Analysis-by-Synthesis Transformer for Single-View 3D Reconstruction Supplementary Material

Dian Jia[1], Xiaoqian Ruan[1], Kun Xia[1,2], Zhiming Zou[1], Le Wang[2], and Wei Tang[1]

[1] University of Illinois Chicago, Chicago, IL, USA
{djia7,xruan9,zzou6,tangw}@uic.edu
[2] Xi'an Jiaotong University, Xi'an, Shaanxi, P.R. China
xiakun@stu.xjtu.edu.cn
lewang@xjtu.edu.cn

1 More Model Details

1.1 Scaled Dot-Product Attention

We first have a brief review of the standard scaled dot-product attention [7]. The input consists of query vectors, key vectors with the dimension of D^{qk}, and value vectors with the dimension of D^{v}. To handle multiple query, key, and value vectors in parallel, the vectors are packed into matrixes $\boldsymbol{Q} = [\boldsymbol{q}_i \in \mathbb{R}^{D^{\mathrm{qk}}} : i = 1, \ldots, N^{\mathrm{q}}]$, $\boldsymbol{K} = [\boldsymbol{k}_i \in \mathbb{R}^{D^{\mathrm{qk}}} : i = 1, \ldots, N^{\mathrm{kv}}]$, $\boldsymbol{V} = [\boldsymbol{v}_i \in \mathbb{R}^{D^{\mathrm{v}}} : i = 1, \ldots, N^{\mathrm{kv}}]$. The scaled dot-product attention is defined as:

$$\texttt{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \texttt{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{D^{\mathrm{qk}}}}\right)\boldsymbol{V} \tag{1}$$

The attention function can not capture the order information of the input, or more specifically, the pixel location. Therefore, we follow the prior work to add the positional embedding [7] to the queries and keys.

1.2 Padding Mask

Padding masks are used in the original Transformer to mask out the semantically meaningless part of the input [7]. In our work, we adopt the padding mask to mask out the background pixels from the input pixel sequence. We convert the predicted saliency map $\boldsymbol{S} \in [0, 1]^{H \times W}$ into a padding mask $\boldsymbol{M}$ with a dimension of WH. Let s_i and m_i respectively denote the ith element of $\boldsymbol{S}$ and $\boldsymbol{M}$. The padding mask is constructed as:

$$m_i = \begin{cases} 0 & \text{if } s_i \geqslant \alpha \\ -\infty & \text{if } s_i < \alpha \end{cases} \tag{2}$$

where α is a thresholding hyper-parameter which is set to 0.5. Then, the masked version of the scaled dot-product attention can be defined as:

$$\texttt{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}, \boldsymbol{M}) = \texttt{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{D^{\mathrm{qk}}}} + \boldsymbol{M}\right)\boldsymbol{V} \tag{3}$$

where the attention weights corresponding to the masked positions will be zero after the softmax function.

1.3 Multi-Head Attention

The research in Transformers has proved that it is beneficial to project the queries, keys, and values into different sub-spaces through multiple projection heads [1,7]. Let N be the number of heads. The output of the ith attention head is calculated as:

$$\boldsymbol{H}_i = \texttt{Attention}\left(\boldsymbol{Q}\boldsymbol{W}_i^{\mathrm{q}}, \boldsymbol{K}\boldsymbol{W}_i^{\mathrm{k}}, \boldsymbol{V}\boldsymbol{W}_i^{\mathrm{v}}, \boldsymbol{M}\right) \tag{4}$$

where $\boldsymbol{W}_i^{\mathrm{q}}$, $\boldsymbol{W}_i^{\mathrm{k}}$, $\boldsymbol{W}_i^{\mathrm{v}}$ are projection matrices with the shape of $D^{\mathrm{qk}} \times \hat{D}^{\mathrm{qk}}$, $D^{\mathrm{qk}} \times \hat{D}^{\mathrm{qk}}$, $D^{\mathrm{v}} \times \hat{D}^{\mathrm{v}}$, respectively. In practice, we adopt $\frac{D^{\mathrm{qk}}}{N}$ and $\frac{D^{\mathrm{v}}}{N}$ as the dimensions of $\hat{D}^{\mathrm{qk}}$ and $\hat{D}^{\mathrm{v}}$.

The multi-head attention concatenates all the attention heads, followed by another linear projection layer $\boldsymbol{W}^{\mathrm{o}} \in \mathbb{R}^{\hat{D}^{\mathrm{v}} \times D^{\mathrm{v}}}$:

$$\texttt{MHA}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}, \boldsymbol{M}) = \texttt{Concat}\left(\boldsymbol{H}_1, \ldots, \boldsymbol{H}_N\right)\boldsymbol{W}^{\mathrm{o}} \tag{5}$$

2 More Loss Function Details

The main paper has detailed the rendering loss and saliency loss. In this section, we provide more details about the four regularization losses: normal consistency loss, Laplacian smoothing loss, cross-instance consistency loss, and uniformity regularization loss.

2.1 Normal Consistency Loss

The formulation of the normal consistency loss follows the previous work [3]. Let $\mathcal{E}$ denote the ensemble of all edges and θ_i denote the angle between a pair of adjacent faces that conjoin along an edge $e_i \in \mathcal{E}$. The normal consistency loss is defined as:

$$\ell^{\mathrm{NORM}} = \sum_{e_i \in \mathcal{E}} \left(\cos\left(\theta_i\right) + 1\right)^2 \tag{6}$$

which encourages the adjacent faces to have similar normal directions.

2.2 Laplacian Smoothing Loss

The Laplacian smoothing loss [3] aims at smoothing the movement between each vertex and its neighborhood. The loss is defined as:

$$\ell^{\mathrm{LAP}} = \|\boldsymbol{\delta}_v - \frac{1}{|\mathcal{N}(v)|} \sum_{v' \in \mathcal{N}(v)} \boldsymbol{\delta}_{v'}\|_2^2 \tag{7}$$

where v denotes a vertex and $\mathcal{N}(v)$ is the set of its neighboring vertices, and $\boldsymbol{\delta}_v$ represents the predicted offset for the vertex v.

2.3 Cross-Instance Consistency Loss

The cross-instance consistency loss [6] helps mitigate ambiguity during learning by modeling the shape/texture similarity between object instances in the same category. It maintains a memory bank that stores the latent codes of the shape and texture generated during each training iteration. From the memory bank, we can identify the instance $\boldsymbol{I}^{\mathrm{sh}}$ whose shape is most similar to that of the current input image and call it the shape neighbor. Then, a new image $\tilde{\boldsymbol{I}}^{\mathrm{sh}}$ is rendered by the shape predicted from the input image and other attributes predicted from the shape neighbor. Similarly, we can identify the texture neighbor $\boldsymbol{I}^{\mathrm{tx}}$ in the memory bank and render $\tilde{\boldsymbol{I}}^{\mathrm{tx}}$ by swapping the texture. Finally, the cross-instance consistency loss penalizes the discrepancy between the rendered images and the neighbor images:

$$\ell^{\mathrm{CROSS}} = \ell^{\mathrm{REN}}\left(\boldsymbol{I}^{\mathrm{sh}}, \tilde{\boldsymbol{I}}^{\mathrm{sh}}\right) + \ell^{\mathrm{REN}}\left(\boldsymbol{I}^{\mathrm{tx}}, \tilde{\boldsymbol{I}}^{\mathrm{tx}}\right) \tag{8}$$

where ℓ^{Render} is the rendering loss as described in the main paper. Note that we only enforce cross-instance consistency loss in the coarse stage so that the model can generate more unique shapes and textures in the refinement stage.

2.4 Uniformity Regularization Loss

Following [4, 6], we employ a uniformity regularization loss ℓ^{UNI} to impose a uniform distribution constraint on the camera multiplex prediction:

$$\ell^{\mathrm{UNI}} = \sum_k |p_k - 1/K| \tag{9}$$

where p_k denotes the probability of the kth pose hypothesis in the predicted camera multiplex and K is the total number of pose hypotheses.

3 Model Size

Our model has 16.05M parameters, representing a 6% increase compared to Unicorn [6] (15.17M).

Fig. 1: Keypoint transfer visualization. (a) Annotated keypoints on the source image. (b) Annotated keypoints on the target image. (c) Transferred keypoints on the target image according to the source and target meshes generated by our model. (d) Transferred keypoints on the rendered target image by our model.

Fig. 2: Ablation visual results on ShapeNet.

4 Keypoint Transfer Results

Qualitative results of keypoint transfer are shown in Fig. 1. Following [5], the keypoints transfer accuracy is evaluated using the PCK metric on CUB-200-2011 [8]. Our model generates meshes on a source image and a target image. Both images are annotated with a set of common keypoints. To calculate the keypoints transfer accuracy, we first map the source keypoints to the corresponding locations on the source mesh, then transfer the source keypoints to the target image according to the target mesh, and finally calculate the percentage of correct keypoints (PCK) by comparing the annotated target keypoints and the transferred target keypoints that fall within a threshold 0.1. The keypoints can be transferred from the source mesh to the target mesh because there are one-to-one correspondences between vertices/faces of the source mesh and those of the target mesh.

Fig. 1 illustrates (a) annotated keypoints on the source image, (b) annotated keypoints on the target image, (c) transferred keypoints on the target image according to the meshes generated by our model, and (d) transferred keypoints on the rendered target image by our model. We can observe that the transferred keypoints are well aligned with the annotated keypoints, indicating that our model has effectively captured the object shape.

5 Ablation Qualitative Results on ShapeNet

In this section, we provide the visualization of the ablation results on ShapeNet [2], as shown in Fig. 2. These visualizations include results from the full model, the model with AT removed, and the model with ST removed. We can observe that the removal of AT and ST significantly affects the quality of the generated 3D shape and texture.

References

1. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
2. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
3. Chen, W., Ling, H., Gao, J., Smith, E., Lehtinen, J., Jacobson, A., Fidler, S.: Learning to predict 3d objects with an interpolation-based differentiable renderer. Advances in neural information processing systems **32** (2019)
4. Henderson, P., Ferrari, V.: Learning single-image 3d reconstruction by generative modelling of shape, pose and shading. International Journal of Computer Vision **128**(4), 835–854 (2020)
5. Li, X., Liu, S., Kim, K., De Mello, S., Jampani, V., Yang, M.H., Kautz, J.: Self-supervised single-view 3d reconstruction via semantic consistency. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16. pp. 677–693. Springer (2020)
6. Monnier, T., Fisher, M., Efros, A.A., Aubry, M.: Share with thy neighbors: Single-view reconstruction by cross-instance consistency. In: European Conference on Computer Vision. pp. 285–303. Springer (2022)
7. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
8. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-ucsd birds 200 (2010)