A. Additional implementation details

A.1. More details of the new dataset.

To collect the dataset, we first pick objects with diverse geometries, then render images with dynamic spotlights of different sizes, colors, energy, blend ratios, and locations. Especially three of the scenes contain varying environmental lighting and independently placed spotlights, while the remaining are with light sources moving with the camera. We made this decision according to the observation that a light field usually changes drastically due to the presence of an active light source which is always controlled/carried by an agent (e.g., in rescue scenes), and it can already yield images with significant appearance variations (Fig. S1a).

A.2. Structure of the feature function

The proposed feature function $\mathcal{F}$ can be incorporated into the original neural radiance field architecture in two ways (please refer to the bottom right of Fig. 4). To minimize additional modifications of the original network, one option is to treat $\mathcal{F}$ as a parallel counterpart to the color rendering function c (Fig. 4 A). Then the only modification with the loss function is to add an additional objective term $\mathcal{L}_{vdn}$. Note that we have $\hat{c} = c(\mathbf{p}, \mathbf{v}, \mathbf{n}, \hat{z})$ and $\hat{\psi} = \mathcal{F}(\mathbf{p}, \mathbf{v}, \mathbf{n}, \hat{z})$, where we denote the sampled point as $\mathbf{p}$, viewing direction as $\mathbf{v}$, radiance of point $\mathbf{p}$ as $\hat{c}$, and feature of point $\mathbf{p}$ as $\hat{\psi}$. Following NeuS [41], the byproduct of SDF network, normal $\mathbf{n}$ and global feature $\hat{z}$ are also fed into neural fields c and $\mathcal{F}$.

Another scheme is to treat the predicted $\hat{\psi}$ as a conditional feature to help better predict the radiance field c. That is, to make the color function depend on the feature branch. We first produce the feature by $\hat{\psi} = \mathcal{F}(\mathbf{p}, \mathbf{v}, \mathbf{n}, \hat{z})$, then concatenate it with $\hat{z}$ and feed them into the color rendering branch. In this way, the function of c turns into $\hat{c} = c(\mathbf{p}, \mathbf{v}, \mathbf{n}, \hat{z}, \hat{\psi})$.

A.3. Adaptation for out-of-domain scenes

Given limited diversity of training data for monocular depth estimation, the off-the-shelf network parameters of the distillation network ψ may not be optimal on some out-of-domain scenes. To alleviate this problem, we can first train a network without updating the feature function $\mathcal{F}$ for 20k iterations to learn a coarse scene. Then, the depth map d_k^r for each image I_k is extracted from the SDF network to serve as the target for $\psi(I_k)$ (Eq. (4)). After finetuning ψ for 100 steps, the features $\psi^l(I_k)$ from the distillation network can be utilized for the joint training with view-dependence normalization.

A.4. Warm-up for view-dependence normalization

In practice, the view-dependence normalization loss $\mathcal{L}_{vdn}$ is gradually added into the training after 5000 iterations. Namely, we increase the weight of it from 0 to λ_{vdn} based on the Sigmoid activation. This scheme in our observation can prevent the surfaces from losing details under a large weighting of the depth feature term. In our experiments, we choose λ_{color}, λ_{vdn}, λ_{reg}, and λ_{msk} as 1.0, 1.0 , 0.1 and 0.1 respectively.

A.5. Mask loss details

As mentioned in the main paper, if masks are available, the training is to minimize:

$$\mathcal{L} = \lambda_{color}\mathcal{L}_{color} + \lambda_{vdn}\mathcal{L}_{vdn} + \lambda_{reg}\mathcal{L}_{reg} + \lambda_{msk}\mathcal{L}_{msk} \tag{S1}$$

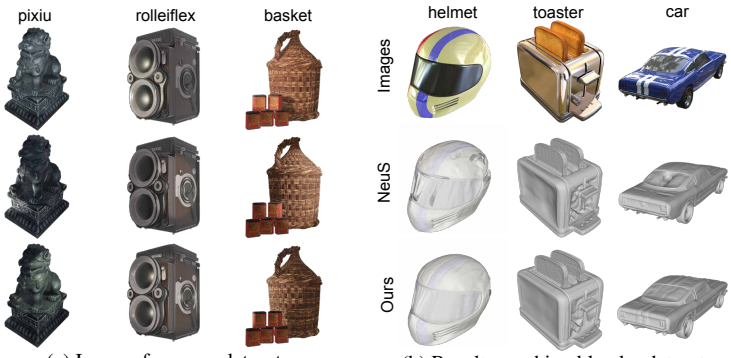
In the case that surface points could be temporally invisible due to occlusion, we simply disable $\mathcal{L}_{msk}$ in training (λ_{msk}=0). For example, in Fig. 8b, rays passing through the camera center to the masked pixels (in gray) may intersect the teeth behind the mouth gag, thus the geometry should not be bounded by the mask loss.

A.6. Camera parameters

Following NeRF- - [43], our optimization is based on the pinhole camera model.

Regarding the intrinsic matrix, it can be expressed as the focal length f and the principal point (cx, cy). We assume all images share the same intrinsic and consider the center of the sensor as the camera principal points, that is, $cx = W/2$, $cy = H/2$, where W, H denote the width and the height of input images. Thus, f is the only parameter of the intrinsic matrix that we need to estimate.

The extrinsic matrix, namely camera poses, can be expressed as a transformation matrix $T = [R|t] \in \mathbb{R}^{3\times4}$ from camera's coordinate system to the world's, where $R \in \mathbb{R}^{3\times3}$ denotes the camera rotation and $t \in \mathbb{R}^3$ denotes the translation. According to Rodrigues' formula, the rotation matrix can be recovered by $R = \mathbf{I} + (1 - cos(\theta))(\mathbf{u}^\wedge)^2 + sin(\theta)\mathbf{u}^\wedge$, where $\mathbf{I}$ is the identity matrix, $\mathbf{u}$, θ are the unit vector and module length of rotation vector $\phi \in \mathbb{R}^3$, and $(\cdot)^\wedge$ is the skew operator that converts a vector to a skew matrix. Thus we can optimise the extrinsic matrices with trainable parameters ϕ and t.



(a) Images from our dataset. (b) Results on shiny blender dataset.

Figure S1

		random loc. & env. lighting			glossy obj. (shiny blender)		
	Method	basket	pixiu	rolleiflex	car	helmet	toaster
IoU ↑	NeuS	0.706	0.494	0.440	0.650	0.362	0.373
	Ours	0.736	0.693	0.442	0.659	0.475	0.454
f-score ↑	NeuS	0.793	0.666	0.691	0.778	0.518	0.472
	Ours	0.821	0.871	0.721	0.839	0.645	0.590
L2CD ↓	NeuS	0.241	4.311	3.014	0.441	12.305	4.827
	Ours	0.230	0.390	1.879	0.309	4.266	4.370

Table S1. Quantitative results under independently moving light source (left) and of glossy objects from shiny blender (right).

B. Additional results

B.1. Results on general lighting conditions and glossy objects

We report the quantitative comparison on the three scenes with independent spotlights and varying ambient light in Tab. S1 (left). The improvement of our method is still valid across all metrics. Also, please note that the NeROIC dataset we employed also comes with globally changing illumination Fig. 5.

Tab. S1 (right) shows quantitative results on three objects of synthetic shiny blender dataset [40], and the corresponding qualitative results can be found in Fig. S1. As seen, our method can alleviate the ambiguity caused by glossy surfaces significantly.

B.2. Effect of the two variants for the feature branch

We carry out experiments with the two realization schemes of the feature function $\mathcal{F}$ and show the results in Fig. S2. Both of the two designs (Ours A and Ours B) have positive effects over vanilla NeuS. For convenience, we choose scheme A as default, which we use to report the setting and results in the main paper.

B.3. Ablation on out-of-domain adaptation

In Fig. S3, we perform an additional ablation on the effect of adaptation for the distillation network. As observed, in some scenes the features without adaptation (w/o ada.) may fail to help recover better details (lego/ mic, orange boxes), e.g., erroneous concave-convex relation (lego, blue boxes). Sometimes, the surface reconstructed for the teeth can be even worse if the adaptation is not performed. As designed, when it comes to out-of-domain scenes, the additional steps of adaptation can effectively close the domain gap and thus ensure that our view-dependence normalization can robustly improve the geometry.

B.4. Results on the DTU dataset

We also report the results on the DTU dataset [20]. The reconstructions are measured with Chamfer Distance using DTU evaluation code and compared with several existing works (in Tab. S2). Our method (developed from NeuS) achieves much better quality than NeuS as well as other methods. We visualize these surfaces in Fig. S4. Both qualitative and quantitative comparisons prove that our method leads to a significant improvement. In conclusion, the proposed method works well in general scenarios, either with dynamic (Tab. 1) or static light fields.

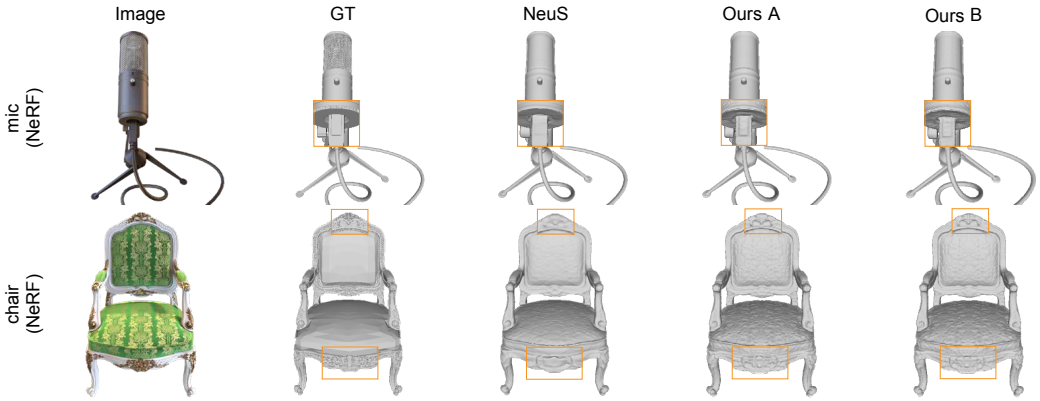


Figure S2. Comparison between different realizations of the feature function. Reconstructions from the two variants (Ours A and Ours B) have more accurate details compared to the baseline NeuS, and the two variants have similar results.

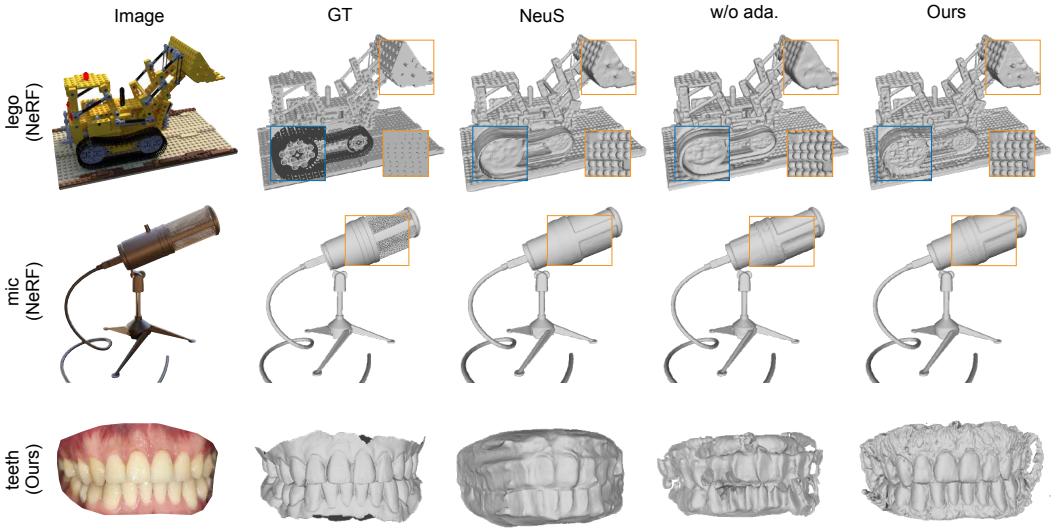


Figure S3. Ablation study on the adaptation scheme. Training data of the first two rows are from the synthetic dataset of NeRF [30], and the last row is from our intra-oral scan. Without adaption (w/o ada.), the model may fail to estimate an accurate geometry or it could lead to the lack of details. With the adaptive finetuning from a coarse scene geometry, the distillation network provides features more beneficial to the task of geometric reconstruction.

SCAN	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122	mean
COLMAP [38]	0.45	0.91	0.37	0.37	0.9	1	0.54	1.22	1.08	0.64	0.48	0.59	0.32	0.45	0.43	0.65
IDR [50]	1.63	1.87	0.63	0.48	1.04	0.79	0.77	1.33	1.16	0.76	0.67	0.9	0.42	0.51	0.53	0.9
VolSDF [49]	1.14	1.26	0.81	0.49	1.25	0.7	0.72	1.29	1.18	0.7	0.66	1.08	0.42	0.61	0.55	0.86
NeuS [41]	1.37	1.21	0.73	0.4	1.2	0.7	0.72	1.01	1.16	0.82	0.66	1.69	0.39	0.49	0.51	0.87
NeuralWarp [9]	0.49	0.71	0.38	0.38	0.79	0.81	0.82	1.2	1.06	0.68	0.66	0.74	0.41	0.63	0.51	0.68
GeoNeuS [13]	**0.38**	**0.54**	**0.34**	0.36	0.80	**0.45**	**0.41**	1.03	**0.84**	**0.55**	**0.46**	**0.47**	**0.29**	**0.36**	**0.35**	**0.51**
Ours	0.44	0.72	0.53	**0.34**	**0.65**	0.55	0.54	**0.85**	0.86	0.81	0.49	0.77	0.30	0.41	0.38	0.58

Table S2. Quantitative results on the DTU dataset. All numbers of the baseline methods are from their original papers, except that the IDR and COLMAP results are from GeoNeus [13]. The best results are marked in bold, and the second underlined. When applied to NeuS, our view-dependence normalization significantly improves over the original NeuS results.

Method	pixiu	airforce	starwars1	rolleiflex	boat	basket	ZIL	mechanical
NeRF	0.530	0.482	0.564	0.263	0.756	0.615	0.557	0.644
NeROIC	0.533	0.539	0.603	0.280	0.706	0.598	0.641	0.645
NeRF-W	0.811	0.863	0.461	0.563	0.718	0.599	0.661	0.664
NeuS	0.918	0.905	0.764	0.752	0.723	0.834	0.549	0.654
GeoNeuS	0.432	0.814	0.396	0.662	0.723	0.748	0.636	0.604
NeuS+$\mathcal{F}$	**0.937**	**0.927**	**0.972**	**0.825**	**0.847**	**0.876**	**0.735**	**0.709**

Table S3. Per-scene f-score on our new dataset.

Method	pixiu	airforce	starwars1	rolleiflex	boat	basket	ZIL	mechanical
NeRF	0.3514	0.387	0.522	0.238	0.687	0.513	0.505	0.643
NeROIC	0.373	0.421	0.580	0.263	0.669	0.513	0.603	0.587
NeRF-W	0.345	0.483	0.566	0.259	0.703	0.497	0.632	**0.664**
NeuS	**0.881**	0.770	0.607	0.483	0.649	0.707	0.443	0.572
GeoNeuS	0.456	0.672	0.407	0.424	0.601	0.556	0.529	0.495
NeuS+$\mathcal{F}$	0.864	**0.772**	**0.736**	**0.515**	**0.751**	**0.755**	**0.628**	0.641

Table S4. Per-scene IoU on our new dataset.

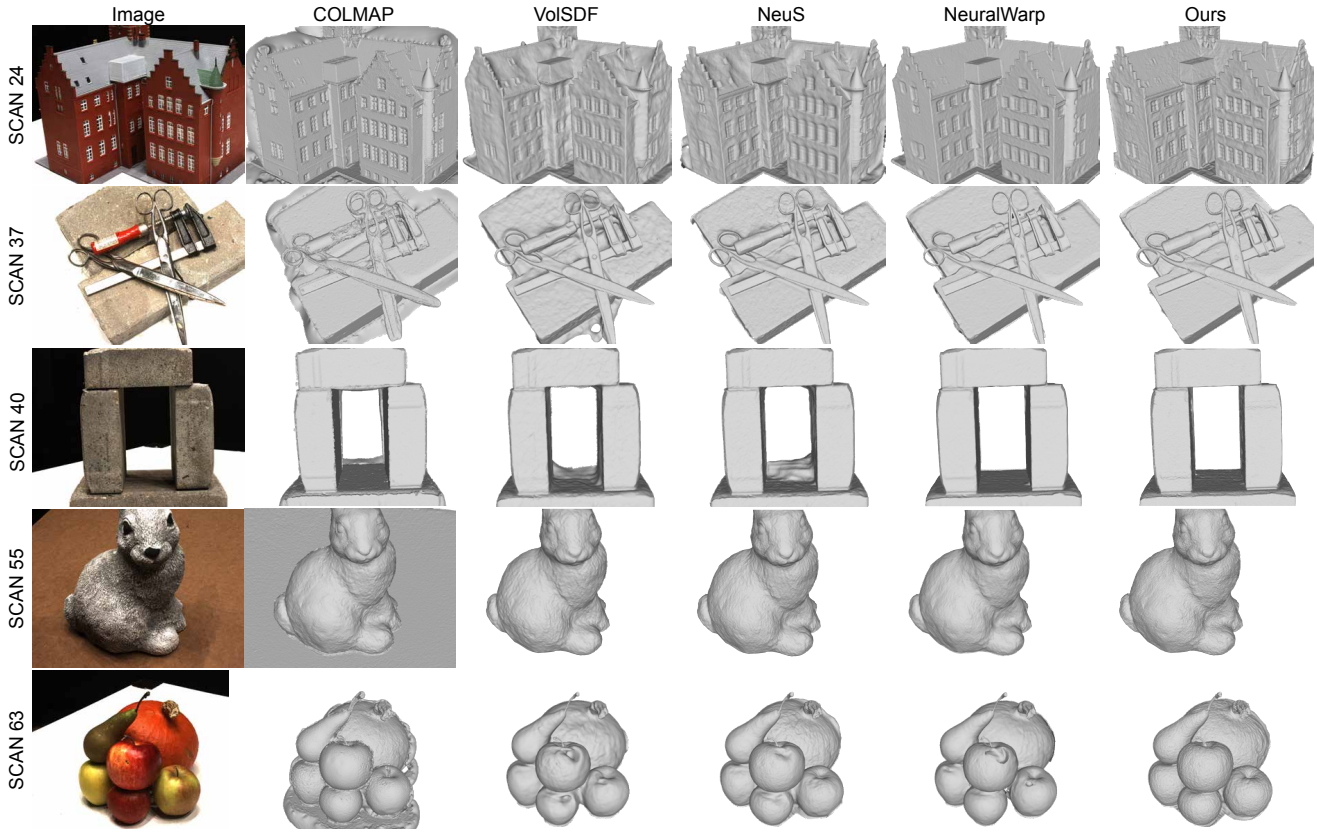


Figure S4. Reconstruction on the DTU dataset (1/2).

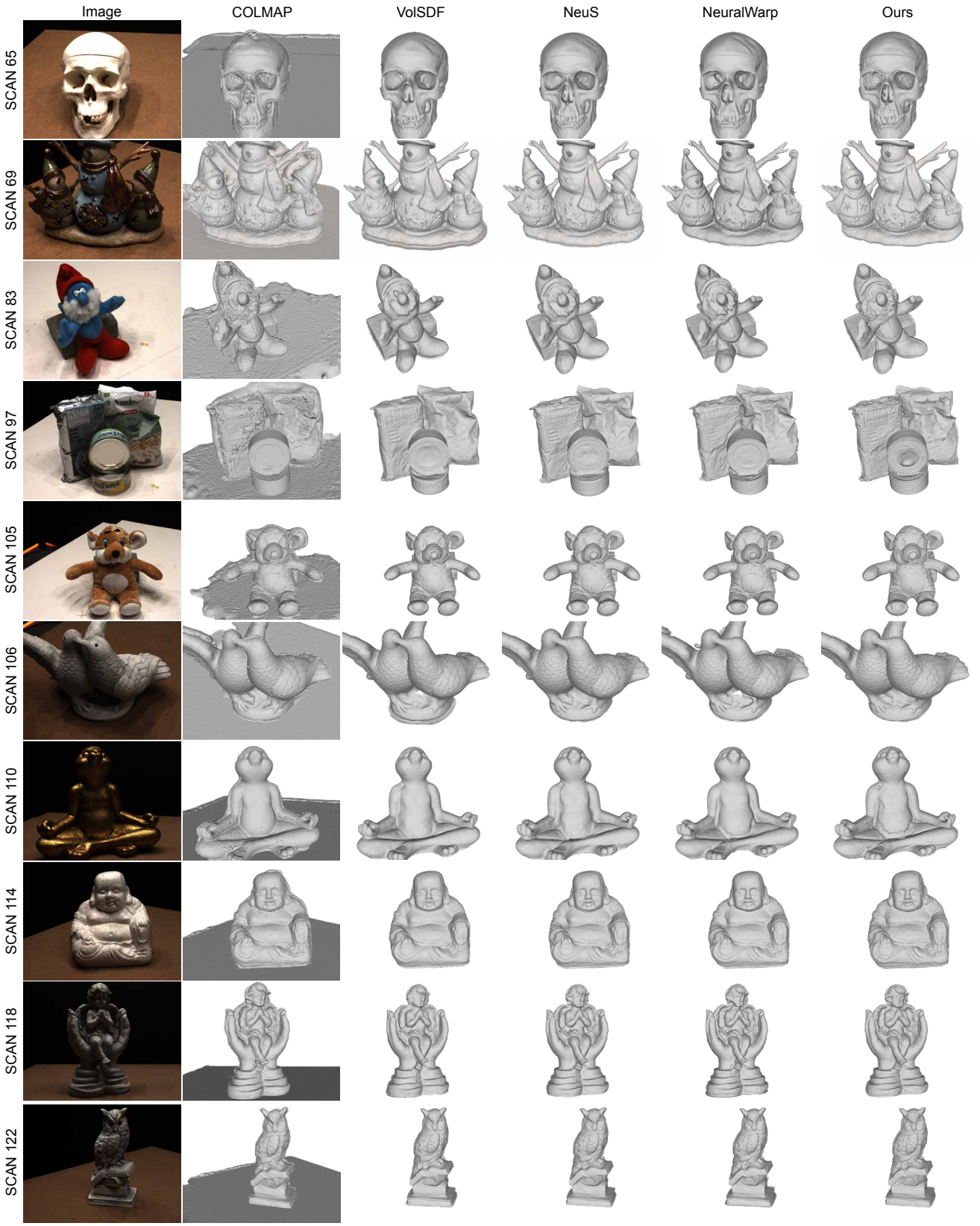


Figure S4. Reconstruction on the DTU dataset (2/2).