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

Abstract

Shape completion aims to recover the full 3D geometry of an object from a partial observation. This problem is inherently multi-modal since there can be many ways to plausibly complete the missing regions of a shape. Such diversity would be indicative of the underlying uncertainty of the shape and could be preferable for downstream tasks such as planning. In this paper, we propose a novel conditional generative adversarial network that can produce many diverse plausible completions of a partially observed point cloud. To enable our network to produce multiple completions for the same partial input, we introduce stochasticity into our network via style modulation. By extracting style codes from complete shapes during training, and learning a distribution over them, our style codes can explicitly carry shape category information leading to better completions. We further introduce diversity penalties and discriminators at multiple scales to prevent conditional mode collapse and to train without the need for multiple ground truth completions for each partial input. Evaluations across several synthetic and real datasets demonstrate that our method achieves significant improvements in respecting the partial observations while obtaining greater diversity in completions.

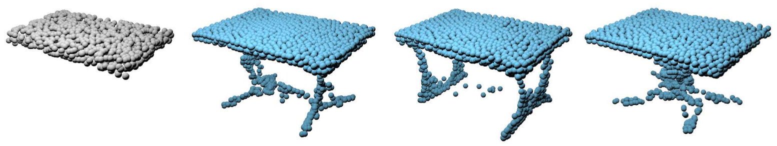


Figure 1: Given a partially observed point cloud (gray), our method is capable of producing many plausible completions (blue) of the missing regions.

1 Introduction

With the rapid advancements in 3D sensing technologies, point clouds have emerged as a popular representation for capturing the geometry of real-world objects and scenes. Point clouds come from sensors such as LiDAR and depth cameras, and can find applications in various domains such as robotics, computer-aided design, augmented reality, and autonomous driving. However, the 3D geometry produced by such sensors is typically sparse, noisy, and incomplete, which hinders their effective utilization in many downstream tasks.

This motivates the task of 3D shape completion from a partially observed point cloud, which has seen significant research in the past few years [1, 2, 3, 4, 5, 6, 7]. Many early point cloud completion works mostly focus on generating a single completion that matches the ground truth in the training set, which does not take into account the potential uncertainty underlying the complete point cloud given the partial view. Ideally, an approach should correctly characterize such uncertainty – generating mostly similar completions when most of the object has been observed, and less similar completions when less of the object has been observed. A good characterization of uncertainty would be informative for downstream tasks such as planning or active perception to aim to reduce such uncertainty.

The task of completing a 3D point cloud with the shape uncertainty in mind is called *multi-modal shape completion* [8, 9], which aims to generate diverse point cloud completions (not to be confused with multi-modality of the input, e.g. text+image). A basic idea is to utilize the diversity coming from generative models such as generative adversarial networks (GAN), where diversity, or the avoidance of *mode collapse* to always generate the same output, has been studied extensively. However, early works [8, 9] often obtain diversity at a cost of poor fidelity to the partial observations due to their simplistic completion formulation that decodes from a single global latent vector. Alternatively, recent diffusion-based methods [10, 11, 12] and auto-regressive methods [13, 14] have shown greater generation capability, but suffer from slow inference time.

In this paper, we propose an approach to balance the diversity of the generated completions and fidelity to the input partial points. Our first novelty comes from the introduction of a *style encoder* to encode the global shape information of complete objects. During training, the ground truth shapes are encoded with this style encoder so that the completions match the ground truth. However, multiple ground truth completions are not available for each partial input; therefore, only using the ground truth style is not enough to obtain diversity in completions. To overcome this, we randomly sample style codes to provide diversity in the other generated completions. Importantly, we discovered that **reducing** the capacity of the style encoder and adding noise to the encoded ground truth shape leads to improved diversity of the generated shapes. We believe this avoids the ground truth from encoding too much content, which may lead the model to overfit to only reconstructing ground truth shapes.

Besides the style encoder, we also take inspiration from recent work SeedFormer [7] to adopt a coarse-to-fine completion architecture. SeedFormer has shown high-quality shape completion capabilities with fast inference time, but only provides deterministic completions. In our work, we make changes to the layers of SeedFormer, making it more suitable for the multi-modal completion task. Additionally, we utilize discriminators at **multiple scales**, which enable training without multiple ground truth completions and significantly improves completion quality. We further introduce a multi-scale diversity penalty that operates in the feature space of our discriminators. This added regularization helps ensure different sampled style codes produce diverse completions.

With these improvements, we build a multi-modal point cloud completion algorithm that outperforms state-of-the-art in both the fidelity to the input partial point clouds as well as the diversity in the generated shapes. Our method is capable of fast inference speeds since it does not rely on any iterative procedure, making it suitable for real-time applications such as in robotics.

Our main contributions can be summarized as follows:

- We design a novel conditional GAN for the task of diverse shape completion that achieves greater diversity along with higher fidelity partial reconstruction and completion quality.
- We introduce a style-based seed generator that produces diverse coarse shape completions via style modulation, where style codes are learned from a distribution of complete shapes.
- We propose a multi-scale discriminator and diversity penalty for training our diverse shape completion framework without access to multiple ground truth completions per partial input.

2 Related work

2.1 3D shape generation

The goal of 3D shape generation is to learn a generative model that can capture the distribution of 3D shapes. In recent years, generative modeling frameworks have been investigated for shape generation using different 3D representations, including voxels, point clouds, meshes, and neural fields.

One of the most popular frameworks for 3D shape generation has been generative adversarial networks (GANs). Most works have explored point cloud-based GANs [15, 16, 17, 18, 19], while recent works have shown that GANs can be trained on neural representations [20, 21, 22, 23]. To avoid the training instability and mode collapse in GANs, variational autoencoders have been used for learning 3D shapes [24, 25, 26], while other works have made use of Normalizing Flows [27, 28, 29]. Recently, diverse shape generation has been demonstrated by learning denoising diffusion models on point clouds [30], latent representations [31, 32], or neural fields [33].

2.2 Point cloud completion

Point cloud completion aims to recover the missing geometry of a shape while preserving the partially observed point cloud. PCN [1] was among the earliest deep learning-based methods that worked directly on point clouds using PointNet [34] for point cloud processing. Since then, other direct point cloud completion methods [35, 36, 4, 37, 6, 38, 7] have improved completion results by using local-to-global feature extractors and by producing completions through hierarchical decoding.

Recently, SeedFormer [7] has achieved state-of-the-art performance in point cloud completion. Their Patch Seeds representation has shown to be more effective than global shape representations due to carrying learned geometric features and explicit shape structure. Furthermore, their transformer-based upsampling layers enable reasoning about spatial relationships and aggregating local information in a coarse-to-fine manner, leading to improved recovery of fine geometric structures.

GAN-based completion networks have also been studied to enable point cloud completion learning in unpaired [39, 40, 41] or unsupervised [42] settings. To enhance completion quality, some works have leveraged adversarial training alongside explicit reconstruction losses [4, 43].

2.3 Multimodal shape completion

Most point cloud completion models are deterministic despite the ill-posed nature of shape completion. To address this, Wu et al. [8] proposed a GAN framework that learns a stochastic generator, conditioned on a partial shape code and noise sample, to generate complete shape codes in the latent space of a pre-trained autoencoder. Arora et al. [9] attempt to mitigate the mode collapse present in [8] by using implicit maximum likelihood estimation. These methods can only represent coarse geometry and struggle to respect the partial input due to decoding from a global shape latent vector.

ShapeFormer [13] and AutoSDF [14] explore auto-regressive approaches for probabilistic shape completion. Both methods propose compact discrete 3D representations for shapes and learn an auto-regressive transformer to model the distribution of object completions on such representation. However, these methods have a costly sequential inference process and rely on voxelization and quantization steps, potentially resulting in a loss of geometric detail.

Zhou et al. [10] propose a conditional denoising diffusion model that directly operates on point clouds to produce diverse shape completions. Alternatively, DiffusionSDF [11] and SDFusion [12] first learn a compact latent representation of neural SDFs and then learn a diffusion model over this latent space. These methods suffer from slow inference times due to the iterative denoising procedure, while [11, 12] have an additional costly dense querying of the neural SDF for extracting a mesh.

2.4 Diversity in GANs

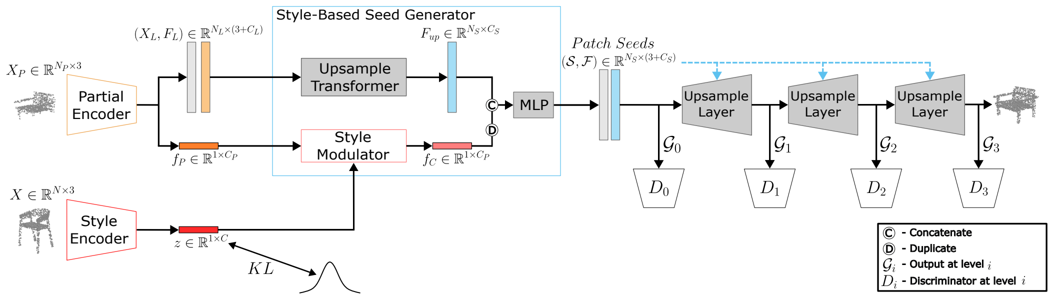
Addressing diversity in GANs has also been extensively studied in the image domain. In particular, style-based generators have shown impressive capability in high-quality diverse image generation where several methods have been proposed for injecting style into generated images via adaptive instance normalization [44, 45, 46, 47] or weight modulation [46, 48]. In the conditional setting, diversity has been achieved by enforcing invertibility between output and latent codes [49] or by regularizing the generator to prevent mode collapse [50, 51, 52] .

3 Method

In this section, we present our conditional GAN framework for diverse point cloud completion. The overall architecture of our method is shown in Figure 2.

Our generator is tasked with producing high-quality shape completions when conditioned on a partial point cloud. To accomplish this, we first introduce a new partial shape encoder which extracts features from the partial input. We then follow SeedFormer [7] and utilize a seed generator to first propose a sparse set of points that represent a coarse completion of a shape given the extracted partial features. The coarse completion is then passed through a series of upsampling layers that utilize transformers with local attention to further refine and upsample the coarse completion into a dense completion.

To obtain diversity in our completions, we propose a style-based seed generator that introduces stochasticity into our completion network at the coarsest level. Our style-based seed generator



Style-Based Seed Generator
$(X_L, F_L) \in \mathbb{R}^{N_L \times (3+C_L)}$
$F_{up} \in \mathbb{R}^{N_S \times C_S}$
Patch Seeds
$(\mathcal{S}, \mathcal{F}) \in \mathbb{R}^{N_S \times (3+C_S)}$
$X_P \in \mathbb{R}^{N_P \times 3}$
Partial Encoder
Upsample Transformer
MLP
Upsample Layer
Upsample Layer
Upsample Layer
Style Modulator
$f_P \in \mathbb{R}^{1 \times C_P}$
$f_C \in \mathbb{R}^{1 \times C_P}$
$\mathcal{G}_0$
$\mathcal{G}_1$
$\mathcal{G}_2$
$\mathcal{G}_3$
D_0
D_1
D_2
D_3
$X \in \mathbb{R}^{N \times 3}$
Style Encoder
$z \in \mathbb{R}^{1 \times C}$
KL
Ⓒ - Concatenate
Ⓓ - Duplicate
$\mathcal{G}_i$ - Output at level i
D_i - Discriminator at level i

Figure 2: Overview of our diverse shape completion framework. A partial encoder is used to extract information from a partial point cloud. During training, a style encoder extracts style codes from complete point clouds, and at inference time style codes are randomly sampled from a normal distribution. Sampled style codes are injected into the partial information to produce diverse Patch Seeds in our style-based seed generator. The generated Patch Seeds are then upsampled into a dense completion through upsampling layers. Furthermore, discriminators and diversity penalties are used at every upsampling layer to train our model.

modulates the partial shape information with style codes before producing a sparse set of candidate points, enabling diverse coarse shape completions that propagate to dense completions through upsampling layers. The style codes used in modulating partial shape information are learned from an object category's complete shapes via a style encoder. Finally, we introduce discriminators and diversity penalties at multiple scales to train our model to produce diverse high-quality completions without having access to multiple ground truth completions that correspond to a partial observation.

3.1 Partial shape encoder

The goal of our partial encoder is to extract shape information in a local-to-global fashion, extracting local information that will be needed in the decoding stage to reconstruct fine geometric structures, while capturing global information needed to make sure a globally coherent shape is generated. An overview of the architecture for our proposed partial encoder is shown in Figure 3a.

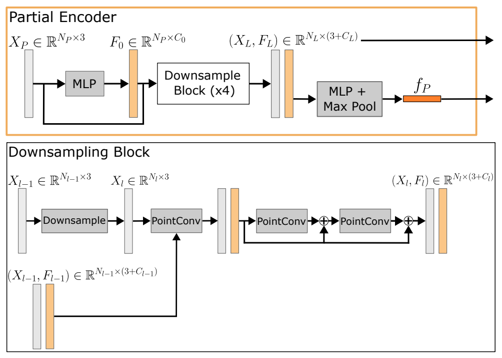
Our partial encoder takes in a partially observed point cloud X_P and first applies a MLP to obtain a set of point-wise features F_0. To extract shape information in a local-to-global fashion, L consecutive downsampling blocks are applied to obtain a set of downsampled points X_L with local features F_L. In each downsampling block, a grid downsampling operation is performed followed by a series of PointConv [53] layers for feature interpolation and aggregation. Following the downsampling blocks, a global representation of the partial shape is additionally extracted by an MLP followed by max pooling, producing partial latent vector f_P.

3.2 Style encoder

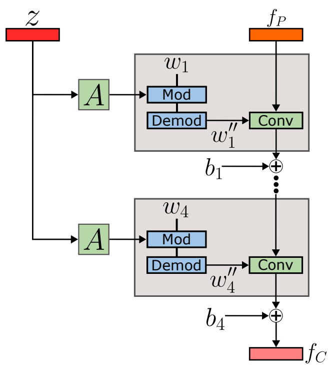
To produce multi-modal completions, we need to introduce randomness into our completion model. One approach is to draw noise from a Gaussian distribution and combine it with partial features during the decoding phase. Another option is to follow StyleGAN [46, 54] and transform noise samples through a non-linear mapping to a latent space $\mathcal{W}$ before injecting them into the partial features. However, these methods rely on implicitly learning a connection between latent samples and shape information. Instead, we propose to learn style codes from an object category's distribution of complete shapes and sample these codes to introduce stochasticity in our completion model. Our style codes explicitly carry information about ground truth shapes, leading to higher quality and more diverse completions.

To do so, we leverage the set of complete shapes we have access to during training. We introduce an encoder E that maps a complete shape $X \in \mathbb{R}^{N \times 3}$ to a global latent vector via a 4-layer MLP followed by max pooling. We opt for a simple architecture as we would like the encoder to capture high level information about the distribution of shapes rather than fine-grained geometric structure.

Instead of assuming we have complete shapes to extract style codes from at inference time, we learn a distribution over style codes that we can sample from. Specifically, we define our style encoder as a learned Gaussian distribution $E_S(z|X) = \mathcal{N}(z|\mu(E(X)), \sigma(E(X)))$ by adding two fully connected layers, μ and σ, to the encoder E to predict the mean and standard deviation of a Gaussian to sample a style code from. Since our aim is to learn style codes that convey information about complete shapes that is useful for generating diverse completions, we train our style encoder with guidance from our


Partial Encoder
$X_P \in \mathbb{R}^{N_P \times 3}$
$F_0 \in \mathbb{R}^{N_P \times C_0}$
$(X_L, F_L) \in \mathbb{R}^{N_L \times (3+C_L)}$
MLP
Downsample Block (x4)
MLP + Max Pool
f_P
Downsampling Block
$X_{l-1} \in \mathbb{R}^{N_{l-1} \times 3}$
$X_l \in \mathbb{R}^{N_l \times 3}$
$(X_l, F_l) \in \mathbb{R}^{N_l \times (3+C_l)}$
Downsample
PointConv
PointConv
PointConv
$(X_{l-1}, F_{l-1}) \in \mathbb{R}^{N_{l-1} \times (3+C_{l-1})}$

(a)



(b)

Figure 3: (a) Architecture of our partial shape encoder. (b) Overview of our style modulator network. For each style-modulated convolution (gray box), w_i and b_i are learned weights and biases of a convolution, w_i'' are the weights after the modulation and demodulation process, and A is a learned Affine transformation.

completion network's losses. To enable sampling during inference, we minimize the KL-divergence between $E_S(z|X)$ and a normal distribution during training. We additionally find that adding noise to our sampled style codes during training leads to higher fidelity and more diverse completions.

3.3 Style-based seed generator

We make use of the Patch Seeds representation proposed in SeedFormer [7], which enables faithfully completing unobserved regions while preserving partially observed structures. Patch Seeds are defined as a set of seed coordinates $\mathcal{S} \in \mathbb{R}^{N_S \times 3}$ and seed features $\mathcal{F} \in \mathbb{R}^{N_S \times C_S}$ produced by a seed generator. In particular, a set of upsampled features $F_{up} \in \mathbb{R}^{N_S \times C_S}$ are generated from the partial local features (X_L, F_L) via an Upsample Transformer [7]. Seed coordinates $\mathcal{S}$ and features $\mathcal{F}$ are then produced from upsampled features F_{up} concatenated with partial latent code f_P via an MLP.

However, the described seed generator is deterministic, prohibiting the ability to generate diverse Patch Seeds that can then produce diverse completions through upsampling layers. We propose to incorporate stochasticity into the seed generator by injecting stochasticity into the partial latent vector f_P. We introduce a style modulator network $M(f_P, z)$, shown in Figure 3b, that injects a style code z into a partial latent vector f_P to produce a styled partial shape latent vector f_C. Following [54], we use weight modulation to inject style into the activation outputs of a network layer, where the demodulated weights w'' used in each convolution layer are computed as:

$$s = A(z), \text{mod: } w'_{ijk} = s_i \cdot w_{ijk}, \text{demod: } w''_{ijk} = w'_{ijk} \Bigg/ \sqrt{\sum_{i,k} w'^2_{ijk} + \epsilon} \tag{1}$$

where A is an Affine transformation, and w is the original convolution weights with i, j, k corresponding to the input channel, output channel, and spatial footprint of the convolution, respectively.

3.4 Coarse-to-fine decoder

Our decoder operates in a coarse-to-fine fashion, which has been shown to be effective in producing shapes with fine geometric structure for the task of point cloud completion [36, 38, 7, 4]. We treat our Patch Seed coordinates $\mathcal{S}$ as our coarsest completion $\mathcal{G}_0$ and progressively upsample by a factor r to produce denser completions $\mathcal{G}_i$ $(i = 1, ..., 3)$. At each upsampling stage i, a set of seed features are first interpolated from the Patch Seeds. Seed features along with the previous layer's points and features are then used by an Upsample Transformer [7] where local self-attention is performed to produce a new set of points and features upsampled by a factor r. We replace the inverse distance weighted averaging used to interpolate seed features from Patch Seeds in SeedFormer with a PointConv interpolation. Since the importance of Patch Seed information may vary across the different layers, we believe a PointConv interpolation is more appropriate than a fixed weighted interpolation as it can learn the appropriate weighting of Patch Seed neighborhoods for each upsampling layer.

Unlike the fully-connected decoders in [8, 9], the coarse-to-fine decoder used in our method can reason about the structures of local regions, allowing us to generate cleaner shape surfaces. A coarse-to-fine design provides us with the additional benefit of having discriminators and diversity penalties at multiple resolutions, which we have found to lead to better completion fidelity and diversity.

3.5 Multi-scale discriminator

During training, we employ adversarial training to assist in learning realistic completions for any partial input and style code combination. We introduce a set of discriminators D_i for $i = \{0, ..., 3\}$, to discriminate against real and fake point clouds at each output level of our generator. Each discriminator follows a PointNet-Mix architecture proposed by Wang et al. [55]. In particular, an MLP first extracts a set of point features from a shape, which are max-pooled and average-pooled to produce f_{max} and f_{avg}, respectively. The features are then concatenated to produce *mix-pooled feature* $f_{mix} = [f_{max}, f_{avg}]$ before being passed through a fully-connected network to produce a final score about whether the point cloud is real or fake.

We also explored more complex discriminators that made use of PointConv or attention mechanisms, but we were unable to successfully train with any such discriminator. This is in line with the findings in [55], suggesting that more powerful discriminators may not guide the learning of point cloud shape generation properly. Thus, we instead use a weaker discriminator architecture but have multiple of them that operate at different scales to discriminate shape information at various feature levels.

For training, we use the WGAN loss [56] with R1 gradient penalty [57] and average over the losses at each output level i. We let $\hat{X}_i = \mathcal{G}_i(X_P, z)$ be the completion output at level i for partial input X_P and sampled style code $z \sim E_S(z|X)$, and let X_i be a real point cloud of same resolution. Then our discriminator loss $\mathcal{L}_D$ and generator loss $\mathcal{L}_G$ are defined as:

$$\mathcal{L}_D = \frac{1}{4}\sum_{i=0}^{3}\Big(\mathbb{E}_{\hat{X}\sim P(\hat{X})}\Big[D_i(\hat{X}_i)\Big] - \mathbb{E}_{X\sim P(X)}[D_i(X_i)] + \frac{\gamma}{2}\mathbb{E}_{X\sim P(X)}\big[\|\nabla D_i(X_i)\|^2\big]\Big) \tag{2}$$

$$\mathcal{L}_G = -\frac{1}{4}\sum_{i=0}^{3}\Big(\mathbb{E}_{\hat{X}\sim P(\hat{X})}[D_i(\hat{X}_i)]\Big) \tag{3}$$

where γ is a hyperparameter ($\gamma = 1$ in our experiments), $P(\hat{X})$ is the distribution of generated shapes, and $P(X)$ is the distribution of real shapes.

3.6 Diversity regularization

Despite introducing stochasticity into the partial latent vector, it is still possible for the network to learn to ignore the style code z, leading to mode collapse to a single completion. To address this, we propose a diversity penalty that operates in the feature space of our discriminator. Our key insight is that for a discriminator to be able to properly discriminate between real and fake point clouds, its extracted features should have learned relevant structural information. Then our assumption is that if two completions are structurally different, the discriminator's global mix-pooled features should be dissimilar as well, which we try to enforce through our diversity penalty.

Specifically, at every training iteration 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. For a single partial input X_P, we produce two different completions $\mathcal{G}_i(X_P, z_1)$ and $\mathcal{G}_i(X_P, z_2)$. We treat our discriminator D_i as a feature extractor and extract the mixed-pooled feature for both completions at every output level i. We denote the mixed-pooled feature corresponding to a completion conditioned on style code z at output level i by $f^z_{mix_i}$, then minimize:

$$\mathcal{L}_{div} = \sum_{i=0}^{3}\frac{1}{\left\|f^{z_1}_{mix_i} - f^{z_2}_{mix_i}\right\|_1} \tag{4}$$

which encourages the generator to produce completions with dissimilar mix-pooled features for different style codes. Rather than directly using the discriminator's mix-pooled feature, we perform pooling only over the set of point features that are not in partially observed regions. This helps avoid penalizing a lack of diversity in the partially observed regions of our completions.

We additionally make use of a partial reconstruction loss at each output level on both completions:

$$\mathcal{L}_{part} = \sum_{z\in\{z_1,z_2\}}\sum_{i=0}^{3} d^{UHD}(X_P,\ \mathcal{G}_i(X_P, z)) \tag{5}$$

where d^{UHD} stands for the unidirectional Hausdorff distance from partial point cloud to completion. Such a loss helps ensure that our completions respect the partial input for any style code z.

To ensure that the completion set covers the ground truth completions in the training set, we choose to always set random complete shape $X_1 = X_{GT}$ and sample $z_1 \sim E_S(z|X_{GT})$, where X_{GT} is the corresponding ground truth completion to the partial input X_P. This allows us to provide supervision at the output of each upsampling layer via Chamfer Distance (CD) for one of our style codes:

$$\mathcal{L}_{comp} = \sum_{i=0}^{3} d^{CD}(X_{GT},\ \mathcal{G}_i(X_P, z_1)) \tag{6}$$

Our full loss that we use in training our generator is then:

$$\mathcal{L} = \lambda_G \mathcal{L}_G + \lambda_{comp}\mathcal{L}_{comp} + \lambda_{part}\mathcal{L}_{part} + \lambda_{div}\mathcal{L}_{div} \tag{7}$$

We set $\lambda_G = 1, \lambda_{comp} = 0.5, \lambda_{part} = 1, \lambda_{div} = 5$, which we found to be good default settings across the datasets used in our experiments.

4 Experiments

In this section, we evaluate our method against a variety of baselines on the task of multimodal shape completion and show superior quantitative and qualitative results across several synthetic and real datasets. We further conduct a series of ablations to justify the design choices of our method.

Implementation Details Our model takes in $N_P = 1024$ points as partial input and produces $N = 2048$ points as a completion. For training the generator, the Adam optimizer is used with an initial learning rate of 1×10^{-4} and the learning rate is linearly decayed every 2 epochs with a decay rate of 0.98. For the discriminator, the Adam optimizer is used with a learning rate of 1×10^{-4}. We train a separate model for each shape category and train each model for 300 epochs with a batch size of 56. All models are trained on two NVIDIA Tesla V100 GPUs and take about 30 hours to train.

Datasets We conduct experiments on several synthetic and real datasets. Following the setup of [8], we evaluate our approach on the Chair, Table, and Airplane categories of the 3D-EPN dataset [58]. Similarly, we also perform experiments on the Chair, Table, and Lamp categories from the PartNet dataset [59]. To evaluate our method on real scanned data, we conduct experiments on the Google Scanned Objects (GSO) dataset [60]. For GSO, we share quantitative and qualitative results on the Shoe, Toys, and Consumer Goods categories. A full description is presented in the supplementary.

Metrics We follow [8] and evaluate with the Minimal Matching Distance (MMD), Total Mutual Difference (TMD), and Unidirectional Hausdorff Distance (UHD) metrics. MMD measures the fidelity of the completion set with respect to the ground truth completions. TMD measures the completion diversity for a partial input shape. UHD measures the completion fidelity with respect to the partial input. We evaluate metrics on $K = 10$ generated completions per partial input. Reported MMD, TMD, and UHD values in our results are multiplied by 10^3, 10^2, and 10^2, respectively.

Table 1: Results on the 3D-EPN dataset. * indicates metric is not reported.

	MMD ↓				TMD ↑				UHD ↓			
Method	Chair	Plane	Table	Avg.	Chair	Plane	Table	Avg.	Chair	Plane	Table	Avg.
SeedFormer [7]	0.45	0.17	0.65	0.42	0.00	0.00	0.00	0.00	1.69	1.27	1.69	1.55
KNN-latent [8]	1.45	0.93	2.25	1.54	2.24	1.13	3.25	2.21	8.94	9.54	12.70	10.39
cGAN [8]	1.61	0.82	2.57	1.67	2.56	2.03	4.49	3.03	8.33	9.59	9.03	8.98
IMLE [9]	*	*	*	*	2.93	**2.31**	4.92	**3.39**	8.51	9.55	8.52	8.86
Ours	**1.16**	**0.59**	**1.45**	**1.07**	**3.26**	1.53	**5.14**	3.31	**4.02**	**3.40**	**4.00**	**3.81**

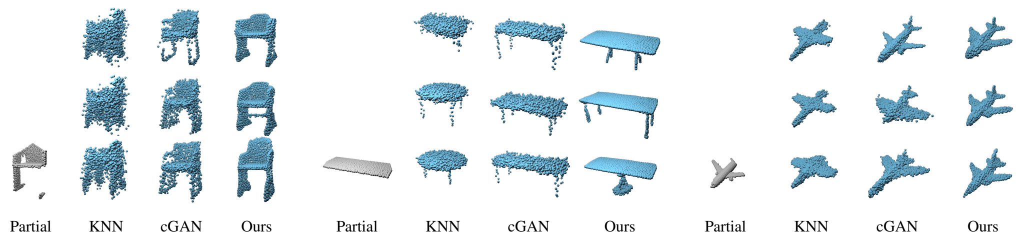


Figure 4: Qualitative comparison of multi-modal completions on the 3D-EPN dataset.

Table 2: Results on the PartNet dataset. * indicates that metric is not reported. † indicates methods that use an alternative computation for MMD and TMD.

	MMD ↓				TMD ↑				UHD ↓			
Method	Chair	Lamp	Table	Avg.	Chair	Lamp	Table	Avg.	Chair	Lamp	Table	Avg.
SeedFormer [7]	0.72	1.35	0.71	0.93	0.00	0.00	0.00	0.00	1.54	1.25	1.48	1.42
KNN-latent [8]	**1.39**	**1.72**	1.30	1.47	2.28	4.18	2.36	2.94	8.58	8.47	7.61	8.22
cGAN [8]	1.52	1.97	1.46	1.65	2.75	3.31	3.30	3.12	6.89	5.72	5.56	6.06
IMLE [9]	*	*	*	*	2.76	5.49	4.45	4.23	6.17	5.58	5.16	5.64
ShapeFormer [10]	*	*	*	**1.32**	*	*	*	3.96	*	*	*	*
Ours	1.50	1.84	**1.15**	1.49	**4.36**	**6.55**	**5.11**	**5.34**	**3.79**	**3.88**	**3.69**	**3.79**
PVD [10] †	**1.27**	**1.03**	1.98	1.43	1.91	1.70	**5.92**	3.18	*	*	*	*
Ours †	1.34	1.55	**1.12**	**1.34**	**5.27**	**7.11**	5.84	**6.07**	*	*	*	*

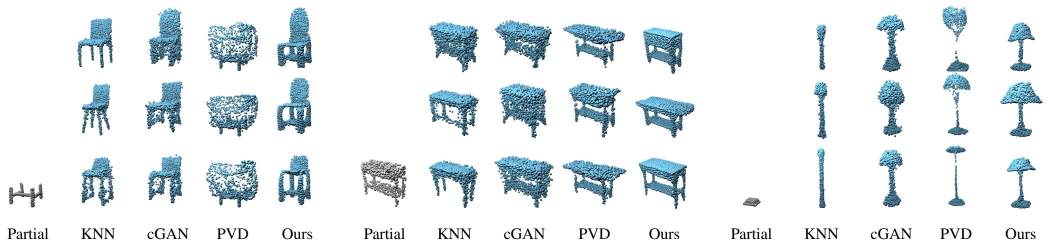


Figure 5: Qualitative results on the PartNet dataset.

Baselines We compare our model against three direct multi-modal shape completion methods: cGAN [8], IMLE [9], and KNN-latent which is a baseline proposed in [8]. We further compare with the diffusion-based method PVD [10] and the auto-regressive method ShapeFormer [13]. We also share quantitative results against the deterministic point cloud completion method SeedFormer [7].

4.1 Results

Results on the 3D-EPN dataset are shown in Table 1. SeedFormer obtains a low UHD implying that their completions respect the partial input well; however, their method produces no diversity as it is deterministic. Our UHD is significantly better than all multi-modal completion baselines, suggesting that we more faithfully respect the partial input. Additionally, our method outperforms others in terms of TMD and MMD, indicating better diversity and completion quality. This is also reflected in the qualitative results shown in Figure 4, where KNN-latent fails to produce plausible completions, while completions from cGAN contain high levels of noise.

In Table 2, we compare against other methods on the PartNet dataset. For a fair comparison with PVD, we also report metrics following their protocol (denoted by †)[10]. In particular, under their protocol TMD is computed on a subsampled set of 1024 points and MMD is computed on the subsampled set concatenated with the original partial input. Once again our method obtains the best diversity (TMD) across all categories, beating out the diffusion-based method PVD and the auto-regressive method ShapeFormer. Our method also achieves significantly lower UHD and shows competitive performance in terms of MMD. Some qualitative results are shown in Figure 5. We find that our method produces cleaner surface geometry and obtains nice diversity in comparison to other methods.

Table 3: Results on Google Scanned Objects dataset. * indicates metric is not reported. † indicates methods that use an alternative computation for MMD and TMD.

	MMD ↓				TMD ↑				UHD ↓			
Method	Shoe	Toys	Goods	Avg.	Shoe	Toys	Goods	Avg.	Shoe	Toys	Goods	Avg.
SeedFormer [7]	0.42	0.67	0.47	0.52	0.00	0.00	0.00	0.00	1.49	1.69	1.55	1.58
cGAN [8]	1.00	2.75	1.79	1.85	1.10	1.87	**1.95**	1.64	5.05	6.61	6.35	6.00
Ours	**0.85**	**1.90**	**0.99**	**1.25**	**1.71**	**2.27**	1.89	**1.96**	**2.88**	**3.84**	**4.68**	**3.80**
PVD [10] †	**0.66**	2.04	1.11	1.27	1.15	2.05	1.44	1.55	*	*	*	*
Ours †	0.90	**1.72**	**1.04**	**1.22**	**2.42**	**2.88**	**2.57**	**2.62**	*	*	*	*

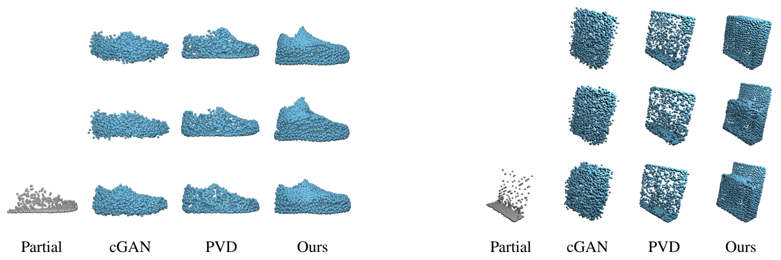


Figure 6: Qualitative comparison of diverse completions on the Google Scanned Objects dataset.

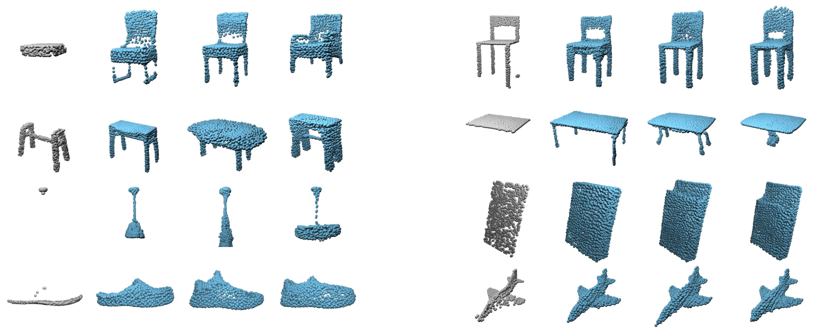


Figure 7: Multi-modal completions (blue) of partial point clouds (gray) produced by our method.

Additionally, we compare our method on real data from the Google Scanned Objects dataset. In Table 3, we show that our method obtains better performance across all the metrics for all three categories. We present a qualitative comparison of completions on objects from the Google Scanned Objects dataset in Figure 6. Completions by cGAN [8] are noisy and lack diversity, while completions from PVD [10] have little diversity and suffer from non-uniform density. Alternatively, our method produces cleaner and more diverse completions with more uniform density.

We further demonstrate the ability of our method to produce diverse high-quality shape completions in Figure 7. Even with varying levels of ambiguity in the partial scans, our method can produce plausible multi-modal completions of objects. In particular, we find that under high levels of ambiguity, such as in the lamp or shoe examples, our method produces more diverse completions. On the other hand, when the object is mostly observed, such as in the plane example, our completions exhibit less variation among them. For more qualitative results, we refer readers to our supplemental.

Finally, we find that our method is capable of inference in near real-time speeds. To produce $K = 10$ completions of a partial input on a NVIDIA V100, our method takes an average of 85 ms while cGAN and KNN-latent require 5 ms. PVD requires 45 seconds which is 500 times slower than us.

Table 4: Ablation on style code dim.

Style Code	MMD $\downarrow$	TMD $\uparrow$	UHD $\downarrow$
512-dim	1.51	3.41	3.98
128-dim	1.54	3.30	4.17
32-dim	1.59	3.89	4.04
16-dim	1.55	3.96	3.97
8-dim	1.51	3.94	3.97
4-dim	1.51	4.00	3.93
8-dim + noise (Ours)	1.50	4.36	3.79

Table 5: Ablation on style code generation.

Method	MMD $\downarrow$	TMD $\uparrow$	UHD $\downarrow$
Gaussian Noise	1.57	3.71	4.59
Mapping Network	1.45	4.03	3.72
Style Encoder (Ours)	1.50	4.36	3.79

Table 6: Ablation on completion network design.

Method	MMD $\downarrow$	TMD $\uparrow$	UHD $\downarrow$
SF	1.45	3.21	3.37
SF + PE	1.56	3.74	3.83
SF + PE + PCI (Ours)	1.50	4.36	3.79

Table 7: Ablation on discriminator architecture.

Method	MMD ↓	TMD ↑	UHD ↓
Single-scale	1.58	2.05	4.27
Multi-scale (Ours)	1.50	4.36	3.79

Table 8: Ablation on loss functions.

	MMD ↓	TMD ↑	UHD ↓
w/o $\mathcal{L}_{comp}$	1.62	4.61	4.58
w/o $\mathcal{L}_{part}$	1.81	5.97	13.03
w/o $\mathcal{L}_{div}$	1.70	0.41	3.57
Ours	1.50	4.36	3.79

4.2 Ablation studies

In Table 4 we examine the dimensionality of our style codes. Our method obtains higher TMD when using smaller style code dimension size. We additionally find that adding a small amount of noise to sampled style codes during training further helps boost TMD while improving UHD. We believe that reducing the style code dimension and adding a small amount of noise helps prevent our style codes from encoding too much information about the ground truth shape, which could lead to overfitting to the ground truth completion. Furthermore, in Table 5, we present results with different choices for style code generation. Our proposed style encoder improves diversity over sampling style codes from a normal distribution or by using the mapping network from StyleGAN [46]. Despite having slightly worse MMD and UHD than StyleGAN's mapping network, we find the quality of completions at test time to be better when training with style codes sampled from our style encoder (see supplementary).

In our method, we made several changes to the completion network in SeedFormer [7]. We replaced the encoder from SeedFormer, which consisted of point transformer [61] and PointNet++ [62] set abstraction layers, with our proposed partial encoder as well as replaced the inverse distance weighted interpolation with PointConv interpolation in the SeedFormer decoder. To justify these changes, we compare the different architectures in our GAN framework in Table 6. We compare the performance of the original SeedFormer encoder and decoder (SF), our proposed partial encoder and SeedFormer decoder (PE + SF), and our full architecture where we replace inverse distance weighted interpolation with PointConv interpolation in the decoder (PE + SF + PCI). Our proposed partial encoder produces an improvement in TMD for slightly worse completion fidelity. Further, we find using PointConv interpolation provides an additional boost in diversity while improving completion fidelity.

The importance of our multi-scale discriminator is shown in Table 7. Using a single discriminator/diversity penalty only at the final output resolution results in a drop in completion quality and diversity when compared with our multi-scale design.

Finally, we demonstrate the necessity of our loss functions in Table 8. Without $\mathcal{L}_{comp}$, our method has to rely on the discriminator alone for encouraging sharp completions in the missing regions. This leads to a drop in completion quality (MMD). Without $\mathcal{L}_{part}$, completions fail to respect the partial input, leading to poor UHD. With the removal of either of these losses, we do observe an increase in TMD; however, this is most likely due to the noise introduced by the worse completion quality. Without $\mathcal{L}_{div}$, we observe TMD drastically decreases towards zero, suggesting no diversity in the completions. This difference suggests how crucial our diversity penalty is for preventing conditional mode collapse. Moreover, we observe that when using all three losses, our method is able to obtain good completion quality, faithfully reconstruct the partial input, and produce diverse completions.

5 Conclusion

In this paper, we present a novel conditional GAN framework that learns a one-to-many mapping between partial point clouds and complete point clouds. To account for the inherent uncertainty present in the task of shape completion, our proposed style encoder and style-based seed generator enable diverse shape completion, with our multi-scale discriminator and diversity regularization preventing mode collapse in the completions. Through extensive experiments on both synthetic and real datasets, we demonstrate that our multi-modal completion algorithm obtains superior performance over current state-of-the-art approaches in both fidelity to the input partial point clouds and completion diversity. Additionally, our method runs in near real-time speed, making it suitable for applications in robotics such as planning or active perception.

While our method is capable of producing diverse completions, it considers only a segmented point cloud on the object. A promising future research direction is to explore how to incorporate additional scene constraints (e.g. ground plane and other obstacles) into our multi-modal completion framework.

Acknowledgments and Disclosure of Funding

This work was supported in part by ONR award N0014-21-1-2052, ONR/NAVSEA contract N00024-10-D-6318/DO#N0002420F8705 (Task 2: Fundamental Research in Autonomous Subsea Robotic Manipulation), NSF grant #1751412, and DARPA contract N66001-19-2-4035.

References

[1] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 international conference on 3D vision (3DV)*, pages 728–737. IEEE, 2018.

[2] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 383–392, 2019.

[3] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11596–11603, 2020.

[4] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 790–799, 2020.

[5] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX*, pages 365–381. Springer, 2020.

[6] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12498–12507, 2021.

[7] 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.

[8] 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.

[9] Himanshu Arora, Saurabh Mishra, Shichong Peng, Ke Li, and Ali Mahdavi-Amiri. Multimodal shape completion via implicit maximum likelihood estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2958–2967, June 2022.

[10] 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, October 2021.

[11] Gene Chou, Yuval Bahat, and Felix Heide. Diffusion-sdf: Conditional generative modeling of signed distance functions. In *The IEEE International Conference on Computer Vision (ICCV)*, 2023.

[12] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tuyakov, Alex Schwing, and Liangyan Gui. SDFusion: Multimodal 3d shape completion, reconstruction, and generation. *arXiv*, 2022.

[13] Xingguang Yan, Liqiang Lin, Niloy J. Mitra, Dani Lischinski, Danny Cohen-Or, and Hui Huang. Shape-former: Transformer-based shape completion via sparse representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[14] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. AutoSDF: Shape priors for 3d completion, reconstruction and generation. In *CVPR*, 2022.

[15] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018.

[16] Chun-Liang Li, Manzil Zaheer, Yang Zhang, Barnabas Poczos, and Ruslan Salakhutdinov. Point cloud gan. *arXiv preprint arXiv:1810.05795*, 2018.

[17] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3859–3868, 2019.

[18] Ruihui Li, Xianzhi Li, Ka-Hei Hui, and Chi-Wing Fu. Sp-gan: Sphere-guided 3d shape generation and manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–12, 2021.

[19] Yingzhi Tang, Yue Qian, Qijian Zhang, Yiming Zeng, Junhui Hou, and Xuefei Zhe. Warpinggan: Warping multiple uniform priors for adversarial 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6397–6405, 2022.

[20] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019.

[21] Marian Kleineberg, Matthias Fey, and Frank Weichert. Adversarial generation of continuous implicit shape representations. *arXiv preprint arXiv:2002.00349*, 2020.

[22] Moritz Ibing, Isaak Lim, and Leif Kobbelt. 3d shape generation with grid-based implicit functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13559–13568, 2021.

[23] X Zheng, Yang Liu, P Wang, and Xin Tong. Sdf-stylegan: Implicit sdf-based stylegan for 3d shape generation. In *Computer Graphics Forum*, volume 41, pages 52–63. Wiley Online Library, 2022.

[24] Jinwoo Kim, Jaehoon Yoo, Juho Lee, and Seunghoon Hong. Setvae: Learning hierarchical composition for generative modeling of set-structured data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15059–15068, 2021.

[25] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[26] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3d point cloud processing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–118, 2018.

[27] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4541–4550, 2019.

[28] Roman Klokov, Edmond Boyer, and Jakob Verbeek. Discrete point flow networks for efficient point cloud generation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 694–710. Springer, 2020.

[29] Janis Postels, Mengya Liu, Riccardo Spezialetti, Luc Van Gool, and Federico Tombari. Go with the flows: Mixtures of normalizing flows for point cloud generation and reconstruction. In *2021 International Conference on 3D Vision (3DV)*, pages 1249–1258. IEEE, 2021.

[30] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021.

[31] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. *arXiv preprint arXiv:2210.06978*, 2022.

[32] Gimin Nam, Mariem Khlifi, Andrew Rodriguez, Alberto Tono, Linqi Zhou, and Paul Guerrero. 3d-ldm: Neural implicit 3d shape generation with latent diffusion models. *arXiv preprint arXiv:2212.00842*, 2022.

[33] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. *arXiv preprint arXiv:2211.16677*, 2022.

[34] 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.

[35] Xin Wen, Tianyang Li, Zhizhong Han, and Yu-Shen Liu. Point cloud completion by skip-attention network with hierarchical folding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1939–1948, 2020.

[36] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. Pf-net: Point fractal network for 3d point cloud completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7662–7670, 2020.

[37] Xin Wen, Peng Xiang, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Pmp-net: Point cloud completion by learning multi-step point moving paths. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7443–7452, 2021.

[38] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5499–5509, 2021.

[39] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13080–13089, 2021.

[40] Zhen Cao, Wenxiao Zhang, Xin Wen, Zhen Dong, Yu-shen Liu, Xiongwu Xiao, and Bisheng Yang. Ktnet: Knowledge transfer for unpaired 3d shape completion. *arXiv e-prints*, pages arXiv–2111, 2021.

[41] 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.

[42] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1768–1777, 2021.

[43] Chulin Xie, Chuxin Wang, Bo Zhang, Hao Yang, Dong Chen, and Fang Wen. Style-based point generator with adversarial rendering for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4619–4628, 2021.

[44] Ting Chen, Mario Lucic, Neil Houlsby, and Sylvain Gelly. On self modulation for generative adversarial networks. *arXiv preprint arXiv:1810.01365*, 2018.

[45] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019.

[46] 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.

[47] Ari Heljakka, Yuxin Hou, Juho Kannala, and Arno Solin. Deep automodulators. *Advances in Neural Information Processing Systems*, 33:13702–13713, 2020.

[48] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. *arXiv preprint arXiv:2103.10428*, 2021.

[49] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *Advances in neural information processing systems*, 30, 2017.

[50] Augustus Odena, Jacob Buckman, Catherine Olsson, Tom Brown, Christopher Olah, Colin Raffel, and Ian Goodfellow. Is generator conditioning causally related to gan performance? In *International conference on machine learning*, pages 3849–3858. PMLR, 2018.

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

[52] 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.

[53] 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.

[54] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[55] 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.

[56] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[57] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.

[58] 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.

[59] 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.

[60] 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.

[61] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021.

[62] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[63] 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.

[64] 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.

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

6 Overview

- Architecture (Section 7): we provide a detailed description of our architecture
- Datasets (Section 8): we provide a detailed description of datasets used in our experiments
- Metrics (Section 9): we formally define our evaluation metrics
- Baseline diversity penalty (Section 10): we discuss an alternative diversity penalty which we treat as a baseline in our ablations
- More results (Section 11): we share more qualitative results of our method
- Limitations (Section 12): we discuss some of the limitations of our method

7 Architecture

7.1 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 [53] 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.

7.2 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 [34] 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:

$$z = \mu + \sigma \cdot \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, I) \tag{8}$$

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:

$$\mathcal{L}_{KL} = \lambda_{KL}\, D_{KL}(E_S(z|X)\, ||\, \mathcal{N}(0, I)) \tag{9}$$

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

7.3 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.

7.4 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 [7] 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 [7] work for a full description of the Upsample Transformer.

7.5 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.

7.6 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 [55]. 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.

8 Datasets

We conduct experiments on data from the ShapeNet [63], PartNet [59], 3D-EPN [58], Google Scanned Objects [60], and ScanNet [64] 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.

8.1 3D-EPN

For the 3D-EPN dataset [58], we evaluate on the Chair, Table, and Airplane categories and follow the train/test splits used in [8]. 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 [63]. 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.

8.2 PartNet

For the PartNet dataset [59], we evaluate on the Chair, Table, and Lamp categories and once again follow the train/test splits used in [8]. 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.

8.3 Google Scanned Objects

For the Google Scanned Objects dataset [60], 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.

8.4 ScanNet

For the ScanNet dataset [64], we use the preprocessed data provided by [41]. 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.

9 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:

$$d^{CD}(P,Q) = \frac{1}{|P|}\sum_{x\in P}\min_{y\in Q}\|x-y\|_2^2 + \frac{1}{|Q|}\sum_{y\in Q}\min_{x\in P}\|x-y\|_2^2 \tag{10}$$

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 $\mathcal{C} = \{c_{ij}\}$.

9.1 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:

$$MMD = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\left(\min_{c\in\mathcal{C}} d^{CD}(t,c)\right) \tag{11}$$

9.2 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 $\mathcal{P}$:

$$TMD = \frac{1}{|\mathcal{P}|} \sum_{i=1}^{|\mathcal{P}|} \left(\sum_{j=1}^{K} \frac{1}{K-1} \sum_{1 \leq l \leq K, l \neq j} d^{CD}(c_{ij}, c_{il}) \right) \tag{12}$$

9.3 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:

$$d^{UHD}(P, Q) = \max_{x \in P} \min_{y \in Q} \|x - y\|_2 \tag{13}$$

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 $j = 1, ..., K$:

$$UHD = \frac{1}{|\mathcal{P}|} \sum_{p_i \in \mathcal{P}} \left(\frac{1}{K} \sum_{j=1}^{K} d^{UHD}(p_i, c_{ij}) \right) \tag{14}$$

10 Baseline diversity penalty

We discuss an alternative diversity penalty which we treat as a baseline in our ablation in Table 9. 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 [51, 52], 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:

$$d^{EMD}(P, Q) = \min_{\phi: P \to Q} \frac{1}{|P|} \sum_{x \in P} \|x - \phi(x)\|_2 \tag{15}$$

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:

$$\mathcal{L}_{div} = \sum_{i=0}^{3} \frac{1}{d^{EMD}(\mathcal{G}_i(X_P, z_1),\ \mathcal{G}_i(X_P, z_2))} \tag{16}$$

Note, by minimizing Equation 16 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.

11 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.

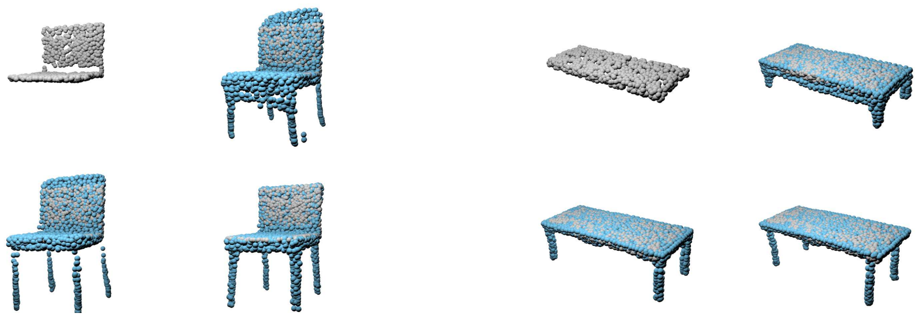


Figure 8: Visualization of partial shapes (gray) overlaid on completions from our method (blue).

11.1 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 8. 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.

11.2 More completions

We share more multi-modal completions produced by our method in Figure 9. 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 10, 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.

11.3 Visualizing style codes

In Figure 11, 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.

11.4 Nearest neighbors of completions

In Figure 12, 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.

11.5 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 9, 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

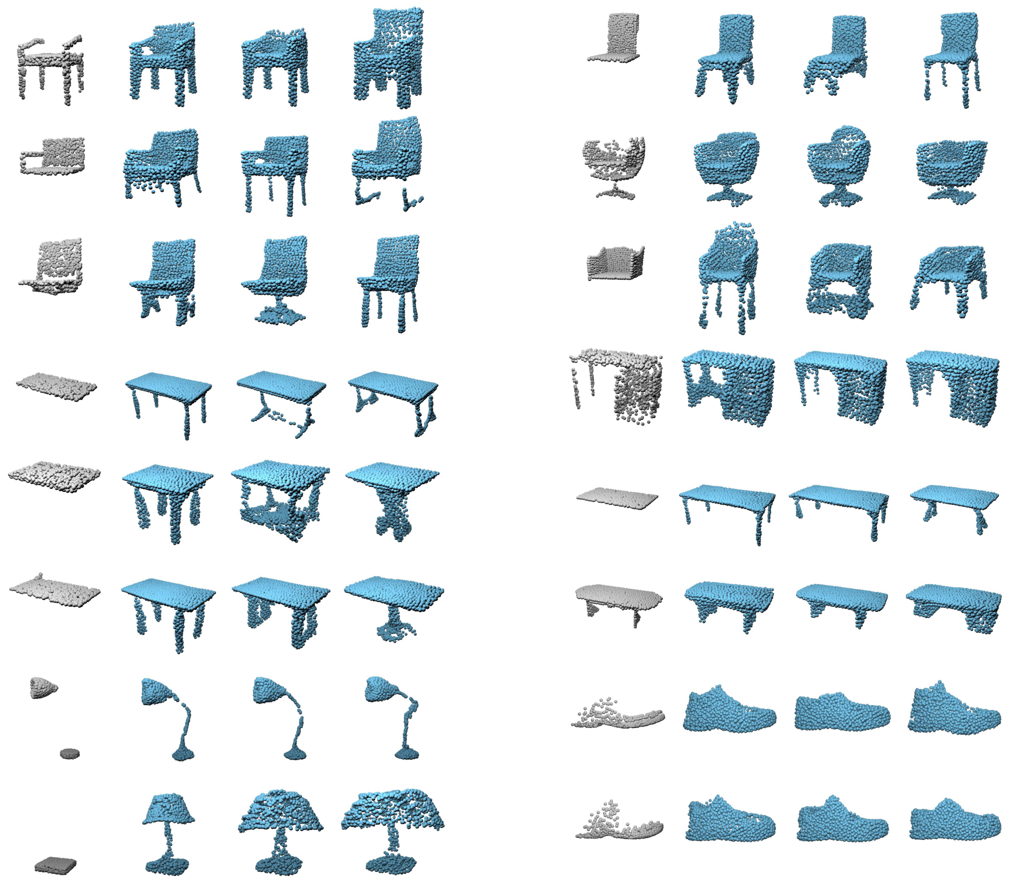


Figure 9: Example multi-modal completions (blue) of partial point clouds (gray) across several different categories from the PartNet, 3D-EPN, and Google Scanned Objects datasets.

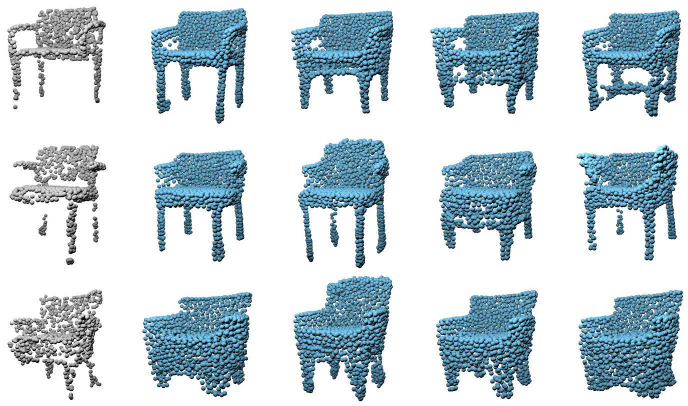


Figure 10: Qualitative results on real scanned chairs from ScanNet.

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 13, 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 [46] 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.

11.6 Failure cases

In Figure 14, 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.

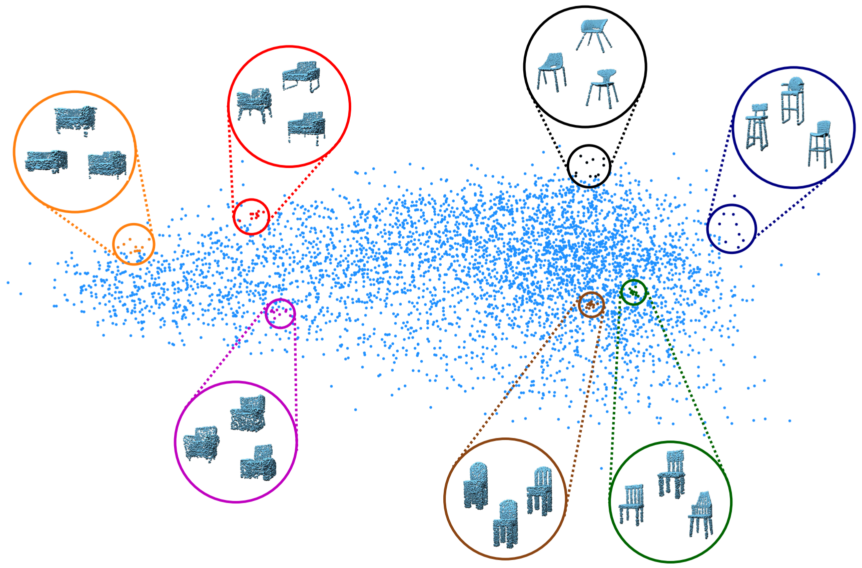


Figure 11: Learned style codes plotted using PCA. We visualize some of the neighborhoods and show that the shapes in the neighborhood share some characteristic/style. It might be concluded that from left to right the chairs are becoming less wide and taller.

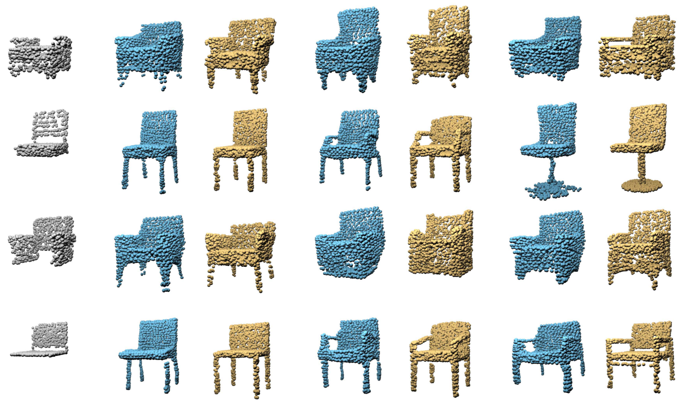


Figure 12: For a partial input (gray), we generate three completions (blue) and each completions nearest neighbor (yellow) from the training set.

Table 9: Ablation on diversity penalty.

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

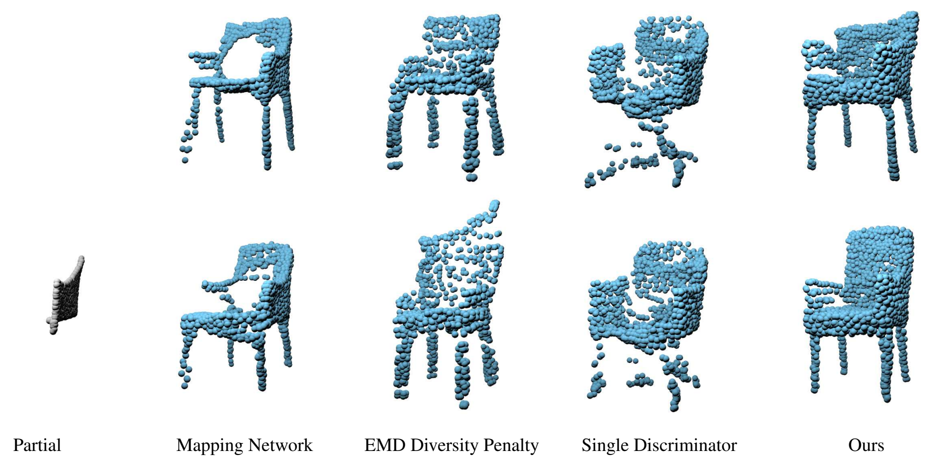


Figure 13: Qualitative comparison of ablated versions of our method.

Figure 14: Failure completion cases with missing/incorrect thin structures (left) and noisy artifacts (right).

12 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.