Supplementary for "TriVol: Point Cloud Rendering via Triple Volumes"

Tao Hu [1*] Xiaogang Xu[1*] Ruihang Chu [1] Jiaya Jia[1,2]
[1] The Chinese University of Hong Kong [2] SmartMore
{taohu,xgxu,rhchu,leojia}@cse.cuhk.edu.hk

1. Experimental Details

1.1. Network Details

The learnable modules in our framework contain the *Feature TriVol decoder* $D = \{D_x, D_y, D_z\}$ and the MLP g in NeRF [6]. The structures of D_x, D_y, and D_z are all the same except for the tensor shape. Fig. 1 describe the architecture and the tensor shape for D_x, and the visualizations for D_y and D_z are similar. The MLP architecture is displayed in Fig. 2.

1.2. Training Details

Our proposed architecture is end-to-end trainable and requires no pre-training of any sub-modules. The models are trained with the AdamW optimizer with β_1 = 0.9 and β_2 = 0.999. The training epoch number for ScanNet, ShapeNet, and GSO is 100. The batch size is 4, and we adopt the distributed training in Pytorch using four RTX 3090 GPUs, i.e., the batch size is 1 on each GPU. The scene and its viewpoint in every batch are randomly selected.

1.3. Dataset Details

ScanNet. For the ScanNet dataset, the mesh in each scene is provided. The multi-view images as well as the corresponding camera parameters are calibrated.
ShapeNet and GSO. For the ShapeNet and GSO datasets, we utilize Blender [5] to render multi-view images from the textured mesh. We use xatlas [11] to get texture coordinates for the mesh in ShapeNet, from where we can warp our 3D mesh into a 2D plane and obtain the corresponding 3D location on the mesh surface for any position on the 2D plane. We then discretize the 2D plane into an image, and for each pixel, we query the texture field using the corresponding 3D location to obtain the RGB color to get the texture map. In the GSO dataset, the texture map is provided.

To generate the multi-view data, we first scale each shape such that the longest edge of its bounding box equals e_m. $e_m = 0.9$ for the *Car* in ShapeNet and *Shoe* in GSO. we then render the RGB images and silhouettes from camera poses sampled from the upper hemisphere of each object. For ShapeNet, we render each mesh with the elevation angles as $\{0°, 30°, 60°\}$, and the rotation angles associated with each elevation angle are $\{0°, 5°, ..., 350°, 355°\}$ with an interval of $5°$, totally 216 camera poses for each mesh. For GSO, we render each mesh with the elevation angles as $\{-60°, -30°, 0°, 30°, 60°\}$, and the rotation angles associated with each elevation angle are $\{0°, 5°, ..., 350°, 355°\}$, totally 360 camera poses for each mesh. For all camera poses in ShapeNet and GSO, we use a fixed radius of 1.2 and the FOV angle of $49.13°$. We render the images in Blender using fixed lighting.

*Equal Contribution.

2. View Consistency

The view consistency issue mentioned in this paper indicates that the appearance of the same object might be distinct under different views. Such an issue usually happens in the method that projects existing points' features into the 2D plane and then trains 2D neural networks to synthesize the 2D image. NPBG++ [7], NPCR [3], and ADOP [8] are representative approaches, and the view inconsistency phenomenon in its results can be distinctly seen in Fig. On the other hand, the view inconsistency does not exist in the outcomes of approaches incorporating 3D feature volume and NeRF-based rendering, *e.g.*, Voxels-128 and our TriVol, as shown in Fig. 3 Furthermore, compared with TriVol, our framework with TriVol can render realistic results efficiently.

3. Experiments on More Datasets

This section provides more visual comparisons between our method and baselines [7, 8, 10] on different datasets, as shown in Fig. 4, Fig. 5, and Fig. 6.

4. Failure Case

As indicated in the limitation section of the main paper, our method still struggles to render clear image regions when there is a large number of missing points, as illustrated in Fig. 7. This is also one of the most challenging issues for current point renderers, and we aim to solve it by

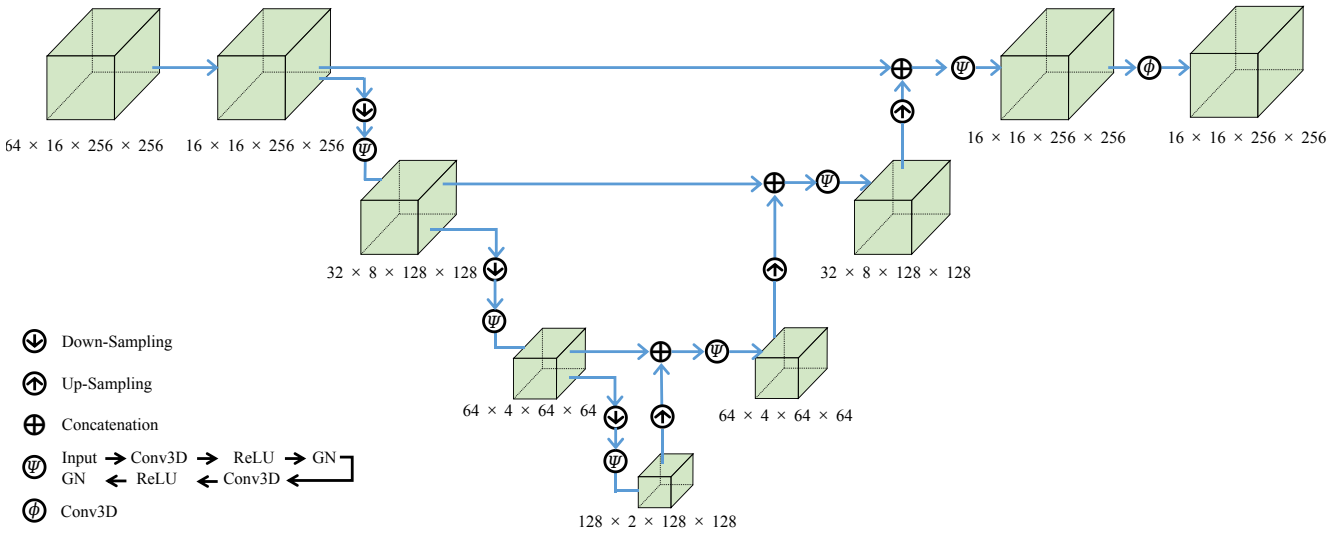


Figure 1. The network structure of TriVol decoder D_x, "Conv3D" means 3D convolutions, "GN" denotes group normalization [9] (the group number is 8). The down-sampling is completed by 3D max pooling, and the up-sampling is implemented by 3D interpolation.

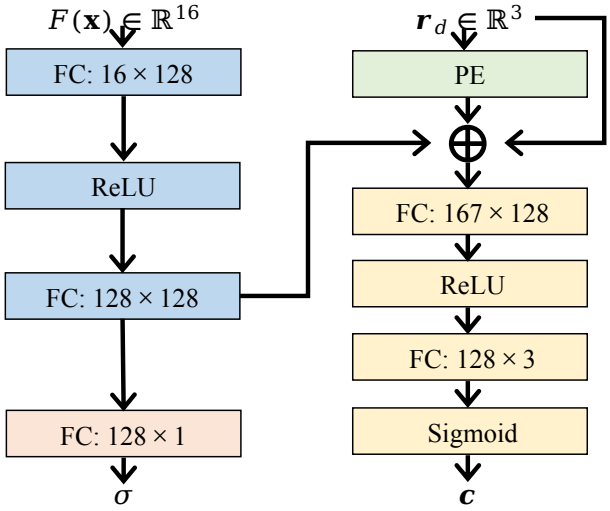


Figure 2. The network architecture for the MLP in NeRF. "FC" represents the fully-connected layer, "PE" denotes the position encoding.

combining our framework with a generative model in future work.

References

[1] 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. *arxiv*, 2015. 3, 4, 5

[2] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017. 5

[3] Peng Dai, Yinda Zhang, Zhuwen Li, Shuaicheng Liu, and Bing Zeng. Neural point cloud rendering via multi-plane projection. In *CVPR*, 2020. 1

[4] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *ICRA*, 2022. 4

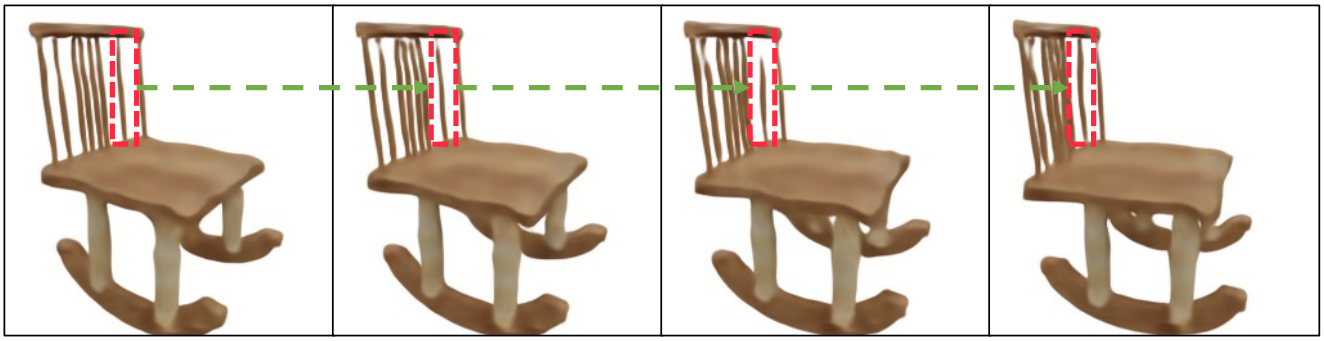
[5] Roland Hess. *Blender Foundations: The Essential Guide to Learning Blender 2.6*. Focal Press, 2010. 1

[6] 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 *ECCV*, 2020. 1

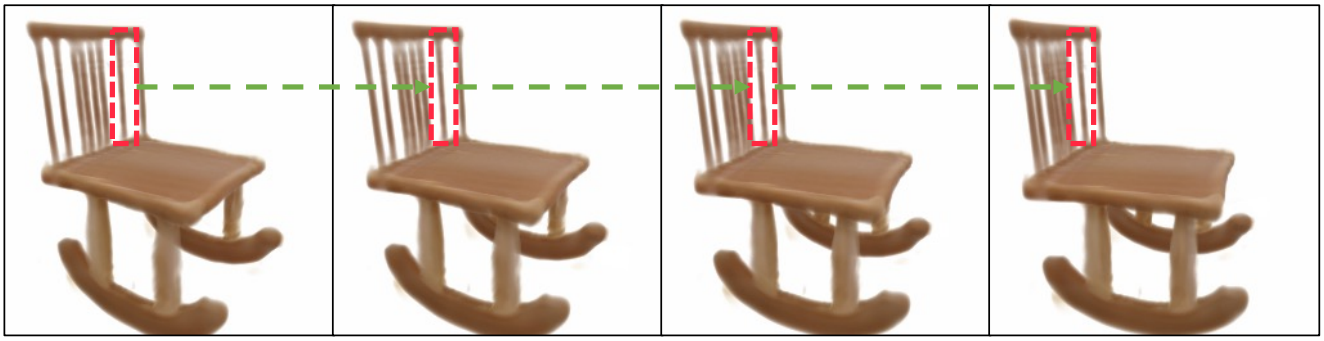
[7] Ruslan Rakhimov, Andrei-Timotei Ardelean, Victor Lempitsky, and Evgeny Burnaev. NPBG++: Accelerating neural point-based graphics. In *CVPR*, 2022. 1

[8] Darius Rückert, Linus Franke, and Marc Stamminger. ADOP: approximate differentiable one-pixel point rendering. *ACM TOG*, 2022. 1

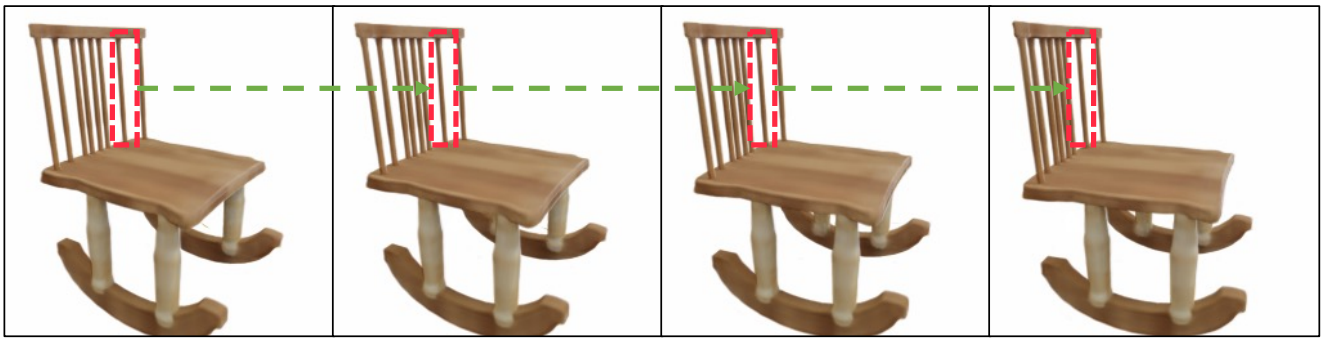
[9] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018. 2



(a) NBPG++: Not Consistent View, Photo-Realistic



(b) Voxel-128: Consistent View, Not Photo-Realistic



(c) Ours: Consistent View, Photo-Realistic

Figure 3. Comparison between ours with baselines on the ShapeNet-Chair dataset [1]. Our method can generate photo-realistic and view-consistent results. Please refer to the demo video for a better visualization.

[10] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-NeRF: Point-based neural radiance fields. In *CVPR*, 2022. 1

[11] Jonathan Young. xatlas. https://github.com/jpcy/xatlas, 2021. 1

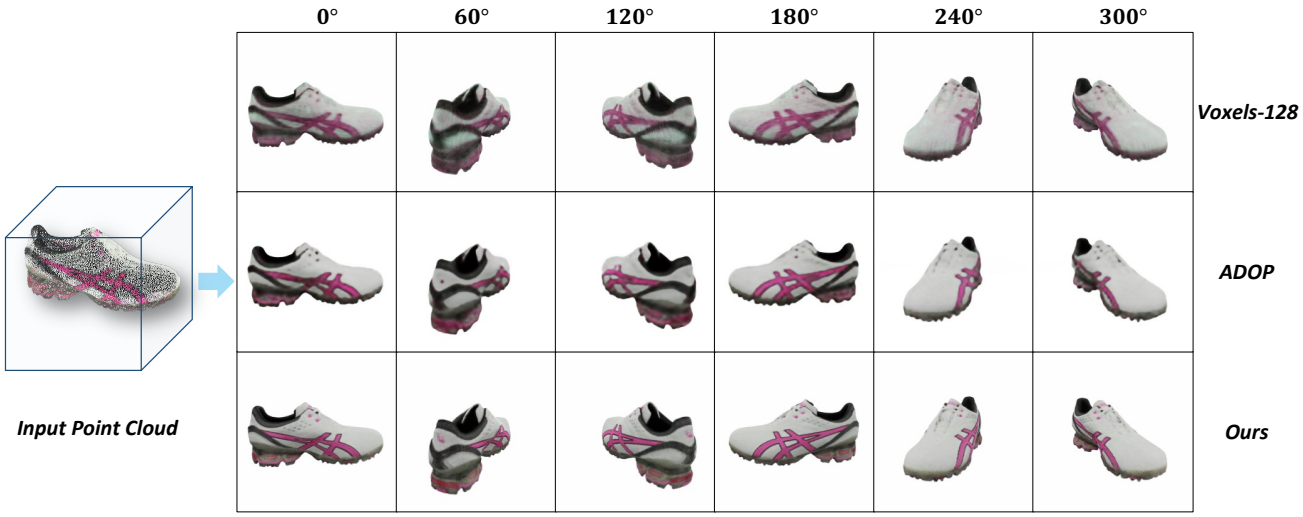


Figure 4. Comparison between ours with baselines on the GSO-Shoe dataset [4]. Please refer to the demo video to visualize the view consistency issue.

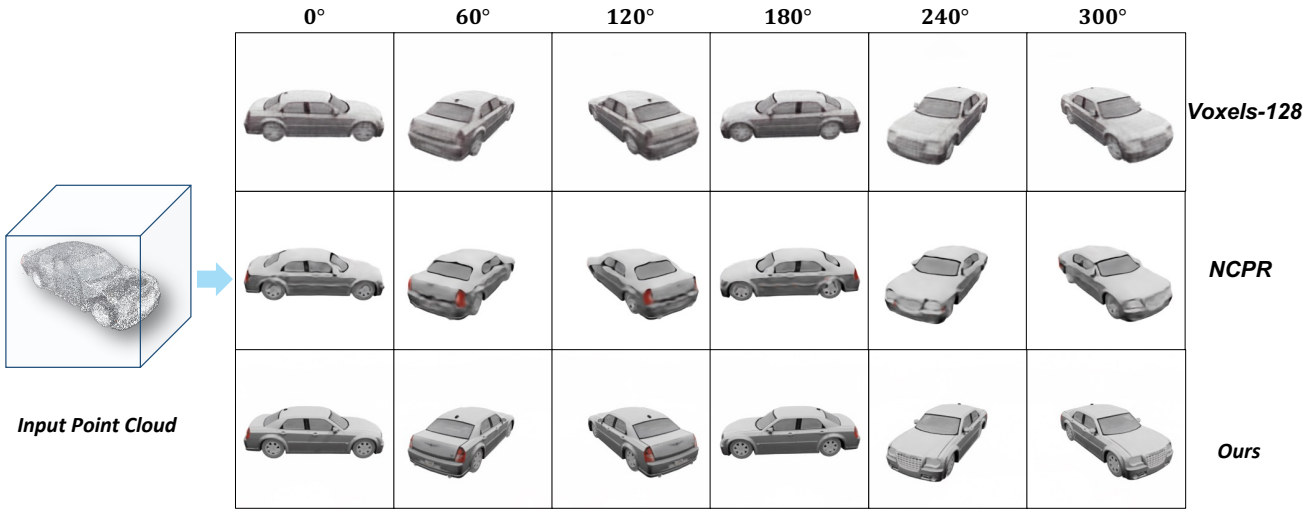


Figure 5. Comparison between ours with baselines on the ShapeNet-Car dataset [1]. Please refer to the demo video to visualize the view consistency issue.

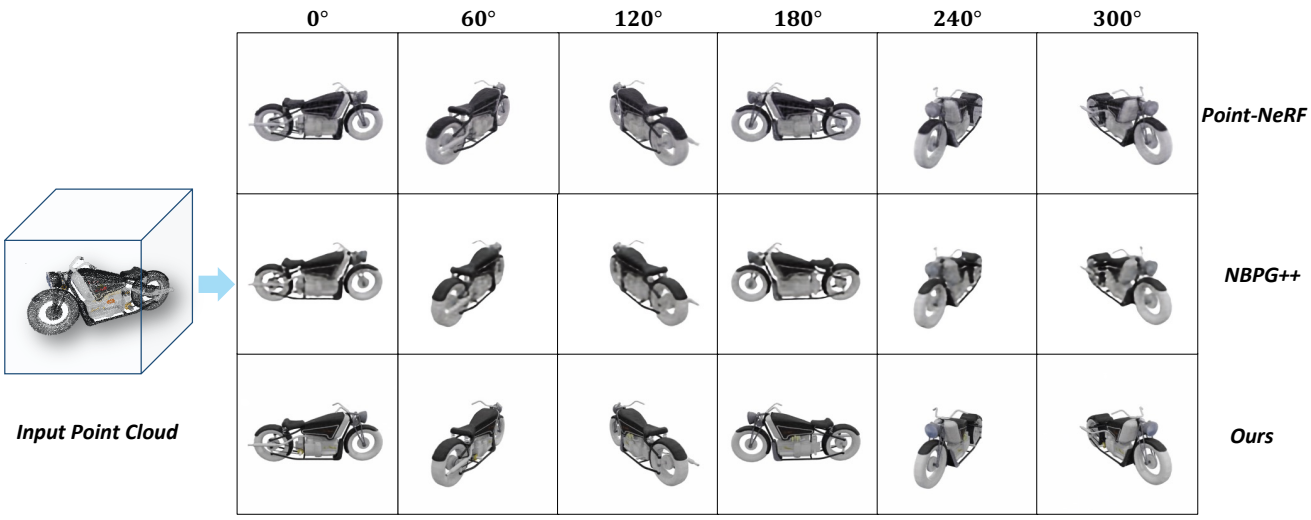


Figure 6. Comparison between ours with baselines on the ShapeNet-Motobike dataset [1]. Please refer to the demo video to visualize the view consistency issue.

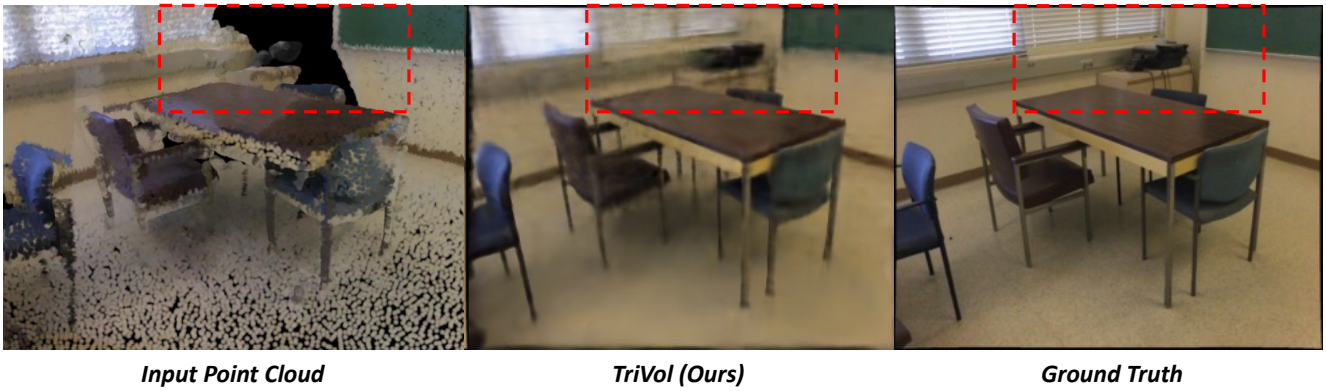


Figure 7. The failure case on the ScanNet dataset [2]. Missing a large number of points usually leads to blurred areas in the rendered images.