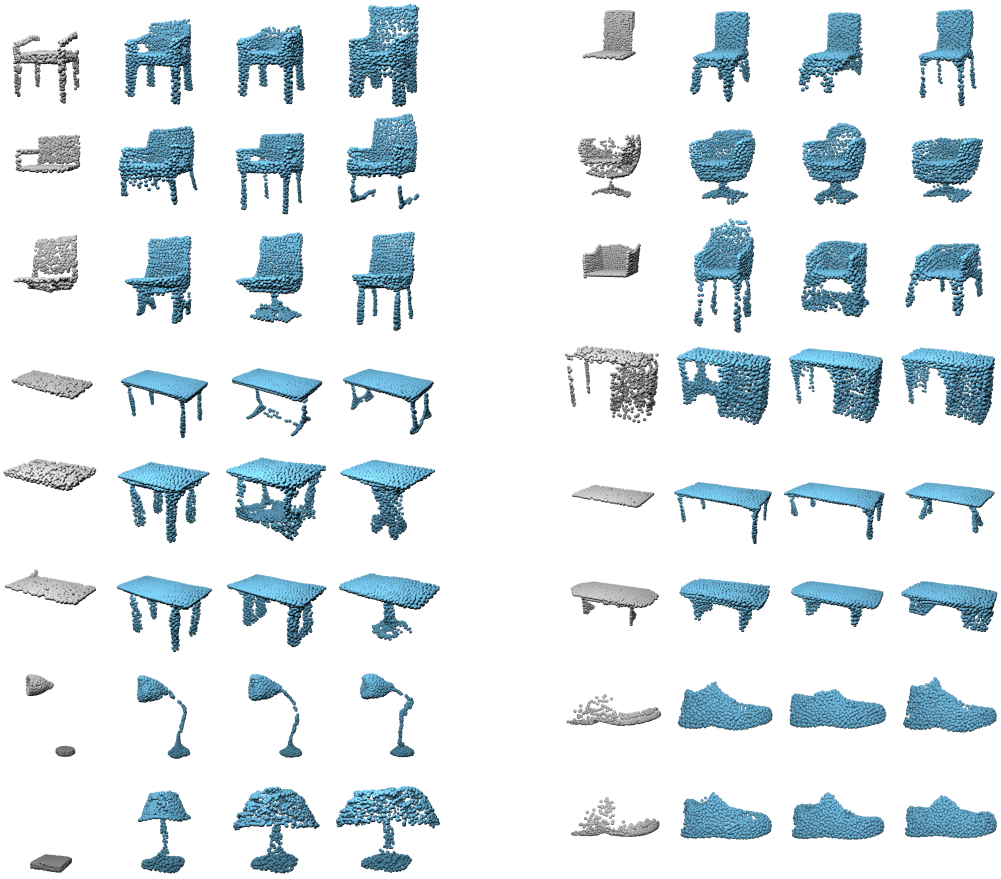

Figure 2: Example multi-modal completions (blue) of partial point clouds (gray) across several different categories from the PartNet, 3D-EPN, and Google Scanned Objects datasets.

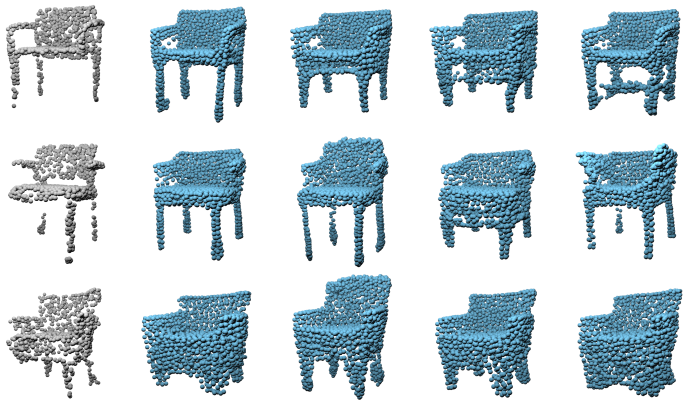

Figure 3: Qualitative results on real scanned chairs from ScanNet.

leads to higher quality and more plausible completions. Interestingly, the EMD diversity penalty obtains a high TMD, suggesting that TMD may be easy to maximize when completion quality is poor due to higher levels of noise in the completions.

In Figure 6, we present a qualitative comparison of some of the ablated versions of our method. When partial inputs have high ambiguity, we find that sampling style codes using the mapping network from StyleGAN [14] produces completions with large regions of the shape missing. Unlike our learned style codes, the style codes produced by the mapping network do not explicitly carry any information

about complete shapes, and thus can't help in producing plausible completions. When using the EMD diversity penalty, completions have non-uniform density and poorly respect the partial input. EMD is sensitive to density and is computed on all points in the shape, including the points in the partially observed regions; thus, we find that the EMD diversity penalty tends to undesirably shift local point densities along the shape surface rather than result in changes in geometry. Using a single discriminator as opposed to our multi-scale discriminator results in completions that are not realistic. Due to our discriminator's weak architecture, having a discriminator at only a single resolution is not enough to properly discriminate between real and fake point clouds.

6.6 Failure cases

In Figure 7, we share some completion failures. We observe that the failed completions by our method are usually either due to missing thin structures or some noisy artifacts.

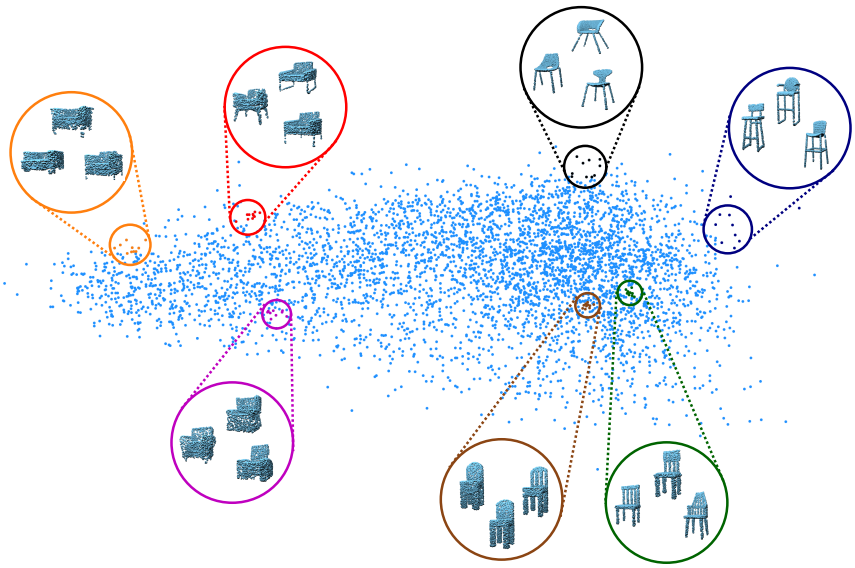

Figure 4: Learned style codes plotted using PCA. We visualize some of the neighborhoods and show that the shapes in the neighborhood share some characteristic/style. It might be concluded that from left to right the chairs are becoming less wide and taller.

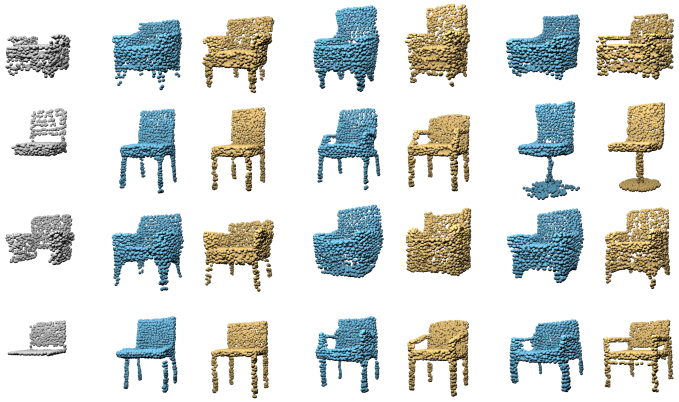

Figure 5: For a partial input (gray), we generate three completions (blue) and each completions nearest neighbor (yellow) from the training set.

Table 1: Ablation on diversity penalty.

Method	MMD ↓	TMD ↑	UHD ↓
EMD	1.82	7.14	6.16
Feat. Diff. (Ours)	1.50	4.36	3.79

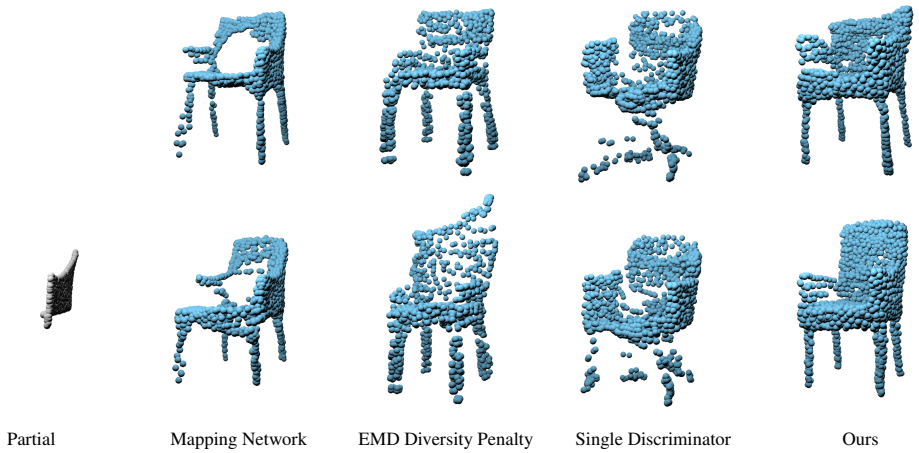

Figure 6: Qualitative comparison of ablated versions of our method.

Figure 7: Failure completion cases with missing/incorrect thin structures (left) and noisy artifacts (right).

7 Limitations

Similar to all other previous works, our method does not consider any external constraints when producing plausible completions. While our method obtains state-of-the-art performance in fidelity to the partial input point clouds and completion diversity, the completions produced by our method are only plausible in the sense that they respect the partial input. This can be problematic when producing completions of objects within a scene as they may violate other scene constraints such as not intersecting with the ground plane or other objects. Taking those constraints into consideration will be interesting future work.

References

[1] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 9621–9630, 2019.

[2] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[3] Haoran Zhou, Yun Cao, Wenqing Chu, Junwei Zhu, Tong Lu, Ying Tai, and Chengjie Wang. Seedformer: Patch seeds based point cloud completion with upsample transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part III*, pages 416–432. Springer, 2022.

[4] He Wang, Zetian Jiang, Li Yi, Kaichun Mo, Hao Su, and Leonidas Guibas. Rethinking Sampling in 3D Point Cloud Generative Adversarial Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop Report*, 2021.

[5] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[6] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[7] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5868–5877, 2017.

[8] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022.

[9] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[10] Rundi Wu, Xuelin Chen, Yixin Zhuang, and Baoquan Chen. Multimodal shape completion via conditional generative adversarial networks. In *The European Conference on Computer Vision (ECCV)*, August 2020.

[11] Xuelin Chen, Baoquan Chen, and Niloy J Mitra. Unpaired point cloud completion on real scans using adversarial training. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[12] Dingdong Yang, Seunghoon Hong, Yunseok Jang, Tianchen Zhao, and Honglak Lee. Diversity-sensitive conditional generative adversarial networks. *arXiv preprint arXiv:1901.09024*, 2019.

[13] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1429–1437, 2019.

[14] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.