# ShapeFormer: Transformer-based Shape Completion via Sparse Representation

Xingguang Yan[1] Liqiang Lin[1] Niloy J. Mitra[2] Dani Lischinski[3] Danny Cohen-Or[4] Hui Huang[1]
[1]Shenzhen University [2]University College London [3]Hebrew University of Jerusalem [4]Tel Aviv University
yanxg.art/shapeformer

## Abstract

*We present ShapeFormer, a transformer-based network that produces a distribution of object completions, conditioned on incomplete, and possibly noisy, point clouds. The resultant distribution can then be sampled to generate likely completions, each exhibiting plausible shape details while being faithful to the input.*

*To facilitate the use of transformers for 3D, we introduce a compact 3D representation, vector quantized deep implicit function (VQDIF), that utilizes spatial sparsity to represent a close approximation of a 3D shape by a short sequence of discrete variables. Experiments demonstrate that ShapeFormer outperforms prior art for shape completion from ambiguous partial inputs in terms of both completion quality and diversity. We also show that our approach effectively handles a variety of shape types, incomplete patterns, and real-world scans.*

## 1. Introduction

Shapes are typically acquired with cameras that probe and sample surfaces. The process relies on line of sight, and, at best, can obtain partial information from the visible parts of objects. Hence, sampling complex real-world geometry is inevitably imperfect, resulting in varying sampling densities and missing parts. This problem of surface completion has been extensively investigated over multiple decades [4]. The central challenge is to compensate for incomplete data by inspecting non-local hints in the observed data to infer missing parts using various forms of priors.

Recently, deep implicit function (DIF) has emerged as an effective representation for learning high-quality surface completion. To learn shape priors, earlier DIFs [12, 43, 49] encode each shape using a single global latent vector. Combining a global code with region-specific local latent codes [13, 14, 22, 27, 37, 51] can faithfully preserve geometric details of the input in the completion. However, when presented with *ambiguous* partial input, for which multiple plausible completions are possible (see Fig. 1), the deterministic nature of local DIF usually fails to produce meaningful completions for unseen regions. A viable alternative is to combine generative models to handle the input uncertainty. However, for representations that contain huge sta-

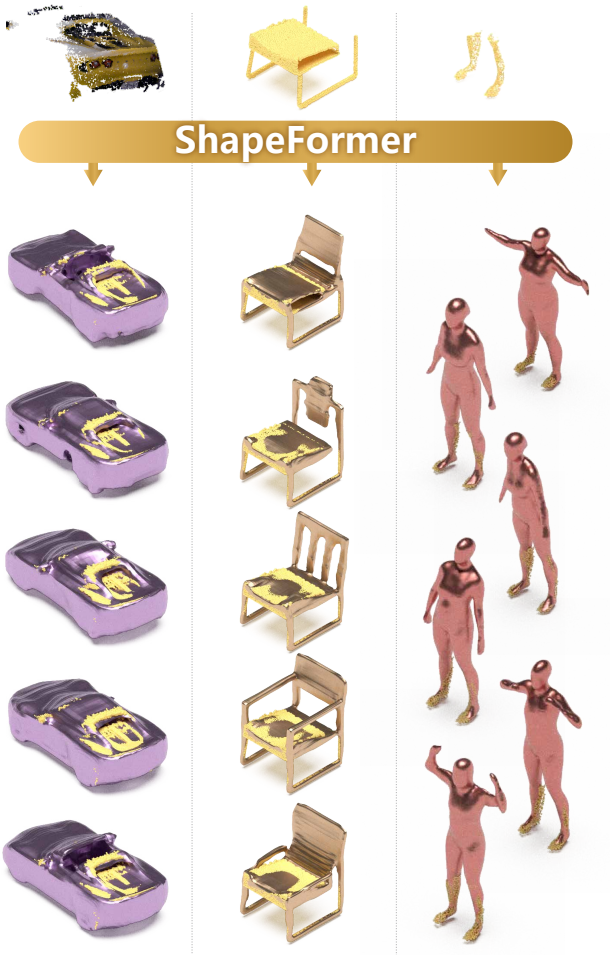



Figure 1. ShapeFormer predicts multiple completions for a real-world scan of a sports car (left column), a chair with missing parts (middle column), and a partial point cloud of human lower legs (right column). The input point clouds are superimposed with the generated shapes to emphasize the faithfulness of the completion to the input point cloud.

tistical redundancy, as in the case of current local methods, such combination [58] excessively allocates model capacity towards perceptually irrelevant details [21, 25].

We present *ShapeFormer*, a transformer-based autoregressive model that learns a *distribution* over possible shape completions. We use local codes to form a sequence of discrete, vector quantized features, which greatly reduces the representation size, while keeping the underlying structure. Applying transformer-based generative models toward such sequences of discrete variables have been shown to be effective for progressive image generation [2, 10] and various image synthesis tasks [10, 23, 54] including pluralistic image completion [65].

However, directly deploying transformers to 3D feature grids leads to a sequence length cubic in the feature resolution. Since transformers have an innate quadratic complexity on sequence length, only using overly coarse feature resolution, while feasible, can barely represent meaningful shapes. To mitigate the complexity, we first introduce Vector Quantized Deep Implicit Functions (*VQDIF*), a novel 3D representation which is both compact and structured, that can represent complex 3D shapes with acceptable accuracy, while being rather small in size. The core idea is to sparsely encode shapes as a sequences of discrete 2-tuples, each representing both the position and content of a non-empty local feature. These sequences can be decoded to deep implicit functions from which high-quality surfaces can subsequently be extracted. Due to the sparse nature of 3D shapes, such encoding reduces the sequence length from cubic to quadratic in the feature resolution, thus enabling effective combination with generative models.

*ShapeFormer* completes shapes by generating complete sequences, conditioned on the sequence for partial observation. It is trained by sequentially predicting the conditional distribution of both location and content over the next element. Unlike image completion [65], where the model is trained with the BERT [2, 20] objective to only predict for unseen regions, in the 3D shape completion setting, the input features may also come from both noisy and incomplete observations, and keeping them intact necessarily yields noisy results. Hence, in order to generate whole complete sequences from scratch while being faithful to the partial observations, we adapt the auto-regressive objective and prepend the partial sequence to the complete one to achieve conditioning. This strategy has been proved effective for conditional synthesis for both text [41] and images [23].

We demonstrate the ability of ShapeFormer to produce diverse high-quality completions for ambiguous partial observations for various shape types, including CAD models and human bodies; for various incomplete sources such as real world scans with missing parts.

In summary, our contributions include: (i) a novel DIF representation based on sequences of discrete variables that compactly represents satisfactory approximations of 3D shapes; (ii) a transformer-based autoregressive model that uses our new representation to predict multiple high-quality completed shapes conditioned on the partial input; and (iii) state-of-the-art results for multi-modal shape completion in terms of completion quality and diversity. The FPD score on PartNet is improved by at most 1.7 compared with prior multi-modal method cGAN [68].

## 2. Related Work

**Shape reconstruction and completion**. 3D reconstruction is a longstanding ill-posed problem in computer vision and graphics. Traditional methods can produce faithful reconstruction from complete input such as point cloud [4], or images [26]. Recently, neural network-based methods have demonstrated an impressive performance toward reconstruction from partial input [30], where the unseen regions are completed with the help of data priors. They can be classified according to their output representation, such as voxels, meshes, point clouds, and deep implicit functions. Since voxels can be processed or generated easily through 3D convolutions thanks to their regularity, they are commonly used in earlier works [17, 19, 31, 57]. However, since their cubic complexity toward resolution, the predicted shapes are either too coarse or too heavy in size for later applications. While meshes are more data-efficient, due to the difficulty of handling mesh topology, mesh-based methods have to either use shape template [40, 55, 66], limiting to a single topology, or produce self-intersecting meshes [29]. Point clouds, in contrast, do not have such a problem and are popularly used lately [1, 24, 60, 69, 70]. However, point clouds need to be non-trivially post-processed using classical methods [5, 35, 38, 39] to recover surfaces due to their sparse nature. Recent works that represent shapes as deep implicit functions have been shown to be effective for high-quality 3D reconstruction [12, 43, 49]. By leveraging local priors, follow-up works [14, 22, 27, 42, 51] can further improve the fidelity of geometric details. However, most current methods are not effective toward ambiguous input due to their deterministic nature. Other methods handle such input by leveraging generative models. They learn the conditional distribution of complete shapes represented as either a single global code [68], which, due to their lack of spatial structure, leads to completions misaligned with the input, or raw point cloud [72], which, due to its statistical redundancy, is only effective for completing simple shapes with a limited number of points. In this paper, we show how building generative models upon our new compact; structured representation enables multi-modal high-quality reconstruction for complex shapes.

**Autoregressive models and Transformers**. Autoregres-

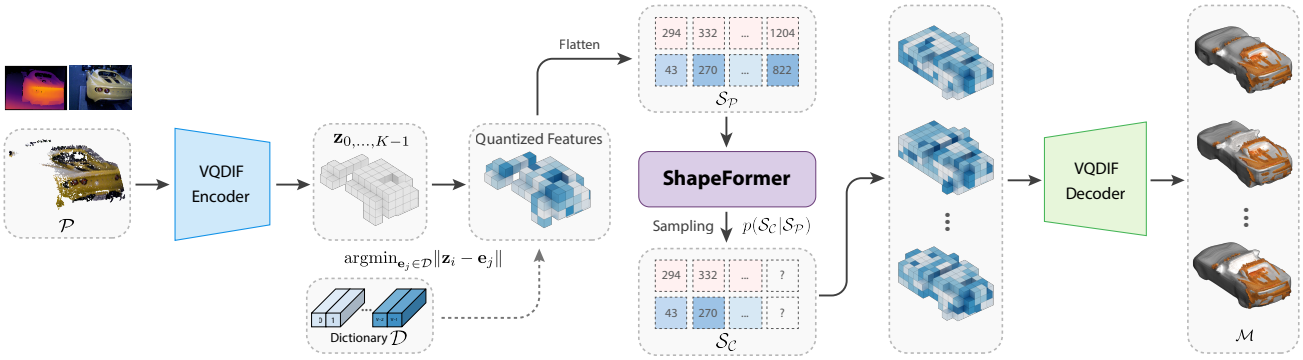



Figure 2. Overview of our shape completion approach. Given a partial point cloud $\mathcal{P}$, possibly from a depth image, as input, our VQDIF encoder first converts it to a sparse feature sequence $\mathbf{z}_{0\ldots K-1}$, replacing them with the indices of their nearest neighbor $\mathbf{e}_j$ in a learned dictionary $\mathcal{D}$, forming a sequence of discrete 2-tuples consisting of the coordinate (pink) and the quantized feature index (blue). We refer to this partial sequence as $\mathcal{S}_\mathcal{P}$ (drawn with dashed lines). The ShapeFormer then takes $\mathcal{S}_\mathcal{P}$ as input and models the conditional distribution $p(\mathcal{S}_\mathcal{C}|\mathcal{S}_\mathcal{P})$. Autoregressive sampling yields a probable complete sequence $\mathcal{S}_\mathcal{C}$. Finally, the VQDIF decoder converts the sequence $\mathcal{S}_\mathcal{C}$ to a deep implicit function, from which the surface reconstruction $\mathcal{M}$ can be extracted. To show the faithfulness of our reconstructions, we super-impose the input point cloud on them.

sive models are generative models that aim to model distributions of high dimensional data by factoring the joint probability distribution to a series of conditional distributions via the chain rule [3]. Using neural networks to parameterize the conditional distribution has been proved to be effective [28, 61] in general, and more specifically to image generation [11, 48, 63]. Transformers [64], known for their ability to model long-range dependencies through self-attentions, have shown the power of autoregressive models in natural languages [7, 52], image generation [10, 50]. Contrary to deterministic masked auto-encoders [32], Transformers can produce diverse image completions [65] that are sharp in masked regions by adopting the BERT [20] training objective. In the 3D domain, autoregressive models have been used to learn the distribution of point clouds [58, 67] and meshes [45]. However, due to the lack of efficient representation, these models can only generate small point clouds or meshes restricted to 1024 vertices. In contrast, by eliminating statistical redundancy, a compressed discrete representation enables generative models to focus on data dependencies at a more salient level [54, 62] and recently allows high-resolution image synthesis [23, 53]. Follow-up works utilize data sparsity to obtain even more compact representations [21, 46]. We explore this direction in the context of surface completion.

# 3. Method

We model the shape completion problem as mapping a partial point cloud $\mathcal{P} \in \mathbb{R}^{N\times 3}$ to a complete, watertight surface mesh $\mathcal{M}$ which matches the cloud. Since this is an ill-posed problem, we seek to estimate the probabilistic distribution of such mesh $p(\mathcal{M}|\mathcal{P})$ utilizing the power of Transformers. Instead of working directly on point clouds, meshes, or feature grids, we approximate shapes as short discrete sequences (see Sec. 3.1) to greatly reduce both the number of variables and the variable bit size, which enables Transformers to complete complex 3D shapes (see Sec. 3.2).

With such compact representation, the conditional distribution becomes $p(\mathcal{S}_\mathcal{C}|\mathcal{S}_\mathcal{P})$, where $\mathcal{S}_\mathcal{P}$ and $\mathcal{S}_\mathcal{C}$ are the sequence encoding of the partial point cloud and the complete shape, respectively. Once such distribution is modeled, we can sample multiple complete sequences $\mathcal{S}_\mathcal{C}$, from which different surface reconstructions $\mathcal{M}$ can be obtained through decoding. This process is illustrated in Fig. 2.

## 3.1. Compact sequence encoding for 3D shapes

We propose VQDIF, whose goal is to approximate 3D shapes with a shape dictionary, with each entry describing a particular type of local shape part inside a cell of volumetric grid $G$ with resolution $R$. With such a dictionary, shapes can be encoded as short sequences of entry indices, describing the local shapes inside all non-empty grid cells, enabling transformers to model the global dependencies more effectively.

We design an auto-encoder architecture to achieve this. The encoder $E$ first maps the input point cloud to a 64 resolution feature grid with local-pooled PointNet and then downsample it to resolution $R$. Unlike the previous strategy for image synthesis [23], the encoder parameters are carefully set to have the least receptive field, reducing the number of non-empty features to the number of non-empty cells, $K$, of the grid $G$. Then these non-empty features are flattened to a sequence of length $K$ in row-major order. Since these features are sparse, we record their locations with their

flattened index $\{c_i\}_{i=0}^{K-1}$. Other orderings are also possible, but they are not as effective as row-major order for learning sequence distribution [23].

Following the idea of neural discrete representation learning [62], we compress the bit size of the feature sequence $\{\mathbf{z}_i\}_{i=0}^{K-1}$ through vector quantization, that is, clamping it to its nearest entry in a dictionary $\mathcal{D}$ of $V$ embeddings $\{\mathbf{e}_j\}_{j=0}^{V}$ and we save the indices of these entries:

$$v_i = \text{argmin}_{j\in[0,V)} \|\mathbf{z}_i - \mathbf{e}_j\|. \tag{1}$$

Thus, we get a compact sequence of discrete 2-tuples representing the 3D shape $\mathcal{S} = \{(c_i, v_i)\}_{i=0}^{K-1}$. Finally, the decoder projects this sequence back to a feature grid and, through a 3D-Unet [18], decodes it to a local deep implicit function $f$ [51], whose iso-surface is the surface reconstruction $\mathcal{M}$ of $\mathcal{P}$.

**Training**. We train the VQDIF by simultaneously minimizing the reconstruction loss and updating the dictionary using exponential moving averages [62], where dictionary embeddings are gradually pulled toward the encoded features. Also, we also adopt commitment loss $\mathcal{L}_{\text{commit}}$ [62] to encourage the encoded features $\mathbf{z}_i$ to stay close to their nearest entry $\mathbf{e}_{v_i}$ in the dictionary, with index $v_i$, thus keeping the range of the embeddings bounded. We define the loss as,

$$\mathcal{L}_{\text{commit}} = \frac{1}{K}\sum_{i=0}^{K-1}(\mathbf{z}_i - \text{sg}[\mathbf{e}_{v_i}])^2, \tag{2}$$

where sg stands for stop gradient operator which prevents the embedding being affected by this loss.

The full training objective for VQDIF is the combination of reconstruction loss of $\mathcal{L}_{\text{commit}}$ with weighting factor $\beta$:

$$\mathcal{L}_{\text{VQDIF}} = \frac{1}{T}\sum_{i=0}^{T-1}\text{BCE}(f(\mathbf{x}_i), o_i) + \beta\mathcal{L}_{\text{commit}}. \tag{3}$$

Here, $T$ is the size of the target set and BCE is the binary cross-entropy loss which measures the discrepancy between the predicted and the ground truth occupancy $o_i$ at target point $\mathbf{x}_i$. During training, we select the target set $\mathcal{T}_x = \{\mathbf{x}_{i=0}^{T-1}\}$ and its occupancy values $\mathcal{T}_o = \{o_{i=0}^{T-1}\}$ in a similar fashion to prior work [43].

### 3.2. Sequence generation for shape completion

We autoregressively model the distribution $p(\mathcal{S}_\mathcal{C}|\mathcal{S}_\mathcal{P})$, by predicting the distribution of the next element conditioned on the previous elements. We also factor out the tuple distribution for each element: $p(c_i, v_i) = p(c_i)p(v_i|c_i)$ . The final factored sequence distribution is as follows:

$$p(\mathcal{S}_\mathcal{C}|\mathcal{S}_\mathcal{P};\theta) = \prod_{i=0}^{K-1} p_{c_i}\cdot p_{v_i}$$
$$p_{c_i} = p(c_i|\mathbf{c}_{<i}, \mathbf{v}_{<i}, \mathcal{S}_\mathcal{P};\theta)$$
$$p_{v_i} = p(v_i|\mathbf{c}_{\le i}, \mathbf{v}_{<i}, \mathcal{S}_\mathcal{P};\theta).$$

Here, $\theta$ indicates model parameters and $p_{c_i}$ and $p_{v_i}$ are the distributions of the coordinate and the index value of the $i$-th element of $\mathcal{S}_\mathcal{C}$, conditioned on previously generated elements and the partial sequence $\mathcal{S}_\mathcal{P}$. Note that $p_{v_i}$ is also conditioned on the current coordinate $c_i$.

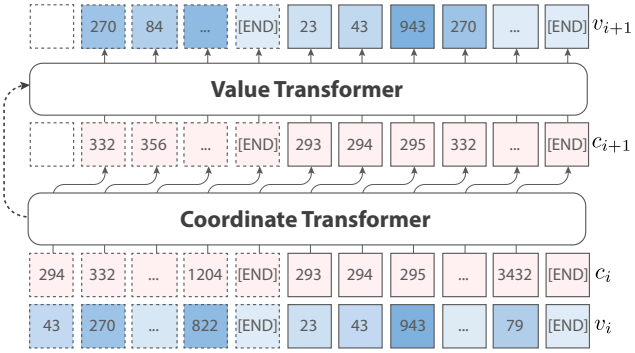



Figure 3. The architecture of the ShapeFormer. The partial sequence $\mathcal{S}_\mathcal{P}$ (dashed boxes) and the complete one $\mathcal{S}_\mathcal{C}$ (solid boxes) both appended with an end token are concatenated before sending their locations, $c_i$ (pink) and values $v_i$ (blue), to a Coordinate Transformer to predict the next location $c_{i+1}$. The Value Transformer takes both $c_{i+1}$ and the former Transformer's output embedding to predict the next value $v_{i+1}$.

Different approaches have been applied to build a transformer model that can predict tuple sequences. Instead of flattening them [58], which in our case doubles the sequence length, we stack two decoder-only transformers to predict the $p_{c_i}$ and $p_{v_i}$ respectively in a similar way to prior works [21, 46, 67], as illustrated in Fig. 3. Unlike in the image completion case [65], where the partial sequence is strictly a part of the complete sequence so that only the missing regions need to be completed. For our case, however, due to the noise or incompleteness of local observations, we would like to predict complete sequences from scratch to fix such data deficiencies. And thanks to the autoregressive structure of the decoder-only transformer, we can achieve conditioning by simply prepending $\mathcal{S}_\mathcal{P}$ before $\mathcal{S}_\mathcal{C}$ to generate complete sequences that are in coordination with the partial one. We also append an additional end token to both sequences to help learning.

**Training and inference**. The training objective of ShapeFormer is to maximize the log-likelihood given both $\mathcal{S}_\mathcal{C}$ and $\mathcal{S}_\mathcal{P}$: $\mathcal{L}_{\text{ShapeFormer}} = -\log p(\mathcal{S}_\mathcal{C}|\mathcal{S}_\mathcal{P};\theta)$. After the model

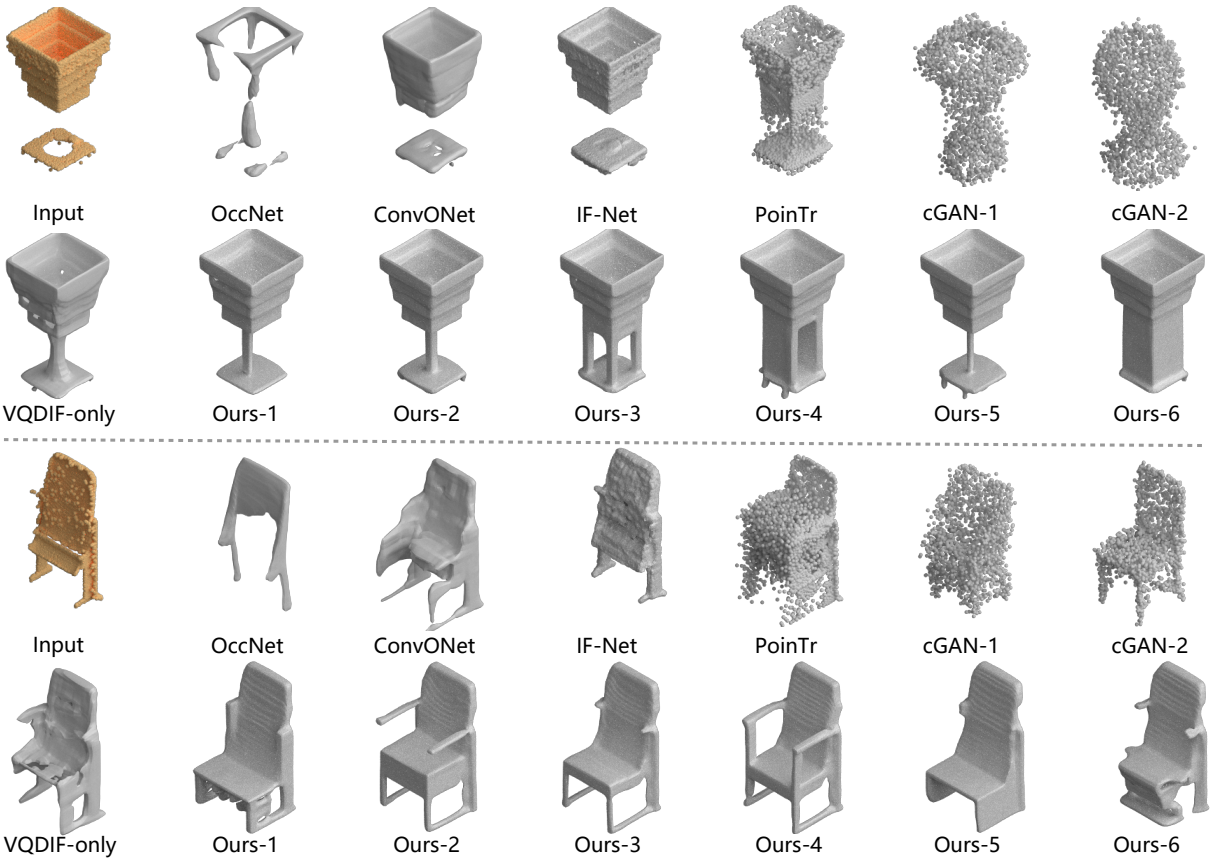



Figure 4. Visual comparison with prior shape completion methods on the ShapeNet dataset. Our method can better handle ambiguous scans and produce completions that are more faithful on both observed and unseen regions.

is trained, ShapeFormer performs shape completion by sequentially sampling the next element of the complete sequence until an end token ([END]) is encountered. Given the partial sequence, we alternatively sample the new coordinate and value index using top-p sampling [34], where only a few top choices, for which the sum of probabilities exceeds a threshold $p_n$, are kept, and the remaining probability mass is redistributed across them. Also, we mask out the invalid choices for coordinate to guarantee a monotonic row-major order.

## 4. Results and Evaluation

In this section, we demonstrate our methods outperforms prior art for shape completion from ambiguous scans and part-level incompleteness (Sec. 4.1) Then we show our approach can effectively handle a variety of shape types, scans of out-of-distribution shapes, and real-world scans from the Redwood dataset [15] (Sec. 4.2). Lastly, we show our VQDIF representation has a significantly smaller size compared with prior DIFs while achieving similar reconstruction accuracy (Sec. 4.3).

| Scan Ambiguity | Low | | | High | | |
|---|---|---|---|---|---|---|
| Method | CD↓ | F1↑ | FPD↓ | CD↓ | F1↑ | FPD↓ |
| OccNet [43] | 1.48 | 63.2 | 0.34 | **2.79** | 50.4 | 3.12 |
| ConvONet [51] | 0.81 | 72.9 | 0.23 | 3.14 | 60.4 | 2.85 |
| IF-Net [14] | 0.79 | **73.8** | 0.25 | 18.4 | 51.5 | 3.66 |
| PoinTr [69] | 0.80 | 70.1 | 0.23 | 3.11 | 59.3 | 3.29 |
| cGAN [68] | 1.33 | 62.1 | 1.36 | 3.49 | 59.3 | 2.55 |
| Ours | 0.74 | 70.3 | 0.24 | 4.72 | 60.5 | **1.45** |
| Ours* | **0.73** | 71.4 | **0.22** | 4.69 | **60.7** | 1.83 |
| VQDIF-only | 0.79 | **73.8** | 0.25 | 3.07 | 60.3 | 3.14 |

Table 1. Quantitative results on ShapeNet with different scan ambiguity. Ours: top-p=0.4 sampling, Ours*: top-p=0 sampling.

Throughout all these experiments, we use feature resolution $R = 16$ for VQDIF and set its loss balancing factor $\beta = 0.01$. We also set the vocabulary of the dictionary $\mathcal{D}$ to be $V = 4096$. We use 20 and 4 blocks for Coordinate and Value Transformers, respectively. All of these blocks have 16 heads self-attention, and the embedding dimension

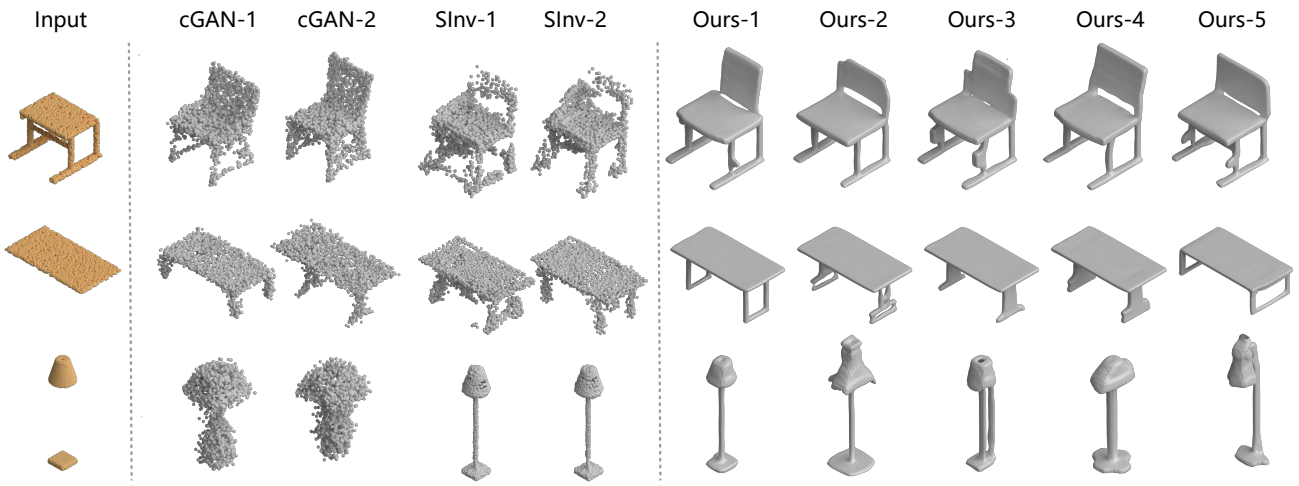



Figure 5. Visual comparison for multi-modal shape completion of Table, Chair, and Lamp categories on PartNet. We can produce diverse completions that better align with the input.

is 1024. We find that a maximum sequence length of 812 is enough for all of our experiments. We set the default probability factor $p = 0.4$ for sampling. Further implementation details such as architecture and training statistics are provided in the supplementary. *Code will be released on acceptance.*

### 4.1. Shape completion results

**Data**. We consider two datasets: 1) ShapeNet [8] for testing on partial scan and 2) PartNet [44] for testing on part-level incompleteness; we follow the same setting as in cGAN [68]. For ShapeNet, following prior works [12, 14, 43, 51], we use 13 classes of the ShapeNet with train/val/test split from 3D-R2N2 [17]. The data are processed and sampled similarly to IMNet [12] and we create partial input for training via random virtual scanning. For evaluation, we first measure the ambiguity score of a partial point cloud $\mathcal{P}$ to its complete counterpart $\mathcal{C}$ as the mean ratio of the distance of each point $x \in \mathcal{C}$ with its nearest neighbor in $\mathcal{P}$ to its distance toward furthest neighbor in $\mathcal{C}$. We uniformly sample 70 viewpoints on a sphere for each shape. Then we create two setups for the dataset according to ambiguity. The high scan ambiguity setup selects scans with the top half ambiguity score and vice versa. More details about this score are provided in the supplementary material.

**Metric**. For the low ambiguity setting, we use Chamfer $L_2$ Distance (*CD*) and F-score%1 (*F1*) [59] to measure how accurate the completion is; this is similar to the previous setup [51]. And in order to evaluate completion quality for high ambiguity setting, we following prior work [56] to use pre-trained PointNet [9] classifier as a feature extractor to compute the Fréchet Point Cloud Distance (FPD) between the set of completion results and ground truth shapes. Additionally, for the PartNet dataset, we follow cGAN [68] and use Unidirectional Hausdorff Distance (UHD) to measure faithfulness toward input, Total Mutual Difference (TMD) to measure diversity, and Minimal Matching Distance (MMD) [1] based on CD.

| **Method** | MMD ↓ | TMD ↑ | UHD ↓ | FPD ↓ |
|---|---|---|---|---|
| cGAN [68] | 1.98 | 3.05 | 3.39 | 2.95 |
| SInv. [71] | 2.14 | 0.62 | 2.32 | 3.45 |
| **Ours** | **1.32** | **3.96** | **0.98** | **1.22** |

Table 2. Quantitative comparison for multi-modal completion on PartNet between our method and prior works. The metrics are averaged across all three categories (Table, Chair, Lamp) and are scaled by $10^3$, $10^2$, $10^2$, $10^1$ respectively.

**Baselines**. We compare our model with a global DIF method OccNet [43], two local DIF methods ConvONet [51] and IF-Net [14], PoinTr [69], which adopts Transformers without autoregressive learning, and multi-modal completion method cGAN [68]. We also compare our VQDIF-only model to illustrate the necessity of ShapeFormer. We train these methods for shape completion in our dataset setting with their official implementation.

**Results on ShapeNet**. As shown in Fig. 4, methods incorporating structured local features can better preserve the input details than those that only operate on global features (OccNet [43], cGAN [68]) And deterministic methods tend to produce averaged shape since they are unable to handle multi-modality. Notice that PoinTr [69] also utilizes the

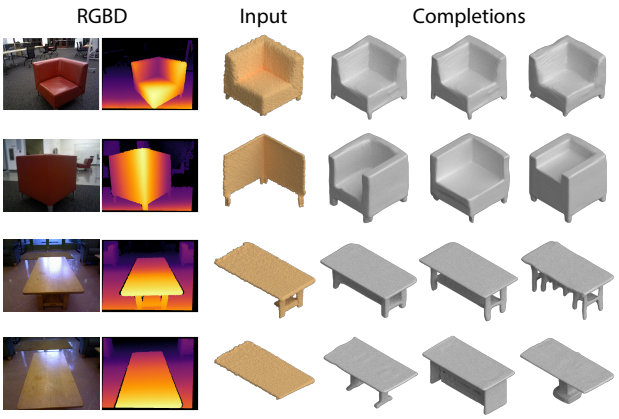



Figure 6. Shape completion results on real-world depth scan from Redwood dataset. ShapeFormer takes partial point clouds converted from depth images and produces multiple possible completions whose variation depends on the uncertainty of viewpoints.

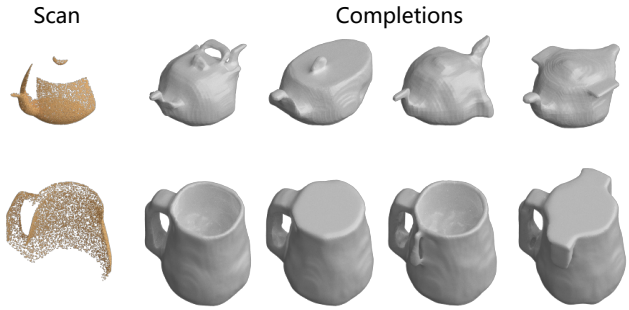



Figure 7. Shape completion results on out-of-distribution shapes. Given a scan of an unseen type of shape, ShapeFormer can produce multiple reasonable completions by generalizing the knowledge learned in the training set.

power of Transformers, but they can not alleviate this problem by adopting Transformers without generative modeling. This phenomenon is more apparent for the chair example, which has higher ambiguity. Our VQDIF-only model also fails to produce good completion in this case. Based on VQDIF, our ShapeFormer resolves ambiguity by factoring the estimation into a distribution, with each sampled shape sharp and plausible. In contrast, the multi-modal method cGAN [68] is unable to produce high-quality shapes due to their unstructured representation.

Further, we generate one completion per input with top-p sampling for quantitative evaluation. As shown in Tab. 1, our method has a much better FPD for high ambiguity scans. Notice that F1 is more reliable than CD when ambiguity is high since CD often treats plausible completions as significant errors. For low ambiguity scans, our method is also competitive toward previous state-of-the-art shape completion methods in terms of accuracy.

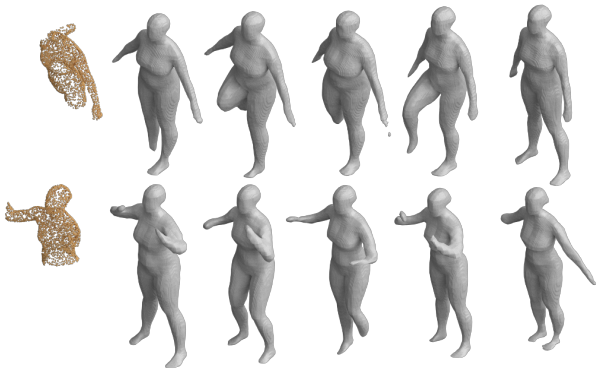


Figure 8. Given partial human body parts (left column), our method generates complete human bodies with various poses (along the rows). Notice how these shapes preserve the poses and geometries of the input scans.

**Results on PartNet**. We compare our model with cGAN and ShapeInversion [71] on PartNet. The latter method can also achieve multi-modal completion through GAN inversion. The quantitative and qualitative comparisons are shown in Tab. 2 and Fig. 5, respectively. Thanks to our structured representation, we achieve much better faithfulness (UHD) and can generate more varied (TMD) high-quality shapes (MMD and FPD) than these GAN-based methods.

### 4.2. More results

**Results on real scans**. We further investigate how our model pre-trained on ShapeNet can be applied to scans of real objects. We test our model on partial point clouds converted from RGBD scans of the Redwood 3D Scans dataset [15]. Figure 6 shows the results for a sofa and a table, both of them have two scans from different views. Notice that our model sensitively captures the uncertainty of a scan, producing a distribution of completions that are faithful to the scan and plausible in unobserved regions. We also show results for a sports car in Fig. 2.

**Results on out-of-distribution objects**. We further evaluate ShapeFormer's generalization by testing scans of unseen types of shapes on our trained model of Sec. 4.1. We pick the novel shapes from the "Famous" dataset collected by Erler et al. [22] which includes many famous geometries for testing, such as the "Utah teapot," and apply virtual scan to get the partial point cloud. Fig. 7 demonstrates our ShapeFormer can grasp general concepts such as symmetry or hollow and filled. Even the model is only trained on the 13 ShapeNet categories, without ever seeing any cups or teapot, it can still successfully produce multiple reasonable completions from the partial scan. Moreover, in the second row, we see the completions of a one side scan of a

| | Occ. | CONet. | IF. | $\text{Ours}_8$ | $\text{Ours}_{16}$ | $\text{Ours}_{32}$ |
|---|---|---|---|---|---|---|
| CD | 3.56 | 0.98 | 0.43 | 1.90 | 0.98 | 0.55 |
| F1 | 68.2 | 89.0 | 97.8 | 77.5 | 88.1 | 96.4 |
| len. | 1 | $32^3$ | $128^3$ | 57 | 217 | 889 |

Table 3. Auto-encoding results for objects in ShapeNet. len. stands for the length of the representation flattened into a sequence.

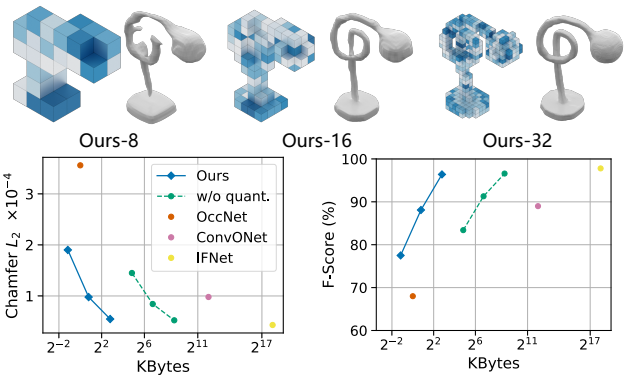



Figure 9. The relation between representation size and reconstruction accuracy. With higher feature resolution, our VQDIF achieves satisfactory accuracy while keeping a rather small byte size.

cup contain two distinct features: the cups might be solid or empty. These examples show the ShapeFormer's potential for general-purpose shape completion, where once we have it trained, we can apply it for all types of shapes.

**Results on human shapes**. In addition to CAD models, we qualitatively evaluate our completion results on scans of human shapes (D-FAUST dataset [6]) using the same setting as Niemeyer et al. [47]. This data is challenging due to the thin structures of the human body and the wide variety of poses. To simulate part level incompleteness, we randomly select a point from the complete cloud and only keep neighboring points within a ball of a fixed radius as partial input. Fig. 8 shows examples of our results. We can see that our completions keep the pose of the observed body parts and generate various possible poses for the unobserved body parts.

### 4.3. Surface reconstruction with VQDIF

Our final experiment evaluates the representation size and reconstruction accuracy of VQDIF. We compare VQDIF of different feature resolutions ($\text{Ours}_8$, $\text{Ours}_{16}$, $\text{Ours}_{32}$) with OccNet, ConvONet, IF-Net, which are retrained to auto-encode the complete shape with their released implementations. As shown in Fig. 9, $\text{Ours}_{32}$ achieves similar accuracy to state-of-the-art local implicit approach IF-Net while being significantly smaller in size

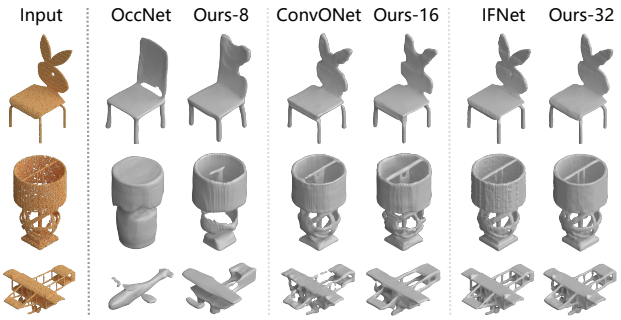



Figure 10. Results for auto-encoding complete shapes. Our VQDIF in different feature resolutions achieves better or similar results compared to the prior DIF methods.

thanks to the sparse and discrete features of VQDIF. The minimum receptive field of our encoder keeps the feature as local as possible, which greatly reduces the feature amount. Then the multi-dimensional feature vectors are quantized and can be referred to using a single integer index, which further reduces the size. The accuracy loss is only salient for lower feature resolution, as seen in the *w/o quant.* comparison, where we train VQDIF without vector quantization. These together allow transformers to learn the relationships between local shape parts effectively. We adopt $\text{Ours}_{16}$ for ShapeFormer since it only has an average sequence length of 217 (see Tab. 3) and its accuracy is already comparable with ConvONet (see Fig. 10).

## 5. Conclusions

We have presented ShapeFormer, a transformer-based architecture that learns a conditional distribution of completions, from which multiple plausible completed shapes can be sampled. By explicitly modeling the underlying distribution, our method produces sharp output instead of regressing to the mean producing an averaged blurred out result. To facilitate generative learning for 3D shape, we propose a new 3D representation VQDIF that can significantly compress the shapes into short sequences of sparse, discrete local features, which in turn enables producing better results, both in terms of quality and diversity, than previous generative methods.

The major factor limiting our method to be applied in fields like robotics is the sampling speed, which is currently 20 seconds per generated complete shape. In the future, we would also like to explore utilizing a more efficient attention mechanism to allow Transformers to learn VQDIF with smaller size, producing even higher quality completions. Moreover, the current method is generic, leveraging advances in language models. More research is required to include geometric or physical reasoning in the process to better deal with ambiguities.

## References


[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J. Guibas. Learning representations and generative models for 3D point clouds. In *International conference on machine learning*, 2017. 2, 6

[2] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers, 2021. 2

[3] Yoshua Bengio and Samy Bengio. Modeling high-dimensional discrete data with multi-layer neural networks. *Advances in Neural Information Processing Systems*, 12:400–406, 2000. 3

[4] Matthew Berger, Andrea Tagliasacchi, Lee M Seversky, Pierre Alliez, Gael Guennebaud, Joshua A Levine, Andrei Sharf, and Claudio T Silva. A survey of surface reconstruction from point clouds. In *Computer Graphics Forum*, volume 36, pages 301–329, 2017. 1, 2

[5] Fausto Bernardini, Joshua Mittleman, Holly Rushmeier, Claudio Silva, and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics*, 5(4):349–359, 1999. 2

[6] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 8

[7] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 3

[8] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository, 2015. 6

[9] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. 6, 4

[10] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020. 2, 3

[11] Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. Pixelsnail: An improved autoregressive generative model. In *International Conference on Machine Learning*, pages 864–872. PMLR, 2018. 3

[12] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 5939–5948, 2019. 1, 2, 6

[13] Zhang Chen, Yinda Zhang, Kyle Genova, Sean Fanello, Sofien Bouaziz, Christian Hane, Ruofei Du, Cem Keskin, Thomas Funkhouser, and Danhang Tang. Multiresolution deep implicit functions for 3d shape representation. In *Proc. Int. Conf. on Computer Vision*, pages 13087–13096, 2021. 1

[14] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*. IEEE, jun 2020. 1, 2, 5, 6, 8

[15] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans, 2016. 5, 7

[16] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers, 2021. 8

[17] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 2, 6

[18] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016. 4

[19] Angela Dai, Charles Ruizhongtai Qi, and Matthias NieBner. Shape completion using 3D-Encoder-Predictor CNNs and shape synthesis. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, 2017. 2

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 3

[21] Sander Dieleman, Charlie Nash, Jesse Engel, and Karen Simonyan. Variable-rate discrete representation learning. *arXiv preprint arXiv:2103.06089*, 2021. 2, 3, 4

[22] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, N. Mitra, and M. Wimmer. Points2surf learning implicit surfaces from point clouds. In *Proc. Euro. Conf. on Computer Vision*, 2020. 1, 2, 7, 4

[23] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis, 2020. 2, 3, 4

[24] Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3D object reconstruction from a single image. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, 2017. 2

[25] J. Fauw, S. Dieleman, and K. Simonyan. Hierarchical autoregressive image models with auxiliary decoders. *ArXiv*, abs/1903.04933, 2019. 2

[26] Yasutaka Furukawa and Carlos Hernández. *Multi-view stereo: A tutorial*, volume 9. 2013. 2

[27] Kyle Genova, F. Cole, Avneesh Sud, Aaron Sarna, and T. Funkhouser. Local deep implicit functions for 3d shape. *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 4856–4865, 2020. 1, 2

[28] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In *International conference on machine learning*, pages 881–889. PMLR, 2015. 3

[29] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Ap-

proach to Learning 3D Surface Generation. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, 2018. 2

[30] Xian-Feng Han, Hamid Laga, and Mohammed Bennamoun. Image-based 3d object reconstruction: State-of-the-art and trends in the deep learning era. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1578–1604, 2019. 2

[31] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. *2017 International Conference on 3D Vision (3DV)*, pages 412–420, 2017. 2

[32] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. 3

[33] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers, 2019. 8

[34] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *Proc. Int. Conf. on Learning Representations*, 2019. 5

[35] Zhiyang Huang, Nathan Carr, and Tao Ju. Variational implicit point set surfaces. *ACM Transactions on Graphics (TOG)*, 38(4):1–13, 2019. 2

[36] Vivek Jayaram and John Thickstun. Parallel and flexible sampling from autoregressive models via langevin dynamics, 2021. 8

[37] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1

[38] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proc. Eurographics Symp. on Geometry Processing*, volume 7, 2006. 2

[39] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. on Graphics*, 32:29:1–29:13, 2013. 2

[40] Or Litany, Alex Bronstein, Michael Bronstein, and Ameesh Makadia. Deformable shape completion with graph convolutional autoencoders, 2018. 2

[41] Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences, 2018. 2

[42] Shi-Lin Liu, Hao-Xiang Guo, Hao Pan, Peng-Shuai Wang, Xin Tong, and Yang Liu. Deep implicit moving least-squares functions for 3d reconstruction. *arXiv preprint arXiv:2103.12266*, 2021. 2

[43] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3Dreconstruction in function space. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 4460–4470, 2019. 1, 2, 4, 5, 6

[44] Kaichun Mo, Shilin Zhu, Angel X. Chang, L. Yi, Subarna Tripathi, L. Guibas, and H. Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 909–918, 2019. 6

[45] Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*, pages 7220–7229. PMLR, 2020. 3

[46] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021. 3, 4

[47] M. Niemeyer, Lars M. Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. *ICCV*, pages 5378–5388, 2019. 8

[48] Aäron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. In *Proc. Conf. on Neural Information Processing Systems*, pages 4797–4805, 2016. 3

[49] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 165–174, 2019. 1, 2

[50] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018. 3

[51] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proc. Euro. Conf. on Computer Vision*, 2020. 1, 2, 4, 5, 6, 8

[52] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 3, 4

[53] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021. 3

[54] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *arXiv preprint arXiv:1906.00446*, 2019. 2, 3

[55] Jason Rock, Tanmay Gupta, Justin Thorsen, JunYoung Gwak, Daeyun Shin, and Derek Hoiem. Completing 3d object shape from one depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2484–2493, 2015. 2

[56] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions, 2019. 6

[57] David Stutz and Andreas Geiger. Learning 3d shape completion from laser scan data with weak supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[58] Yongbin Sun, Yue Wang, Ziwei Liu, Joshua Siegel, and Sanjay Sarma. Pointgrow: Autoregressively learned point cloud generation with self-attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 61–70, 2020. 2, 3, 4

[59] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do

single-view 3D reconstruction networks learn? In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, pages 3405–3414, 2019. 6, 2

[60] Lyne P. Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *Proc. IEEE Conf. on Computer Vision & Pattern Recognition*, 2019. 2

[61] Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. Neural autoregressive distribution estimation. *The Journal of Machine Learning Research*, 17(1):7184–7220, 2016. 3

[62] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proc. Conf. on Neural Information Processing Systems*, 2017. 3, 4

[63] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756. PMLR, 2016. 3

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 3

[65] Ziyu Wan, Jingbo Zhang, Dongdong Chen, and Jing Liao. High-fidelity pluralistic image completion with transformers. *arXiv preprint arXiv:2103.14031*, 2021. 2, 3, 4, 8

[66] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images, 2018. 2

[67] Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. Sceneformer: Indoor scene generation with transformers. *arXiv preprint arXiv:2012.09793*, 2020. 3, 4

[68] Rundi Wu, Xuelin Chen, Yixin Zhuang, and Baoquan Chen. Multimodal shape completion via conditional generative adversarial networks. In *Proc. Euro. Conf. on Computer Vision*, August 2020. 2, 5, 6, 7

[69] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *ICCV*, 2021. 2, 5, 6, 4, 8

[70] 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, 2018. 2

[71] 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 *CVPR*, 2021. 6, 7

[72] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proc. Int. Conf. on Computer Vision*, pages 5826–5835, October 2021. 2

# ShapeFormer: Transformer-based Shape Completion via Sparse Representation

## Supplementary Material


### Abstract

*In this supplementary document, we first give a detailed description of the ambiguity measure, model architectures, and training/testing statistics in Appendix A. Then we show more visual comparisons between our method and previous methods for scans of both **high and low ambiguity** in Appendix B. Lastly, we will give more analysis on our method in Appendix C, such as a discussion of limitations. The code of our model is also included in the supplementary material.*


## A. Implementation Details

### A.1. Ambiguity measure for partial point cloud

The ambiguity for a partial point cloud measures the variety of its potential complete shapes. However, the direct measurement for ambiguity is difficult, if not impossible. In contrast, the incompleteness of a partial cloud toward its complete shape is relatively easy to compute. Although it can not fully reflect ambiguity (e.g., a top scan of a table as incomplete as a bottom scan could have a much greater ambiguity), the ambiguity is still strongly correlated.

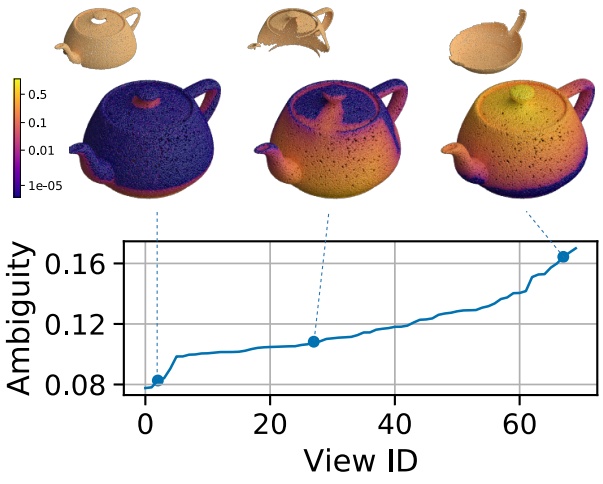
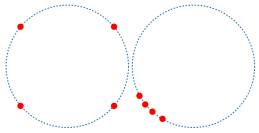



Figure 11. Viewing direction greatly influences the scan ambiguity. Our proposed scores for 70 scans of a teapot are shown in sorted order, with examples marked with their position on the curve. The example contains the scans (in gold insets) and complete shape color-coded scores for each point in it.



Hence, we seek to find a metric on the incompleteness of such a point cloud to indicate its ambiguity. Intuitively, we could use metrics like F-score [59] to measure the ratio of the approximate partial surface area toward the complete area. But as indicated in the inset figure, such measures will fail to differentiate the coverage difference of the partial cloud (red dots) to the complete one (in blue). Instead, we propose to use a metric based on Chamfer-$L_2$, which goes larger as the partial point cloud misses more global structure. Since the partial to complete distance is always negligible, we can only calculate the complete to partial distance. And to compare the ambiguity of scans on different shapes, we normalize the distance of a point according to its farthest distance in the complete shape. More specifically, we define the metric $Amb$ evaluating the ambiguity of scan $\mathcal{C}$ given the complete point cloud as $\mathcal{B}$ as:

$$Amb(B, C) = \frac{1}{B}\Sigma_{x\in\mathcal{B}}\frac{\min_{y\in\mathcal{C}}||x-y||}{\max_{x'\in B}||x-x'||}, \quad (4)$$

Where $B$ is the number of points in the complete cloud.

We sample 70 views for each shape, 64 of which are evenly sampled from the view sphere (via Fibonacci sampling), and the rest are the six orthogonal views. Then we sort these views according to the score. In Fig. 11, we use a teapot as an example to show the score distribution of these 70 scans. For scans with low ambiguity scores, the underlying shape's global structure is either captured or is clearly indicated by the captured shape salient features. For example, the scan covering the teapot's mouth, handle, and body can be completed easily. However, it would be more difficult to infer the complete shape when the score is high since it may have different global structures, and a single explanation is not satisfactory. As shown in the main paper, our method can better handle such scans than existing shape completion methods.

### A.2. Architectures

We show the detailed architecture of VQDIF and ShapeFormer in Figs. 12 and 13, respectively. and the parameters of their sub-modules are listed in Tab. 4

**VQDIF**. As shown in Figure 12, VQDIF is an encoder-decoder architecture, where the encoder maps an input point cloud to a discrete sequence representation $\mathcal{S}$, while the decoder maps such a sequence to a deep implicit function $f(\mathbf{x})$. Unlike the main paper's completion pipeline, both the

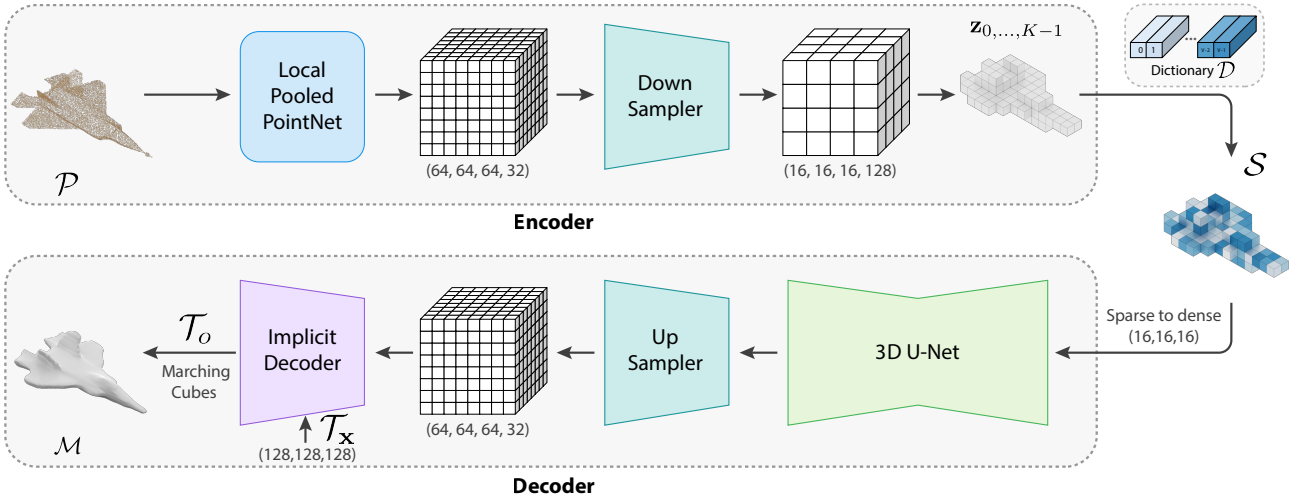



Figure 12. The architecture of VQDIF. The complete point cloud $\mathcal{P}$ is encoded to a feature grid and down-sampled into a lower resolution one. Its non-empty features are then flattened and quantized to form the VQDIF sequence which is then projected back to a feature grid, up-sampled and sent to an implicit decoder, from which the occupancy grid $\mathcal{T}_o$ of probes $\mathcal{T}_\mathbf{x}$ and the reconstruction $\mathcal{M}$ can be obtained.

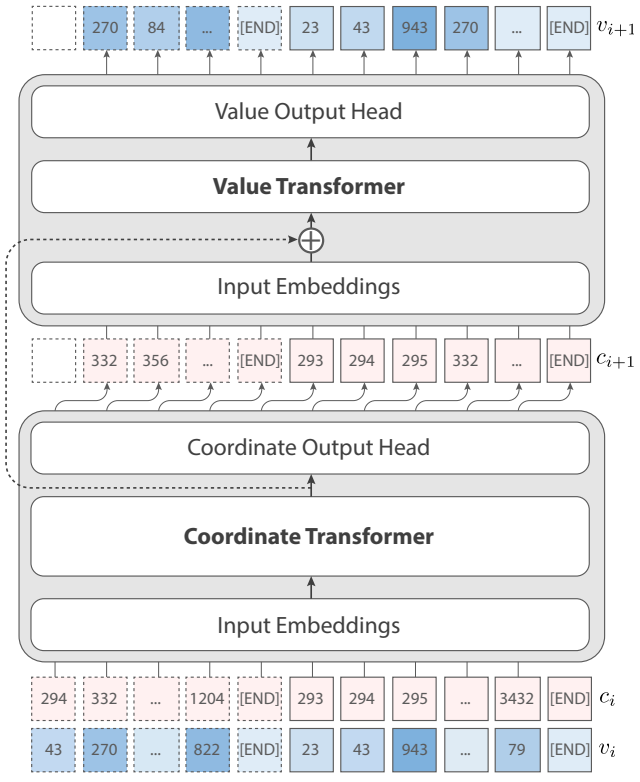



Figure 13. An extended view of ShapeFormer. Different from the figure in the main paper, we show the inside of each Transformer module. The input embeddings are obtained by additively mixing the location and value embeddings. And the output head converts the output embedding into categorical distributions.

| Layer Name | Notes | Input Size |
|---|---|---|
| **VQDIF** | | |
| Local Pooled Pointnet | | $N \times 3$ |
| Downsampler | | |
| ConvLayer | k2s2p0 | $64 \times 64 \times 64 \times 32$ |
| ConvLayer | k1s1p0 | $32 \times 32 \times 32 \times 64$ |
| ConvLayer | k2s2p0 | $64 \times 64 \times 64 \times 32$ |
| ConvLayer | k1s1p0 | $32 \times 32 \times 32 \times 64$ |
| Quantizer | | $16 \times 16 \times 16 \times 128$ |
| UNet3D | | $16 \times 16 \times 16 \times 128$ |
| Upsampler | | $16 \times 16 \times 16 \times 128$ |
| Scaling | nearest mode | $16 \times 16 \times 16 \times 128$ |
| ConvLayer | k3s1p1 | $32 \times 32 \times 32 \times 128$ |
| ConvLayer | k3s1p1 | $32 \times 32 \times 32 \times 64$ |
| Scaling | nearest mode | $32 \times 32 \times 32 \times 64$ |
| ConvLayer | k3s1p1 | $64 \times 64 \times 64 \times 64$ |
| ConvLayer | k3s1p1 | $64 \times 64 \times 64 \times 32$ |
| Upsampler Output | | $64 \times 64 \times 64 \times 32$ |
| Implicit Decoder | | $128^3 \times 3$ |
| Implicit Decoder Output | | $128^3 \times 1$ |
| **ShapeFormer** | | |
| Embedding Blocks | #4M | $K \times 2$ |
| Coordinate Transformer Blocks $\times 20$ | #251M | $K \times 1024$ |
| Coordinate Output Heads | #4M | $K \times 4097$ |
| Embedding Blocks | #4M | $K \times 2$ |
| Value Transformer Blocks $\times 4$ | #50M | $K \times 1024$ |
| Value Output Heads | #4M | $K \times 4097$ |
| **Total params** | #340M | |
| **Trainable params** | #323M | |

Table 4. The detailed architecture information of our method. $N$ is the point size. For both VQDIF and ShapeFormer, we list the input size of their components. For convolutional neural networks, the "k", "s", "p" stands for kernel size, stride, and padding, respectively. Also "ConvLayer" denotes the composition of CNN + ReLU + GroupNorm. We also list the number of parameters for each component and indicate them with #. The sequence length is denoted by K, with a maximum of 812.

encoder and decoder only take complete input during training. The input to the encoder is a point cloud $\mathcal{P} \in \mathbb{R}^{N\times 3}$ representing the dense sampling of a shape or its partial observation. During the training phase, we use complete dense clouds with $N = 32768$ points to train VQDIF to capture local geometric details in the input. At test time, we use the trained encoder to directly encode partial point clouds, which may be sparse or dense.

The encoder first processes the input cloud with a local pooled PointNet [9] to obtain a feature grid. Similar to prior work [51], the local pooled PointNet aggregates features within a grid cell in contrast to the original PointNet, where all point features are pooled together to obtain a global feature. Specifically, we use a grid of resolution 64 with a feature size of 32.

Next, to reduce the number of local features, the high-resolution feature grid is down-sampled to lower resolution $R$, using several consecutive strided convolution blocks. As shown in Tab. 4, the parameters of these blocks are carefully set to have the least receptive field since a large receptive field lets each grid feature cover a larger region, reducing the sparsity of the representation. We can then extract the non-empty features by directly masking the encoded feature grid with the voxelized input point cloud (resolution $R$) thanks to the minimum receptive field. After flattening and quantizing the features (see the main paper), we get the 2-tuple sequence representation directly sent to the decoder. Note that we also save the "empty" feature to project the sequence back to the feature grid in the decoder.

The decoder consists of a 3D U-Net [18], an up-sampler, and an implicit decoder. It first projects the quantized sparse sequence back to a 3D feature grid, which serves as the input for the 3D U-Net. In contrast to the encoder, the decoder is designed to have a large receptive field. This is because, in order for the implicit decoder to infer whether a probe lies inside or outside of the shape, we need global knowledge. This is in alignment with prior works [22, 51]. More specifically, we use a 3-step U-Net to increase the receptive field, which integrates both local and global information. The up-sampler has the same number of scaling stages as the down-sampler, but it has a larger receptive field by design. Lastly, similarly to prior work [51], the implicit decoder consists of multiple ResNet blocks. It takes querying probe points $\mathcal{T}_\mathbf{x}$ and predicts their occupancy probability $\mathcal{T}_o$.

**ShapeFormer**. In Fig. 13, we show the detailed architecture of ShapeFormer. The input to the ShapeFormer consists of the concatenated sequence of $\mathcal{S}_\mathcal{P}$ and $\mathcal{S}_\mathcal{C}$. Since these sequences both have variable lengths, we append an end-token ([END]) to each sequence to indicate when the sequence terminates. Next, as in prior works [21, 46], all these indices are turned into learnable embeddings and are additively combined as the input embedding for ShapeFormer.

The main components of ShapeFormer are two causally-masked transformers, which consist of multiple decoder-only transformer blocks [52]. The first transformer learns to predict the coordinate of the next tuple, conditioned on previous tuples, while the second one learns to predict the value of the next element conditioned on previous tuples and the (predicted) coordinate index of the next element. Thus, the output feature of the first transformer is additively mixed with the input embedding of the second transformer delivering the encoded sequence information.

Each transformer is followed by an output head, which converts the feature produced by the transformer into a categorical distribution of the next sequence element. Both output heads consist of two fully connected layers, followed by a softmax layer to produce categorical conditional distributions for each of the sequence elements: $\{(p_{c_i}, p_{v_i})\}_{i=1}^{K}$. Note that this essentially shifts the complete sequence to the right by one element. For training, we also empirically find randomly masking out the partial sequence will improve generalization.

### A.3. Details on training and sampling

We use Adam optimizer for training both VQDIF and ShapeFormer, and we set the learning rate as $1e-4$ for VQDIF and $1e-5$ for ShapeFormer. We use step decay for VQDIF with step size equal to 10 and $\beta = .9$ and do not apply learning rate scheduling for ShapeFormer. We train our network on a deep learning server with Intel Xeon CPU E5-2680 v4 CPU*56 and 256GB memory with 10 Nvidia Quadro P6000 graphics cards with a GPU memory size of 24GB. It takes 30 hours for our model to converge on our virtual scan dataset and 8 hours on the PartNet dataset. For D-Faust, the converging time is 16 hours. For sampling, we can obtain a single sample sequence in roughly 20 seconds, and we can also sample 24 sequences in parallel in 5 minutes.

## B. More comparisons

We show more visual comparisons between our method and prior state-of-the-art methods in Figs. 14 to 16. Figs. 14 and 15 illustrates results on high-ambiguity scans, In these examples, we can see the averaging effect of the deterministic methods (See the scattering effect in ambiguous regions of the completions of PoinTr [69]). Our method produces significantly better results in terms of quality and diversity.

Also, we demonstrate our method can also achieve competitive accuracy for low-ambiguity scans in Fig. 16. Since there is limited ambiguity for such scans and the goal is to achieve accuracy toward ground truth, we put the ground truth in the first row and only sample 1 completion for each of our sampling strategies (Ours: top-.4 sampling, Ours*:

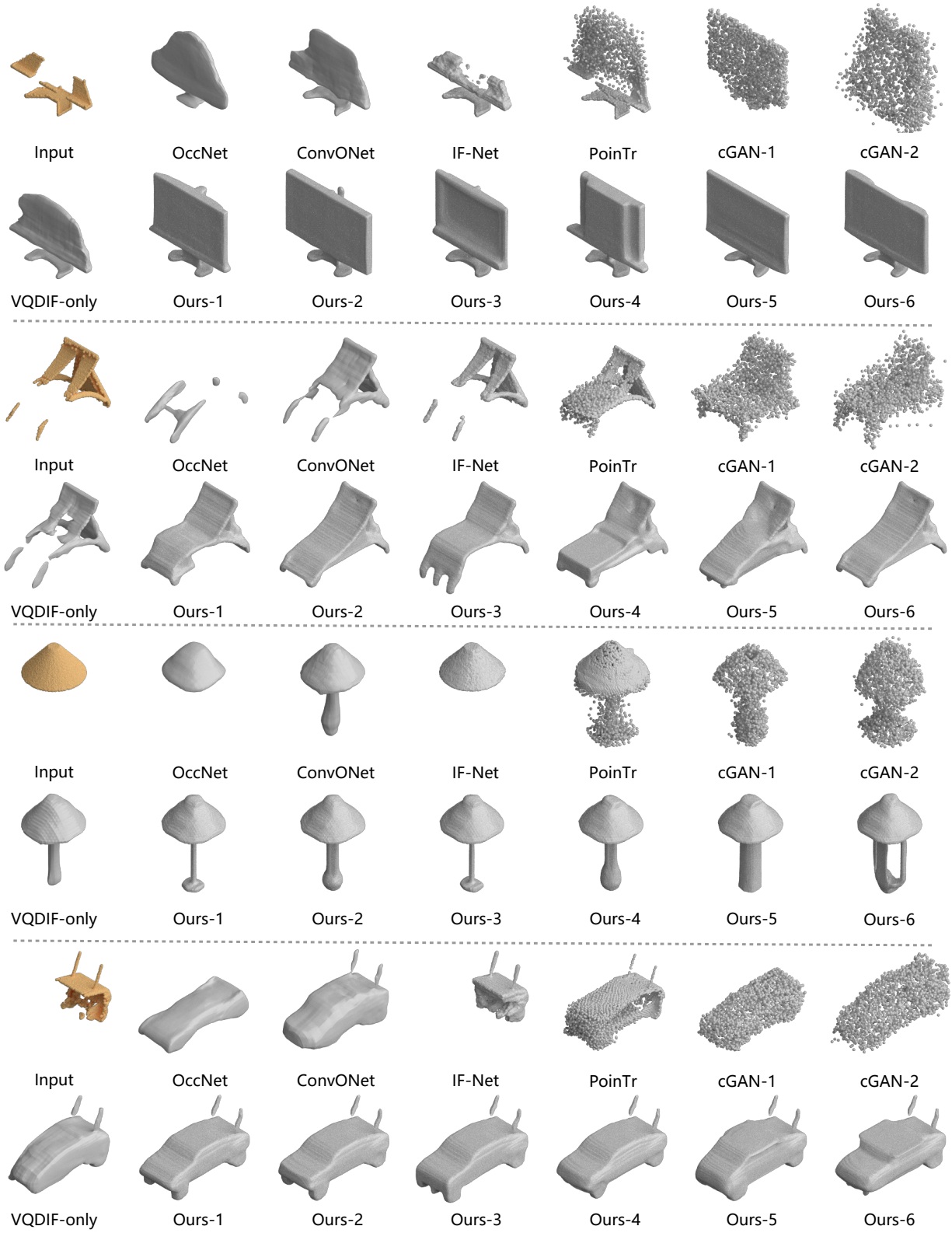



Figure 14. More comparisons on high ambiguity scans of ShapeNet objects.

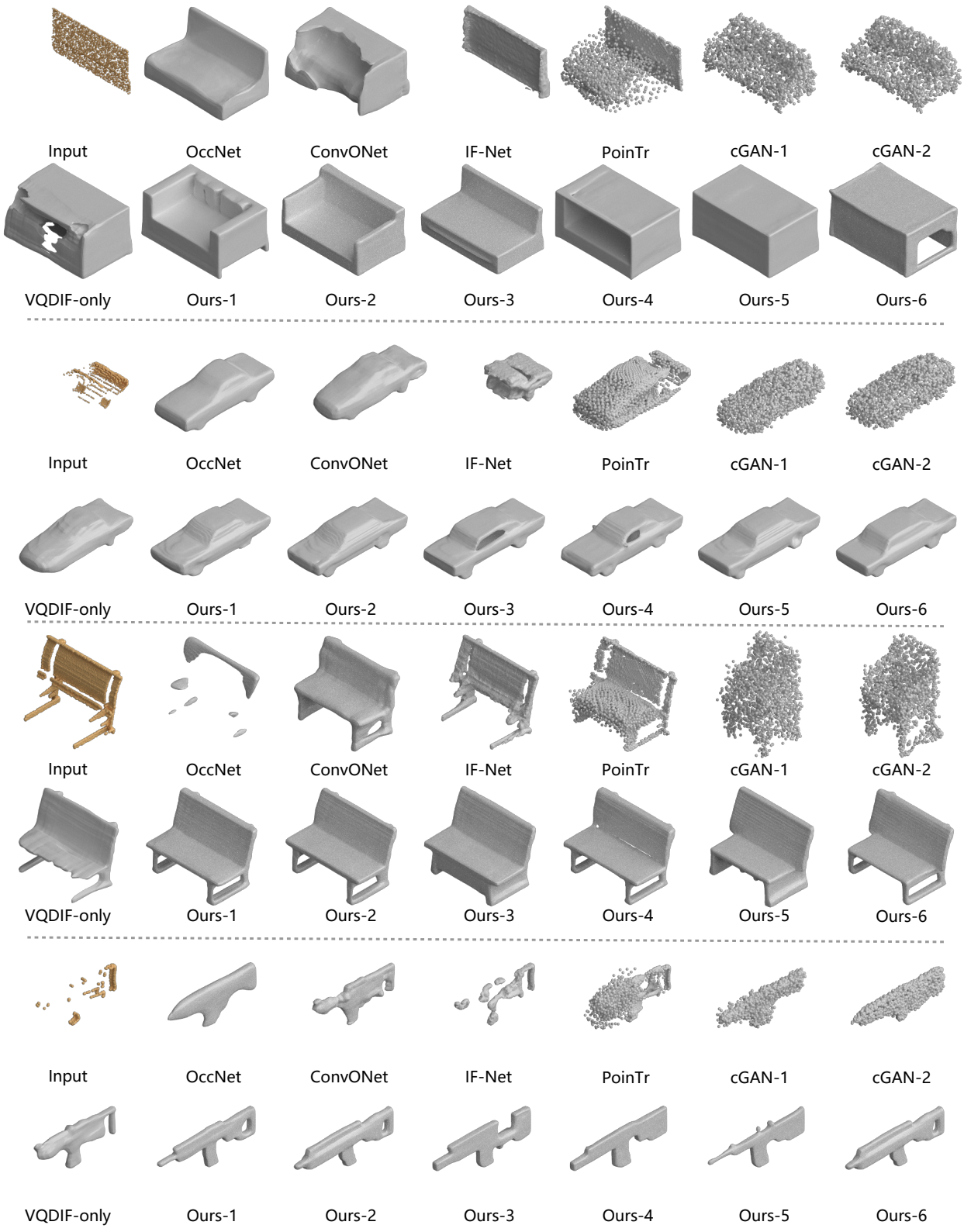



Figure 15. More comparisons on high ambiguity scans of ShapeNet objects.

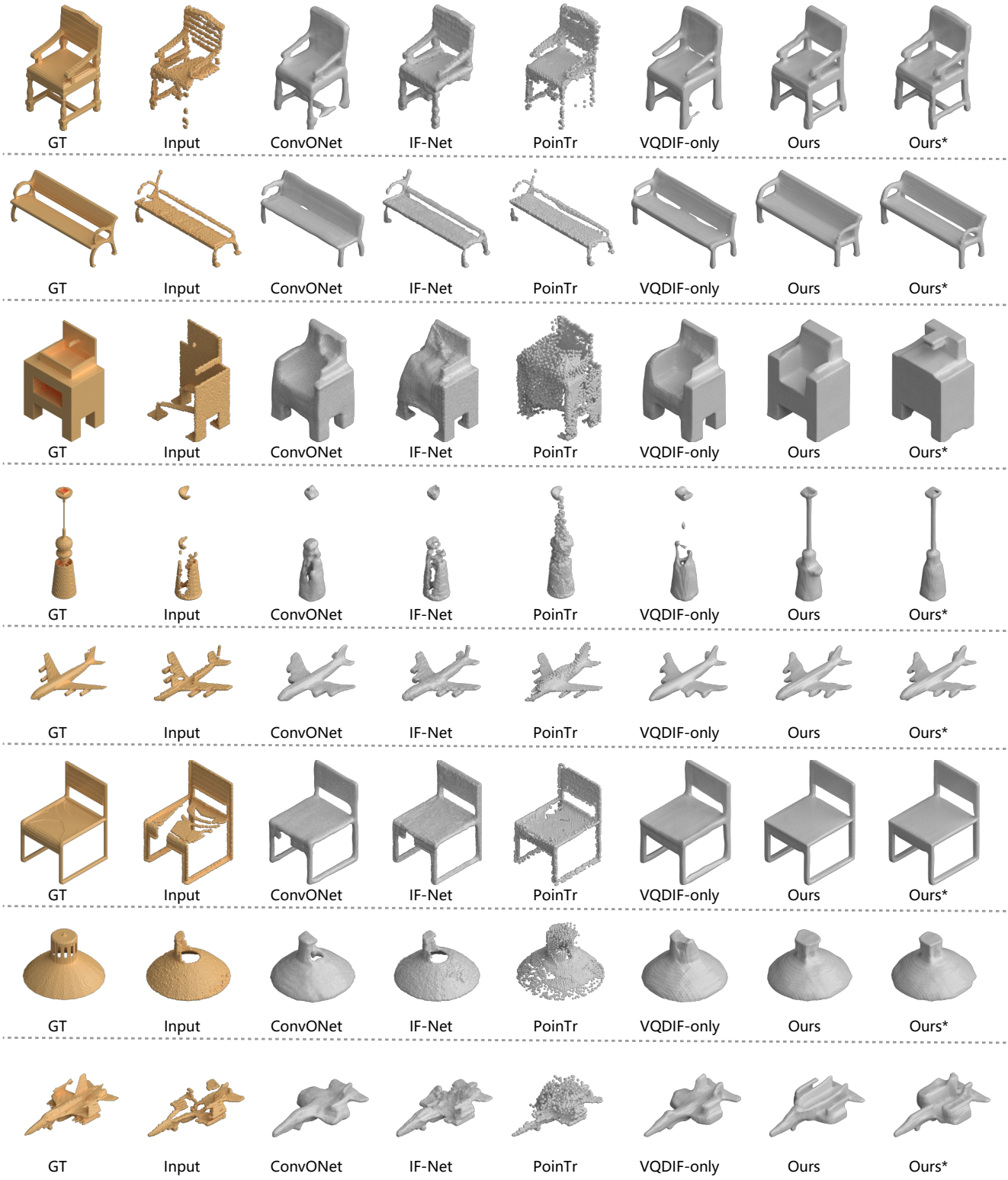



Figure 16. More comparisons on low ambiguity scans of ShapeNet objects. Ours=top-.4 sampling, Ours*=top-.0 sampling (best sampling).

top-.0, e.g., best sampling). Also, we only compare state-of-the-art deterministic methods: ConvONet [51], IF-Net [14], and PoinTr [69] in these examples. As we can see, even the scans cover most areas of the ground truth shape; prior works can still produce unsatisfactory results for unseen regions. In contrast, our method can always produce more accurate, high-quality completions. Moreover, since Ours* always picks the coordinate and value indices with the highest probability, it often produces slightly more accurate shapes.

## C. More analysis

**Discussion of Limitation**. ShapeFormer inherits the typical limitations of transformer-based autoregressive models. Mainly, the representation length cannot be too long, and thus the method currently can only use VQDIF with $R = 16$, which may fail to complete and reconstruct shapes with intricate structures; an example is shown in Figure 17. Another related limitation is the sampling speed, which prevents interactive applications.

Figure 17. An example of a shape completion failure case of ShapeFormer. The intricate details present in the input (second from left) are not preserved in the completions (gray shapes). The leftmost image shows the ground truth shape.

There are two research avenues to alleviate these problems: (i) Investigating more efficient attention mechanisms to reduce the transformer's quadratic complexity in the sequence length $K$ to $O(K\sqrt{K})$ [33] or even $O(K)$ [16]. (ii) Designing an adaptive quantization scheme for the point clouds, which enables Transformers to focus dependencies on a lower local level while using higher-level features for faraway regions. (iii) Adopt advanced sampling techniques for autoregressive models such as parallel sampling [36].

Moreover, since we generate sequences of complete shapes from scratch, our results may slightly alter the input geometry to overcome the potential sparsity and noise. Besides using higher resolution quantized features to obtain more accurate generation, another possible improvement to this issue is to include high-resolution features of the input in the decoding procedure as in a recent image inpainting technique [65].