PET-NeuS: Positional Encoding Tri-Planes for Neural Surfaces Supplementary Material

Yiqun Wang
Chongqing University and KAUST

Ivan Skorokhodov
KAUST

Peter Wonka
KAUST

In these supplementary materials, we first provide a derivation for incorporating positional encoding into tri-planes. We then provide ablation studies for geometric initialization, frequency bands, other architectures, and EMD evaluation. We also show the details of self-attention convolution and the multi-scale architecture we compared to. Furthermore, we conduct experiments and show some examples of other real-world scenes. We finally show more qualitative comparisons to supplement the main text.

A. Derivations for Positional Encoding Tri-Planes

In this section, we derive Eq. 13 in the main text to incorporate positional encodings into tri-planes. Recall the 3D function $f\left(x,y,z\right)$ can be expanded as follows.

$$\begin{aligned} f\left(x,y,z\right) &= \sum_{k=-K}^{K} c_k\left(x,y\right)\Theta_k^z && (18)\\ &= \sum_{k=-K}^{K}\sum_{n=-N}^{N} b_{nk}\left(x\right)\Theta_n^y\Theta_k^z && (19)\\ &= \sum_{k=-K}^{K}\sum_{n=-N}^{N}\sum_{m=-M}^{M} a_{mnk}\Theta_m^x\Theta_n^y\Theta_k^z && (20)\end{aligned}$$

where m, n, and k are the different frequencies for x, y, z with maximum number of scales $M, N, K \mapsto \infty$ and

$$\Theta_t^v = \begin{cases} \cos\left(tv\right) & t>0\\ 1 & t=0\\ \sin\left(tv\right) & t<0 \end{cases} \tag{21}$$

The idea is to use $\cos(tv)$ and $\sin(tv)$ to represent the function $f\left(x,y,z\right)$. We first expand Θ_m^x to represent $f\left(x,y,z\right)$ into the form of $\cos\left(mx\right)$ and $\sin\left(mx\right)$ as follows.

$$\begin{aligned} f = &\sum_{k=-K}^{K}\sum_{n=-N}^{N}\left(\sum_{m=1}^{M} a_{mnk}\cos\left(mx\right)\right)\Theta_n^y\Theta_k^z && (22)\\ &+\sum_{k=-K}^{K}\sum_{n=-N}^{N}\left(\sum_{m=1}^{M} a_{(-m)nk}\sin\left(mx\right)\right)\Theta_n^y\Theta_k^z && (23)\\ &+\sum_{k=-K}^{K}\sum_{n=-N}^{N}\left(a_{0nk}\right)\Theta_n^y\Theta_k^z && (24)\end{aligned}$$

There are three terms in the equation. The first and second terms can be rewritten as a combination of $\cos\left(mx\right)$ and $\sin\left(mx\right)$ as follows.

$$\begin{aligned} &\sum_{k=-K}^{K}\sum_{n=-N}^{N}\left(\sum_{m=1}^{M} a_{mnk}\cos\left(mx\right)\right)\Theta_n^y\Theta_k^z && (25)\\ &=\sum_{m=1}^{M}\left(\sum_{k=-K}^{K}\sum_{n=-N}^{N} a_{mnk}\Theta_n^y\Theta_k^z\right)\cos\left(mx\right) && (26)\\ &\sum_{k=-K}^{K}\sum_{n=-N}^{N}\left(\sum_{m=1}^{M} a_{(-m)nk}\sin\left(mx\right)\right)\Theta_n^y\Theta_k^z && (27)\\ &=\sum_{m=1}^{M}\left(\sum_{k=-K}^{K}\sum_{n=-N}^{N} a_{(-m)nk}\Theta_n^y\Theta_k^z\right)\sin\left(mx\right) && (28)\end{aligned}$$

We define $\hat{g}_m(y,z) = \sum\limits_{k=-K}^{K}\sum\limits_{n=-N}^{N} a_{mnk}\Theta_n^y\Theta_k^z$ and $\hat{g}'_m(y,z) = \sum\limits_{k=-K}^{K}\sum\limits_{n=-N}^{N} a_{(-m)nk}\Theta_n^y\Theta_k^z$. In this way, the function can be expanded in the following form.

$$\begin{aligned} f = &\left(\sum_{m=1}^{M}\hat{g}_m(y,z)\cos\left(mx\right)\right) && (29)\\ &+\left(\sum_{m=1}^{M}\hat{g}'_m(y,z)\sin\left(mx\right)\right) && (30)\\ &+\sum_{k=-K}^{K}\sum_{n=-N}^{N}\left(a_{0nk}\right)\Theta_n^y\Theta_k^z && (31)\end{aligned}$$

However, $\cos(mx)$ and $\sin(mx)$ do not appear in the third term. We therefore would like to find some other way to express the third term using trigonometric functions. We observe that we can alternately expand the other two terms Θ_n^y and Θ_k^z in $f(x,y,z)$ of Eq. 3 in the same way and add them together as follows.

$$3f = \sum_{m=1}^{M} \hat{g}_m(y,z)\cos(mx) \tag{32}$$
$$+ \sum_{m=1}^{M} \hat{g}'_m(y,z)\sin(mx) \tag{33}$$
$$+ \sum_{k=-K}^{K}\sum_{n=-N}^{N} (a_{0nk})\,\Theta_n^y\Theta_k^z \tag{34}$$
$$+ \sum_{n=1}^{N} \hat{h}_n(x,z)\cos(ny) \tag{35}$$
$$+ \sum_{n=1}^{N} \hat{h}'_n(x,z)\sin(ny) \tag{36}$$
$$+ \sum_{k=-K}^{K}\sum_{m=-M}^{M} (a_{m0k})\,\Theta_m^x\Theta_k^z \tag{37}$$
$$+ \sum_{k=1}^{K} \hat{w}_k(x,y)\cos(kz) \tag{38}$$
$$+ \sum_{k=1}^{K} \hat{w}'_k(x,y)\sin(kz) \tag{39}$$
$$+ \sum_{k=-N}^{N}\sum_{n=-M}^{M} (a_{mn0})\,\Theta_m^x\Theta_n^y \tag{40}$$

where

$$\hat{h}_n(x,z) = \sum_{k=-K}^{K}\sum_{m=-M}^{M} a_{mnk}\Theta_m^x\Theta_k^z \tag{41}$$
$$\hat{h}'_n(x,z) = \sum_{k=-K}^{K}\sum_{m=-M}^{M} a_{m(-n)k}\Theta_m^x\Theta_k^z \tag{42}$$
$$\hat{w}_k(x,y) = \sum_{n=-N}^{N}\sum_{m=-M}^{M} a_{mnk}\Theta_m^x\Theta_n^y \tag{43}$$
$$\hat{w}'_k(x,y) = \sum_{n=-N}^{N}\sum_{m=-M}^{M} a_{mn(-k)}\Theta_m^x\Theta_n^y \tag{44}$$

We further expend Eq. 17 as follows.

$$\sum_{k=-K}^{K}\sum_{n=-N}^{N} (a_{0nk})\,\Theta_n^y\Theta_k^z \tag{45}$$
$$= \sum_{k=-K}^{K}\left(\sum_{n=1}^{N} a_{0nk}\cos(ny)\right)\Theta_k^z \tag{46}$$
$$+ \sum_{k=-K}^{K}\left(\sum_{n=1}^{N} a_{0(-n)k}\sin(ny)\right)\Theta_k^z \tag{47}$$
$$+ \sum_{k=-K}^{K} (a_{00k})\,\Theta_k^z \tag{48}$$
$$= \sum_{n=1}^{N}\left(\sum_{k=-K}^{K} a_{0nk}\Theta_k^z\right)\cos(ny) \tag{49}$$
$$+ \sum_{n=1}^{N}\left(\sum_{k=-K}^{K} a_{0(-n)k}\Theta_k^z\right)\sin(ny) \tag{50}$$
$$+ \sum_{k=-K}^{K} (a_{00k})\,\Theta_k^z \tag{51}$$

Similarly, we can expand Eq. 20, and Eq. 23.

$$\sum_{k=-K}^{K}\sum_{m=-M}^{M} (a_{m0k})\,\Theta_m^x\Theta_k^z \tag{52}$$
$$= \sum_{k=1}^{K}\left(\sum_{m=-M}^{M} a_{m0k}\Theta_m^x\right)\cos(kz) \tag{53}$$
$$+ \sum_{k=1}^{K}\left(\sum_{m=-M}^{M} a_{m0(-k)}\Theta_m^x\right)\sin(kz) \tag{54}$$
$$+ \sum_{m=-M}^{M} (a_{m00})\,\Theta_m^x \tag{55}$$

and

$$\sum_{k=-N}^{N}\sum_{n=-M}^{M} (a_{mn0})\,\Theta_m^x\Theta_n^y \tag{56}$$
$$= \sum_{m=1}^{M}\left(\sum_{n=-N}^{N} a_{mn0}\Theta_n^y\right)\cos(mx) \tag{57}$$
$$+ \sum_{m=1}^{M}\left(\sum_{n=-N}^{N} a_{(-m)n0}\Theta_n^y\right)\sin(mx) \tag{58}$$
$$+ \sum_{n=-N}^{N} (a_{0n0})\,\Theta_n^y \tag{59}$$

Therefore the function can be rewritten as follows.

$$3f = \sum_{m=1}^{M}\left(\hat{g}_m(y,z) + \sum_{n=-N}^{N} a_{mn0}\Theta_n^y\right)\cos(mx) \quad (60)$$

$$+ \sum_{m=1}^{M}\left(\hat{g}'_m(y,z) + \sum_{n=-N}^{N} a_{(-m)n0}\Theta_n^y\right)\sin(mx) \quad (61)$$

$$+ \sum_{n=1}^{N}\left(\hat{h}_n(x,z) + \sum_{k=-K}^{K} a_{0nk}\Theta_k^z\right)\cos(ny) \quad (62)$$

$$+ \sum_{n=1}^{N}\left(\hat{h}'_n(x,z) + \sum_{k=-K}^{K} a_{0(-n)k}\Theta_k^z\right)\sin(ny) \quad (63)$$

$$+ \sum_{k=1}^{K}\left(\hat{w}_k(x,y) + \sum_{m=-M}^{M} a_{m0k}\Theta_m^x\right)\cos(kz) \quad (64)$$

$$+ \sum_{k=1}^{K}\left(\hat{w}'_k(x,y) + \sum_{m=-M}^{M} a_{m0(-k)}\Theta_m^x\right)\sin(kz) \quad (65)$$

$$+ \sum_{m=-M}^{M}\left(a_{m00}\right)\Theta_m^x \quad (66)$$

$$+ \sum_{n=-N}^{N}\left(a_{0n0}\right)\Theta_n^y \quad (67)$$

$$+ \sum_{k=-K}^{K}\left(a_{00k}\right)\Theta_k^z \quad (68)$$

Now we ignore the constant coefficient 3 and the function f can be rewritten by the combination of positional encoding, which can be learned by an MLP network as follows.

$$f(x,y,z) = MLP\left(\left\{\begin{array}{c} \cos(mx) \\ \sin(mx) \\ g_m(y,z)\cos(mx) \\ g'_m(y,z)\sin(mx) \\ \cos(ny) \\ \sin(ny) \\ h_n(x,z)\cos(ny) \\ h'_n(x,z)\sin(ny) \\ \cos(kz) \\ \sin(kz) \\ w_k(x,y)\cos(kz) \\ w'_k(x,y)\sin(kz) \end{array}\right\}_{mnk}^{\textbf{flatten}}\right) \quad (69)$$

where

$$g_m(y,z) = \hat{g}_m(y,z) + \sum_{n=-N}^{N} a_{mn0}\Theta_n^y \quad (70)$$

$$g'_m(y,z) = \hat{g}'_m(y,z) + \sum_{n=-N}^{N} a_{(-m)n0}\Theta_n^y \quad (71)$$

$$h_n(x,z) = \hat{h}_n(x,z) + \sum_{k=-K}^{K} a_{0nk}\Theta_k^z \quad (72)$$

$$h'_n(x,z) = \hat{h}'_n(x,z) + \sum_{k=-K}^{K} a_{0(-n)k}\Theta_k^z \quad (73)$$

$$w_k(x,y) = \hat{w}_k(x,y) + \sum_{m=-M}^{M} a_{m0k}\Theta_m^x \quad (74)$$

$$w'_k(x,y) = \hat{w}'_k(x,y) + \sum_{m=-M}^{M} a_{m0(-k)}\Theta_m^x \quad (75)$$

B. Additional Ablations

B.1. Geometric Initialization

In this section, we provide an ablation study for geometric initialization. We compare our geometric initialization method (Our Geo Init) with two different settings. The first one is to initialize tri-planes from random noise and the 3-layer MLP with standard geometric initialization [1] (MLP Geo Init). The random noise is from a normal distribution $\mathcal{N}(0,1)$. In the second setting, the standard geometric initialization is not used during the initialization stage (No Geo Init). We show a representative example in Fig. 6 to compare the difference. We can observe that the method without geometry initialization causes the foreground and background to stick together and inconsistent reconstruction results on the belly. Using random noise to initialize tri-planes and standard geometric initialization to initialize MLP results in high-frequency details that are artifacts on the generated surface model. We also show a comparison of chamfer distance in Tab. 4. We run three times for each setting and take the average as a metric to evaluate the different initialization methods. Our geometric initialization leads to more consistent surface reconstruction results.

Table 4. Ablations for geometric initialization.

Method	1	2	3	**Mean**
No Geo Init	1.86	1.79	1.92	1.86
MLP Geo Init	1.39	1.29	1.33	1.34
Our Geo Init	1.04	1.03	1.09	1.05

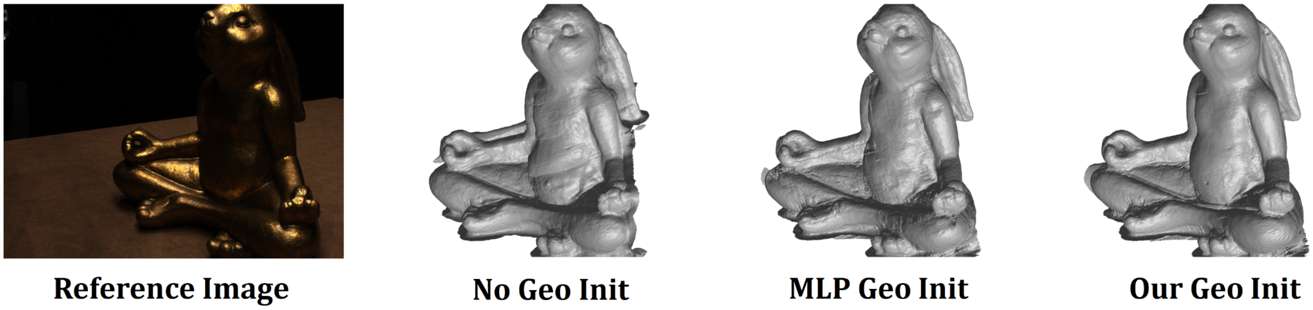

Figure 6. Qualitative evaluation for geometric initialization. First column: reference images. Second to the fourth column: no geometric initialization, geometric initialization for MLP, and our geometric initialization.

Table 5. Additional ablations and EMD evaluation on DTU.

Method	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	**Mean**
+ MSA + SAC (2+2 FB)	0.72	0.89	1.05	0.38	0.92	0.86	0.81	1.40	1.17	0.78	0.60	1.55	0.41	0.55	0.50	0.84
+ MSA + MPE (4FB)	0.98	0.86	0.82	0.38	0.90	0.82	0.74	1.36	1.16	0.75	0.55	1.61	0.35	0.53	0.50	0.82
+ SAC + MPE (2FB)	0.59	0.78	0.70	0.36	0.89	0.80	0.75	1.35	1.12	0.68	0.53	1.10	0.34	0.51	0.50	0.73
+ SAC + MPE (4FB)	0.56	0.75	0.68	0.36	0.87	0.76	0.69	1.33	1.08	0.66	0.51	1.04	0.34	0.51	0.48	**0.71**
+ SAC + MPE (6FB)	0.64	0.76	0.71	0.36	0.87	0.83	0.76	1.36	1.13	0.67	0.54	1.07	0.34	0.52	0.51	0.74
Earth Mover Distance (EMD) evaluation on DTU																
NeuS	1.03	1.08	0.85	0.96	1.22	0.80	0.84	0.98	1.01	0.88	0.71	0.87	0.75	0.80	0.82	0.91
Ours	0.87	0.92	0.71	0.93	1.01	0.85	0.85	0.95	1.02	0.84	0.70	0.82	0.74	0.80	0.81	0.85

B.2. Frequency Bands

We conduct an ablation study to investigate the impact of different frequency bands. Tab. 5 provides Chamfer distance results for 2, 4, and 6 frequency bands with the window sizes of {8}, {4,8,16}, and {1,2,4,8,16}, respectively. Four frequency bands performed the best in our experiments.

B.3. Other Combination of Architectures

We run the experiment for MSA+SAC and MSA+MPE and report the results in Tab. 5. To construct four frequency bands (4FB) for MSA+SAC, we use two-resolution triplanes for MSA and apply two-frequency-band SAC on each tri-plane. Compared with SAC+MPE (4FB) in Tab. 5, SAC+MPE method is superior to MSA+SAC and MSA+MPE.

B.4. EMD evaluation

We follow the very recent SotA methods (e.g., NeuS, VolSDF, IDR), which use *only* Chamfer distance to evaluate the surface quality. For additional EMD evaluation, we compare our method with NeuS and provide the results in Tab. 5. For each method, we evenly sampled 10K points on the reconstructed surface to obtain the point cloud. The results are consistent with Chamfer distance, and our method outperforms NeuS on DTU by 7% on average.

C. Method Details

C.1. Multi-Scale Architecture

Here, we discuss the details of the multi-scale architecture we compared to in the main text. One general strategy to achieve multi-scale behavior is to use a multi-resolution data structure. Therefore, it was our idea to use tri-planes with different resolutions. We use Tri-planes with four different resolutions in the multi-scale architecture to mimic four frequency bands in the SAC (self-attention convolution) structure for a fair comparison. For the highest resolution tri-plane representing high frequency, we used the same resolution as SAC, i.e. 512 dimensions. However, for the low-frequency tri-planes, we use three different tri-planes with different resolutions, namely 256×256, 128×128, and 64×64. We then concatenate features generated by these tri-planes to obtain multi-scale tri-plane features. We show the results in Table 3 in the main text and find that our results using the self-attention convolution outperform the results using the multi-scale architecture. An intuitive explanation

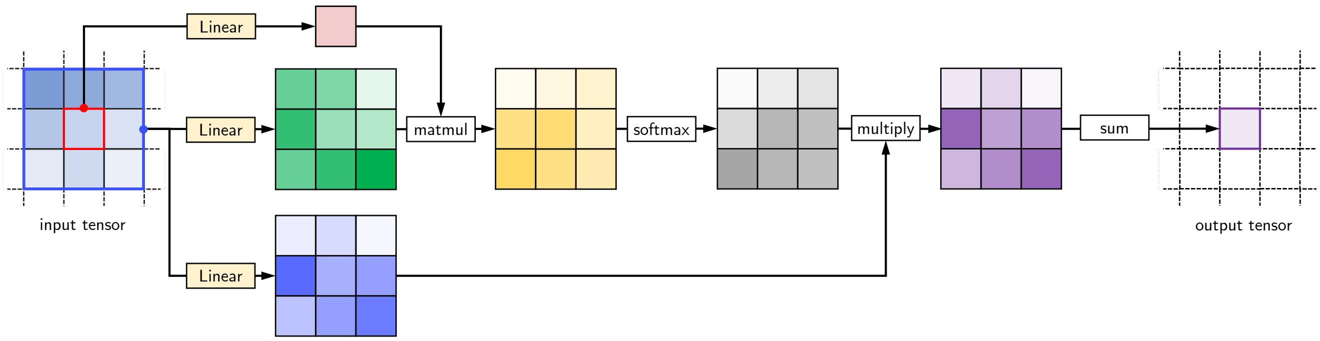

Figure 7. Self-Attention Convolution for a window size $k = 3$.

is that the smoothing kernel can deal with the influence of noise more effectively.

C.2. Self-Attention Convolution

We use the vanilla SAC mechanism from [2] and its detailed diagram is provided in Fig. 7. Larger window sizes represent lower frequency and smaller window sizes represent higher frequency. These window sizes are used as different kernels in the self-attention convolution and the corresponding convolved features are produced. Adding the original tri-plane features, we can get four triplane features with frequencies as our output feature maps.

D. Other Real-world Examples

We conduct an experiment on another real-world dataset namely CO3D [3]. This dataset provides data for real-world scenes. Many scenes are captured with portable devices and camera poses are extracted using COLMAP [4]. 3D reconstruction methods from datasets with this acquisition method are more general, but also introduce greater noise and challenges. We select some representative examples (bench, bicycle, and hydrant) and show the comparison with NeuS [5] on these examples in Fig. 8. Experimental results show that our method can reconstruct detailed features, such as the net structure of a bench, the rear wheel of a bicycle, and the rivets of a fire hydrant.

E. Additional Qualitative Comparisons

In this section, we provide more qualitative comparisons with NeuS [5] and HF-NeuS [6] for surfaces and images on the NeRF synthetic dataset (Fig. 9 and Fig. 10) and the DTU dataset (Fig. 11 and Fig. 12).

References

[1] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2565–2574, 2020. 3

[2] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. *Advances in Neural Information Processing Systems*, 32, 2019. 5

[3] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021. 5

[4] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 5

[5] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 2021. 5

[6] Yiqun Wang, Ivan Skorokhodov, and Peter Wonka. Improved surface reconstruction using high-frequency details. *Advances in Neural Information Processing Systems*, 2022. 5

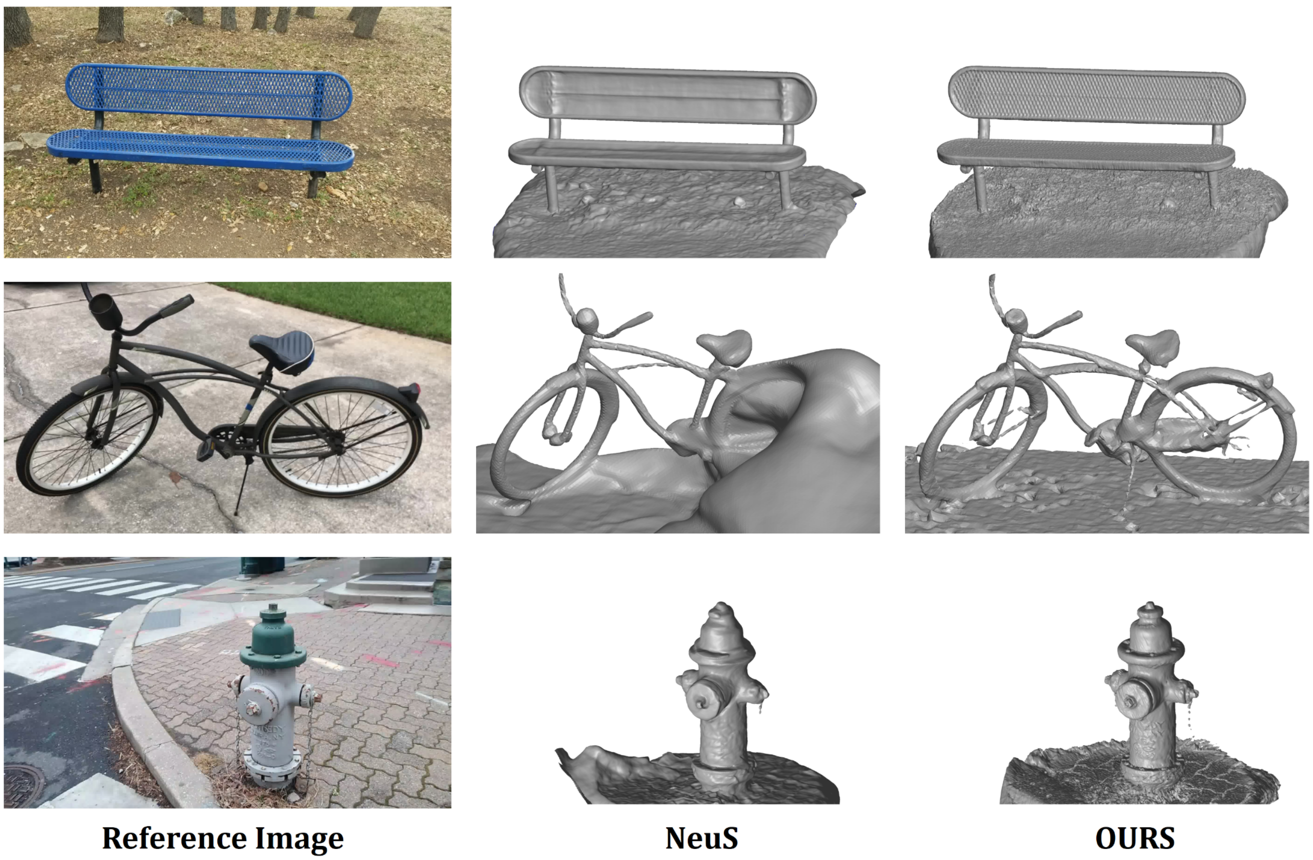

Figure 8. Qualitative evaluation on CO3D dataset. First column: reference images. Second to the third column: NeuS and PET-NeuS.

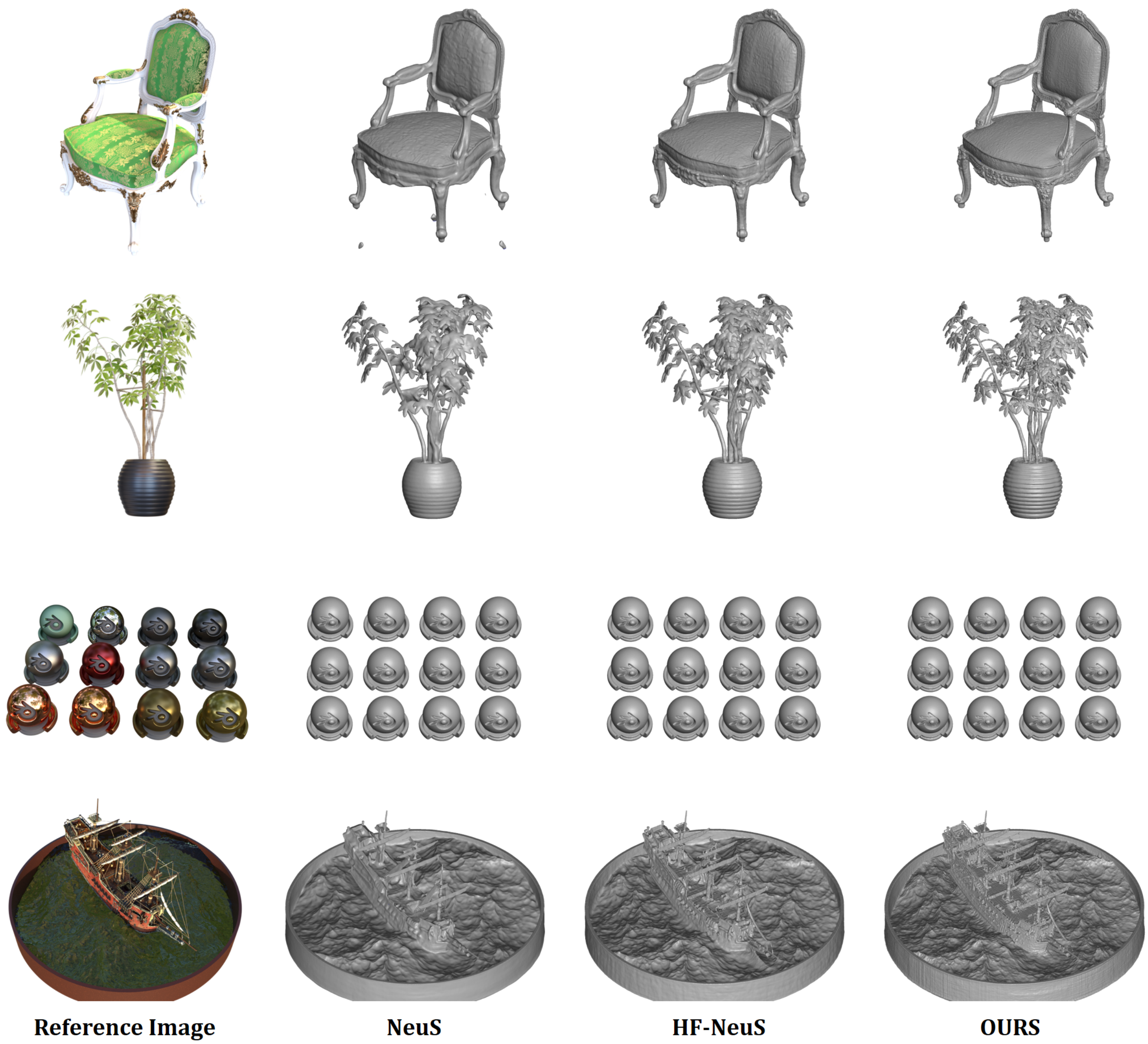

Figure 9. Qualitative evaluation on the NeRF synthetic dataset. First column: reference images. Second to the fourth column: NeuS, HF-NeuS, and PET-NeuS.

Figure 10. Qualitative evaluation on NeRF synthetic dataset. First column: reference images. Second to the fourth column: the generated images from NeuS, HF-NeuS, and PET-NeuS.

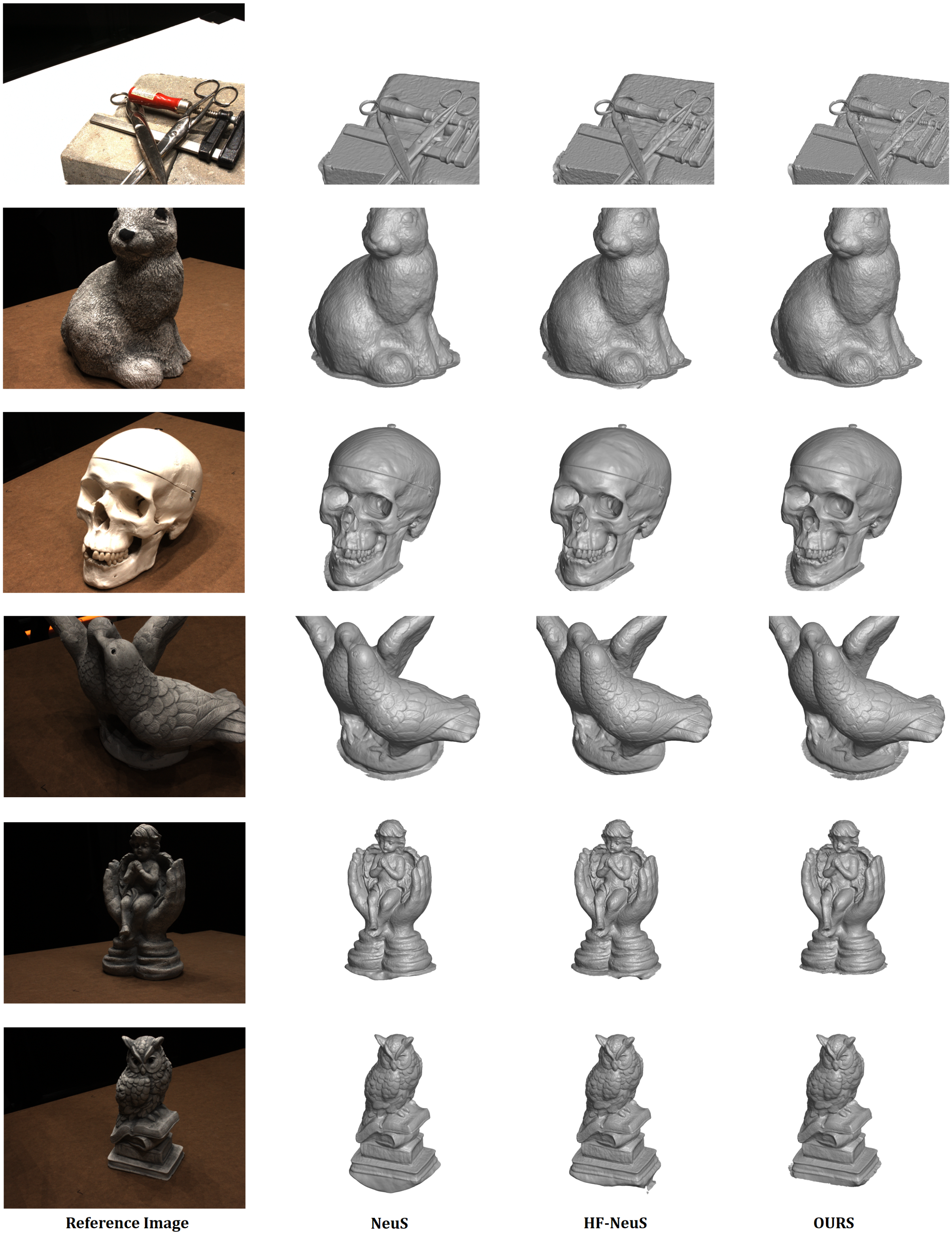

Figure 11. Qualitative evaluation on DTU dataset. First column: reference images. Second to the fourth column: NeuS, HF-NeuS, and PET-NeuS.

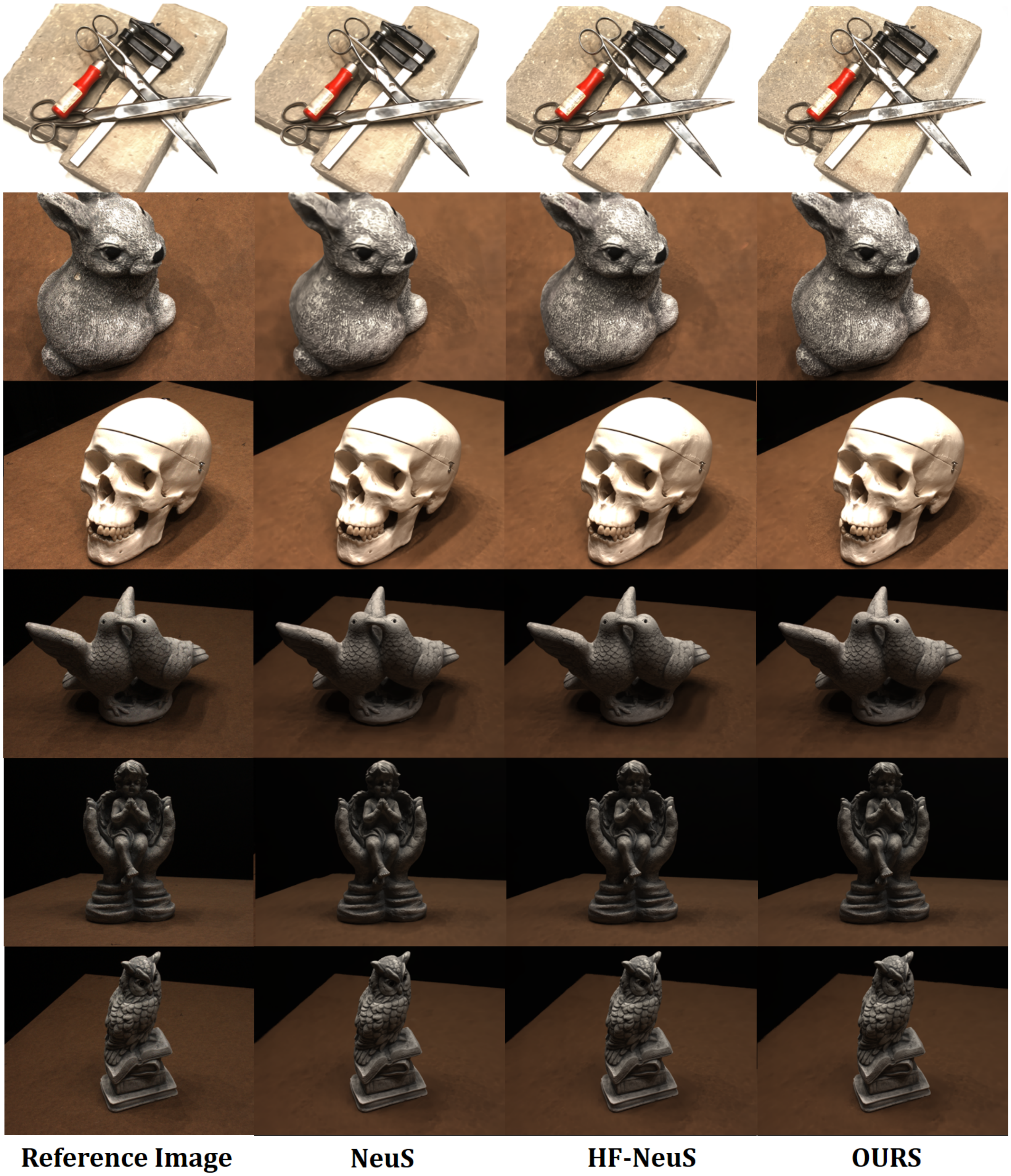

Figure 12. Qualitative evaluation on DTU dataset. First column: reference images. Second to the fourth column: the generated images from NeuS, HF-NeuS, and PET-NeuS.