Supplementary Material for: Fast and Explicit Neural View Synthesis

Pengsheng Guo Miguel Angel Bautista Alex Colburn Liang Yang
Daniel Ulbricht Joshua M. Susskind Qi Shan
Apple
{pengsheng_guo,mbautistamartin,alexcolburn,ericyang,dulbricht,jsusskind,qshan}@apple.com

A. Limitations and Future Work

Though our model shows very promising reconstruction results and great efficiency, a few limitations still exist to be tackled as future work.

First, training a high fidelity as well as scene/category-agnostic representation remains an open problem for both explicit and implicit methods. The resolution of images that can be rendered by our model is capped by the resolution of the explicit voxel-like scene representation. Different from our approach, continuous representations like radiance fields are resolution-free by definition; however, they are either slow to obtain or slow to render. As a future work, we are planning to tackle this problem by increasing scene representation capacity for complex regions of space while minimizing the computational resources for empty regions. This can be achieved via a more flexible and sparse explicit space representations including octrees [18, 19, 23], mixture of volumetric primitives [11], and scene graph [14, 12]. In addition, it is also possible to enhance our model with super-resolution methods [20, 15].

Second, our current model cannot produce view-dependent lighting effects. We leave this as future work to be tackled with a physically based renderer that utilizes a more informative scene representation that incorporates view direction, light sources, material properties, and surface information [1, 25, 17]. Specifically, we can utilize the Lambertian RGBA model as basis to form an albedo map, and accumulate additional view dependent lighting onto separate surface representations including material properties such roughness, metalness, and surface properties such as normals and displacement maps. Additionally, we can also leverage techniques such as spherical harmonics [5] or a learned MLP to synthesize the color with view-dependent specular effects. By doing so during rendering time, we can leverage more advanced rendering techniques such as deferred rendering to better estimate the radiance field that captures both incoming light and material properties.

B. Social Impact

Our work represents a step towards producing highly realistic generative models of the world. Such a goal can have both positive and negative social impacts. On the positive side, our model can enable interactive art creation, improved AR/VR experiences, etc. On the negative side, better generative models of 3D objects can potentially be misused to produce technology similar to deep fakes that have become a concern for misrepresenting person identity. Releasing code, models, and technical papers can help the community prepare for this kind of misuse by detecting fake content or ensuring content is certified.

C. Experiment Details

C.1. Implementation Details

The 2D feature encoder utilizes the U-Net [16] with a ResNet-18 [6] backbone, initialized with ImageNet pre-trained weights. The intermediate feature channels are $64, 128, 256, 512$ at each spatial resolution level during downsampling. At the upsampling stage, we perform deconvolution and fuse the resulting features with skip encoder features at the same level through concatenation followed by two consecutive convolution blocks. The final 2D feature map has the same spatial resolution as the input image, with a 32 channel feature map.

At the inverse projection stage, the 2D feature map is back-projected into a 3D voxel space that has halved spatial height and width with respect to the input feature map. To enable back-projection, we first tile the 2D features into a 3D cube to prepare for sampling the inverse projection. Next, we leverage the camera intrinsic parameters to compute the homography matrix to map the cube to the viewing frustum. The grid mapping function is obtained by multiplying the homography matrix with individual points from the fixed voxel mesh grid set. Lastly, we apply the *grid_sample* function to perform the inverse projection.

The 3D feature decoder consists of a 3D U-Net with ResNet-3D blocks. Following common practice, when spatial resolution is downscaled by 2 times, we double the

Table 1: **Quantitative results on unseen-category view synthesis.** Our model bypasses ENR [4] baseline by a noticeable margin. Our two-view model has a very large improvement from single-view counterpart, which indicates the good generalizability of the model.

Metrics	Methods	bench	cbnt.	disp.	lamp	spkr.	rifle	sofa	table*	phone	boat	mean
PSNR↑	ENR	22.59	21.41	16.99	22.29	20.13	23.22	23.16	20.00	20.15	25.81	21.50
	Ours (Single-View)	23.10	22.27	17.01	22.15	20.76	23.22	24.20	20.54	19.59	25.77	21.90
	Ours (Two-View)	25.11	24.54	20.85	25.19	22.72	27.74	26.24	23.34	23.80	29.17	24.80
SSIM↑	ENR	0.845	0.799	0.674	0.819	0.768	0.900	0.848	0.759	0.806	0.896	0.807
	Ours (Single-View)	0.865	0.819	0.686	0.822	0.785	0.902	0.872	0.792	0.796	0.898	0.825
	Ours (Two-View)	0.909	0.875	0.818	0.905	0.838	0.963	0.913	0.871	0.897	0.950	0.893
LPIPS↓	ENR	0.182	0.199	0.273	0.203	0.202	0.143	0.166	0.206	0.182	0.154	0.190
	Ours (Single-View)	0.135	0.156	0.237	0.175	0.173	0.117	0.123	0.152	0.176	0.128	0.150
	Ours (Two-View)	0.108	0.122	0.153	0.118	0.140	0.072	0.098	0.106	0.107	0.087	0.107

channel numbers. The output 3D feature map has 16 channels with the same spatial resolution as the input image. At the final layer, we apply one 3D convolution to transform the tensor map into 4 channels and sigmoid activations to produce the RGBα volume.

At the amortized rendering stage, we combine the rotation transformation and the perspective deformation into one by left matrix multiplication, where rotation transformation transfers a volume from the source view to the target view and perspective deformation maps the viewing frustum to the output cube. After similar mesh grid multiplication and grid sampling steps as in inverse projection, we obtain the voxel volume used for alpha blending. The alpha blending process leverages *cumprod* method to calculate the accumulated transmittance between the near plane and the current point.

We use the ReLU nonlinearity and GroupNorm [22] normalization throughout the model for each nonlinear transformation. We train the network for 150 epochs with Adam optimizer and a fixed learning rate of 0.0016, the loss weighting for SSIM [21] loss is set to be 0.05. We only perform data normalization with ImageNet statistics at the preprocessing stage.

C.2. Computation Resources

For the category-specific view synthesis, each model is trained using 30 V100 GPUs on an internal cluster. It takes 5 days to train the single-chair dataset, 2.5 days to train the single-car dataset, and 1.5 days to train the multiple-chair dataset.

For the category-agnostic and unseen-category view synthesis, each model is trained using 30 V100 GPUs on an internal cluster, and it takes us 2.5 days to train each dataset.

We are able to perform the evaluation on a single GPU in minutes compared to PixelNeRF[24], which takes days for evaluation.

D. Additional Experiments & Visualizations

D.1. Additional Experiments for Unseen-Category View Synthesis

To further demonstrate the generalization capacity of our method, we extend the baselines for unseen-category view synthesis experiments to include ENR[4]. Our proposed approach beats ENR [4] in all metrics as shown in Table 1. Figure 1 also shows that our model is able to synthesize objects with more clean geometry (less blurry artifacts) as seen from the visualizations of boat (row 1), bench (row 7) objects. In addition, we also report results for a two-view model. During training, we randomly sample pairs of source images as input; during evaluation, we sample two images offset by 90 degrees azimuth as input. Figure 1 shows this model is not biased towards the training categories anymore, instead, the model is learning to perform view synthesis given two views of an unseen object.

D.2. Additional Experiments for 3D Reconstruction

We further demonstrate the 3D reconstruction performance of our model by comparing with two **supervised** 3D reconstruction baselines V-LSMs[8] and 3D-R2N2 [3]. These baselines use voxel occupancy as supervision, while our model only relies on 2D self-supervision. To conduct a fair comparison, we downsampled the predicted voxel from 64^3 to 32^3 and utilize the same thresholding strategy as reported in Section 4.2. Table 2 indicates that our model obtains better mIoU on 9 out of 13 categories of objects and better mean mIoU across categories.

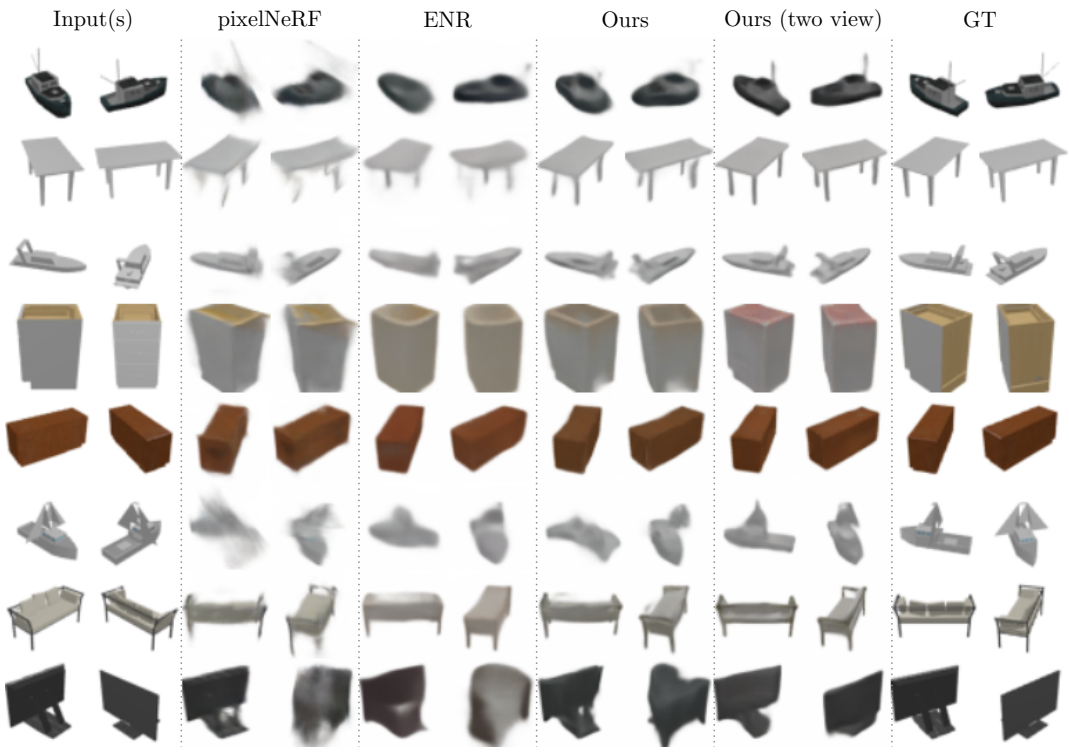

Figure 1: **Qualitative results on unseen-category datasets.** All methods except ours (two view) tasks a single input view (left image under Input(s) column) and perform novel view synthesis.

Table 2: **Quantitative results for 3D geometry reconstruction on 13 ShapeNet classes.** Our model has better mIoU metric compared to V-LSMs[8] and 3D-R2N2 [3] (1-view) that leverage groudtruth voxel occupancies as the supervision signal.

Metrics	Methods	plane	bench	cbnt.	car	chair	disp.	lamp	spkr.	rifle	sofa	table*	phone	boat	mean
mIoU↑	3D-R2N2 w/pose [3] (1 view)	56.7	43.2	61.8	77.6	50.9	44.0	40.0	56.7	56.5	58.9	51.6	65.6	53.1	55.1
	V-LSMs[8] (1 view)	**61.1**	50.8	65.9	79.3	**57.8**	53.9	48.1	63.9	**69.7**	67.0	**55.6**	67.7	58.3	61.5
	Ours (1 view)	57.7	**54.7**	**76.0**	**80.4**	57.0	**60.6**	**51.8**	**74.1**	60.2	**72.3**	53.8	**72.0**	**60.0**	**63.25**

D.3. Real World View Synthesis

We also test our model generalization capabilities when transferring to a new target domain. We use the model trained with ShapeNet synthetic car objects from Section 4.1.1 of the main text and perform novel view synthesis on real car images from [10]. Following the same protocol as pixelNeRF [24], we masked out the background and paint as white using PointRend[9], and perform view synthesis in input view coordinate space. Figure 2 suggests that our model can predict plausible novel views of the real cars. The geometry is similar to pixelNeRF[24] results with less artifacts and blurry effects.

D.4. Increasing Number of Source Views on Category-specific Multiple Chairs

We analyze the capability of our model in terms of handling different numbers of source views. In this experiment, we use a model trained with two source views for evaluation. As shown in Table 3 and Figure 3, our model's performance decreases on the one view setting due to having to deal with a large number of degrees of freedom. As the number of source view increases, the model is able to satisfy more constraints; our model achieves the best performance in the three-view setting.

D.5. Real World DTU dataset

We evaluate the performance of our model in the real world DTU MVS dataset [7]. As opposed to the ShapeNet[2] dataset, DTU contains a smaller number of

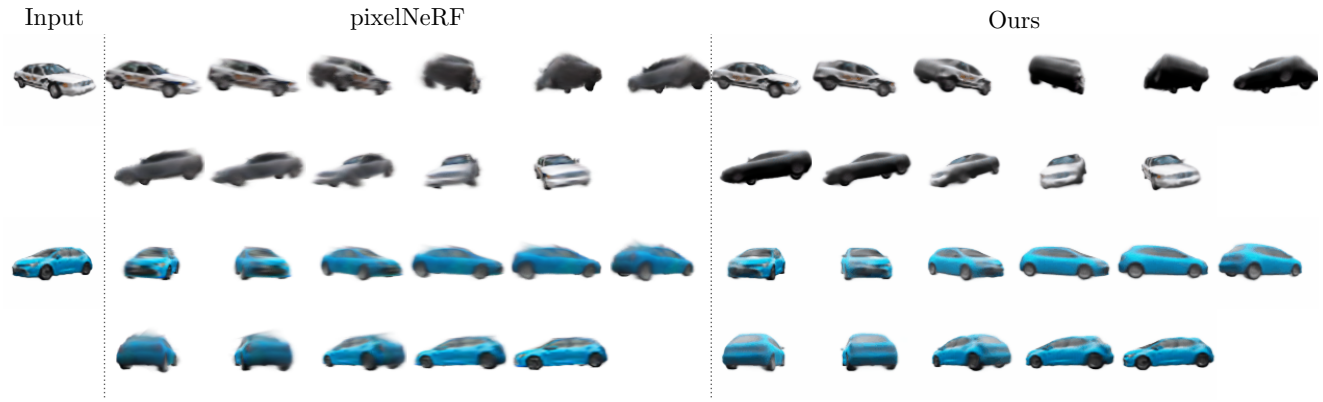

Figure 2: **Qualitative results for Real Car Image.** We use the model trained with ShapeNet virtual cars objects and perform novel view synthesis on real car images. Our model is able to synthesize plausible novel views, similar to pixelNeRF[24] but with less blurry effects.

Table 3: **Quantitative results on category-specific multiple chairs with increased number of views.**

Methods	1-view			2-view			3-view		
	PSNR↑	SSIM↑	LPIPS ↓	PSNR↑	SSIM↑	LPIPS ↓	PSNR↑	SSIM↑	LPIPS ↓
Ours	17.02	0.747	0.241	24.13	0.907	0.098	26.18	0.935	0.076

scenes with very refined object textures, various lighting conditions and higher resolution images. This increased level of complexity requires our model to learn a good prior over shapes and textures to generalize across scenes given limited training samples. For this experiment, we follow the protocol in [24] and use 88 training scenes and 15 test scenes. The images resolution is 300×400. During training, we randomly sample three images of the same object as input; during evaluation, we used a fixed set of input images. To better accommodate the scene in our voxel scene representation, we tune the physical size through the voxel distance between the object to camera used in inverse projection and perspective deformation steps.

As shown in Fig. 4, our model is able to synthesize accurate novel views given 3 inputs of unseen scenes while doing so at 1900x the speed of pixelNeRF [24]. With a larger voxel distance between the object to camera (190 v.s. 80), the physical size of the voxel scene representation is smaller, the object texture is more refined under the same voxel resolution. In that case, we are not able to estimate the background table correctly as the table is beyond the scope of the voxel representation. Literature like [13] solve this problem by masking out the background and focus on reconstructing the foreground object. As a result, the reconstruction accuracy of our method is slightly impact when compared pixelNeRF as shown in Table 4. Efficiency wise, as pixelNeRF's rendering time increases linearly with the resolution of the image, our model is able to increase per-view inference and rendering speed by over 600x, per-object inference and rendering speed by over 1900x. As discussed in the limitations section, an exciting future work to mitigate the small gap in performance could be to include a multi-resolution voxel representation [18].

D.6. Additional Visualizations

We include additional visualizations for every experiment:

- Figure 5: category-specific single chair dataset with one source view;
- Figure 6: category-specific single chair dataset with two source views;
- Figure 7: category-specific single car dataset with one source view;
- Figure 8: category-specific single car dataset with two source views;
- Figure 9: category-specific multiple chairs dataset;
- Figure 10: category-agnostic dataset;
- Figure 11: unseen-category dataset.

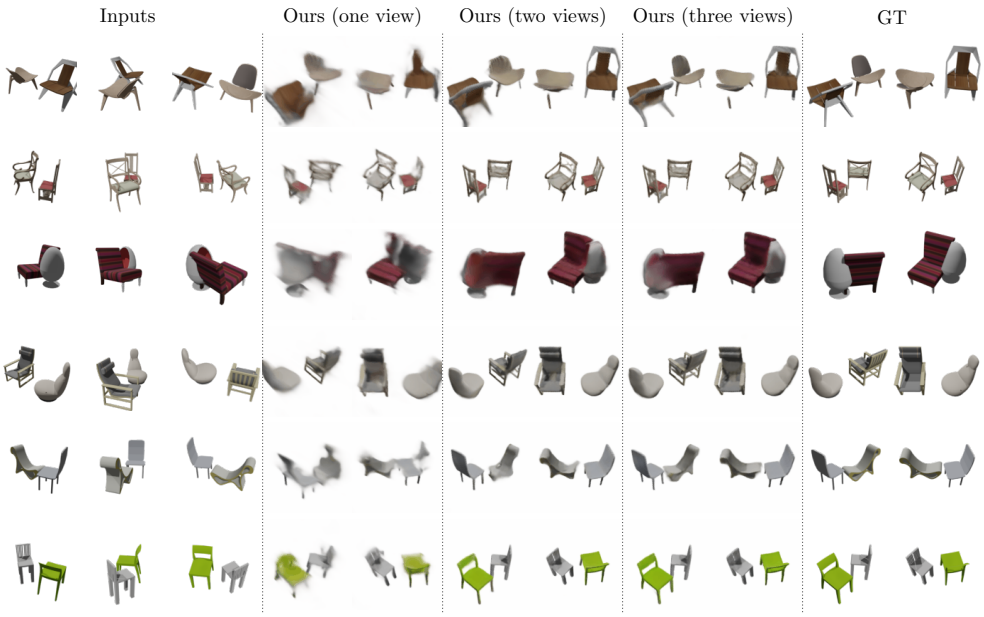

Figure 3: **Qualitative results on category-specific multiple chairs with an increased number of views.** We apply our model only trained with two input source viewers to a different number of input views settings during evaluation. Our model can predict better geometries and textures with the increasing number of views.

Table 4: **Quantitative results on unseen test set scenes for DTU dataset.** Our model sacrifices performance in the evaluation metrics mainly due to the limited physical size of the voxel scene representation. Meanwhile, it increases inference and rendering speed by 600x per-view wise and 1900x per object wise compared to pixelNeRF.

Methods	3-view			Inference and Rendering Time (s)	
	PSNR↑	SSIM↑	LPIPS ↓	Per-view	Per-object (46 views)
NeRF (partial set)	9.85	0.374	0.622	-	-
pixelNeRF (partial set)	19.24	0.687	0.399	-	-
pixelNeRF	18.99	0.680	0.420	35.9782	1655
Ours (190)	16.49	0.660	0.469		
Ours (160)	16.92	0.657	0.471	0.0557	0.8523
Ours (80)	17.58	0.645	0.494		

We intentionally use the same random indices for different view settings of the single chair/car dataset. From the visualizations in Figure 5 v.s. Figure 6 and Figure 7 v.s. Figure 8, it can be easily observed that our model can synthesize better geometry and texture at novel target pose with more information given.

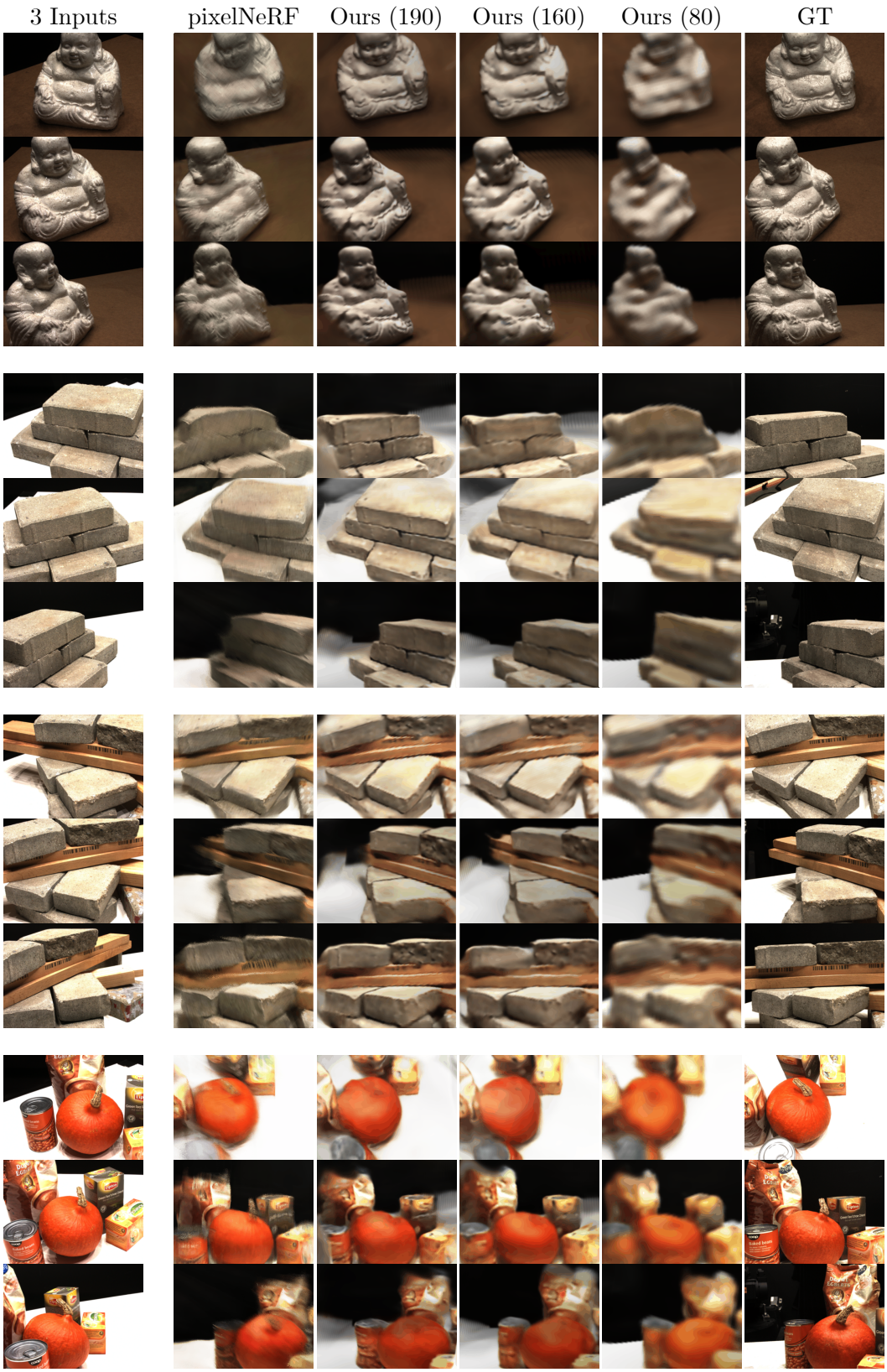

Figure 4: **Qualitative results on unseen test set scenes for DTU dataset.** Our model is able to synthesize reasonable novel views given 3 inputs of unseen scenes while doing so at 1900x the speed of pixelNeRF.

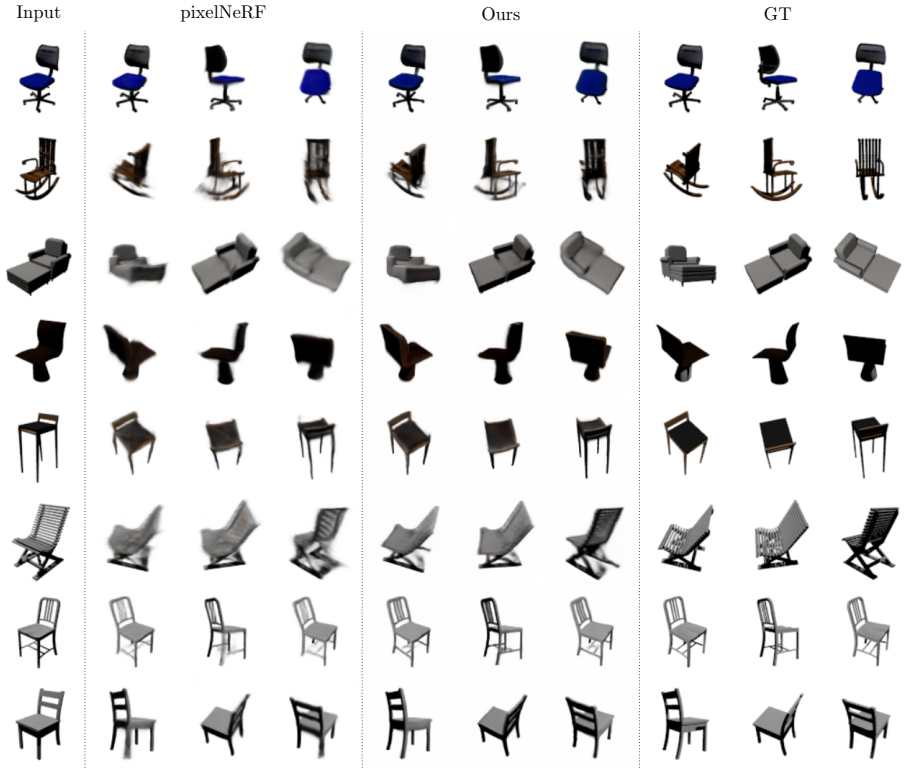

Figure 5: **Qualitative results on category-specific single chair (single-view)**

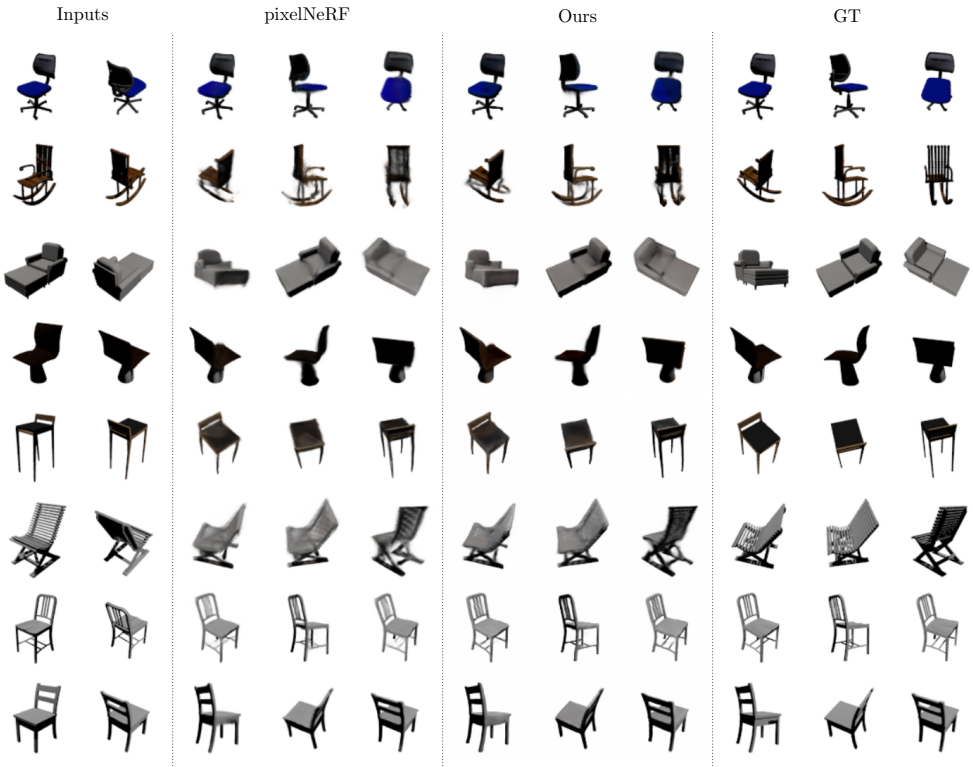

Figure 6: **Qualitative results on category-specific single chair (two-view)**

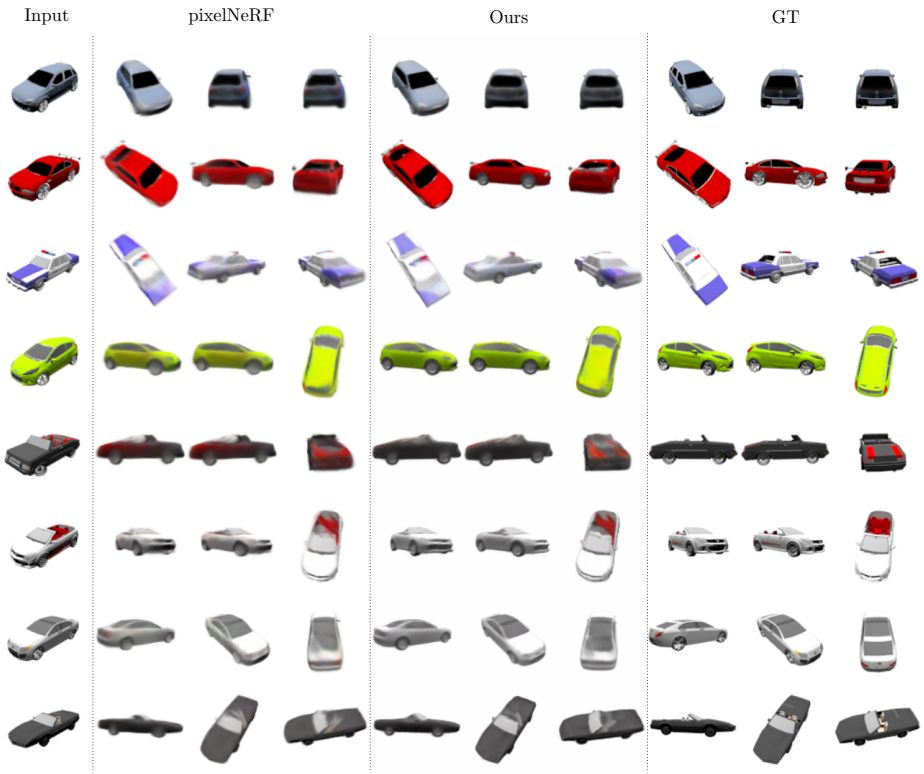

Figure 7: **Qualitative results on category-specific single car (single-view)**

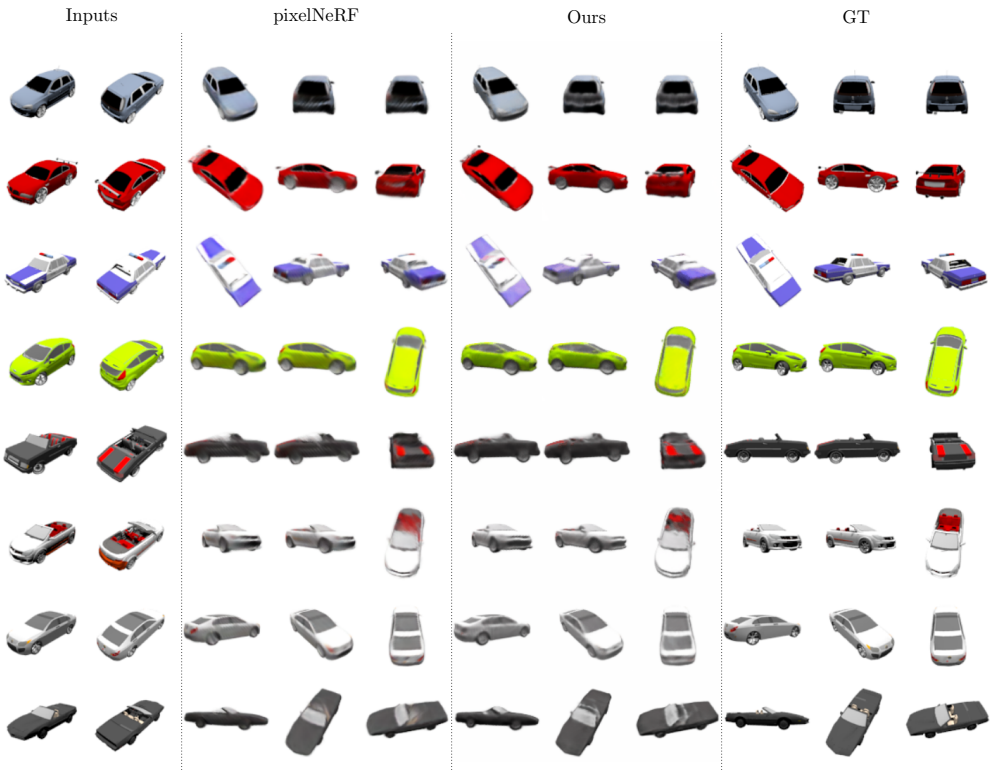

Figure 8: **Qualitative results on category-specific single car (two-view)**

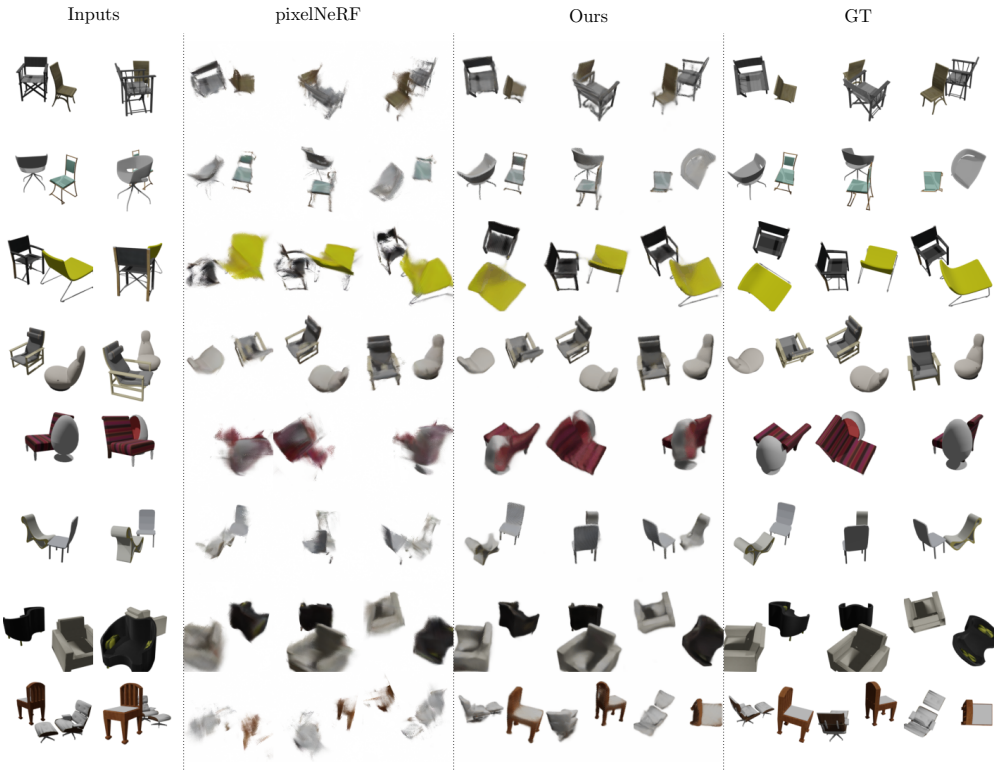

Figure 9: **Qualitative results on category-specific multiple chairs (two-view)**

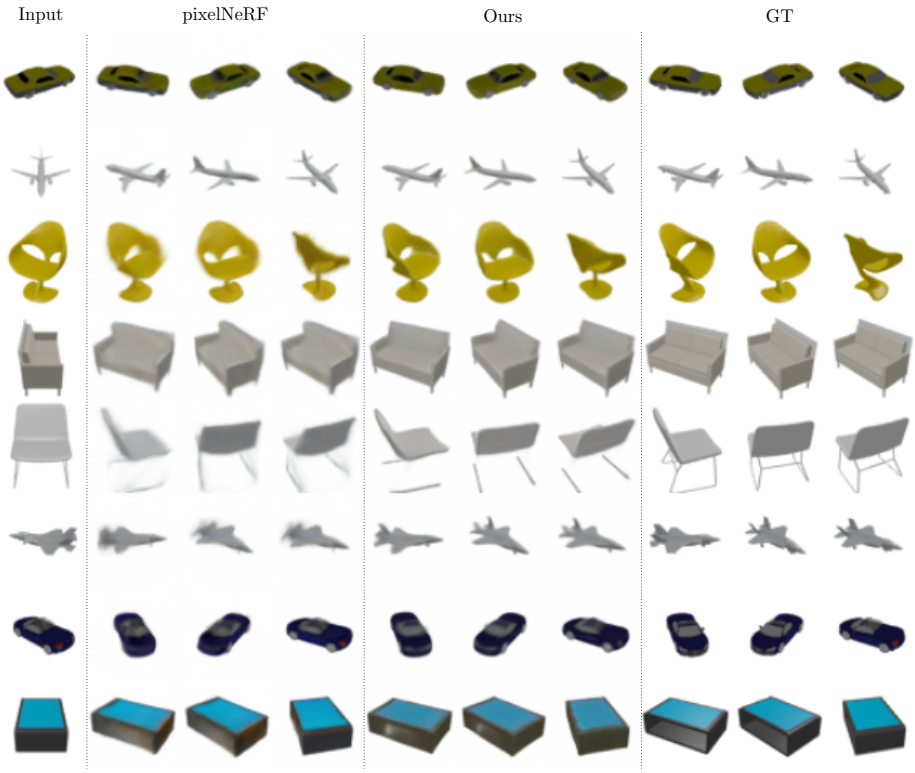

Figure 10: **Qualitative results on multiple-category dataset (one-view)**

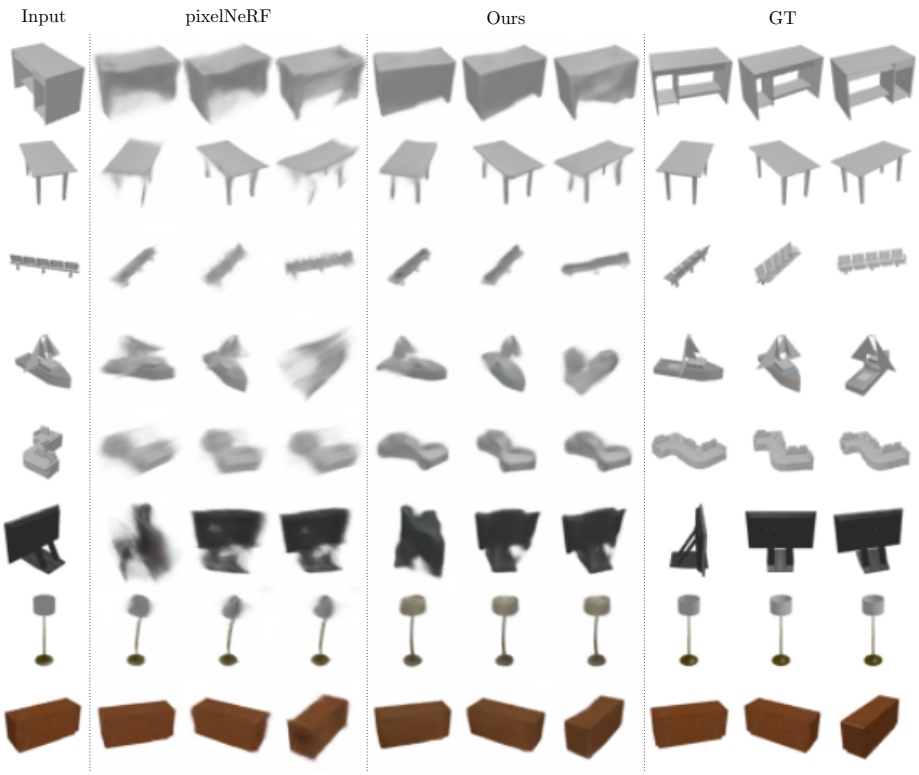

Figure 11: **Qualitative results on unseen-category dataset (one-view)**

References

[1] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. *arXiv preprint arXiv:2012.03918*, 2020.

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[3] 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.

[4] Emilien Dupont, Miguel Angel Bautista, Alex Colburn, Aditya Sankar, Carlos Guestrin, Josh Susskind, and Qi Shan. Equivariant neural rendering. *arXiv preprint arXiv:2006.07630*, 2020.

[5] Robin Green. Spherical harmonic lighting: The gritty details. In *Archives of the game developers conference*, volume 56, page 4, 2003.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[7] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014.

[8] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine, 2017.

[9] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering, 2020.

[10] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[11] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *arXiv preprint arXiv:2103.01954*, 2021.

[12] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. *arXiv preprint arXiv:2011.12100*, 2020.

[13] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020.

[14] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2856–2865, 2021.

[15] Nathanaël Carraz Rakotonirina and Andry Rasoanaivo. Esrgan+: Further improving enhanced super-resolution generative adversarial network. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3637–3641. IEEE, 2020.

[16] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[17] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021.

[18] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2088–2096, 2017.

[19] Peng-Shuai Wang, Yang Liu, and Xin Tong. Deep octree-based cnns with output-guided skip connections for 3d shape and scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 266–267, 2020.

[20] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018.

[21] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[22] Yuxin Wu and Kaiming He. Group normalization, 2018.

[23] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. *arXiv preprint arXiv:2103.14024*, 2021.

[24] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. *arXiv preprint arXiv:2012.02190*, 2020.

[25] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *arXiv preprint arXiv:2106.01970*, 2021.