Neural-PBIR Reconstruction of Shape, Material, and Illumination —Supplementary Material—

1. Technical Details

In what follows, we elaborate technical details of our Neural-PBIR pipeline's three main stages. Code will be released upon internal approval for future extension and reproduction.

1.1. Neural Surface Reconstruction

Sharpness term in unbiased volume rendering. Following NeuS [7], we use a scaled sigmoid σ_s function in the SDF for alpha activation:

$$\alpha_i = \max\left(0,\ \frac{\sigma_s(S(\boldsymbol{x}_i)) - \sigma_s(S(\boldsymbol{x}_{i+1}))}{\sigma_s(S(\boldsymbol{x}_i))}\right), \quad (1)$$

where:

- S is the signed-distance function;
- $\sigma_s(y) = (1+\exp(-sy))^{-1}$ with $s > 0$ being the *sharpness term*;
- $\{\boldsymbol{x}_i = \boldsymbol{o} + t_i\boldsymbol{v}\}_{i=1}^N$ (with $0 < t_1 < t_2 < \ldots < t_N$) are the N sampled points on the camera ray originated at the camera's location $\boldsymbol{o}$ with viewing direction $\boldsymbol{v}$.

Specifically, we start with $s = 30$ if foreground masks are provided (*e.g.*, for the synthetic and DTU datasets) and $s = 5$ otherwise (*e.g.* for our measured real-world dataset). In practice, we use a *scheduled* sharpness s (instead of updating s with gradient descent) as we find it more stable. Then, we update the sharpness s by setting $s \leftarrow \min(s + 0.02,\ 300)$ after each iteration.

Background modeling. As stated in Sec. 3.1 of the main paper, we use two sets of $V^{(\mathrm{sdf})}$ and $V^{(\mathrm{feat})}$ grids to model the foreground and the background (via Eq. (3) in the main paper), respectively. Specifically, the foreground region is defined as the volume inside a (predetermined) small bounding box. The background, on the other hand, is the volume inside a much larger bounding box.[1] Given a camera ray, we categorize the sample points $\{\boldsymbol{x}_i = \boldsymbol{o} + t_i\boldsymbol{v}\}_{i=1}^N$ along the ray as foreground or background and then evaluate signed distance $S(\boldsymbol{x}_i)$ and radiance $L_\mathrm{o}(\boldsymbol{x}_i, -\boldsymbol{v})$ for each $\boldsymbol{x}_i$ using the corresponding grids.

Thanks to the background scene volume, our method can work without external mask supervision (e.g., our own real-world dataset).

Points sampling on rays. When sampling 3D points $\{\boldsymbol{x}_i = \boldsymbol{o} + t_i\boldsymbol{v}\}_{i=1}^N$ along a camera ray, we use $t_i = i\Delta t$ with Δt being half the size of a grid voxel for all $i = 1, 2, \ldots, N$.

Coarse-to-fine optimization. For better efficiency and more coherent results, when optimizing the $V^{(\mathrm{sdf})}$ and $V^{(\mathrm{feat})}$ grids, we leverage a coarse-to-fine scheme by doubling the number of voxels every 1k iterations for the first 10k iterations. The final voxel resolutions are 300^3 for the foreground grids (which contain the object of interest) and 160^3 for the background ones.

Optimization details. We optimizing the $V^{(\mathrm{sdf})}$ and $V^{(\mathrm{feat})}$ grids for the foreground and the background jointly using the Adam [5] method with $\beta = (0.9, 0.99)$ and $\epsilon = 10^{-12}$ in 20k iterations. When computing the loss, we use weights $w_\mathrm{lap} = 10^{-8}$ and $w_\mathrm{pp\_rgb} = 0.01$. Also, when using the running means to update the threshold t in the adaptive Huber loss, we set the momentum to 0.99 and clamp t to a minimum of 0.01.

When training the SDF grids $V^{(\mathrm{sdf})}$, we use an initial learning rate of 0.01 that then decays to 0.001 at 10k iterations. When training the outgoing radiance field L_o, we use a learning rate of 0.001 for the MLPs and 0.1 for the feature grids $V^{(\mathrm{feat})}$.

1.2. Neural Distillation of Material and Lighting

Initialization. We initialize the roughnesses to $M_\mathrm{r}[v] = 0.25$ for each vertex v. For per-vertex albedo, we initialize $M_\mathrm{a}[v]$ to the median of the outgoing radiance from the

[1] In practice, we use background bounding boxes that are $16\times$ as large as the foreground ones.

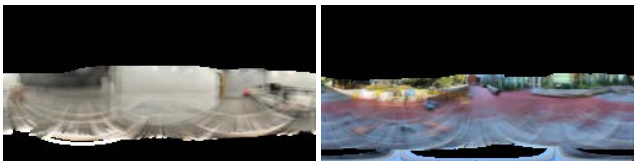


Figure 1: **Examples of the averaged background observation.** The black region indicates missing observation.

teacher model L_{o}:

$$M_{\mathrm{a}}[v] = \mathrm{Median}\left\{L_{\mathrm{o}}(\boldsymbol{x}[v], \boldsymbol{\omega}_{\mathrm{o}}) \mid \boldsymbol{\omega}_{\mathrm{o}} \in \Omega,\ (\boldsymbol{\omega}_{\mathrm{o}} \cdot M_{\mathrm{n}}[v]) > 0\right\}, \quad (2)$$

where $\boldsymbol{x}[v]$ and $M_{\mathrm{n}}[v]$ indicate, respectively, the position and the normal of vertex v, and Ω is the predetermined set of outgoing directions.

Fresnel term. In addition to albedo and roughness, we also need the Fresnel term F_0 [4] to model specular reflection. Following MII, we assume the object to be reconstructed is dielectric and make F_0 constant. We set $F_0 = 0.02$ for all synthetic data since it is used by MII's open-source implementation, and $F_0 = 0.04$ for real-world data since it is the industrial standard.

Averaged background constraint. Recap that we regularize our SG-based illumination $L_{\mathrm{env}}^{\mathrm{SG}}$ to be similar to the averaged background observation $(L_{\mathrm{env}}^{\mathrm{SG}})'$. We now detail how the latter is obtained.

First, we gather all "background" training pixels (toward which the camera rays miss our reconstructed mesh). Then, we compute $(L_{\mathrm{env}}^{\mathrm{SG}})'$ as an environment map under the latitude-and-longitude representation as follows. For each background pixel with intensity I and viewing direction $\boldsymbol{v}$, we set the value of the corresponding pixel j in the environment map $(L_{\mathrm{env}}^{\mathrm{SG}})'$—based on latitude and longitude coordinates of $\boldsymbol{v}$—as $(L_{\mathrm{env}}^{\mathrm{SG}})'[j] = I$. When multiple pixels (from different camera locations) contribute to one pixel j of $(L_{\mathrm{env}}^{\mathrm{SG}})'$, we set $(L_{\mathrm{env}}^{\mathrm{SG}})'[j]$ using the average intensity of all such pixels.

We show some examples of the averaged background observations $(L_{\mathrm{env}}^{\mathrm{SG}})'$ in Fig. 1. We only compute the regularization loss for the observed viewing directions.

Optimization details. To optimize per-vertex appearance parameters, we use the Adam method with $\beta = (0.9, 0.999)$ and $\epsilon = 10^{-8}$ in 2k iterations. When computing losses, we use the weights $w_{\mathrm{v\_reg}} = 0.1$ and $w_{\mathrm{bg}} = 10$. We use a learning rate 0.01 for per-vertex attributes and 0.001 for the spherical Gaussian (SG) parameters (representing the illumination $L_{\mathrm{env}}^{\mathrm{SG}}$).

1.3. Physics-Based Inverse Rendering

Optimization details. Initialized using the mesh M_0 predicted by the surface reconstruction stage as well as albedo/roughness maps $T_{\mathrm{a}}^{(0)}$, $T_{\mathrm{r}}^{(0)}$ (for surface reflectance) and SG-based illumination $L_{\mathrm{env}}^{\mathrm{SG}}$ produced by the neural distillation stage, our physics-based inverse rendering (PBIR) stage involves the following three steps:

1. We jointly optimize (using 1k iterations) the albedo/roughness maps T_{a}, T_{r} and the SG parameters $L_{\mathrm{env}}^{\mathrm{SG}}$ while keeping the mesh geometry fixed.
2. We first pixelize the SG-based $L_{\mathrm{env}}^{\mathrm{SG}}$ into an environment map L_{env} and then perform joint per-pixel optimizations (using 1k iterations) for the albedo, roughness, and environment maps T_{a}, T_{r}, and L_{env}.
3. We jointly optimize (using 500 iterations) all maps and the mesh geometry (per-vertex).

In practice, when optimizing albedo and roughness maps T_{a} and T_{r} in all three steps, we use the Adam optimizer with $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and the learning rates 10^{-2} for T_{a} and 5×10^{-3} for T_{r}. When computing losses, we use $w_{\mathrm{mask}} \approx 10$ and $w_{\mathrm{reg}} \approx 0.1$ (which we slightly adjust for each example).

Additionally, in the first step, we use the Adam optimizer [5] for the SG parameters with $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and learning rates around 0.001 (which we slightly adjust per example). In the second step, to suppress the impact of Monte Carlo noises during environment map optimization, we utilize the AdamUniform optimizer [6] with $\lambda = 1$ and a learning rate of 0.01. In the last step, when optimizing the mesh geometry, we again use the AdamUniform optimizer with $\lambda = 100$.

2. Additional Results and Evaluations

2.1. Additional Results

Video for view synthesis and relighting. Since results of novel-view synthesis and relighting are best viewed animated, we encourage readers to see our supplementary video (video.mp4) for a more convincing comparison on our five real-world objects.

Similar to the results shown in Fig. 6 of the main paper, our method significantly outperforms nvdiffrec-mc [2] and MII [8]. nvdiffrec-mc's geometry and material reconstructions contain heavy artifacts. Despite nvdiffrec-mc showing better novel-view results than MII (main paper's Fig. 4), the artifacts become visually prominent under novel illuminations as can be seen in the video and main paper's Fig. 6. MII offers better overall albedo than nvdiffrec-mc but suffer from over-blurring in both geometry and material reconstructions. Overall, our results show significant better quality in both geometry and material.

Outdoor illumination. The five real-world objects presented the main paper are captured under indoor lighting. In Fig. 2, we showcase the results of two of these objects re-captured under outdoor illumination. Same as the results under indoor lighting, our reconstructions are more detailed, allowing their rerenderings (under novel views) to achieve better PSNR and SSIM.

Synthetic MII dataset. The authors of MII have kindly shared their rendered results for us to compare. As their evaluation scripts are unavailable, we use our own implementation for all the quantitative results. Due to the different implementation of the evaluation metrics, MII's quantitative results presented in our main paper differ slightly from those reported in their paper.

In Figs. 3 to 6, we show more qualitative results on the synthetic MII dataset. Overall, our method offers more detailed albedo reconstructions than the baseline methods. On the other hand, none of the methods performs well on roughness estimation—likely due to the lack of robust priors. The qualitative results are consistent with the quantitative comparison in Tab. 1 of the main paper.

Our synthetic dataset. Since the MII dataset does not contain groundtruth meshes, it is difficult to evaluate the accuracy of reconstructed shapes. To address this, we create two extra synthetic scenes—*buddha* and *lion*—with groundtruth meshes for evaluation. For each scene, the training set includes 190 posed images with masks. The testing set consists of visualizations of groundtruth albedo, roughness, and renderings of the object under seven novel lighting conditions in 10 poses.

Table 1 shows quantitative comparisons between our method and the baselines. In addition to metrics used in the MII dataset, we also measure Chamfer distances [1] between optimized and groundtruth shapes (normalized so that the groundtruth has unit bounding boxes). Our method again outperforms the baselines.

As shown in Figs. 7 and 8, since the background is fully visible (i.e., each pixel of L_{env} is visible as the background of at least one input image), our method is capable of reconstructing the environment map almost perfectly. Because of this, our albedo reconstructions are not hindered by the albedo-light ambiguity—as demonstrated in Tab. 1 where the error metrics barely change with or without albedo alignment. We note that this might not apply to all scenarios, for instance, the background might not be fully visible, as shown in the MII dataset. Reconstructing indoor lighting perfectly is also challenging even if the background is completely visible, because it breaks the assumption of environmental (i.e., distant) lighting.

2.2. Additional Evaluations

Surface quality on the DTU dataset. We show quantitative results breakdown for the 15 scenes from DTU dataset [3] in Tab. 2. We use the official evaluation script to measure Chamfer distances. Please note that our results evaluated here are directly from the shape reconstruction stage. We skip evaluating the shape refinement of our physics-based inverse rendering on DTU dataset as DTU exhibit vary light occlusion from robot arms.

Usefulness of shape refinement. Lastly, we demonstrate the usefulness of our shape refinement (as the last step of the physics-based inverse rendering stage) via an ablation. As shown in Fig. 9 and Tab. 1, our shape refinement improves the accuracy of reconstructed object geometries.

References

[1] Harry G Barrow, Jay M Tenenbaum, Robert C Bolles, and Helen Cf Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings: Image Understanding Workshop*, pages 21–27. Science Applications, Inc, 1977. 3

[2] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, light & material decomposition from images using monte carlo rendering and denoising. In *NeurIPS*, 2022. 2, 4

[3] Rasmus Ramsbøl Jensen, Anders Lindbjerg Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *CVPR*, 2014. 3, 11

[4] Brian Karis and Epic Games. Real shading in unreal engine 4. *Proc. Physically Based Shading Theory Practice*, 4(3):1, 2013. 2

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1, 2

[6] Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. Large steps in inverse rendering of geometry. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 40(6), Dec. 2021. 2

[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. In *NeurIPS*, 2021. 1

[8] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *CVPR*, 2022. 2, 4

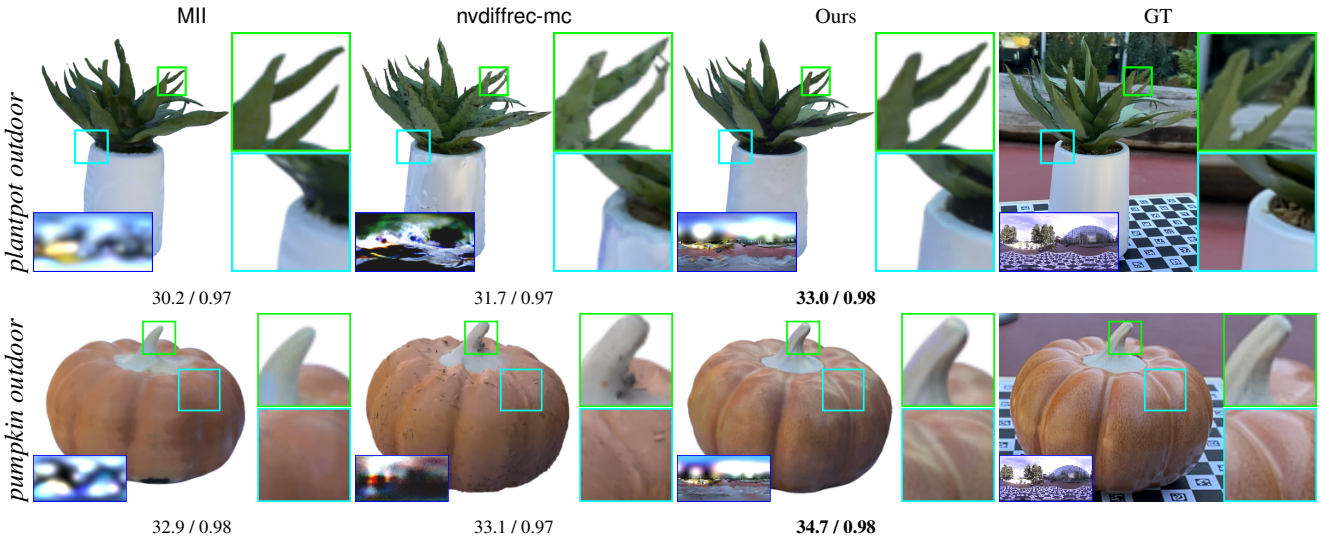


Figure 2: **Novel-view interpolation on our additional two real outdoor data.** We report the average PSNR↑ and SSIM↑ below each image. The results show that our method achieves good quality and outperforms previous arts under outdoor lighting as well.

	Speed	Relighting			Aligned albedo			Albedo			Rough.	Shape
Method	Time↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	MSE↓	CD↓
nvdiffrec-mc [2]	∼ 2 h	21.77	0.936	0.071	33.29	0.964	0.037	16.14	0.910	0.068	0.013	9.42e-5
MII [8]	∼ 10 h	24.94	0.952	0.051	30.92	0.962	0.044	19.80	0.923	0.065	0.003	5.92e-5
Ours - Distilled only	< 15 m	33.90	0.976	0.034	34.09	0.971	0.034	34.09	0.972	0.034	0.005	2.61e-5
Ours - w/o shape ref.	∼ 45 m	34.18	0.980	0.028	35.57	0.983	0.026	35.57	0.983	0.026	0.003	2.61e-5
Ours - Full	∼ 1 h	**35.30**	**0.982**	**0.026**	**37.69**	**0.985**	**0.023**	**37.68**	**0.985**	**0.023**	**0.002**	**2.56e-5**

Table 1: **Relighting, material reconstruction, and mesh quality on our synthetic dataset.** We compare our method with MII and nvdiffrec-mc. The highest performing number is presented in bold, while the second best is underscored. We measure the shape quality using Chamfer distances (CD).

Figure 3: **Qualitative comparisons of** *air_balloons* **from the MII dataset.**

Figure 4: **Qualitative comparisons of *chair* from the MII dataset.**

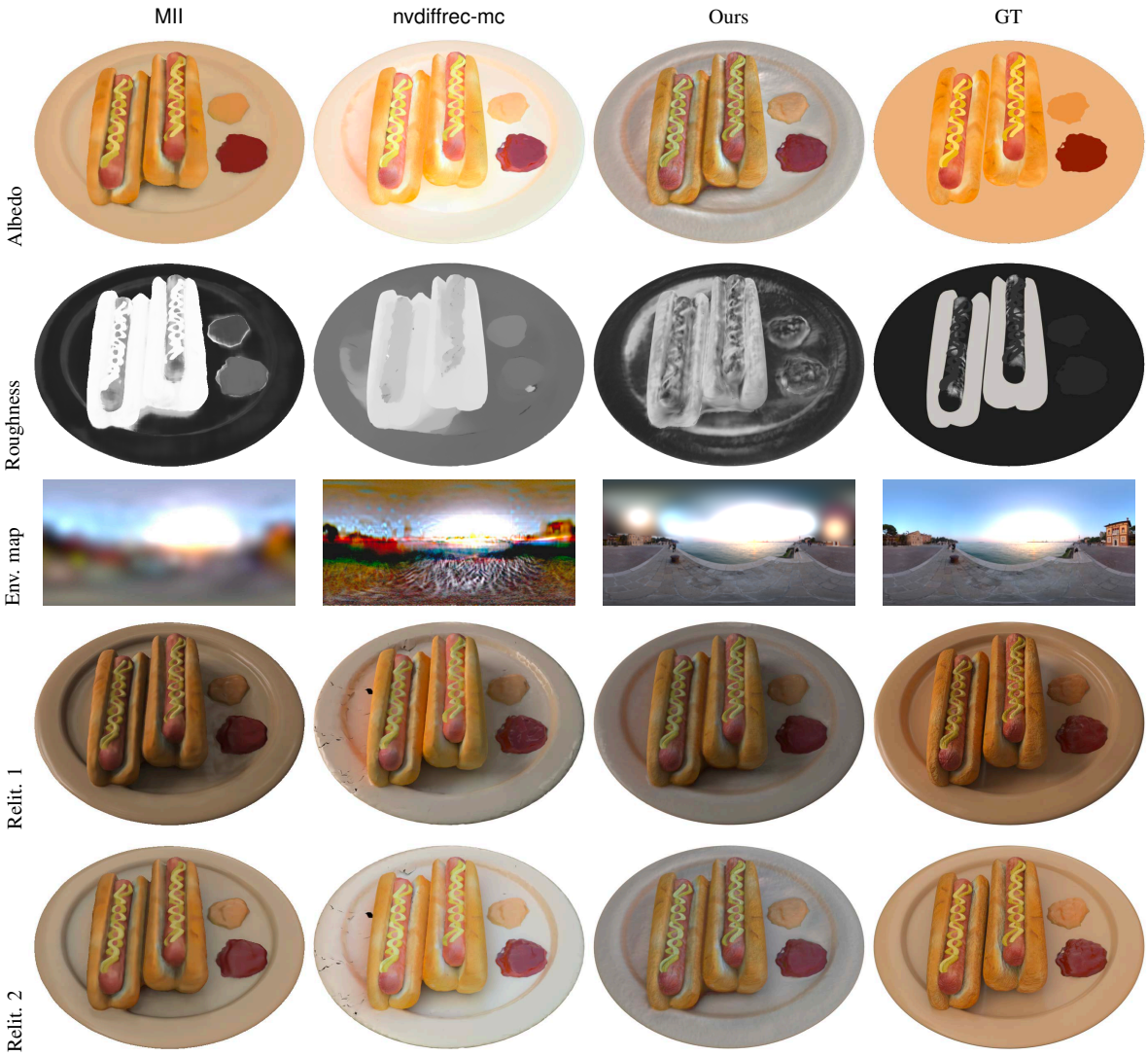


Figure 5: **Qualitative comparisons of** *hotdog* **from the MII dataset.**

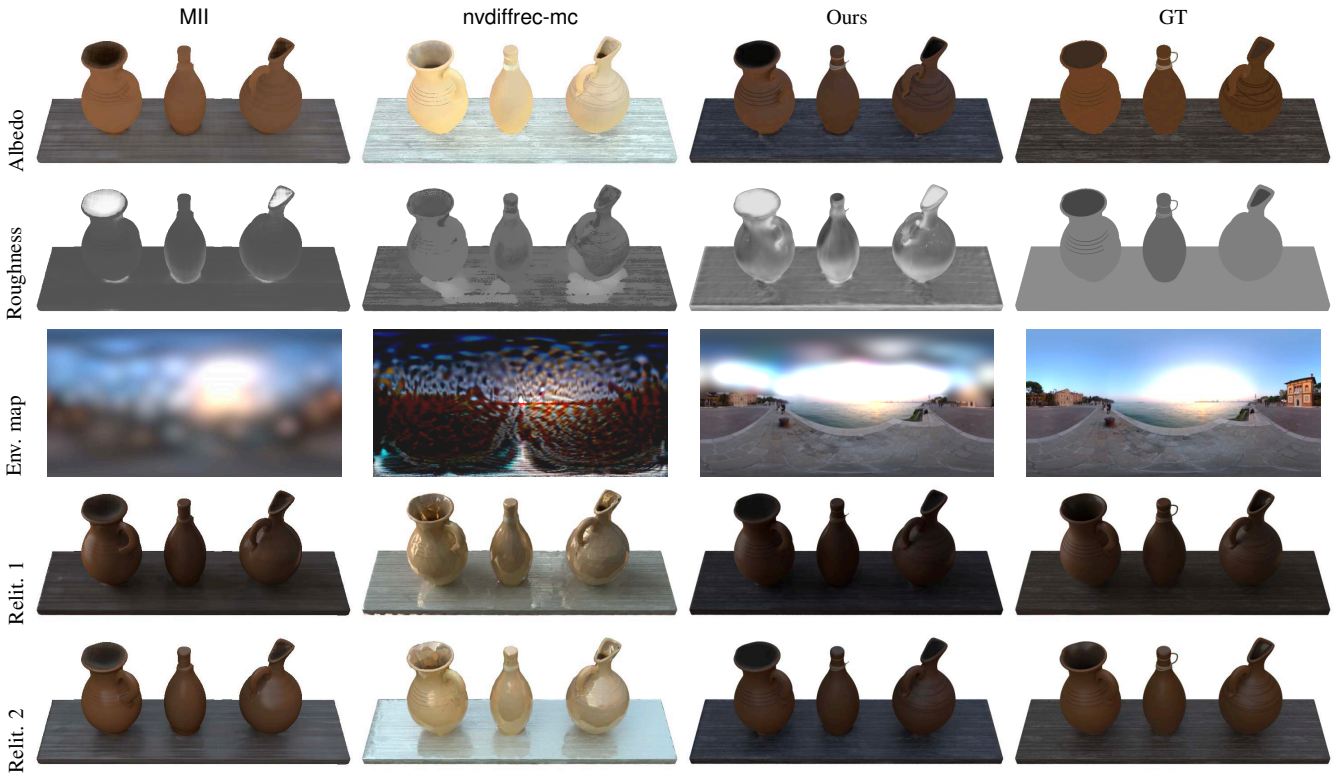


Figure 6: **Qualitative comparisons of *jugs* from the MII dataset.**

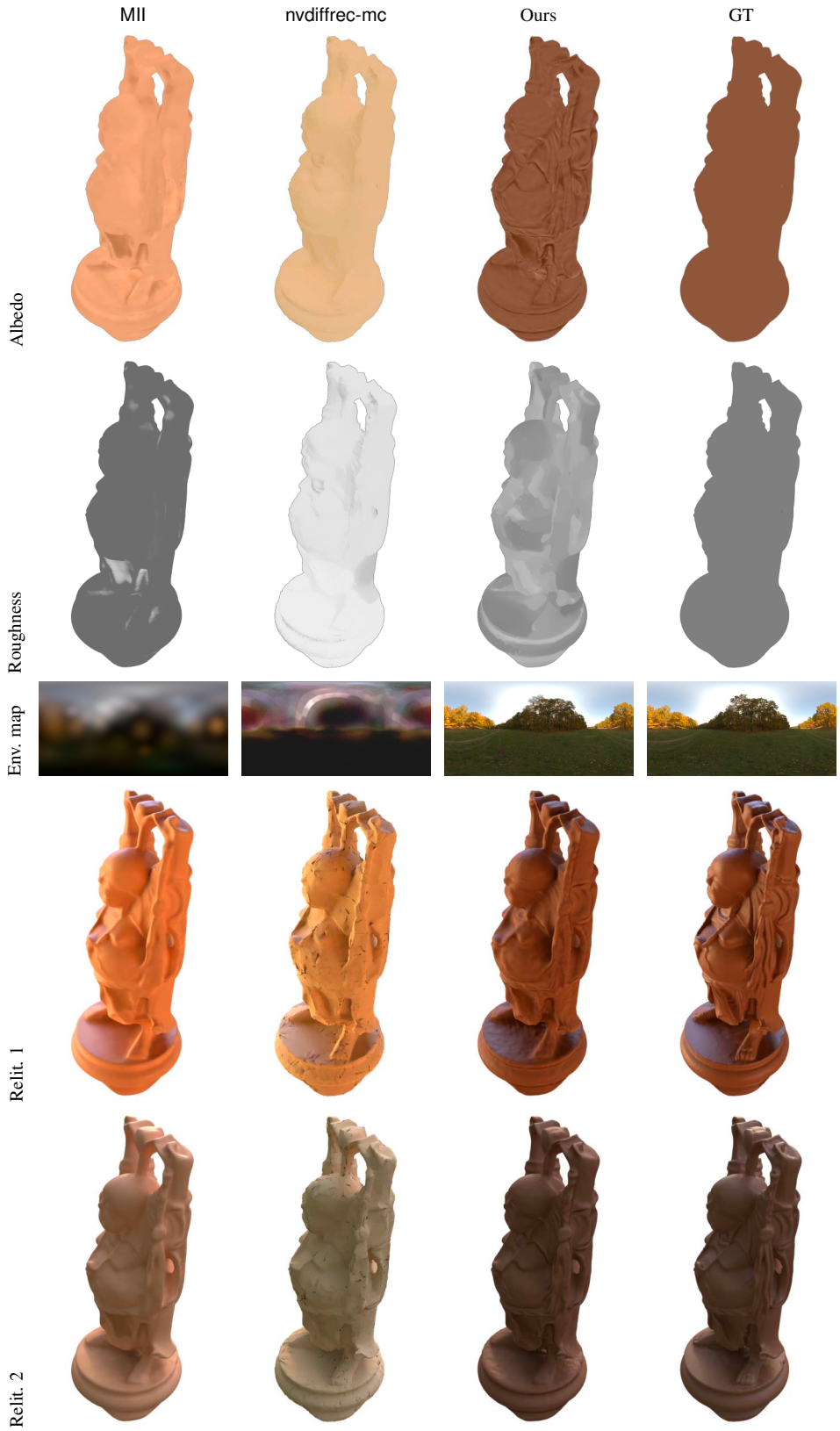


Figure 7: **Qualitative comparisons of** *buddha* **from our dataset.**

Figure 8: **Qualitative comparisons of *lion* from our dataset.**

Method	Time	avg.	24	37	40	55	63	65	69	83	97	105	106	110	114	118	122
COLMAP	1 h	1.36	0.81	2.05	0.73	1.22	1.79	1.58	1.02	3.05	1.40	2.05	1.00	1.32	0.49	0.78	1.17
NeuS	5.5 h	0.77	0.83	0.98	0.56	0.37	1.13	**0.59**	**0.60**	1.45	**0.95**	0.78	**0.52**	1.43	**0.36**	0.45	**0.45**
Ours	**5 m**	**0.66**	**0.52**	**0.72**	**0.36**	**0.35**	**0.97**	0.68	0.61	**1.27**	1.06	**0.71**	**0.52**	**0.78**	**0.36**	**0.43**	0.56

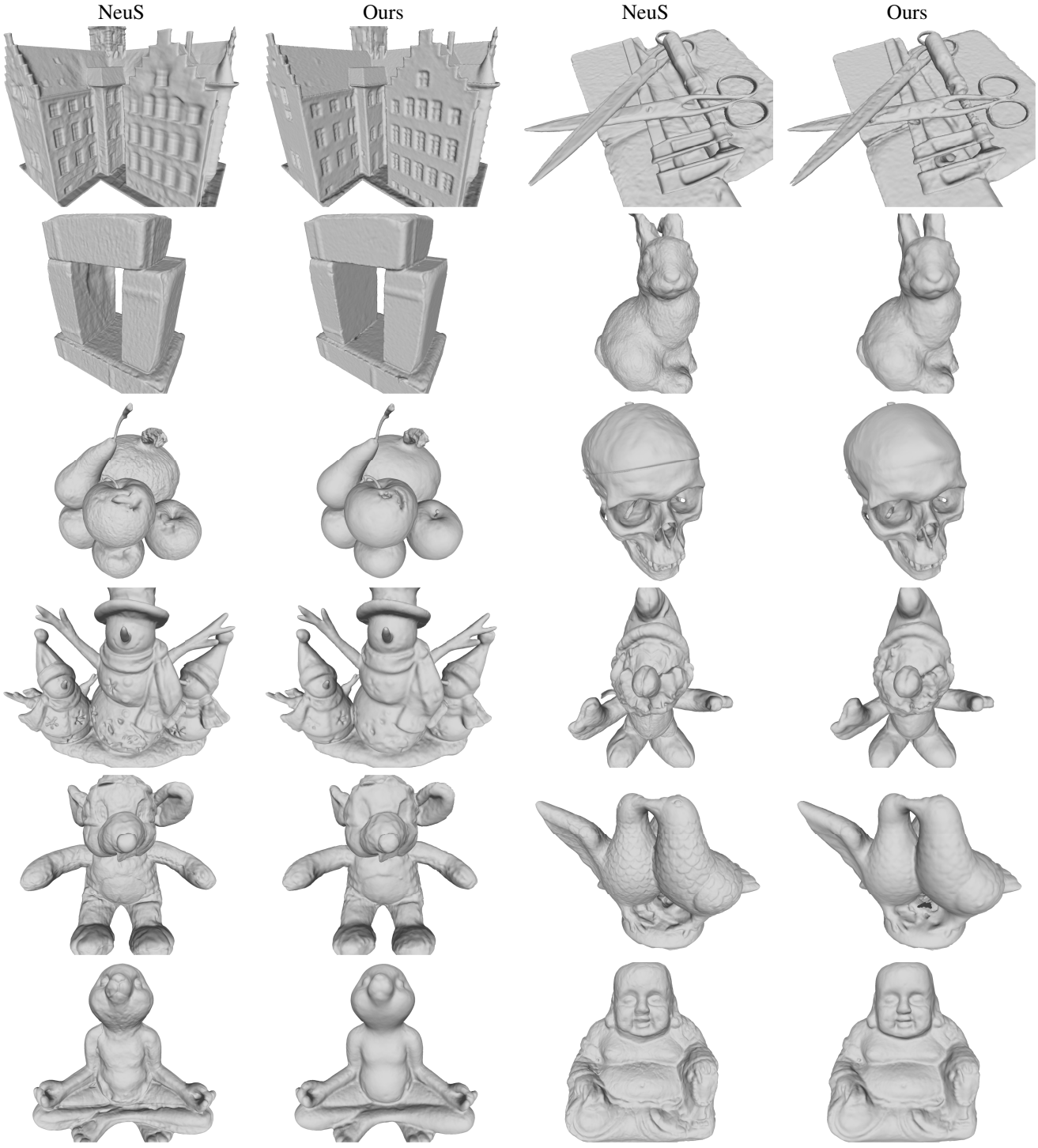
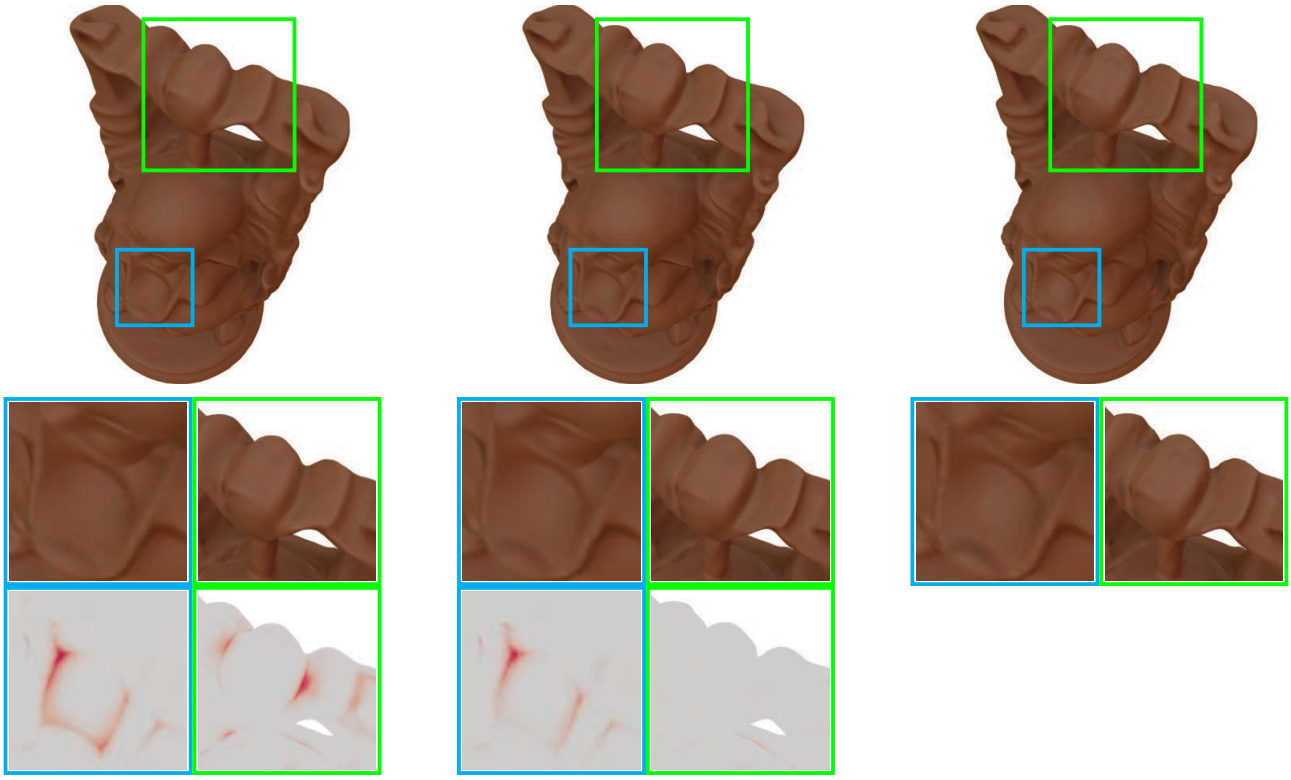


Table 2: **Quantitative results breakdown and visualization on the DTU MVS dataset [3].** We use official evaluation script to measure Chamfer distances (in mm). Our results are typically smoother with some details missing. We do not apply PBIR shape refinement as DTU exhibits significant lighting variation. See Fig. 9 and the main paper for the experiments about shape refinement.



w/o shape refinement
(34.78 / 1.98e-5)

w/ shape refinement
(36.13 / 1.91e-5)

GT

Figure 9: **Usefulness of our shape refinement in the physics-based inverse rendering (PBIR) stage.** To showcase the effectiveness of our shape refinement, we employ the *buddha* scene and present zoom-in renderings along with Chamfer distance visualizations where darker colors indicate higher errors. Additionally, we report the PSNR for relighting and the Chamfer distance, presented at the bottom of our results.