Unsupervised 3D Shape Completion through GAN Inversion

Junzhe Zhang[1,3] Xinyi Chen[2,3] Zhongang Cai[3,4] Liang Pan[1]
Haiyu Zhao[3,4] Shuai Yi[3,4] Chai Kiat Yeo[2] Bo Dai[1] Chen Change Loy[1]
[1]S-Lab, Nanyang Technological University [2]Nanyang Technological University
[3]SenseTime Research [4]Shanghai AI Laboratory
{junzhe001,xchen032}@e.ntu.edu.sg,{liang.pan,asckyeo,bo.dai,ccloy}@ntu.edu.sg
{caizhongang,zhaohaiyu,yishuai}@sensetime.com
https://junzhezhang.github.io/projects/ShapeInversion/

Abstract

Most 3D shape completion approaches rely heavily on partial-complete shape pairs and learn in a fully supervised manner. Despite their impressive performances on in-domain data, when generalizing to partial shapes in other forms or real-world partial scans, they often obtain unsatisfactory results due to domain gaps. In contrast to previous fully supervised approaches, in this paper we present ShapeInversion, which introduces Generative Adversarial Network (GAN) inversion to shape completion for the first time. ShapeInversion uses a GAN pre-trained on complete shapes by searching for a latent code that gives a complete shape that best reconstructs the given partial input. In this way, ShapeInversion no longer needs paired training data, and is capable of incorporating the rich prior captured in a well-trained generative model. On the ShapeNet benchmark, the proposed ShapeInversion outperforms the SOTA unsupervised method, and is comparable with supervised methods that are learned using paired data. It also demonstrates remarkable generalization ability, giving robust results for real-world scans and partial inputs of various forms and incompleteness levels. Importantly, ShapeInversion naturally enables a series of additional abilities thanks to the involvement of a pre-trained GAN, such as producing multiple valid complete shapes for an ambiguous partial input, as well as shape manipulation and interpolation.

1. Introduction

3D shape completion estimates the complete geometry from a partial shape in the form of a partial point cloud, and is important to many downstream applications such as robotics navigation [11, 24] and scene understanding [9, 14]. Most works [28, 21, 15, 30, 32] for shape completion are trained in a fully supervised manner with paired partial-complete data. While they obtain promising results on in-domain data, it is challenging for these methods to generalize to out-of-domain data, which are real-world scans or data with different partial forms, as shown in Fig. 1 (a)-(d).

We take an unsupervised approach in this study. Inspired by the success of GAN inversion in 2D tasks such as image restoration and editing, we propose to apply GAN inversion to 3D shape completion for the first time, which we refer to as **ShapeInversion**. Specifically, given a partial input, ShapeInversion looks for a latent code in the GAN's latent space that gives a complete shape that best reconstructs the input. By incorporating prior knowledge stored in the pre-trained GAN, no assumptions on the input partial forms are made, thus ShapeInversion generalizes well to inputs of various partial forms and real-world scans. Moreover, the involvement of GAN in ShapeInversion brings several side-benefits, including giving multiple reasonable complete shapes for some partial input, as well as shape jittering and shape manipulation.

While ShapeInversion shares some similarity with GAN inversion methods for 2D images, the former possesses several intrinsic challenges due to the nature of 3D data: (1) Unlike 2D images that follow a grid-like structure, where the positions of pixels are well defined, point clouds of different 3D shapes are highly unstructured. Often, GANs trained on 3D shapes would generate point clouds with significant non-uniformity, *i.e.*, points are unevenly distributed over the shape surface. Such non-uniformity may lead to shapes with undesired holes, undermining the completeness of our predictions. (2) The unordered nature of point clouds makes the completion task significantly different from 2D image inpainting. In 2D image inpainting, one can easily measure the reconstruction consistency between the visible regions of partial input and predicted output given the lattice-aligned pixel correspondences. Such comparison is challenging in 3D shape completion since the corresponding regions of two 3D shapes may reside at different locations in the 3D space. Without accurate point correspon-

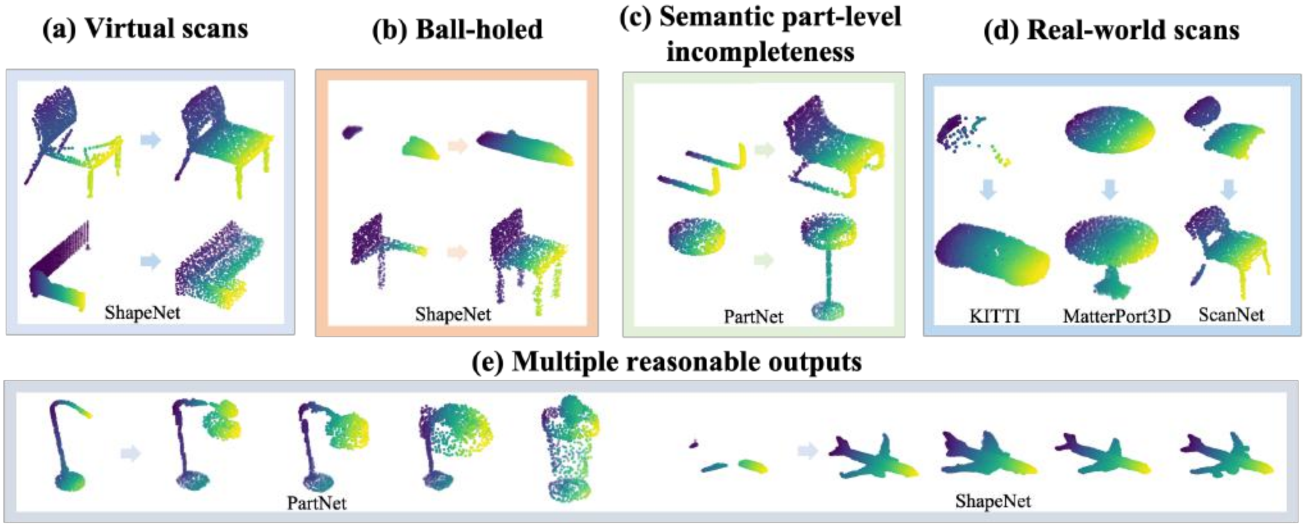

Figure 1. ShapeInversion incorporates the prior captured by a well-trained GAN. It shows exceptional generalization ability for shape completion: is invariant to partial form changes, *i.e.*, **(a)** virtual scans, generated by back-projecting 2.5D depth images into 3D, **(b)** ball-holed partial shapes, generated by removing points within a random ball from a complete shape (PF-Net [15]), **(c)** semantic part-level incompleteness, generated by randomly removing some semantic parts (PartNet [23]); and generalizes well to **(d)** real-world scans. Moreover, it can give **(e)** multiple valid outputs when there is ambiguity in the partial shape

dences, GAN inversion would suffer from poor reconstruction and in turn jeopardize the shape completion task.

We present two new components to address these unique challenges. First, to improve the uniformity of estimated point clouds, we introduce a simple and effective uniform loss, **PatchVariance**. The loss samples small patches, to ensure the planar assumption, over the object surface, and penalize the variance of average distances between the patch centers and their respective nearest neighbors. Unlike existing methods [18, 35] that are typically conducted at the patch level, ours is a soft regularizer that enhances uniformity at the object level on-the-fly while training GANs. As a result, we achieve improved uniformity across all categories, ranging from bulk to fine structures while preserving the shape plausibility and variety.

Second, we devise an effective masking mechanism, **k-Mask**, to estimate the point correspondences between the partial input and predicted shape. To mitigate the ambiguous correspondences caused by the unordered nature of point cloud, our method lets each point in the partial input look for its k-nearest neighbors from the predicted shape. The indices of all these k-nearest neighbors define the mask of the visible regions, from which we can compute for reconstruction loss. Our method is dynamic, thus performing better than baseline approaches that use predefined voxels or distance thresholds. It shows high robustness even when the semantics parts between the partial input and the predicted shape are not within a close vicinity in the space.

ShapeInversion demonstrates compelling performance for shape completion in different scenarios. First, on a common benchmark derived from ShapeNet, it outperforms the SOTA unsupervised method pcl2pcl [7] by a significant margin, and is comparable to various supervised methods. Second, our method shows considerable generalization ability and robustness when it comes to real-world scans or variation in partial forms and incompleteness levels, whereas supervised methods exhibit significant performance drops due to domain mismatches. Third, given more extreme incompleteness that causes ambiguity, our method is able to provide multiple valid complete shapes, all of which remain faithful to the visible parts presented in the partial input.

2. Related Work

3D Shape Completion. 3D shape completion has played an important role for robotics [11, 24] and perception [9, 14]. Since the pioneering work PCN [36], point cloud-based shape completion has seen significant development compared to other representation forms like meshes and voxel grids, due to its flexibility and popularity as a raw data format. Most existing approaches are trained in a fully-supervised manner with partial shapes of a particular form [10, 15, 7, 23, 32, 37], and paired complete shapes. Owing to the coarse-to-fine strategy [28, 21, 15, 30, 34], they achieve impressive results on in-domain data, but may fail to sufficiently generalize to real-world scans or partial shapes in other forms. Recently, pcl2pcl [7] proposes an unsupervised method with unpaired data, *e.g.*, complete shapes obtained from 3D models and real-world partial scans. It trains two separate auto-encoders, for reconstructing complete shapes and partial ones respectively, and learns a mapping from the latent space of partial shapes to that of the complete ones. In view of ambiguity at high incompleteness levels, its follow-up work [33] is able to output multiple plausible complete shapes, conditioned by an additional latent vector drawn from Gaussian distribution. Our approach also lies in the unsupervised regime, and can also give multiple reasonable complete shapes thanks to the involvement of a pre-trained GAN. Moreover, we achieve more faithful results, particularly for real scans.

GAN Inversion. State-of-the-art GANs, *e.g.*, BigGAN [4] and StyleGAN [16], are typically trained on a large number of images and capture rich knowledge of images including low-level statistics, image semantics, and high-level concepts. GAN inversion uses a well-trained GAN as effective prior to reconstruct images with high-fidelity. This appealing nature of GAN prior has been extensively exploited on various image restoration and manipulation tasks [3, 2, 25, 13]. In general, the method aims to find a latent vector that best reconstructs the given image with a pre-trained GAN. Typically, the latent vector can be optimized based on gradient descent [22, 20], or projected by an extra encoder from the image space [39, 17]. Moreover, the introduced encoder can serve as a better initialization prior to gradient descent [2]. Zhu *et al.* [38] learn a domain-guided encoder, which is used to regularize the latent vector optimization for semantically meaningful editing. While mainstream approaches fix the parameters of the generator during inversion, recent approaches chose to perturb [3] or fine-tune [25] the generator when updating the latent vector to address the gap between the approximated manifold and the real one. Our approach is the first to apply GAN inversion to shape completion. Unlike image-based tasks where the degradation transform is typically straightforward, transforming a complete shape into a partial one in the 3D space is ill-posed.

3. Method

A GAN that is well-trained on 3D shapes of a particular category, *e.g.*, chairs or cars, captures rich shape geometries and semantics of this distribution. In this study, we wish to incorporate a well-trained GAN as an effective prior for shape completion, in particular, to handle partial shapes of a wide range of varieties and to generalize to unseen shapes. The GAN prior can be exploited through GAN inversion. Despite its notable success in various image restoration and manipulation tasks, it has not been explored for shape completion.

Here, we formally introduce the use of GAN inversion in our task. After a generator G with parameters θ is trained on 3D shapes in the form of point clouds, it can generate a shape $\mathbf{x}_c \in \mathbb{R}^{m\times 3}$ from a latent vector $\mathbf{z} \in \mathbb{R}^d$. GAN inversion aims to find the latent vector that best reconstructs a given shape $\mathbf{x}_{in}$ using G:

$$\mathbf{z}^* = \underset{\mathbf{z}\in\mathbb{R}^d}{\arg\min}\,\mathcal{L}(G(\mathbf{z};\boldsymbol{\theta}), \mathbf{x}_{in}), \quad \mathbf{x}_c^* = G(\mathbf{z}^*;\boldsymbol{\theta}) \tag{1}$$

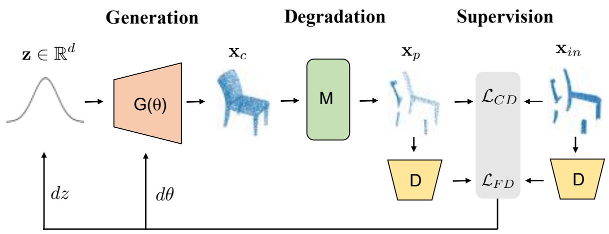

Figure 2. GAN inversion for shape completion. A latent vector $\mathbf{z}$ is used by the pre-trained generator G to reconstruct a complete shape $\mathbf{x}_c$. The degradation function M (Sec. 3.2) then transforms $\mathbf{x}_c$ into a partial shape $\mathbf{x}_p$. The supervision signal includes the Chamfer Distance and the Feature Distance (Sec. 3.3) between $\mathbf{x}_p$ and the input partial shape $\mathbf{x}_{in}$. ShapeInversion looks for a latent vector $\mathbf{z}$ and finetunes the parameters θ of G that best reconstruct the complete shape corresponding to $\mathbf{x}_{in}$ via gradient descent

While mainstream approaches usually fix the generator during inversion, we follow the very recent approaches [25, 3] to fine-tune the generator while updating the latent vector on-the-fly, which is shown to improve the results of GAN inversion. Thus, the formulation becomes:

$$\boldsymbol{\theta}^*, \mathbf{z}^* = \underset{\mathbf{z},\boldsymbol{\theta}}{\arg\min}\,\mathcal{L}(G(\mathbf{z};\boldsymbol{\theta}), \mathbf{x}_{in}) \tag{2}$$

The inversion process starts with an initialization stage, in which hundreds of latent vectors are sampled randomly, and the z with the smallest $\mathcal{L}$ value is selected as the initial value for fine-tuning. Then both z and θ are updated via gradient descent according to Eq. (2). In the scenario of shape completion, we aim to infer a complete shape $\mathbf{x}_c$ from a given partial shape $\mathbf{x}_{in}$, where the distance is computed at the observation space, *i.e.*, we would need to transform a complete shape into a partial form via a degradation function M, as shown in Eq. (3). Thus, it is essential for M to provide precise point correspondences for the sake of an accurate reconstruction loss. The inversion stage is shown in Fig. 2.

$$\mathbf{z}^* = \underset{\mathbf{z}\in\mathbb{R}^d}{\arg\min}\,\mathcal{L}(M(G(\mathbf{z};\boldsymbol{\theta})), \mathbf{x}_{in}) \tag{3}$$

3.1. Enhancing Point Cloud Uniformity

Compared to images where the generated pixels are arranged in a regular lattice, 3D shapes are represented by points in a continuous 3D space without a common structure. As a result, 3D GANs often generate point clouds with significant non-uniformity, where the points are often unevenly distributed over the shape surface. Such non-uniformity is detrimental to shape completion given the number of points in each point cloud is fixed (typically 2048 for existing GANs): point concentration in one region inevitably leads to sparsity or even holes in other regions.

tree-GAN as a Case Study. The latest state-of-the-art point cloud generation method, tree-GAN [27], employs

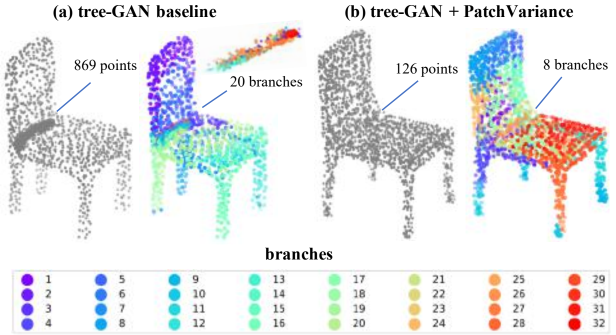

Figure 3. Visualization of uniformity with the use of PatchVariance. Darker regions in the grayscale image have a higher density of points. tree-GAN uses a tree structure to generate 3D shapes. We group points into *branches* by their parent nodes: branches that are more distantly related are further apart on the color spectrum. For the tree-GAN baseline, nearly half of the points and 20 out of the 32 branches cluster around the joint of different parts, with fewer branches to cover the rest of the shape surface

a tree-structured graph convolution network (TreeGCN) as the generator, where the information passes from the ancestor nodes instead of the neighbor nodes. As branching occurs between every two layers in TreeGCN, the child nodes sharing the same parent node would be more geometrically related to each other. Although it outperforms previous approaches [1, 29] in terms of fidelity and coverage, the non-uniformity issue remains unsolved, as illustrated in Fig. 3 (a). For a clearer visualization, we colorize the points based on their relative relationships on TreeGCN. It shows that points with distant relationships might clutter in the 3D space. Without a proper regularization, points of different branches would tend to form a Gaussian-like distribution, such that more points are gathered around the geometric center of an object or the joints between different semantic parts, resulting in highly non-uniform shapes.

The non-uniformity of shapes' point clouds is a long-standing problem. Studies in point cloud upsampling [18, 35] propose some forms of uniformity losses, such as *repulsion loss*, on point cloud patches. Besides, MSN [21] proposes *expansion penalty* to reduce overlapping of the surface elements. However, these methods regularize each part of the shape separately, without enforcing a consensus across all parts to achieve an overall uniformity. In view of their weaknesses, we propose a new uniform loss, **PatchVariance**, to regularize the uniformity of the entire shape during tree-GAN training, in addition to its adversarial loss.

$$\mathcal{L}_{patch} = Var(\{\rho_j\}_{j=1}^{n}), \quad \rho_j = \frac{1}{k}\Sigma_{i=1}^{k} dist_{ij}^2 \qquad (4)$$

Specifically, we randomly sample n seed positions over the object surface via Farthest Point Sampling (FPS), and then form small patches by including the k-nearest neighbors for each seed. Regardless of fine or bulk structure, these small patches shall scatter similarly. Thus we compute the average distance between each seed and its k-nearest neighbors and penalize the variance of all patches' average distances, as shown in Eq. (4).

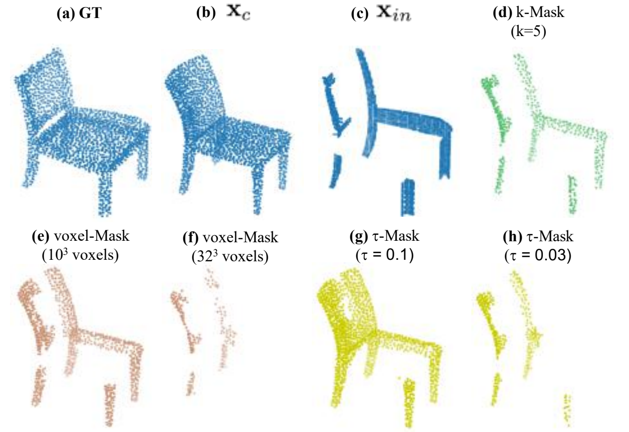

Figure 4. Better degradation function produces partial shapes x_p that are more similar to the input shape x_{in}. Our proposed k-Mask provides accurate point correspondence between x_{in} and the generated complete shape x_c. In contrast, voxel-mask and τ-mask are sensitive to the hyperparameters, *i.e.*, the voxel size or the distance threshold respectively: large values result in noisy degradation, *e.g.*, the chair's back has excessive points in **(e)** and **(g)**; small values lead to missing parts when correpsonding semantic parts reside differently in the 3D space, *e.g.*, the chair's legs are falsely masked off in **(f)** and **(h)**. Note that $\mathbf{x}_{in}$ has 2048 points in ShapeNet benchmark, while k-Mask is shown to be robust to partial shapes of any density

As illustrated in Fig. 3 (b), PatchVariance significantly improves the uniformity of the generated shape. More evaluation and comparison with other uniform losses is covered in the ablation study. Note that the proposed uniform loss is generic: it directly works on the generated shape and is invariant to the GAN architecture. Cross-validation of PatchVariance on r-GAN [1] (with MLP-based generator) can be found in the Supplementary Materials.

3.2. Degradation in the 3D Space

For shape completion, we define a degradation transform M to best approximate the transformation from a generated complete shape $\mathbf{x}_c = G(z)$ to a partial shape $\mathbf{x}_p$, such that the corresponding regions between $\mathbf{x}_c$ and the given partial shape $\mathbf{x}_{in}$ can be precisely compared. We find that defining such a degradation function is ill-posed due to the unique unstructured nature of point clouds.

One may intuitively relate it to the image inpainting task, where a 2D binary mask m is typically provided to degrade a complete image to the observation space through element-wise product: $x_{masked} = x \odot m$, given that the pixel correspondences between any image pair are consistent with the pixel locations. In contrast, corresponding regions of two 3D structures may reside at different locations in the 3D space, while directly voxelizing $\mathbf{x}_{in}$ to form a 3D tensor indicating the voxel occupancy would inevitably lead to information loss. More importantly, as it is likely that $\mathbf{x}_c$ is quite different from $\mathbf{x}_{in}$ especially in the early GAN inversion stage, the corresponding semantic parts do not fall in the same voxels, hence leading to problematic degradation. See Fig. 4 (e) and (f) for an illustration.

In this work, we introduce an accurate and robust mask as the degradation function M, which we refer to as **k-Mask**. An accurate and robust degradation shall be based on the knowledge of corresponding points between $\mathbf{x}_{in}$ and a general $\mathbf{x}_c$. In fact, point correspondences are ambiguous, far less straightforward, and variant to different generated shapes. To this end, we dynamically obtain the point correspondences between $\mathbf{x}_{in}$ and a specific $\mathbf{x}_c$ based on the Euclidean distance. In view of the correspondence ambiguity, we opt for multiple corresponding points for a robust design. Specifically, for each point p_i in $\mathbf{x}_{in}$, we look for its k-nearest neighbors from $\mathbf{x}_c$, denoted as $N_k^{\mathbf{x}_c}(p_i)$. Consequently, $\mathbf{x}_p$ can be constructed by the union of these k-nearest neighbors, as shown in Eq. (5).

$$\mathbf{x}_p = \bigcup_{i=1}^{n} \{q_j \in N_k^{\mathbf{x}_c}(p_i) \mid p_i \in \mathbf{x}_{in}\} \tag{5}$$

Alternative Design Variants. We also provide other alternative masks for comparison. As stated above, the **voxel-Mask** is an intuitive design that directly extends the 2D binary mask to the 3D domain. Voxelization of $\mathbf{x}_{in}$ gives its voxel occupancy, such that $\mathbf{x}_p$ simply consists of all the points of $\mathbf{x}_c$ that correspond to the occupied voxels in $\mathbf{x}_{in}$. **τ-Mask** determines the corresponding regions based on a predefined threshold. Eq. (6) describes $\mathbf{x}_p$, which consists of all points of $\mathbf{x}_c$ where the L2 distance from its nearest neighbor is within a threshold τ.

$$\mathbf{x}_p = \{q \in \mathbf{x}_c \mid \min_{p \in \mathbf{x}_{in}} ||p - q||_2 < \tau\} \tag{6}$$

As illustrated in Fig. 4, k-Mask provides an accurate and robust degradation whereas the other masks fail to achieve these two goals concurrently. This is because both voxel-Mask and τ-Mask leverage essentially fixed distance thresholds and are thus unable to adapt to changes in point density in certain regions. This observation is in line with the preference of k-NN over ball query in popular point feature extractors [31, 19].

3.3. Loss Function for Inversion

Chamfer Distance (CD) and Earth Mover's Distance (EMD) are the most commonly used structural losses for shape completion, with the latter being more sensitive to details and the density distribution [21]. However, unlike typical training processes of supervised shape completion that measure the distance between two complete shapes, our GAN inversion process compares a specific degraded shape against the given partial shape, which may contain a different number of points, thus making EMD infeasible. We follow the CD-T variant [30, 28] that computes the squared L2 distance, as shown in Eq. (7).

$$\begin{aligned}\mathcal{L}_{CD}(\mathbf{x}_p, \mathbf{x}_{in}) = &\frac{1}{|\mathbf{x}_p|}\sum_{p \in \mathbf{x}_p} \min_{q \in \mathbf{x}_{in}} ||p - q||_2^2 \\ + &\frac{1}{|\mathbf{x}_{in}|}\sum_{q \in \mathbf{x}_{in}} \min_{p \in \mathbf{x}_p} ||p - q||_2^2\end{aligned} \tag{7}$$

As structural losses are typically only concerned about low-level regularity of the point cloud, we also perform feature matching at the observation space hoping to align the geometries more semantically. Following the recent practice in [25], we make use of the discriminator, a network that is trained together with the generator during pretraining. We take the feature from the intermediate layer immediately after max-pooling, which captures more geometric details, and compute the L1 distance as the **Feature Distance** loss, as shown in Eq. (8).

$$\mathcal{L}_{FD} = ||D(\mathbf{x}_p) - D(\mathbf{x}_{in})||_1 \tag{8}$$

The overall loss function is shown in Eq. (9), which is used in both shape completion and reconstruction of complete shapes.

$$\mathcal{L} = \mathcal{L}_{CD}(\mathbf{x}_p, \mathbf{x}_{in}) + \mathcal{L}_{FD}(\mathbf{x}_p, \mathbf{x}_{in}) \tag{9}$$

4. Experiments

We start with an ablation study (Sec. 4.1) and then evaluate ShapeInversion through extensive experiments. Besides shape completion on the virtual scan benchmark (Sec. 4.2), we also compare its generalization with other methods on cross-domain partial shapes (Sec. 4.3) and real-world partial scans (Sec. 4.4). In addition, we also provide qualitative results on multiple valid output under ambiguity (Sec. 4.5) and shape manipulation of completed shapes (Sec. 4.6).

Datasets. To facilitate a comprehensive evaluation, we conduct experiments on both synthetic and real-world partial shapes. The following three forms of synthetic partial shapes are: **a)** virtual scans (*e.g.*, in PCN [36] and CRN [30]) **b)** ball-holed partial shapes (*e.g.*, in PF-Net [15]) and **c)** semantic part-level incompleteness (PartNet [23]), as shown in Fig. 1 (a)-(c). They are all derived from ShapeNet [6]. For real-world scans, we evaluate on objects extracted from three sources: **i)** KITTI (cars) [12], **ii)** ScanNet (chairs and tables) [8], and **iii)** MatterPort3D (chairs and tables) [5], as shown in Fig. 1 (d). Note that we follow the standard practice in the field of shape completion to assume the input is always canonically oriented.

Evaluation Metrics. In Sec. 4.1, we evaluate the fidelity and uniformity of the set of generated shapes against those

Table 1. Effectiveness of PatchVariance on the shape uniformity. PatchVariance achieves the lowest MMD-EMD ↓ (scaled by 10^3) across all the eight categories from ShapeNet, indicating the best uniformity and fidelity for the generated shapes

Methods	Plane	Cabinet	Car	Chair	Lamp	Sofa	Table	Boat	Average
tree-GAN baseline	30.7	52.9	38.4	58.6	59.6	41.2	57.1	42.9	47.7
tree-GAN + expansion penality [21]	39.7	68.7	41.0	59.3	66.7	55.4	66.5	40.3	54.7
tree-GAN + repulsion loss [35]	29.8	54.5	36.9	53.2	61.3	44.9	56.1	40.7	47.2
tree-GAN + PatchVariance (ours)	**28.1**	**35.0**	**30.9**	**45.9**	**52.1**	**35.5**	**47.7**	**36.9**	**39.0**

in the test set using *Minimum Matching Distance-Earth Mover's Distance (MMD-EMD)* [1, 27]. EMD is highly indicative of uniformity as it conducts bijective matching of points between two point clouds. With ground truth in Sec. 4.2 and Sec. 4.3, we evaluate the shape completion performance using *CD* and *F1* score following pcl2pcl [7], where F1 is the harmonic average of the *accuracy* and the *completeness*. Without ground truth in Sec. 4.4, we use *Unidirectional Chamfer Distance (UCD)* and *Unidirectional Hausdorff Distance (UHD)* [7, 33] from the partial input $\mathbf{x}_{in}$ to the generated shape $\mathbf{x}_c$.

Implementation Details. In all experiments, ShapeInversion uses the same tree-GAN that is pre-trained on the ShapeNet train set for complete shape generation. Although tree-GAN is able to generate multi-class 3D point clouds, we follow pcl2pcl and MPC [33], and train single-class models for each class for better fidelity. The resolution of the predicted complete shape is 2048 for all the following experiments. More details can be found in the Supplementary Materials.

4.1. Ablation Study

We first investigate the merit of each module in our framework, covering both the pre-training and the GAN inversion stage.

Effectiveness of PatchVariance. We compare our PatchVariance against expansion penalty [21] and repulsion loss [35]. As shown in Tab. 1, PatchVariance achieves the best result across all categories. From Fig. 5, we can observe that expansion penalty leads to more unevenly distributed point clouds while it penalizes branch expansion, and repulsion loss enforces uniformity at local regions only whereas a global uniformity is obtained with PatchVariance.

Effectiveness of k-Mask and Feature Distance. Tab. 2 shows the ablation study during the GAN inversion stage. Replacing k-Masks with other alternative degradation functions shows significant degradation. The choice of k-nearest neighbors of points in the partial shape provides an accurate and robust degradation, and better adapts to variations in local point density. The use of feature distance provides more semantic information to complement the structural loss, significantly boosting the performance.

4.2. Shape Completion on Virtual Scan Benchmark

We compare with existing supervised and unsupervised methods on the common virtual scan benchmark generated from ShapeNet, first proposed by PCN [36]. For a fair comparison, all the baseline methods are trained with virtual scans provided by CRN [30] (with corresponding complete shapes from ShapeNet train set). Tab. 3 shows that ShapeInversion outperforms the other unsupervised method pcl2pcl by a large margin across all the eight categories, and is comparable to the various supervised methods. Note that the impressive performance of various supervised methods is in part attributed to the coarse-to-fine strategy, some of which even calibrates the coarse output with the partial input during the refining stage [21, 30]. ShapeInversion, in contrast, performs completion in a single stage and achieves comparable results. Besides Fig. 1 (a), more qualitative results can be found in the Supplementary Materials.

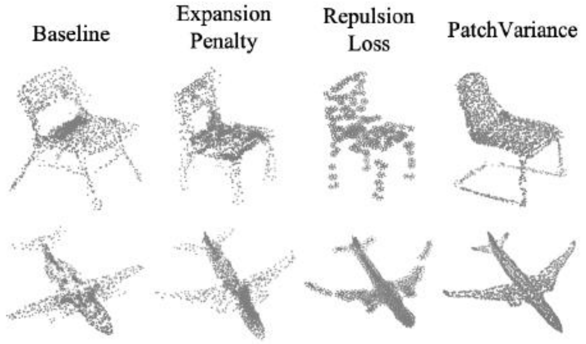

Figure 5. Visualization of randomly generated shapes using various methods for uniformity. Each generated shape contains 2048 points, where darker regions indicates higher point density. PatchVariance achieves the best uniformity

Table 2. Effectiveness of various degradation functions and Feature Distance. Note that the results with these masks are obtained at their respective optimal hyperparameters: 10^3 voxels for voxel-Mask, $\tau = 0.03$ for τ-Mask, and k = 5 for k-Mask

Methods	CD ($\times 10^4$) ↓	acc. ↑	comp. ↑	F1 ↑
Ours w/ voxel-Mask	19.3	84.7	79.7	81.5
Ours w/ τ-Mask	18.9	82.9	81.6	81.6
Ours w/o $\mathcal{L}_{FD}$	16.3	83.6	81.7	81.9
Ours	**14.9**	**85.0**	**84.0**	**83.9**

4.3. Robustness to Varying Partial Forms

To mimic various causes of partial shapes such as occlusion and self-occlusion, various partial forms such as ball-holed partial shapes and virtual scans are considered in different works. We demonstrate the robustness of ShapeInversion under three different partial forms in Tab. 4. Su-

Table 3. Shape completion results on ShapeNet benchmark. The numbers shown are [CD ↓ / F1 ↑], where CD is scaled by 10^4. ShapeInversion outperforms pcl2pcl by a large margin, and is comparable to the various supervised methods. *sup.*: supervised methods; *unsup.*: unsupervised methods

	Methods	Plane	Cabinet	Car	Chair	Lamp	Sofa	Table	Boat	Average
sup.	PCN [36]	3.5/96.5	11.3/86.4	6.4/94.0	11.0/86.0	11.6/84.6	11.5/85.2	10.4/89.4	7.4/91.7	9.1/89.2
	TopNet [28]	4.1/96.0	12.9/84.1	7.8/91.3	13.4/82.3	14.8/79.4	16.0/80.8	12.9/85.7	8.9/89.3	11.4/86.1
	MSN [21]	2.9/97.4	12.5/85.5	7.1/92.3	10.6/86.8	9.3/88.6	12.0/83.3	9.6/91.3	6.5/93.1	8.8/89.8
	CRN [30]	2.3/98.3	11.4/86.2	6.2/93.8	8.8/89.7	8.5/90.2	11.3/85.1	9.3/92.9	6.1/94.2	8.0/91.3
unsup.	pcl2pcl [7]	9.8/89.1	27.1/68.4	15.8/80.0	26.9/70.4	25.7/70.4	34.1/58.4	23.6/79.0	15.7/77.8	22.4/74.2
	Ours	**5.6/94.3**	**16.1/77.2**	**13.0/85.8**	**15.4/81.2**	**18.0/81.7**	**24.6/78.4**	**16.2/85.5**	**10.1/87.0**	**14.9/83.9**

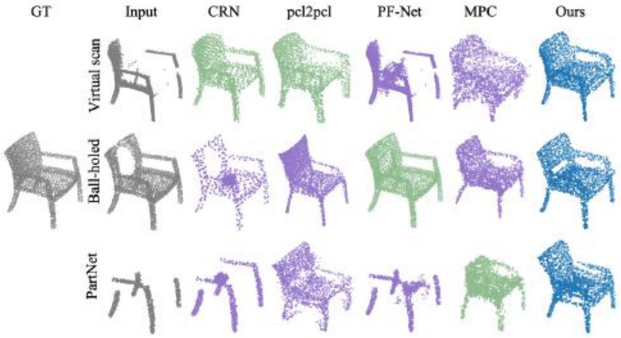

Figure 6. Visualization of cross-domain validation. Different partial forms of the same object are tested. In-domain results are in green whereas out-of-domain ones are in purple. Supervised methods, CRN and PF-Net, show significant performance drop with domain change; unsupervised methods pcl2pcl and MPC show relatively better results for out-of-domain inputs. In contrast, ShapeInversion constantly provides plausible and accurate outputs for all partial forms. Note that CRN leverages the partial input during the refinement stage; PF-Net only predicts the missing regions and combines the partial input as the final output

pervised methods may bias towards the partial forms seen in the training pairs and in turn give poor results on out-domain data, even with auxiliary adversarial loss (*e.g.*, CRN). The unsupervised pcl2pcl performs better than the supervised methods. For ShapeInversion, the GAN is pretrained with complete shapes only, and the degradation via k-Mask during the inversion stage is invariant to partial form changes. In this way, ShapeInversion achieves the best results across almost all the domains. See Fig. 6 and Fig. 1 (a)-(c) for qualitative results.

Note that PF-Net is trained to generate missing regions only for the ball-holed partial shape, which is not compatible with other partial forms with multiple missing regions; although MPC is able to give multiple outputs in view of ambiguity in the partial shape, we report results from its single output for a fair comparison. To further ensure fairness, we remove the shapes from the PartNet test split that are present in the ShapeNet train set.

Table 4. Cross-domain validation. We follow the literature to train each method on a certain partial form (*source*) and cross-validate on other partial forms (*targets*). For each target domain, the SOTA in-domain results are listed at the first line for reference. Methods, especially supervised ones, usually perform well on the in-domain data but suffer large performance drops on the out-of-domain data, whereas ShapeInversion gives the best results for almost all the cross-domain tests, highlighting its robustness to partial form changes. The metric is CD↓ ($\times 10^4$)

Target	Methods	Source	Chair	Table	Lamp
Virtual scan	CRN	Virtual scan	8.8	9.3	8.5
	MPC [33]	PartNet	45.9	88.9	63.0
	Ours	-	**15.4**	**16.2**	**18.0**
Ball-holed	PF-Net [15]	Ball-holed	11.9	9.9	23.1
	MSN	Virtual scan	79.6	46.6	55.4
	CRN	Virtual scan	44.7	52.9	52.1
	pcl2pcl	Virtual scan	18.6	18.5	21.2
	MPC	PartNet	44.7	28.9	69.5
	Ours	-	**10.1**	**16.0**	**17.3**
PartNet	MPC	PartNet	40.0	51.0	82.0
	MSN	Virtual scan	198.0	143.2	229.9
	CRN	Virtual scan	177.4	140.6	185.9
	pcl2pcl	Virtual scan	51.0	**76.6**	111.2
	Ours	-	**36.8**	77.8	**100.8**

4.4. Completion of Real-World Scans

We investigate the generalization of ShapeInversion further on real-world data extracted from MatterPort3D, ScanNet, and KITTI. Besides the domain gap from virtual scans, these real scans tend to be noisier and more incomplete, *e.g.*, KITTI cars. We quantitatively evaluate the performance of ShapeInversion and pcl2pcl in Tab. 5 using UCD and UHD. Despite that pcl2pcl is retrained with real-world scans, our approach significantly outperforms pcl2pcl in terms of UCD and achieves comparable UHD given that pcl2pcl is trained via UHD. With the further addition of UHD into the loss function, ShapeInversion achieves better UHD results with a small compromise on the UCD performance. The completion results in Fig. 8 reveals that pcl2pcl tends to ignore the geometric details in the partial shape, whereas our results remain highly plausible and faithful.

4.5. Multiple Valid Outputs under Ambiguity

With more severely incomplete input, there is more than one complete shape that makes sense. Our framework can naturally give multiple valid and diversified outputs, as we can inverse from multiple initial values of z, which are se-

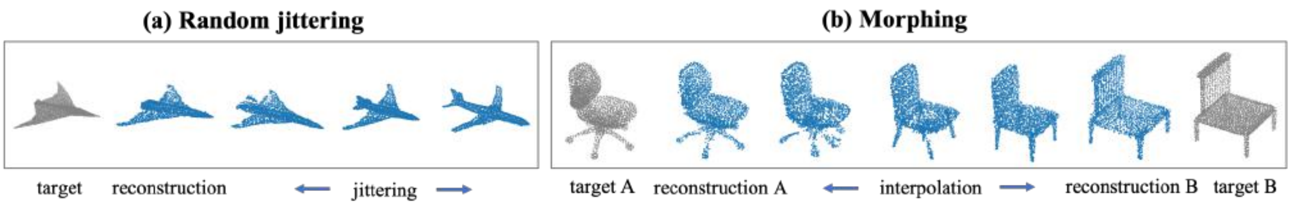

Figure 7. ShapeInversion enables manipulation of complete shapes: **(a)** changing an object into other plausible shapes of different geometries; **(b)** making a sound transition from one shape to another

Table 5. Shape completion results on the real scans. As there is no corresponding ground truth, we evaluate the results using [UCD ↓ / UHD ↓], where UCD is scaled by 10^4 and UHD is scaled by 10^2

Methods	ScanNet		MatterPort3D		KITTI
	Chair	Table	Chair	Table	Car
pcl2pcl	17.3/10.1	9.1/11.8	15.9/10.5	6.0/11.8	9.2/14.1
Ours	**3.2**/10.1	**3.3**/11.9	**3.6**/10.0	**3.1**/11.8	**2.9**/13.8
Ours+UHD	4.0/**9.3**	6.6/**11.0**	4.5/**9.5**	5.7/**10.7**	5.3/**12.5**

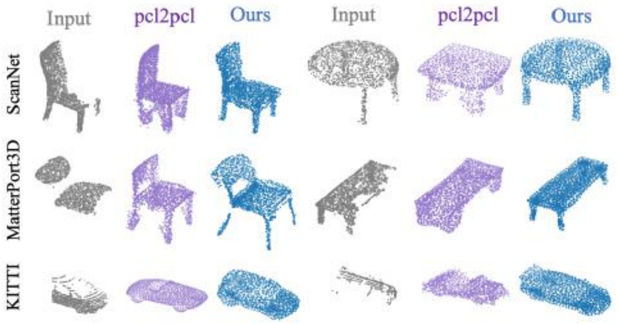

Figure 8. Shape completion on real-world partial scans. Note that pcl2pcl is retrained with real-world partial shapes (together with synthetic complete shapes in an unpaired manner [7]). In contrast, ShapeInversion does not use any real scans, yet, reconstructs high-fidelity shapes that are more faithful to the partial input

lected from hundreds of initial values via FPS, subject to the loss $\mathcal{L}$ being smaller than a threshold $\tau_{\mathcal{L}}$. As demonstrated in Fig. 1 (e) and Fig. 9, ShapeInversion provides multiple reasonable outputs, where each of them faithfully reflects the details in the partial shape.

There exists a trade-off between the diversity and fidelity of the output shapes. In contrast to MPC [33] where the trade-off is predefined during the training by the weights of different losses, our framework offers a more flexible diversity-fidelity trade-off, *e.g.*, we can opt for higher diversity for a particular partial shape by simply choosing a large $\tau_{\mathcal{L}}$ and reducing the number of iterations of inversion.

4.6. Shape Manipulation

Shape manipulation enables interesting applications such as generative design. We show that ShapeInversion can be readily extended to **random jittering** and **morphing**, giving plausible new shapes and sound transition from one shape to another respectively, as shown in Fig. 7. These can be realized efficiently upon shape reconstruction: jittering of a given shape is achieved by introducing perturbation in the latent space; morphing between two given shapes is achieved by interpolation between their corresponding latent vectors $\mathbf{z}$ and generator parameters θ.

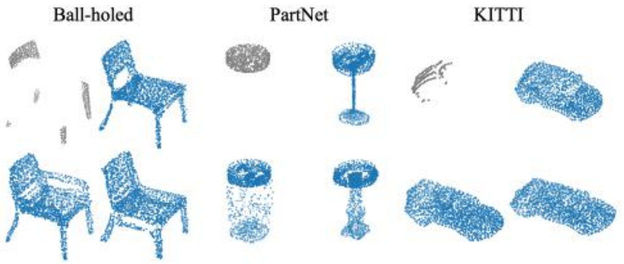

Figure 9. ShapeInversion can give multiple valid outputs when higher incompleteness level of partial shapes impose ambiguity

5. Conclusion

We introduce ShapeInversion for unsupervised point cloud completion. ShapeInversion addresses the domain gaps between virtual and real-world partial scans, and among various simulated partial shapes through GAN inversion. As the very first GAN inversion approach for 3D shape completion, we introduce two new components to address the unique challenges posed by the nature of points clouds: an effective uniform loss, PatchVariance, and an accurate and robust degradation function, k-Mask. With the incorporation of rich knowledge of shape geometries and semantics captured in a well-trained GAN, it achieves remarkable generalization for real-world scans and partial inputs of various forms and incompleteness levels. Moreover, our framework also brings several side-benefits, including giving multiple reasonable complete shapes for one partial input, as well as shape jittering and shape interpolation.

So far, both shape completion and manipulation are conducted on a model pre-trained with a single category. Future works can focus on improving the fidelity of multi-class models, which could provide more possibilities such as cross-category shape completion via a conditional GAN.

Acknowledgements. This research was conducted in collaboration with SenseTime. This work is supported by A*STAR through the Industry Alignment Fund - Industry Collaboration Projects Grant.

A. Additional Details of Evaluation Metrics

Minimum Matching Distance (MMD) [1] measures the fidelity of a set of generated shapes $\mathbb{S}_c$ with respect to the set of ground truth shapes $\mathbb{S}_{gt}$. Specifically, we match every shape $\mathbf{x}_{gt}$ in $\mathbb{S}_{gt}$ to the generated shape $\mathbf{x}_c$ in $\mathbb{S}_c$ with minimum distance, and report the average macthed distance.

Earth Mover's Distance (EMD) [26] is the solution, *i.e.*, the minimum cost, to transport one shape's point cloud to another. It is defined in Eq. (10), where ϕ is a bijection. The bijection is highly indicative of uniformity of the generated shapes. We use the implementation provided by MSN [21].

$$d_{EMD}(\mathbf{x}_A, \mathbf{x}_B) = \min_{\phi:\mathbf{x}_A \to \mathbf{x}_B} \frac{1}{|\mathbf{x}_A|} \sum_{p \in \mathbf{x}_A} ||p - \phi(p)||_2 \tag{10}$$

Unidirectional Chamfer Distance (UCD) measures the squared L2 distance from the partial input shape $\mathbf{x}_p$ to the complete output $\mathbf{x}_c$, as shown in Eq. (11).

$$d_{UCD}(\mathbf{x}_p, \mathbf{x}_c) = \frac{1}{|\mathbf{x}_p|} \sum_{p \in \mathbf{x}_p} \min_{q \in \mathbf{x}_c} ||p - q||_2^2 \tag{11}$$

Unidirectional Hausdorff Distance (UHD) [7, 33], similarly, measures the single-sided Hausdorff distance:

$$d_{UHD}(\mathbf{x}_p, \mathbf{x}_c) = \max_{p \in \mathbf{x}_p} \min_{q \in \mathbf{x}_c} ||p - q||_2^2 \tag{12}$$

F1 score, following pcl2pcl [7], is defined as the harmonic average of the **accuracy** and the **completeness**, where accuracy measures the fraction of points in the $\mathbf{x}_c$ that are matched in the corresponding ground truth shape $\mathbf{x}_{gt}$, and completeness measures the fraction of points in the $\mathbf{x}_{gt}$ that are matched in $\mathbf{x}_c$. Here, the status of matching is determined by a threshold $\epsilon = 0.03$ for L2 distance.

B. Additional Details of Datasets

In Sec. 4.3 of the main paper, both the virtual scans and the ball-holed partial shapes follow the same train-test split of ShapeNet [6], whereas the PartNet [23] follows the train-test split provided by MPC [33]. As some shapes in the PartNet test set are present in the ShapeNet train set, we remove them for evaluating the PartNet target domain. The ball-holed partial shapes in this section are generated as described in PF-Net [15], *i.e.*, removing exactly 512 points from the complete shape. But in Fig. 1 and Fig. 9 of the main paper, higher incompleteness levels are tested to assess the robustness of ShapeInversion.

The real-world partial scans are provided by pcl2pcl [7]. Both ScanNet [8] and KITTI [12] are split into train set and test set, and the mapping GAN of the pcl2pcl framework, which maps the latent space of partial shapes to that of the complete ones, is retrained on the real scan train set. Note that MatterPort3D [5] does not have its own train set, so pcl2pcl is evaluated on MatterPort3D using the model retrained on ScanNet, given that both these two datasets are captured by depth cameras. In contrast, ShapeInversion does not need the real scan train set, and is directly evaluated on the test set.

C. Additional Details on Implementation

Pre-training. We follow Shu *et al.* [27] to train the tree-GAN baseline. The PatchVariance module has $n = 100$ patches, each with $k = 30$ points. This would ensure the entire generated shape (with 2048 points) is sampled via FPS. This setting works for r-GAN [1] as well, as shown in Fig. 10. tree-GAN is trained on eight Nvidia V100 GPUs for 2000 epochs. For the four categories with more than 3000 shapes, *i.e.*, plane, car, chair, and table, the batch size is 512; for the other four categories with fewer shapes, *i.e.*, cabinet, lamp, sofa, and boat, the batch size is 128 to train for enough iterations.

Inversion. The k in k-Mask is 5. We manually split the inversion stage into four sub-stages, each with different learning rates for z and θ: $\alpha_z = [1\times10^{-2}, 1\times10^{-4}, 1\times10^{-5}, 1\times10^{-6}]$, and $\alpha_\theta = [2\times10^{-7}, 1\times10^{-6}, 1\times10^{-6}, 2\times10^{-7}]$. For bulk structures, *i.e.*, car, couch, cabinet, and plane, each sub-stage consists of 30 iterations; for thin structures, *i.e.*, chair, lamp, table, and boat, each sub-stage consists of 200 iterations.

D. Additional Qualitative Results

We show more qualitative results on shape completion for virtual scans (Fig. 12), cross-data validation (Fig. 13), real scans (Fig. 11), shape jittering (Fig. 14), shape morpging (Fig. 15), as well as on the effectiveness of PatchVariance on r-GAN (Fig. 10).

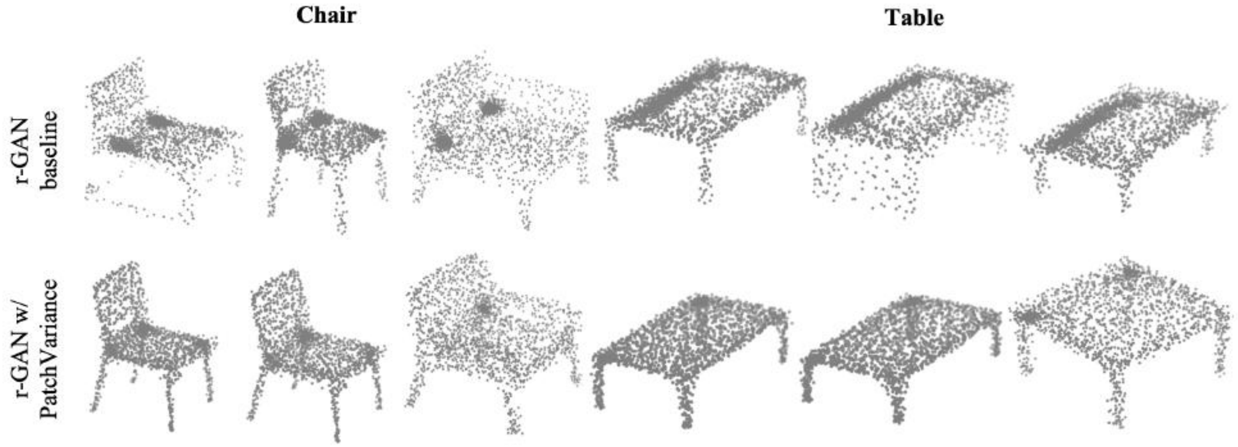

Figure 10. Validation of the significance of PatchVariance on r-GAN [1]. Non-uniformity issue is a common problem faced by point cloud GANs, our PatchVariance loss directly works on the generated point cloud, and is invariant to the architecture of the generator, hence it effectively enhances the uniformity of generated point clouds for r-GAN as well. Darker regions have a higher density of points

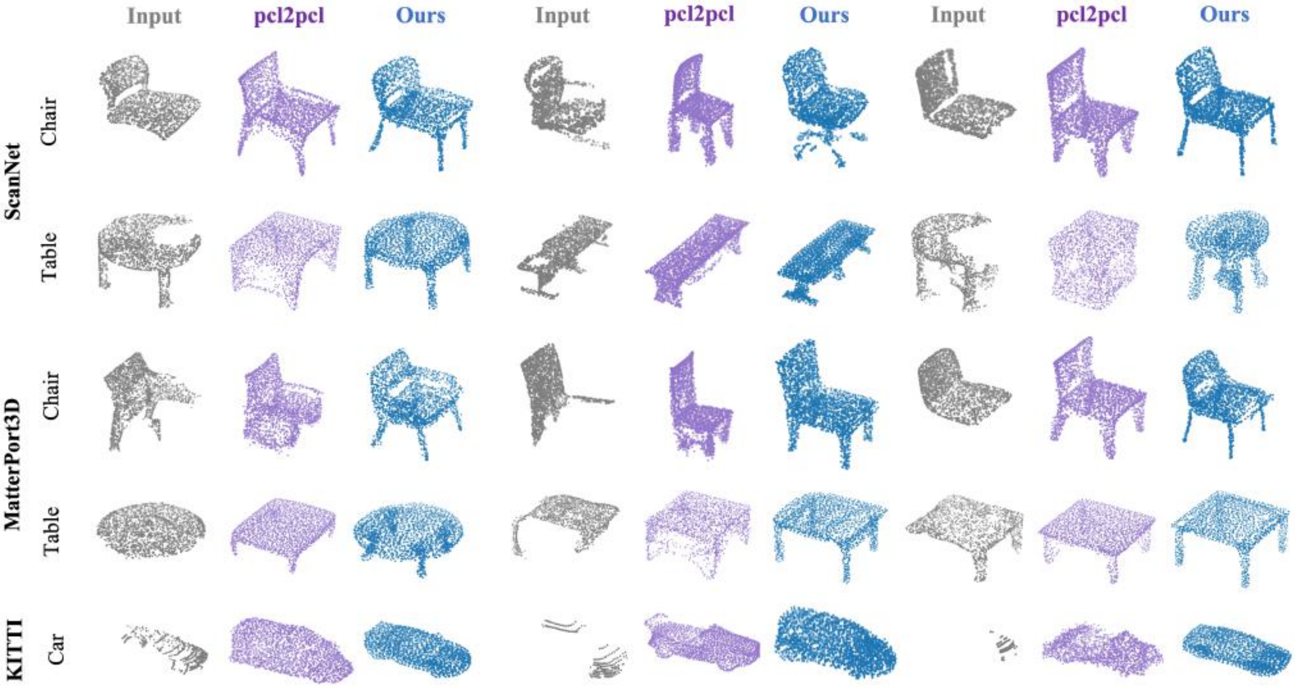

Figure 11. More qualitative results for shape completion on real-world partial scans. Note that pcl2pcl is retrained with real-world partial shapes (together with synthetic complete shapes in a unpaired manner [7]). In contrast, ShapeInversion does not use any real scans, yet, reconstructs high-fidelity shapes that are more faithful to the partial input

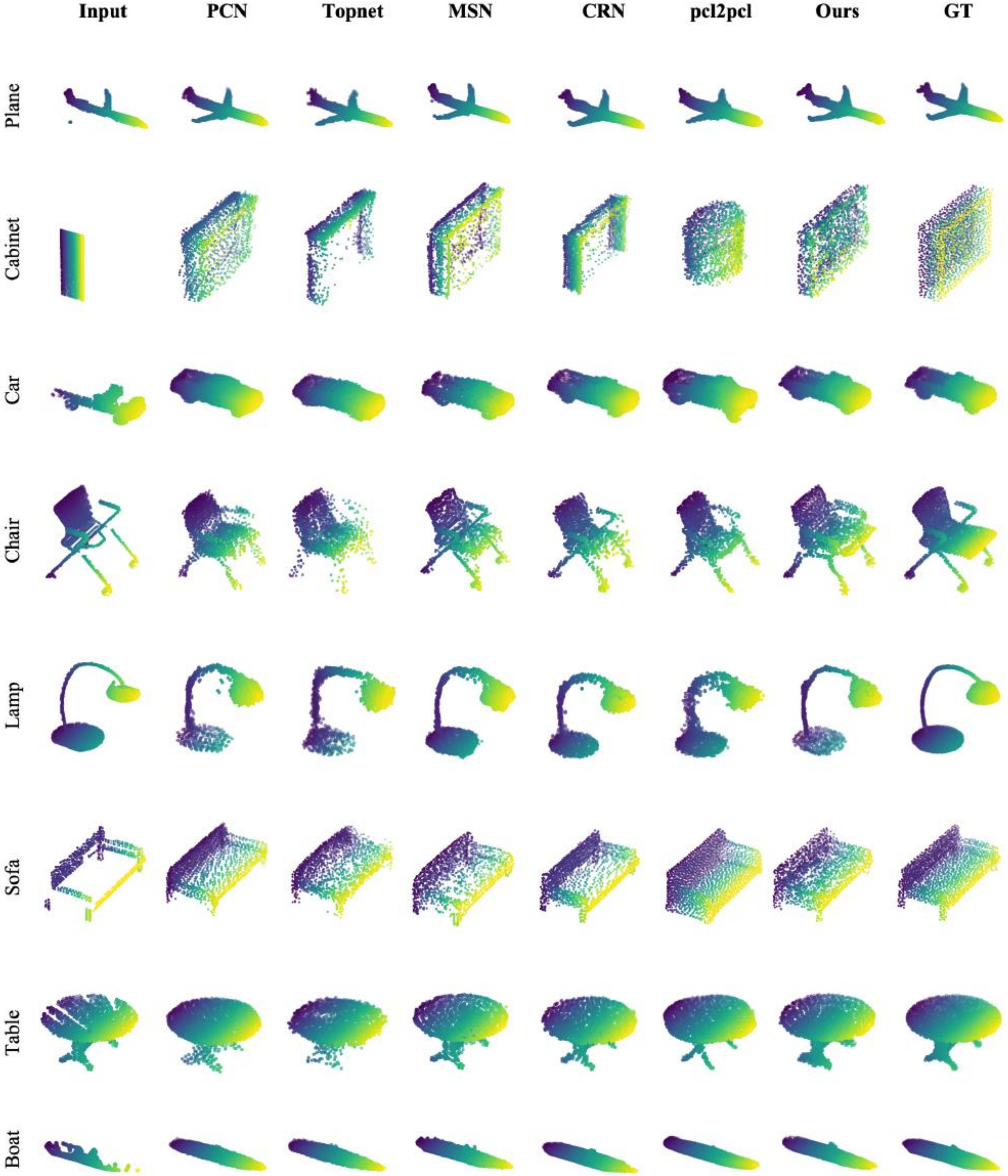

Figure 12. Qualitative comparison on the ShapeNet benchmark. The resolution of partial inputs, complete shapes, and ground truths are all 2048 points

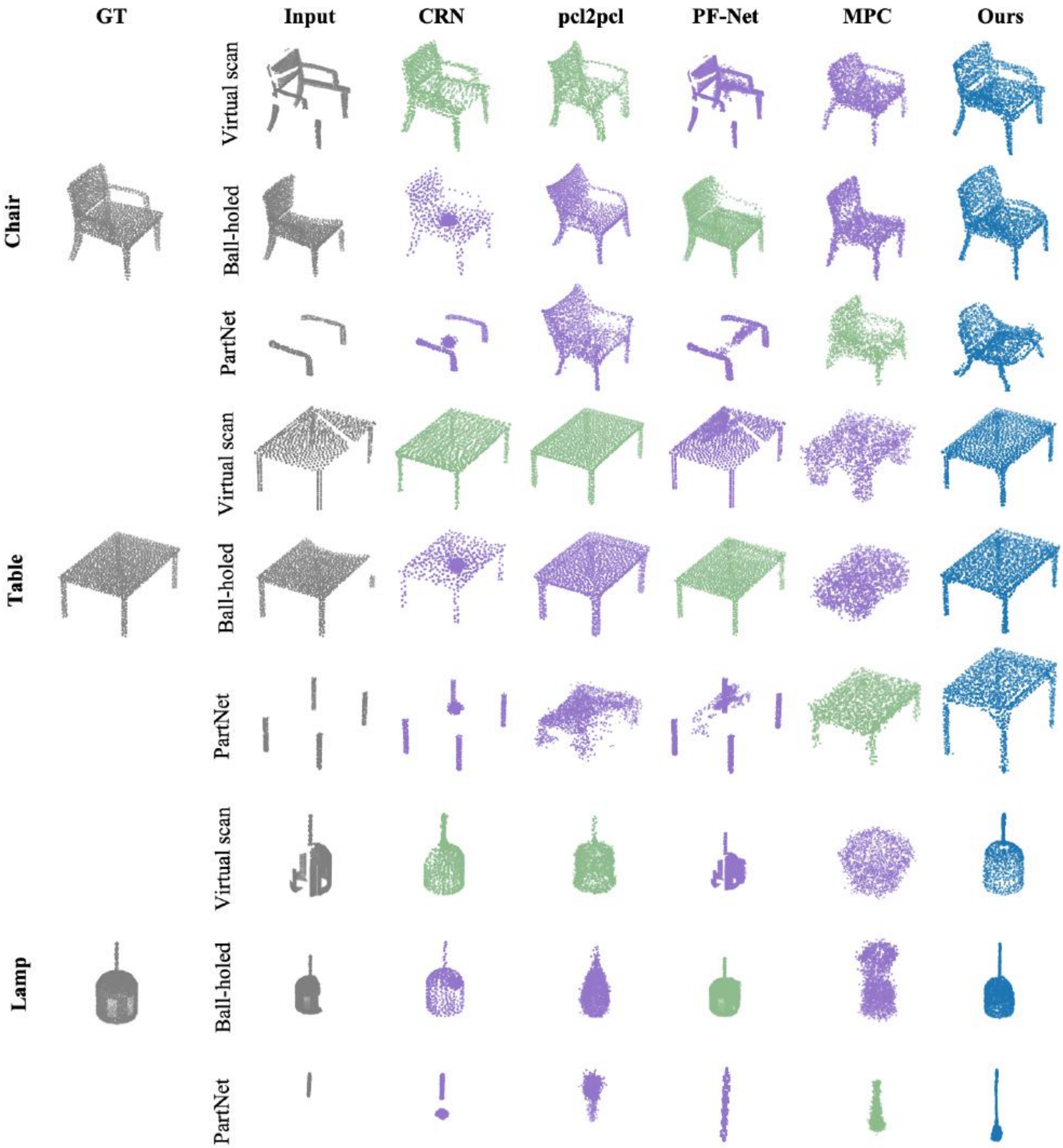

Figure 13. Additional visualization of cross-domain validation. Different partial forms of the same object are tested. In-domain results are in dark green whereas out-of-domain ones are in purple. Supervised methods CRN and PF-Net show significant performance drop with domain change; unsupervised methods pcl2pcl and MPC show relatively better results for out-of-domain inputs. ShapeInversion constantly provides plausible and accurate outputs for all partial forms. Note that CRN leverages the partial input during the refinement stage; PF-Net only predicts the missing regions and combines the partial input as the final output

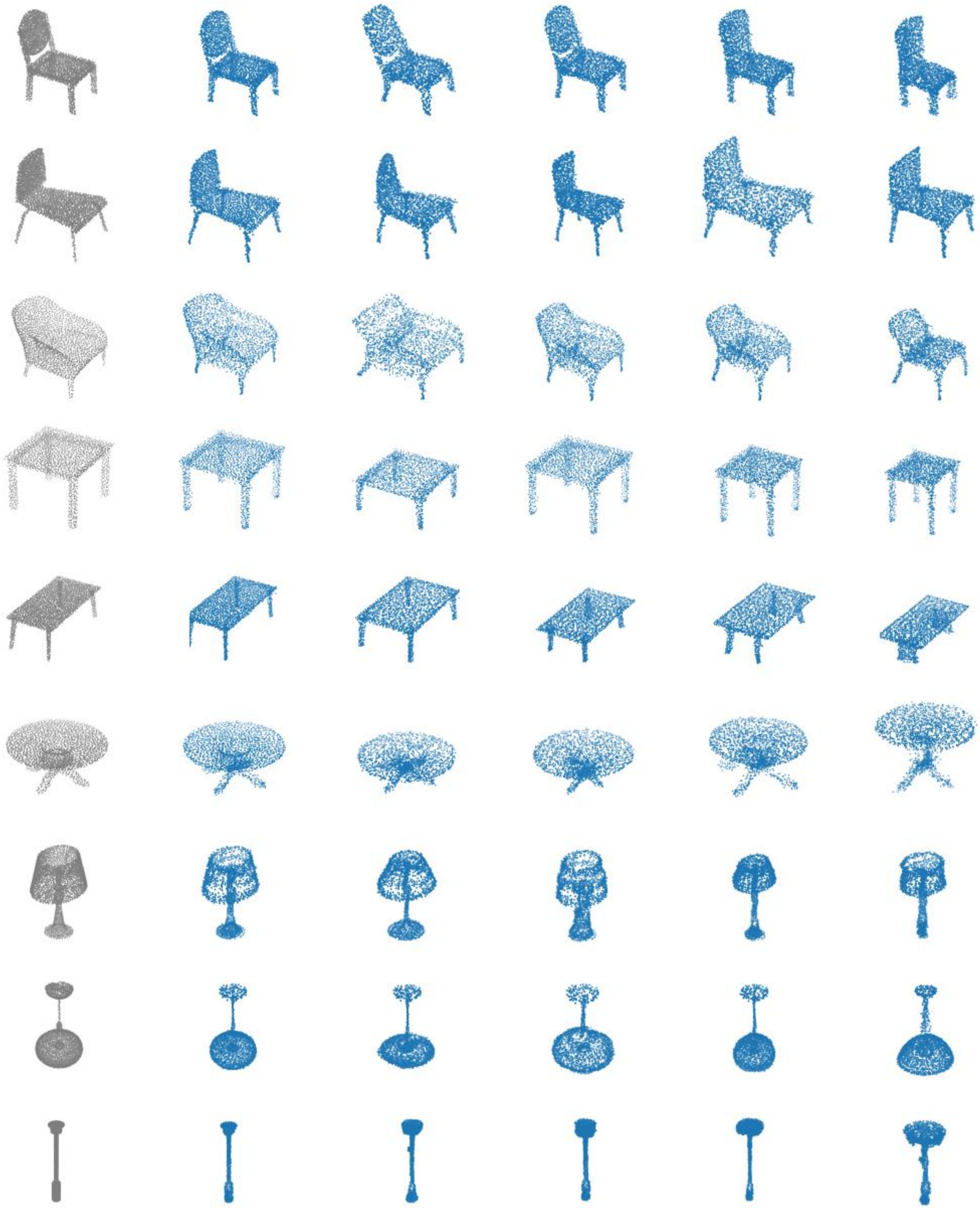

Figure 14. Additional qualitative results for shape random jittering by ShapeInversion. Random jittering changes one shape into other plausible shapes of different geometries by introducing perturbation in the latent space

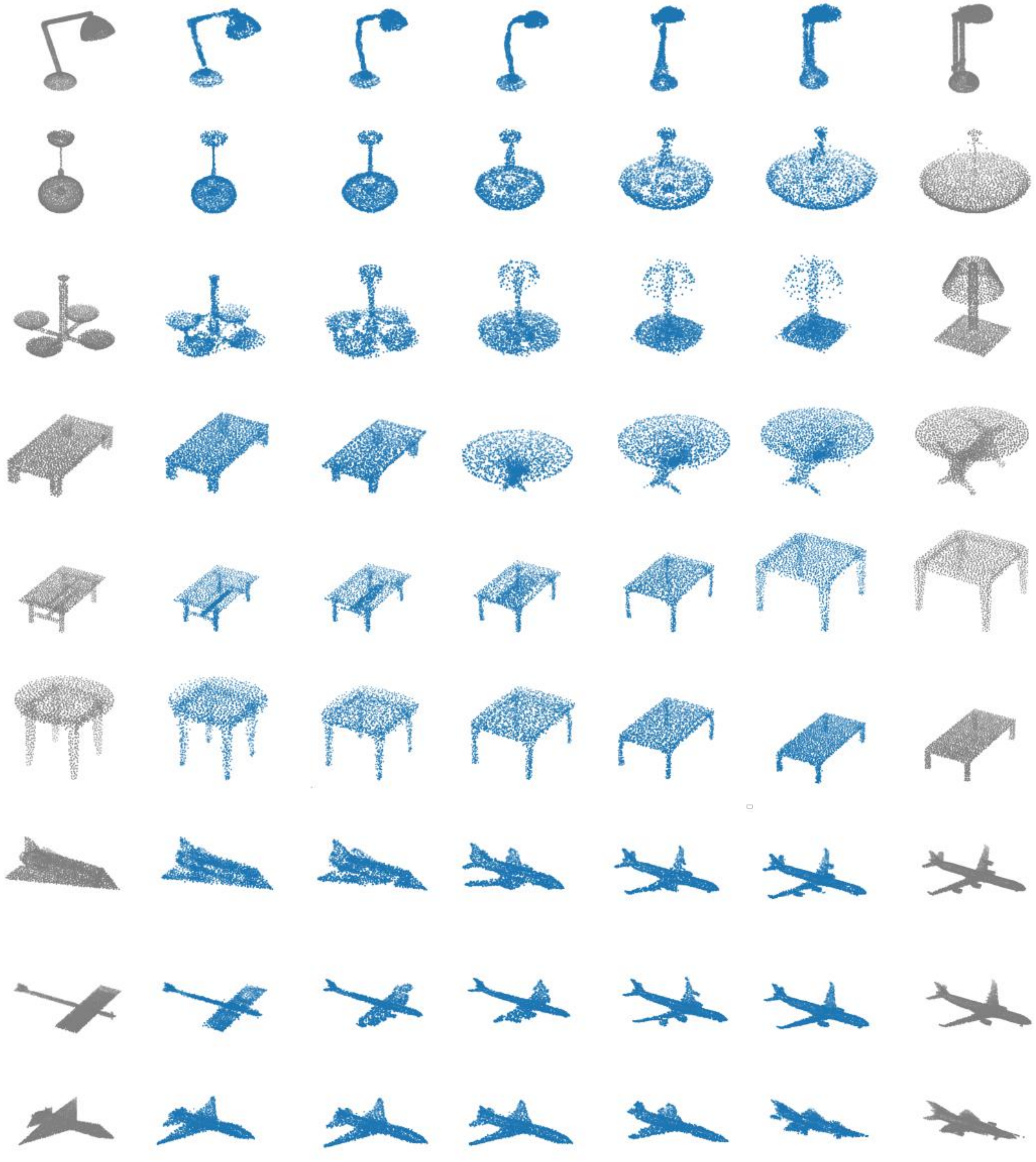

Figure 15. Additional qualitative results for shape interpolation (morphing) by ShapeInversion. A sound transition from one shape to another is achieved by interpolation between their corresponding latent vectors z and generator parameters θ

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In *ICML*, 2018. 4, 6, 9, 10

[2] David Bau, Hendrik Strobelt, William Peebles, Bolei Zhou, Jun-Yan Zhu, Antonio Torralba, et al. Semantic photo manipulation with a generative image prior. In *SIGGRAPH*, 2019. 3

[3] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a GAN cannot generate. In *ICCV*, 2019. 3

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019. 3

[5] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *3DV*, 2017. 5, 9

[6] 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. 5, 9

[7] Xuelin Chen, Baoquan Chen, and Niloy J Mitra. Unpaired point cloud completion on real scans using adversarial training. In *ICLR*, 2020. 2, 6, 7, 8, 9, 10

[8] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017. 5, 9

[9] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. ScanComplete: Large-scale scene completion and semantic segmentation for 3D scans. In *CVPR*, 2018. 1, 2

[10] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3D-encoder-predictor CNNs and shape synthesis. In *CVPR*, 2017. 2

[11] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *ECCV*, 2014. 1, 2

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012. 5, 9

[13] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code GAN prior. In *CVPR*, 2020. 3

[14] Ji Hou, Angela Dai, and Matthias Nießner. 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In *CVPR*, 2019. 1, 2

[15] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. PF-Net: Point fractal network for 3D point cloud completion. In *CVPR*, 2020. 1, 2, 5, 7, 9

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 3

[17] Qi Lei, Ajil Jalal, Inderjit S Dhillon, and Alexandros G Dimakis. Inverting deep generative models, one layer at a time. In *NeurIPS*, 2019. 3

[18] Ruihui Li, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. PU-GAN: A point cloud upsampling adversarial network. In *ICCV*, 2019. 2, 4

[19] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution on X-transformed points. In *NeurIPS*, 2018. 5

[20] Zachary C Lipton and Subarna Tripathi. Precise recovery of latent vectors from generative adversarial networks. *arXiv preprint arXiv:1702.04782*, 2017. 3

[21] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. In *AAAI*, 2019. 1, 2, 4, 5, 6, 7, 9

[22] Fangchang Ma, Ulas Ayaz, and Sertac Karaman. Invertibility of convolutional generative networks from partial measurements. In *NeurIPS*, 2018. 3

[23] 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 *CVPR*, 2019. 2, 5, 9

[24] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 1, 2

[25] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. In *ECCV*, 2020. 3, 5

[26] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vis.*, 40(2):99–121, 2000. 9

[27] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3D point cloud generative adversarial network based on tree structured graph convolutions. In *ICCV*, 2019. 3, 6, 9

[28] Lyne P Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. TopNet: Structural point cloud decoder. In *CVPR*, 2019. 1, 2, 5, 7

[29] Diego Valsesia, Giulia Fracastoro, and Enrico Magli. Learning localized generative models for 3D point clouds via graph convolution. In *ICLR*, 2018. 4

[30] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *CVPR*, 2020. 1, 2, 5, 6, 7

[31] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics*, 38(5):1–12, 2019. 5

[32] Xin Wen, Tianyang Li, Zhizhong Han, and Yu-Shen Liu. Point cloud completion by skip-attention network with hierarchical folding. In *CVPR*, 2020. 1, 2

[33] Rundi Wu, Xuelin Chen, Yixin Zhuang, and Baoquan Chen. Multimodal shape completion via conditional generative adversarial networks. In *ECCV*, 2020. 3, 6, 7, 8, 9

[34] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. GRNet: Gridding residual network for dense point cloud completion. In *ECCV*, 2020. 2

[35] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. PU-Net: Point cloud upsampling network. In *CVPR*, 2018. 2, 4, 6

[36] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. PCN: Point completion network. In *3DV*, 2018. 2, 5, 6, 7

[37] Wenxiao Zhang, Qingan Yan, and Chunxia Xiao. Detail preserved point cloud completion via separated feature aggregation. In *ECCV*, 2020. 2

[38] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain GAN inversion for real image editing. In *ECCV*, 2020. 3

[39] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *ECCV*, 2016. 3