PET-NeuS: Positional Encoding Tri-planes for Neural Surfaces

Anonymous authors
Paper under double-blind review

Abstract

The signed distance function (SDF) represented by an MLP network is commonly used for multi-view neural surface reconstruction. We build on the successful recent method NeuS to extend it by three new components. The first component is to borrow the Tri-plane representation from EG3D and represent signed distance fields as a mixture of tri-planes and MLPs instead of representing it with MLPs only. Discretizing the scene space with Tri-planes leads to a more expressive data structure but involving tri-planes will introduce noise due to discrete discontinuities. The second component is to use a new type of positional encoding with learnable weights to combat noise in the reconstruction process. We divide the features in the tri-plane into multiple frequency bands and modulate them with sin and cos functions of different frequency. The third component is to use learnable convolution operations on the tri-plane features using self-attention convolution to produce features with different frequency. The experiments show that PET-NeuS achieves high-fidelity surface reconstruction on standard datasets. Following previous work and using the Chamfer metric as the most important way to measure surface reconstruction quality, we are able to improve upon the NeuS baseline by 25% on Nerf-synthetic (0.84 compared to 1.12) and by 14% on DTU (0.75 compared to 0.87). The qualitative evaluation reveals how our method can better control the interference of high-frequency noise.

1 Introduction

Implicit neural functions, or neural fields, have received a lot of attention in recent research. The seminal paper NeRF (Mildenhall et al., 2020) combines implicit neural functions with volume rendering, enabling high-quality novel view synthesis. Inspired by NeRF, NeuS (Wang et al., 2021) and VolSDF (Yariv et al., 2021) introduce a signed distance function into the volume rendering formula and regularize the signed distance function, so that smooth surface models can be reconstructed. However, these methods use pure MLP networks to encode signed distance functions. Although these two methods can reconstruct smooth surfaces, they both leave room for improvement when it comes to reconstructing surface details.

One research direction (Yu et al. (2021); Reiser et al. (2021); Müller et al. (2022); Chen et al. (2022); Chan et al. (2022)) explores explore data structures such as tri-planes or voxel grids that are suitable to improve the NeRF framework, in terms of speed or reconstruction quality. However, data structures that are successful for novel view synthesis may not be bring immediate success when employed for surface reconstruction as shown in Fig. 1. While a greater expressiveness to encode local details is useful to better fit the input data, due to the discontinuities caused by discretization, these data structures may generate a large amount of unavoidable noise interference. These noise disturbances can seriously affect the fidelity of the reconstructed surface.

In our work, we explore how to increase expressiveness to encode local features while at the same time reducing the impact of noise interference. We choose to build on the tri-plane data structure since it has fewer discretization discontinuities than a voxel grid. In addition, the tri-planes can be easier scaled to higher resolutions.

In our work, we build on EG3D and NeuS to propose a novel framework, called PET-NeuS. First, we propose a method to integrate the tri-plane data structure into a surface reconstruction framework in order to be able to model an SDF with more local details. Second, since the source of noise

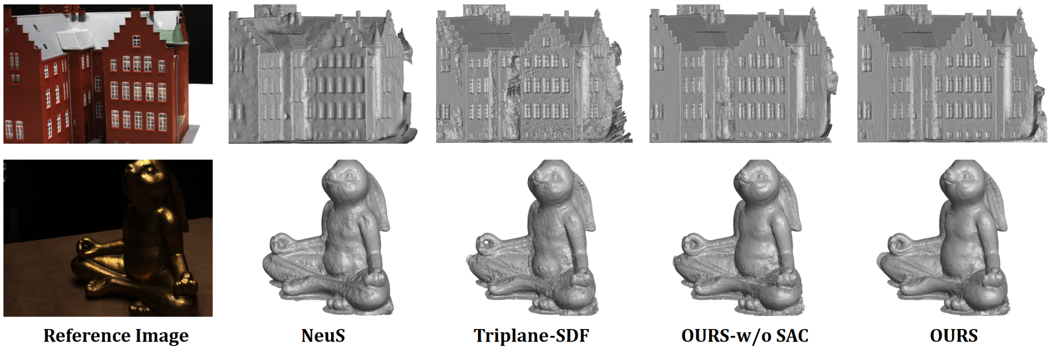

Figure 1: The challenge of using the tri-plane representation directly. First column: reference image. Second to the fifth column: NeuS, Learning SDF using tri-planes, OURS without self-attention convolution, and OURS.

are discretization discontinuities, that is, the features between tri-plane pixels do not share learnable parameters, we use positional encoding to modulate the tri-plane features, thereby enhancing the continuity of the learnable features. Third, the positional encoding involves functions of different frequencies. In order to better match different frequencies, we propose to use multi-scale self-attention convolution kernels with different window sizes to perform convolution in the spatial domain to generate features of different frequency bands. This further increases the fidelity of the surface reconstruction while suppressing noise.

We experiment on two datasets to verify the effectiveness of our method, the DTU dataset and the NeRF-Synthetic dataset. Since the DTU dataset contains non-Lambertian surfaces, the ability of the network to resist noise interference can be verified. The NeRF-Synthetic dataset has many sharp features, which can verify that our framework can effectively utilize its improved local expressiveness to better reconstruct local details. We show superior performance compared to state-of-the-art methods on both datasets.

In summary, our contributions are as follows:

- We propose to train neural implicit surfaces with a tri-plane architecture to enable the reconstructed surfaces to better preserve fine-grained local features.
- We derive a novel positional encoding strategy to be used in conjunction with tri-plane features in order to reduce the noise interference caused by discretization discontinuities.
- We utilize self-attention convolution to produce tri-plane features with different frequency bands to match the positional encoding of different frequencies, further improving the fidelity of surface reconstruction.

2 Related Work

2.1 Implicit Neural Representation

Learning continuous implicit representations with neural networks to represent 3D scenes has recently gained a lot of attention. IM-Net (Chen & Zhang, 2019) and occupancy networks (Mescheder et al., 2019) propose to learn an occupancy function representing shapes using MLPs. Unlike them, DeepSDF (Park et al., 2019) utilizes MLP networks to construct signed distance functions for shapes. In order to improve the representation ability of the models, Peng et al. (2020); Chibane et al. (2020a) use 3D convolution on the local voxels to learn local shape features and construct the occupancy function and signed distance function of the shapes respectively. Due to locality, implicit neural function representations can model fine-grained scenes. Subsequently, some works (Atzmon & Lipman, 2020; Chibane et al., 2020b) focus on solving the problem of learning implicit functions on shapes with boundary, while others (Martel et al., 2021; Takikawa et al., 2021; Williams et al., 2021) further exploit voxel representations to improve the quality of modeling. Then the seminal

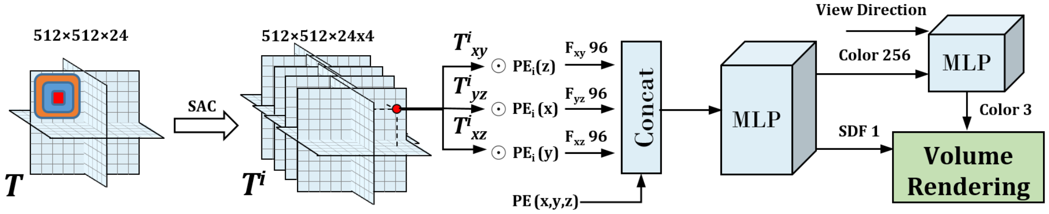

Figure 2: Our PET-NeuS framework consists of a tri-plane architecture, two types of positional encoding, self-attention convolution (SAC), and MLP mapping blocks.

work, NeRF (Mildenhall et al., 2020), incorporates the implicit neural function into the volume rendering formula, thus achieving high-fidelity rendering results. Due to the implicit neural function representing the scene, the method produces excellent results for novel view synthesis. Some follow-up works (Sitzmann et al., 2020; Barron et al., 2021; Verbin et al., 2021) use multiscale techniques to learn fine-grained details. Recently, many works (Yu et al., 2021; Reiser et al., 2021; Müller et al., 2022; Chen et al., 2022; Chan et al., 2022) use voxel grids or a factored representation (e.g. tri-planes) to further improve training speed or rendering quality.

2.2 Neural Surface Reconstruction from Multi-view Images

Surface reconstruction from multiple views is a popular topic in 3D vision. Traditional algorithms for multi-view surface reconstruction usually use discrete voxel-based representations (De Bonet & Viola, 1999; Seitz & Dyer, 1999; Kutulakos & Seitz, 2000; Broadhurst et al., 2001; Izadi et al., 2011; Nießner et al., 2013) or reconstruct point clouds (Barnes et al., 2009; Furukawa & Ponce, 2009; Schönberger et al., 2016; Schonberger & Frahm, 2016; Galliani et al., 2016). Discrete voxel representations suffer from resolution and memory overhead, while point-based methods require additional consideration of missing point clouds and additional surface reconstruction steps. Recently, some methods based on neural implicit surfaces have emerged to reconstruct shapes using continuous neural implicit function from multi-view images. Surface rendering and volume rendering are two key techniques. DVR (Niemeyer et al., 2020) and IDR (Yariv et al., 2020) adopt surface rendering to model the occupancy functions or signed distance functions for 3D shapes respectively. The methods based on surface rendering need to compute precise location of the surface to render images and gradient decent is applied only on the surface. NeRF-based methods like UNISURF (Oechsle et al., 2021), VolSDF (Yariv et al., 2021), and NeuS (Wang et al., 2021) incorporate occupancy functions or the signed distance functions into the volume rendering equation. Since the implicit function can be regularized by the Eikonal loss, the reconstructed surface can maintain smoothness. The NeuralPatch method by Darmon et al. (2022) is a post-processing step to VolSDF. It binds the colors in the volume to nearby patches with a homography transformation. Since the computation of patch warping relies on accurate surface normals, we consider the algorithm as a post-process that can be applied to any method. HF-NeuS Wang et al. (2022) introduces an additional MLP for modeling a displacement field to learn high-frequency details and further improve the surface fidelity. We choose VolSDF, NeuS, and HF-NeuS as our state-of-the-art competitors.

3 Method

Given N images including pose information, our goal is to reconstruct the geometry of the scene represented by a signed distance function. We build on the NeuS framework (Wang et al., 2021) and first introduce how to use tri-planes and MLPs to learn a neural implicit function of a surface and propose a simple approach to initialize the tri-plane features. Second, we show a theoretical derivation to explain how to embed positional encoding in the tri-plane. Finally, we introduce how to learn multi-frequency tri-plane features. The framework is shown in Fig. 2

3.1 SDF-based Tri-planes with Geometric Initialization

Here we describe how to integrate the tri-plane data structure proposed in EG3D (Chan et al., 2022) into the NeuS framework (Wang et al. (2021), Wang et al. (2022)). Signed distance field functions (SDFs) are the most common way of representing surfaces with implicit functions. The input of such a function is a triple of three-dimensional coordinates (x, y, z), and the output is the signed distance d_s to the surface.

$$(x, y, z) \mapsto d_s = sdf\,(x, y, z)\,, \quad x, y, z, d_s \in \mathbb{R} \tag{1}$$

To model the SDF, we use a discrete representation of three 2D feature maps, called tri-planes (Chan et al., 2022). A tri-plane T is composed of three learnable feature maps T_{xy}, T_{yz}, T_{xz}, and the size of each feature map is $R \times R \times n_f$. R is the resolution of the feature map and n_f is the number of channels of the feature map. These three planes are orthogonal to each other, and they are positioned such that the three planes jointly intersect at the origin $(0, 0, 0)$ in the scene space and have a length of L. Given a 3D coordinate (x, y, z), we can project the 3D coordinate onto the three planes and interpolate three feature vectors $T(x, y, z) = (\boldsymbol{v}_{xy}, \boldsymbol{v}_{yz}, \boldsymbol{v}_{xz})$. Then a small MLP is used to predict the distance value $d_s = MLP(\boldsymbol{v}_{xy}, \boldsymbol{v}_{yz}, \boldsymbol{v}_{xz})$ taking these feature vectors as input. A second MLP is used to predict a view-dependent color value.

Subsequently, we need to incorporate the sdf into the volume rendering equation. There are multiple ways to do this. We chose the modeling approach of HF-NeuS for a fair comparison, which models transparency as the transformed SDF and obtains the density value σ as follows.

$$\sigma(x, y, z) = s\left(\Psi_s\left(sdf\,(x, y, z)\right) - 1\right) \nabla sdf\,(x, y, z) \cdot \mathbf{d} \tag{2}$$

where Ψ_s is the sigmoid function with scale parameter s and $\mathbf{d}$ is the viewing direction. The volume rendering integral is then approximated using α-composition $\alpha_i = 1 - exp\,(-\sigma_i \delta_i)$, where δ_i is the distance in scene space between adjacent samples. We can use the volume rendering equation in NeRF (Mildenhall et al., 2020) to render the color value of a pixel corresponding to a ray.

While an SDF can be obtained by training using an L1 loss that minimizes the difference of volume rendering color and ground truth color, how to get a reasonable neural implicit function highly depends on the initialization of the tri-plane features.

Since the tri-plane is a 2D projection representation, directly initializing the tri-plane features as a circle with the remaining MLP networks initialized by the geometric initialization (Atzmon & Lipman, 2020) cannot guarantee that the initialized SDF is a sphere. We therefore propose a simple approach to leverage the MLP initialization method (Atzmon & Lipman, 2020) to initialize the tri-plane features. We build a five-layer MLP whose input feature dimension is 3 (for inputing coordinates) and the output feature dimension is the tri-plane channel dimension n_f. The MLP is initialized using the geometric initialization (Atzmon & Lipman, 2020). For each pixel on the plane, there is a scene coordinate corresponding to the pixel. We select $R \times R$ grid points on each plane and input the resulting coordinates of the points into the MLP. The features output by the MLP are used as the initial tri-plane features. In this way, the SDF corresponding to the initialized tri-plane features represents a sphere. In addition, since the MLP is used during initialization and not optimized during training, this

3.2 Incorporating Positional Encoding into Tri-planes

Positional encoding is very important in neural rendering. A popular method uses sin and cos functions to map coordinate information to multiple different frequencies, so that the network can better capture the characteristics of different frequency bands. However, the tri-plane is an interpolation-based representation indexed by 3D spatial coordinates, so it is difficult to intuitively utilize the positional encoding information. We first propose a modeling approach. The proposed method expresses the implicit function as a weighted sum of the positional encoding by mathematical derivation. We then propose how to introduce positional encoding into the tri-plane representation.

The goal of learning neural implicit functions is to learn the mapping of coordinates to the function values. For a continuous unary function with compact support, it can be expanded as a Fourier series as follows. We could use an MLP network to encode the coefficients of this series (in this case even a single linear layer without activation function).

$$f\left(x\right) = a_0 + \sum_{m=1}^{M} a_m \cos\left(mx\right) + \sum_{m=1}^{M} a_{(-m)} \sin\left(mx\right) \tag{3}$$

$$= MLP(\{\cos(mx), \sin(mx)\}_m) \tag{4}$$

where the coefficients can be computed as follows.

$$a_0 = \frac{1}{2\pi} \int_{2\pi} f\left(x\right) dx \tag{5}$$

$$a_m = \frac{1}{\pi} \int_{2\pi} f\left(x\right) \cos(mx) dx \tag{6}$$

$$a_{(-m)} = \frac{1}{\pi} \int_{2\pi} f\left(x\right) \sin(mx) dx \tag{7}$$

We can simplify the above equation as follows.

$$f\left(x\right) = \sum_{m=-M}^{M} a_m \Theta_m^x \tag{8}$$

$$\tag{9}$$

where

$$\Theta_t^v = \begin{cases} \cos\left(tv\right) & t > 0 \\ 1 & t = 0 \\ \sin\left(tv\right) & t < 0 \end{cases} \tag{10}$$

Similarly, in the three-dimensional coordinate space, the function can be expanded. We can make the following derivation.

$$f\left(x, y, z\right) = \sum_{k=-K}^{K} c_k\left(x, y\right) \Theta_k^z \tag{11}$$

$$= \left(c_0(x, y) + \sum_{k=1}^{K} c_k(x, y) \cos\left(kz\right) + \sum_{k=1}^{K} c_{-k}(x, y) \sin\left(kz\right) \right) \tag{12}$$

$$= \sum_{k=-K}^{K} \sum_{n=-N}^{N} b_{nk}\left(x\right) \Theta_n^y \Theta_k^z \tag{13}$$

$$= \sum_{k=-K}^{K} \left(b_{0k}\left(x\right) + \sum_{n=1}^{N} b_{nk}\left(x\right) \cos\left(ny\right) + \sum_{n=1}^{N} b_{(-n)k}\left(x\right) \sin\left(ny\right) \right) \Theta_k^z \tag{14}$$

$$= \sum_{k=-K}^{K} \sum_{n=-N}^{N} \sum_{m=-M}^{M} a_{mnk} \Theta_m^x \Theta_n^y \Theta_k^z \tag{15}$$

$$= \sum_{k=-K}^{K} \sum_{n=-N}^{N} \left(a_{0nk} + \sum_{m=1}^{M} a_{mnk} \cos\left(mx\right) + \sum_{m=1}^{M} a_{(-m)nk} \sin\left(mx\right) \right) \Theta_n^y \Theta_k^z \tag{16}$$

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

We observe that the above function can be replaced with an MLP network with different input as follows.

$$f\left(x, y, z\right) = MLP\left(\left\{ \begin{matrix} \cos\left(mx\right) & \cos\left(ny\right) & \cos\left(kz\right) \\ \sin\left(mx\right) & \sin\left(ny\right) & \sin\left(kz\right) \\ g_m\left(y, z\right) \cos\left(mx\right) & h_n\left(x, z\right) \cos\left(ny\right) & w_k\left(x, y\right) \cos\left(kz\right) \\ g'_m\left(y, z\right) \sin\left(mx\right) & h'_n\left(x, z\right) \sin\left(ny\right) & w'_k\left(x, y\right) \sin\left(kz\right) \end{matrix} \right\}_{mnk}^{\textbf{flatten}} \right) \tag{17}$$

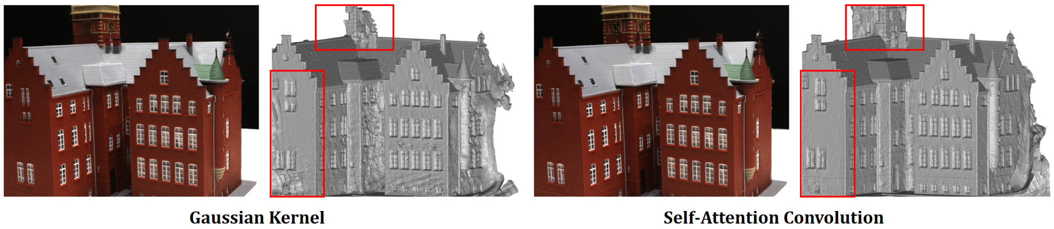

Figure 3: Comparing Gaussian kernels with our self-attention kernels. For each method, the left shows the reconstructed image and the right the reconstructed surface.

To represent the surface as SDF, the function f can be regarded as sdf. In pure MLP networks, $sdf(x, y, z) = MLP(PE(x, y, z))$. Since g, h, and w are all highly nonlinear functions, the number of MLP layers needs to be large. A simple idea of using positional encoding with a tri-plane is to directly concatenate tri-plane features and positional encoding and input the features into an MLP, i.e. $sdf(x, y, z) = MLP([PE(x, y, z), T(x, y, z)])$. Then the tri-plane feature $T(x, y, z)$ is used to encode these highly nonlinear functions, for example, one of the terms $g_m\,(y, z) \cos{(mx)}$, which then only requires fewer layers of the MLP on top of the tri-plane features. However, since the tri-planes do not know any positional encoding information, they cannot effectively fit base signals of different frequencies, such as the above example term. Due to discrete discontinuities of the tri-plane representation and the absence of frequency constraints, the tri-plane features will introduce high-frequency noise. From Eq. 17, we observe that the coefficients g, h, and w of positional encoding is consistent with the tri-plane features since these coefficients are all binary functions of the two-dimensional coordinates. We propose to regard g, h, and w function as the tri-plane features and modulate/multiply them with sin and cos functions of different frequency. In this way, the output features of tri-planes contain a frequency bound and the base function can be fitted more easily. To be specific, our definition is as follows.

$$sdf(x, y, z) = MLP([PE(x, y, z), T_{xy}(x, y) \odot PE(z), T_{yz}(y, z) \odot PE(x), T_{xz}(x, z) \odot PE(y)]) \tag{18}$$

where $\odot$, e.g. $T_{xy}(x, y) \odot PE(z)$, is the component-wise multiplication. Using our modeling method, not only can the output features of the tri-planes contain frequency information to suppress high-frequency noise, but also reduced nonlinearity and learning complexity of triplane features.

3.3 Learning Self-attention Features with Different Frequency on Tri-planes

We could directly set the number of channels of the tri-plane features according to the dimension of the positional encoding. However, in order to better learn features of different frequencies, we propose to generate tri-plane features with different frequency bands.

The product in the frequency domain is equal to the convolution in the time domain. One simple way to generate multi-frequency tri-plane features is to smooth the tri-plane features with a set of fixed Gaussian smoothing kernels. When experimenting with this simple method for generating features with different frequency bands as shown in Fig. 3, we noticed that this method smooths features across depth discontinuities, e.g. foreground to background. Since the plane is an orthogonal projection of the 3D space, the e.g. foreground features will be affected by the background features due to convolution on the plane, resulting in the wrong structure of the generated surface although the synthesized image is reasonable. Therefore, we looked for alternative dynamic convolution operations that could be performed. Inspired by window self-attention convolution (Ramachandran et al., 2019) and the Swin Transformer architecture (Liu et al., 2021), we propose to use self-attention convolution with different window sizes to generate features in different frequency bands. We found that using either a sliding window or a shifted window exponentially increases the computational cost of training. To reduce the computational cost, we use a single-layer self-attention convolution and directly divide the tri-plane into non-overlapping patches regularly with different window sizes (4, 8, 16 in praxis). Since the subsequent MLP will combine features of different scales, the features across-windows will also interact. We selected features generated by convolution with three window sizes and concatenate them with the original features to form tri-plane features for four frequency

Table 1: Quantitative results on the NeRF-synthetic dataset. Chamfer distance on the left and PSNR on the right.

Method	Chair	Ficus	Lego	Materials	Mic	Ship	**Mean**	Chair	Ficus	Lego	Materials	Mic	Ship	**Mean**
VOLSDF	1.26	1.54	2.83	1.35	3.62	2.92	2.37	25.91	24.41	26.99	28.83	29.46	25.65	26.86
NeuS	0.74	1.21	2.35	1.30	3.89	2.33	1.97	27.95	25.79	29.85	29.36	29.89	25.46	28.05
HF-NeuS	0.69	1.12	0.94	1.08	0.72	2.18	1.12	28.69	26.46	30.72	29.87	30.35	25.87	28.66
OURS	**0.65**	**0.71**	**0.58**	**1.05**	**0.49**	**1.57**	**0.84**	**29.57**	**27.39**	**32.40**	**29.97**	**33.08**	**26.83**	**29.87**

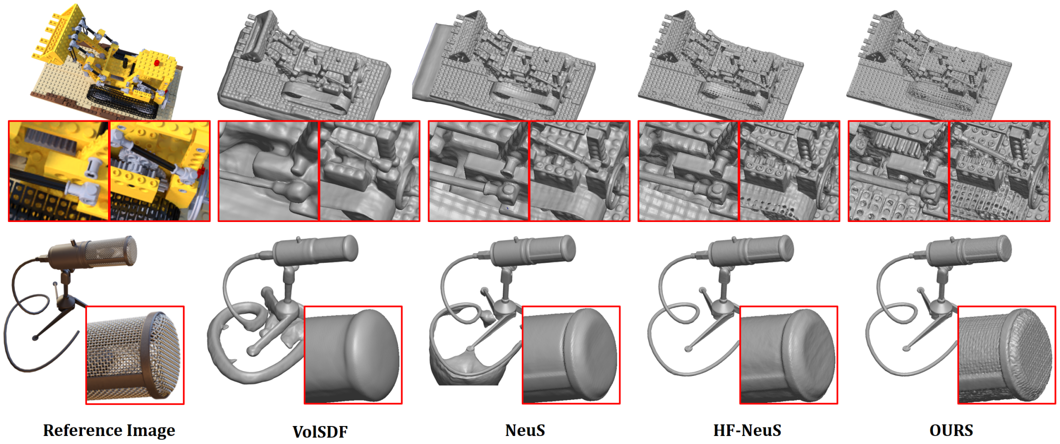

Figure 4: Qualitative evaluation on the Lego and Mic models. First column: reference images. Second to the fifth column: VolSDF, NeuS, HF-NeuS, and OURS.

bands as follows.

$$T = \left\{T^i\right\}, i = 0, 1, 2, 3. \quad T^i = \left\{T^i_{xy}, T^i_{yz}, T^i_{xz}\right\} \tag{19}$$

We multiply the features of the four frequency bands with the corresponding low-frequency to high-frequency positional encoding, so as to achieve the goal of generating adaptive features for different frequencies.

3.4 Optimization

We use two different losses in the training (identical to what has been used in previous work NeuS and HF-NeuS). The first one is the color reconstruction loss. The second is the Eikonal loss (Gropp et al., 2020). We found that total variation regularization (Lombardi et al., 2019) (TVloss) can also regularize the SDF like Eikonal loss, but Eikonal loss is especially suitable for learning SDF. Color reconstruction loss is the L1 distance between ground truth colors and the volume rendered colors of sampled pixel set S .

$$\mathcal{L}_{color} = \frac{1}{|S|} \sum_{s \in S} \left\| \hat{C}_s - C_s \right\|_1 \tag{20}$$

Eikonal loss is a regularization loss on sampled point set I that constrain the implicit function and make the SDF smooth.

$$\mathcal{L}_{reg} = \frac{1}{|I|} \sum_{i \in I} \left[\left(\left\| \nabla sdf(x_i, y_i, z_i) \right\|_2 - 1 \right)^2 \right] \tag{21}$$

We employ both loss functions to train our network with a hyperparameter λ. Note that in all settings we do not provide masks and ignore mask loss in the training.

$$\mathcal{L} = \mathcal{L}_{color} + \lambda \mathcal{L}_{reg} \tag{22}$$

Table 2: Quantitative results on the DTU dataset. Chamfer distance on top and PSNR on the bottom.

Method	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	**Mean**
VOLSDF	1.14	1.26	0.81	0.49	1.25	0.70	0.72	1.29	1.18	0.70	0.66	**1.08**	0.42	0.61	0.55	0.86
NeuS	1.37	1.21	0.73	0.40	1.20	0.70	0.72	**1.01**	1.16	0.82	0.66	1.69	0.39	**0.49**	0.51	0.87
HF-NeuS	0.76	1.32	**0.70**	0.39	1.06	**0.63**	**0.63**	1.15	1.12	0.80	0.52	1.22	**0.33**	0.49	**0.50**	0.77
OURS	**0.61**	**0.96**	0.77	**0.34**	**0.98**	0.85	0.71	1.35	**1.04**	**0.66**	**0.50**	1.11	0.39	0.53	0.53	**0.75**
VOLSDF	26.28	25.61	26.55	26.76	31.57	31.50	29.38	33.23	28.03	32.13	33.16	31.49	30.33	34.90	34.75	30.38
NeuS	28.20	27.10	28.13	28.80	32.05	33.75	30.96	34.47	29.57	32.98	35.07	32.74	31.69	36.97	37.07	31.97
HF-NeuS	29.15	27.33	28.37	28.88	32.89	**33.84**	**31.17**	34.83	**30.06**	33.37	35.44	33.09	32.12	37.13	37.32	32.33
OURS	**30.02**	**27.52**	**29.03**	**29.76**	**33.68**	33.49	30.91	**35.17**	29.42	**33.45**	**36.65**	**33.62**	**32.30**	**38.52**	**37.56**	**32.74**

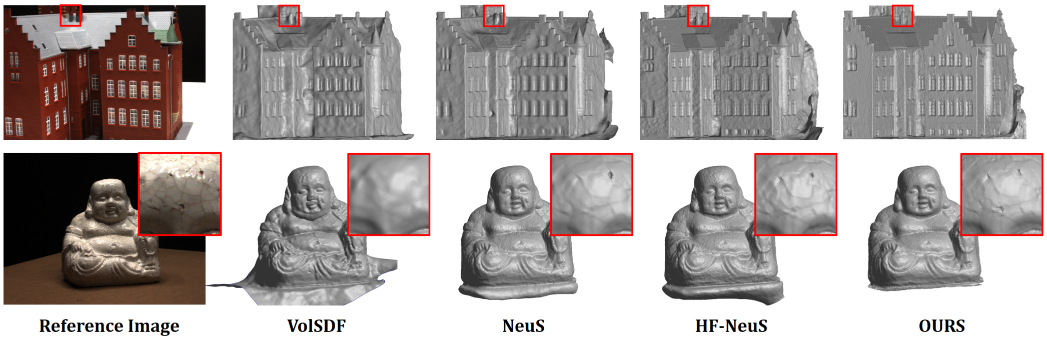

Figure 5: Qualitative evaluation on DTU house and Buddha models. First column: reference images. Second to the fifth column: VolSDF, NeuS, HF-NeuS, and OURS.

4 Results

Datasets. The NeRF synthetic dataset contains posed multi-view images of 800 × 800 resolution with detailed and sharp features. The DTU dataset is a real dataset that contains posed multi-view images of 1600 × 1200 resolution. We select the same 15 models as shown in other works for a fair comparison. The DTU dataset contains non-Lambertian surfaces which are testing for methods sensitive to noise. Besides the DTU dataset, 6 challenging scenes are selected from the NeRF-synthetic dataset (Mildenhall et al., 2020). Ground truth surfaces and camera poses are provided in both datasets.

Baselines. Three state-of-the-art baselines are considered: VolSDF (Yariv et al., 2021) embeds an SDF into the density function and employs an error bound by using a sampling strategy. The training time is 12 hours on the DTU dataset. NeuS (Wang et al., 2021) incorporates an SDF into the weighting function and uses sigmoid functions to control the slope of the function. The training time is 16 hours on the DTU dataset. HF-NeuS (Wang et al., 2022) builds on NeuS using offset functions. The training time is 20 hours on the DTU dataset. Since NeuS and VolSDF compared to older methods and demonstrated better results for surface reconstruction, we do not compare with methods such as NeRF (Mildenhall et al., 2020), IDR (Yariv et al., 2020), or UNISURF (Oechsle et al., 2021).

Evaluation metrics. For the DTU dataset, we follow the official evaluation protocol to evaluate the Chamfer distance. For the NeRF synthetic dataset, we compute the Chamfer distance between the ground truth shape and reconstructed surface. For completeness, PSNR metric is used to measure the quality of reconstructed images. However, we would like to emphasize that the Chamfer distance is the most important metric for comparing surface reconstruction methods.

Implementation details. We use two MLPs to model the SDF and color function on top of tri-plane features. Each MLP consists of only 3 layers. The hyperparameter for the Eikonal regularization is $\lambda = 0.1$. We normalize scenes to fit $L = 3.0$. The resolution of each tri-plane is 512 × 512. The number of tri-plane feature channels is $n_f = 24$. Our three window sizes are set to 4, 8, and 16. We use positional encoding with 8 scales, which means $M = M = K = 8$. The Adam optimizer

Table 3: Ablation study results (Chamfer distance).

Method	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	**Mean**
TriSDF	1.23	1.12	1.28	0.46	1.08	0.90	0.84	1.40	1.21	0.85	0.58	1.76	0.42	0.59	0.56	0.95
TriSDF+MPE	0.69	1.04	0.96	0.42	1.00	0.87	0.75	1.38	1.19	0.81	0.55	1.38	0.41	0.54	0.55	0.84
TriSDF+PE+SCF	0.61	0.96	0.77	0.34	0.98	0.85	0.71	1.35	1.04	0.66	0.50	1.11	0.39	0.53	0.53	0.75

with a learning rate $5e^{-4}$ is utilized for network training using a single NVIDIA TITAN A100 40GB graphics cards. The training time is 9 hours on the DTU dataset, which is faster than all competitors.

4.1 Comparison

We first report quantitative comparisons on the NeRF-synthetic dataset (Mildenhall et al., 2020). In Table 1, we show Chamfer distance on the left and the PSNR values on the right. The results show that our proposed framework PET-NeuS has the best surface reconstruction quality compared to all other methods. This means that our network has the ability to better preserve local features. Besides outperforming other baselines in terms of quantitative error, we also show the visual effect of the improved reconstruction (Fig. 4). We find the reconstruction of the bumps and the wheel holes of Lego model and the grid of the Mic model to be particularly impressive. The reconstructed fine-grained structures are a lot better than what can be achieved with previous work.

The quantitative results on the DTU dataset (Jensen et al., 2014) are shown in Table 2. We show Chamfer distance on the top and the PSNR values on the bottom. For the Chamfer distance, PET-NeuS surpasses NeuS and VolSDF. Compared with HF-NeuS, PET-NeuS is comparable or slightly better. Our PSNR outperforms all other competitors. The qualitative results compared with other methods are shown in Fig. 5. The reconstructed surfaces by PET-NeuS preserve fine-grained details. For instance, the holes between the eyes of the Buddha and the windows are more obvious.

4.2 Ablation study

In Table 3, we conduct an ablation study to analyze the effect of each component. “TriSDF” refers to using tri-planes and MLPs to encode SDFs, which is $sdf(x,y,z) = MLP([PE(x,y,z),T(x,y,z)])$ as discussed in the method section. ”MPE” means we modulate tri-plane features with positional encoding as in Eq. 18. ”SAC” refers to generating features with different frequencies using self-attention convolution. We conduct experiments on the DTU dataset quantitatively. From the results, we can observe that the result of using only ”TriSDF” still results in a large geometric error. We believe that this is due to the discretization discontinuities. Modulating tri-plane features using positional encoding can suppress noise interference. Using self-attention convolution will match the positional encoding on different frequencies, which can further improve the geometric fidelity. From Fig. 1, we can also observe an improvement in reconstruction quality.

5 Conclusion and Limitations

We propose PET-NeuS, a novel tri-plane based method for multi-view surface reconstruction. By modulating tri-plane features using positional encoding and producing tri-plane features with different frequencies using self-attention convolution, our surface reconstruction can reduce noise interference while maintaining high fidelity. PET-NeuS produces fine-grained surface reconstruction and outperforms other state-of-the-art competitors in qualitative and quantitative comparisons. One limitation of our method is that it still requires long computation times. It would be an exciting avenue of future work to improve computation times by one or two orders of magnitude without drastically sacrificing quality. Another limitation that we observed is a trade-off between reconstructing fine details and adding high frequency noise to otherwise flat surface areas. As we experimented with many versions of our framework, we observed that network architectures that are more expressive to model surface detail tend to be more prone to overfitting and hallucinating details, e.g. in areas of high-frequency changes in light transport. It would be interesting to investigate this trade-off from a theoretical perspective in future work. Finally, we would like to state that we do not expect a noteworthy negative societal impact due to research on surface reconstruction.

REFERENCES

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*, pp. 2565–2574, 2020.

Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009.

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5855–5864, 2021.

Adrian Broadhurst, Tom W Drummond, and Roberto Cipolla. A probabilistic framework for space carving. In *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, volume 1, pp. 388–393. IEEE, 2001.

Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16123–16133, 2022.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. *arXiv preprint arXiv:2203.09517*, 2022.

Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5939–5948, 2019.

Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6970–6981, 2020a.

Julian Chibane, Gerard Pons-Moll, et al. Neural unsigned distance fields for implicit function learning. *Advances in Neural Information Processing Systems*, 33:21638–21652, 2020b.

François Darmon, Bénédicte Bascle, Jean-Clément Devaux, Pascal Monasse, and Mathieu Aubry. Improving neural implicit surfaces geometry with patch warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6260–6269, 2022.

Jeremy S De Bonet and Paul Viola. Poxels: Probabilistic voxelized volume reconstruction. In *Proceedings of International Conference on Computer Vision (ICCV)*, pp. 418–425, 1999.

Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009.

Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Gipuma: Massively parallel multi-view stereo reconstruction. *Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e. V*, 25(361-369):2, 2016.

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020.

Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 559–568, 2011.

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*, pp. 406–413, 2014.

Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38(3):199–218, 2000.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019.

Julien NP Martel, David B Lindell, Connor Z Lin, Eric R Chan, Marco Monteiro, and Gordon Wetzstein. Acorn: Adaptive coordinate networks for neural scene representation. *arXiv preprint arXiv:2105.02788*, 2021.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4460–4470, 2019.

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 *European conference on computer vision*, pp. 405–421. Springer, 2020.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022.

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*, pp. 3504–3515, 2020.

Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)*, 32(6):1–11, 2013.

Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5589–5599, 2021.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 165–174, 2019.

Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision*, pp. 523–540. Springer, 2020.

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.

Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14335–14345, 2021.

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

Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pp. 501–518. Springer, 2016.

Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020.

Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11358–11367, 2021.

Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. *arXiv preprint arXiv:2112.03907*, 2021.

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.

Yiqun Wang, Ivan Skorokhodov, and Peter Wonka. Improved surface reconstruction using high-frequency details. *arXiv preprint arXiv:2206.07850*, 2022.

Francis Williams, Matthew Trager, Joan Bruna, and Denis Zorin. Neural splines: Fitting 3d surfaces with infinitely-wide neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9949–9958, 2021.

Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020.

Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34, 2021.

Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *arXiv preprint arXiv:2112.05131*, 2021.

A Appendix

In this appendix, we show more qualitative comparisons for surfaces and images in Fig. 6, Fig. 7, Fig. 8, and Fig. 9.

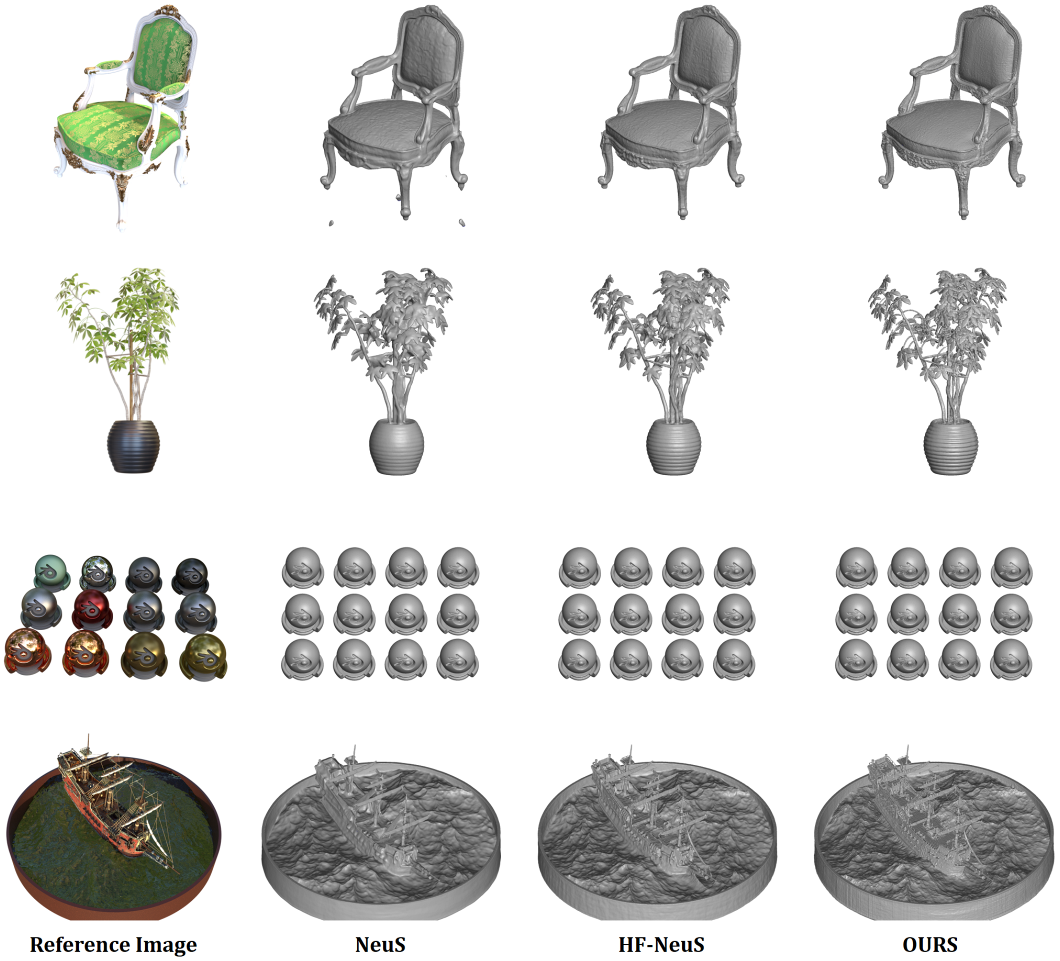

Figure 6: Qualitative evaluation on NeRF synthetic dataset. First column: reference images. Second to the fifth column: VolSDF, NeuS, HF-NeuS, and OURS.

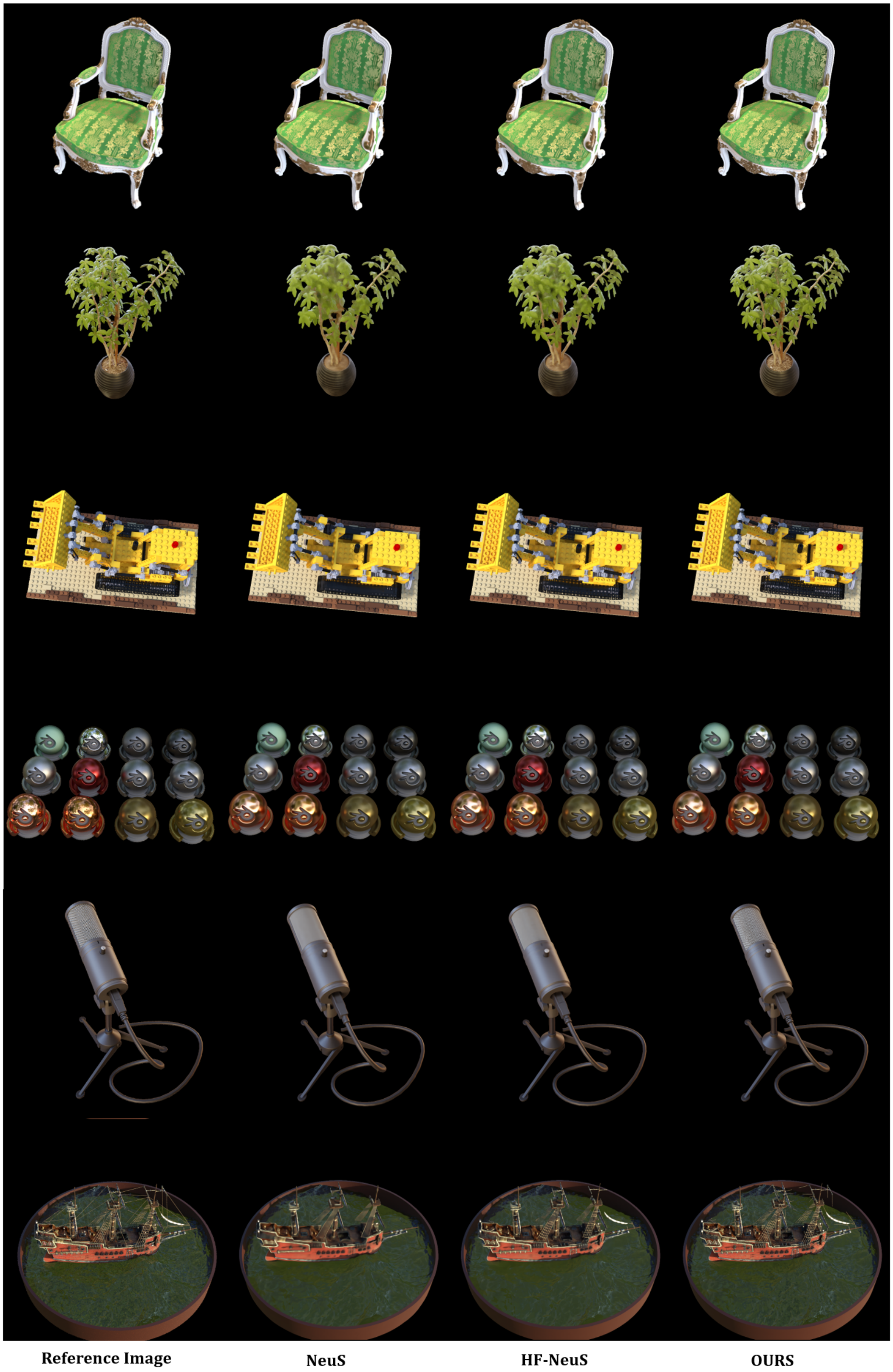

Figure 7: Qualitative evaluation on NeRF synthetic dataset. First column: reference images. Second to the fifth column: the generated images from VolSDF, NeuS, HF-NeuS, and OURS.

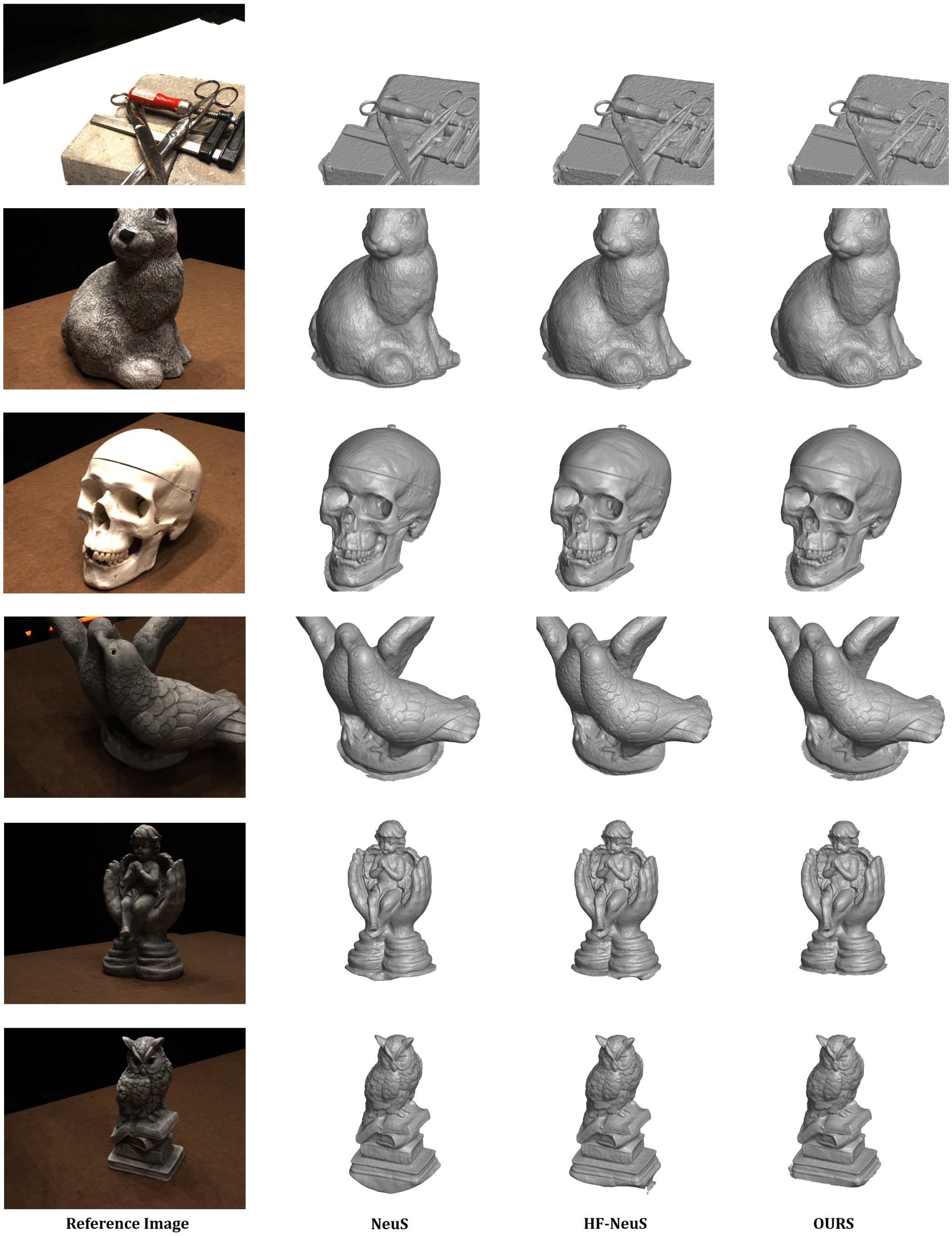

Figure 8: Qualitative evaluation on DTU dataset. First column: reference images. Second to the fifth column: VolSDF, NeuS, HF-NeuS, and OURS.

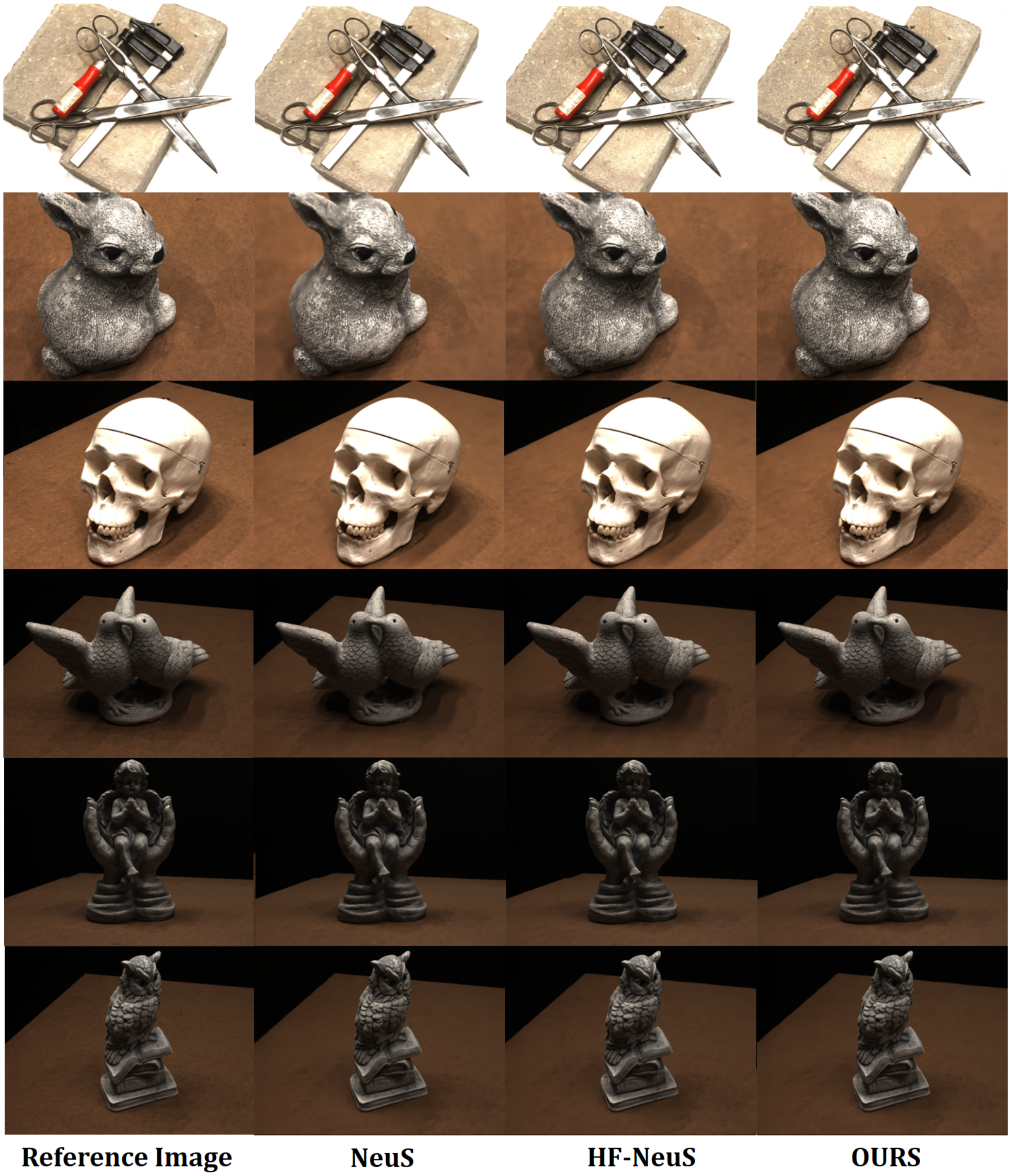

Figure 9: Qualitative evaluation on DTU dataset. First column: reference images. Second to the fifth column: the generated images from VolSDF, NeuS, HF-NeuS, and OURS.