Multi-view 3D Reconstruction with Transformer

Dan Wang, Xinrui Cui†, Xun Chen, Zhengxia Zou, Tianyang Shi,
Septimiu Salcudean *Fellow, IEEE*, Z. Jane Wang *Fellow, IEEE*, and Rabab Ward *Fellow, IEEE*

***Abstract*—Deep CNN-based methods have so far achieved the state of the art results in multi-view 3D object reconstruction. Despite the considerable progress, the two core modules of these methods - multi-view feature extraction and fusion, are usually investigated separately, and the object relations in different views are rarely explored. In this paper, inspired by the recent great success in self-attention-based Transformer models, we reformulate the multi-view 3D reconstruction as a sequence-to-sequence prediction problem and propose a new framework named 3D Volume Transformer (VolT) for such a task. Unlike previous CNN-based methods using a separate design, we unify the feature extraction and view fusion in a single Transformer network. A natural advantage of our design lies in the exploration of view-to-view relationships using self-attention among multiple unordered inputs. On ShapeNet - a large-scale 3D reconstruction benchmark dataset, our method achieves a new state-of-the-art accuracy in multi-view reconstruction with fewer parameters (70% less) than other CNN-based methods. Experimental results also suggest the strong scaling capability of our method. Our code will be made publicly available.**

I. Introduction

Learning 3D object representation from multi-view images is a fundamental and challenging problem in 3D modeling, virtual reality, and computer animation. Recently, deep learning approaches have greatly promoted the research in multi-view 3D reconstruction problem, where the deep convolutional neural network (CNN) based approaches have so far achieved state-of-the-art accuracy in this task [1], [2], [3].

To learn effective 3D representation from multiple input views, most recent CNN-based approaches follow the design principle of divide-and-conquer, where a common practice is to introduce a CNN for feature extraction and fusion module for integrating the features or reconstruction results from multiple views. Despite the strong connection between the two modules, their methodology designs are investigated separately. Also, during the CNN feature extraction stage, the object relations in different views are rarely explored. Although some recent approaches have introduced recurrent neural network (RNN) to learn object relationships between different views [4], [5], such a design lacks computational efficiency and the input views to the RNN model are permutation-sensitive [6], which is difficult to be compatible with a set of unordered input views. It is also shown in recent researches that CNN-based reconstruction methods may suffer from the model scaling problem. For example, when the number of model inputs exceeds a certain scale (e.g. 4), the accuracy of the model will be saturated, showing the difficulty of learning complementary knowledge through a large set of independent CNN feature extraction units [2], [3].

Considering the above challenges, we propose a new framework named "3D Volume Transformer (VolT)", and explore the potential of recent great success of self-attention based language model for the multi-view 3D object reconstruction task. We reformulate the multi-view 3D reconstruction as a sequence-to-sequence prediction problem and unify the feature extraction and view fusion in a single Transformer network. On one hand, in multi-view 3D reconstruction, from a particular view, we can only see part of the underlying 3D structure. On the other hand, in a Transformer model, the self-attention mechanism has recently shown its great power on learning complex semantic abstractions within an arbitrary number of input tokens [7], [8] and is naturally suitable for exploring the view-to-view relationships of a 3D object's different semantic parts. Given all this, the structure of Transformer [9], [10] becomes a natural and attractive solution for the multi-view 3D reconstruction.

Our Transformer-based framework contains a 2D-view Transformer encoder and a 3D-volume Transformer decoder, as presented in Figure 1. In the proposed framework, the 2D-view encoder encodes and fuses the multiple 2D-view information by exploring their "2D-view × 2D-view" relationships of the different inputs. The 3D-volume decoder decodes and fuses the multi-view features from the encoder and generate a 3D probabilistic voxel output for each of the spatial query tokens. The attention layers in the decoder learns "2D-view × 3D-volume" relationships between each of the output voxel grids and input views. Meanwhile, volume attention layers in the decoder further learn "3D-volume × 3D-volume" relationships by exploiting correlations amongst different 3D locations. By using the above unified design, the "2D-view × 2D-view", "3D-volume × 3D-volume", and "2D-view × 3D-volume" relationships can be jointly explored by multiple attention layers in both the encoder and decoder networks.

On basis of the above encoder-decoder structure design, we further investigate the "attention uniformity" problem in a Transformer model and propose a effective solution for enhancing the effectiveness of a Transformer model in the multi-view reconstruction task. In Transformers, self-attention possesses a solid inductive bias towards "token uniformity" [11], which encourages feature representations of input tokens converge. However, this convergence may further cause the problem of "attention uniformity" in deeper layers, which makes a Transformer model loses expressive power speedily with respect to network depth [11]. We show that in the multi-view 3D reconstruction task, this problem is particu-

† Corresponding author: Xinrui Cui (xinruic@ece.ubc.ca).
D. Wang, X. Cui, Septimiu Salcudean, Z. Jane Wang, and Rabab Ward are with University of British Columbia, Canada. e-mail: {danw, xinruic, Tims, zjanew, rababw}@ece.ubc.ca. X. Chen is with University of Science and Technology of China. e-mail: xunchen@ustc.edu.cn. Z. Zou is with University of Michigan, Ann Arbor. e-mail: zzhengxi@umich.edu. T. Shi is with NetEase Fuxi AI Lab. e-mail: shitianyang@corp.netease.com.

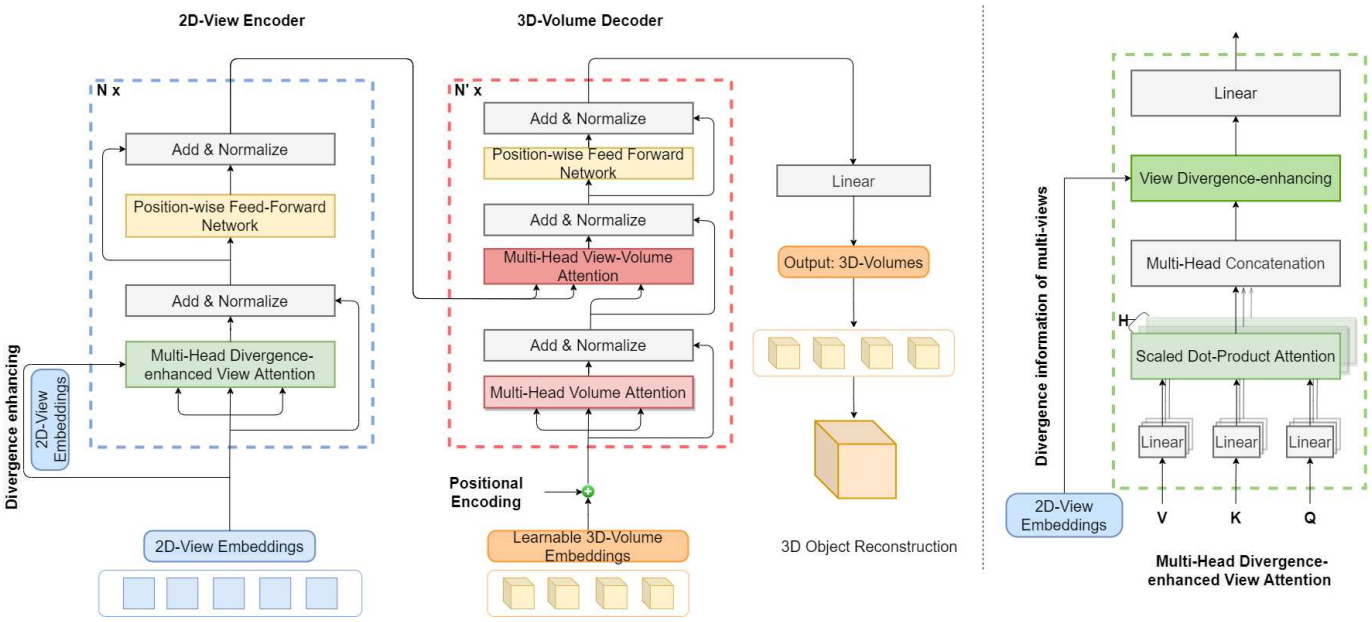

Fig. 1. Illustration of EVolT for Multi-view 3D Object Reconstruction (left). The proposed view-divergence enhancing function in our EVolT (right).

larly prominent and will limit the Transformer's capability of exploring and abstracting multi-view associations at a deeper level. In our experiment, we found that when directly adopting the vanilla Transformer [9] as our backbone for multi-view 3D reconstruction, the increase in depth will cause degradation of reconstruction accuracy when the model exceeds a certain depth. To tackle it, we further propose the divergence-enhanced Transformer that can slow down the divergence decay in the self-attention layers by enhancing the discrepancy of the embeddings from different views.

The contributions can be summarized as follows:

- We propose a brand new framework VolT for multi-view 3D object reconstruction. Different from the previous CNN based methods that using a separate design of feature extraction + view fusion, we unify these two stages into a single Transformer network and re-frame the 3D reconstruction as a "sequence-to-sequence" prediction problem.
- The proposed method can jointly and naturally explore multi-level correspondence and associations between the 2D input views and 3D output volume with in a single unified framework.
- We investigate the problem of "divergence decay" in the proposed 3D Volume Transformer layers and propose a view-divergence enhancing operation in our self-attention layers to avoid such degradation.
- Our method achieves a new state-of-the-art for multi-view 3D reconstruction on ShapeNet with only 30% amount of parameters of other recent CNN-based methods. Our method also shows better scaling capability on the number of input views.

II. Related Work

A. Multi-view 3D Reconstruction

Reconstructing an object's 3D shape from multi-view images has long been a research hot-spot in both computer vision and computer graphics. Traditional methods [12], [13] of this field are typically designed based on hand-crafted geometric features. Some representatives of early methods like Structure from Motion (SfM) [12], Simultaneous Localization and Mapping (SLAM) [13] need extrinsic camera parameters, which are not always feasible to obtain. Recently, CNN-based approaches, without requiring viewpoint labels or camera calibration, have quickly become the main stream of multi-view 3D reconstruction [4], [5] and have achieved the state of the art reconstruction accuracy.

In CNN-based methods, a 2D-CNN view encoder, a 3D-CNN view decoder, and a multi-view fusion model are usually separately designed for 3D reconstruction. Among them, the fusion plays an central role in the integration of multi-view feature information. Previous multi-view fusion methods can be roughly grouped into three categories, i.e., pooling-based fusion, learnable weighted-sum fusion and RNN-based fusion. The The pooling-based fusion, including max-pooling fusion and average-pooling fusion, only learns partial information of multiple views and ignores the view associations [14], [15]. The learnable weighted-sum fusion models are introduced to resolve these problems [1], [2], [3]. The RNN-based fusion methods like 3D-R2N2 [4] and LSM [5] can learn effective view-to-view relations but are computational expensive and permutation-variant [6]. In this paper, different from the above CNN-based methods, we propose a Transformer-based 3D reconstruction method, which unifies the feature extraction and view fusion in a single model and naturally explore the relationship between different input views.

B. Transformer

In natural language processing, Transformer models have achieves great success in a variety of tasks such as machine translation, text classification, and question answering [16]. The key to the Transformer is the multi-head self-attention operation, which aggregates features among every token pairs of the embedding sequence. Recently, Transformer has been also successfully adapted to the computer vision field [17], [10], [18] and shows promising application prospective. DETR [17] provides a new framework for object detection that combines a 2D CNN with a Transformer, and directly predicts (in parallel) the final object detection as a sequence of language tokens. ViT [10] applies Transformer directly to sequences of image patches for the image classification task without using CNNs features and have achieves comparable and even higher image classification accuracy when pretrained on large-scale dataset. In CNN-based multi-view 3D reconstruction methods, it is still a difficult task to design a fusion model that can explore the deep relationship between views while maintaining the permutation-invariant capability. A natural advantage of the Transformer in multi-view 3D reconstruction is that its token embedding can be abstracted and learned layer by layer in a disorderly manner, which can naturally ease the pain points of CNN-based methods.

无序学习

III. Methodology

A. Framework

The proposed 3D volume Transformer model consists of a 2D-view encoder and a 3D-volume decoder. The 2D-view encoder encodes the relevant information amongst different views via view attention layers. The 3D-volume decoder learns global correlations of different spatial locations in volume attention layers and predicts the final 3D volume output. We uniformly split the 3D space into a set of tokens and the predicted volumes for each token are finally stitched into the final 3D reconstruction output. Fig. 1 shows an overview of the proposed framework.

In this paper, we implement three different versions of method based on the proposed framework: Vanilla 3D Volume Transformer (**VolT**), Vanilla 3D Volume Transformer+ (**VolT+**), and view-divergence-Enhance 3D Volume Transformer (**EVolT**).

- **VolT:** A baseline implementation of the proposed method using vanilla Transformer model as our baseline and using standard VGG16 features as our initial view embeddings.
- **VolT+:** Using 2D-view embeddings obtained from an advanced pretrained CNN compared with VolT. We use it to testify the impact of 2D-view embeddings on our Transformer-based framework for multi-view 3D reconstruction.
- **EVolT:** A full implementation of our method adopting the proposed view-divergence enhancing function in the proposed 3D Volume Transformer framework.

Here, to obtain 2D-view initial embeddings, we use a pretrained CNN that is shared among multi-views. Note that we can also build "EVolT+" with an advanced CNN for 2D-view embedding learning in VolT+ to further improve the performance of EVolT. However, we prefer to keep the small parameter size of the EVolT and emphasize the advantage of the Transformer.

B. Divergence-enhanced 2D-view Encoder

Suppose $\mathcal{I} = \{\mathbf{I}^1, \mathbf{I}^2, \cdots, \mathbf{I}^M\}$, where M denote the multi-view image set of an object to be reconstructed. For each $\mathbf{I}^m$, we first use a pretrained view-shared CNN to obtain a set of initial view embedding $\mathbf{x}^m \in \mathbb{R}^{1\times d}$, where d is the feature dimension. Then, the 2D-view encoder takes in the initial view embeddings $\mathbf{X}_0 = [\mathbf{x}^1; \mathbf{x}^2; \cdots; \mathbf{x}^M] \in \mathbb{R}^{M\times d}$ and refines the multi-view representations by exploring a global relationship amongst multiple views using a series of self-attention layers. Here, to keep permutation invariant for the view sequence $\mathcal{X}$, the positional encodings of a standard Transformer are removed. We build our divergence-enhanced 2D-view encoder based on DETR [17] by stacking $N = 6$ basic blocks. Each basic block consists of a multi-head divergence-enhanced view attention layer (denoted as MH-DEAtt, Eq. (2)) and a position-wise feed-forward network (FFN, Eq. (4)). The 2D-view encoder is formulated as follows:

$$\mathbf{X}_0 = [\mathbf{x}^1; \mathbf{x}^2; \cdots; \mathbf{x}^M] \tag{1}$$
$$\mathbf{X}_l = \text{MH-DEAtt}(\mathbf{X}_{l-1}, \mathbf{X}_0) \tag{2}$$
$$\hat{\mathbf{X}}_l = \text{Norm}(\mathbf{X}_l + \mathbf{X}_{l-1}) \tag{3}$$
$$\mathbf{X}_l = \text{FFN}(\hat{\mathbf{X}}_l) \tag{4}$$
$$\mathbf{X}_l = \text{Norm}(\mathbf{X}_l + \hat{\mathbf{X}}_l) \tag{5}$$

where "Norm" denotes layer normalization and l is the index of a basic block ($l = 1, \cdots, L$). The embeddings of the layer L are used as the output of our 2D-view encoder.

As shown in the right side of Figure 1, the scaled dot-product attention (denoted as Attn) aggregates the feature representations amongst multiple views by learning view-to-view relationships. Meantime, we propose a view-divergence enhancing function (DiView) to ease the discrepancy degradation of the multi-view representations in deeper layers. Specifically, DiView introduces skip connections and concatenates the internal view features with the input view embeddings in the feature dimension. The MH-DEAtt layer is defined as follows

$$\begin{aligned} \text{MH-DEAtt}(\mathbf{X}, \mathbf{X}_0) &= \text{DiView}(\mathbf{A}, \mathbf{X}_0)\mathbf{W}_{view} \\ \text{where } \mathbf{A} &= \text{cat}(\mathbf{A}^1, \cdots, \mathbf{A}^H) \\ \mathbf{A}^h &= \text{Attn}(\mathbf{Q}^h, \mathbf{K}^h, \mathbf{V}^h) \end{aligned} \tag{6}$$

where "cat" denotes the concatenation operation and h is the number of head in MH-DEAtt layer. $\mathbf{W}_{view} \in \mathbb{R}^{(Hd_k+d)\times d}$ denotes the parameter matrix of the linear function, and d_k is the feature dimension in each head. In the h-th head, M queries stacked in $\mathbf{Q}^h \in \mathbb{R}^{M\times d_k}$ are projected from M view embeddings stacked in $\mathbf{X}$ with the parameter matrix $\mathbf{W}_Q^h \in \mathbb{R}^{d\times d_k}$. Similarly, the keys and values stacked in

$\mathbf{K}^h \in \mathbb{R}^{M\times d_k}$ and $\mathbf{V}^h \in \mathbb{R}^{M\times d_k}$ are obtained with parameter matrices $\mathbf{W}_K^h \in \mathbb{R}^{d\times d_k}, \mathbf{W}_V^h \in \mathbb{R}^{d\times d_k}$, respectively:

$$\mathbf{Q}^h = \mathbf{X}\mathbf{W}_Q^h; \quad \mathbf{K}^h = \mathbf{X}\mathbf{W}_K^h; \quad \mathbf{V}^h = \mathbf{X}\mathbf{W}_V^h. \tag{7}$$

Specifically, in the Attention function "Attn", the output for a query is represented as an attention-score weighted sum of the values $\mathbf{V}$. Therefore, the Attn function is formulated as

$$\text{Attn}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}, \tag{8}$$

where d_k is a scalar for normalization.

C. 3D-volume Decoder

The 3D-volume decoder in our framework learns the global correlation amongst different spatial locations and explore the relationship between the view and spatial domains. Given an object, we denote $\mathbf{Y}_0 = [\mathbf{y}^1;\mathbf{y}^2;\cdots;\mathbf{y}^N]$ as a sequence learnable 3D-volume queries at the input end of the decoder, where $\mathbf{y}^n \in \mathbb{R}^{1\times d}$ corresponds to the n-th 3D-volume. Here, positional encodings $\mathbf{E}^{pos}$ are added to 3D-volume embeddings to keep the position information in the spatial domain. In the decoder, a basic block contains a volume attention layer, a view-volume attention layer, and a FFN. The decoder can be formulated as follows:

$$\mathbf{Y}_0 = [\mathbf{y}^1;\mathbf{y}^2;\cdots;\mathbf{y}^N] + \mathbf{E}^{pos} \tag{9}$$
$$\mathbf{Y}_l = \text{MH-VolAttn}(\mathbf{Y}_{l-1}) \tag{10}$$
$$\hat{\mathbf{Y}}_l = \text{Norm}(\mathbf{Y}_l + \mathbf{Y}_{l-1}) \tag{11}$$
$$\mathbf{Y}_l = \text{MH-ViewVolAttn}(\hat{\mathbf{Y}}_l, \mathbf{X}_L) \tag{12}$$
$$\tilde{\mathbf{Y}}_l = \text{Norm}(\mathbf{Y}_l + \hat{\mathbf{Y}}_l) \tag{13}$$
$$\mathbf{Y}_l = \text{FFN}(\tilde{\mathbf{Y}}_l) \tag{14}$$
$$\mathbf{Y}_l = \text{Norm}(\mathbf{Y}_l + \tilde{\mathbf{Y}}_l) \tag{15}$$

where MH-VolAttn (in Eq.(10)) and MH-ViewVolAttn (in Eq.(12)) denote the multi-head volume attention layer and the multi-head view-volume attention layer, respectively. We use the embeddings at the layer L as the output of the decoder.

In our decoder, the MH-VolAttn layer learns global dependencies amongst different 3D-volumes, and is calculated as follows:

$$\begin{aligned}&\text{MH-VolAttn}(\mathbf{Y}) = \text{cat}(\mathbf{A}^1,\cdots,\mathbf{A}^H)\mathbf{W}_{vol}\\ &\text{where } \mathbf{A}^h = \text{Attn}(\mathbf{Y}\mathbf{W}_Q^h, \mathbf{Y}\mathbf{W}_K^h, \mathbf{Y}\mathbf{W}_V^h.\end{aligned} \tag{16}$$

The MH-ViewVolAttn layer integrates the relevant information across the view and spatial domains, and is calculated as follows:

$$\begin{aligned}&\text{MH-ViewVolAttn}(\mathbf{Y},\mathbf{X}_L) = \text{cat}(\mathbf{A}^1,\cdots,\mathbf{A}^H)\mathbf{W}\\ &\text{where } \mathbf{A}^h = \text{Attn}(\mathbf{Y}\mathbf{W}_Q^h, \mathbf{X}_L\mathbf{W}_K^h, \mathbf{X}_L\mathbf{W}_V^h)fi\end{aligned} \tag{17}$$

where $\mathbf{W}_{vol} \in \mathbb{R}^{Hd_k\times d}$ and $\mathbf{W} \in \mathbb{R}^{Hd_k\times d}$ are the parameter matrices of the corresponding linear functions.

Finally, after the 3D-volume decoder, we use a linear function to project the output embeddings of each 3D volume to their 3D output space. Then the predicted 3D volumes are reshaped and grouped to the final reconstruction output.

TABLE I
PARAMETER SIZES OF COMPETING METHODS AND PRETRAINED CNNS FOR 2D-VIEW EMBEDDINGS IN COMPETING METHODS.

	Pretrained CNN used for 2D-view embeddings	Param. (M)
Pix2Vox-A [1]	VGG16 [20]	114.24
Pix2Vox++/A [3]	ResNet50 [21]	96.31
VolT	VGG16 [20]	28.63
VolT+	2D-CNN+3D-DCNN	96.76
EVolT	VGG16 [20]	29.03

IV. EXPERIMENT

A. Dataset

We utilize the ShapeNet dataset [19] to evaluate the proposed methods and other comparison methods. We follow 3D-R2N2 [4] and use the same setting for a fair comparison. Specifically, we use a subset of ShapeNet which consists of 13 categories and 43,783 common 3D objects. For each 3D object, 24 2D-images are rendered from different viewing angles circling around the object.

B. Evaluation Metrics

1) IoU: The mean Intersection-over-Union (IoU) calculates the matching degree between predicted 3D voxel grids and their ground-truth grids. A higher IoU value means a better reconstruction result. For each voxel grid, the IoU is defined as:

$$\text{IoU} = \frac{\sum_{(i,j,k)} \text{I}(y(i,j,k) > t)\text{I}(\bar{y}(i,j,k))}{\sum_{(i,j,k)} \text{I}[\text{I}(y(i,j,k) > t) + \text{I}(\bar{y}(i,j,k))]}, \tag{18}$$

where $y(i,j,k)$ denotes the predicted occupancy probability which is binarized with an optimal fixed voxelization-threshold t for compared methods. $\bar{y}(i,j,k)$ is the ground truth at (i,j,k). $\text{I}(\cdot)$ is an indicator function.

2) F-Score: Compared with IoU, F-score [22], [3] explicitly evaluates the distance between object surfaces, which is more interpretable. F-score is formally defined as the harmonic mean between precision $\text{P}(d)$ and recall $\text{R}(d)$ with a distance threshold d:

$$\text{F-Score}(d) = \frac{2\text{P}(d)\text{R}(d)}{\text{P}(d) + \text{R}(d)} \tag{19}$$

A higher F-score with a stringent distance threshold indicates a better reconstruction result.

In F-Score, $\text{P}(d)$ estimates the reconstruction accuracy by counting the portion of reconstructed points lying within the distance $d = 1\%$ to the ground truth. $\text{R}(d)$ quantifies the reconstruction completeness by counting the percentage of ground-truth points lying within the distance d to the reconstruction. These two metrics are defined as follows:

$$\text{P}(d) = \frac{1}{|\mathcal{R}|}\sum_{\mathbf{r}\in\mathcal{R}}[e_{\mathbf{r}\to\mathcal{G}} < d], e_{\mathbf{r}\to\mathcal{G}} = \min_{\mathbf{r}\in\mathcal{G}}\|\mathbf{r}-\mathbf{g}\| \tag{20}$$

$$\text{R}(d) = \frac{1}{|\mathcal{G}|}\sum_{\mathbf{g}\in\mathcal{G}}[e_{\mathbf{g}\to\mathcal{R}} < d], e_{\mathbf{g}\to\mathcal{R}} = \min_{\mathbf{r}\in\mathcal{R}}\|\mathbf{g}-\mathbf{r}\|, \tag{21}$$

where $[\cdot]$ is the Iverson bracket. $\mathcal{G}$ is the ground-truth point set and $\mathcal{R}$ is the reconstructed point set being evaluated. We apply F-Score with the same setting in [3].

TABLE II
COMPARISON OF 24-VIEW RECONSTRUCTION ON SHAPENET USING IOU AND F-SCORE. THE BEST SCORE FOR EACH CATEGORY IS IN BOLD.

	24-view IoU					24-view F-Score@1%				
Category	Pix2Vox-A	Pix2Vox++/A	VolT	VolT+	EVolT	Pix2Vox-A	Pix2Vox++/A	VolT	VolT+	EVolT
airplane	0.731	0.729	0.719	0.725	**0.741**	0.635	0.614	0.604	0.618	**0.636**
bench	0.679	0.686	0.678	0.682	**0.707**	0.525	0.522	0.513	0.525	**0.548**
cabinet	0.822	0.829	0.825	0.825	**0.832**	0.448	0.456	0.452	0.455	**0.464**
car	0.880	0.883	0.884	0.885	**0.894**	0.598	0.598	0.604	0.609	**0.624**
chair	0.620	0.647	0.645	0.641	**0.681**	0.318	0.341	0.339	0.340	**0.373**
display	0.599	0.613	0.635	0.613	**0.674**	0.320	0.335	0.366	0.339	**0.403**
lamp	0.475	0.493	0.478	0.481	**0.520**	0.335	0.351	0.320	0.338	**0.366**
speaker	0.751	0.762	0.762	0.753	**0.772**	0.309	0.326	0.327	0.317	**0.339**
rifle	0.676	0.686	0.663	0.693	**0.711**	0.615	0.624	0.597	0.634	**0.653**
sofa	0.764	0.782	0.781	0.776	**0.800**	0.427	0.454	0.449	0.448	**0.478**
table	0.644	0.666	0.649	0.658	**0.675**	0.398	0.419	0.407	0.418	**0.431**
telephone	0.837	0.849	0.857	0.850	**0.867**	0.659	0.666	0.678	0.675	**0.687**
watercraft	0.655	0.668	0.670	0.670	**0.693**	0.441	0.460	0.456	0.470	**0.494**
Overall	0.706	0.720	0.714	0.716	**0.738**	0.462	0.473	0.468	0.475	**0.497**

TABLE III
COMPARISON OF MULTI-VIEW RECONSTRUCTION ON SHAPENET USING IOU AND F-SCORE. THE BEST SCORE FOR EACH VIEW NUMBER IS IN BOLD.

F-Score@1%	24	23	22	21	20	18	16	14	12	8	6	4
3D-R2N2 [4]	-	-	-	-	0.383	-	0.382	-	0.382	0.383	-	0.378
AttSets [2]	-	-	-	-	0.448	-	0.447	-	0.445	0.444	-	0.430
Pix2Vox-A [1]	0.462	0.462	0.462	0.462	0.462	0.461	0.461	0.461	0.460	0.458	0.456	0.452
Pix2Vox++/A [3]	0.473	-	-	-	0.462	-	0.461	-	0.460	0.459	-	**0.457**
VolT	0.468	0.467	0.467	0.465	0.464	0.461	0.459	0.456	0.450	0.430	0.410	0.356
VolT+	0.475	0.475	0.474	0.474	0.474	0.473	0.472	0.471	0.469	**0.464**	**0.460**	0.451
EVolT	**0.497**	**0.496**	**0.495**	**0.494**	**0.492**	**0.489**	**0.486**	**0.481**	**0.475**	0.448	0.423	0.358
IoU												
3D-R2N2 [4]	-	-	-	-	0.636	-	0.636	-	0.636	0.635	-	0.625
AttSets [2]	0.694	-	-	-	0.693	-	0.692	-	0.688	0.685	-	0.675
Pix2Vox-A [1]	0.706	0.706	0.706	0.706	0.706	0.705	0.705	0.705	0.704	0.702	0.700	0.697
Pix2Vox++/A [3]	0.720	-	-	-	0.719	-	0.718	-	0.717	**0.715**	-	**0.708**
VolT	0.714	0.713	0.712	0.711	0.711	0.708	0.706	0.703	0.699	0.681	0.662	0.605
VolT+	0.716	0.716	0.716	0.715	0.715	0.714	0.714	0.713	0.711	0.707	0.704	0.695
EVolT	**0.738**	**0.738**	**0.737**	**0.735**	**0.735**	**0.732**	**0.729**	**0.726**	**0.720**	0.698	0.675	0.609

3) Divergence measurement for multi-view representations: We also define a metric to explore the convergence of multi-view representations in different layers. Since the convergence has a positive correlation with the divergence decay of multi-view attentions, we utilize a similarity measure based on multi-view attentions to evaluate the divergence enhancing ability in our method.

In each view attention layer, an attention-score matrix $\mathbf{S} = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})$ contains view-to-view attention vectors. The m-th row of $\mathbf{S}$, denoted as $\mathbf{s}_m$, is an attention-score vector where each element represents its attention weight to another view. For 3D reconstruction of a specific object, the Euclidean distance measuring the similarity of multi-view attentions, is defined as

$$D = \frac{1}{N_{view}} \sum_m \|\mathbf{s}_m - \bar{\mathbf{s}}\|_2 \quad \text{where} \quad \bar{\mathbf{s}} = \frac{1}{N_{view}} \sum_m \mathbf{s}_m. \tag{22}$$

Here, a small D means a more considerable similarity and the convergence of multi-view representations.

C. Implementation Details

We set the batch size to 64 and the view image size to 224×224 for training. The 3D spatial size of the voxelized output is set to $32 \times 32 \times 32$. The VolT and its two variants VolT+, and EVolT are trained by an AdamW optimizer [23] with a β_1 of 0.9 and a β_2 of 0.999.

Table I shows the parameter sizes of competing methods and pretrained CNNs for the initial view embeddings used in different competing methods. Compared with Pix2Vox-A [1] and Pix2Vox++/A [3], the parameter size of EVolT is only around 30% of them. To obtain the reported best results, Pix2Vox-A and Pix2Vox++/A both adopt an additional 3D-CNN-based refiner containing another 3D-CNN and 3D-DCNN. In contrast, our proposed end-to-end methods do not need additional refiner and can also achieve the best results. To testify the effect of the transformer architecture, in VolT+, we apply an advanced CNN feature extraction model for 2D-view embeddings from the 2D-CNN and 3D-DCNN without the last layer in Pix2Vox-A.

D. Multi-view 3D Object Reconstruction

*1) **Quantitative results**:* Here, we show the quantitative results of compared methods on ShapeNet using different evaluation metrics. Table II shows the comparison of 24-view object reconstruction on ShapeNet using IoU and F-Score metrics. The highest value for each category is highlighted in bold. This table shows that EVolT reaches the highest IoU and F-score amongst the compared methods. VolT gets moderate

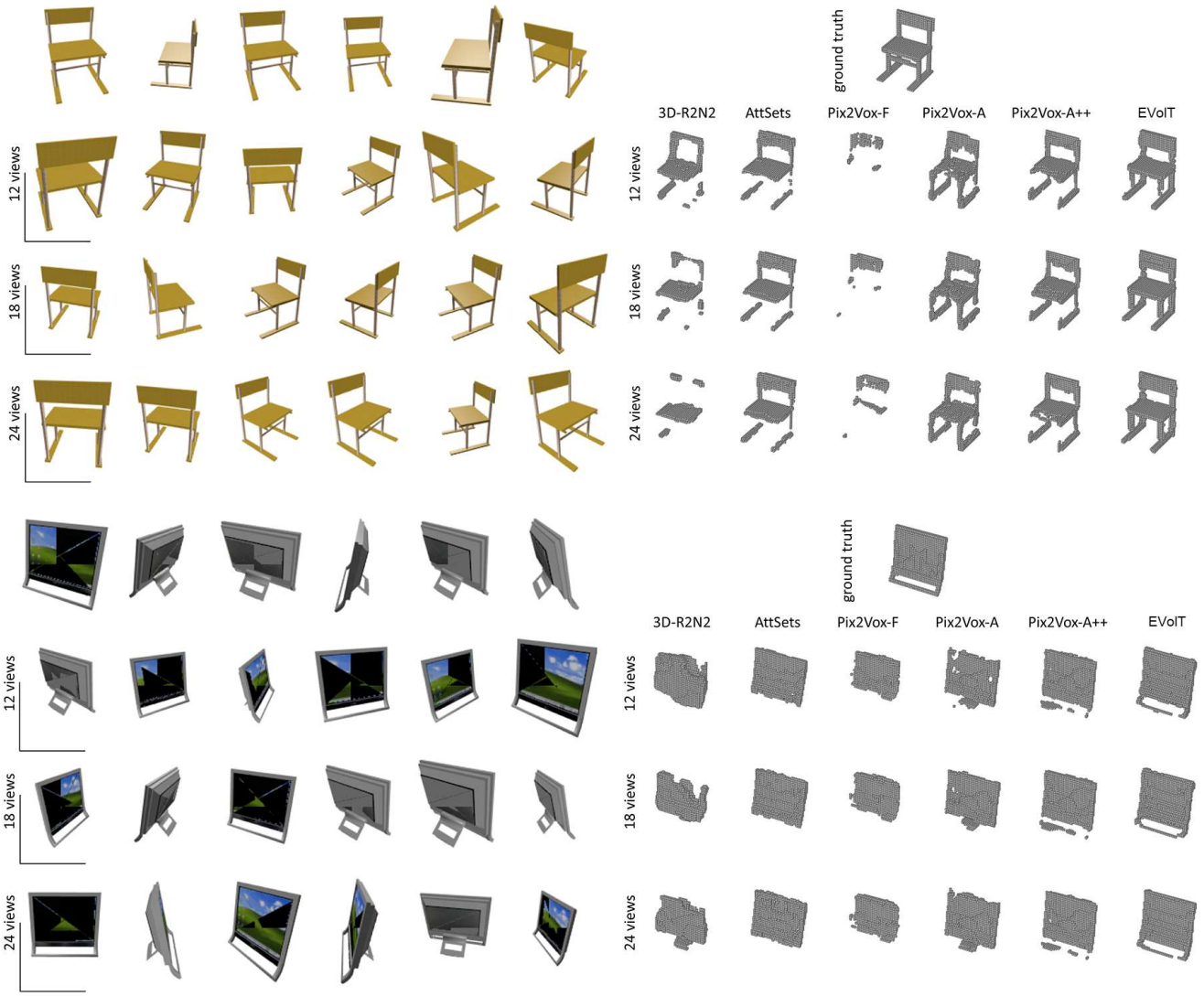

Fig. 2. Qualitative 3D object reconstruction results on ShapeNet based on different number of input 2D-view images.

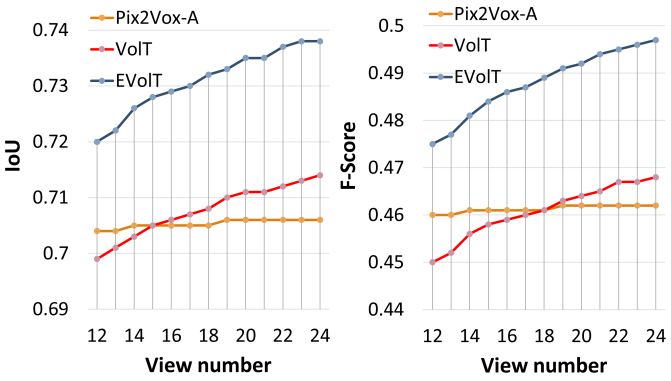

Fig. 3. Effect of the view-divergence enhancing function on 3D reconstruction results.

results between Pix2Vox-A and Pix2Vox++/A. VolT+ works better than VolT because it uses better initial features. However, VolT+ still falls behind EVolT even the EVolT is simply based on the plain VGG features. These observations indicate that the view-divergence enhancing function in EVolT plays an indispensable role in increasing its performance against the compared methods.

Table III shows the multi-view object reconstruction results on ShapeNet. The best score for each number of views is highlighted in bold. This table shows that the performances of our methods increase appreciably as the number of views increases. In comparison, other compared methods increase slightly when the view number enlarges. For example, the mean IoU of EVolT increases by 0.04 from 8 views to 24 views, which is eight times the improvement of Pix2Vox++/A. This observation indicates that the proposed Transformer-based methods has better scaling ability and can learn a more comprehensive 3D representation with the increase of view number. We can also see from this table that our proposed methods get the best F-Score when the view number is larger than 6 and get the best IoU when the view number is higher

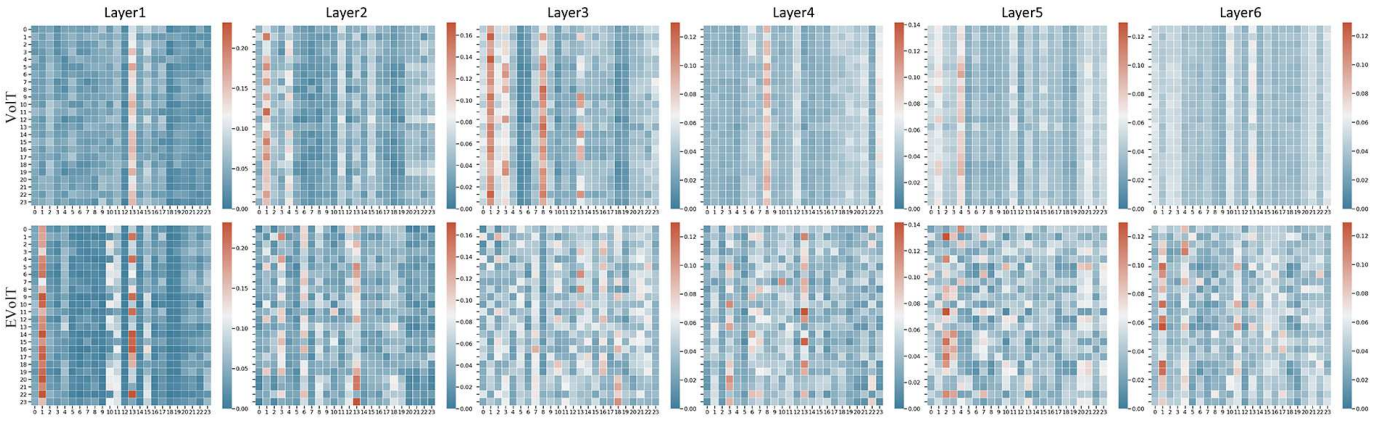

Fig. 4. Multi-view attention-matrix visualization in VolT and EVolT.

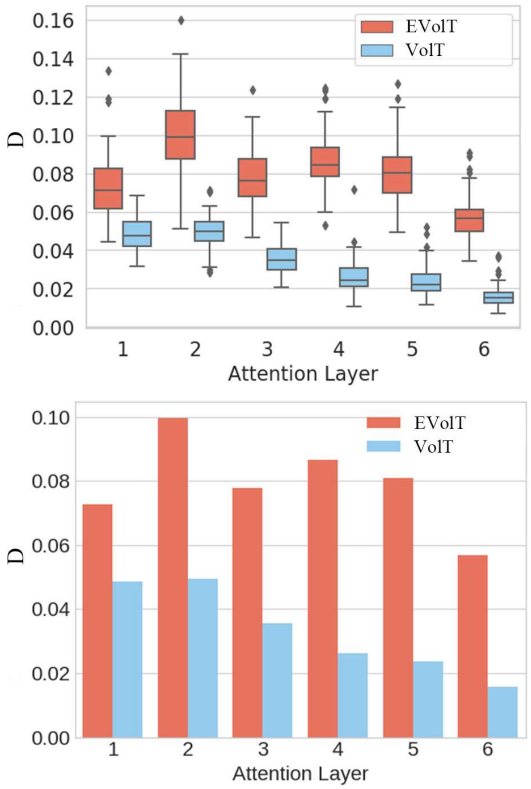

Fig. 5. Discrepancy amongst multi-view representations in VolT and EVolT.

than 12.

*2) **Qualitative results**:* In Figure 2, we show the qualitative results of 3D object reconstruction of different methods on ShapeNet. In each object sample, we provide object reconstruction results from different number of input views, i.e., 12 views, 18 views, and 24 views. The first two rows on the left part of Figure 2 show the 12 input views of an object, and the corresponding reconstruction results of competing methods are shown at the second row on the right. Similarly, the first three rows on the left part are the 18 input views corresponding to the results on the right.

Figure 2 shows that EVolT can obtain more accurate and complete 3D reconstruction against compared methods. For example, the EVolT results in the last column successfully recover chair legs and monitor stand while other methods only show incomplete parts. More qualitative results can be found in our Supplementary Material.

E. Ablation Study

The following ablation experiments are made to verify the effectiveness of the proposed view-divergence enhancing function.

1) Effect on 3D reconstruction accuracy: In Figure 3, we quantitatively evaluate the influence of the view-divergence enhancing function on 3D reconstruction results by comparing EVolT with VolT and Pix2Vox-A. From Figure 3, we can observe that EVolT significantly outperforms VolT that achieves better results than Pix2Vox-A. This indicates the positive effect of the view-divergence enhancing function on 3D reconstruction results.

2) Effect on the view divergence: In Figure 4, we visualize the view-to-view attention matrix in different layers by VolT and EVolT. We set the input view number to 24 in this experiment. In the attention matrix at each layer, the m-th row shows an attention vector where each element is the attention weight of the m-th view to another view. From the top of Figure 4, we can observe that rows become more similar in a standard transformer as the attention layers go deeper. As a comparison, in EVolT, we can see the diversity of multi-view attention still keeps in deep layers, which means that the divergence enhancement function in EVolT can effectively slow down the convergence degradation of multi-views in deeper layers.

In Figure 5, the similarity measurement score D in Eq. 22 is also recorded to analyze the convergence amongst multi-view representations in each layer. For 100 randomly chosen objects, we plot D in different layers displayed in the left of Figure 5, and the right shows the average values of D. A small D suggests a significant convergence amongst multi-view representations. As shown in Figure 5, the value of D obtained by VolT declines gradually with deepening the 2D-

view encoder layer while the value of D of EVolT at the same layer keeps higher than that of VolT.

The ablation studies indicate that the view-divergence enhancing function plays an essential role in improving the proposed EVolT performance and relieving the convergence amongst multi-view representations in different layers.

V. Conclusion and Future Works

In this paper, we propose a Transformer-based framework for multi-view 3D reconstruction and achieves state-of-the-art accuracy on ShapeNet with fewer parameters than other CNN-based methods. We propose three versions of the method (VolT, VolT+ and EVolT) to explore view and spatial domain relationships for multi-view 3D reconstruction. Meantime, we explore the problem of divergence decay for the multi-view information in deeper layers and proposed view-divergence enhancing function to ease such a problem. In our future work, we will work on exploring the interpretability of the proposed framework and give its explanation for 3D reconstruction. We also plan to build an interpretable way to visualize and understand the correspondence between the latent representation and multiple input views.

References

[1] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang, "Pix2vox: Context-aware 3d reconstruction from single and multi-view images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2690–2698.

[2] B. Yang, S. Wang, A. Markham, and N. Trigoni, "Robust attentional aggregation of deep feature sets for multi-view 3d reconstruction," *International Journal of Computer Vision*, vol. 128, no. 1, pp. 53–73, 2020.

[3] H. Xie, H. Yao, S. Zhang, S. Zhou, and W. Sun, "Pix2vox++: multi-scale context-aware 3d object reconstruction from single and multiple images," *International Journal of Computer Vision*, vol. 128, no. 12, pp. 2919–2935, 2020.

[4] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. 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.

[5] A. Kar, C. Häne, and J. Malik, "Learning a multi-view stereo machine," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[6] O. Vinyals, S. Bengio, and M. Kudlur, "Order matters: Sequence to sequence for sets," in *International Conference on Learning Representations (ICLR)*, 2016.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[8] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," *arXiv preprint arXiv:2012.12877*, 2020.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021.

[11] Y. Dong, J.-B. Cordonnier, and A. Loukas, "Attention is not all you need: Pure attention loses rank doubly exponentially with depth," 2021.

[12] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer, "A survey of structure from motion," *Acta Numerica*, vol. 26, pp. 305–364, 2017.

[13] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: a survey," *Artificial intelligence review*, vol. 43, no. 1, pp. 55–81, 2015.

[14] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang, "DeepMVS: Learning multi-view stereopsis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2821–2830.

[15] D. Paschalidou, O. Ulusoy, C. Schmitt, L. Van Gool, and A. Geiger, "Raynet: Learning volumetric 3d reconstruction with ray potentials," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3897–3906.

[16] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165*, 2020.

[17] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[18] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever, "Generative pretraining from pixels," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1691–1703.

[19] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[22] M. Tatarchenko, S. R. Richter, R. Ranftl, Z. Li, V. Koltun, and T. Brox, "What do single-view 3d reconstruction networks learn?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[23] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations (ICLR)*, 2019.

Supplementary Material: Multi-view 3D Reconstruction with Transformer

I. Comparison of Structures in Competing Methods

Table 6 gives a comparison between the proposed transformer-based 3D reconstruction methods and existing CNN-based methods from the perspective of components and structure.

CNN-based methods	Components	Feature Extraction Module	2D-CNN Encoder	Encode each 2D-view image into its view-specific feature representation. (2D-CNN: 2D Convolutional Neural Network)
			3D-DCNN Decoder	Decode a latent feature representation into a 3D voxel grid representing the 3D shape. (3D-DCNN: 3D Deconvolutional Neural Network)
		Multi-view Fusion Module		Aggregate multi-view representations to a fused representation for the 3D object reconstruction. Here, each view representation is learned from the corresponding 2D-view image.
		Refiner		An optional component: a residual network aims to correct wrongly recovered parts of a predicted 3D volume output. It is another "3D-CNN encoder+ 3D-DCNN decoder" with the U-Net connections.
	Structure	3D-R2N2		2D-CNN Encoder + RNN-based Fusion + 3D-DCNN Decoder
		AttSets		2D-CNN Encoder + Weighted-sum Fusion + 3D-DCNN Decoder
		Pix2Vox++/A; Pix2Vox-A		2D-CNN Encoder + 3D-DCNN Decoder + Weighted-sum Fusion + Refiner
	Design a feature extraction module and a multi-view fusion module separately.			
Our Transformer-based methods	Components	2D-view Transformer Encoder		Encode and fuse the multiple 2D-view information by exploring the "2D-view X 2D-view" relationships in 2D-View Attention layers.
		3D-volume Transformer Decoder		Decode and fuse the multi-view features from 2D-view encoder into each 3D-volume representation by learning "2D-view X 3D-volume" relationships in View-Volume Attention layers. Meanwhile, Volume Attention layers in the decoder further learn "3D-volume X 3D-volume" relationships by exploiting correlations amongst different 3D locations.
	Structure	2D-view Transformer Encoder + 3D-volume Transformer Decoder		
	Unify the feature extraction and multi-view fusion in Transformer based on the attention mechanism. By using the above unified design, "2D-view X 2D-view", "3D-volume X 3D-volume", and "2D-view X 3D-volume" relationships can be jointly explored by multiple attention layers in both the encoder and decoder.			

Fig. 6. Comparison of CNN-based methods and the proposed Transformer-based methods.

II. Additional Results

A. View Divergence

In Figure 7, we plot the estimated probability density of the D value at different attention layers for VolT and EVolT. We use kernel density estimation (KDE) to compute the probability density and explore the convergence of multi-view representations in different attention layers. A small D means a more considerable convergence of multi-view representations.

In each view attention layer, the probability density function $\hat{p}(D)$ of D is estimated as

$$\begin{aligned} \hat{p}(D) &= \frac{1}{N_{object} N_{view} h} \sum_{i}^{N_{object}} \sum_{m}^{N_{view}} K(\frac{D_m^i - D}{h}) \\ \text{where} \quad D_m^i &= \|\mathbf{s}_m^i - \frac{1}{N_{view}} \sum_{m}^{N_{view}} \mathbf{s}_m^i\|_2 \end{aligned} \tag{23}$$

where $\mathbf{s}_m^i$ is the attention vector of the m-th view for the i-th object. The number of random objects is set to $N_{object} = 100$. The input view number is set to $N_{view} = 24$. Here, we used the Gaussian kernel $K(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{x^2}{2})$. h is computed by the rule of thumb of Scott.

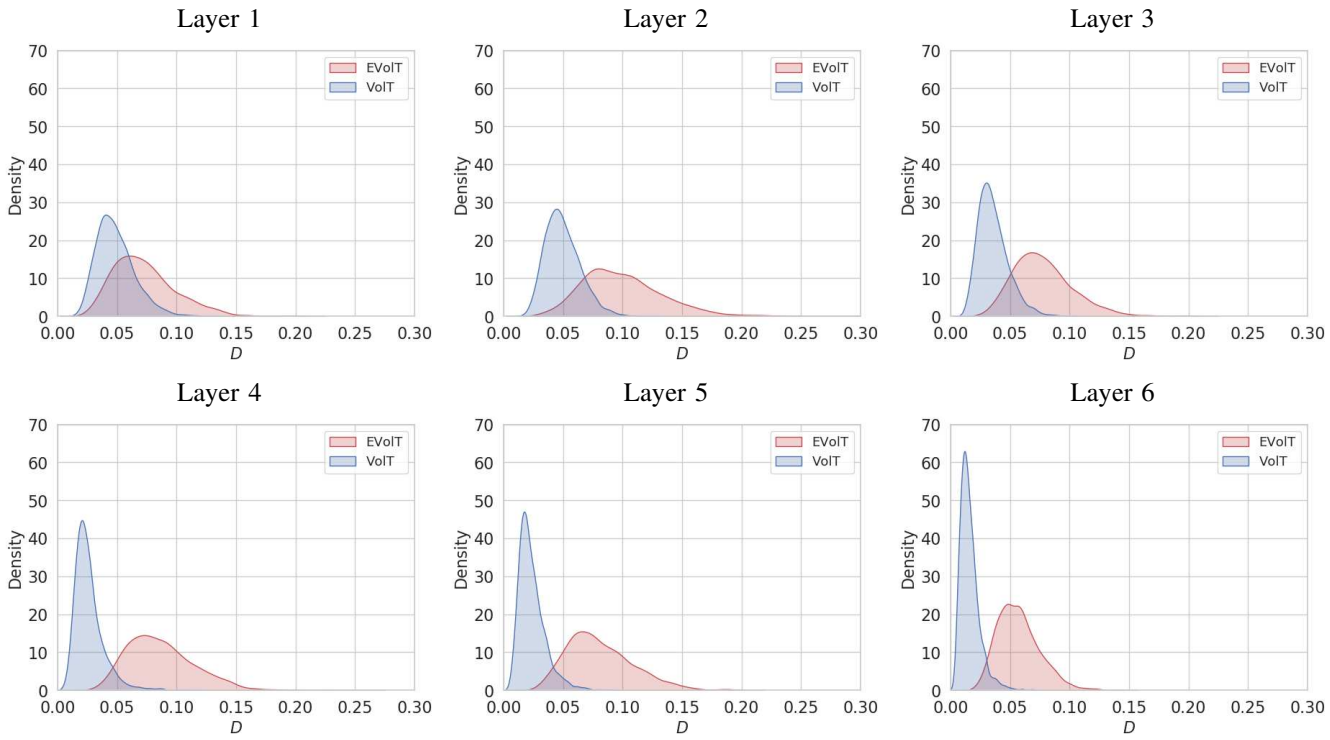

Fig. 7. Kernel density estimation of D value in different attention layers for VolT and EVolT.

It is shown in Figure 7 that the density of EVolT has a much larger variance than that of the VolT. Also, as the attention layers go deeper, the D value of the VolT gradually moves closer to 0 while the EVolT can still cover a larger range of D values. This indicates that the divergence enhancement function in EVolT can effectively slow down the convergence degradation of multi-views in deeper layers.

B. Qualitative Results

We provide more object reconstruction results of competing methods, as shown in Figure 8, 9, 10, and 11. In each object sample, we provide object reconstruction results from different number of input views, i.e., 12 views, 18 views, and 24 views. The first two rows on the left part of Figure 8 show the 12 input views of an object, and the corresponding reconstruction results of competing methods are shown at the second row on the right. Similarly, the first three rows on the left part are the 18 input views corresponding to the results on the right. The qualitative comparison suggests the superiority of the proposed method in terms of the reconstruction topology and details.

Fig. 8. Qualitative reconstruction results of competing methods for bench (top), aeroplane (middle), and sofa (bottom).

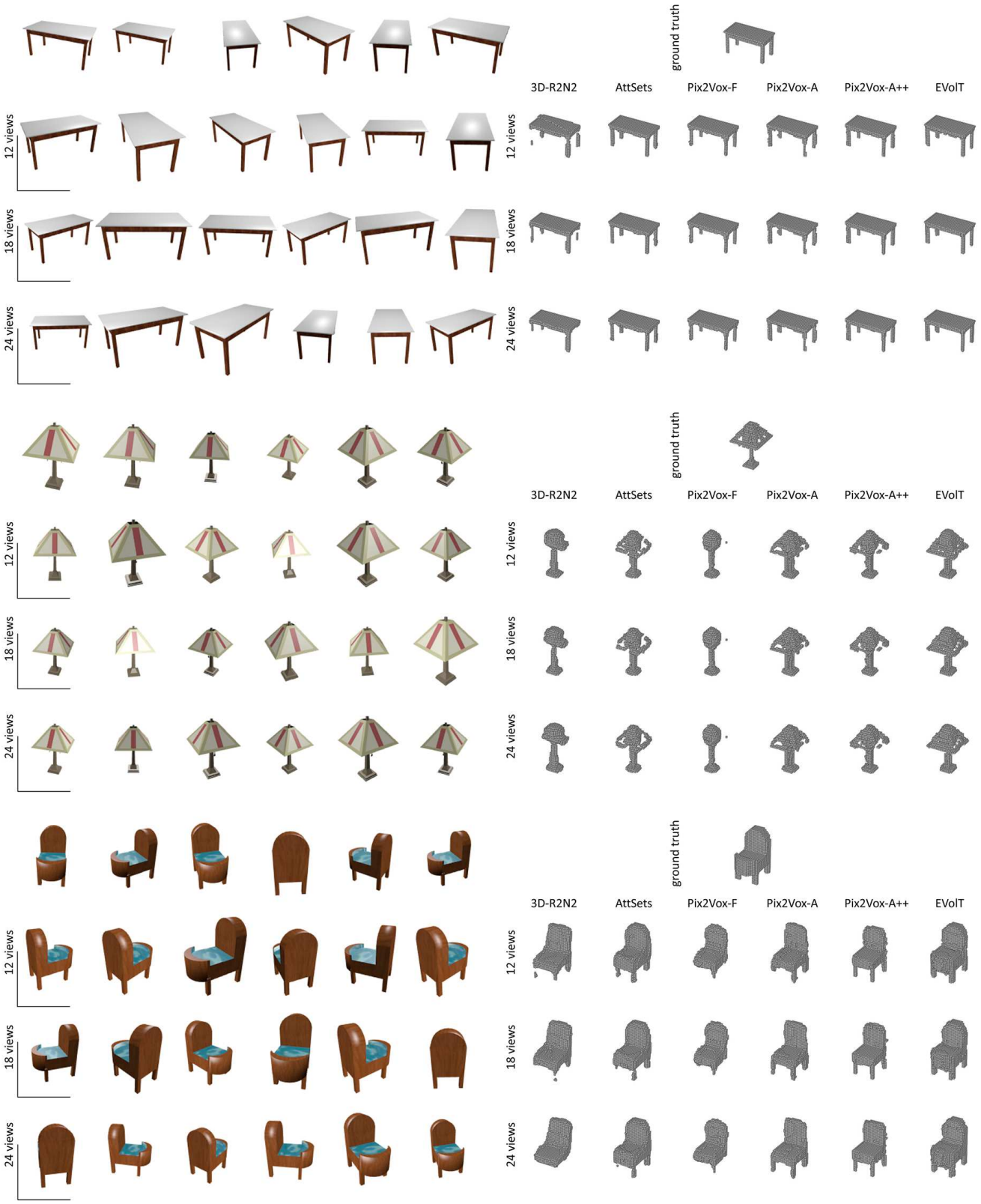

Fig. 9. Qualitative reconstruction results of competing methods for table (top), lamp (middle), and chair (bottom).

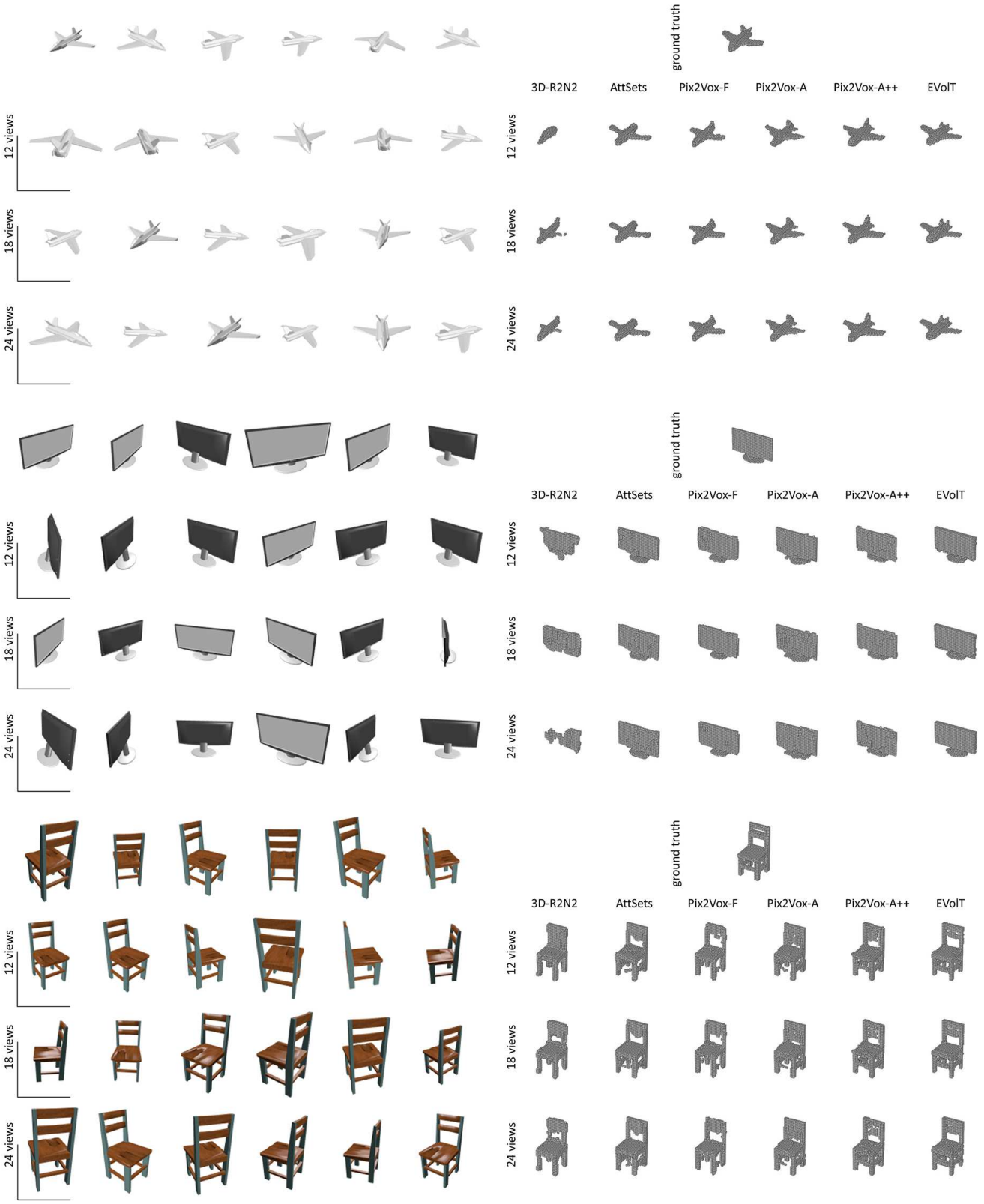

Fig. 10. Qualitative reconstruction results of competing methods for aeroplane (top), display (middle), and chair (bottom).

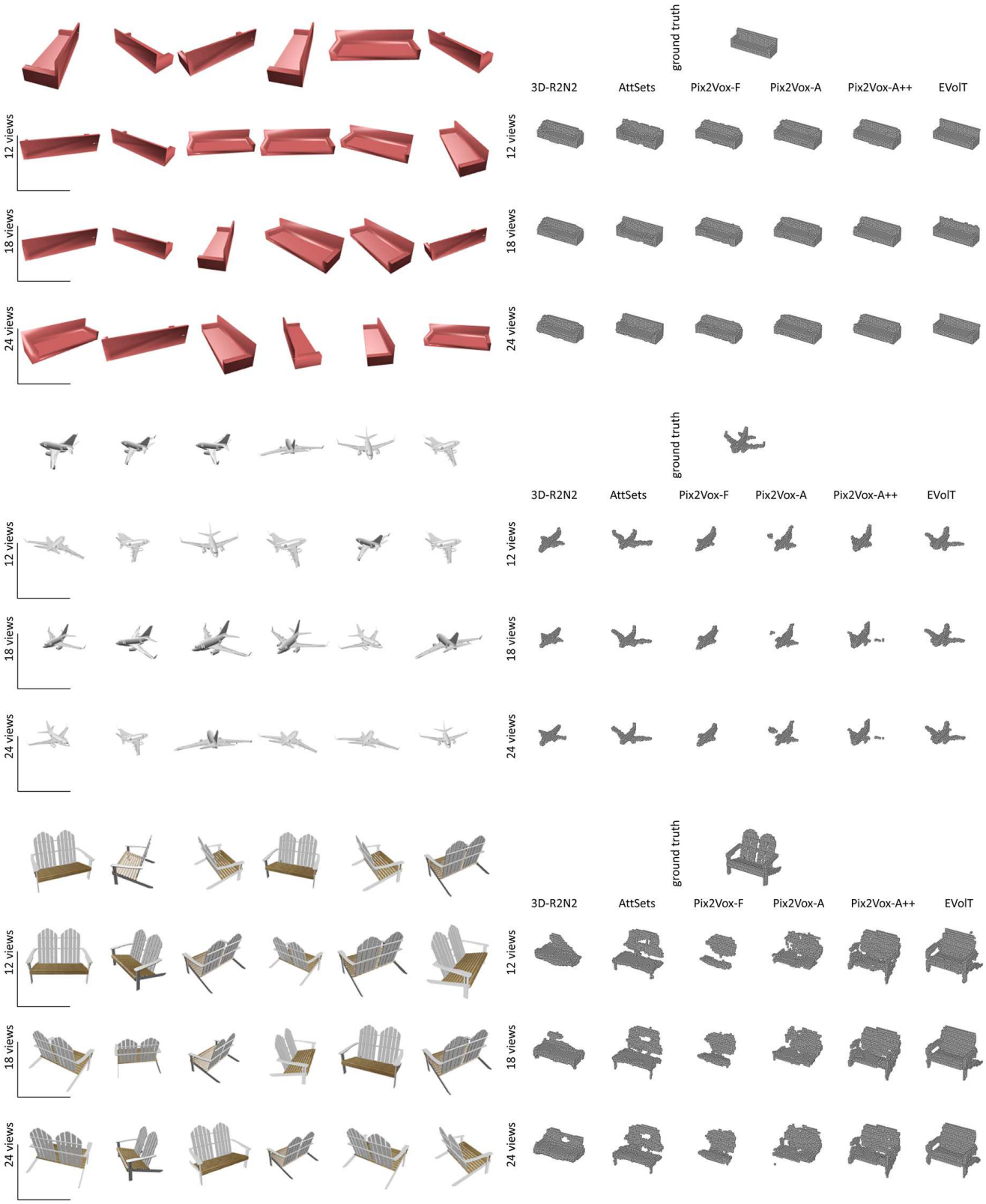

Fig. 11. Qualitative reconstruction results of competing methods for sofa (top), aeroplane (middle), and bench (bottom).