MULTI-VIEW 3D RECONSTRUCTION FROM VIDEO WITH TRANSFORMER

Yijie Zhong, Zhengxing Sun, Yunhan Sun, Shoutong Luo, Yi Wang, Wei Zhang

State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing China

ABSTRACT

Multi-view 3D reconstruction is the base for many other applications in computer vision. Video provides multi-view images and temporal information, which can help us better complete the reconstruction goal. Redundant information handling in video and multi-view feature extraction and fusion become the key issues in the shape prior extraction for reconstruction. In this paper, inspired by the recent great success in Transformer models, we propose a transformer-based 3D reconstruction network. We formulate the multi-view 3D reconstruction into three parts: frame encoder, fusion module, and shape decoder. We apply several special used tokens and perform the fusion progressively in the encoder phase, called patch-level progressive fusion module. These tokens describe which part of the object the frame should focus on and the local structural detail progressively. Then we further design a transformer fusion module to aggregate the structure information. Finally, multi-head attention is utilized to build the transformer-based decoder to reuse the shallow features from encoder. In experiments not only can ours method achieve competitive performance, but it also has low model complexity and computation cost.

***Index Terms*—** Multi-view 3D reconstruction, Sequential modeling, Transformer-based model

1. INTRODUCTION

Image-based 3D reconstruction is a long-standing and important problem in computer vision, and it is also the key to computer perception of the real world [1, 2]. Many recent works focus on reconstruction tasks using only a single image [3, 4], but there is a serious ambiguity in a single image. So single-image 3D reconstruction has long been an ill-posed problem [5]. Another part of the works is based on the input of multiple images to extract the shape prior for 3D reconstruction, where how to fuse the information from different images and accomplish the final shape prediction is the core of each method. We consider video sequences as a special form of multiple images, in which the temporal information can help us to better complement and interact with frames. And at the same time the video will also bring a lot of redundant information. Therefore, **how to make full use of the video sequences and complete the information fusion between images** and **how to deal with the redundant information in the video to complete the reconstruction task efficiently** are two main issues in our extraction of shape prior.

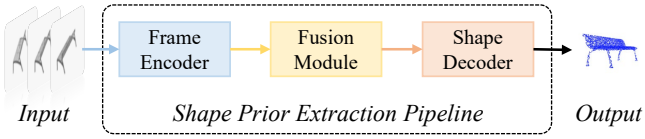

Fig. 1. Extracting the shape prior from multi-view images is split into three important parts.

We divide the whole process of shape prior extraction into three processes, as shown in Fig. 1. First, a video encoder encodes the image sequences to obtain the embedding vectors for each frames. Then a fusion module follows to obtain the embedding, which describes the 3D shape, for the input object based on the complementary information of all images. Finally, the shape decoder finishes the prediction of the shape to get the three-dimensional output.

For the first step of encoding the input images, existing methods [6, 7] usually apply a shared encoder to encode each image independently, and the the information fusion is completely dependent on the subsequent fusion module. The bias and loss of information generated in this step will have an impact on the next processing. There are also methods that use a large encoder to encode all input images at once [8, 9], which performs sufficient information interaction but also results in a large amount of redundant computation. They do not handle redundant information in image sequences properly. In this paper, we propose an efficient multi-image encoder to process input video. Recently Transformer has shown its great advantage in various tasks in computer vision [10, 11, 12], and its ability to capture long-range dependence is of great help to multi-view 3D reconstruction. The information interaction between images is mainly reflected in the fact that **a part of the object may appear only in one view point and is not visible in another view**. The Tokens (patches) sliced from the image in Visual Transformer are also well suited for local information completion in our task. And the use of task-wise tokens can well reduce the model size and the amount of computation. Thus, we propose a transformer-based patch-level progressive fusion (PLPF) module. The Frame Tokens in it extracts the information of each frame after the initial fusion, making them focus on the parts that can be accurately

predicted under their respective perspectives. Then Structure Tokens extracts the embedding of the object part of interest in each frame and they are used for subsequent fusion module.

For the second fusion step, some methods [13] use global pooling operations to get a simple global feature, but such operations cause the loss of a large amount of information. And they default to the same weight for each frame, while in fact the information presented in different views contributes differently to the final shape predictions. Some methods [14, 15] again use recurrent neural networks (RNN) to process image sequences, but this results in the overall network not being able to execute in parallel and is less efficient. In addition RNN favors short-range dependence and is difficult to handle long-range relationships. For video input, the information at shorter intervals is more redundant, and the overall shape of the object can only be observed when the difference in viewpoint increases, making long-range dependence more important in our task. Thus, we propose a transformer-based fusion module. We use several task-wise tokens to stand for the final fusion results, which represent the shape of the target object, to achieve efficient and accurate integration.

For the final decoding step, the simplest way is to finish the prediction directly using multi-layer perceptron (MLP), which leads to low quality because a single global feature that has lost detail information. PSG [16] proposed an hourglass decoding approach that introduces multi-scale features from the encoding phase. However it is only designed for single image input. There are also many methods [6, 17] that add refine modules to refine the predicted shapes after the common encode-decode network to improve the accuracy. Such methods are only able to obtain small enhancement because they do not solve the essential problem in the feature encoding and fusion stages, making the extracted features with more information about the shape of the object. According to this, we propose a transformer-based decoder module with the help of previous mentioned Frame Token, Structure Token, and Task-wise Token. In addition to the simple decoding of task-wise tokens by using MLP, we add two branches to optimize them using the multi-head attention module. We use tokens from encoder as *key* and *query* and task-wise tokens as *value* to further dig the valid information for the shape prediction from the video input. The small number of tokens ensures that no large computational cost is incurred in this process.

2. METHOD

In this section, we explain our proposed transformer-based multi-view 3D reconstruction method. We describe frame encoder, fusion module, and the decoder in detail, respectively.

2.1. Patch-level Progressive Fusion Encoder

The structure of the commonly used encoder type and our proposed encoder can be seen in Fig. 2. A shared convolutional neural network (CNN) is popularly used as a backbone

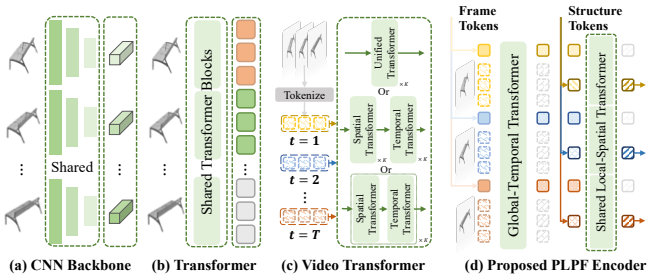

Fig. 2. Different ways to encode the input video sequence. Ours patch-level progressive fusion encoder is shown in (d).

in multi-view 3D reconstruction, as shown in Fig. 2(a). Recently, Transformer demonstrates its powerful encoding capabilities. We also try on a shared vision transformer backbone [18, 19], as shown in Fig. 2(b). They simply stack several convolution layers or transformer blocks, so this encoding type does not take advantage of multiple images input and represents an image with only a single feature which loses lots of information and local details. Some works [20, 21] have also used the Transformer architecture for multi-image or video encoding, which can be seen in Fig. 2(c), so we also try to use this type of encoder structure to observe its information interaction performance for 3D reconstruction task. It is worth noting that they always have large number of parameters. In Fig. 2(d), for more efficient feature extraction and interaction, we propose a patch-level progressive fusion (PLPF) encoder. In different stages of our proposed encoder, we use special tokens to deal with the redundant information and also apply different fusion strategies.

Let $\{I_1, I_2, \cdots, I_n\}$ denote the input video sequences which contains n frames, where I_i represents a single image. Then we can get the tokens $T_i = \{T_i^1, T_i^2, \cdots, T_i^m\}$ of i-th frame, where $m = HW/p^2$ and $T_i^j \in \mathbb{R}^{3p^2}$. H and W represent the height and the width of input images. p means the size of a patch to construct a token. In order to make each frame individually focus on the parts that can be more accurately predicted from its own perspective, we add a Frame Token F_i for each frame. Then we perform a global temporal transformer to interact with all input frames and conduct the first-time fusion. This operation can be considered as:

$$G_{out} = \Phi_{global}([T_1, F_1; T_2, F_2; \cdots; T_n, F_n]), \qquad (1)$$

where Φ_{global} means the global transformer and G_{out} has the same size of the input features. At this point we get the embedding of each frame F_i, which contains information about each frame itself as well as auxiliary information about that perspective from other perspectives. Thus we only use $\{F_i\}_{i=1}^n$ for the subsequent process, which also reduces the computation cost. Then we can work on each frame to get a shape embedding for each frame for its part of interest. Specifically, we add several Structure Token S_i for each frame and use a shared local spatial transformer to handle the

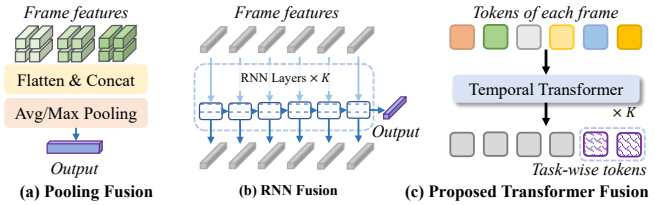

Fig. 3. Different ways to fuse the information from images.

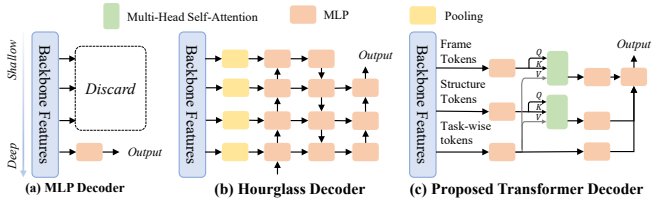

Fig. 4. Different ways to decode the final shape prediction.

shape structure inside each frame, which can be described as:

$$L_{out} = [\Phi_{local}(F_1, S_1); \cdots; \Phi_{local}(F_n, S_n)], \quad (2)$$

where Φ_{local} means the local transformer and L_{out} has the same size of the input in this stage. Finally, we achieve the features of different parts of the objects and $\{S_i\}_{i=1}^n$ in L_{out} is considered as the input of the next fusion module.

2.2. Transformer Fusion Module

As shown in Fig. 3(a), a normal fusion approach flattens the features of each frame into one-dimension and concatenates them. Then a global pooling operation is used to obtain a uniform feature representation of the input object. This approach results in the loss of local details and does not take into account the difference between frames. RNN (Fig. 3(b)) becomes a sound alternative solution. The output of the recurrent unit in the last frame is considered as the final fusion feature. However, it operates inefficiently and is more difficult to capture long-range dependence. To solve this problem, we propose a transformer fusion module with several additional task-wise tokens for reconstruction in Fig. 3(c).

First, we add the position embedding on the input S_i of each frame to emphasize the frame order. Then, we add several task-wise tokens W_i for each frame and use a k-layers transformer for information fusion and interaction. In the final output, we discard S_i of each frame and use only task-wise tokens W_i for subsequent process to achieve higher efficiency.

2.3. Transformer-based Decoder

Decoding in 3D reconstruction is usually performed using MLP or 3D deconvolution layers, as shown in Fig. 4(a). They can no longer use shallow features to guide the prediction process. Using only a single global feature for prediction also leads to a decrease in accuracy. Thus, PSG [16] proposed hourglass decode structure, which conducts the encode-decode operations recurrently. It has stronger representation power and can mix global and local information evidences. However, this structure cannot be applied directly here as the multi-image encoder has multiple output features. So, we design a transformer-based decoder combined with the output of our previous network.

To use the guidance of shallow features, we take out the Frame Token T_i and Structure Token S_i. The multi-head attention mechanism is very effective in interacting with features and updating the input features. Therefore, we use the previously extracted $T = \{T_i\}_{i=1}^n$ and $S = \{S_i\}_{i=1}^n$ to guide and interact with the Task-wise Token $W = \{W_i\}_{i=1}^n$ which is responsible for the final reconstruction. We consider the guidance tokens as *key* and *query* in multi-head attention to update W as the *value*. A single head in the vanilla multi-head attention (MHA) can be written as:

$$Attention(Q, K, V) = softmax(QK^T)V. \quad (3)$$

In our proposed decoder, it can be considered as:

$$Attention(S, S, W) = (softmax(S^T S)W^T)^T, \quad (4)$$

where $S \in \mathbb{R}^{n \times d}$ and $W \in \mathbb{R}^{k \times d}$, where d is feature dimension and k is the number of tokens. We perform the same on T. This operation further allows each token to enhance the object part it focuses on. Finally, we concatenate all the output tokens to predict the 3D shape by using several fully connected layers, as shown in Fig. 3(c).

3. EXPERIMENT

3.1. Experimental setup

Dataset. We evaluate our method on 3D models from the commonly used ShapeNet [22]. We use a subset of it, which contains 13 shape classes. The dataset split strategy follow the setup of [14]. We use random lighting to render 30 frames of motion sequence at random initial position.

Evaluation metric. We use the Chamfer distance (CD) as our main evaluation metric, since it has been shown to be well correlated with human judgment of shape similarity.

Implementation details. In this paper, we use the popular PyTorch framework to implement our method. We use the Adamw optimizer. The weight decay is set to 0.05 for loss optimization. The learning rate is set to 5e-4. We apply the warmup and cosine learning rate strategy. The batch size is set to 24 and the input images are resized to 64×64.

Compared methods. For the CNN backbone, we apply the ResNet [23] series. For the image Transformer backbone, we apply the PVT [18] and PVT-v2 [19] series. For the video Transformer backbone, we apply the ViViT [20] and MViT [21] series. For the RNN, we use the LSTM unit.

3.2. Comparison experiments

The improvement not only come from Transformer. As can be seen in Table 1, we find that in each series as the

Table 1. Chamfer Distance for different encoder backbone. All the experiments are with Pooling fusion and MLP decoder. The best of each backbone series are in underline. The best of all results are in **bold**. The values are reported multiplied by 100.

Encoder	airplane	bench	cabinet	car	chair	lamp	monitor	rifle	sofa	speaker	table	telephone	vessel	Mean	Param.	FLOPs
ResNet-18	2.69	6.79	8.08	3.94	5.76	10.38	**7.33**	3.23	6.99	10.01	9.97	5.00	3.44	6.43	34.52M	4.52G
ResNet-34	**2.68**	6.84	7.95	**3.93**	5.79	10.25	**7.33**	3.22	7.03	10.01	9.58	4.99	3.44	6.39	44.63M	9.05G
ResNet-50	2.69	6.79	8.10	**3.93**	5.79	9.43	7.37	**3.21**	7.04	10.10	9.59	**4.95**	**3.43**	6.34	47.64M	10.24G
PVT-tiny	2.73	6.78	7.96	3.98	5.76	9.90	7.46	3.31	7.05	9.96	9.91	5.03	3.46	6.41	36.06M	4.59G
PVT-small	2.69	6.77	7.98	**3.93**	5.76	9.42	7.56	3.28	7.01	10.04	9.90	5.01	3.45	6.37	47.31M	9.03G
PVT-medium	2.77	6.79	8.02	3.94	5.99	9.74	7.41	3.24	7.05	10.01	9.96	5.30	3.51	6.44	67.04M	15.81G
PVT-v2-b0	2.69	6.83	8.03	3.94	5.76	9.42	7.45	3.22	**6.96**	**9.90**	**9.57**	5.06	3.46	6.33	26.62M	1.34G
PVT-v2-b1	2.71	6.75	8.10	3.95	**5.73**	9.45	7.37	3.23	6.97	9.93	9.70	5.04	3.44	6.34	36.84M	5.04G
PVT-v2-b2-li	2.69	**6.72**	**7.96**	3.94	5.74	**9.40**	7.43	3.24	6.98	10.06	9.59	5.02	3.44	**6.32**	48.19M	9.59G

Table 2. Chamfer Distance for different ways of fusion and decoder. The best results other than our method are in underline. The best results are in **bold**. The values are reported multiplied by 100.

Encoder	Fusion	Decoder	airplane	bench	cabinet	car	chair	lamp	monitor	rifle	sofa	speaker	table	telephone	vessel	Mean	Param.	FLOPs
	Pooling	MLP	2.69	6.79	8.08	3.94	5.76	10.38	7.33	3.23	6.99	10.01	9.97	5.00	3.44	6.43	34.52M	4.52G
ResNet-18	RNN	MLP	2.64	6.75	8.03	3.91	5.60	9.32	7.29	3.21	6.87	9.75	9.61	4.98	3.41	6.26	47.12M	4.96G
	Trans	MLP	2.61	6.71	8.00	3.83	5.54	9.12	7.26	3.17	6.84	9.72	9.41	4.96	3.34	6.19	34.54M	4.58G
	Pooling	MLP	2.73	6.78	7.96	3.98	5.76	9.90	7.46	3.31	7.05	9.96	9.91	5.03	3.46	6.41	36.06M	4.59G
PVT-tiny	RNN	MLP	2.61	6.76	7.79	3.86	5.69	8.91	7.33	3.23	6.82	9.57	9.85	4.97	3.32	6.21	48.66M	5.03G
	Trans	MLP	2.60	6.62	7.73	3.82	5.50	8.90	7.00	3.15	6.81	9.51	9.40	4.90	3.31	6.10	36.07M	4.65G
	Pooling	MLP	2.69	6.83	8.03	3.94	5.76	9.42	7.45	3.22	6.96	9.90	9.57	5.06	3.46	6.33	26.62M	1.34G
PVT-v2-b0	RNN	MLP	2.65	6.81	7.86	3.89	5.68	8.94	7.33	3.21	6.80	9.48	9.47	4.90	3.42	6.19	39.22M	1.78G
	Trans	MLP	2.62	6.69	7.81	3.85	5.54	8.84	7.31	3.20	6.78	9.47	9.36	4.87	3.39	6.13	26.64M	1.41G
MViT-base	Trans	-	2.61	6.68	7.85	3.84	5.58	8.97	7.24	3.17	6.88	9.65	9.44	4.95	3.34	6.17	59.52M	8.70G
ViViT-Model1	Trans	-	2.63	6.68	7.79	3.84	5.59	9.19	7.24	3.15	6.81	9.70	9.43	4.92	3.33	6.18	26.05M	1.56G
ViViT-Model2	Trans	-	2.63	6.73	7.77	3.83	5.59	9.14	7.20	3.16	6.85	9.71	9.43	4.97	3.34	6.18	28.94M	1.65G
Ours	Trans	MLP	2.51	6.58	7.66	3.71	5.39	8.82	6.86	3.13	6.63	9.34	9.26	4.73	3.21	5.99	12.37M	1.26G
Ours	Trans	Trans	**2.40**	**6.44**	**7.57**	**3.66**	**5.24**	**8.63**	**6.70**	**3.05**	**6.47**	**9.16**	**9.13**	**4.61**	**3.09**	**5.86**	15.00M	1.32G

depth of the network increases, the performance improvement is very limited, although it is probable to be improved. Comparing the CNN with Transformer encoder, we find that the great encoding power of Transformer structure for single-image tasks is not fully demonstrated at this point. So a simple application of Transformer will not meet our needs. Considering the model complexity, we use the network with the smallest version in each series in the other experiments.

The need for fusion in the encoding phase. We conduct experiments on recent video Transformer network. The results are shown in the fourth part of Table 2. It can be seen that the use of encoders designed for video sequences has resulted in a significant performance improvement. This also shows that it is necessary to perform feature fusion at the encoding phase. And ViViT and MViT are not designed for the 3D reconstruction task we are targeting.

Different fusion module. In Table 2, we replace the fusion method with RNN and our proposed Transformer fusion module for our experiments. Instead of a simple fusion approach (Pooling), uniquely designed fusion modules can often be found to achieve substantial performance gains. Our proposed module use less computation cost to get better fusion performance compared to RNN. This benefits from capturing the long-range dependence and the task-wise tokens.

Our proposed Transformer-based model. Benefit from our proposed PLPF encoder, Transformer fusion module, and the Transformer-based decoder, we achieve a state-of-the-art performance against current methods with low computation cost and model complexity. The results can be seen in Table 2.

4. CONCLUSION

In this paper, we divide the shape prior extraction for multi-view 3D reconstruction into three important parts: frame encoder, fusion module, and shape decoder. And we build an efficient Transformer-based model for video input. Specifically, we propose PLPF encoder, a transformer fusion module, and a transformer decoder. The frame tokens and structure tokens in PLPF describe which part of the object the frame should focus on and the local structural detail progressively. And these tokens minimize model complexity. The task-wise tokens in transformer fusion module aggregate the shape information of the object from structure tokens. The decoder applies the multi-head attention to utilize the local information from encoder. Experiments show the great sequential modeling ability with fewer parameters of our proposed method.

Acknowledgement. This work is supported by: The National Natural Science Foundation of China No.42075139, 42077232; The Science and technology program of Jiangsu Province No.BE2020082;The Innovation Fund of State Key Laboratory for Novel Software Technology No.ZZKT2022A18.

5. REFERENCES

[1] George Fahim, Khalid Amin, and Sameh Zarif, "Single-view 3d reconstruction: A survey of deep learning methods," *Comput. Graph.*, vol. 94, pp. 164–190, 2021.

[2] 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 TPAMI*, vol. 43, no. 5, pp. 1578–1604, 2021.

[3] Eldar Insafutdinov and Alexey Dosovitskiy, "Unsupervised learning of shape and pose with differentiable point clouds," in *NeurIPS*, 2018, pp. 2807–2817.

[4] Priyanka Mandikal and Venkatesh Babu Radhakrishnan, "Dense 3d point cloud reconstruction using a deep pyramid network," in *WACV*. 2019, pp. 1052–1060, IEEE.

[5] Yunjie Wu, Zhengxing Sun, Youcheng Song, Yunhan Sun, Yijie Zhong, and Jinlong Shi, "Shape-pose ambiguity in learning 3d reconstruction from images," in *AAAI*. 2021, pp. 2978–2985, AAAI Press.

[6] Haozhe Xie, Hongxun Yao, Shengping Zhang, Shangchen Zhou, and Wenxiu Sun, "Pix2vox++: Multi-scale context-aware 3d object reconstruction from single and multiple images," *Int. J. Comput. Vis.*, vol. 128, no. 12, pp. 2919–2935, 2020.

[7] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang, "Pix2vox: Context-aware 3d reconstruction from single and multi-view images," in *ICCV*. 2019, pp. 2690–2698, IEEE.

[8] Dan Wang, Xinrui Cui, Xun Chen, Zhengxia Zou, Tianyang Shi, Septimiu Salcudean, Z. Jane Wang, and Rabab Ward, "Multi-view 3d reconstruction with transformer," *CoRR*, vol. abs/2103.12957, 2021.

[9] Zai Shi, Zhao Meng, Yiran Xing, Yunpu Ma, and Roger Wattenhofer, "3d-retr: End-to-end single and multi-view 3d reconstruction with transformers," *CoRR*, vol. abs/2110.08861, 2021.

[10] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020, vol. 12346, pp. 213–229.

[11] 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 *ICLR*. 2021, OpenReview.net.

[12] Salman H. Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah, "Transformers in vision: A survey," *CoRR*, vol. abs/2101.01169, 2021.

[13] Zhaoliang Lun, Matheus Gadelha, Evangelos Kalogerakis, Subhransu Maji, and Rui Wang, "3d shape reconstruction from sketches via multi-view convolutional networks," in *3DV*. 2017, pp. 67–77, IEEE Computer Society.

[14] 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 *ECCV*, 2016, vol. 9912, pp. 628–644.

[15] Abhishek Kar, Christian Häne, and Jitendra Malik, "Learning a multi-view stereo machine," in *NIPS*, 2017, pp. 365–376.

[16] Haoqiang Fan, Hao Su, and Leonidas J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *CVPR*. 2017, pp. 2463–2471, IEEE Computer Society.

[17] Kaiqi Wang, Ke Chen, and Kui Jia, "Deep cascade generation on point sets," in *IJCAI*, 2019, pp. 3726–3732.

[18] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," *CoRR*, vol. abs/2102.12122, 2021.

[19] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao, "Pvtv2: Improved baselines with pyramid vision transformer," *CoRR*, vol. abs/2106.13797, 2021.

[20] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid, "Vivit: A video vision transformer," *CoRR*, vol. abs/2103.15691, 2021.

[21] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer, "Multiscale vision transformers," *CoRR*, vol. abs/2104.11227, 2021.

[22] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing 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," *CoRR*, vol. abs/1512.03012, 2015.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.