MetaLayer: A Meta-learned BSDF Model for Layered Materials –Supplemental Document–

JIE GUO*, State Key Lab for Novel Software Technology, Nanjing University, China
ZERU LI*, State Key Lab for Novel Software Technology, Nanjing University, China
XUEYAN HE, State Key Lab for Novel Software Technology, Nanjing University, China
BEIBEI WANG, Nankai University and Nanjing University of Science and Technology, China
WENBIN LI, State Key Lab for Novel Software Technology, Nanjing University, China
YANWEN GUO†, State Key Lab for Novel Software Technology, Nanjing University, China
LING-QI YAN, University of California, Santa Barbara, United States of America

1 GRADIENT BACK-PROPAGATION FROM BSDFNET TO METANET

In this section, we provide the detailed analysis for gradient back-propagation from BSDFNet to MetaNet.

Let $\boldsymbol{\delta}_{i+1}$ be the input of the $i+1$-th layer. Then, we have

$$\boldsymbol{\delta}^{+}_{i+1} = \mathbf{W}^{+}_{i}(\mathbf{v}^{*}_{i} \oplus \mathbf{v}^{+}_{i} \oplus \mathbf{v}^{\lambda}_{i}) + \mathbf{b}^{+}_{i} \tag{1}$$

$$\boldsymbol{\delta}^{\lambda}_{i+1} = \mathbf{W}^{*}_{i}(\mathbf{v}^{+}_{i} \oplus \mathbf{v}^{\lambda}_{i}) + \mathbf{b}^{*}_{i} \tag{2}$$

and

$$\mathbf{v}^{+}_{i+1} = a(\boldsymbol{\delta}^{+}_{i+1}) \tag{3}$$

$$\mathbf{v}^{\lambda}_{i+1} = a(\boldsymbol{\delta}^{\lambda}_{i+1}) \tag{4}$$

where a is the activation function. We further rephrase $\boldsymbol{\delta}^{+}_{i+1}$ and $\boldsymbol{\delta}^{\lambda}_{i+1}$ as

$$\boldsymbol{\delta}^{+}_{i+1} = \mathbf{W}^{+*}_{i}\mathbf{v}^{*}_{i} + \mathbf{W}^{++}_{i}\mathbf{v}^{+}_{i} + \mathbf{W}^{+\lambda}_{i}\mathbf{v}^{\lambda}_{i} + \mathbf{b}^{+}_{i} \tag{5}$$

$$\boldsymbol{\delta}^{\lambda}_{i+1} = \mathbf{W}^{*+}_{i}\mathbf{v}^{+}_{i} + \mathbf{W}^{*\lambda}_{i}\mathbf{v}^{\lambda}_{i} + \mathbf{b}^{*}_{i} \tag{6}$$

via splitting matrix $\mathbf{W}^{+}_{i}$ into three submatrices: $\mathbf{W}^{+*}_{i}$, $\mathbf{W}^{++}_{i}$ and $\mathbf{W}^{+\lambda}_{i}$; splitting matrix $\mathbf{W}^{\lambda}_{i+1}$ into two submatrices: $\mathbf{W}^{*+}_{i}$ and $\mathbf{W}^{*\lambda}_{i}$. The splitting is performed along the row of each matrix and ensures that all matrix-vector multiplications in the above equations are valid.

Suppose we calculate the L2 loss between the prediction $\mathbf{v}_L$[1] and the ground truth $\mathbf{y}$, i.e.,

$$\mathcal{L} = \frac{1}{2}(\mathbf{v}_L - \mathbf{y})^2 = \frac{1}{2}\left(a(\boldsymbol{\delta}^{\lambda}_{L}) - \mathbf{y}\right)^2 \tag{7}$$

where L denotes the last layer of BSDFNet. Then, the gradients of $\mathcal{L}$ w.r.t. the input of the final layer are

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}^{\lambda}_{L}} = a'(\boldsymbol{\delta}^{\lambda}_{L})(a(\boldsymbol{\delta}^{\lambda}_{L}) - \mathbf{y}) \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}^{+}_{L}} = 0. \tag{8}$$

For the gradients of $\mathcal{L}$ w.r.t. the input of the i-th layer, we have the following two recursive formulas:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}^{\lambda}_{i}} &= \left(\frac{\partial \boldsymbol{\delta}^{\lambda}_{i+1}}{\partial \boldsymbol{\delta}^{\lambda}_{i}}\right)^{\top} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}^{\lambda}_{i+1}} + \left(\frac{\partial \boldsymbol{\delta}^{+}_{i+1}}{\partial \boldsymbol{\delta}^{\lambda}_{i}}\right)^{\top} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}^{+}_{i+1}} \\ &= \left[\mathbf{W}^{*\lambda}_{i}\mathrm{diag}(a'(\boldsymbol{\delta}^{\lambda}_{i}))\right]^{\top} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}^{\lambda}_{i+1}} + \left[\mathbf{W}^{+\lambda}_{i}\mathrm{diag}(a'(\boldsymbol{\delta}^{\lambda}_{i}))\right]^{\top} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}^{+}_{i+1}}\end{aligned} \tag{9}$$

and

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}^{+}_{i}} &= \left(\frac{\partial \boldsymbol{\delta}^{\lambda}_{i+1}}{\partial \boldsymbol{\delta}^{+}_{i}}\right)^{\top} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}^{\lambda}_{i+1}} + \left(\frac{\partial \boldsymbol{\delta}^{+}_{i+1}}{\partial \boldsymbol{\delta}^{+}_{i}}\right)^{\top} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}^{+}_{i+1}} \\ &\left[\mathbf{W}^{*+}_{i}\mathrm{diag}(a'(\boldsymbol{\delta}^{+}_{i}))\right]^{\top} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}^{\lambda}_{i+1}} + \left[\mathbf{W}^{++}_{i}\mathrm{diag}(a'(\boldsymbol{\delta}^{+}_{i}))\right]^{\top} \frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}^{+}_{i+1}}.\end{aligned} \tag{10}$$

With Eqs. 8-10, we can get the gradients of $\mathcal{L}$ w.r.t. the input of any layer in BSDFNet. These gradients are expected to be back-propagated to MetaNet.

*Joint first authors.
†Corresponding author.

Authors' addresses: Jie Guo, State Key Lab for Novel Software Technology, Nanjing University, China, guojie@nju.edu.cn; Zeru Li, State Key Lab for Novel Software Technology, Nanjing University, China, lzr@smail.nju.edu.cn; Xueyan He, State Key Lab for Novel Software Technology, Nanjing University, China, xueyanhe@smail.nju.edu.cn; Beibei Wang, Nankai University and Nanjing University of Science and Technology, China, beibei.wang@njust.edu.cn; Wenbin Li, State Key Lab for Novel Software Technology, Nanjing University, China, liwenbin@nju.edu.cn; Yanwen Guo, State Key Lab for Novel Software Technology, Nanjing University, China, ywguo@nju.edu.cn; Ling-Qi Yan, lingqi@cs.ucsb.edu, University of California, Santa Barbara, Santa Barbara, United States of America.

0730-0301/2023/9-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

[1] $\mathbf{v}_L$ is a scalar (a single channel reflectance) in our implementation.

Now, let us derive the gradient of $\mathcal{L}$ w.r.t. the parameters of MetaNet Θ_M. Since $M(\Gamma, \Theta_M) = \{\mathbf{v}^*, \mathbf{W}^*, \mathbf{b}^*\}$, we have

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \Theta_M} &= \sum_i \left(\left(\frac{\partial \mathbf{v}_i^*}{\partial \Theta_M} \right)^\top \frac{\partial \mathcal{L}}{\partial \mathbf{v}_i^*} + \left(\frac{\partial \mathrm{vec}\mathbf{W}_i^*}{\partial \Theta_M} \right)^\top \frac{\partial \mathcal{L}}{\partial \mathrm{vec}\mathbf{W}_i^*} + \left(\frac{\partial \mathbf{b}_i^*}{\partial \Theta_M} \right)^\top \frac{\partial \mathcal{L}}{\partial \mathbf{b}_i^*} \right) \\
&= \sum_i \left(\left(\frac{\partial \mathbf{v}_i^*}{\partial \Theta_M} \right)^\top \frac{\partial \mathcal{L}}{\partial \mathbf{v}_i^*} + \left(\frac{\partial \mathrm{vec}\mathbf{W}_i^{*+}}{\partial \Theta_M} \right)^\top \frac{\partial \mathcal{L}}{\partial \mathrm{vec}\mathbf{W}_i^{*+}} \right. \\
&\quad \left. + \left(\frac{\partial \mathrm{vec}\mathbf{W}_i^{*\wr}}{\partial \Theta_M} \right)^\top \frac{\partial \mathcal{L}}{\partial \mathrm{vec}\mathbf{W}_i^{*\wr}} + \left(\frac{\partial \mathbf{b}_i^*}{\partial \Theta_M} \right)^\top \frac{\partial \mathcal{L}}{\partial \mathbf{b}_i^*} \right) \\
&= \sum_i \left(\left(\frac{\partial \mathbf{v}_i^*}{\partial \Theta_M} \right)^\top (\mathbf{W}_i^{+*})^\top \underbrace{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}_i^+}}_{\triangle} \right. \\
&\quad + \left(\frac{\partial \mathrm{vec}\mathbf{W}_i^{*+}}{\partial \Theta_M} \right)^\top \mathrm{vec}\left(\underbrace{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}_{i+1}^+}}_{\triangle} (a(\boldsymbol{\delta}_i)^+)^\top \right) \\
&\quad \left. + \left(\frac{\partial \mathrm{vec}\mathbf{W}_i^{*\wr}}{\partial \Theta_M} \right)^\top \mathrm{vec}\left(\underbrace{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}_{i+1}^\wr}}_{\triangle} (a(\boldsymbol{\delta}_i)^\wr)^\top \right) + \left(\frac{\partial \mathbf{b}_i^*}{\partial \Theta_M} \right)^\top \underbrace{\frac{\partial \mathcal{L}}{\partial \boldsymbol{\delta}_{i+1}^\wr}}_{\triangle} \right).
\end{aligned} \tag{11}$$

Note that the terms marked by $\triangle$ in the above equation are gradients back-propagated from BSDFNet to MetaNet. We also note that activation functions appear in the above equation which may approach zero (i.e., $a(\boldsymbol{\delta}_i)^+) = 0$ and $a(\boldsymbol{\delta}_i)^\wr) = 0$) when ReLU is adopted. In this case, the impact of loss will be lowered, leading to slow convergence for MetaNet since $\partial\mathcal{L}/\partial\Theta_M$ may approach zero. With the directly predicted neurons $\mathbf{v}_i^*$ (in the first term of the above equation's right-hand side), the value of $\partial\mathcal{L}/\partial\Theta_M$ can be significantly improved, enabling fast and stable convergence.

2 MORE VALIDATION OF MATERIAL EDITING

To validate the ability of material editing, we conduct a comprehensive analysis of our model's performance on the full set of parameterized materials (including layered materials beyond the scope of the training set). To generate a new layered material, we directly edit the material's physical parameters, including the surface properties of two interfaces (the surface roughness: α_1 and α_2, the relative index of refraction (IOR): η_1 and η_2) and the scattering properties of the internal medium (the extinction coefficient: σ_t and the single-scattering albedo: ρ). Pairwise comparisons of our results and the corresponding reference images generated by Guo et al.'s method [Guo et al. 2018] are provided in Figs. 1 to 10.

3 MORE COMPARISONS TO PRIOR WORK

In Fig. 11, we provide more comparisons against NBRDF [Sztrajman et al. 2021] and an extended version of NBRDF (NBRDF+). For a fair comparison, NBRDF+ has the same number of trainable parameters as our BSDFNet. The error maps are generated by evaluating the per-pixel errors with respect to the reference images which are generated by Guo et al.'s bidirectional method [Guo et al. 2018].

4 MORE COMPARISONS OF DIFFERENT POSITIONAL ENCODING METHODS

In Fig. 12, we provide more rendering results from different positional encoding methods. We compare two different groups of coordinates: $(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$ and $(\boldsymbol{\omega}_h, \boldsymbol{\omega}_d)$. Compared with no encoding and traditional sinusoidal encoding, our Rusinkiewicz spherical harmonics encoding shows obvious advantage in preserving high-frequency angular details.

REFERENCES

Yu Guo, Miloš Hašan, and Shuang Zhao. 2018. Position-Free Monte Carlo Simulation for Arbitrary Layered BSDFs. *ACM Trans. Graph.* 37, 6, Article 279 (dec 2018), 14 pages.

Alejandro Sztrajman, Gilles Rainer, Tobias Ritschel, and Tim Weyrich. 2021. Neural BRDF Representation and Importance Sampling. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 332–346.

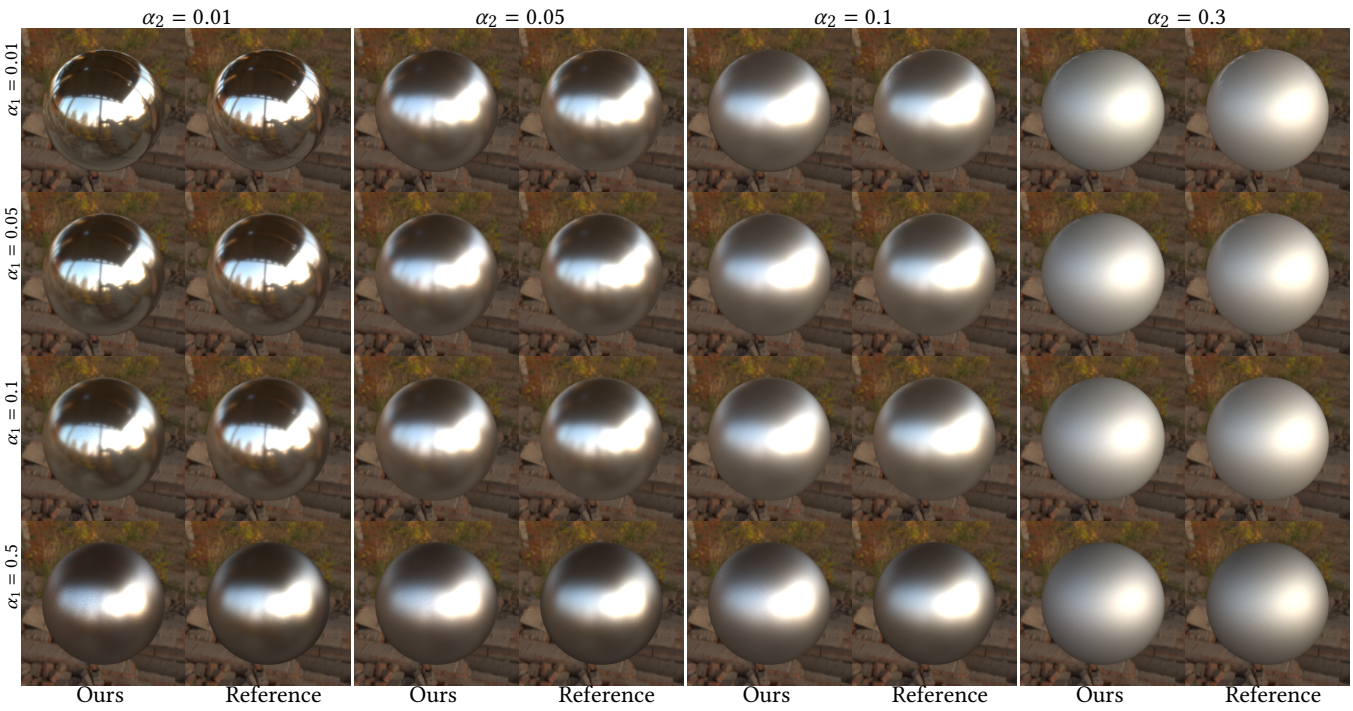

Fig. 1. Editing the surface roughness: α_1 and α_2. The IOR of the bottom conductive surface is set by the value of silver (Ag).

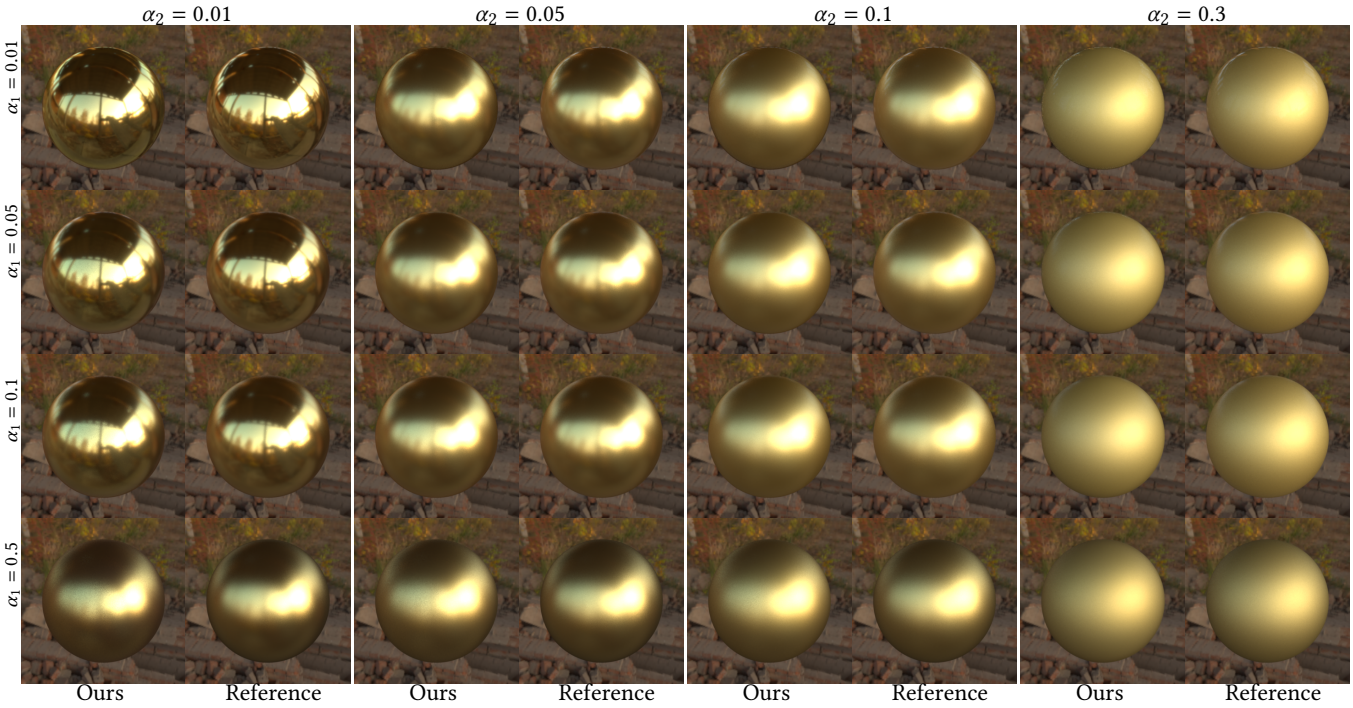

Fig. 2. Editing the surface roughness: α_1 and α_2. The IOR of the bottom conductive surface is set by the value of gold (Au).

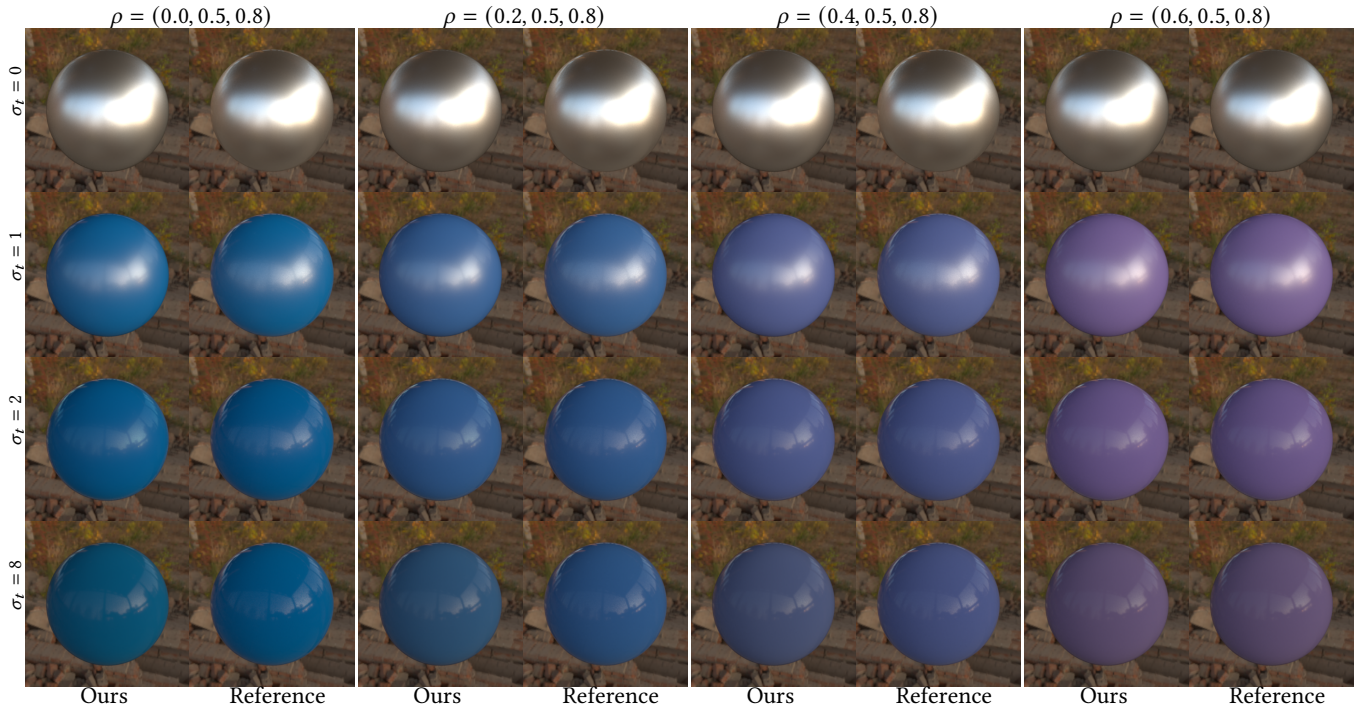

Fig. 3. Editing the extinction coefficient: σ_t and the single-scattering albedo: ρ. The IOR of the bottom conductive surface is set by the value of silver (Ag) and α_1 is set as 0.01.

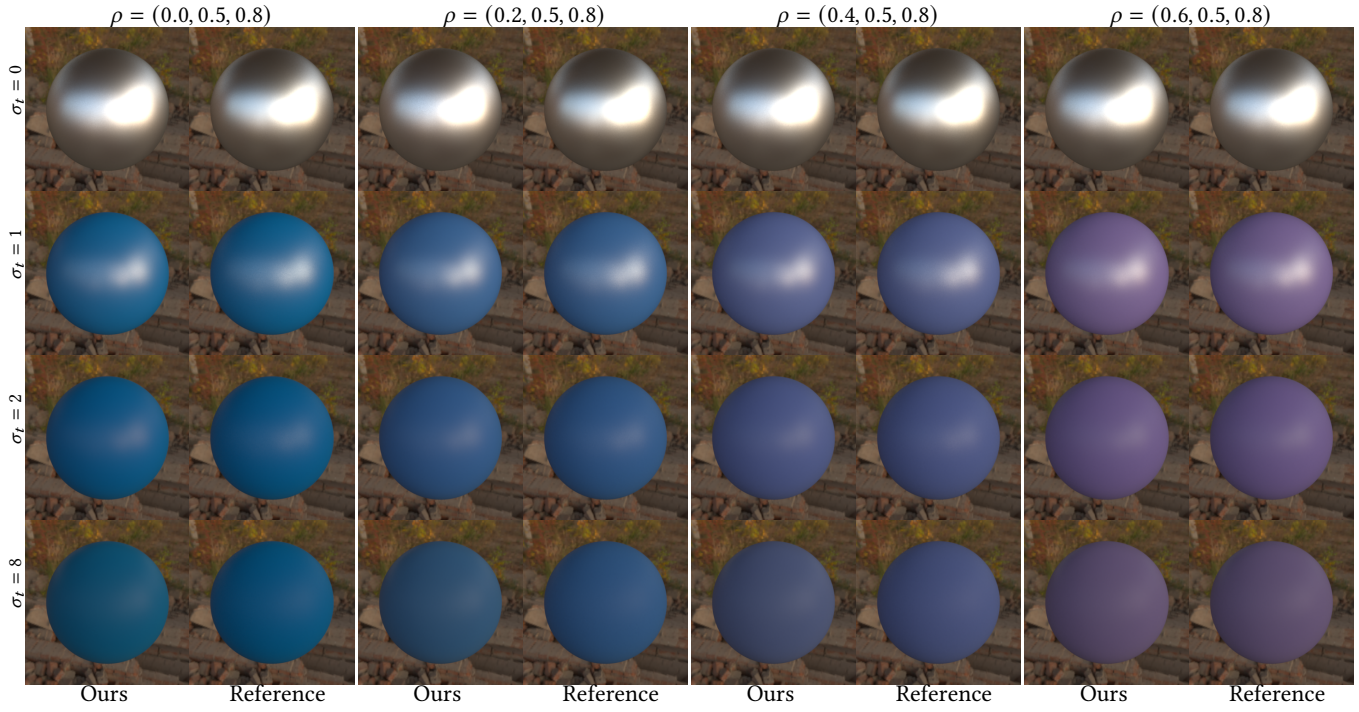

Fig. 4. Editing the extinction coefficient: σ_t and the single-scattering albedo: ρ. The IOR of the bottom conductive surface is set by the value of silver (Ag) and α_1 is set as 0.3.

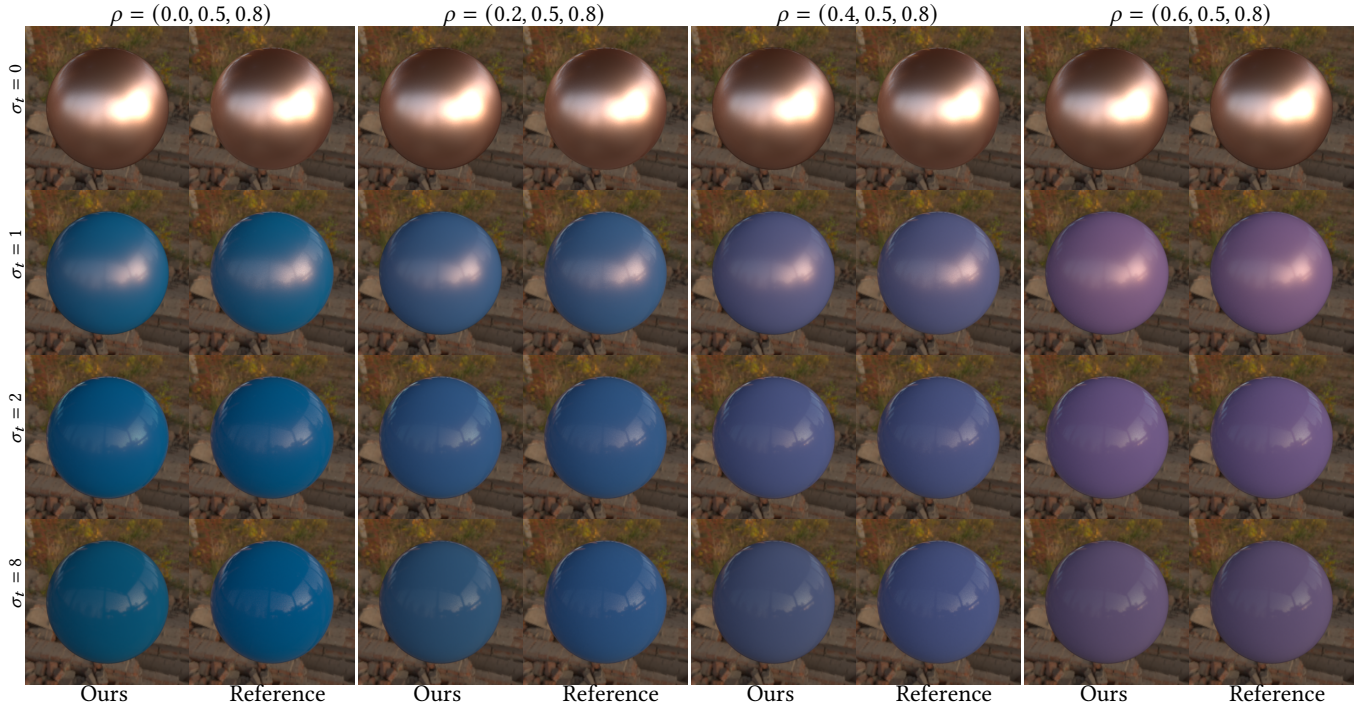

Fig. 5. Editing the extinction coefficient: σ_t and the single-scattering albedo: ρ. The IOR of the bottom conductive surface is set by the value of copper (Cu) and α_1 is set as 0.01.

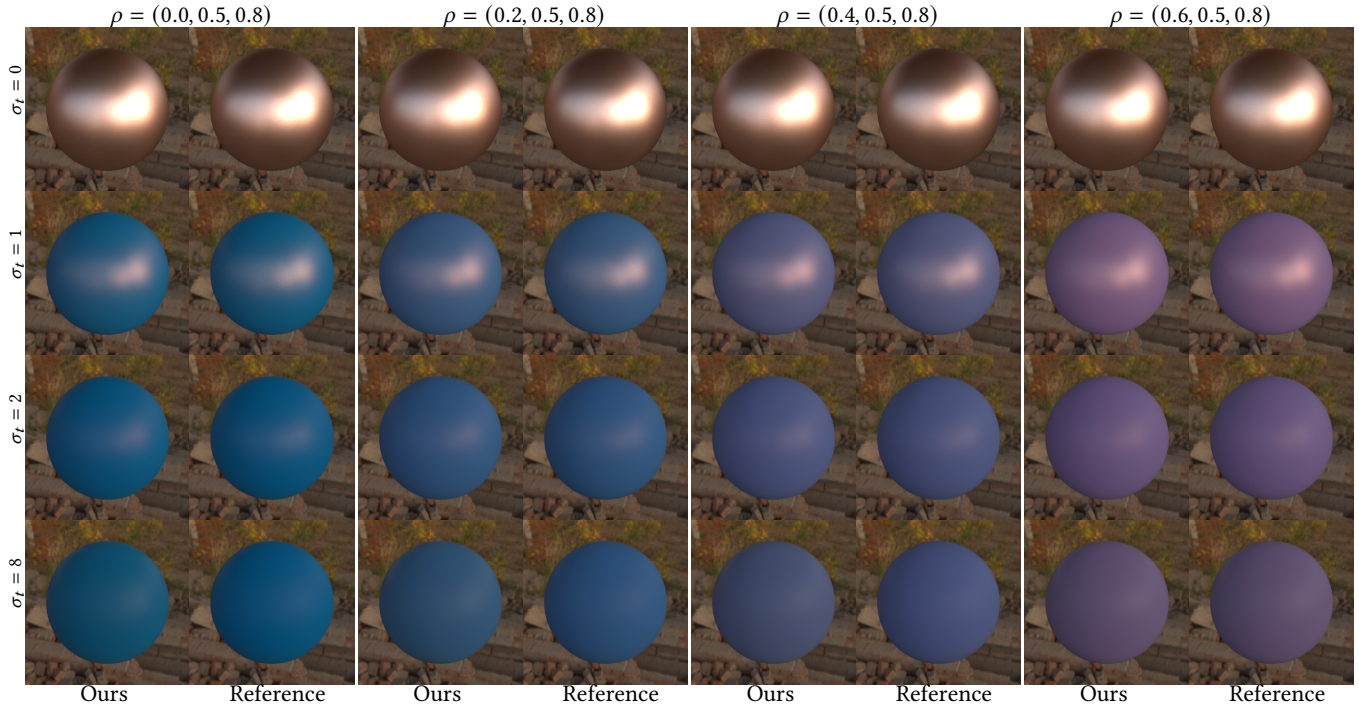

Fig. 6. Editing the extinction coefficient: σ_t and the single-scattering albedo: ρ. The IOR of the bottom conductive surface is set by the value of copper (Cu) and α_1 is set as 0.3.

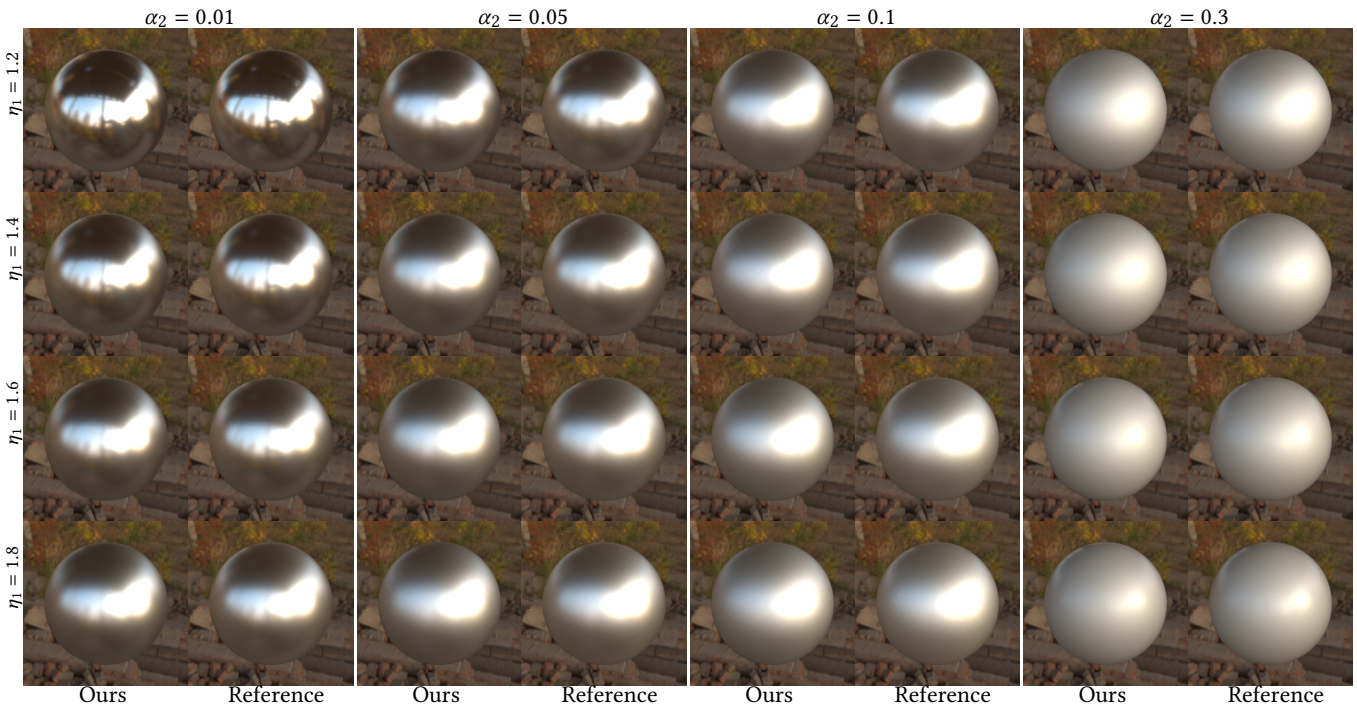

Fig. 7. Editing the roughness α_2 and IOR η_1. The IOR of the bottom conductive surface is set by the value of silver (Ag) and $\sigma_t = 0$, $\alpha_1 = 0.1$.

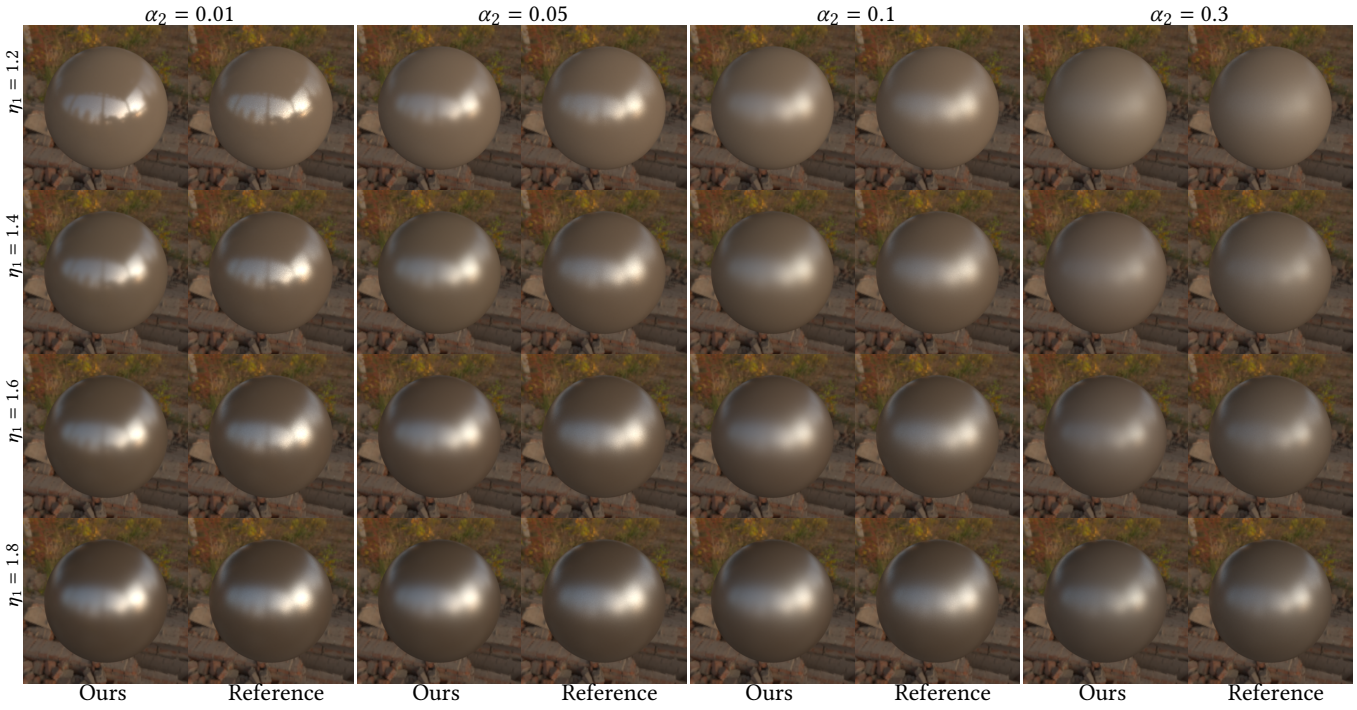

Fig. 8. Editing the roughness α_2 and IOR η_1. The IOR of the bottom conductive surface is set by the value of silver (Ag) and $\sigma_t = 1$, $\alpha_1 = 0.1$.

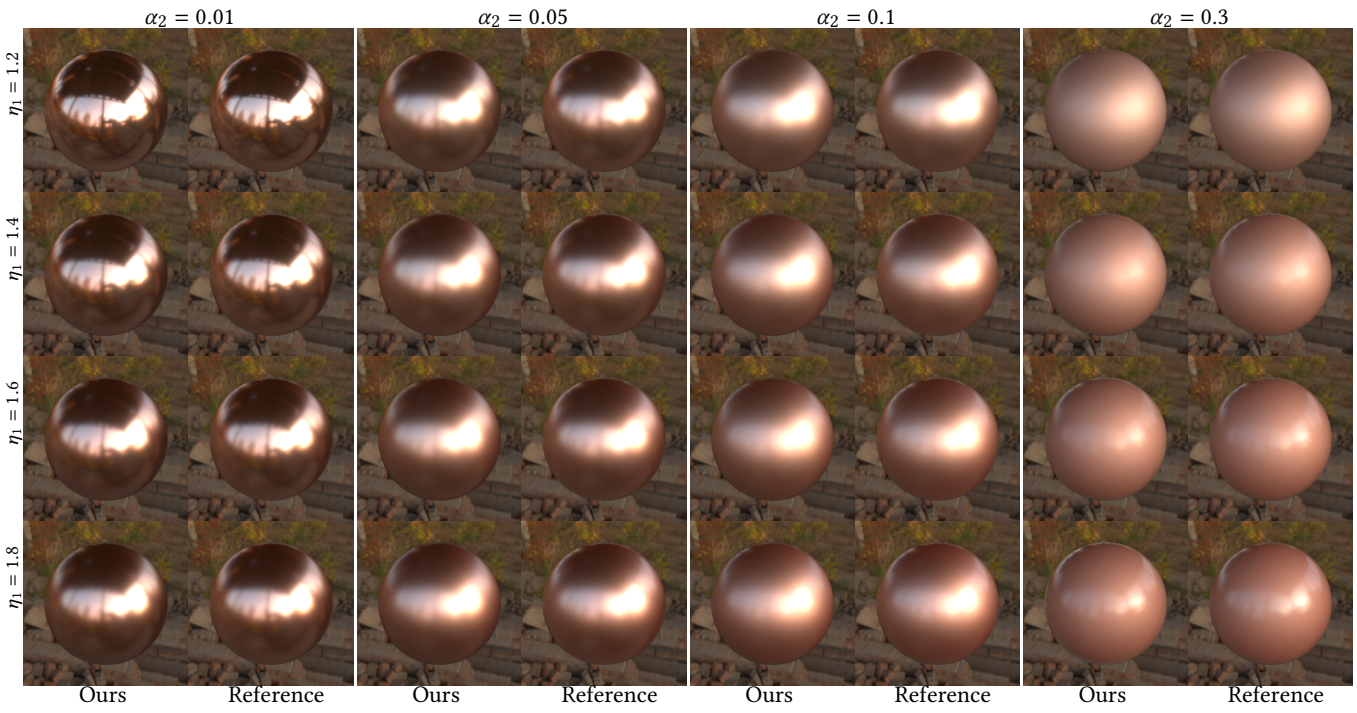

Fig. 9. Editing the roughness α_2 and IOR η_1. The IOR of the bottom conductive surface is set by the value of copper (Cu) and $\sigma_t = 0$, $\alpha_1 = 0.05$.

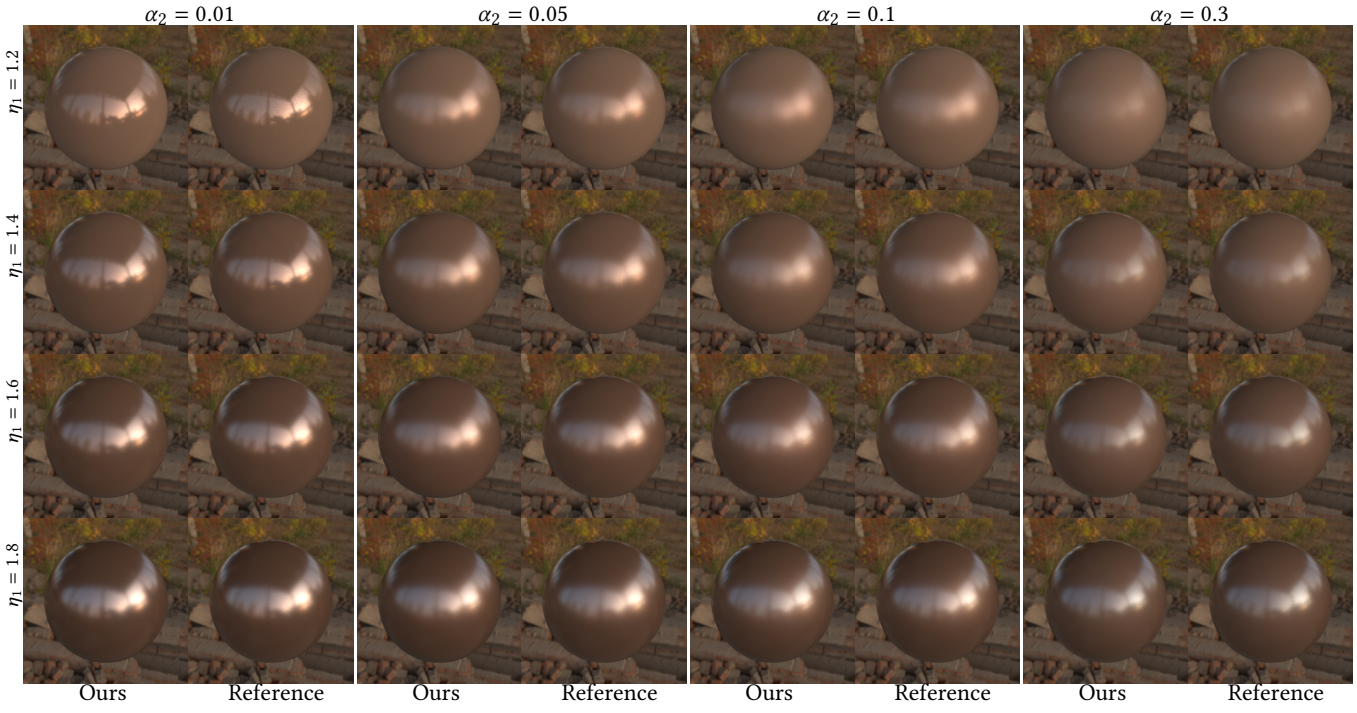

Fig. 10. Editing the roughness α_2 and IOR η_1. The IOR of the bottom conductive surface is set by the value of copper (Cu) and $\sigma_t = 1$, $\alpha_1 = 0.05$.

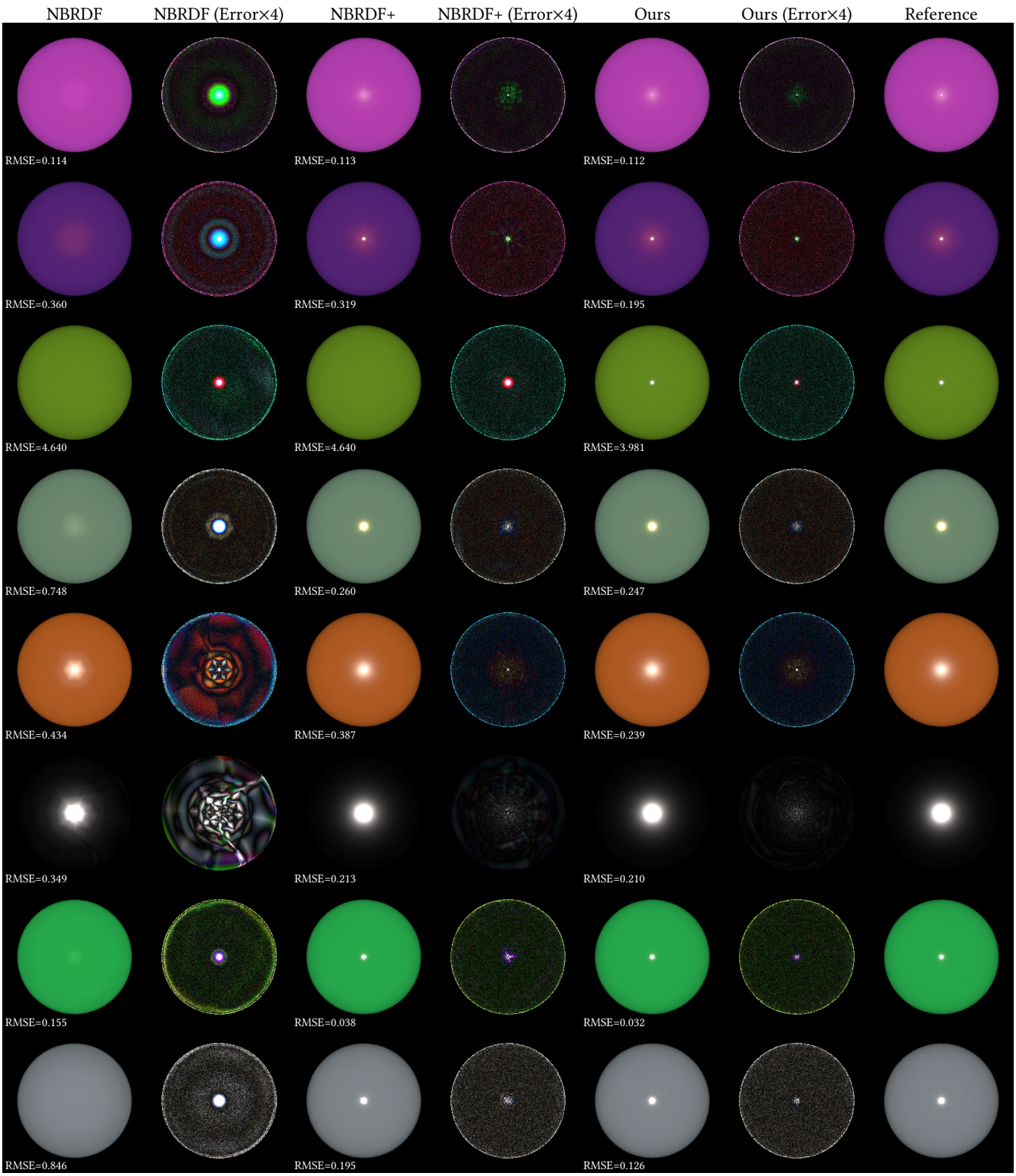

Fig. 11. Comparison against NBRDF [Sztrajman et al. 2021] and an extended version of NBRDF (NBRDF+). RMSEs (root mean square errors) are provided for each method in comparison.

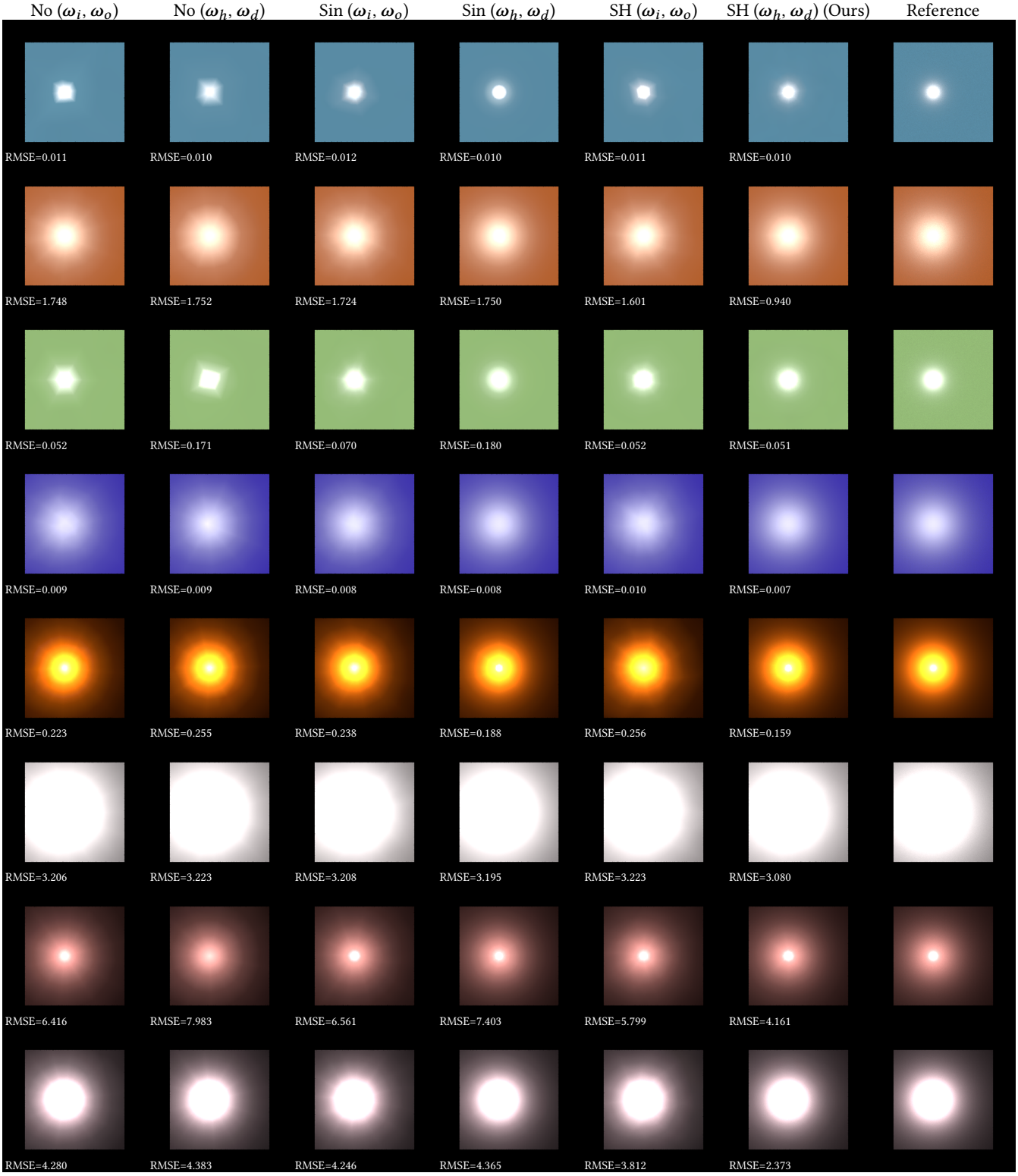

Fig. 12. Comparison of different positional encoding methods. No=no encoding, Sin=sinusoidal encoding, and SH = spherical harmonics encoding.