

Supplementary Materials: Neural Layered BRDFs

Jiahui Fan
Nanjing University of Science and
Technology
China
fjh@njust.edu.cn

Jian Yang[†]
Nanjing University of Science and
Technology
China
csjyang@njust.edu.cn

Beibei Wang[†]
Nankai University,
Nanjing University of Science and
Technology
China
beibei.wang@nankai.edu.cn

Miloš Hašan
Adobe Research
USA

Ling-Qi Yan
University of California, Santa
Barbara
USA
lingqi@cs.ucsb.edu

To efficiently simulate the appearance of layered materials, we propose a compact latent representation for layered materials and a neural network that performs layering operation in this latent space. In the main contents of this paper, we show that our evaluation network can produce better results in considerably shorter time, compared to previous work. In this supplementary material, we introduce show some other operations that can be performed on it, including interpolation, mipmapping and importance sampling, and then show the results of them.

In Sec. 1.1, we introduce the interpolation and mipmapping of our latent representation. In Sec. 1.2, we show the method to perform importance sampling on our latent representation evaluation. Finally, we show the results of above in Sec. 2, together with some extra quality validation for the representation network.

1 METHODOLOGIES

In this section, we introduce some other operations that work properly in our proposed latent space, including interpolation, mipmapping and importance sampling. These operations are key to our purpose to edit and render layered BRDFs efficiently.

1.1 Latent space interpolation and mipmapping

By interpolating two or more given BRDFs, a new BRDF can be obtained, showing a natural transition effect between the input materials. We interpolate the BRDFs by performing the interpolation

of the latent vectors of two input BRDFs. Further, we extend the BRDF interpolation operation to level-of-detail rendering. Recall that we use a multi-channel texture to define SVBRDFs, where each texel is a latent vector instead of an RGB/RGBA value. We build a mipmap of our latent texture as a preprocessing, then we use standard trilinear interpolation to query the mipmap at appropriate levels during rendering, which successfully avoids the aliasing even at a low sampling rate (1 spp). During rendering, we compute the pixel's footprint for each shading point, and then query the mipmapped latent texture with trilinear interpolation, using the footprint size to find a proper level in the mipmap.

1.2 Importance sampling network

Importance sampling is a critical operation for including a BRDF in a practical path tracing system. Specifically, for a given incoming direction, we want to choose an outgoing direction with a pdf roughly proportional to the outgoing BRDF lobe as a function on the hemisphere; we also need to be able to evaluate the sampling pdf for a given direction. To introduce a sampling operation for Neural BRDFs represented in our latent space, our approach is to use an analytic proxy distribution to mimic the actual BRDF lobe.

Generally, we use a weighted sum of a Gaussian lobe and a Lambertian lobe to fit any pdf, and we train a small network to predict the parameters of our pdf proxy directly from latent codes. Our pdf proxy is defined as

$$\text{pdf}(\omega_i, \omega_o) = (1 - w)G_\sigma(h_x, h_y) + wL(\omega_o) \quad (1)$$

where (h_x, h_y) represents the projected half vector (its z -component is dropped), and G_σ is a Gaussian function with standard deviation σ , normalized on the projected hemisphere. L is the Lambertian pdf on the outgoing hemisphere (i.e., $\cos \theta_o / \pi$), where θ_o is the angle between the outgoing direction ω_o and the macro surface normal.

We implement the importance sampling network and use it in the rendering as follows:

Network design. Our sampling network is a simple four-layer MLP (with 128, 512, 128 and 32 hidden units individually) with ReLU layers as activation. Generally, it takes any BRDF latent vector and an incoming direction as input, and outputs the sampling parameters σ and w of this BRDF. In practice, for each latent vector,

[†]Corresponding authors. Email: beibei.wang@nankai.edu.cn.

[†]Corresponding authors. Email: csjyang@njust.edu.cn.

Jiahui Fan, Beibei Wang and Jian Yang are with PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGGRAPH '22 Conference Proceedings, August 7–11, 2022, Vancouver, BC, Canada

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9337-9/22/08...\$15.00

<https://doi.org/10.1145/3528233.3530732>

we uniformly generate 40×40 different incoming directions and predict the corresponding parameters via this network, then we take the averaged pdf parameters predicted by the network from these individual incoming directions.

Data preparation. To train this network, we define the concept of a *generalized normal distribution function* (GNDF) of the BRDF, and match our predicted pdf proxy with it. The name of GNDF is chosen because for a microfacet BRDF, its GNDF has a similar (though not identical) shape to its NDF. The GNDF is the normalized average of the BRDF, in half-angle space, over all incoming directions ω_i ; it is thus a 2D function of the half-vector. In practice, we estimate the GNDF by uniformly sampling 40×40 incoming vectors on the upper hemisphere and averaging the resulting 2D BRDF lobes over half-angle space. Totally, we randomly choose 3,000 two-layer BRDFs from the dataset that we used to train the representation and layering networks and another 300 two-layer BRDFs for validation. We first project them into the latent space and then we compute the ground-truth GNDFs.

Training. We train the importance sampling network by minimizing the difference between our pdf proxy (Equation 1) and the GNDF, in the following sense. To achieve this, we sample 40×40 points on the ground-truth pdf and our proxy pdf (Equation 1), and then minimize their difference by the Kullback-Leibler divergence (KLD) loss. The KLD loss is defined as

$$KLDLoss = \frac{1}{M} \sum_M \left(S(f^{gt}) (\log S(f^{gt}) - \log S(f^{pred})) \right), \quad (2)$$

where S denotes the softmax function and M denotes the sample count on a distribution. We start at a learning rate of 3×10^{-5} and shrink it by 0.7 for every 3 epochs. We trained this network for 10 epochs in total, which costs less than 1 hour on an RTX 2080Ti video card.

Integration into the rendering framework. Our neural based importance sampling strategy works well with modern rendering framework. For any BRDF, we need to precompute the parameters of our proxy (Equation 1) in advance. In rendering, when we sample a BRDF according to our proxy, we firstly generate a random number to choose between the Lambertian and the Gaussian components, with the probability of diffuse ratio w . Then we importance-sample the chosen component to obtain the outgoing direction ω_o (in case of the Gaussian lobe, this is done by first sampling the half vector h and transforming it into outgoing direction). We finally calculate the pdf value for the chosen outgoing direction; this uses the Jacobian term of the half-angle transform, as detailed by Walter et al. [2007]. We validate the results of our sampling methods in Figures 4 and 5.

Note again that by definition, our pdf proxy only depends on σ and w . Hence, even though the importance sampling takes an incoming direction as input, the pdf proxy parameters are independent of it, and can be precomputed only once using our sampling network for a given BRDF before rendering. Therefore, our importance sampling is very fast, because no network inference is performed on the fly.

2 RESULTS

In this section, we firstly show some extra results and comparisons about our latent representation evaluation, both on analytic BRDFs and some measured BRDFs/BTFs. Then, we show the results of the interpolation and mipmapping of our latent representation. At last, we give some rendering comparison to show that our importance sampling strategy works as expected.

2.1 Interpolation and mipmapping

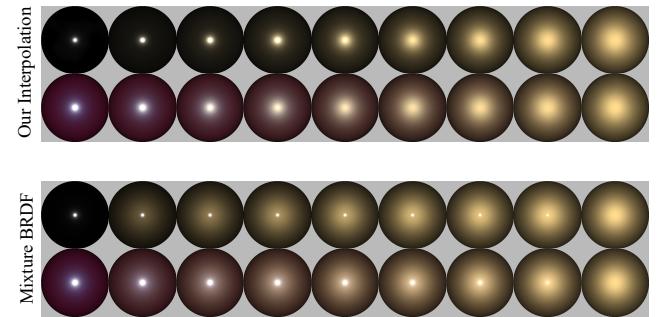


Figure 1: Comparison between our latent space interpolation and linear interpolation (blending/mixture of BRDFs).

Figure 1 shows a visualization of our interpolation, compared to naive linear interpolation of the BRDF values themselves. Our latent space interpolation gives more natural results. For example, during the interpolation between two BRDFs with low roughness and high roughness, we naturally expect one lobe with intermediate roughness, rather than a mixture of both.

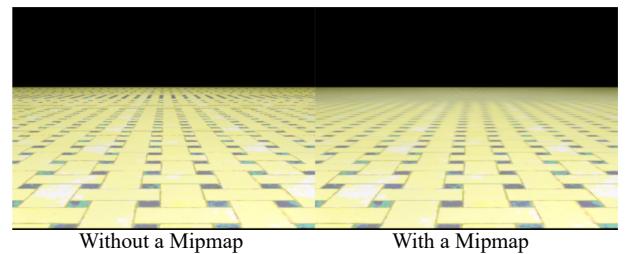


Figure 2: Results using our method without and with latent texture mipmapping, both rendered at 1 spp.

In Figure 2, we use a mipmap which is generated from the multi-channel latent BRDF texture as a precomputation, and then it is queried on the fly in the standard way, using the appropriate level with trilinear interpolation. The mipmap reduces aliasing even with very low sampling rates.

2.2 Sampling network

In Figure 3, we compare the pdf approximation among our method, Lambertian sampling and GGX sampling with the generalized NDF reference. From the curves, we can see that our pdf proxy is able

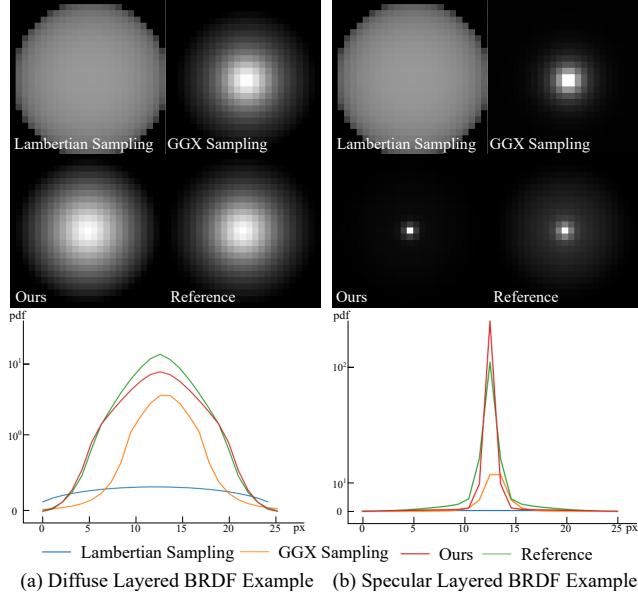


Figure 3: Comparison of different pdf approximation strategies. (Top row) Visualization of the normalized NDFs predicted by different methods for importance sampling. (Bottom row) Visualization of 1D curves showing the pdf values, extracted horizontally through the center of the corresponding NDFs. Our pdf proxy produces the closest results to the generalized NDF reference, compared to those by Lambertian sampling or GGX sampling.

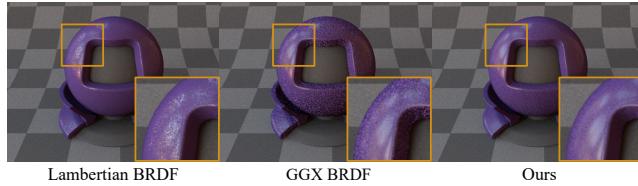


Figure 4: Comparison among three sampling strategies. Left: sampling the outgoing Lambertian lobe. Middle: sampling according to the GGX lobe with parameters obtained from the top layer. Right: our method, sampling two lobes predicted by our sampling network. All results are produced using BRDF sampling only, with 256 spp. Our method has the least variance.

to predict pdfs that are suitable to sample both specular BRDFs and diffuse BRDFs, while pdfs from Lambertian sampling and GGX sampling are far away from the reference.

In Figure 4, we compare the rendered results using our sampling network against sampling parametric lobes like Lambertian lobes or GGX, with equal number of samples per pixel. Our method produces the best results. Also note that, since our fitted GNDF is independent of the incoming direction, and only dependent on the underlying BRDF, we can precompute and store the fits as a preprocessing. In this way, we avoid the expensive inference of the sampling network

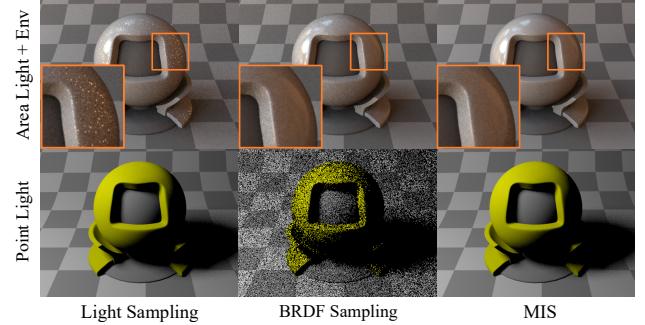


Figure 5: Our method can be naturally applied in the MIS framework. We compare between light sampling only, BRDF sampling only and MIS under two different lighting configurations (large light sources vs. a small light source). In both cases, MIS produces the best results, as expected.

when drawing samples at rendering time. Therefore, our sampling method is as efficient as sampling an analytic BRDF lobe.

Having enabled BRDF sampling, our method automatically enables MIS. In Figure 5, we compare the results rendered with light sampling only, BRDF sampling only, and MIS combining light sampling and BRDF sampling. As expected, MIS further improves the sampling quality.

2.3 Extra quality validation for the representation network

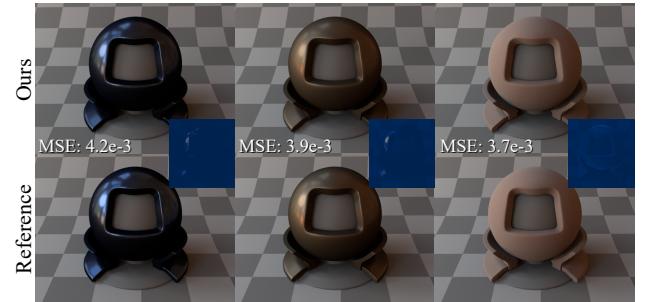


Figure 6: Our method is able to represent both rough and smooth BRDFs from the MERL dataset [Matusik 2003]. The reference is rendered using interpolated BRDF queries from the dataset directly.

Our representation network can produce fairly close results to the reference in rendering, and outperforms many previous methods. In Figure 10, we compare our representation network against Rainer et al. [2020], Sztrajman et al. [2021] and Guo et al. [2018] (reference) on varying materials. For their results, we use the pretrained model provided by Rainer et al. [2020] and train the model for Sztrajman et al. [2021] using their released code. Rainer et al. [2020] cannot handle high-frequency materials well, and suffers from visible artifacts.

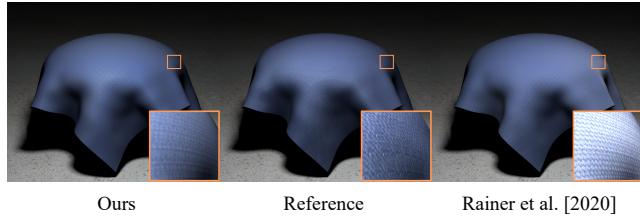


Figure 7: Comparison of the representation ability on a measured SVBRDF/BTF using our method (left), Rainer et al. [2020] (right) and the reference. The BTF data is from the UBO2014 BTF dataset [Weinmann et al. 2014]. While the global appearance is good using both methods, our method produces better grazing-angle result than Rainer et al. [2020]. All insets are with +2.8 exposure adjustment.

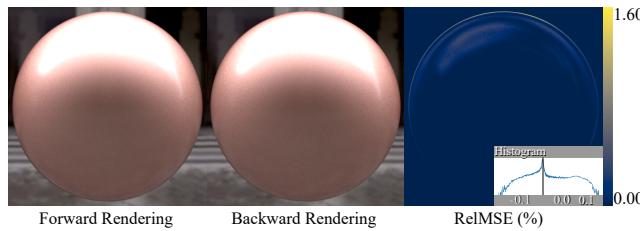


Figure 8: Rendering results by exchanging the incoming and outgoing directions and their difference. Our network has already preserved reciprocity well without imposing explicit regulation on it.

Although Sztrajman et al. [2021] produces visually similar results to ours, their method has $6\times$ more storage cost than our method (even when using the dimensionality reduction), which makes it less practical for SVBRDFs. Even though Sztrajman et al. [2021] can produce lower error in some cases (rows 2–4), there are still obvious color differences from the reference.

Our representation network is suitable not only for analytic BRDF data, but can generalize to measured BRDFs, such as those from the MERL dataset [Matusik 2003]. In Figure 6, we show the rendering results by our network on MERL dataset [Matusik 2003]. We also show that our representation network is able to represent SVBRDFs/BTFs in Figure 7 (note again that Sztrajman et al. [2021] would be difficult to apply for this purpose).

Our network does not force reciprocity during training. However, since our training data is symmetrically sampled by both the incoming and outgoing directions on the projected hemisphere, the reciprocity is naturally preserved in our network. In Figure 8, we show the rendering results by exchanging the incoming and outgoing directions. The histogram of difference shows that our network can preserve reciprocity reasonably well.

2.4 Tradeoff between network size and representation accuracy

Our representation network, which consists of several repeated MLP blocks, has the generalized ability to represent a large range

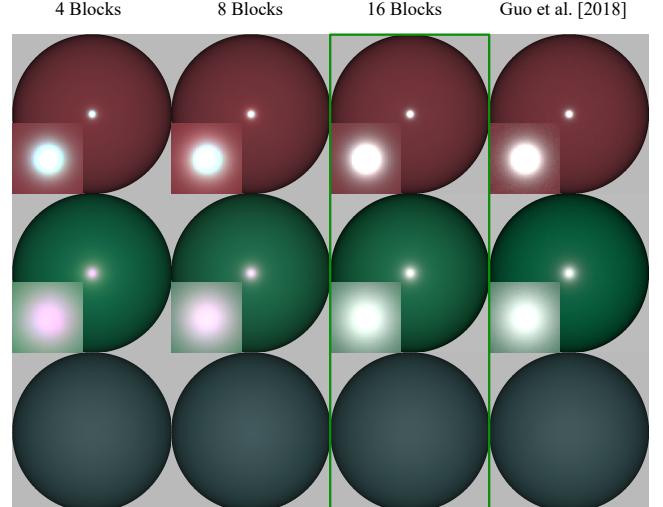


Figure 9: The time cost and accuracy comparison among three different network settings (4, 8 and 16 blocks) on both the validation dataset and a Matpreview test scene (not shown in this figure). Our current choice (16 blocks, marked in green) is powerful enough to represent some difficult BRDFs, while smaller networks struggle to show the highlight precisely.

of common BRDFs. In Figure 9, we show the tradeoff between the network size and the representation accuracy. We compare the outgoing radiance visualization, time cost, validation error and test scene MSE among our current setting and two smaller networks. From our experiments, we conclude that smaller networks bring higher computational efficiency, but at the cost of losing representation accuracy. The smaller networks may also produce wrong color in the highlights of specular BRDFs, and have higher error in both the validation dataset and the test scene. The results show that our current design choice is necessary to preserve the appearances of complex BRDFs, with a reasonable computational overhead.

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.
- Wojciech Matusik. 2003. *A data-driven reflectance model*. Ph.D. Dissertation. Massachusetts Institute of Technology.

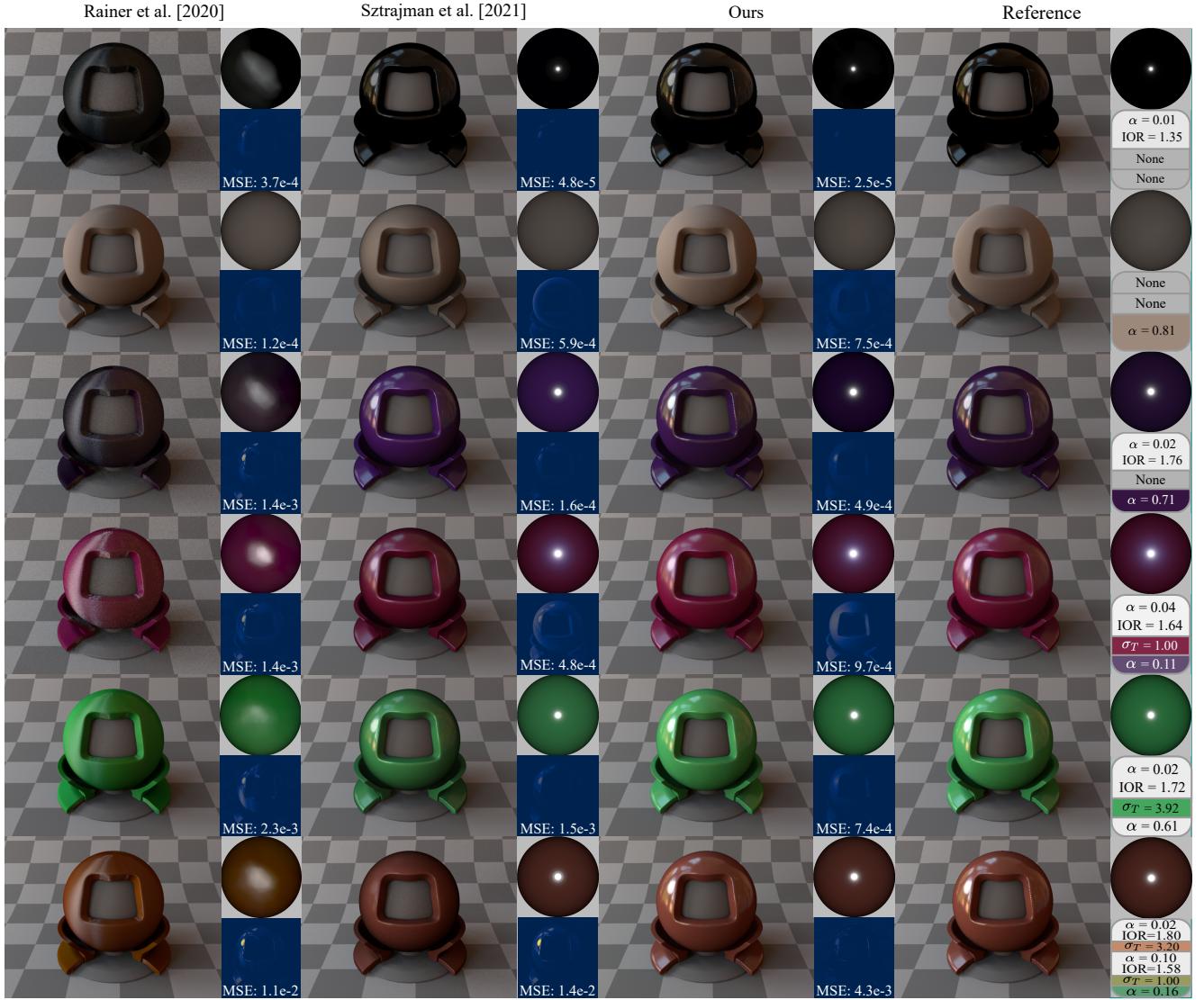


Figure 10: Comparison of the representation abilities with different BRDF configurations between Rainer et al. [2020], Sztrajman et al. [2021], and our representation network. The visualization of the outgoing radiance and the difference maps are also provided. For single-layer materials, the reference is produced by the microfacet model. For layered materials, the reference images are rendered using Guo et al. [2018]. Both Sztrajman et al. [2021] and our method produce similar results to the reference, while Rainer et al. [2020] is less accurate. Note that, although the method of Sztrajman et al. [2021] can produce smaller numerical error in some cases, it also produces obvious color differences. Also note that we focus on representation in this figure; the multi-layer BRDFs being projected are ground truth, not the outputs of our layering network.

Gilles Rainer, Abhijeet Ghosh, Wenzel Jakob, and Tim Weyrich. 2020. Unified Neural Encoding of BTFs. *Computer Graphics Forum (Proceedings of Eurographics)* 39, 2 (June 2020). <https://doi.org/10.1111/cgf.13921>

Alejandro Sztrajman, Gilles Rainer, Tobias Ritschel, and Tim Weyrich. 2021. Neural BRDF Representation and Importance Sampling. *Computer Graphics Forum* n/a, n/a (2021). <https://doi.org/10.1111/cgf.14335>

Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet Models for Refraction Through Rough Surfaces (*EGSR 07*). 195–206.

Michael Weinmann, Juergen Gall, and Reinhard Klein. 2014. Material Classification Based on Training Data Synthesized Using a BTF Database. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014*,

Proceedings, Part III. Springer International Publishing, 156–171.