Supplementary Document for NeISF++: Neural Incident Stokes Field for Polarized Inverse Rendering of Conductors and Dielectrics

This is the supplementary document for NeISF++. We discuss some details that are not shown in the main text due to space limitations. The rest of this document is structured as follows:

- Sec. A: Our pBRDF
- Sec. B: The correctness of our polarimetric renderer
- Sec. C: The neural fields used in the geometry initialization and joint optimization stages.
- Sec. D: Our synthetic and real multi-view HDR polarized datasets
- Sec. E: Additional results
- Sec. F: Implementation details

A. Our pBRDF

This section is the complete description of our material model. Our rendered Stokes vectors $\mathbf{s}$ can be obtained as follows:

$$\mathbf{s} = m\mathbf{s}_{\text{dif}} + \mathbf{s}_{\text{spec}}, \tag{1}$$

where $\mathbf{s}_{\text{dif}}$ and $\mathbf{s}_{\text{spec}}$ are the Stokes vectors from the diffuse and specular polarization, m is the binary indicator which determines the existence of the diffuse polarization. It should be 0 when rendering conductors and 1 when rendering dielectrics. The diffuse and specular components can be represented as follows:

$$\mathbf{s}_{\text{dif}} = \int_{\Omega} \mathbf{R}_{\text{dif}}^{\text{cam}} \cdot \mathbf{M}_{\text{dif}} \cdot \mathbf{s}_{\text{dif}}^{\text{r}} \, d\boldsymbol{\omega_i}, \tag{2}$$

$$\mathbf{s}_{\text{spec}} = \int_{\Omega} \mathbf{R}_{\text{spec}}^{\text{cam}} \cdot \mathbf{M}_{\text{spec}} \cdot \mathbf{s}_{\text{spec}}^{\text{r}} \, d\boldsymbol{\omega_i}. \tag{3}$$

Where $\mathbf{s}_{\text{dif}}^{\text{r}}$ and $\mathbf{s}_{\text{spec}}^{\text{r}}$ are the already rotated incident Stokes vectors of diffuse and specular components. $\mathbf{R}_{\text{dif}}^{\text{cam}}$ is the rotation Mueller matrix, which rotates Stokes vectors to the camera's reference frame:

$$\mathbf{R}_{\text{dif}}^{\text{cam}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(2\phi_{\text{dif}}) & \sin(2\phi_{\text{dif}}) \\ 0 & -\sin(2\phi_{\text{dif}}) & \cos(2\phi_{\text{dif}}) \end{bmatrix}. \tag{4}$$

ϕ_{dif} is the rotation angle. $\mathbf{M}_{\text{dif}}$ is the material Mueller matrix of diffuse polarization, which describes the process of light entering the object, scattering multiple times under the surface, and finally leaving the object. It can be formulated as follows:

$$\mathbf{M}_{\text{dif}} = (\frac{\mathrm{a}}{\pi}\cos\theta_i)\mathbf{F}_o^T \cdot \mathbf{D} \cdot \mathbf{F}_i^T. \tag{5}$$

a is the diffuse albedo, $\theta_{i,o}$ denotes the incident/outgoing angle, $\mathbf{D} \in \mathbb{R}^{3\times3}$ is a depolarizer:

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{6}$$

the Fresnel transmission term $\mathbf{F}_{i,o}^T \in \mathbb{R}^{3\times3}$ is defined as:

$$\mathbf{F}_{i,o}^T = \begin{bmatrix} T_{i,o}^+ & T_{i,o}^- & 0 \\ T_{i,o}^- & T_{i,o}^+ & 0 \\ 0 & 0 & T_{i,o}^\times \end{bmatrix}, \tag{7}$$

where $T_{i,o}^+ = (T_{i,o}^\perp + T_{i,o}^\parallel)/2$, $T_{i,o}^- = (T_{i,o}^\perp - T_{i,o}^\parallel)/2$, and $T_{i,o}^\times = \sqrt{T_{i,o}^\perp T_{i,o}^\parallel}$. Where $T_{i,o}^\perp$ is the perpendicular term of the transmission coefficient:

$$T_{i,o}^\perp = \frac{4\cos\theta_{i,o}\sqrt{\eta^2 - \sin^2\theta_{i,o}}}{(\cos\theta_{i,o} + \sqrt{\eta^2 - \sin^2\theta_{i,o}})^2}, \tag{8}$$

and $T_{i,o}^\parallel$ is the parallel term of the transmission coefficient:

$$T_{i,o}^\parallel = \frac{4\eta^2\cos\theta_{i,o}\sqrt{\eta^2 - \sin^2\theta_{i,o}}}{(\eta^2\cos\theta_{i,o} + \sqrt{\eta^2 - \sin^2\theta_{i,o}})^2}, \tag{9}$$

where η is the real part of the refractive index. Then, we can rewrite Eq. 2 as follows:

$$\mathbf{s}_{\text{dif}} = \int_{\Omega} \frac{\mathrm{a}}{\pi}\cos\theta_i \begin{bmatrix} T_o^+T_i^+ & T_o^+T_i^- & 0 \\ T_o^-T_i^+\cos(2\phi_{\text{dif}}) & T_o^-T_i^-\cos(2\phi_{\text{dif}}) & 0 \\ -T_o^-T_i^+\sin(2\phi_{\text{dif}}) & -T_o^-T_i^-\sin(2\phi_{\text{dif}}) & 0 \end{bmatrix} \cdot \mathbf{s}_{\text{dif}}^{\text{r}} \, d\boldsymbol{\omega_i}. \tag{10}$$

When rendering RGB images, Eq. 10 should be repeated three times with separate diffuse albedos. Similarly, $\mathbf{R}_{\text{spec}}^{\text{cam}}$ with the rotation angle ϕ_{spec} is as follows:

$$\mathbf{R}_{\text{spec}}^{\text{cam}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(2\phi_{\text{spec}}) & \sin(2\phi_{\text{spec}}) \\ 0 & -\sin(2\phi_{\text{spec}}) & \cos(2\phi_{\text{spec}}) \end{bmatrix}. \tag{11}$$

$\mathbf{M}_{\text{spec}}$ is the material Mueller matrix, which describes the single bounce mirror reflection. It can be written as follows:

$$\mathbf{M}_{\text{spec}} = k_s \frac{DG}{4\cos\theta_o} \mathbf{F}^R, \tag{12}$$

where k_s is the specular coefficient. GGX distribution function D [14] is defined as follows:

$$D = \frac{r^2}{\pi \cos^4\theta_h (r^2 + \tan^2\theta_h)^2}, \tag{13}$$

where r is the roughness, θ_h is the angle between the halfway vector and surface normal. Smith G function [6] is as follows:

$$G = \left(\frac{2}{1+\sqrt{1+r^2\tan^2\theta_i}}\right)\left(\frac{2}{1+\sqrt{1+r^2\tan^2\theta_o}}\right). \tag{14}$$

Fresnel reflection term $\mathbf{F}^R \in \mathbb{R}^{3\times3}$ is as follows:

$$\mathbf{F}^R = \begin{bmatrix} R^+ & R^- & 0 \\ R^- & R^+ & 0 \\ 0 & 0 & R^\times \cos\Delta \end{bmatrix}, \tag{15}$$

$R^+ = (R^\perp + R^\parallel)/2$, $R^- = (R^\perp - R^\parallel)/2$, and $R^\times = \sqrt{R^\perp R^\parallel}$. $R^\perp$ is the perpendicular term of the reflection coefficient:

$$R^\perp = |(r^\perp)^2|, \tag{16}$$

$$r^\perp = \frac{\cos\theta_o - (\eta - ik)\cos\theta_t}{\cos\theta_o + (\eta - ik)\cos\theta_t}, \tag{17}$$

and $R^\parallel$ is the parallel term of the reflection coefficient:

$$R^\parallel = |(r^\parallel)^2|, \tag{18}$$

$$r^\parallel = \frac{(\eta - ik)\cos\theta_o - \cos\theta_t}{(\eta - ik)\cos\theta_o + \cos\theta_t}, \tag{19}$$

where η and k are the real and imaginary components of the complex refractive index. $\cos\theta_t$ is the angle of transmission, and it can be calculated using Snell's law:

$$\cos\theta_t = \sqrt{1 - \frac{\sin^2\theta_o}{(\eta - ik)^2}}. \tag{20}$$

Δ is the phase shift. in Baek pBRDF [2] $\cos\Delta$ is hard coded as -1 or 1 when the incident angle is less or larger than the Brewster angle. However, this hard coding does not hold for the complex refractive index. We implement the phase shift based on the Fresnel wave theory [3]. Let $x^\perp$ and $y^\perp$ be the real and imaginary component of $r^\perp$:

$$r^\perp = x^\perp - iy^\perp, \tag{21}$$

and similarly:

$$r^\parallel = x^\parallel - iy^\parallel. \tag{22}$$

Then, the cosine value of the phase shift can be calculated as follows:

$$\begin{aligned} \cos\Delta &= \cos(\arg(r^\parallel) - \arg(r^\perp)) \\ &= \cos(\delta^\parallel - \delta^\perp) \\ &= \cos\delta^\parallel \cos\delta^\perp + \sin\delta^\parallel \sin\delta^\perp \\ &= \frac{x^\parallel x^\perp + y^\parallel y^\perp}{\sqrt{(x^\parallel)^2 + (y^\parallel)^2}\sqrt{(x^\perp)^2 + (y^\perp)^2}} \end{aligned} \tag{23}$$

Then Eq. 3 can be rewritten as follows:

$$\mathbf{s}_{\text{spec}} = \int_\Omega k_s \frac{DG}{4\cos\theta_o} \begin{bmatrix} R^+ & R^- & 0 \\ R^-\cos 2\phi_{\text{spec}} & R^+\cos 2\phi_{\text{spec}} & R^\times \sin 2\phi_{\text{spec}} \cos\psi \\ -R^-\sin 2\phi_{\text{spec}} & -R^+\sin 2\phi_{\text{spec}} & R^\times \cos 2\phi_{\text{spec}} \cos\psi \end{bmatrix} \cdot \mathbf{s}_{\text{spec}}^{\text{r}}\, d\boldsymbol{\omega_i}, \tag{24}$$

where ϕ_{spec} is the rotation angle from the reference frame of $\mathbf{M}_{\text{spec}}$ to the camera axis. Similar to the diffuse term, Eq. 24 also needs to be repeated three times with separate specular coefficients and refractive indexes when rendering RGB images.

B. Polarimetric Renderer

Our polarimetric renderer is implemented in PyTorch [11]. To verify correctness, we compare the rendering results with Mitsuba 3 [7], a physically-based renderer that implements a large number of BRDF plugins. We compare with the *rough conductor* BRDF plugin for conductors, and *polarized plastic* BRDF plugin for dielectrics. These two BRDFs can be considered as two special cases as our general pBRDF. For simplicity, we only render a sphere under uniform illumination. For the conductor, we use the refractive index of gold and set the roughness $r = 0.1$. For the dielectric, we set the refractive index $ior = 1.5$, the diffuse albedo $\boldsymbol{a} = [0.5, 0.5, 0.5]$, and the roughness $r = 0.5$. As shown in Fig. 1, the rendered results of our renderer are consistent with Mitsuba 3. The error on the sphere's boundary mainly comes from the aliasing problem.

C. Neural Fields

As briefly mentioned in the main paper, the training is divided into the geometry initialization and joint optimization

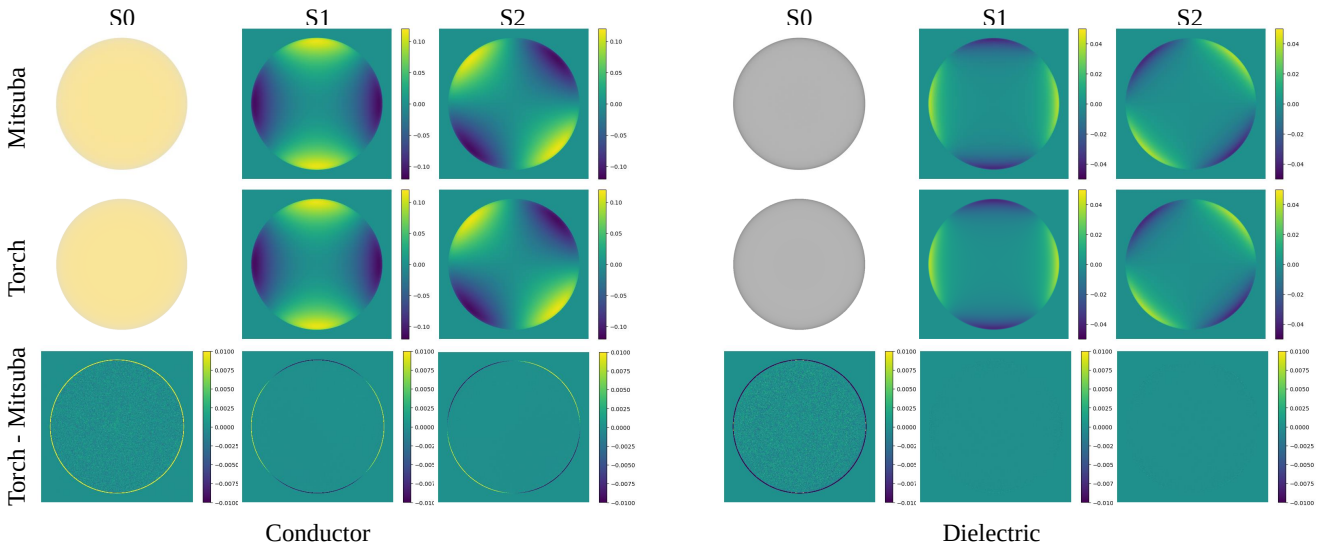

Figure 1. The rendering results between our renderer (denoted as "Torch" in the figure) and Mitsuba 3.0 [7].

stages. In this section, we provide a complete description of these two stages. For each pixel, we cast a single ray along the current direction $\boldsymbol{\omega}_o$. We denote $\{\mathbf{x}^j\}_{j=1}^N$ as N samples along the ray.

C.1. Geometry Initialization

The geometry initialization stage is a variant of VolSDF [15]. It consists of three MLPs, which are the SDF net f_{sdf}, the DoLP net f_{DoLP}, and the outgoing intensity net f_{oi}. The SDF net models the signed distance function:

$$f_{\text{sdf}}(\mathbf{x}^j) = d^j, \mathbf{v}^j \tag{25}$$

where d^j is the signed distance from the nearest surface, $\mathbf{v}^j$ is the feature vector. For simplicity, we omit the cosine embedding [10] in equations. The normal $\mathbf{n}^j$ of the sampled point $\mathbf{x}^j$ is the gradient of f_{sdf}:

$$\frac{\nabla_{\mathbf{x}^j} f_{\text{sdf}}(\mathbf{x}^j)}{||\nabla_{\mathbf{x}^j} f_{\text{sdf}}(\mathbf{x}^j)||_2} = \mathbf{n}^j. \tag{26}$$

The DoLP net f_{DoLP} models the appearance of the DoLP $\boldsymbol{\rho}^j$. Because the DoLP is a view-dependent property, the direction information $\boldsymbol{\omega}_o$ must also be given to the MLP:

$$f_{\text{DoLP}}(\mathbf{x}^j, \boldsymbol{\omega}_o, \mathbf{n}^j, \mathbf{v}^j) = \boldsymbol{\rho}^j. \tag{27}$$

Similar to the original VolSDF [15], we also model the intensity as a position and view-dependent function:

$$f_{\text{oi}}(\mathbf{x}^j, \boldsymbol{\omega}_o, \mathbf{n}^j, \mathbf{v}^j) = \mathbf{I}^j. \tag{28}$$

The final DoLP and intensity are computed via alpha-blending:

$$\boldsymbol{\rho} = \sum_{j=1}^{N} w^j \boldsymbol{\rho}^j, \tag{29}$$

$$\mathbf{I} = \sum_{j=1}^{N} w^j \mathbf{I}^j \tag{30}$$

The weight w^j is computed via:

$$w^j = T^j(1 - \exp(-\sigma^j \delta^j)), \tag{31}$$

where

$$T^j = \exp(-\sum_{l=1}^{j-1} \sigma^l \delta^l), \tag{32}$$

δ is the distance between two adjacent samples. σ is the density, following VolSDF [15], we model it as:

$$\sigma^j = \gamma \Psi_\beta(d^j), \tag{33}$$

where Ψ is the cumulative distribution function of the Laplace distribution. γ and β are two learnable parameters.

C.2. Joint Optimization

The joint optimization stage consists of eight MLPs. The SDF net f_{sdf} is initialized by the geometry initialization stage, and will continue to be trained in the joint optimization stage. The normal $\mathbf{n}^j$ is still computed using Eq. 26. The BRDF fields have four MLPs:

$$f_{\text{alb}}(\mathbf{x}^j) = \mathbf{a}^j, \tag{34}$$

$$f_{\text{rough}}(\mathbf{x}^j) = r^j, \tag{35}$$

$$f_{\text{eta}}(\mathbf{x}^j) = \boldsymbol{\eta}^j, \tag{36}$$

$$f_{\text{k}}(\mathbf{x}^j) = \mathbf{k}^j, \tag{37}$$

which estimates the albedo $\mathbf{a}^j$, the roughness r^j, the real part of the refractive index $\boldsymbol{\eta}^j$, and the imaginary part of the refractive index $\mathbf{k}^j$ of the point $\mathbf{x}^j$. The parameters used for rendering are computed using alpha-blending:

$$\mathbf{n} = \sum_{j=1}^{N} w^j \mathbf{n}^j, \tag{38}$$

$$\mathbf{a} = \sum_{j=1}^{N} w^j \mathbf{a}^j, \tag{39}$$

$$r = \sum_{j=1}^{N} w^j r^j, \tag{40}$$

$$\boldsymbol{\eta} = \sum_{j=1}^{N} w^j \boldsymbol{\eta}^j, \tag{41}$$

$$\mathbf{k} = \sum_{j=1}^{N} w^j \mathbf{k}^j. \tag{42}$$

Following NeISF [9], we model the lighting as incident Stokes fields, which consists of three MLPs:

$$f_{\text{ii}}(\mathbf{x}, \boldsymbol{\omega}_i) = \mathbf{s}^{\text{r}}_{\text{spec}}[0] = \mathbf{s}^{\text{r}}_{\text{dif}}[0], \tag{43}$$

$$f_{\text{spec}}(\mathbf{x}, \boldsymbol{\omega}_i) = \mathbf{s}^{\text{r}}_{\text{spec}}[1, 2], \tag{44}$$

$$f_{\text{dif}}(\mathbf{x}, \boldsymbol{\omega}_i) = \mathbf{s}^{\text{r}}_{\text{dif}}[1]. \tag{45}$$

where $\mathbf{s}^{\text{r}}_{\text{spec}}[0]$ and $\mathbf{s}^{\text{r}}_{\text{dif}}[0]$ are the first element (unpolarized intensity) of the incident Stokes vectors of the specular and diffuse polarization. They should have the same value, thus we use a single MLP f_{ii} to model them. f_{spec} models the second and third elements of the incident Stokes vectors of the specular polarization term. f_{dif} models the second element of the incident Stokes vector of the diffuse polarization term. Note that for the diffuse polarization term, modeling the third element is meaningless. Because it will be canceled out in our polarimetric rendering equation 10. After estimating all parameters, we render the outgoing Stokes vectors using the pBRDF defined in Sec. A.

D. Datasets

D.1. Synthetic

Our synthetic dataset was rendered by Mitsuba 3 [7]. Following NeISF [9], we placed the object inside a modified Cornell Box [5]. Specifically, we set the roughness of the wall to small values (less than 0.1), so that light can bounce many times and becomes polarized before interacting with the object. The scene setup was designed to mimic our daily indoor scenes, where light bounces multiple times before interacting with the object. The object contains both dielectrics and conductors. For the dielectric part, we used *polarized plastic* pBRDF plugin as the material model. The refractive index was set to 1.5, and the specular coefficient was set to 1.0. The diffuse albedo and roughness were provided by texture maps. For the conductor part, we use the *rough conductor* pBRDF plugin as the material model. The specular coefficient was set to 1.0, and the complex reflective index was provided by some predefined reflective index for conductors like gold, sliver, and copper. For the geometry, we used objects from Stanford 3D scan dataset [4] and some open-sourced 3D objects. We used two light sources in our scene: one was the area light under the ceiling, and the other was the distant light provided by the environment map. For each object, we rendered 110 images. Each image contains a 9-channel Stokes vectors map, a 3-channel surface normal map, a 3-channel albedo map, a 1-channel roughness map, a 6-channel refractive index map, a 1-channel object mask, and a 1-channel conductor-dielectric mask. The cameras were uniformly distributed on the hemisphere around the object. We used 100 images for training and 10 images for testing. The resolution for the synthetic dataset is 700 $\times$ 700, and we rendered 4096 samples per pixel. Examples of our dataset can be found in Fig. 2.

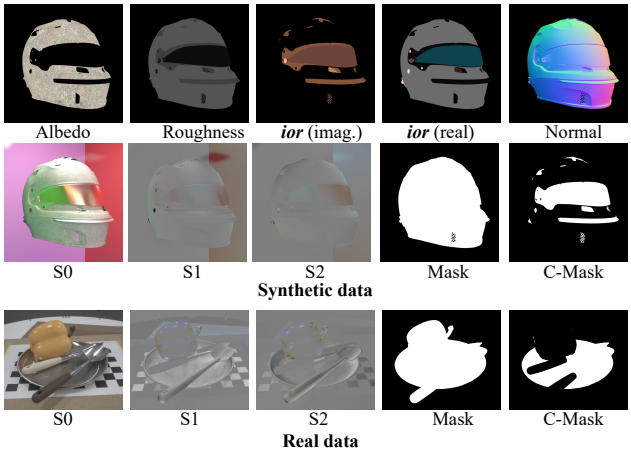

Figure 2. Examples of the proposed real and synthetic dataset.

D.2. Real-world

Because the capturing setting of real data is almost the same as NeISF [9], this section is heavily borrowed from it. Images of the real-world dataset were captured by a FLIR BFS-U3-51S5PC-C polarization camera with a Sony IMX250MYR sensor. For each viewpoint, we captured eight images with the exposure time [4, 8, 16, 32, 64, 128, 256, 512] ms. Then, we composite them to obtain one HDR image $\mathbf{I}_{\text{HDR}}$ in the raw image domain. We apply the alpha blending to minimize the noise in the composited image $\mathbf{I}_{\text{HDR}}$ as below.

$$I_{\text{HDR}} = \alpha \cdot g_{\text{short}} \cdot I_{\text{short}} + (1-\alpha) \cdot g_{\text{long}} \cdot I_{\text{long}}, \quad (46)$$

where I_{short} and I_{long} are intensities of a short-exposure and a long-exposure image, respectively. g_{short} and g_{long} are gain values to equalize the level of each image. These are calculated from the ratio of the exposure time. We get the optimal weight α by minimizing the noise variance σ^2_{HDR}. This can be represented using noise variances of two images, $\sigma^2_{\text{short}}, \sigma^2_{\text{long}}$,

$$\sigma^2_{\text{HDR}}(\alpha) = \alpha^2 \cdot g^2_{\text{short}} \cdot \sigma^2_{\text{short}} + (1-\alpha)^2 \cdot g^2_{\text{long}} \cdot \sigma^2_{\text{long}}. \quad (47)$$

Note that our main paper uses σ as density in Eq. (9). However, we denote σ^2 as noise variance in this section. The optimal weight $\hat{\alpha}$ that minimizes Eq. 47 can be simply obtained by solving:

$$\frac{\partial \sigma^2_{\text{HDR}}(\alpha)}{\partial \alpha} = 0. \quad (48)$$

The optimal weight $\hat{\alpha}$ is:

$$\hat{\alpha} = \underset{\alpha}{\text{argmin}}\ \sigma^2_{\text{HDR}}(\alpha) \quad (49)$$

$$= \frac{g^2_{\text{long}} \cdot \sigma^2_{\text{long}}}{g^2_{\text{short}} \cdot \sigma^2_{\text{short}} + g^2_{\text{long}} \cdot \sigma^2_{\text{long}}}. \quad (50)$$

Here, the variances of noise in the raw image, $\sigma^2_{\text{short}}, \sigma^2_{\text{long}}$, are assumed to follow a shot noise model where the noise variance has a linear relationship with the expected intensity μ [16],

$$\sigma^2 = a \cdot \mu + b. \quad (51)$$

Parameters a and b can be estimated by fitting the mean intensity and variance obtained from a series of raw images of a scene with different brightness to Eq. 51. Thus, we can estimate the noise variance of the raw image by substituting the intensity $\mathbf{I}$ to Eq. 51. We recursively apply Eq. 46 to expand to eight images' composition.

After demosaicing, we can get four polarized images with the polarization angle $[0°, 45°, 90°, 135°]$, and we denote them as $\hat{\mathbf{I}}_0, \hat{\mathbf{I}}_{45}, \hat{\mathbf{I}}_{90}, \hat{\mathbf{I}}_{135}$. The Stokes vectors $\hat{\mathbf{s}}$ can be calculated as follows:

$$\hat{\mathbf{s}}[0] = (\hat{\mathbf{I}}_0 + \hat{\mathbf{I}}_{45} + \hat{\mathbf{I}}_{90} + \hat{\mathbf{I}}_{135})/2, \quad (52)$$

$$\hat{\mathbf{s}}[1] = \hat{\mathbf{I}}_0 - \hat{\mathbf{I}}_{90}, \quad (53)$$

$$\hat{\mathbf{s}}[2] = \hat{\mathbf{I}}_{45} - \hat{\mathbf{I}}_{135}. \quad (54)$$

For each object, we took roughly 90 viewpoints for training and 2-4 viewpoints for evaluation. The resolution of the captured Stokes vectors is 1224 × 1024. Due to the limited computational resources, we re-scaled the resolution to 612 × 512 before training. In addition, we manually created an object mask and a conductor-dielectric mask using Photoshop [1] for each viewpoint. The camera poses were calculated by COLMAP [12, 13].

E. Additional Results

The ablation study of polarization cues To show the contribution of the polarization cues, we conduct experiments on an ablated model that does not have access to the polarization cues. We name this ablated model "Ours-unpol". Because this model can not access polarization cues, the geometry initialization stage is the original VolSDF [15]. For the joint optimization stage, f_{spec} and f_{dif} are removed. The outgoing intensity $\mathbf{s}[0]$ is rendered by the following equations:

$$\mathbf{s}[0] = \mathbf{s}_{\text{dif}}[0] + \mathbf{s}_{\text{spec}}[0] \quad (55)$$

$$\mathbf{s}_{\text{dif}}[0] = \int_\Omega \frac{\mathrm{a}}{\pi} T_o^+ T_i^+ \mathbf{s}^{\mathrm{r}}_{\text{dif}}[0] \cos\theta_i \, d\boldsymbol{\omega_i}, \quad (56)$$

$$\mathbf{s}_{\text{spec}}[0] = \int_\Omega \mathrm{k_s} \frac{DGR^+}{4\cos\theta_o} \mathbf{s}^{\mathrm{r}}_{\text{spec}}[0] \, d\boldsymbol{\omega_i}. \quad (57)$$

We show the geometry and material reconstruction results by comparing the synthetic data in Fig. 3 and real data in Fig. 4.

Novel view synthesis results Another important application of inverse rendering is novel view synthesis. In Fig. 5, we demonstrate our re-rendered results compared with views that were not shown to the model during training. It is observed that our method can reconstruct high-quality surface reflections even for glossy areas.

F. Implementation Details

F.1. Sphere sampling

The polarimetric rendering equations 2, 3 are solved by a fixed Fibonacci sphere sampling:

$$\mathbf{s} = \frac{2\pi}{|S_L|} \sum_{S_L} \mathbf{R}^{\text{cam}}_{\text{dif}} \cdot \mathbf{M}_{\text{dif}} \cdot \mathbf{s}^{\mathrm{r}}_{\text{dif}} + \mathbf{R}^{\text{cam}}_{\text{spec}} \cdot \mathbf{M}_{\text{spec}} \cdot \mathbf{s}^{\mathrm{r}}_{\text{spec}}, \quad (58)$$

where S_L is the set of the sampled incident light over the hemisphere.

F.2. Ray marching algorithm

The light-object interaction point is calculated using a ray marching algorithm. Given the ray origin r_o and the direction r_d, we iteratively march the ray to compute the interaction point $\mathbf{x}$ using Algo. 1. The number of steps is hard coded as 100 for both training and inference.

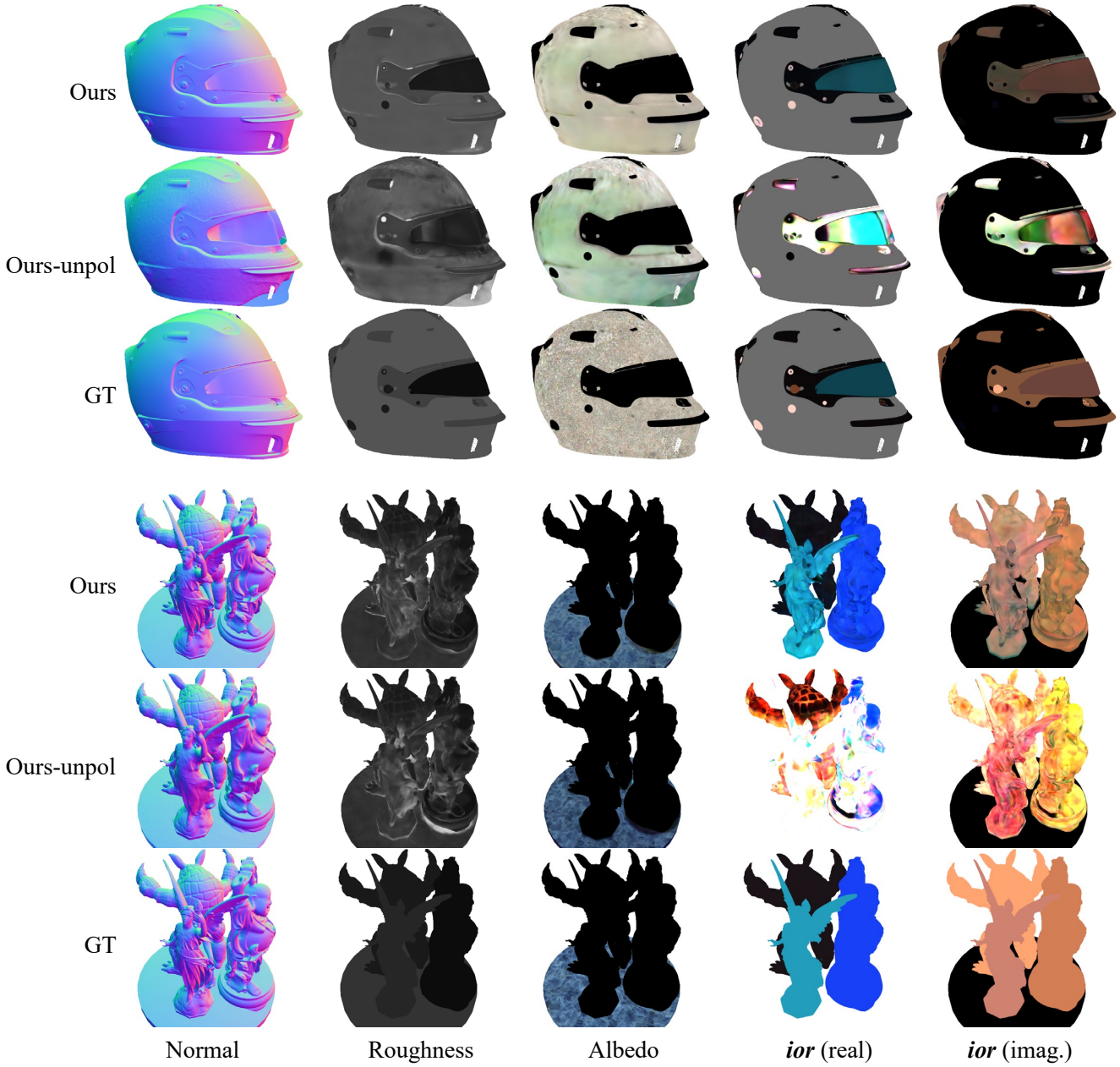

Figure 3. Ablation study of the contribution of polarization cues on the synthetic data. It is shown that without polarization cues, the accuracy of geometry and material reconstruction is much worse than our method.

F.3. Network architecture

We show the network architecture of the geometry initialization in Fig. 6, and the joint optimization in Fig. 7. The dimension of positional encoding [10] is set to 6 for all networks. The signed distance network f_{sdf} is directly taken from VolSDF [15]. The left networks are modified from NeILF++ [17]. The output dimension for f_{DoLP}, f_{oi}, f_{alb}, f_{ii}, f_{dif}, f_{eta}, f_{k} are 3, for f_{rough} is 1, and for f_{spec} is 6.

F.4. Training details

We define all pixels from the foreground are trained once as one epoch. The geometry initialization stage was trained for 20 epochs, and the joint optimization stage was also trained for 20 epochs. The batch size was set to 2,048, and we used Adam [8] optimizer with a learning rate set to 5e-4 and decayed exponentially to 5e-5. For the joint optimization stage, We clipped the gradient norm of the signed distance field net f_{sdf} with the maximum norm set

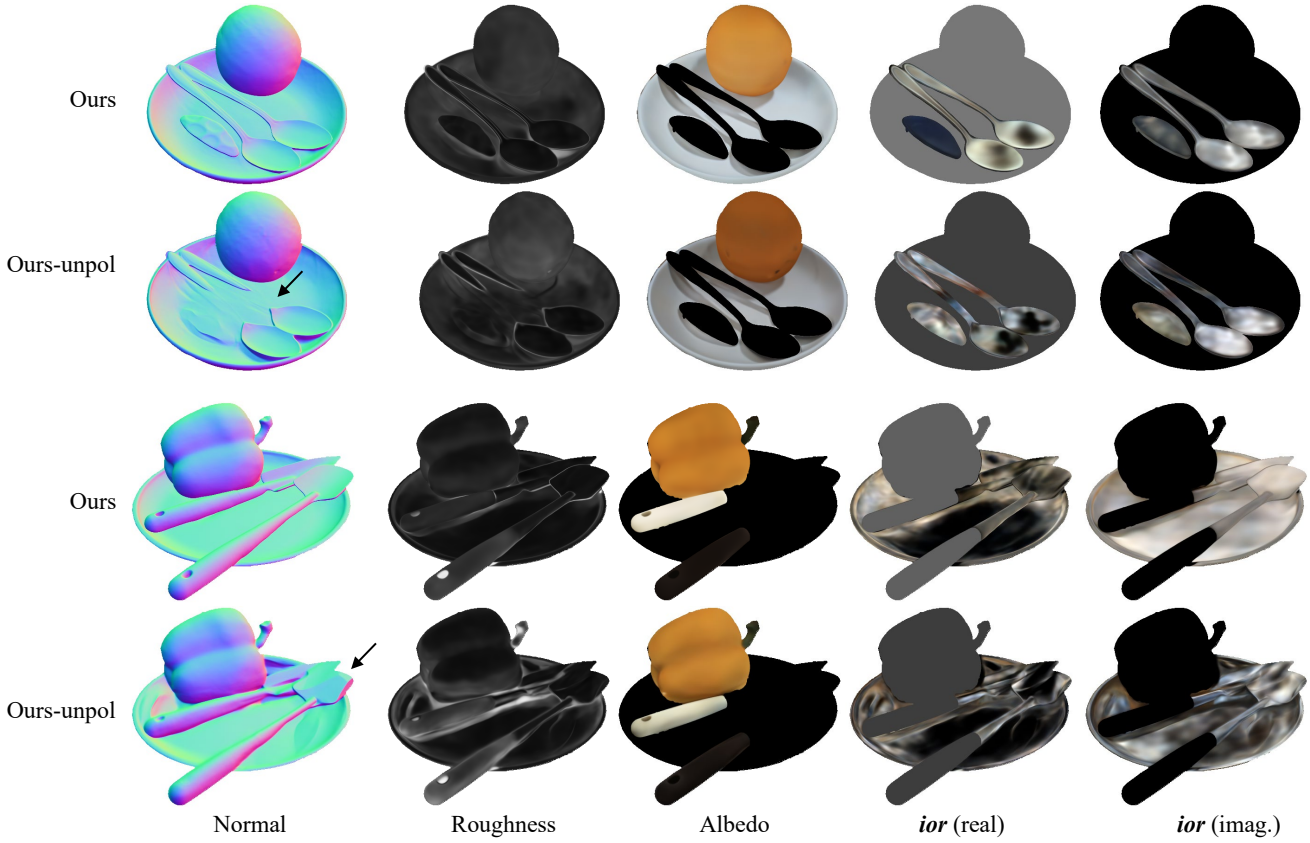

Figure 4. Ablation study of the contribution of polarization cues on the real data.

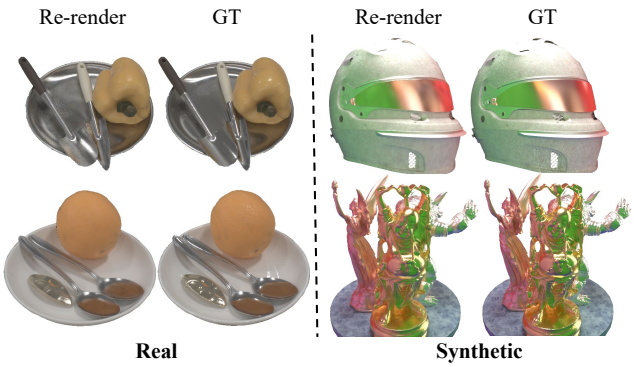

Figure 5. Novel view synthesis results.

to 0.1. The number of samples along a ray is set to 64. The number of sampled incident rays of each interaction point is 128. For the weights of the training losses. In the geometry initialization stage, we set the DoLP weight $\lambda_{\boldsymbol{\rho}} = 10$, the intensity weight $\lambda_{\mathbf{I}} = 0.5$, and the weight for the Eikonal regularization $\lambda_{\text{Eik}} = 0.1$. In the joint optimization stage, we set the DoLP weight $\lambda_{\boldsymbol{\rho}_s} = 5$, the Stokes vectors weight $\lambda_{\mathbf{s}} = 1$, and the weight for the Eikonal regularization $\lambda_{\text{Eik}} = 0.1$. However, due to the sensor's design,

Algorithm 1 Ray Marching Algorithm

$t \leftarrow 0$
for step $= 1$ **to** max_step **do**
  $\mathbf{x} \leftarrow r_o + t \cdot r_d$
  $t \leftarrow t + f_{\text{sdf}}(\mathbf{x})$
end for
$\mathbf{x} \leftarrow r_o + t \cdot r_d$
return $\mathbf{x}$

we observe a strong noise on the DoLP images for the real data, especially for the specular reflection. Because pixels in polarization sensors have a larger Bayer pattern than the conventional RGB sensor. Thus, we empirically reduce the DoLP weight $\lambda_{\boldsymbol{\rho}_s}$ to 0 or 0.5 for real-world data according to the noise level.

References

[1] Adobe Inc. Adobe photoshop. 5

[2] Seung-Hwan Baek, Daniel S Jeon, Xin Tong, and Min H Kim. Simultaneous acquisition of polarimetric svbrdf and normals. *ACM TOG*, 37(6):268–1, 2018. 2

[3] Edward Collett. Field guide to polarization. Spie Bellingham, 2005. 2

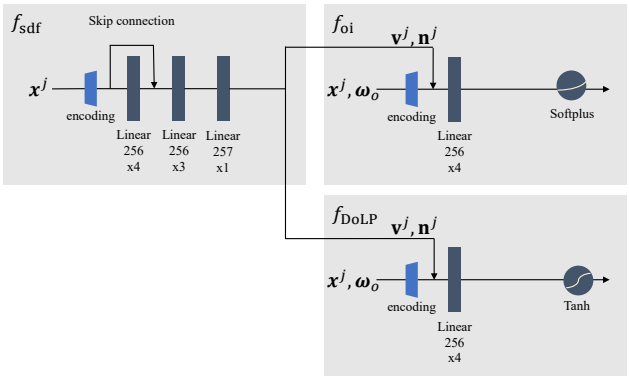

Figure 6. The network architecture of the geometry initialization stage.

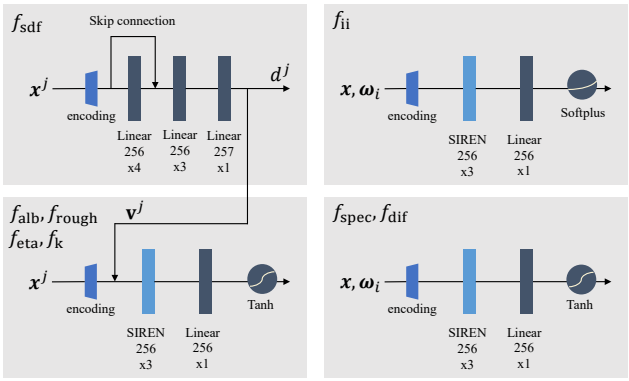

Figure 7. The network architecture of the joint optimization stage.

[4] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996. 4

[5] Cindy M Goral, Kenneth E Torrance, Donald P Greenberg, and Bennett Battaile. Modeling the interaction of light between diffuse surfaces. *SIGGRAPH*, 18(3):213–222, 1984. 4

[6] Eric Heitz. Understanding the masking-shadowing function in microfacet-based brdfs. *Journal of Computer Graphics Techniques*, 3(2):32–91, 2014. 2

[7] Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, and Delio Vicini. Dr.jit: A just-in-time compiler for differentiable rendering. *ACM TOG*, 41(4), 2022. 2, 3, 4

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[9] Chenhao Li, Taishi Ono, Takeshi Uemori, Hajime Mihara, Alexander Gatto, Hajime Nagahara, and Yusuke Moriuchi. Neisf: Neural incident stokes field for geometry and material estimation. In *CVPR*, pages 21434–21445, 2024. 4

[10] 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 *ECCV*, pages 405–421, 2020. 3, 6

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035. 2019. 2

[12] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. 5

[13] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, pages 501–518, 2016. 5

[14] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 195–206, 2007. 2

[15] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *NeurIPS*, 34:4805–4815, 2021. 3, 5, 6

[16] Masakazu Yoshimura, Junji Otsuka, Atsushi Irie, and Takeshi Ohashi. Rawgment: noise-accounted raw augmentation enables recognition in a wide variety of environments. In *CVPR*, pages 14007–14017, 2023. 5

[17] Jingyang Zhang, Yao Yao, Shiwei Li, Jingbo Liu, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. Neilf++: Inter-reflectable light fields for geometry and material estimation. In *ICCV*, pages 3601–3610, 2023. 6