Multi-view 3D Reconstruction with Transformers

Dan Wang[1] Xinrui Cui[2*] Xun Chen[2]
Zhengxia Zou[3] Tianyang Shi[4] Septimiu Salcudean[1] Z. Jane Wang[1] Rabab Ward[1]
[1] University of British Columbia [2] University of Science and Technology of China
[3] University of Michigan, Ann Arbor [4] NetEase Fuxi AI Lab

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 - view feature extraction and multi-view fusion, are usually investigated separately, and the relations among multiple input views are rarely explored. Inspired by the recent great success in Transformer models, we reformulate the multi-view 3D reconstruction as a sequence-to-sequence prediction problem and propose a framework named 3D Volume Transformer. 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, our method achieves a new state-of-the-art accuracy in multi-view reconstruction with fewer parameters (70% less) than CNN-based methods. Experimental results also suggest the strong scaling capability of our method. Our code will be made publicly available.

1. 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, where the deep convolutional neural network (CNN) based approaches have so far achieved the state of the art results in this task [26, 28, 27].

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 single-view feature extraction and multi-view fusion 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 that processes each view image separately, the relations in different views are rarely explored. Although some recent approaches have introduced recurrent neural network (RNN) to learn object relationships among different views [5, 12], such a design lacks computational efficiency, and the input views of the RNN model are permutation-sensitive [22], which is not 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 input views exceeds a particular scale (e.g., 4), the accuracy of methods will be saturated, showing the difficulty of learning complementary knowledge from a large set of independent CNN feature extraction units [28, 27].

Considering the above challenges, we propose a new framework named "3D Volume Transformer" and explore the potential of the self-attention-based Transformer 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 the one hand, in multi-view 3D reconstruction, we need to learn the underlying 3D representation by exploring the relationships among multiple input views since we can only see part of the 3D structure from a particular view. On the other hand, in a Transformer model, the self-attention mechanism has recently shown its great power in learning complex semantic abstractions within an arbitrary number of input tokens [6, 20] 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 [21, 8] 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. The 2D-view Transformer

*Corresponding Author. Email:xinruic@ece.ubc.ca

encoder encodes and fuses multiple 2D-view information by exploring their "2D-view $\times$ 2D-view" relationships of different inputs. The 3D-volume Transformer decoder decodes and fuses the multi-view features from the encoder and generates a 3D probabilistic voxel output for each spatial query token. The attention layers in the decoder learn "2D-view $\times$ 3D-volume" relationships between each of the output voxel grids and input views. Meanwhile, volume attention layers in the decoder further learn "3D-volume $\times$ 3D-volume" relationships by exploiting correlations among different 3D locations. By using the above unified design, the "2D-view $\times$ 2D-view", "3D-volume $\times$ 3D-volume", and "2D-view $\times$ 3D-volume" relationships can be jointly explored by multiple attention layers in both the encoder and decoder networks.

Based on the above encoder-decoder Transformer structure design, we further investigate the "attention uniformity" problem in a Transformer model and propose an 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" [7], which encourages feature representations of input tokens to 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 [7]. We show that this problem is particularly prominent in the multi-view 3D reconstruction task and will limit the Transformer's capability to explore and abstract multi-view associations at a deeper level. To tackle it, we further propose the divergence-enhanced Transformer that can slow down the divergence decay in 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 Transformer-based framework for multi-view 3D object reconstruction. Different from the previous CNN-based methods that use 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 within our encoder-decoder Transformer structure.
- We investigate the problem of "divergence decay" in the proposed framework 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 recent CNN-based methods. Our method also shows better scaling capability on the number of input views.

2. Related Work

2.1. Multi-view 3D Reconstruction

Multi-view 3D object reconstruction has long been a research hot-spot in both computer vision and computer graphics. Traditional methods [30, 9] of this field are typically designed based on hand-crafted geometric features. Some representatives of early methods like Structure from Motion (SfM) [30], Simultaneous Localization and Mapping (SLAM) [9] can produce 3D reconstruction with satisfactory quality. However, they typically capture multiple images of the same object using well-calibrated cameras, which is not practical in some situations. Recently, CNN-based approaches, without requiring complex camera calibration, have gained increasing attention in 3D reconstruction [5, 12] and have shown promising results.

In CNN-based methods, a 2D-CNN single-view encoder, a 3D-CNN single-view decoder, and a multi-view fusion model are usually separately designed for 3D reconstruction. Among them, the fusion model plays a 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 pooling-based fusion only learns partial information of multiple views and ignores the view associations [11, 15]. The learnable weighted-sum fusion models are introduced to resolve these problems [26, 28, 27]. The RNN-based fusion methods [5, 12] can learn effective view-to-view relations but are computationally expensive and permutation-variant [22].

In this paper, different from the above CNN-based methods, we propose a Transformer-based 3D reconstruction method that unifies the feature extraction and view fusion in a single model and naturally explores the relationship between input views.

2.2. Transformer

In natural language processing, Transformer models have achieved great success in various tasks such as machine translation, text classification, and question answering [1]. The key to the Transformer is the multi-head self-attention operation, which aggregates features among every token pair of the embedding sequence. Recently, Transformer has also been successfully adapted to the computer vision field [2, 8, 4] and has shown promising application prospects. DETR [2] 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

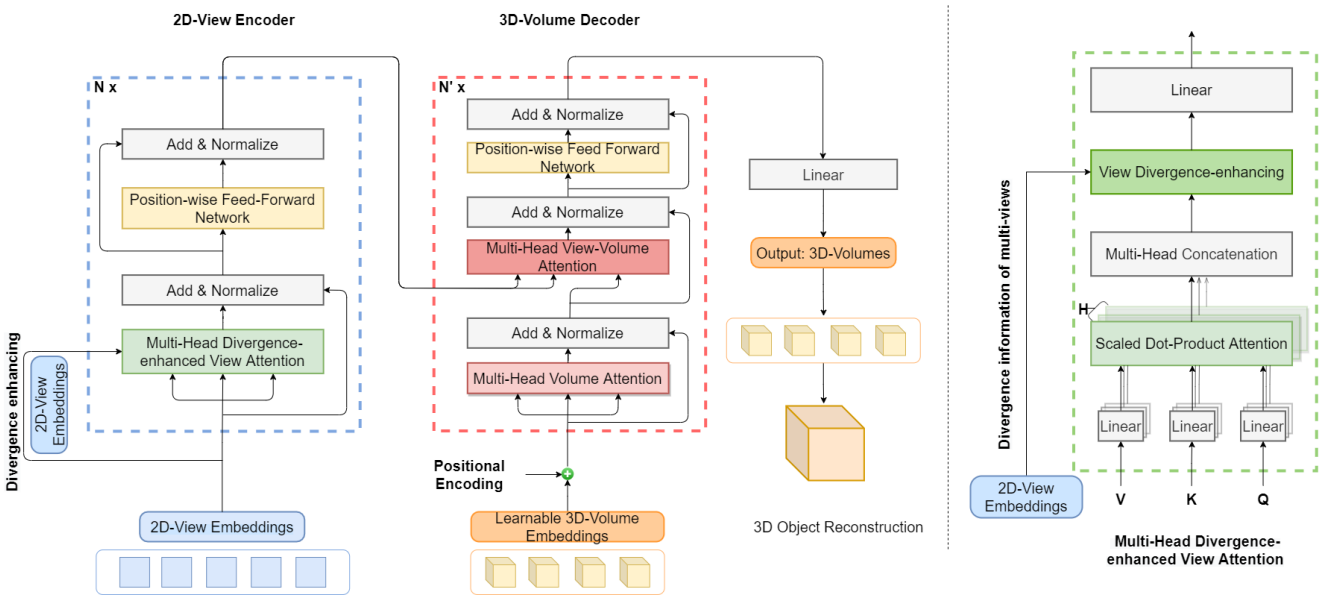

Figure 1. Illustration of EVolT for Multi-view 3D Object Reconstruction (left). The proposed view-divergence enhancing function in our EVolT (right).

sequence of tokens. ViT [8] applies Transformer directly to sequences of image patches for the image classification task without using CNN features and achieves comparable and even higher image classification accuracy when pretrained on a large-scale dataset.

In CNN-based multi-view 3D reconstruction methods, it is still challenging to design a fusion model that can explore the deep relationship between views while maintaining the permutation-invariant capability. A natural advantage of 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.

2.3. 3D Representation

There are different 3D representations, i.e., coordinate-based implicit representation [14, 18, 29, 24, 3], voxel-[17, 26] and mesh-based [23] representations. In terms of learning 3D representation, our method is voxel-based and trained across multi-scenes for 3D reconstruction instead of optimizing 3D scenes individually [17, 23, 14, 18]. Besides, our method is learned without requiring camera parameters, while coordinate-based implicit 3D representation methods are primarily for view synthesis and require camera parameters [14, 29, 24].

3. Methodology

The proposed 3D volume Transformer model, as shown in Figure 1, consists of a 2D-view encoder and a 3D-volume decoder. The inputs are multi-view images of an object. The 2D-view encoder encodes the relevant information among different views via view attention layers. The 3D-volume decoder learns global correlations of different spatial locations in volume attention layers and decodes the relationships between the view and spatial domains. In the decoder, we uniformly split the 3D space into a set of tokens as inputs. The predicted volumes for each token are finally stitched into the final 3D reconstruction as the output. The output contains occupancy voxel predictions where each defines the object-occupancy probability in its voxel.

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-Enhanced 3D Volume Transformer (**EVolT**).

- **VolT:** A baseline implementation of our method employing vanilla Transformer model and using standard VGG16 [16] 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 reconstruction.
- **EVolT:** A full implementation of our method adopting the proposed view-divergence enhancing function and using standard VGG16 [16] for 2D-view embeddings.

Here, to obtain 2D-view initial embeddings, we use a pretrained CNN shared among multiple views.

3.1. Divergence-enhanced 2D-view Encoder

Suppose $\mathcal{I} = \{\mathbf{I}^1, \mathbf{I}^2, \cdots, \mathbf{I}^M\}$ denotes the multi-view image set of an object to be reconstructed. For each view $\mathbf{I}^m$, we first use a pretrained view-shared CNN to obtain its 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 global relationships among multiple views using a series of self-attention layers. Here, to keep permutation invariant for the view sequence, the positional encodings of a standard Transformer are removed. We build our divergence-enhanced 2D-view encoder based on DETR [2] 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. (3)). The 2D-view encoder is formulated as follows:

$$\mathbf{X}_0 = [\mathbf{x}^1; \mathbf{x}^2; \cdots; \mathbf{x}^M], \tag{1}$$

$$\bar{\mathbf{X}}_l = \text{Norm}(\text{MH-DEAtt}(\mathbf{X}_{l-1}, \mathbf{X}_0) + \mathbf{X}_{l-1}), \tag{2}$$

$$\mathbf{X}_l = \text{Norm}(\text{FFN}(\bar{\mathbf{X}}_l) + \bar{\mathbf{X}}_l), \tag{3}$$

where "Norm" denotes layer normalization and l is the index of a basic block ($l = 1, \cdots, L$). The embeddings of the block 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, Eq. (5)) aggregates the feature representations among multiple views by learning view-to-view relationships. In the meantime, we propose a view-divergence enhancing function (DiView, Eq. (4)) 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:

$$\begin{aligned} &\text{MH-DEAtt}(\mathbf{X}_{l-1}, \mathbf{X}_0) = \text{DiView}(\mathbf{A}, \mathbf{X}_0)\mathbf{W}_{view}, \\ &\text{where}\mathbf{A} = \text{cat}(\mathbf{A}^1, \cdots, \mathbf{A}^H), \\ &\quad \mathbf{A}^h = \text{Attn}(\mathbf{Q}^h, \mathbf{K}^h, \mathbf{V}^h). \end{aligned} \tag{4}$$

Here, "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 $\mathbf{X}_{l-1}$ with the parameter matrix $\mathbf{W}_Q^h \in \mathbb{R}^{d\times d_k}$ ($\mathbf{Q}^h = \mathbf{X}_{l-1}\mathbf{W}_Q^h$). 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.

Specifically, in the Attention function "Attn", the output for a query is represented as an attention-score weighted sum of the values. The Attn function is formulated as

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}, \tag{5}$$

3.2. 3D-volume Decoder

The 3D-volume decoder learns the global correlation among different spatial locations and explores the relationship between the view and spatial domains. Given an object, we denote $[\mathbf{y}^1; \mathbf{y}^2; \cdots; \mathbf{y}^N]$ as a sequential 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. The 3D-volume embeddings, denoting a set of 3D sub-volumes for an object, are optimized end-to-end together with the network parameters and shared across all potential inputs (like object queries in [2]). These volume embeddings are not view-conditional variables but can provide a global prior for the dataset.

Positional encodings $\mathbf{E}^{pos}$ are added to 3D-volume embeddings to keep the position information in the spatial domain. Each positional encoding informs each sub-volume of its 3D spatial location in an object. It is calculated in a similar way to [2] using sine and cosine functions of different frequencies. 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:

$$\mathbf{Y}_0 = [\mathbf{y}^1; \mathbf{y}^2; \cdots; \mathbf{y}^N] + \mathbf{E}^{pos}, \tag{6}$$

$$\bar{\mathbf{Y}}_l = \text{Norm}(\text{MH-VolAttn}(\mathbf{Y}_{l-1}) + \mathbf{Y}_{l-1}), \tag{7}$$

$$\hat{\mathbf{Y}}_l = \text{Norm}(\text{MH-ViewVolAttn}(\bar{\mathbf{Y}}_l, \mathbf{X}_L) + \bar{\mathbf{Y}}_l), \tag{8}$$

$$\mathbf{Y}_l = \text{Norm}(\text{FFN}(\hat{\mathbf{Y}}_l) + \hat{\mathbf{Y}}_l), \tag{9}$$

where MH-VolAttn (in Eq.(7)) and MH-ViewVolAttn (in Eq.(8)) denote the multi-head volume attention layer and the multi-head view-volume attention layer, respectively.

In our decoder, the MH-VolAttn layer learns global dependencies among different 3D volumes as follows:

$$\begin{aligned} &\text{MH-VolAttn}(\mathbf{Y}_{l-1}) = \text{cat}(\mathbf{A}^1, \cdots, \mathbf{A}^H)\mathbf{W}_{vol}, \\ &\text{where } \mathbf{A}^h = \text{Attn}(\mathbf{Y}_{l-1}\mathbf{W}_Q^h, \mathbf{Y}_{l-1}\mathbf{W}_K^h, \mathbf{Y}_{l-1}\mathbf{W}_V^h). \end{aligned} \tag{10}$$

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

$$\begin{aligned} &\text{MH-ViewVolAttn}(\bar{\mathbf{Y}}_l, \mathbf{X}_L) = \text{cat}(\mathbf{A}^1, \cdots, \mathbf{A}^H)\mathbf{W}, \\ &\text{where } \mathbf{A}^h = \text{Attn}(\bar{\mathbf{Y}}_l\mathbf{W}_Q^h, \mathbf{X}_L\mathbf{W}_K^h, \mathbf{X}_L\mathbf{W}_V^h). \end{aligned} \tag{11}$$

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 decoder block L, we use a linear function to project the embeddings of each 3D volume to their

Table 1. Parameter sizes and pretrained CNNs for the initial 2D-view embeddings in competing methods.

	Param. (M)	Pretrained CNN used for 2D-view embeddings
Pix2Vox-A [26]	114.24	VGG16 [16]
Pix2Vox++/A [27]	96.31	ResNet50 [10]
VolT	28.63	VGG16 [16]
VolT+	96.76	2D-CNN+3D-DCNN
EVolT	29.03	VGG16 [16]

3D output space. Then the predicted 3D volumes are reshaped and grouped to the reconstruction output. We use binary cross-entropy between the voxel label and the output as our loss function.

4. Experiment

4.1. Dataset

We utilize the ShapeNet dataset [25] to evaluate the proposed methods and other compared methods. We follow 3D-R2N2 [5] 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 the object. For each category, we follow [27] and randomly split samples into 70% training, 10% validation, 20% test. Categories in training are the same as those in evaluation. During training, the input-view number can be varied.

4.2. Evaluation Metrics

4.2.1 IoU

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

$$\text{IoU} = \frac{\sum_{(i,j,k)} \text{I}(y(i,j,k) > t)\text{I}(\bar{y}(i,j,k))}{\sum_{(i,j,k)} \text{I}[\text{I}(y(i,j,k) > t) + \text{I}(\bar{y}(i,j,k))]}, \quad (12)$$

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.

4.2.2 F-Score

Compared with IoU, F-score [19, 27] 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:

$$\text{F-Score}(d) = \frac{2\text{P}(d)\text{R}(d)}{\text{P}(d) + \text{R}(d)}, \quad (13)$$

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:

$$\text{P}(d) = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{r} \in \mathcal{R}} [\min_{\mathbf{g} \in \mathcal{G}} \|\mathbf{r} - \mathbf{g}\| < d], \quad (14)$$

$$\text{R}(d) = \frac{1}{|\mathcal{G}|} \sum_{\mathbf{g} \in \mathcal{G}} [\min_{\mathbf{r} \in \mathcal{R}} \|\mathbf{g} - \mathbf{r}\| < d]. \quad (15)$$

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 [27].

4.2.3 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 mean absolute deviation (MAD) measuring the similarity of multi-view attentions, is calculated as $D = \frac{1}{N_{view}} \sum_m \|\mathbf{s}_m - \bar{\mathbf{s}}\|_2$, where $\bar{\mathbf{s}} = \frac{1}{N_{view}} \sum_m \mathbf{s}_m$. Here, a small D means a more considerable similarity and the convergence of multi-view representations.

4.3. 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 [13] with a β_1 of 0.9 and a β_2 of 0.999.

Table 1 shows the parameter sizes and pretrained CNNs for the initial view embeddings used in different competing methods. Compared with Pix2Vox-A [26] and Pix2Vox++/A [27], the parameter size of EVolT is only around 30% of them. To obtain the reported best results,

Table 2. Comparison of 24-views reconstruction on ShapeNet using IoU and F-Score. The best score for each category is in bold.

	24-views IoU					24-views F-Score@1%				
Category	Pix2-Vox-A	Pix2-Vox++/A	VolT	VolT+	EVolT	Pix2-Vox-A	Pix2-Vox++/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 3. 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 [5]	-	-	-	-	0.383	-	0.382	-	0.382	0.383	-	0.378
AttSets [28]	-	-	-	-	0.448	-	0.447	-	0.445	0.444	-	0.430
Pix2Vox-A [26]	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 [27]	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 [5]	-	-	-	-	0.636	-	0.636	-	0.636	0.635	-	0.625
AttSets [28]	0.694	-	-	-	0.693	-	0.692	-	0.688	0.685	-	0.675
Pix2Vox-A [26]	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 [27]	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

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 impact of 2D-view embeddings on our Transformer-based framework, 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.

4.4. Multi-view 3D Object Reconstruction

4.4.1 Quantitative results

Here, we show the quantitative results of compared methods on ShapeNet using different evaluation metrics. Table 2 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 among the compared methods. VolT gets moderate 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 EVolT is 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 3 shows the multi-view object reconstruction results on ShapeNet. The best score for each number of views (column label) is highlighted in bold. This table shows that the performances of our methods increase appreciably as the number of views increases. In comparison, other com-

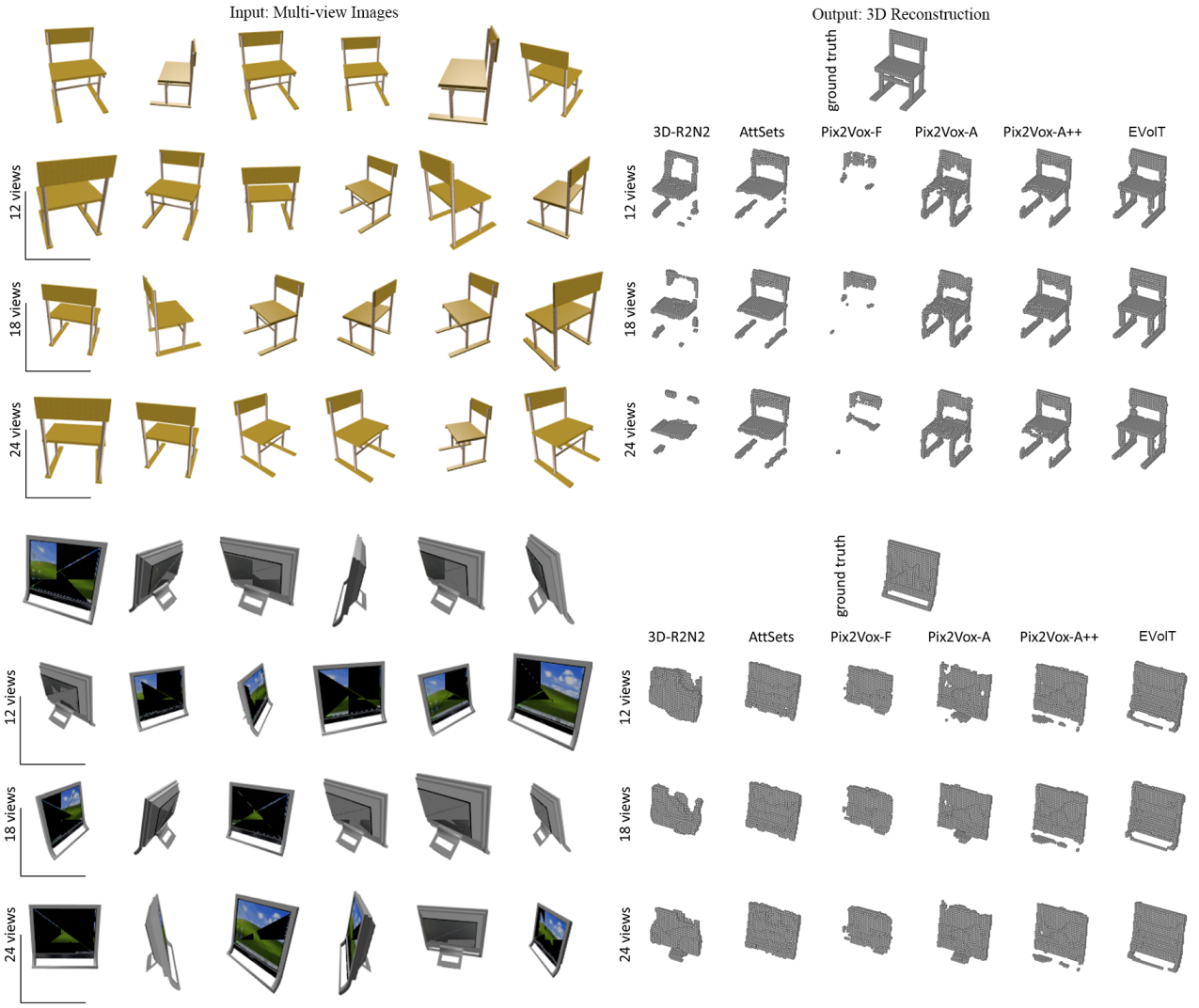

Figure 2. Qualitative 3D object reconstruction results on ShapeNet based on different number of input 2D-view images.

pared 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 have 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 than 12.

4.4.2 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 numbers of input views, i.e., 12 views, 18 views,

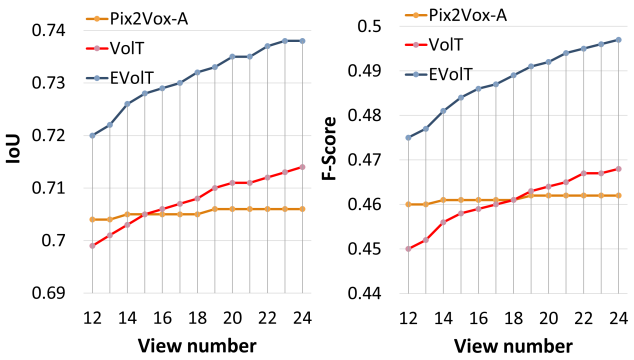

Figure 3. Effect of the view-divergence enhancing function on 3D reconstruction results.

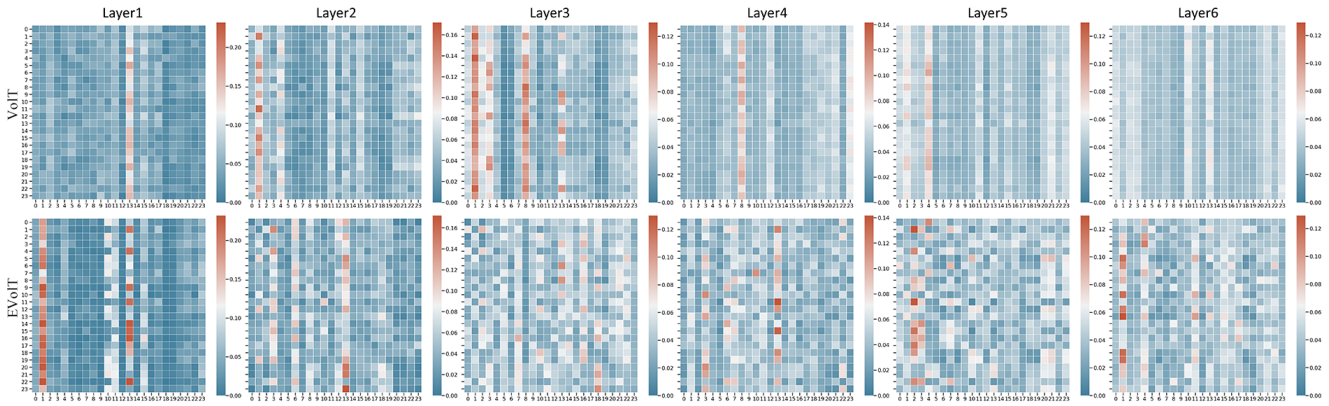

Figure 4. Multi-view attention-matrix visualization in VolT and EVolT.

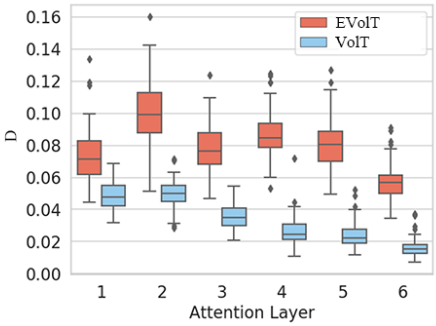

Figure 5. Divergence among multi-view representations in VolT and EVolT.

and 24 views. The first two rows on the left part of Figure 2 show the 12 input views of an object. The corresponding reconstruction results of competing methods are shown in the second row on the right. Similarly, the first three rows on the left 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.

4.5. Ablation Study

4.5.1 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.

4.5.2 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 enhancing function in EVolT can effectively slow down the convergence degradation of multi-views in deeper layers.

In Figure 5, the similarity measurement score D is also recorded to analyze the convergence among multi-view representations in each layer. For 100 randomly chosen objects, we plot D in different layers displayed in Figure 5. A small D suggests a significant convergence among 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 among multi-view representations in different layers.

5. Conclusion

This paper proposes a Transformer-based framework for multi-view 3D reconstruction, which achieves state-of-the-art accuracy on ShapeNet with fewer parameters than CNN-based methods. The proposed framework explores view and spatial domain relationships for multi-view 3D reconstruction. We also explore the problem of divergence decay for the multi-view information in deeper layers and propose a view-divergence enhancing function to ease such a problem.

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 2

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 2, 4

[3] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-GAN: Periodic implicit generative adversarial networks for 3D-aware image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5799–5809, 2021. 3

[4] 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

[5] 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. 1, 2, 5, 6

[6] 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. 1

[7] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. *arXiv preprint arXiv:2103.03404*, 2021. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 1, 2, 3

[9] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial intelligence review*, 43(1):55–81, 2015. 2

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[11] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. DeepMVS: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2821–2830, 2018. 2

[12] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Advances in Neural Information Processing Systems*, volume 30, 2017. 1, 2

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 5

[14] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 3

[15] Despoina Paschalidou, Osman Ulusoy, Carolin Schmitt, Luc Van Gool, and Andreas Geiger. RayNet: Learning volumetric 3D reconstruction with ray potentials. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3897–3906, 2018. 2

[16] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 3, 5

[17] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhofer. Deepvoxels: Learning persistent 3D feature embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2437–2446, 2019. 3

[18] Vincent Sitzmann, Michael Zollhoefer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 3

[19] Maxim Tatarchenko, Stephan R. Richter, Rene Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3D reconstruction networks learn? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[20] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 1

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. 1

[22] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. Order matters: Sequence to sequence for sets. In *International Conference on Learning Representations (ICLR)*, 2016. 1, 2

[23] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2Mesh: Generating 3D mesh models from single RGB images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018. 3

[24] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. IBRNet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 3

[25] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 5

[26] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2Vox: Context-aware 3D reconstruction from single and multi-view images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2690–2698, 2019. 1, 2, 3, 5, 6

[27] Haozhe Xie, Hongxun Yao, Shengping Zhang, Shangchen Zhou, and Wenxiu Sun. Pix2Vox++: multi-scale context-aware 3D object reconstruction from single and multiple images. *International Journal of Computer Vision*, 128(12):2919–2935, 2020. 1, 2, 5, 6

[28] Bo Yang, Sen Wang, Andrew Markham, and Niki Trigoni. Robust attentional aggregation of deep feature sets for multi-view 3D reconstruction. *International Journal of Computer Vision*, 128(1):53–73, 2020. 1, 2, 6

[29] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4578–4587, June 2021. 3

[30] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion. *Acta Numerica*, 26:305–364, 2017. 2