## A. Preliminaries

**Volume rendering**. The radiance $C$ of the pixel corresponding to a given ray $\boldsymbol{r}(t) = \boldsymbol{o} + t\boldsymbol{d}$ at the origin $\boldsymbol{o} \in \mathbb{R}^3$ towards direction $\boldsymbol{d} \in \mathbb{S}^2$ is calculated using the volume rendering equation, which involves an integral along the ray with boundaries $t_n$ and $t_f$ ($t_n$ and $t_f$ are parameters to define the near and far clipping plane). This calculation requires the knowledge of the volume density $\sigma$ and directional color $\boldsymbol{c}$ for each point within the volume.

$$C(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\sigma(\boldsymbol{r}(t))\mathbf{c}(\boldsymbol{r}(t), \boldsymbol{d})dt \tag{14}$$

The volume density $\sigma$ is used to calculate the accumulated transmittance $T(t)$:

$$T(t) = \exp\left(-\int_{t_n}^{t_f} \sigma(\boldsymbol{r}_s)ds\right) \tag{15}$$

It is then used to compute a weighting function $w(t) = T(t)\sigma(\boldsymbol{r}(t))$ to weigh the sampled colors along the ray $\boldsymbol{r}(t)$ to integrate into radiance $C(\boldsymbol{r})$.

**Surface rendering**. The radiance $L_o(\boldsymbol{x}, \boldsymbol{\omega}_o)$ reflected from a surface point $\boldsymbol{x}$ in direction $\boldsymbol{\omega}_o = -\boldsymbol{d}$ is an integral of bidirectional reflectance distribution function (BRDF) and illumination over half sphere $\Omega$, centered at normal $\boldsymbol{n}$ of the surface point $\boldsymbol{x}$:

$$L_o(\boldsymbol{x}, \boldsymbol{\omega}_o) = \int_{\Omega} L_i(\boldsymbol{x}, \boldsymbol{\omega}_i) f_r(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o)(\boldsymbol{\omega}_i \cdot \boldsymbol{n})\mathrm{d}\boldsymbol{\omega}_i \tag{16}$$

where $L_i(\boldsymbol{x}, \boldsymbol{\omega}_i)$ is the illumination on $\boldsymbol{x}$ from the incoming light direction $\boldsymbol{\omega}_i$, and $f_r$ is BRDF, which is the proportion of light reflected from direction $\boldsymbol{\omega}_i$ towards direction $\boldsymbol{\omega}_o$ at the point $\boldsymbol{x}$.

## B. Implementation details

Our full model is composed of several MLP networks, each one of them having a width of 256 hidden units unless otherwise stated. In Stage 1, the SDF network $\mathsf{S}_\theta$ is composed of 8 layers and includes a skip connection at the 4-th layer, similar to NeuS [42]. The input 3D coordinate $\boldsymbol{x}$ is encoded using positional encoding with 6 frequency scales. The diffuse color network $\mathsf{M}_d$ utilizes a 4-layer MLP, while the input surface normal $\boldsymbol{n}$ is positional-encoded using 4 scales. For the specular color network $\mathsf{M}_s$, a 4-layer MLP is employed, and the reflection direction $\boldsymbol{\omega}_r$ is also positional-encoded using 4 frequency scales. In the first stage, we exclusively focus on decomposing the highlight (largely white) areas. To reduce the complexity of considering color, we assume that the specular radiance is in grayscale and only consider changes in brightness. We can incorporate color information in later stages to obtain a more detailed specular reflection model. Like NeuS, the background is modeled by NeRF++.

In Stage 2, the light visibility network $\mathsf{M}_\nu$ has 4 layers. To better encode the input 3D coordinate $\boldsymbol{x}$, positional encoding with 10 frequency scales is utilized. The input view direction $\omega_i$ is also positional-encoded using 4 scales. The indirect light network $\mathsf{M}_{\text{ind}}$ in stage 2 comprises 4 layers.

In stage 3, the encoder part of the BRDF network consists of 4 layers, and the input 3D coordinate is positional-encoded using 10 scales. The output latent vector $\boldsymbol{z}$ has 32 dimensions, and we impose a sparsity constraint on the latent code $\boldsymbol{z}$, following IndiSG [56]. The decoder part of the BRDF network is a 2-layer MLP with a width of 128, and the output has 4 dimensions, including the diffuse albedo $\boldsymbol{d}_a \in \mathbb{R}^3$ and roughness $r \in \mathbb{R}$. Finally, the specular albedo network $\mathsf{M}_{sa}$ uses a 4-layer MLP, where the input 3D coordinate $\boldsymbol{x}$ is positional-encoded using 10 scales, and the input reflection direction $\boldsymbol{\omega}_r$ is positional-encoded using 4 scales.

The learning rate for all three stages begins with a linear warm-up from 0 to $5 \times 10^{-4}$ during the first 5K iterations. It is controlled by the cosine decay schedule until it reaches the minimum learning rate of $2.5 \times 10^{-5}$, which is similar to NeuS. The weights $\lambda_{\text{sur}}$ for the surface color loss are set for 0.1, 0.6, 0.6, 0.6 and 0.01 for DTU, SK3D, Shiny, Glossy, and the IndiSG dataset, respectively. For all datasets, the Fresnel value f in the rendering equation is set to 0.02. We train our model for 300K iterations in the first stage, which takes 11 hours in total. For the second and third stages, we train for 40K iterations, taking around 1 hour each. The training was performed on a single NVIDIA RTX 4090 GPU.

## C. Training strategies of stage 1

In our training process, we define three loss functions, namely volume radiance loss $\mathcal{L}_{\text{vol}}$, surface radiance loss $\mathcal{L}_{\text{sur}}$, and regularization loss $\mathcal{L}_{\text{reg}}$. The volume radiance loss $\mathcal{L}_{\text{vol}}$ is measured by calculating the $\mathcal{L}_1$ distance between the ground truth colors $C^{\text{gt}}$ and the volume radiances $C^{\text{vol}}$ of a subset of rays $\mathcal{R}$, which is defined as follows.

$$\mathcal{L}_{\text{vol}} = \frac{1}{|\mathcal{R}|}\sum_{\boldsymbol{r}\in\mathcal{R}} \|C^{\text{vol}}_{\boldsymbol{r}} - C^{\text{gt}}_{\boldsymbol{r}}\|_1 \tag{17}$$

The surface radiance loss $\mathcal{L}_{\text{sur}}$ is measured by calculating the $\mathcal{L}_1$ distance between the ground truth colors $C^{\text{gt}}$ and the surface radiances $C^{\text{sur}}$. During the training process, only a few rays have intersection points with the surface. We only care about the set of selected rays $\mathcal{R}'$, which satisfies the condition that each ray exists point whose SDF value is less than zero and not the first sampled point. The loss is defined as follows.

$$\mathcal{L}_{\text{sur}} = \frac{1}{|\mathcal{R}'|}\sum_{\boldsymbol{r}\in\mathcal{R}'} \|C^{\text{sur}}_{\boldsymbol{r}} - C^{\text{gt}}_{\boldsymbol{r}}\|_1 \tag{18}$$

$\mathcal{L}_{\text{reg}}$ is an Eikonal loss term on the sampled points. Eikonal loss is a regularization loss applied to a set of sampling points

$X$, which is used to constrain the noise in signed distance function (SDF) generation.

$$\mathcal{L}_{\text{reg}} = \frac{1}{|\mathcal{X}|} \sum_{\boldsymbol{x} \in \mathcal{X}} (\|\nabla \mathsf{S}_\theta(\boldsymbol{x})\|_2 - 1)^2 \tag{19}$$

We use weights $\lambda_{\text{sur}}$ and $\lambda_{\text{reg}}$ to balance the impact of these three losses. The overall training weights are as follows.

$$\mathcal{L} = \mathcal{L}_{\text{vol}} + \lambda_{\text{sur}} \mathcal{L}_{\text{sur}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}} \tag{20}$$

## D. Details of stage 2

At this stage, we focus on predicting the lighting visibility and indirect illumination of a surface point $\boldsymbol{x}$ under different incoming light direction $\boldsymbol{\omega}_i$ using the SDF in the first stage. Therefore, we need first to calculate the position of the surface point $\boldsymbol{x}$. In stage one, we have calculated two sampling points $\boldsymbol{r}(t_{i'-1}), \boldsymbol{r}(t'_i)$ near the surface. As Geo-NeuS [11], we weigh these two sampling points to obtain a surface point $\boldsymbol{x}$ as follows.

$$\boldsymbol{x} = \frac{\mathsf{S}_\theta(\boldsymbol{r}(t_{i'-1}))\boldsymbol{r}(t'_i) - \mathsf{S}_\theta(\boldsymbol{r}(t'_i))\boldsymbol{r}(t_{i'-1})}{\mathsf{S}_\theta(\boldsymbol{r}(t_{i'-1})) - \mathsf{S}_\theta(\boldsymbol{r}(t'_i))} \tag{21}$$

**Learning lighting visibility.** Visibility is an important factor in shadow computation. It calculates the visibility of the current surface point $\boldsymbol{x}$ in the direction of the incoming light $\boldsymbol{\omega}_i$. Path tracing of the SDF is commonly used to obtain a binary visibility (0 or 1) as used in IndiSG [56], but this kind of visibility is not friendly to network learning. Inspired by NeRFactor [55], we propose to use an integral representation with the continuous weight function $w(t)$ (from 0 to 1) for the SDF to express light visibility. Specifically, we establish a neural network $\mathsf{M}_\nu : (\boldsymbol{x}, \boldsymbol{\omega}_i) \mapsto \nu$, that maps the surface point $\boldsymbol{x}$ and incoming light direction $\boldsymbol{\omega}_i$ to visibility, and the ground truth value of light visibility is obtained by integrating the weights $w_i$ of the SDF of sampling points along the incoming light direction and can be expressed as follows.

$$\nu^{gt} = 1 - \sum_{i=1}^{n} w_i \tag{22}$$

The weights of the light visibility network are optimized by minimizing the loss between the calculated ground truth values and the predicted values of a set of sampled incoming light directions $\Omega_i \subset \mathbb{S}^2$. This pre-integrated technique can reduce the computational burden caused by the integration for subsequent training.

$$\mathcal{L}_{\text{vis}} = \frac{1}{|\Omega_i|} \sum_{\boldsymbol{\omega} \in \Omega_i} \|\nu_{\boldsymbol{\omega}} - \nu^{\text{gt}}_{\boldsymbol{\omega}}\|_1 \tag{23}$$

**Learning indirect illumination.** Indirect illumination refers to the light that is reflected or emitted from surfaces in a scene and then illuminates other surfaces, rather than directly coming from a light source, which contributes to the realism of rendered images. Following IndiSG [56], we parameterize indirect illumination $I(\boldsymbol{x}, \boldsymbol{\omega}_i)$ via $K_i = 24$ Spherical Gaussians (SGs) as follows.

$$I(\boldsymbol{x}, \boldsymbol{\omega}_i) = \sum_{k=1}^{K_i} I_k(\boldsymbol{\omega}_i \mid \boldsymbol{\xi}^i_k(\boldsymbol{x}), \lambda^i_k(\boldsymbol{x}), \boldsymbol{\mu}^i_k(\boldsymbol{x})) \tag{24}$$

where $\boldsymbol{\xi}^i_k(\boldsymbol{x}) \in \mathbb{S}^2$, $\lambda^i_k(\boldsymbol{x}) \in \mathbb{R}_+$, and $\boldsymbol{\mu}^i_k(\boldsymbol{x}) \in \mathbb{R}^3$ are the lobe axis, sharpness, and amplitude of the $k$-th Spherical Gaussian, respectively. For this, we train a network $\mathsf{M}_{\text{ind}} : \boldsymbol{x} \mapsto \{\boldsymbol{\xi}^i_k, \lambda^i_k, \boldsymbol{\mu}^i_k\}_{k=1}^{K_i}$ that maps the surface point $\boldsymbol{x}$ to the parameters of indirect light SGs. Similar to learning visibility, we randomly sample several directions $\boldsymbol{\omega}_i$ from the surface point $\boldsymbol{x}$ to obtain (pseudo) ground truth $I^{\text{gt}}(\boldsymbol{x}, \boldsymbol{\omega}_i)$. Some of these rays have intersections $\boldsymbol{x}'$ with other surfaces, thus, $\boldsymbol{\omega}_i$ is the direction pointing from $\boldsymbol{x}$ to $\boldsymbol{x}'$. We query our proposed color network $\mathsf{M}_c$ to get the (pseudo) ground truth indirect radiance $I^{\text{gt}}(\boldsymbol{x}, \boldsymbol{\omega}_i)$ as follows.

$$I^{\text{gt}}(\boldsymbol{x}, \boldsymbol{\omega}_i) = \mathsf{M}_c(\boldsymbol{x}', \boldsymbol{n}', \boldsymbol{\omega}_i, \boldsymbol{v}_f) \tag{25}$$

where $\boldsymbol{n}'$ is the normal on the point $\boldsymbol{x}'$. We also use $\mathcal{L}_1$ loss to train the network.

$$\mathcal{L}_{\text{ind}} = \frac{1}{|M|} \sum_{m \in M} \|I(\boldsymbol{x}, \boldsymbol{\omega}_m) - I^{\text{gt}}_m(\boldsymbol{x}, \boldsymbol{\omega}_m)\|_1 \tag{26}$$

## E. Details of stage 3

The combination of light visibility and illumination SG is achieved by applying a ratio to the lobe amplitude of the output SG, while preserving the center position of the SG. We randomly sample $K_s = 32$ directions within the SG lobe and compute a weighted average of the visibility with different directions.

$$\begin{aligned} &\nu(\boldsymbol{x}, \boldsymbol{\omega}_i) \otimes E_k(\boldsymbol{\omega}_i \mid \boldsymbol{\xi}^e_k, \lambda^e_k, \boldsymbol{\mu}^e_k) \\ \approx &E_k(\boldsymbol{\omega}_i \mid \boldsymbol{\xi}^e_k, \lambda^e_k, \frac{\sum_{s=1}^{K_s} E_k(\boldsymbol{\omega}_s)\nu(\boldsymbol{x}, \boldsymbol{\omega}_s)}{\sum_{s=1}^{K_s} E_k(\boldsymbol{\omega}_s)} \boldsymbol{\mu}^e_k) \end{aligned} \tag{27}$$

Here, we offer intuitive explanations for why the incorporation of specular albedo in the model results in a decrease in lighting prediction. The increase in the model's complexity is the primary reason. Specular albedo introduces a more detailed modeling of surface reflection characteristics, requiring additional parameters and learning capacity. This raises the difficulty of training the model, potentially resulting in overfitting or training instability, thereby affecting the accurate prediction of lighting.

# F. Additional results

## F.1. Additional results for the main text

We conduct a more in-depth comparison of our method with the already published work NeRO. For DTU datasets, our findings demonstrate that NeRO performs less effectively than our approach on real datasets DTU, especially in the regular scenes of DTU shown in Fig. 10. NeRO fails to address the negative impact of partial highlights on the geometry. For example, the highlighted region of the skull model is reconstructed as overly flat, while the Buddha model loses numerous details that should have been retained. Similar issues are also observed in the two plush toy scenes. Moreover, the presence of shadows causes NeRO to mistakenly reconstruct shadowed areas as real objects and fill them in (bricks and skull models). Fig. 11 shows three other scenes (helmet, teapot, and car) from the Shiny dataset. We have demonstrated that our method performs better than other methods on the Shiny dataset. NeRO exhibits defects in the dents and highlights of the helmet, as well as in the wheels of the car. Furthermore, we extend our comparison to include more scenes in the glossy dataset in Fig. 12 and Tab. 5, where although NeRO performs better, our method is also capable of mitigating the impact of highlights on geometry. Our method demonstrates comparable results to NeRO. Moreover, compared to NeuS, the results show a significant improvement. Note that the real datasets include bear, bunny, coral, and vase. The rest of the others are synthetic. As for material representation, we adopted the spherical Gaussians to represent the Ward BRDF model [45], while NeRO uses the spherical harmonics to represent the Disney BRDF model [5]. Tab. 6 and Fig. 14 show that we outperform NeRO in materials and rendering.

In order to prove the effectiveness of the combination of volume rendering and decoupled surface rendering, in Fig. 13, we additionally present the qualitative evaluation for more objects. It is evident that surface rendering is essential for decomposing diffuse and specular components, ensuring smooth reconstruction of glossy surfaces with complex reflections.

Fig. 15 illustrates the qualitative results of material reconstruction on the other scenes of the IndiSG dataset, highlighting the effectiveness of our method. For completeness, we visualize the decomposition of diffuse and specular in the first stage in Fig. 16. In the first stage, the decomposition of diffuse and specular is not a true BRDF model. This is because the MLP in the first stage is used solely for predicting the components of diffuse and specular reflection, rather than predicting material properties such as albedo and roughness. The decision to directly predict colors instead of material properties in the first stage serves two purposes: reducing model complexity by focusing on the direct prediction of specular reflection color, and optimizing geometry for better reconstruction. By decomposing highlights through the network in the first stage, surfaces with specular reflections can be reconstructed more effectively, demonstrated by the presence of flower pot ablation, and without encountering the concavity issues observed in other methods.

In Fig. 17, from the DTU data, we can observe that our method can separate the specular reflection component from the diffuse reflection component, as seen in the highlights on the apple, can, and golden rabbit. Even when faced with a higher intensity of specular reflection, as demonstrated in the example showcased in SK3D, our method excels at preserving the original color in the diffuse part and accurately separating highlights into the specular part.

In Fig. 18, we show the diffuse albedo and rendering results of NVDiffrec, IndiSG, and our method. The rendering results indicate that our method can restore the original appearance with specular highlights more accurately, such as the reflections on the helmet and toaster compared to the IndiSG and NVDiffrec methods. The material reconstruction results show that our diffuse albedo contains less specular reflection information compared to other methods, indicating our method has a better ability to suppress decomposition ambiguity caused by specular highlights.

Additionally, in Fig. 19, Fig. 20, and Fig. 21, we presented all components, the rendering, albedo, roughness, diffuse color, specular color, light visibility, indirect light, and environment light results for the IndiSG, DTU and SK3D datasets, respectively. An interesting observation is that our reconstructed environment maps have the capability to represent multiple direct light illuminants, as demonstrated in the DTU dataset.

In Fig. 22, we additionally showcase the visualization results of relighting compared with the IndiSG method. IndiSG and ours yield different predictions for material, resulting in variations in the relighting results, but the relighting results generated by our method exhibit richer details. Our method demonstrates the practical utility employed in the relighting scenarios.

In Fig. 23, we show that introducing specular albedo also makes the sausage appear smoother and closer to its true color roughness, represented by black. In terms of lighting, when not using specular albedo, the lighting reconstruction achieves the best result, indicating a clearer reconstruction of ambient illumination. In summary, our ablation study highlights the importance of taking into account various factors when reconstructing materials and illumination from images. By evaluating the performance of different modules, we can better understand their role in improving the reconstruction quality.

In stage 3, if we do not consider indirect illumination during the training process, the predicted results for rendering, material, and lighting will all experience a decline. The qualitative results are shown in Fig. 24.

Table 5. Comparison with NeRO and NeuS on Glossy dataset.

| Glossy | bear | bunny | coral | maneki | vase | angel | bell | cat | horse | luyu | potion | tbell | teapot | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| NeuS | 0.0074 | 0.0022 | 0.0016 | 0.0091 | 0.0101 | 0.0035 | 0.0146 | 0.0278 | 0.0053 | 0.0066 | 0.0393 | 0.0348 | 0.0546 | 0.0167 |
| NeRO | **0.0033** | **0.0012** | **0.0014** | **0.0024** | **0.0011** | **0.0034** | **0.0032** | **0.0044** | **0.0049** | **0.0054** | **0.0053** | **0.0035** | **0.0037** | **0.0033** |
| Ours | 0.0034 | 0.0017 | **0.0014** | 0.0027 | 0.0023 | **0.0034** | 0.0054 | 0.0059 | 0.0052 | 0.0060 | 0.0058 | **0.0035** | 0.0105 | 0.0044 |

Table 6. Comparison of material rendering on IndiSG dataset.

| | Baloons | | | Hotdog | | |
|---|---|---|---|---|---|---|
| | albedo | rough | render | albedo | rough | render |
| NeRO | 14.65 | 18.91 | 23.84 | 11.54 | 18.42 | 26.95 |
| Ours | **25.79** | **19.75** | **33.89** | **30.72** | **23.10** | **36.71** |

## F.2. Additional experiments

To have a fair comparison with Geo-NeuS on the DTU dataset, we incorporate the components of Geo-NeuS based on the additional data (the point clouds from SfM and image pairs) used in Geo-NeuS into our method. As shown in Tab. 7, our approach can further enhance the surface reconstruction quality on datasets where highlights are less pronounced.

Table 7. Quantitative results in terms of Chamfer distance on DTU [17].

| | DTU 63 | DTU 97 | DTU 110 | **Mean** |
|---|---|---|---|---|
| Geo-NeuS [11] | 0.96 | 0.91 | 0.70 | 0.86 |
| Factored-NeuS (ours) | 0.99 | 1.15 | 0.89 | 1.01 |
| Factored-NeuS (ours w/ Geo) | **0.95** | **0.89** | **0.69** | **0.84** |

We conduct another experiment to compare our modeling approach with Ref-NeRF and $S^3$-NeRF [47]. The experimental quantitative and qualitative results are shown in Fig. 25. Ref-NeRF utilizes volume rendering colors for diffuse and specular components. If we directly combine SDF and the architecture of Ref-NeRF, it is challenging to eliminate the influence of highlights. Furthermore, if we applied the construction method of $S^3$-NeRF, which involves integrating surface rendering colors into volume rendering, to modify our model structure, we found that this modeling approach cannot address the issue of geometric concavity caused by highlights.

For chrome-like materials, We increase the Fresnel value to 0.75 in the rendering formula of stage 3 to test the impact of this operation. We show the results and their PSNR value in Fig. 26, we observed that increasing the Fresnel value indeed leads to the better reconstruction of objects with chrome-like materials. For the Toaster model, we observed a significant improvement in PSNR with an increased Fresnel value. However, we also noticed that solely increasing the Fresnel value can result in the degradation of texture details. For instance, in the Coffee model, although the highlights on the spoon are better reconstructed, the text on the cup deteriorates. One of our future directions is to address this issue more effectively.

## G. Limitations

In certain scenarios, our method still faces difficulties. For mesh reconstruction, despite improvements on the glossy parts in the DTU 97 tin model, the overall Chamfer distance does not significantly decrease due to the small proportion of glossy parts. However, for scenes with large areas of glossy parts, such as the flower pot model, our improvements are more pronounced and surpass Geo-NeuS. As seen in Appx Fig. 15, the reconstructed albedo of the chair still lacks some detail. The nails on the chair and the textures on the pillow are not accurately captured in the reconstructed geometry. A future research direction is how to effectively decompose materials for fine structures, such as nails on the backrest of a chair.

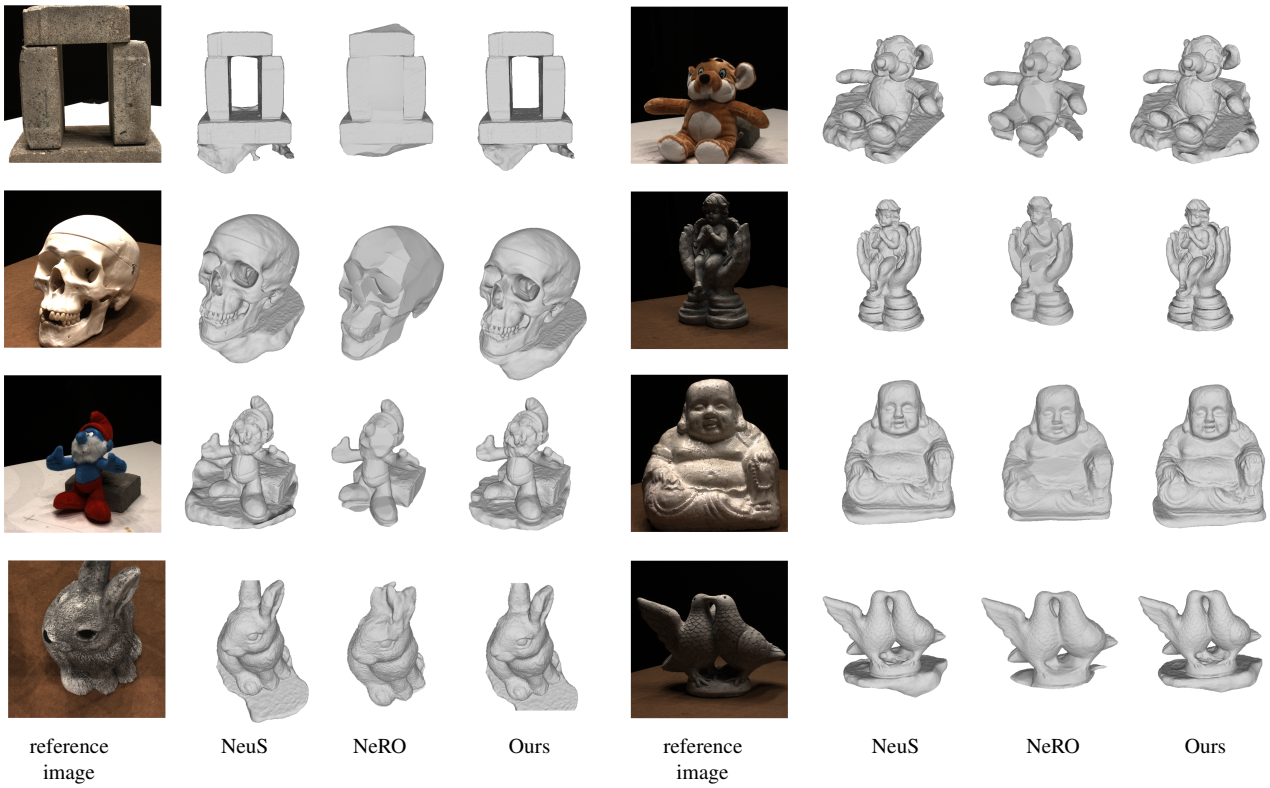



Figure 10. Qualitative results on regular scenes from DTU. with NeRO and NeuS on DTU dataset. Results show that NeRO fails to address the negative impact of partial highlights on the geometry. Moreover, the presence of shadows causes NeRO to mistakenly reconstruct shadowed areas as real objects and fill them in (bricks and skull models).

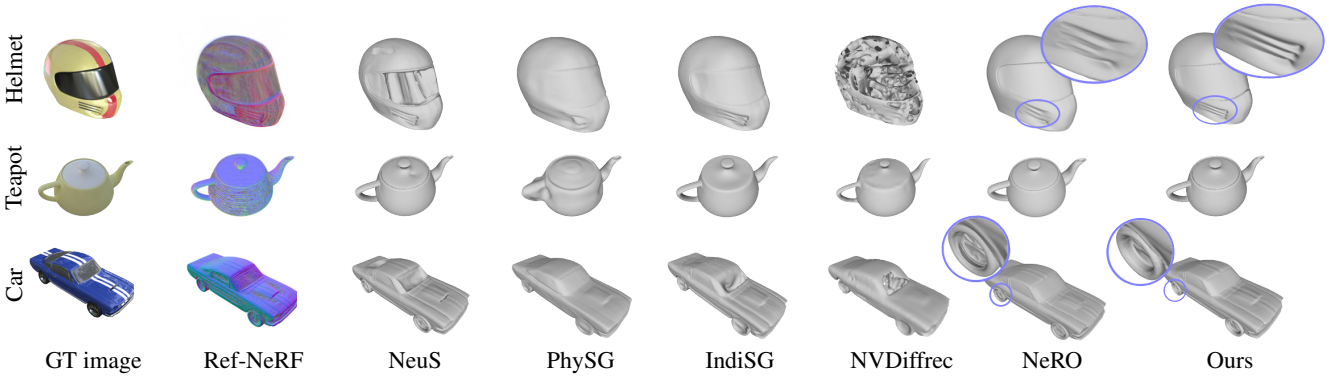



Figure 11. Qualitative results on other scans (helmet, teapot, and car) from the Shiny dataset [39]. NeRO exhibits defects in the dents and highlights of the helmet, as well as in the wheels of the car.

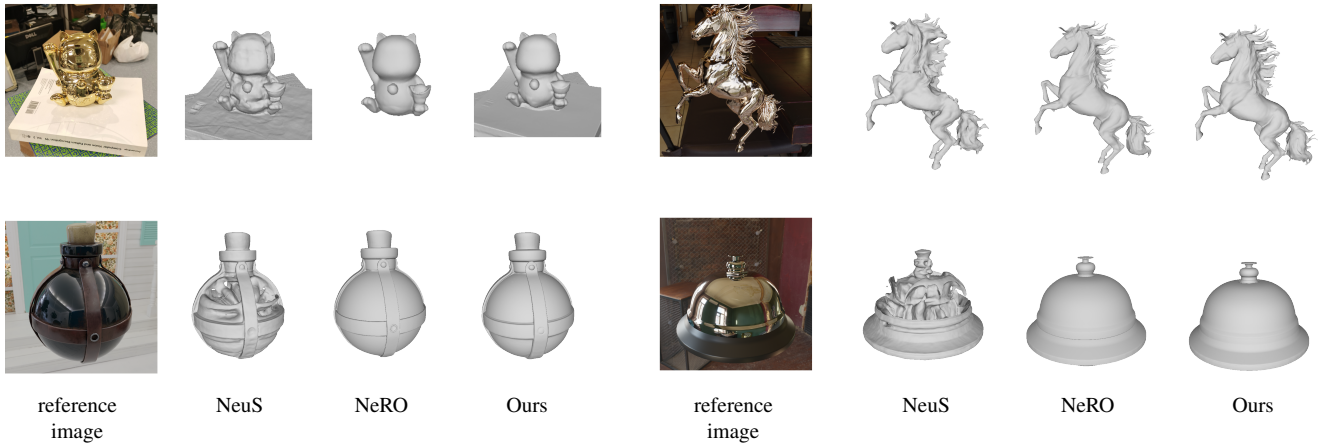



Figure 12. Qualitative results on other scenes (bell, potion, horse, and tbell) compared with NeRO and NeuS on the Glossy dataset.

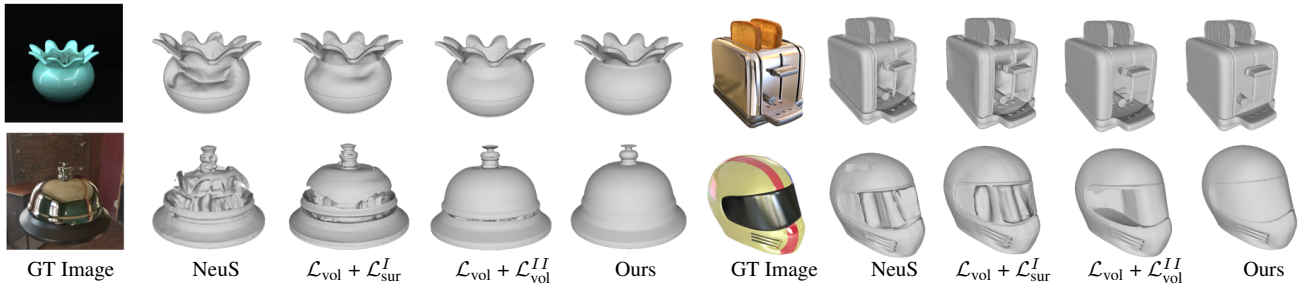



Figure 13. Qualitative ablation evaluation for more objects.

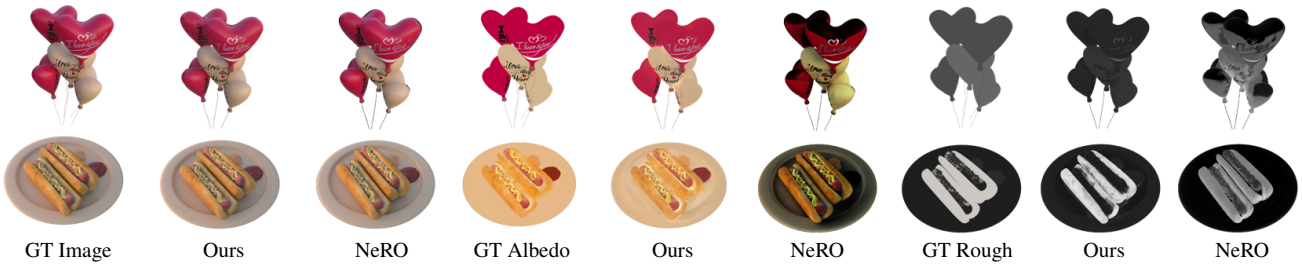



Figure 14. Qualitative comparison with NeRO in terms of material reconstruction and rendering quality.

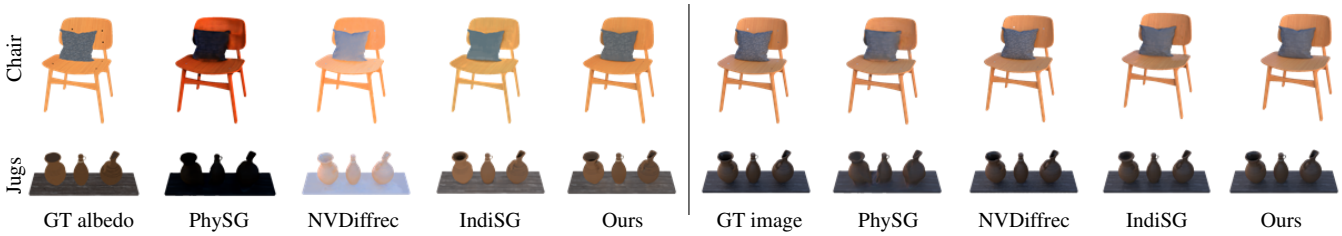



Figure 15. Qualitative results for other scenes (chair and jugs) on IndiSG dataset in terms of albedo reconstruction (left) and novel view synthesis quality (right).


Figure 16. Diffuse and specular decomposition results in the first stage.

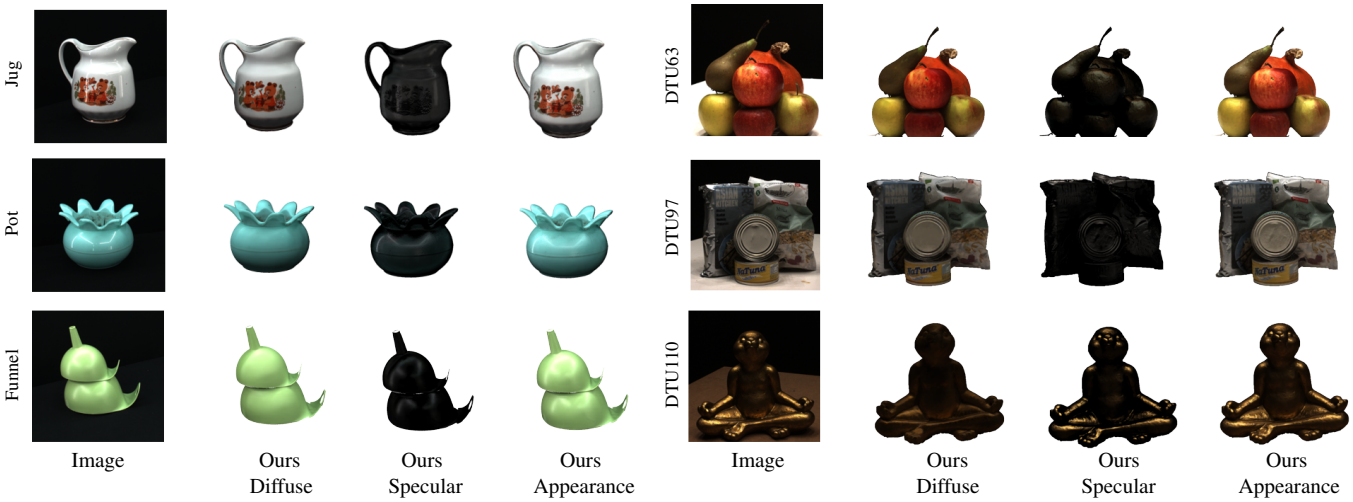



Figure 17. Qualitative results for the SK3D (left) and DTU (right) datasets in the third stage. We can observe that our method can separate the specular reflection component from the diffuse reflection component, as seen in the highlights on the apple, can, and golden rabbit. Even when faced with a higher intensity of specular reflection in SK3D, our method can separate them very well.

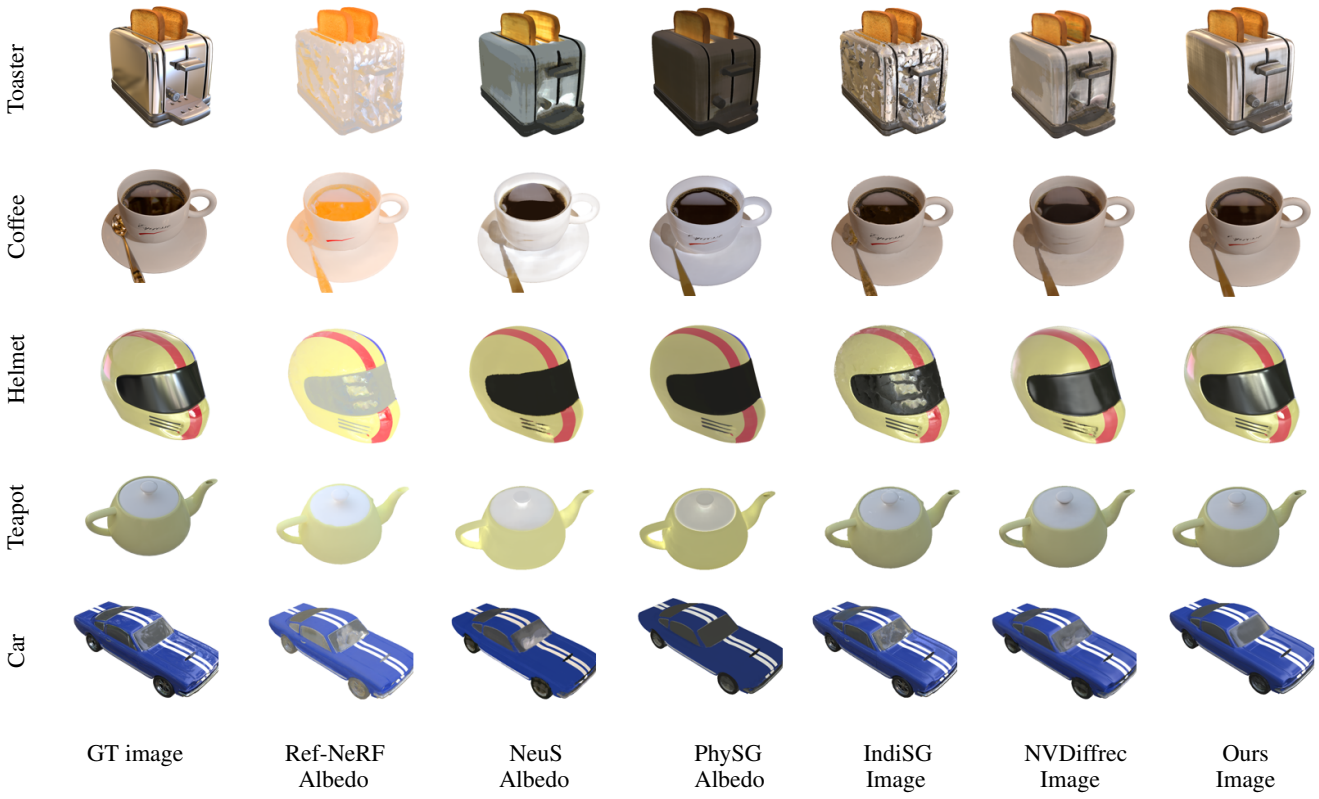



Figure 18. Qualitative results of materials reconstruction for the Shiny dataset, where albedo refers to the diffuse albedo. The results indicate that our method can restore the original appearance with specular highlights more accurately, such as the reflections on the helmet and toaster, and has a better ability to suppress decomposition ambiguity caused by specular highlights.

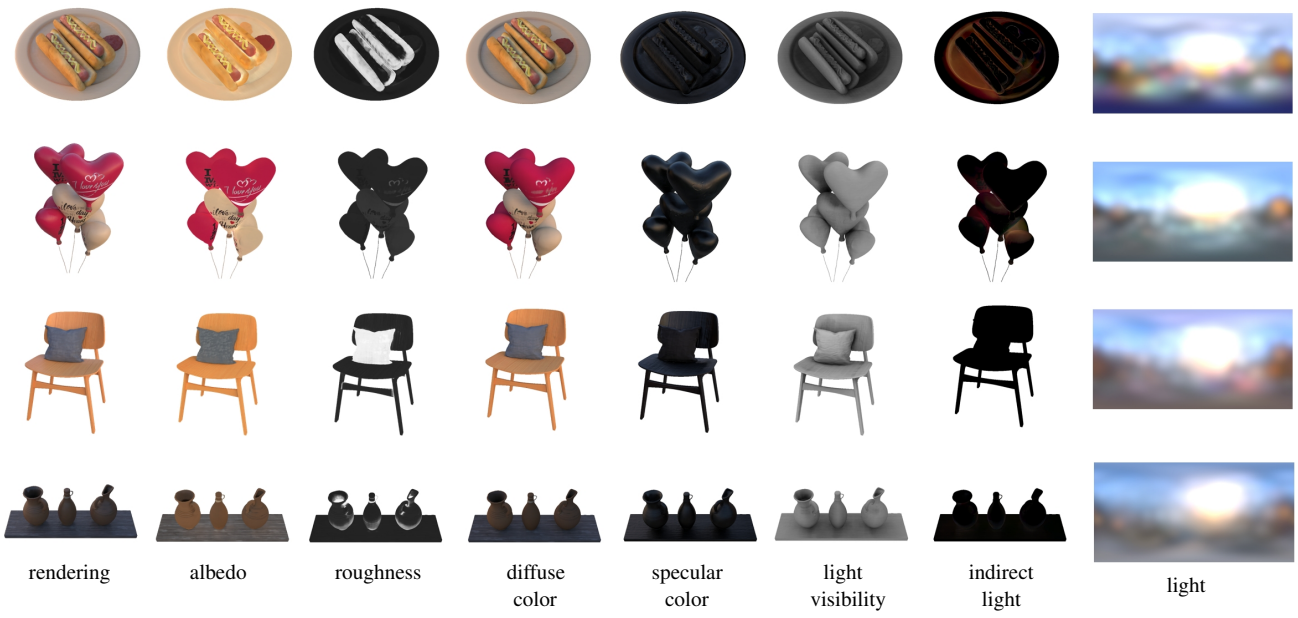



Figure 19. Visualization of all components on IndiSG dataset.

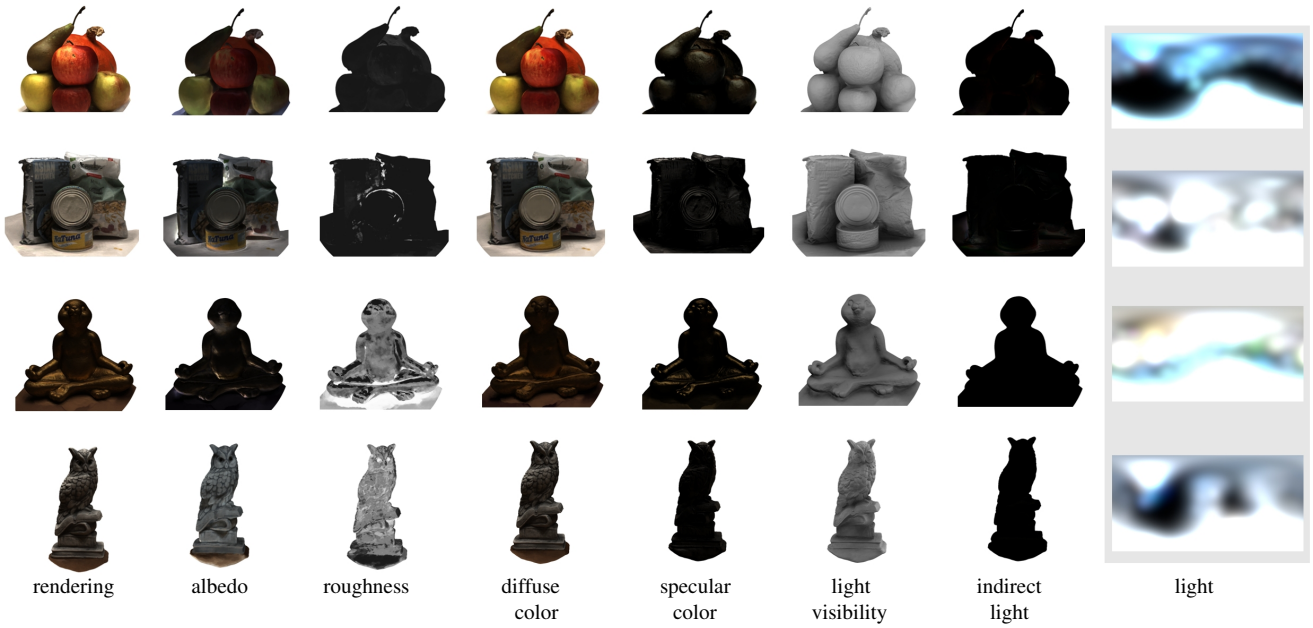



Figure 20. Visualization of all components on DTU dataset.


Figure 21. Visualization of all components on SK3D dataset.

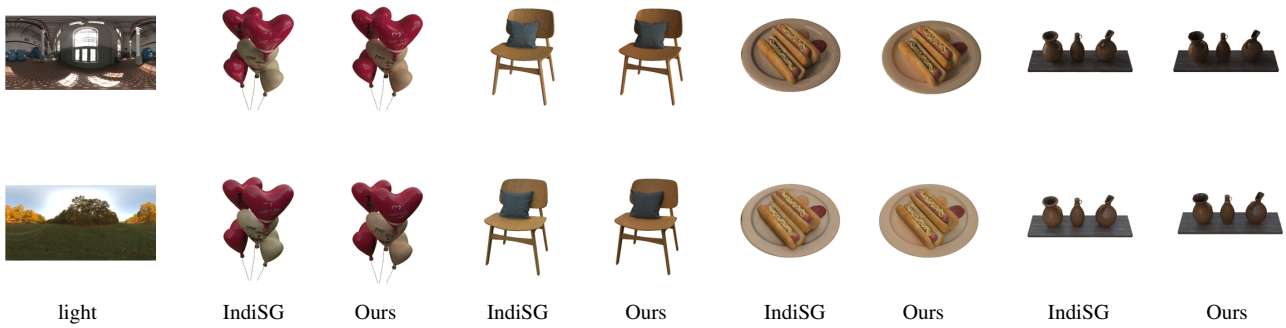



Figure 22. Relighting comparison with IndiSG.

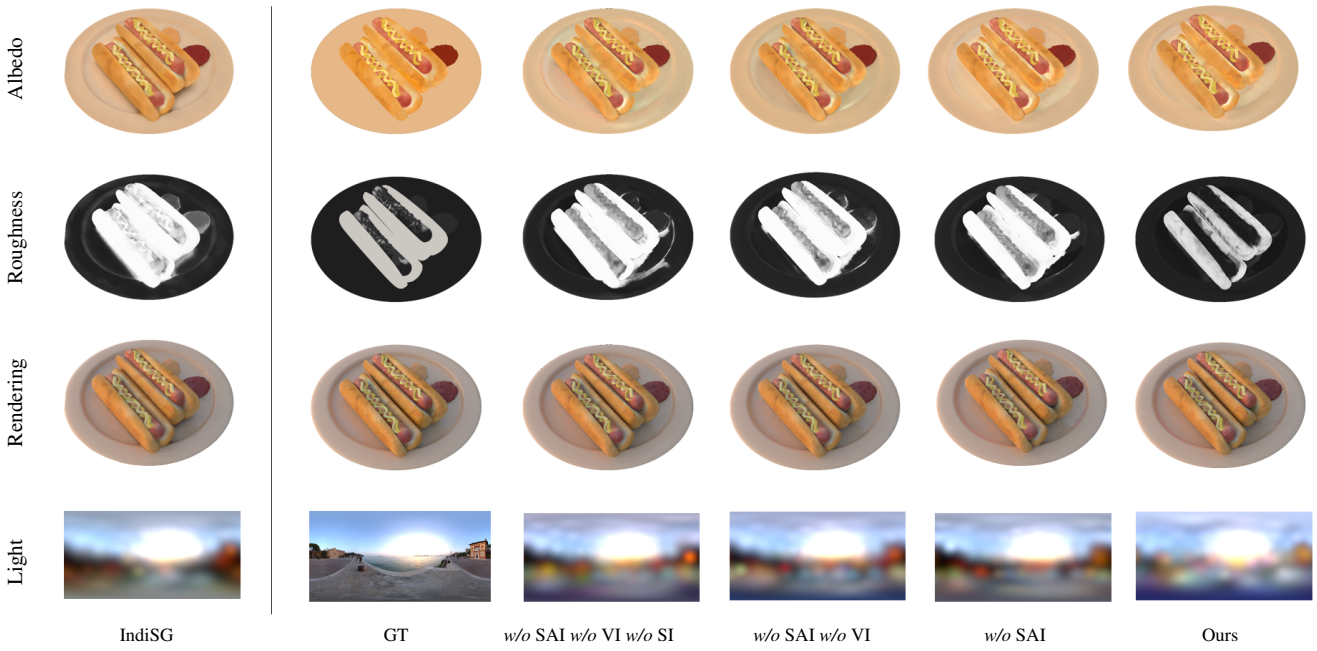



Figure 23. Ablation study of material and illumination reconstruction. We show that introducing specular albedo also makes the sausage appear smoother and closer to its true color roughness, represented by black. In terms of lighting, when not using specular albedo, the lighting reconstruction achieves the best result, indicating a clearer reconstruction of ambient illumination.

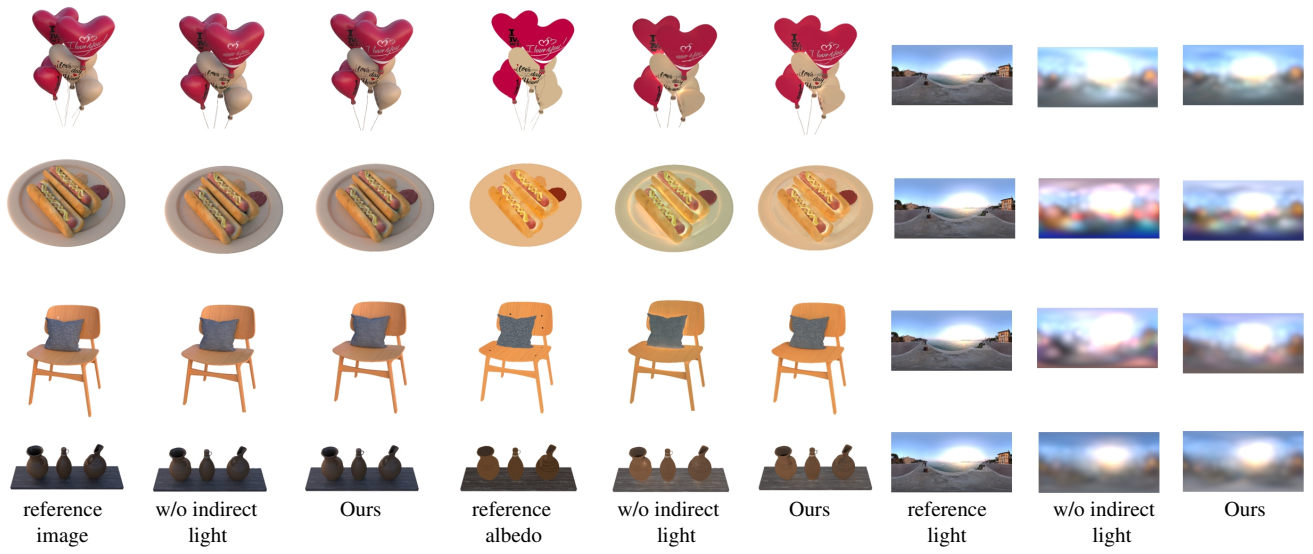
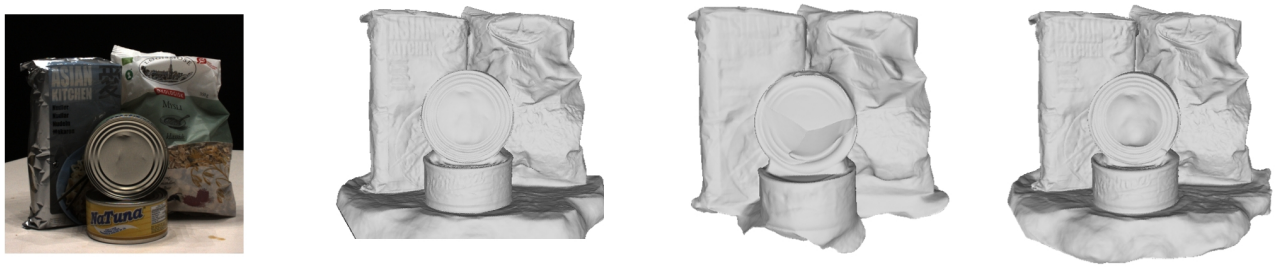



Figure 24. Ablation study of indirect light.



reference image | Ours 1.15 | Ours with Ref-NeRF 2.17 | Ours with $S^3$-NeRF 1.23

Figure 25. Comparison with Ref-NeRF and $S^3$-NeRF.

GT image | f=0.02 22.33 | f=0.75 24.31 | GT image | f=0.02 30.43 | f=0.75 30.18

Figure 26. Adjusting Fresnel value to model chrome-like appearance.