ENVIDR: Implicit Differentiable Renderer with Neural Environment Lighting Supplement

Ruofan Liang[1,2] Huiting Chen[1] Chunlin Li[1] Fan Chen[1] Selvakumar Panneer[3] Nandita Vijaykumar[1,2]

[1]University of Toronto [2]Vector Institute [3]Intel Labs

A. Additional SDF Regularizations

In addition to the Eikonal term [13] that is commonly used for constraining the learned SDF field, neural surface methods also require special weight initialization or warm-up steps to stabilize the learning of implicit surfaces [44, 39, 47]. However, unlike prior works that use a large fully implicit MLP [44, 39] or attach spatial coordinates to the grid-interpolated geometry features [47], ENVIDR's geometry MLP uses an NGP-like [26] model that uses multi-level hash encoding as the only input to the tiny geometry MLP. Therefore, the geometric initialization [2] that assumes spatial coordinates as MLP inputs cannot be applied to our model. Thus, our model requires new ways to initialize its learned geometry to better represent continuous and smooth surfaces.

To demonstrate the importance of incorporating special constraints when learning implicit neural surfaces, we first show the results of models that do not include additional SDF constraints. Figure 11 shows the surface normals of two such models, one trained solely with L1 photometric loss ("L1") and the other with L1 loss and Eikonal term ("L1+Eikonal"). Both of these models fail to capture accurate surface geometry on glossy regions, and their corresponding SDF curves oscillate around the zero-level SDF, which results in a compositing weight w_i (obtained from Eqn. 2) distribution with multiple peaks along the ray. These scattered compositing weights further cause the reconstructed surface to "collapse" into the actual object. To avoid this surface collapse, additional regularizations are needed on the initial SDF predictions.

In this work, we employ two SDF regularization terms in the early training steps to stabilize the learning of the initial SDF. The first regularization term is to force the SDF predictions of positions inside object surfaces to be away from zero-level SDF, which can suppress the multiple peaks on the compositing weight curves. To avoid over-suppression of the existing SDF predictions that are far away from the zero-level SDF, we employ a modified Cauchy loss [16] on the SDF-converted (via Eqn. 1) density values σ_i:

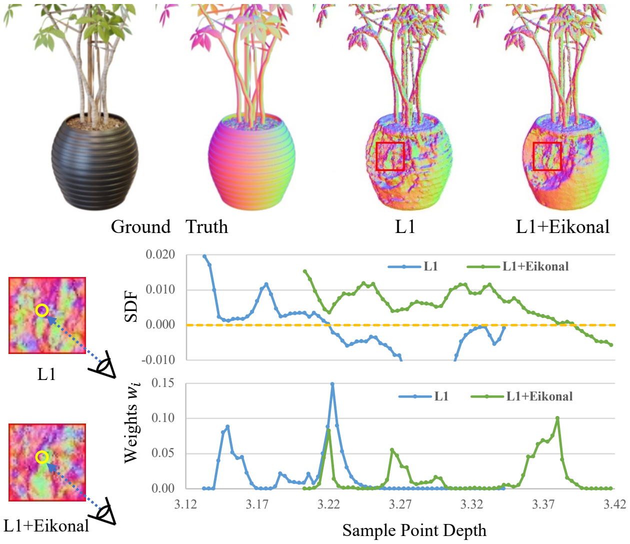
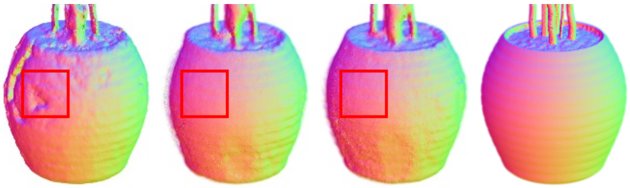


Figure 11: The normals of the surfaces learned by models without additional constraints. The results are from models after only 20k training iterations. We also plot the curves of SDF and compositing weight w_i of the sampled points along the ray for rendering the pixel shown on the left.

$$\mathcal{L}_n = \frac{1}{N}\sum_i \log\left(1 + \frac{(1-\beta\sigma_i^2}{c^2}\right) \tag{15}$$

where β is the parameter used in Eqn. 1 and c is a hyperparameter that controls the loss scale which we set to 4. This regularization term is uniformly applied to all the sampled N points per iteration. The effectiveness of this regularization is shown in Figure 12a. Compared to the results in Figure 11, $\mathcal{L}_n$ is able to help our model to get a better initial surface structure.

The second regularization is to eliminate the fluctuations of SDF curve segments that are close to zero-level SDF. The zero level set of SDF denotes the actual position of the surface, and frequent fluctuations near the surface can significantly impact the quality of surface geometry as well as sur-



(a) $\mathcal{L}_n$ Only (b) $\mathcal{L}_b$ Only (c) $\mathcal{L}_n + \mathcal{L}_b$ (d) 200k iter.

Figure 12: The estimated surface normal with our regularization terms. (a-c) show the results after 20k iterations, and (d) shows the results of the model with our full regularizations after 200k training iterations ($\mathcal{L}_n$ & $\mathcal{L}_b$ stop at 40k).

face normals. To ensure that our estimated SDF has a stable decreasing curve when it hits the surface at ray-sampled points, we propose a back-face suppression regularization. This regularization penalizes SDF curve segments with positive slopes and high corresponding compositing weights, as shown in the following equation:

$$\mathcal{L}_b = \sum_i w_i \max(\Delta s_i, 0) \frac{\Delta s_i}{\delta_i^2 + \Delta s_i^2} \tag{16}$$

In this equation, $\Delta s_i = s(\mathbf{x}_{i+1}) - s(\mathbf{x}_i)$ represents the difference in estimated SDF between two adjacent sampled points $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, δ_i is the actual distance between $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$. The compositing weight w_i for volume rendering can be obtained from Equation 2. Figure 12b demonstrates the effectiveness of this regularization term. We can observe that the smoother estimated surface is achieved with the same number of training iterations compared to the results obtained using only $\mathcal{L}_n$. However, it should be noted that the object's surface shrinks slightly compared to the other results.

By combining the two regularization terms introduced above, we are able to achieve a more stable and accurate SDF estimation. Specifically, we formulate the combination of the two regularization terms as:

$$\mathcal{L}_{reg} = \lambda_n \mathcal{L}_n + \lambda_b \mathcal{L}_b \tag{17}$$

where λ_n and λ_b are weight hyperparameters. Setting overly large values of λ_n and λ_b may prohibit the model from learning fine-grained geometry details. However, they can be beneficial in providing smooth surfaces for the later training steps. The effectiveness of the combined regularization can be seen in Figure 12c. By using these additional SDF regularizations during the early training steps, we can produce much better initial surfaces for accurate reconstruction and illumination.

B. Implementation Details

Architecture and hyperparameters. The geometry MLP $\boldsymbol{F}_g$ used in ENVIDR is similar to Instant-NGP [26]. We employ a hash encoding with 16 grid levels where each level

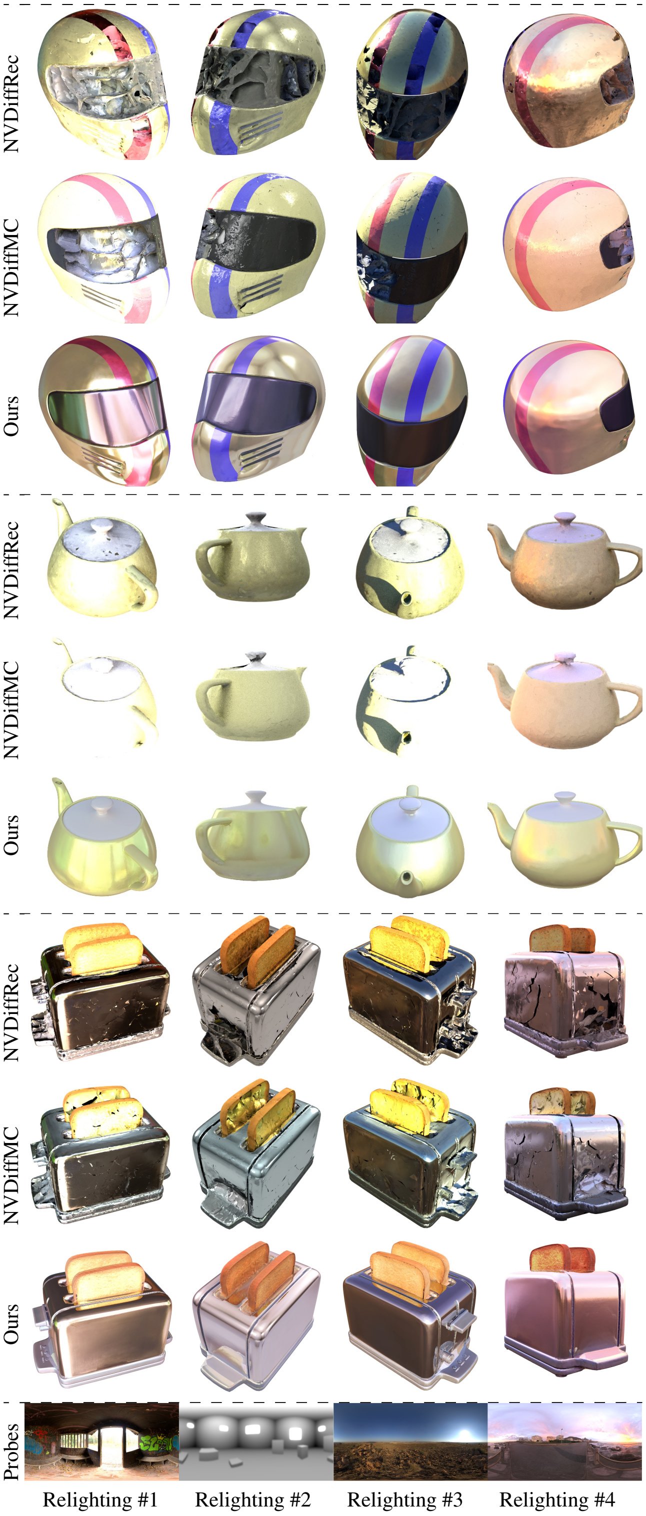


Figure 13: Additional relighting results for various scenes. Due to the unavailability of Blender files for these objects, reference relighting images are not provided for comparison. NVDIFFREC and NVDIFFRECMC are rendered by Blender Cycles.

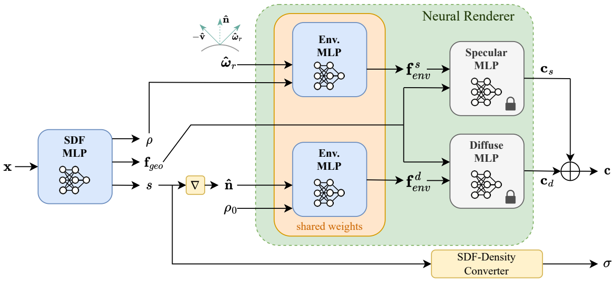


Figure 14: The model structure for general scenes. Compared to the structure for sphere rendering, 1) the specular and diffuse MLPs are pre-trained and their weights are frozen after pre-training; 2) material attributes are implicitly learned by SDF MLP, instead of as model inputs.

encodes a 2-d feature vector. The MLP $\boldsymbol{F}_g$ itself is a tiny 3-layer MLP with 64 neurons per hidden layer. The output geometry feature $\mathbf{f}_{geo}$ is a 12-d feature vector. The environment MLP $\boldsymbol{E}$ is a relatively large MLP with 4 layers and 256 neurons per layer (still smaller than a 1k HDR light probe image). The integrated directional encoding (IDE) [36] utilized by $\boldsymbol{E}$ encodes unit directions using the first 5 bands of spherical harmonics. The output environment feature $\mathbf{f}_{env}$ is also a 12-d feature vector. Specular MLP $\boldsymbol{R}_s$ is a 3-layer MLP with 64 neurons per hidden layer, while Diffuse MLP $\boldsymbol{R}_d$ is a 2-layer MLP with 32 neurons per hidden layer. We implemented our model using a PyTorch version of Instant-NGP [1].

Training details. For the neural renderer's training (Fig. 3), we use Filament PBR engine [30] to randomly synthesize new frames on the fly, with varying materials and environment lights (we use only 11 light probe images, collected by Filament [2]). We train the neural renderer for 100k iterations, each with 32000 sampled rays, using the Adam optimizer with an initial learning rate of 0.001. This training takes about 3 hours on a single RTX3090 GPU. For the training of representing general scenes (Fig. 14), we train our model for 200k iterations with 4096 sampled rays per iteration, using the Adam optimizer with an initial learning rate of 0.0005. We first train the model only with a photometric loss for 4k iterations to obtain a coarse geometry for ray-sampling acceleration. We then apply our additional SDF regularizations for about 40k iterations with exponentially decaying loss weights. The Eikonal loss term is added to the training after the first 10k training iterations. If the indirect illumination module is used, we initiate the extra raymarching pass after the first 40k training iterations. The training speed depends on the complexity of the target scene, but most scenes can be trained within 3 hours (5 hours if indirect illumination is used) on RTX3090.

Runtime. The time required to render a single 800×800 image is approximately between 0.5 to 1.2 seconds (without indirect illumination) on a single RTX3090 GPU. If the indirect illumination pass is enabled, rendering may take 1.6 times longer. Although our code base is not yet optimized for runtime performance, further optimizations are possible to achieve faster rendering.

C. Additional Results

In this section, we present additional experimental results to demonstrate ENVIDR's ability to reconstruct and render glossy surfaces. We also provide a demo video on our web page to showcase results in motion.

C.1. The Generalizability of Neural Renderer

The current implicit neural surface models [39, 44, 42] usually have a similar model structure with an SDF MLP for geometry learning and a color MLP for rendering. Since the color rendering part is relatively independent of the design of the geometry learning model, therefore, our neural renderer with *the same pre-trained weights* can also be integrated with different neural surface models to achieve similar rendering results, as illustrated in Fig. 15.

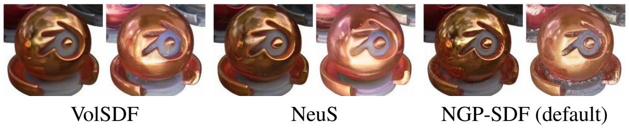

VolSDF NeuS NGP-SDF (default)

Figure 15: The rendering and relighting results of our neural renderer combined with different geometry MLP backbones. VolSDF and NeuS use their original SDF MLPs, NGP-SDF is our default model described in the paper.

C.2. Additional Comparisons

In the main paper, our comparison does not include some previous inverse rendering methods based on neural fields such as NeRD [6] and Neural-PIL [7]. This is because we find these methods usually fail to capture high-frequency reflections and have lower rendering quality. They are also more expensive to run (e.g., over 24 hours of training per scene). Fig. 16 shows our additional comparison.

C.3. The Rotation of Environment Lighting

Since our model is able to decompose the environment illumination from the training images. To demonstrate the correctness of the decomposition and versatility of our neural renderer, we further show the results of rendering scenes with a rotating environment. Suppose we want to apply a

[1] https://github.com/ashawkey/torch-ngp

[2] https://github.com/google/filament/tree/main/third_party/environments

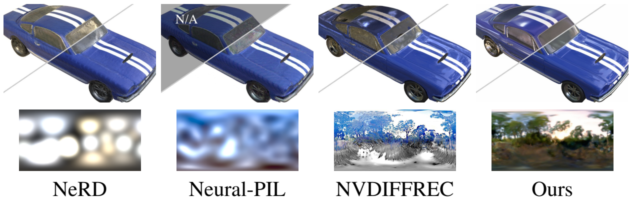


Figure 16: The comparison of inverse rendering results. Both novel view synthesis (lower right) and scene relighting (upper left) are shown. The lower part shows the light probes estimated by corresponding methods. Neither NeRD nor Neural-PIL captures the high-frequency details from the environment lighting.

rotation $\mathbf{R} \in \mathbf{SO}(3)$ to the environment map, we can simply use the rotated directional vectors $\hat{\boldsymbol{\omega}_r}' = \mathbf{R}^{-1}\hat{\boldsymbol{\omega}_r}$ as MLP inputs. The visual results in Fig. 17 show that our model also enables high quality rendering without flickering on rotating environments.

C.4. Additional Relighting

Additional relighting results on challenging synthetic shiny scenes are demonstrated in Figure 13. ENVIDR is capable of synthesizing lighting effects that are comparable to Blender's path-tracing rendering. while maintaining significantly better surface geometry compared to the two baseline models. However, these relighting results also reveal some limitations of our method. For instance, in the "teapot" example, our current method cannot synthesize shadowing effects caused by surface occlusions. We plan to address this in our future work.

C.5. Per-Scene Decompositions

Figure 17 shows the rendering decomposition of all evaluated synthetic scenes. ENVIDR can effectively decompose both the view-independent diffuse color and view-dependent specular color from multiview training images. Moreover, it can successfully distinguish between different types of materials present in the scene, such as metallic materials in the ”toaster” and ”coffee” scenes. In addition, our model also captures high-fidelity environment light probes from these shiny objects.

References

[1] Matthew Anderson, Ricardo Motta, Srinivasan Chandrasekar, and Michael Stokes. Proposal for a standard default color space for the internet—srgb. In *Color and imaging conference*, volume 1996, pages 238–245. Society for Imaging Science and Technology, 1996. 5

[2] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2565–2574, 2020. 11

[3] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 1, 2

[4] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5470–5479, 2022. 2

[5] Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. Neural reflectance fields for appearance acquisition. *arXiv preprint arXiv:2008.03824*, 2020. 2

[6] Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. Nerd: Neural reflectance decomposition from image collections. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12684–12694, 2021. 2, 13

[7] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan Barron, and Hendrik Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. *Advances in Neural Information Processing Systems*, 34:10691–10704, 2021. 1, 2, 3, 4, 8, 13

[8] Robert L Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (ToG)*, 1(1):7–24, 1982. 4

[9] Abe Davis, Marc Levoy, and Fredo Durand. Unstructured light fields. In *Computer Graphics Forum*, volume 31, pages 305–314. Wiley Online Library, 2012. 2

[10] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2367–2376, 2019. 2

[11] James AD Gardner, Bernhard Egger, and William AP Smith. Rotation-equivariant conditional spherical neural fields for learning a natural illumination prior. *arXiv preprint arXiv:2206.03858*, 2022. 2, 3, 4

[12] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, 1996. 2

[13] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 5, 11

[14] Yuan-Chen Guo, Di Kang, Linchao Bao, Yu He, and Song-Hai Zhang. Nerfren: Neural radiance fields with reflections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18409–18418, 2022. 1, 2

[15] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, light & material decomposition from images using monte carlo rendering and denoising. *arXiv preprint arXiv:2206.03380*, 2022. 1, 2, 6, 8

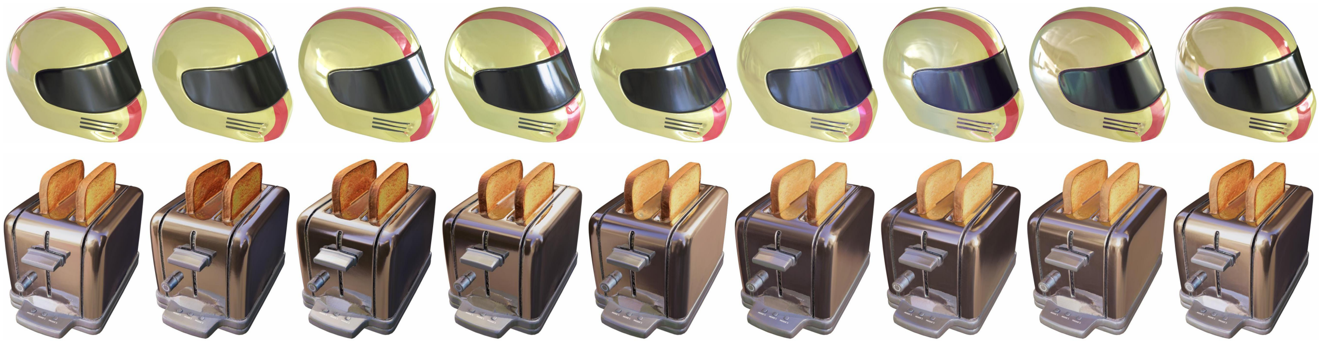


Figure 17: Fixed view, rotating environment light by 360 degrees. The helmet scene uses its original environment light, and the toaster scene uses a relit environment light. The **video** is also provided.

[16] Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5875–5884, 2021. 6, 11

[17] James T Kajiya. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150, 1986. 1, 3

[18] Georgios Kopanas, Thomas Leimkühler, Gilles Rainer, Clément Jambon, and George Drettakis. Neural point catacaustics for novel-view synthesis of reflections. *ACM Transactions on Graphics (TOG)*, 41(6):1–15, 2022. 1, 2

[19] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996. 2

[20] Ruofan Liang, Jiahao Zhang, Haoda Li, Chen Yang, and Nandita Vijaykumar. Spidr: Sdf-based neural point fields for illumination and deformation. *arXiv preprint arXiv:2210.08398*, 2022. 2, 3, 4, 8

[21] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022. 1

[22] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 2020. 1

[23] Stephen Robert Marschner. *Inverse rendering for computer graphics*. Cornell University, 1998. 2

[24] Nelson Max. Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):99–108, 1995. 3

[25] 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*, 2020. 1, 2, 3, 6

[26] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. 1, 2, 3, 5, 11, 12

[27] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8280–8290, 2022. 1, 2, 6, 7, 8

[28] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 3

[29] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1

[30] Mathias Agopian Romain Guy. Filament: a real-time physically based rendering engine. `https://github.com/google/filament`, 2018. 3, 5, 13

[31] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021. 2, 3

[32] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022. 2

[33] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. *arXiv preprint arXiv:2302.04264*, 2023. 2

[34] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. In *Computer Graphics Forum*, volume 39, pages 701–727. Wiley Online Library, 2020. 2

[35] Kushagra Tiwary, Askhat Dave, Nikhil Behari, Tzofi Klinghoffer, Ashok Veeraraghavan, and Ramesh Raskar. Orca: Glossy objects as radiance field cameras. *arXiv preprint arXiv:2212.04531*, 2022. 1

[36] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. *arXiv preprint arXiv:2112.03907*, 2021. 1, 2, 4, 6, 7, 13

[37] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. *Rendering techniques*, 2007:18th, 2007. 3, 4

[38] Jingwen Wang, Tymoteusz Bleja, and Lourdes Agapito. Go-surf: Neural feature grid optimization for fast, high-fidelity rgb-d surface reconstruction. *arXiv preprint arXiv:2206.14735*, 2022. 5

[39] 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. *Advances in Neural Information Processing Systems*, 2021. 2, 5, 11, 13

[40] Suttisak Wizadwongsa, Pakkapon Phongthawee, Jiraphon Yenphraphai, and Supasorn Suwajanakorn. Nex: Real-time view synthesis with neural basis expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8534–8543, 2021. 2

[41] Daniel N Wood, Daniel I Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H Salesin, and Werner Stuetzle. Surface light fields for 3d photography. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 287–296, 2000. 2

[42] Tong Wu, Jiaqi Wang, Xingang Pan, Xudong Xu, Christian Theobalt, Ziwei Liu, and Dahua Lin. Voxurf: Voxel-based efficient and accurate neural surface reconstruction. *arXiv preprint arXiv:2208.12697*, 2022. 3, 13

[43] Xiuchao Wu, Jiamin Xu, Zihan Zhu, Hujun Bao, Qixing Huang, James Tompkin, and Weiwei Xu. Scalable neural indoor scene rendering. *ACM Transactions on Graphics (TOG)*, 41(4):1–16, 2022. 1, 2

[44] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 1, 2, 3, 5, 6, 11, 13

[45] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. *Advances in Neural Information Processing Systems*, 33:2492–2502, 2020. 3

[46] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5752–5761, 2021. 1

[47] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in Neural Information Processing Systems*, 2022. 3, 5, 11

[48] Kai Zhang, Fujun Luan, Qianqian Wang, Kavita Bala, and Noah Snavely. Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5453–5462, 2021. 1, 2, 8

[49] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6

[50] Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul Debevec, William T. Freeman, and Jonathan T. Barron. Nerfactor: Neural factorization of shape and reflectance under an unknown illumination. *ACM Trans. Graph.*, 40(6), dec 2021. 1, 2

[51] Yuanqing Zhang, Jiaming Sun, Xingyi He, Huan Fu, Rongfei Jia, and Xiaowei Zhou. Modeling indirect illumination for inverse rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18643–18652, 2022. 3

[52] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 2

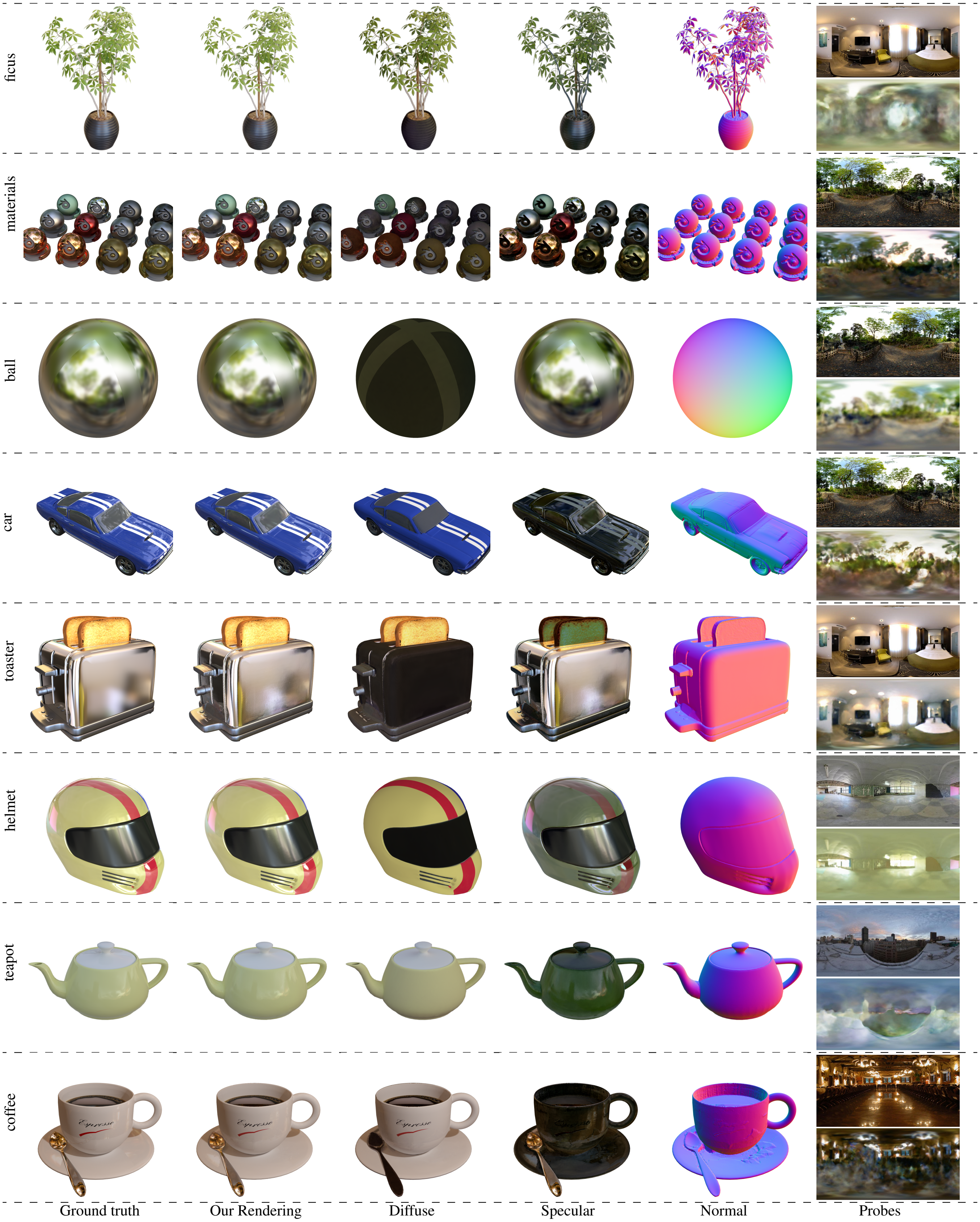


Figure 17: Our decompositions. In the “Probes” column, the upper row shows the reference, and the lower row shows our estimation.