Real-Time Neural BRDF with Spherically Distributed Primitives Supplemental Material

Yishun Dou[2] Zhong Zheng[2] Qiaoqiao Jin[1] Bingbing Ni[1,2] Yugang Chen[2] Junxiang Ke[2]
[1] Shanghai Jiao Tong University, Shanghai 200240, China [2] Huawei
yishun.dou@gmail.com nibingbing@sjtu.edu.cn

1. Additional Implementation Details

This section describes additional implementation details, including training strategies, tiny neural network, pruning of sphere grid, and importance sampling.

Training Strategies. Given bidirections $\boldsymbol{\omega}_i, \boldsymbol{\omega}_o$ and ground truth reflectance values y that are obtained via measurement or simulation, the optimization problem is:

$$\underset{\mathcal{D},\mathcal{S}_g,\Phi}{\arg\min}\ \mathbb{E}_{\boldsymbol{\omega}_{i,o},y}||m(\psi(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o;\mathcal{D}[\mathcal{S}_g]);\Phi)-y||. \quad (1)$$

This cannot be solved directly by usual gradient methods since the indexing $\mathcal{D}[\mathcal{S}_g]$ is non-differentiable. To this end, we use a proxy soften matrix $\mathcal{C} \in \mathbb{R}^{2n\times 2^b}$ from which the integer indices $\mathcal{S}_g = \arg\max_j \mathcal{C}[j]$ can be obtained from row-wise argmax, following [15]. We can then replace the hard indexing $\mathcal{D}[\mathcal{S}_g]$ with a matrix product $\sigma(\mathcal{C})\mathcal{D}$, where $\sigma(\cdot)$ is the softmax function applied row-wise on matrix $\mathcal{C}$.

For efficient training and better convergence, we expect the argmax results will almost freeze after a training period and the gradient that is backpropagated from a grid point of $\mathcal{S}_g$ mainly affects single corresponding primitive. That is, we should adjust the softness during training. Accordingly, we introduce a tunable temperature hyperparameter τ to the softmax. We obtain the following optimization problem:

$$\underset{\mathcal{D},\mathcal{S}_g,\Phi}{\arg\min}\ \mathbb{E}_{\boldsymbol{\omega}_{i,o},y}||m(\psi(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o;\sigma_\tau(\mathcal{C})\mathcal{D});\Phi)-y||, \quad (2)$$

where the temperature softmax is $\sigma_\tau(\boldsymbol{x}_i) = \exp(\boldsymbol{x}_i/\tau)/\sum_j \exp(\boldsymbol{x}_j/\tau)$. The hyperparameter τ controls the softness of the probability distribution. When τ gets lower, the biggest value in $\boldsymbol{x}$ gets more probability. In our case, τ is initialized with 1.0 and decreased linearly with the training epoch until the minimum 0.5. We also use the straight-through estimator [1] to make the loss be aware of the hard indexing during training, *i.e.* we use Eq. (1) for forward pass and Eq. (2) for backward pass. We jointly train the codebook entries, the indices, and the MLP by applying Adam [7]. The learning rate is initialized to 10^{-2}, which is then decayed using multistep scheduler.

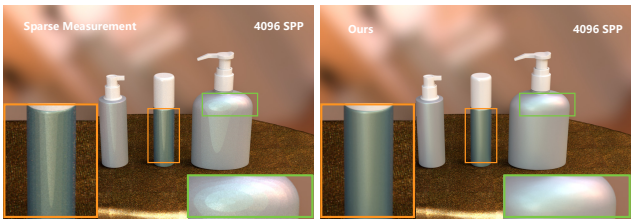

Figure 1. **NeuBRDF trained on sparse measurements.** Our approach can generalize to sparse measurements for both the BRDF (bottles) [5] and SVBRDF (table cloth) [9]. *Bottles:* With our grid pruning, unsupervised pixels are replaced with their parent, resulting in a smooth rendering. *Table Cloth:* NeuBRDF shows faithful rendering effect compared with the measured counterpart.

Tiny Neural Network. A lightweight neural network is made possible when the informative features are fed. Our fully connected neural network comprises of four fully connected layers. The two hidden layers have 128 neurons each with ReLU activation functions. The output layer reduces the 128 dimensions to three RGB values. Moreover, we add sparsity constraints to the weight matrices in hidden layers. After convergence, the 128 neurons in hidden layers are then structured pruned [6] to 64 neurons. We then freeze other learnable parameters and only finetune the pruned neural network.

Importance Sampling. The importance sampling of a reflectance model facilities the convergence of Monte Carlo path tracing. For BRDF, we implement importance sampling with an additional analytic lobe following recent neural-based works [3, 13]. For SVBRDF and BTF, we use cosine-hemisphere importance sampling.

Sphere Grid Pruning. The measured database is typically acquired using a gonioreflectometer with motorized robotic arms. The acquisition setups, such as the angular resolution of sensors and lights, vary widely among different databases. Due to the uniform pixelation of our spherical grid, our framework can adapt to data with different sampling strategies. However, there may be some grid points

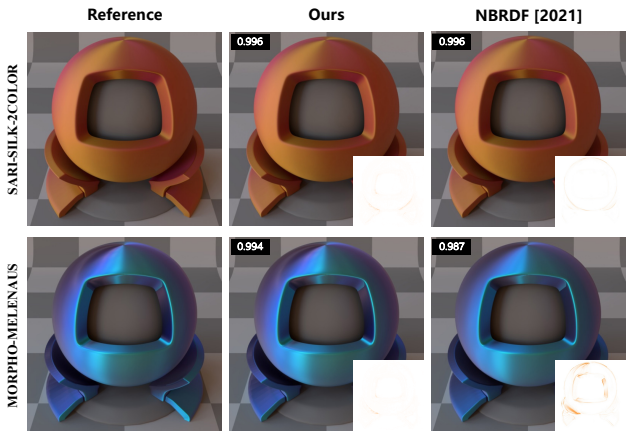

Figure 2. **Quality comparison in representing anisotropic materials from RGL [2] database.** Our method well preserves the highly frequent changes on appearance, especially in the complicated MORPHO-MELENAUS. We show the SSIM value in top-left corner and SSIM pixel-value in bottom-right, following NBRDF [13] for fair comparison.

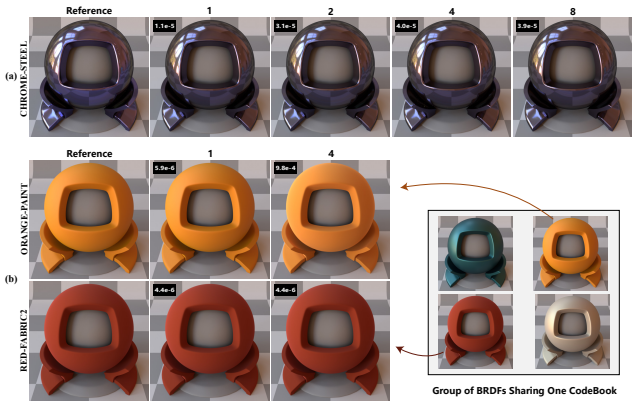

Figure 3. **Sharing codebook across multiple instances.** The digits above images represent the number of instances sharing one codebook. We show the MSE in top-left corner. (a) Representation quality degenerates as the size of belonging material group increases. (b) We demonstrate the ability of neural primitives generalizing to different colors.

that are never supervised during training. For efficiency at runtime and meaningful feature query, we discard these points before deployment. That is, a grid point is removed if there is no training direction $\boldsymbol{\omega}$ locating at the surrounding four pixels. Recall that the HEALPix is hierarchical, *i.e.* a pixel is divided into four at next level, we can thus naturally replace these four pixels with their parent, which is similar with pruning an octree in [14].

This pruning strategy is useful for isotropic materials or sparse measurements. For the isotropic materials, such as those in MERL [8], only about quarter spherical grids are preserved after pruning [1]. For the sparse measurements, our method behaves more like an analytic method, where we can simply infer any given direction, rather than resorting to time-consuming nearest neighbor search and interpolation like rendering directly with measured data. Figure 1 depicts a rendering result of the NeuBRDF model trained on sparse measurements.

It is expensive to calculate the weights according to the great-circle distance. We found that $1/4$ is sufficient for the default sphere resolution ($N_{side} = 64$). However, for the sparse measurements that could trigger sphere grid pruning, we instead use the Euclidean distance for large pixels.

2. Additional Experiments

Anisotropic BRDF. Our sphere data structure is designed to support both isotropic and anisotropic materials. This flexibility owes much to the factorized design and grid pruning. In Fig. 2, we demonstrate the representation quality of anisotropic materials from the RGL database. NBRDF [13] struggles in representing high-frequency changes in some anisotropic BRDFs (such as MORPHO-MELENAUS) due to the low frequency spectral bias [12] of the pure-MLP in NBRDF. In contrast, benefiting from the well-characterized *bidirection encoding*, our method can reproduce the high-frequency reflectance distribution effective.

Sparse Measurements. Rendering a sparse measured BRDF directly may encounter visual artifact, and a time-consuming nearest neighbor search is necessary for rendering smooth results. In order to evaluate the scalability to sparse measurement of NeuBRDF, we employ sparse BRDFs database of special coating [5] (hundreds of samples) and sparse SVBRDF of UBO2019 [9] (100 samples per pixel). The rendering results are shown in Fig. 1. The smooth surface reflectance benefits from the inherent hierarchy of HEALPix, which enables the adaptive grid pruning for arbitrary sparse measurement.

Shared Codebook. The neural reflectance primitives stored in a standalone codebook are independent of the per-instance sphere feature-grid, which offers the opportunity for maximizing the memory-quality by sharing codebook between material instances. We illustrate the representation quality in Fig. 3: (a) shows the degeneration of rendering quality as the number of instances sharing one codebook increase; (b) shows the comparison of a cluster of materials represented with and without shared codebook. Figure 8 (top right, in main article) illustrates the overall degeneration of MERL when sharing codebook, evaluated on the raw reflectance data.

[1] For isotropic materials, we assign the $\boldsymbol{\omega}_o.\phi$ to azimuth difference with the range $[0, \pi]$ and leave $\boldsymbol{\omega}_i.\phi = 0$. Therefore, after pruning, about half pixels are kept in south hemisphere and only pixels surrounding 0 longitude line are kept in north hemisphere.

Figure 4. **More results rendered with our NeuBRDF.**

Performance. The inference speed of our neural reflectance model is roughly agnostic to the material. More results rendered with our reflectance model are shown in Fig. 4.

3. Discussions

MLP Decoder. From the perspective of implicit neural functions (INR), the difference between parametric-encoding methods (*e.g.*, instant-NGP [10]) and ours is obvious in the intention of MLP. Conceptually, the MLP in our framework is used to parse/decode the factorized components (and compensate the losses caused by factorization). Whereas for methods with parametric encoding, the MLPs are typically used for local geometric feature decoding [11], multi-resolution aggregation [14], or handling non-Lambertian effect [10] for neural radiance application.

Importance of Iso-Latitude. For the commonly used Fibonacci sphere pixelation, it's difficult to get a hemisphere since it violates the property of iso-latitude and thus there is no explicit equator. Furthermore, the features (shared by four quadrilaterals) near the equator are trained on the samples with imbalanced polar angle distribution due to the non-iso-latitude pixelation. Because of the changes of Fresnel term around equator ($\theta \sim \pi/2$) are often larger than those around poles ($\theta \sim 0$), the reflectance changes severely, which further overwhelms the shared features.

Hemisphere *v.s.* Projected Hemisphere or Cubemap. Essentially, the projected hemisphere is the same as the hemisphere surface: one uses (r, ϕ), and the other uses (θ, ϕ) to express a point on the sphere. The grid discretization on projected hemisphere should also meet the three requirements: spherical uniform distribution, fast indexing, and hierarchy. Similarly, for a cubemap, when a uniform grid on a square is mapped to the sphere, the grid is not uniformly distributed on the sphere.

The Necessity of the Primitives. The neural reflectance primitive (*i.e.* vectors in codebook) is designed to reduce the memory footprint. Even if only one material is represented, the lack of this primitive solution can still result in significant memory overhead. The primitive sharing is designed to maximize the quality-cost tradeoff. A pre-cluster provides an *optional* clue about which materials can form a shared codebook for better results, which, in other words, means that this step can be omitted. With minor modification of the convolutional auto-encoder, the material cluster is applicable to both of MERL, RGL, and UBO.

Ours *vs.* Neural Biplane [4]: They encode the half-vector using a feature grid and leave the difference vector for the MLP to approximate, thus requiring a heavy MLP decoder (FLOPs: Ours-10.4k *vs.* 350k). Additionally, they focus on BTF, while we aim at a all-purpose representation.

Ours *vs.* Real-Time Neural Appearance Models [16]: They only encode the spatial variation (textures) and leave the $\boldsymbol{\omega}_i, \boldsymbol{\omega}_o$ for the MLP to approximate. Neither of these two recent methods include spherical encoding. In contrast, the proposed spherical parametric encoding allows us to use a smaller MLP and achieves real-time performance.

Multiresolution Sphere Grid. The hierarchy is primarily used to support post grid pruning in our original design, for isotropic materials and sparse measurements. Interestingly, we try the multiresolution representation, and find better results returned in our experiments. However, the behaviour of *ang2pix* is not as straightforward as indexing a multi-resolution octree, where One-*ang2pix*-Call Multiresolution-Pixel-Return requires extra computation or memory cost. As a result, multiresolution sphere grid incurs more evaluation cost, for which we finally chose the simplest single level. Nevertheless, further studies on the multiresolution structure is promising, based on a better *ang2pix* strategy.

References

[1] Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013. 1

[2] Jonathan Dupuy and Wenzel Jakob. An adaptive parameterization for efficient material acquisition and rendering. *ACM Transactions on graphics (TOG)*, 37(6):1–14, 2018. 2

[3] Jiahui Fan, Beibei Wang, Milos Hasan, Jian Yang, and Ling-Qi Yan. Neural layered brdfs. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–8, 2022. 1

[4] Jiahui Fan, Beibei Wang, Milos Hasan, Jian Yang, and Ling-Qi Yan. Neural biplane representation for btf rendering and acquisition. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 3

[5] Alejandro Ferrero, Esther Perales, Ana M Rabal, J Campos, Francisco Miguel Martínez-Verdú, Elizabet Chorro, and A Pons. Color representation and interpretation of special effect coatings. *JOSA A*, 31(2):436–447, 2014. 1, 2

[6] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. *arXiv preprint arXiv:2103.13630*, 2021. 1

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[8] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Transactions on Graphics*, 22(3):759–769, 2003. 2

[9] Sebastian Merzbach, Max Hermann, Martin Rump, and Reinhard Klein. Learned fitting of spatially varying brdfs. *Computer Graphics Forum*, 38(4), 2019. 1, 2

[10] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022. 3

[11] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision*, pages 523–540. Springer, 2020. 3

[12] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019. 2

[13] Alejandro Sztrajman, Gilles Rainer, Tobias Ritschel, and Tim Weyrich. Neural brdf representation and importance sampling. In *Computer Graphics Forum*, pages 332–346. Wiley Online Library, 2021. 1, 2

[14] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11367, 2021. 2, 3

[15] Towaki Takikawa, Alex Evans, Jonathan Tremblay, Thomas Müller, Morgan McGuire, Alec Jacobson, and Sanja Fidler. Variable bitrate neural fields. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 1

[16] Tizian Zeltner, Fabrice Rousselle, Andrea Weidlich, Petrik Clarberg, Jan Novák, Benedikt Bitterli, Alex Evans, Tomáš Davidovič, Simon Kallweit, and Aaron Lefohn. Real-time neural appearance models. *arXiv preprint arXiv:2305.02678*, 2023. 3