Ref-NeuS: Ambiguity-Reduced Neural Implicit Surface Learning for Multi-View Reconstruction with Reflection Supplemental Material

Wenhang Ge[1,2] Tao Hu[4] Haoyu Zhao[1] Shu Liu[5] Ying-Cong Chen[1,2,3,*]

[1]HKUST(GZ) [2]HKUST(GZ)-SmartMore Joint Lab
[3]HKUST [4]CUHK [5]SmartMore

1. Optimization and Additional Model Details

Optimization Details. We used Adam [2] as our optimizer. For the first 5,000 iterations, the learning rate was linearly increased from 0 to 5×10^{-4} using a warm-up strategy. After that, we controlled it using the cosine decay schedule to the minimum learning rate of 2.5×10^{-5}. We trained each model for 200,000 iterations, which took a total of 7 hours on a single NVIDIA RTX3090Ti GPU. For the novel view synthesis task, we trained each model for 1,000,000 iterations over 80 hours using a smaller batch size with fewer sampled rays on a single NVIDIA RTX3090Ti GPU. To ensure consistency with the reconstruction baselines, we used single-image batching with 512 sampled rays for all reconstruction tasks. For novel view synthesis, we used single-image batching with 1024 sampled rays, limited to the GPU memory, instead of 4096×4 as used in Ref-NeRF. On each ray, we sampled 64 coarse points, 64 fine points, and 32 points to model the background, as in NeRF++ [9].

Network architecture. Our network architecture is similar to NeuS [7], comprising of a geometry network and a radiance network to encode SDF and view-dependent radiance, respectively. The geometry network parametrizes the signed distance function and consists of 8 hidden layers with a hidden size of 256. Instead of ReLU, we used Softplus with β = 100 for all hidden layers. We used a skip connection [3] to connect the input with the output of the fourth layer. The geometry network takes the spatial position $\mathbf{x}$ of points as input and outputs the signed distances to the object. In addition, the geometry network produces a geometry feature with dimension 256, which is further used as input to the radiance network to acquire view-dependent radiance. The radiance network comprises 4 hidden layers of size 256, which parametrize view-dependent radiance. It takes as input the spatial position $\mathbf{x}$, the normal vector $\hat{\mathbf{n}}$, the reflection direction $\boldsymbol{\omega}_r$, and the 256-dimensional geometry feature vector. We applied positional encoding with 6 frequencies to the spatial location $\mathbf{x}$ and 4 frequencies to the view direction $\boldsymbol{\omega}_r$.

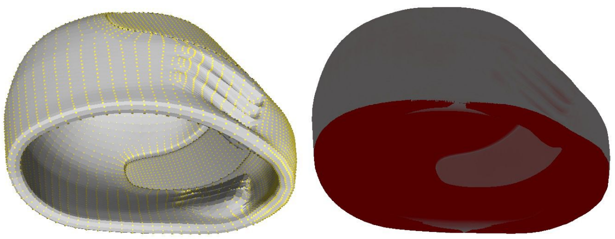


Figure 1. In the ground truth meshes (left) of ShinyBlender, there are two layers. However, on multi-view images, the inner layer is invisible. This means the completeness is biased, as most of the points in the inner layer contribute to meaningless error (right). Points with large errors are marked in red.

2. Evaluation Details

Meshes. For the ShinyBlender [6] and Blender [3] datasets, the ground truth meshes were exported from Blender files. Due to the original models' small scales with a radius around 1, we exported them with a scale factor of 150. For the fish from SLF [8] and the cans/corncho1 from Bag of Chips [4], we increased the meshes' sizes by 100 and 1000 times, respectively, resulting in similar scales for all ground truth meshes. During training, we normalized the object to a unit sphere. During inference, we transferred the meshes to the original space to compute the Chamfer Distance.

Since the original meshes contain too few points, we upsampled the points in each triangle to obtain dense point clouds for evaluation. Finally, the Chamfer Distance was computed by

$$d(S_1, S_2) = \frac{1}{S_1}\sum_{x \in S_1} \min_{y \in S_2} ||x-y||_2^2 + \frac{1}{S_2}\sum_{y \in S_2} \min_{x \in S_1} ||y-x||_2^2, \tag{1}$$

[*] Corresponding author.

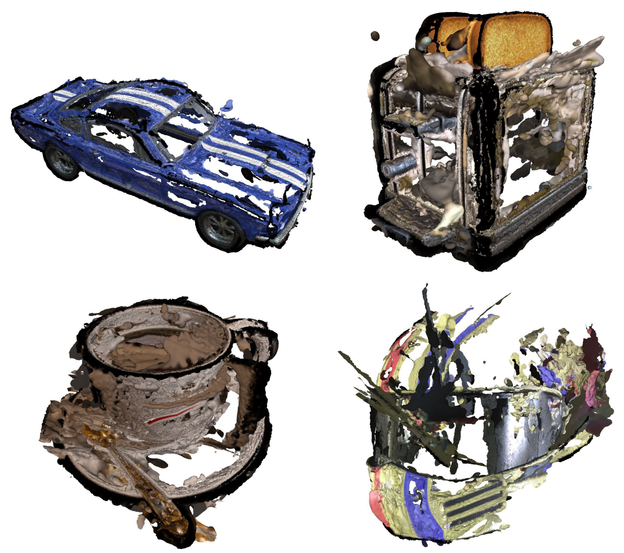


Figure 2. Visualization of the reconstruction results by COLMAP on the ShinyBlender dataset. COLMAP failed to reconstruct objects with reflective surfaces, as the multi-view consistency was not plausible.

where the first term is used to test *accuracy*, and the second term validates *completeness* [1]. S_1 and S_2 are the recovered point clouds upsampled from meshes and ground truth dense point clouds, respectively. For the ShinyBlender dataset, shown in Fig. 1, the ground truth dense point clouds include two layers. However, the inner layer is invisible on multi-view images and cannot be reconstructed, resulting in a biased completeness, so only accuracy is reported.

Surface Normals. To compute the surface normal for a pixel p, we compute the normals of sampled points along the ray $\mathbf{r}$ derived from the SDF as follows:

$$\hat{\mathbf{n}}_\mathbf{i} = \frac{\nabla f(\mathbf{x_i})}{||\nabla f(\mathbf{x_i})||}. \tag{2}$$

Then, the volume rendering procedure is performed to aggregate these normals, forming a single surface normal:

$$\hat{\mathbf{n}}(\mathbf{r}) = \sum_{i=1}^{N} T_i \alpha_i \hat{\mathbf{n}}_i. \tag{3}$$

We used the normalized normals $\overline{\mathbf{n}}(\mathbf{r}) = \frac{\hat{\mathbf{n}}(\mathbf{r})}{||\hat{\mathbf{n}}(\mathbf{r})||}$ for evaluating MAE for all pixels.

3. Results of using training views

In the main text, we used original 200 test views for reconstruction training, here we show the results of using 100 training views for reconstruction training in Table 1. The conclusion is similar to that obtained using the test views.

4. Detailed Results of Ablation Study

We reported the quantitative metrics (accuracy and MAE) for each scene of ShinyBlender in Table 2.

5. Additional Results on diffuse materials

We also carried out experiments on non-reflective objects to show that reflection score will not cause performance degradation of non-reflective objects, where DTU scenes (i.e., scan55, scan83, scan105, scan106, scan114, scan118) were used. The results are reported in Table 3.

6. Additional Results on scenes with both diffuse and shiny materials

Previous reconstructed scenes are either shiny or diffuse materials. To show the robustness of our method, we further diverse the scenes with both diffuse and shiny materials. Given that such scenes are uncommon in existing datasets, besides materials for Blender dataset, we further employed Blender to render multi-view images that combine helmet for Shinyblender and hotdog from Blender. The comparison is presented in Fig. 9. Our method can reconstruct the shiny objects better, while do not lead to performance drop on diffuse materials.

7. Additional Visualizations

Visualizations of COLMAP. We visualized the reconstruction results of COLMAP [5], an MVS-based method on the Shiny Blender dataset, as shown in Fig. 2. COLMAP fails to recover reflective surfaces, indicating that the multi-view consistency is not reasonable in reflective scenes, leading to severe missing parts and artifacts.

Visualizations of ablation study. We visualized how the reflection-aware photometric loss and reflection direction-dependent radiance improve surface quality in Fig. 3. Due to the reflective surfaces, the toaster is extremely challenging even for human perception. It is challenging to distinguish where the real surface lies. NeuS reconstructs the toaster with severe missing parts due to reflection. "Ref-Neus w/o Ref" reconstructs the surface with fewer missing parts, indicating that reflection-aware photometric loss can localize the reflective surfaces and alleviate the ambiguity. Our full model, Ref-NeuS, achieves better reconstruction results without missing parts.

Visualizations of Ref-NeuS. We present additional visualization results of different objects in Fig. 4, Fig. 5, Fig. 6, and Fig. 7 to demonstrate the effectiveness of our Ref-NeuS. Our Ref-NeuS achieves better reconstruction quality compared to NeuS.

Visualizations of results on Hulk. The real-world objects used in our experiments were captured under strict

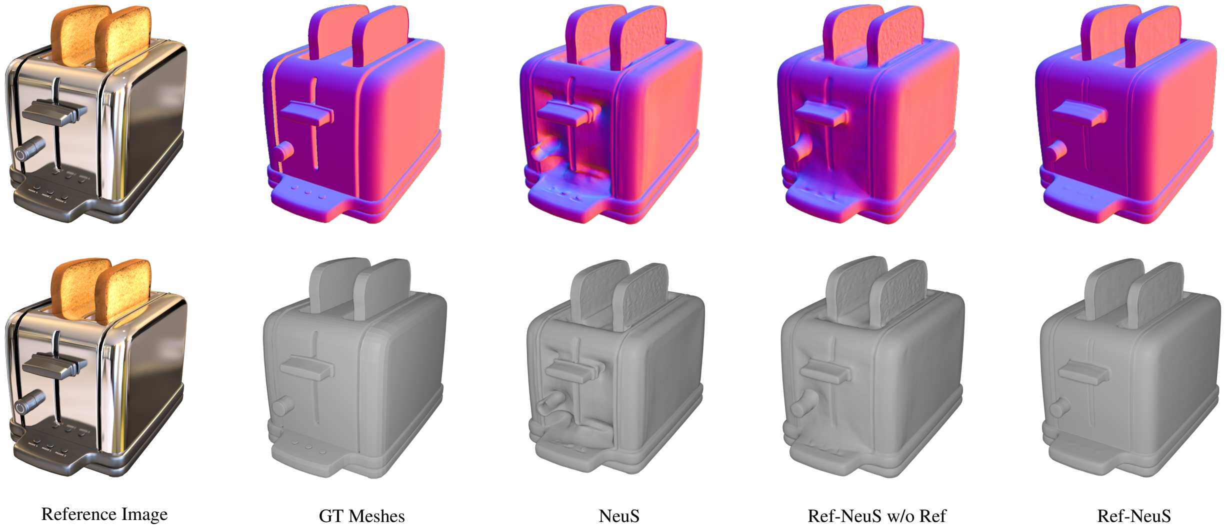


Figure 3. Surface geometry and surface normals of ablation models on toaster of ShinyBlender.

Methods	helmet		toaster		car	
	Acc↓	MAE↓	Acc↓	MAE↓	Acc↓	MAE↓
Neus	0.92	0.88	3.34	2.73	0.72	1.08
Ref-NeuS	0.33	0.39	0.45	1.56	0.36	0.77

Table 1. Results of using training views for reconstruction training.

conditions in a lab-controlled environment. Differently, we captured the Hulk with glossy surfaces using an iPad in a natural environment, capturing both the object and its surroundings with lighting illumination and ambient light. As we captured the object with the iPad moving around it, the light source may have been occluded, resulting in shadows on the surfaces. We show the results in Fig. 8, we can still achieve better performance than NeuS.

8. Running Time

We demonstrate that estimating the reflection score will not significantly increase the running time. There are two main steps that contribute to the increase in running time. The first step involves the intermediate meshes. Since we extract intermediate meshes with a resolution of 128, it only takes approximately 0.35 seconds for each mesh extraction. The second step involves projection and distance computation for visibility identification. To obtain pixel colors, we project the predicted surface point onto visible source images. This step does not incur notable extra computational cost, with only 0.012 seconds per step. In Table 4, we present the total running time.

9. Failure Case

Figure 10 displays the reconstruction results of the coffee object using ShinyBlender, which is a failure case of our method. This object contains water surfaces that possess different reflection coefficients compared to solid objects. Merely substituting the dependency of the radiance network with reflection direction, without considering the object material, can result in artifacts. This motivates future work on how to better model view-dependent radiance while taking the material into consideration. However, incorporating reflection-aware photometric loss can still improves the reconstruction quality over NeuS. We present the results of “Ref-NeuS w/o Ref” for Ref-NeuS on the coffee object of ShinyBlender.

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision (IJCV)*, 2016. 2

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[3] 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. *Communications of the ACM*, 2021. 1

[4] Jeong Joon Park, Aleksander Holynski, and Steven M Seitz. Seeing the world in a bag of chips. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

Method	RS	Vis	Ref	helmet		toaster		coffee		car		mean	
				Acc	MAE	Acc	MAE	Acc	MAE	Acc	MAE	Acc	MAE
NeuS				1.33	1.12	3.26	2.87	1.42	1.99	0.73	1.10	1.69	1.77
NeuS w/ RS	✓			0.75	0.85	2.14	2.23	1.11	1.58	0.66	1.05	1.17	1.43
NeuS w/ Ref			✓	0.41	0.69	0.59	1.59	3.87	2.74	0.55	0.97	1.36	1.50
Ref-NeuS w/o Ref	✓	✓		0.43	0.71	1.43	2.12	0.77	0.99	0.58	1.00	0.80	1.21
Ours (full)	✓	✓	✓	**0.29**	**0.38**	**0.42**	**1.47**	**0.77**	**0.99**	**0.37**	**0.80**	**0.46**	**0.91**

Table 2. Detailed quantitative metrics of ablation study on Shiny Blender dataset.

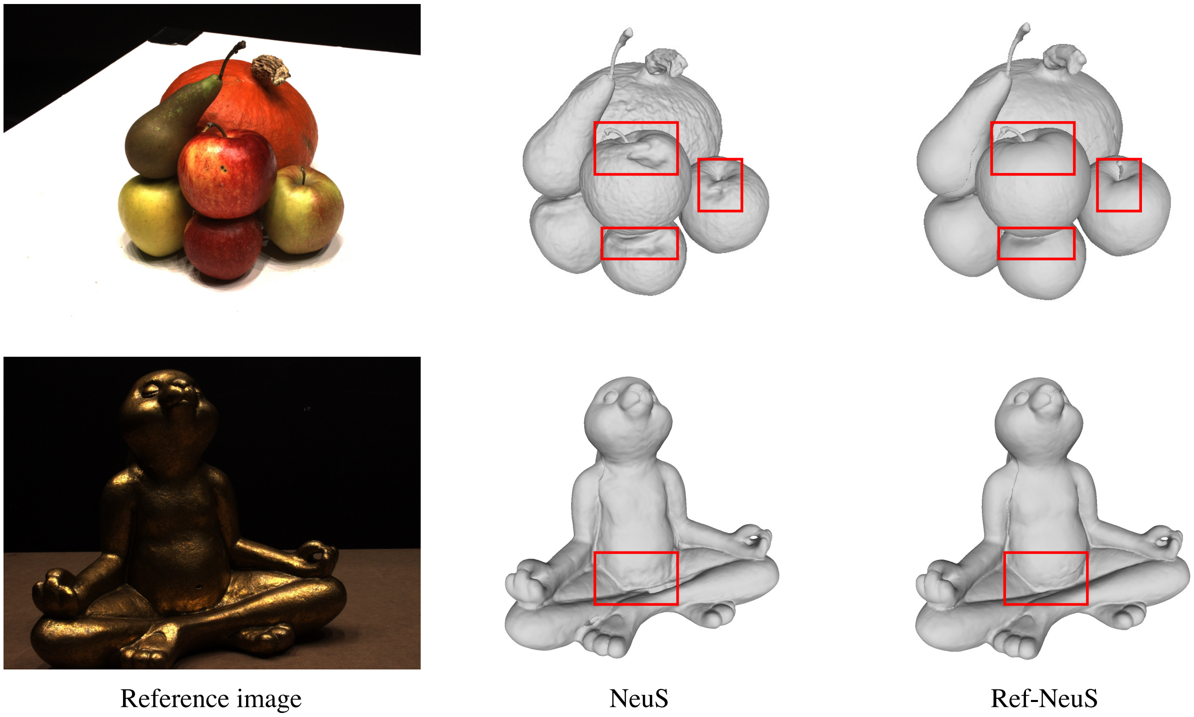


Figure 4. Qualitative comparison with NeuS on scan63 and scan 110 of DTU dataset.

scene	55	83	105	106	114	118
NeuS	0.37	1.45	0.78	0.52	0.36	0.45
NeuS w/ RS	0.36	1.27	0.72	0.51	0.36	0.46

Table 3. The results on non-reflective objects.

setting	Running time [h]
NeuS	7
Ref-NeuS	7.5

Table 4. Comparison of running time.

[5] 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 (ECCV)*, 2016. 2

[6] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[7] 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. *arXiv preprint arXiv:2106.10689*, 2021. 1

[8] Daniel N Wood, Daniel I Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H Salesin, and Werner Stuetzle. Surface light fields for 3d photography. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000. 1

[9] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 1

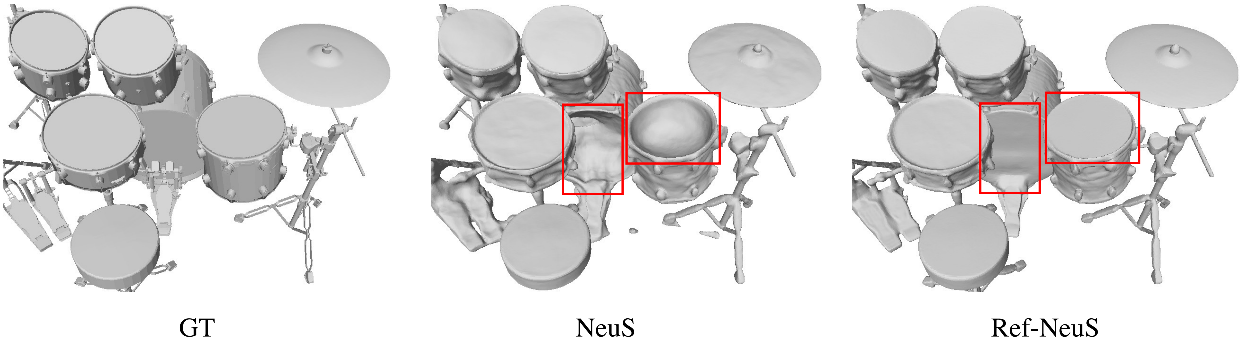


Figure 5. Qualitative comparison with NeuS on drums of Blender dataset.

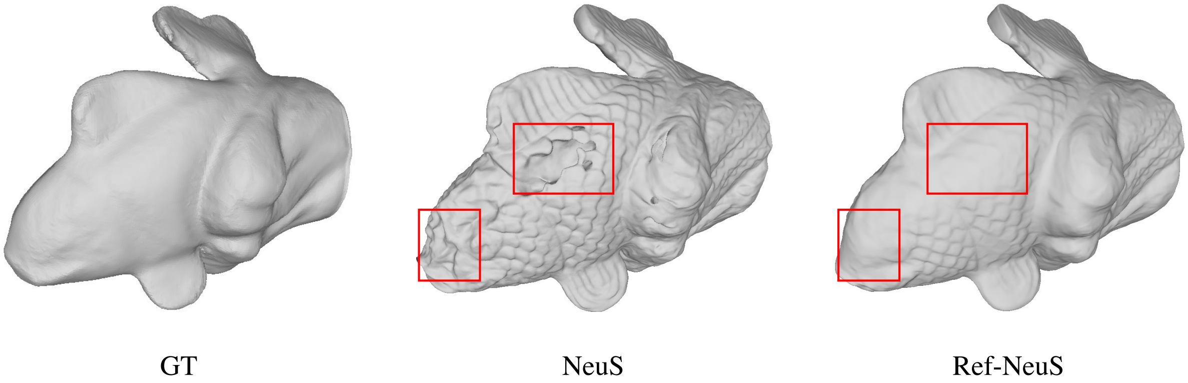


Figure 6. Qualitative comparison with NeuS on fish of SLF dataset.

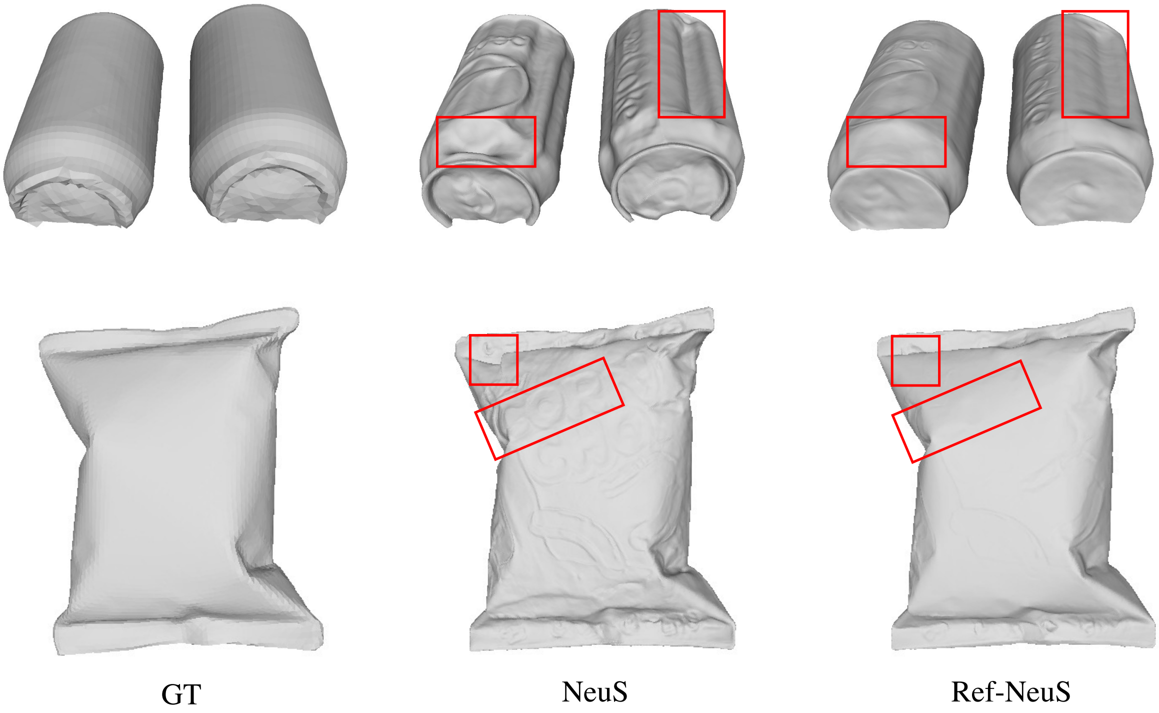


Figure 7. Qualitative comparison with NeuS on cans and corncho1 of Bag of Chips dataset.

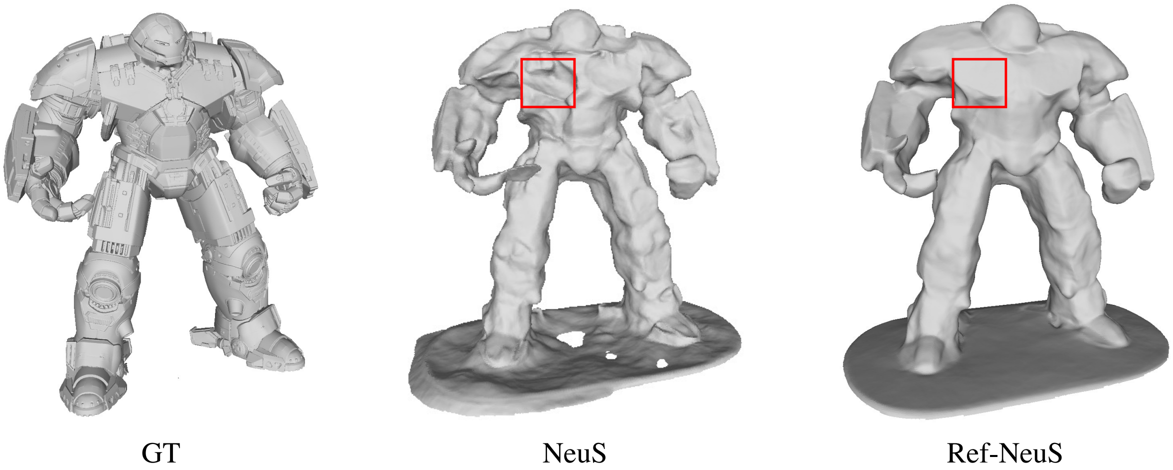


Figure 8. Qualitative comparison with NeuS on Hulk.

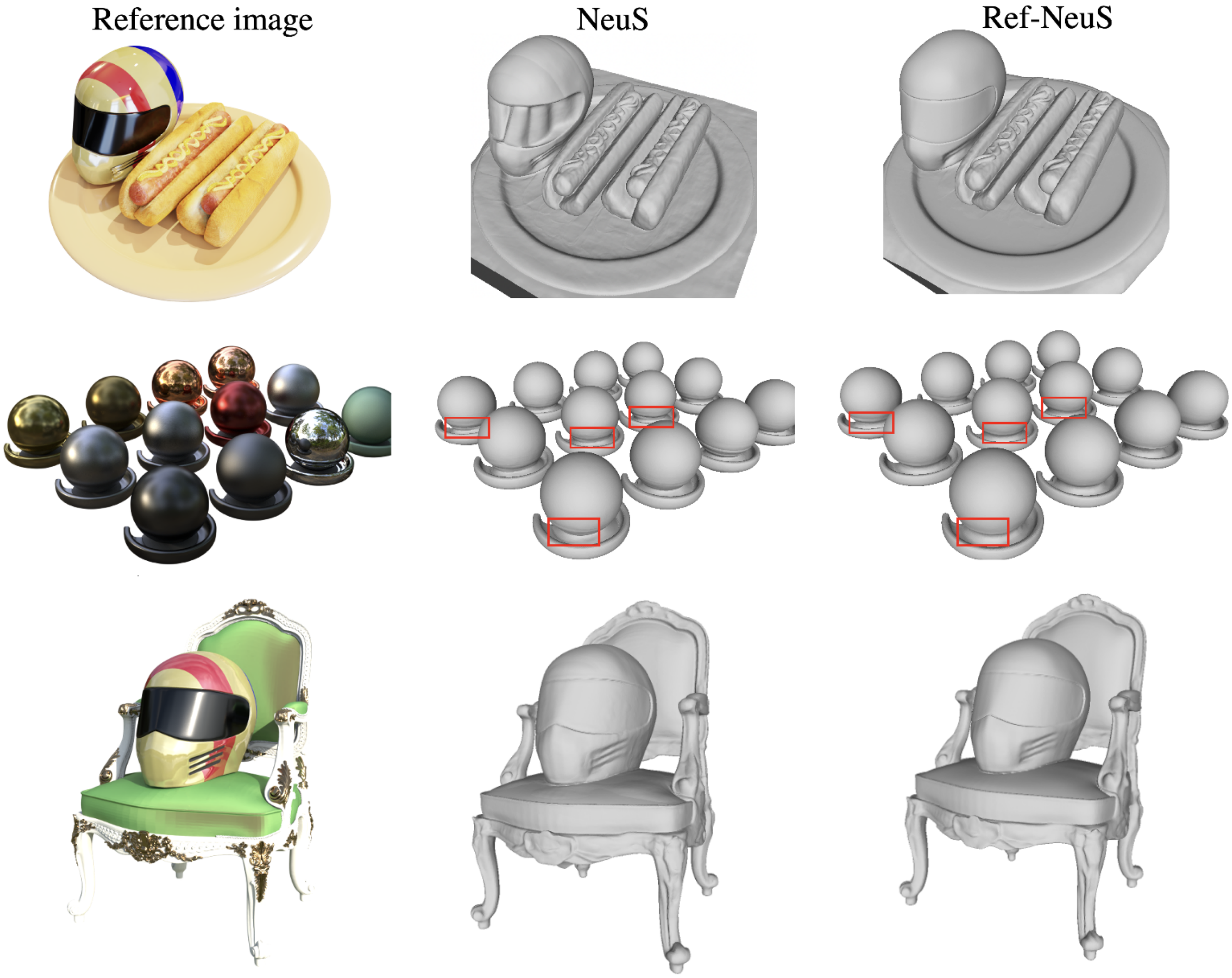
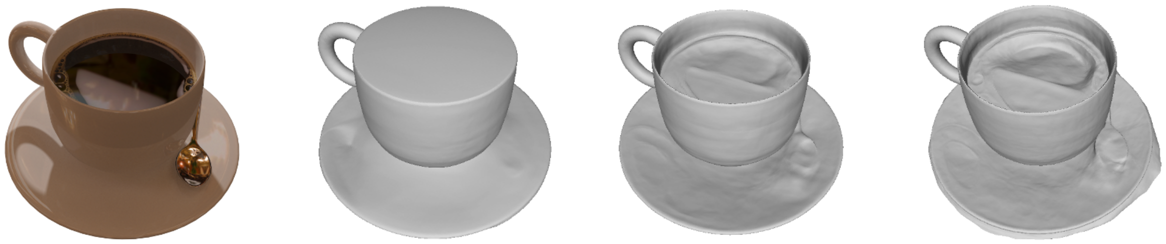


Figure 9. Comparison on scenes with both diffuse and shiny materials.



Reference image

Ref-NeuS

Ref-NeuS w/o Ref

NeuS

Figure 10. A failure case on coffee of ShinyBlender.