Real-Time Neural BRDF with Spherically Distributed Primitives

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

Abstract

We propose a neural reflectance model (NeuBRDF) that offers highly versatile material representation, yet with light memory and neural computation consumption towards achieving real-time rendering. The results depicted in Fig. 1, rendered at full HD resolution on a contemporary desktop machine, demonstrate that our system achieves real-time performance with a wide variety of appearances, which is approached by the following two designs. Firstly, recognizing that the bidirectional reflectance is distributed in a sparse high-dimensional space, we propose to project the BRDF into two low-dimensional components, i.e. two hemisphere feature-grids for incoming and outgoing directions, respectively. Secondly, we distribute learnable neural reflectance primitives on our highly-tailored spherical surface grid. These primitives offer informative features for each hemisphere component and reduce the complexity of the feature learning network, leading to fast evaluation. These primitives are centrally stored in a codebook and can be shared across multiple grids and even across materials, based on low-cost indices stored in material-specific spherical surface grids. Our NeuBRDF, agnostic to the material, provides a unified framework for representing a variety of materials consistently. Comprehensive experimental results on measured BRDF compression, Monte Carlo simulated BRDF acceleration, and extension to spatially varying effects demonstrate the superior quality and generalizability achieved by the proposed scheme.

1. Introduction

Reflectance model is one of the most critical factors that affects the rendering photorealism. A common choice for real-time rendering is hand-crafted analytical BRDF models [3, 39], while they often face challenges in accurately reproducing realistic appearances. In terms of methods that offer high realism, measured BRDF models tend to incur prohibitive storage and transmission overhead, particularly for anisotropic and spatially varying materials. On the other hand, Monte Carlo simulation models rely on tracing a large number of light paths with multiple bounces, which limits their applicability in real-time applications.

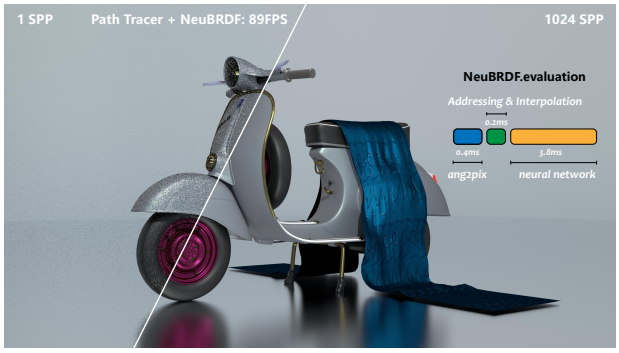


Figure 1. NeuBRDF achieves real-time framerates (i9 12900k and RTX 3090) and can cover a wide range of appearances. We show the time breakdown of evaluating this reflectance model in a 5-maxdepth path tracer at full resolution (1 SPP, 1920×1080).

One appealing approach is to approximate the reflectance using an implicit neural function [27], which offers compact storage and holds promise for acceleration through the recently emerged feature-grid encoding [26, 35]. However, the 4D input of the BRDF poses challenges in constructing such an acceleration structure. We propose to address this challenge through a factorization that decomposes the BRDF into two 3D hemisphere feature-grids, which are decoded by a tiny MLP. Our system is designed according to the following principles:

- **Efficiency.** BRDF evaluation at shading point would occur millions of times at full HD resolution. Therefore, operations on spherical feature-grid need to be straightforward, and the neural network must remain lightweight.
- **Mild memory cost.** The feature-grid is material-specific and often trades memory for fast evaluation. Hence, we should maximize the memory-quality tradeoff.
- **Generalization.** We aim to establish a unified framework capable of accommodating a broad spectrum of materials. We strive for a solution that is agnostic to the materials, whether they are isotropic or anisotropic, measured or simulated, spatially varying or not, and whether the sampling is dense or sparse.

*Equal contribution.
†Corresponding author: Bingbing Ni.

The first principle motivates us to construct an informative feature-grid that reduces the complexity of the feature learning network to a much smaller scale. However, this endeavor encounters challenges in sphere pixelation. Thus, we delve into commonly used sphere discretization methods, extensively discussed and experimented within this paper. The selected HEALPix [10] sphere pixelation is subsequently customized for constructing our reflectance feature-grid. The second principle presents challenges for the methods that directly store separate features at each grid [26, 34]. To maximize the memory utilization, we propose utilizing neural reflectance primitives. These primitives are centrally stored in a codebook and can be shared across multiple grids and even across materials. Consequently, the sphere grid only needs to store low-cost indices pointing to primitives and a material could be approximated by decoding a combination of these primitives. This design strikes a good balance between representation capability and memory compactness. All the aforementioned components in our system are devised in accordance with the third principle. In summary, NeuBRDF enjoys the following good properties:

- NeuBRDF effectively represents measured or offline simulated materials and can be evaluated in **real-time**.
- NeuBRDF offers a **unified framework** for representing a rich diversity of materials consistently.
- **Memory-quality tradeoffs** can be easily attained.

2. Background and Related Work

Our approach is related to previous works in BRDF factorization, feature-based neural representations, neural reflectance models, and sphere pixelation.

BRDF Factorization. Suitable analytic models are not always available for a desired effect, and directly tabulating reflectance data is prohibitive due to the high dimensionality. To overcome this limitation, some pioneers have proposed to factorize the BRDF function into lower dimensional factors, such as spherical harmonics [40], Zernike polynomials [17] and so on. Factors are weighted summed or multiplied [19, 24] to approximate the high dimensional BRDF function. Similarly, Bagher *et al*. [1] define a non-parametric factor microfacet model using tabulated definitions for three functional components (D, G, and F). A commonality among these methods is that they often fall in linear transformation (or with one non-linearity). In contrast, our factorization achieved via auto-decoder learning is capable of modeling complex non-linear functions. In other words, the relationship between existing factorization and ours is similar to that of PCA and auto-encoder.

Feature-based Neural Representation. Feature-based approaches often use differentiable feature primitives to represent the geometric shape. We extend the idea to represent the 4D BRDF. In terms of the geometry or Lambertian effect, feature-based approaches discretize the 3D spatial space into a multi-resolution regular grid and store local features in an octree-based volume [34], a hash table [26], or a dictionary [35, 36]. Representing BRDFs with such feature-based strategy can be challenging to harness, as BRDFs have a higher input dimension (4D) compared to the geometry. And directly expanding the 3D spatial grid into higher dimension to capture BRDFs is prohibitive for its large memory footprint.

Neural Reflectance Models. There are two kinds of approaches for neural-based reflectance modeling. One kind of approaches involve employing one neural network to represent only one material. For example, Sztrajman *et al*. [33] utilize various lightweight networks to represent BRDFs of different materials. To represent spatially-varying BRDFs (SVBRDFs) or bidirectional texture functions (BTFs), material-specific multi-resolution neural textures [18] are used. Most recently, Fan *et al*. [6] approximate BTFs with functions of 2D spatial coordinates and half-vector coordinates. However, the half-vector poses a great risk of bidirectional reflectance loss and thus requires a heavy MLP decoder for correction. The other kind of approaches involve using a single neural network to represent a variety of materials. However, they may lead to a decline in the quality of representation [31]. To achieve both compact representation and high-quality recovery, CNN-based autoencoder is proposed to be utilized for multiple BRDFs [13], where the compactness and quality is further improved using neural processes by [41]. Recently, Fan *et al*. [5] propose to perform material layering in the neural space with the latent code compressed by a neural network.

Sphere Pixelation. The feature-grid relies on a well-characterized spatial pixelation, such as the octree for the geometric shape representation [34]. Besides, representing materials spurs a demand of exploiting a sphere pixelation to accommodate spherical distributed features. We employ the Hierarchical Equal Area isoLatitude Pixelation (HEALPix) [10], which was designed for efficient and incremental discretization of full-sky maps in application to the satellite missions to measure the cosmic microwave background in astrophysics [30]. It provides a deterministic, uniform, and hierarchical sampling method for the sphere surface. Other sphere pixelation methods, such as longitude-latitude lattice, Layered Sukharev grid [32], and Fibonacci lattice [9], are inadequate in terms of efficiency or effectiveness, and will be discussed in this article.

3. Method

Our goal is to represent bidirectional reflectance distribution function (BRDF) using a neural framework that maps bidirections to reflectance values and can be evaluated in *real-time* with *mild memory cost*. The key idea is to build a neural framework that consists of a low-dimensional tailored feature-grid and a tiny neural network to approximate

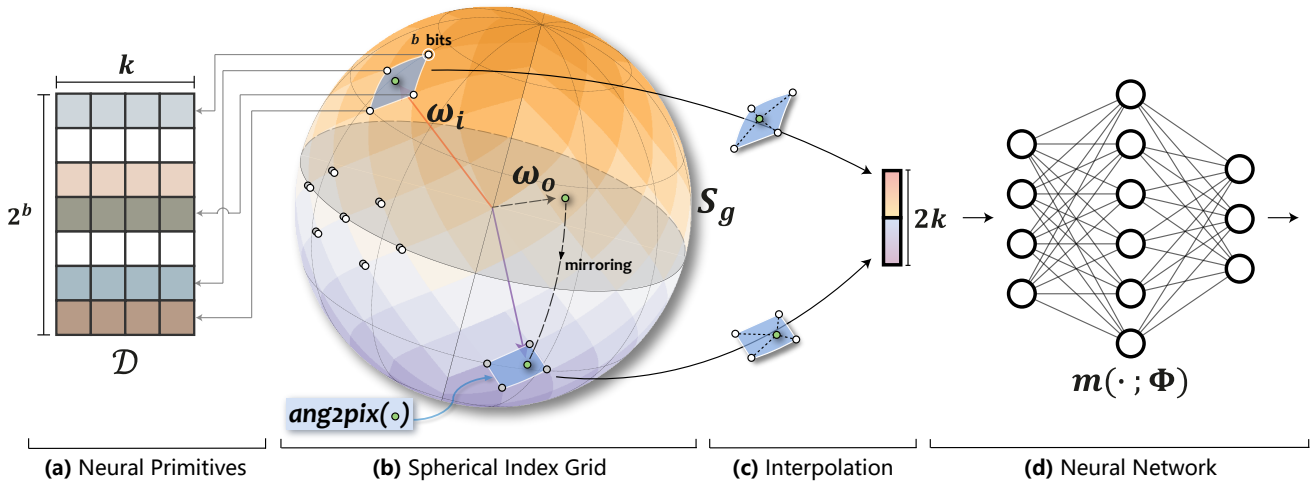


Figure 2. **Illustration of the neural BRDF with spherically distributed primitives.** **(a)** Learnable neural reflectance primitives are centrally stored in a codebook $\mathcal{D}$. **(b)** Given the incident direction $\boldsymbol{\omega}_i$ and outgoing $\boldsymbol{\omega}_o$, we find the surrounding pixels independently on a spherical index grid $\mathcal{S}_g$. For all resulting corner indices, we look up the corresponding k-dimensional neural primitive from the codebook. **(c)** Interpolating them according to the relative position of $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_o$ within the respective curvilinear quadrilateral pixels. **(d)** A tiny MLP takes the concatenated features and outputs final bidirectional reflectance value.

the high-dimensional bidirectional reflectance distributions. In this section, we discuss the algorithmic choices that are key to satisfy the design principles outlined in Sec. 1.

3.1. Algorithm Overview

Evaluating the NeuBRDF consists of a **query** of spherical index grid $\mathcal{S}_g$ by directions $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_o$, **lookup** of the codebook $\mathcal{D}$, and **inference** of a tiny MLP; see Fig. 2.

Fast Evaluation. To enable effective and efficient BRDF evaluation at rendering, we leverage the implicit field based on the *feature-grid*. In contrast to the *global* methods (*e.g.* NeRF [25]) that consist entirely of an MLP, the feature-grid allows the use of a much smaller network. However, it is prohibitive to expand the commonly used 3D spatial grid (*e.g.* Instant-NGP [26]) in scene representation directly into higher dimensions to capture BRDFs, since a 3D one already incurs a large memory footprint.

We consider that the bidirectional reflectance is distributed in a very sparse high-dimensional space, and therefore propose to factorize the BRDF into low-dimensional components. Concretely, we project the 4D BRDF onto two compact 3D hemispherical surface feature-grids along the outgoing $\boldsymbol{\omega}_o$ and incident $\boldsymbol{\omega}_i$ directions. These two factors are used to compute the final reflectance via a tiny neural network, leading to efficient evaluation.

Memory Policy. A tradeoff of feature-grid representations is that they can be quickly evaluated, but typically have a large memory overhead to cache the bulky features. Although our factorization-based solution offers dimensionality reduction, caching the 3D hemispherical features still incurs significant overhead. Furthermore, the learned feature-grid is instance-specific, which means that the memory cost grows with the number of materials in the rendering scenes.

To this end, we incorporate the vector quantization (VQ) compression technique. The features on the hemisphere surface grid points are replaced with indices of a learnable codebook [35]. Notably, we could further maximize the memory-quality tradeoff by reusing one codebook for an arbitrary number of materials, where each material has an exclusive indices grid. We therefore term the prototype vectors in the codebook the *neural reflectance primitives*. The primitives, indices, and tiny MLP network are trained jointly in an end-to-end manner, following [35].

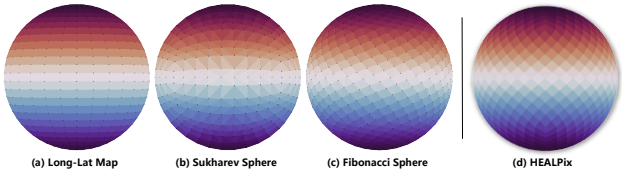


Figure 3. **Comparison among different sphere pixelations.**

3.2. Spherical Feature-Grid

Neural implicit fields in previous works are mainly used to represent low-dimensional signals, such as images and SDFs, thus typically a 3D grid is sufficient for the variants that base on feature-grid. Even for the applications of representing 5D radiance fields, the feature-grids are still built in 3D space, *i.e.* only the geometry and Lambertian effect are involved, where the non-Lambertian effect is represented via additional spherical harmonics [8] or the subsequent MLP [26]. In our case, the 4D BRDF is projected along the outgoing $\boldsymbol{\omega}_o$ and incident $\boldsymbol{\omega}_i$ directions, resulting in two factors that are used to compute reflectance value.

Each factor is represented by features distributed on a hemisphere surface $\mathcal{H}$, where the features are located in some discretized grid. Although there are several spherical discretization methods, whether they meet the needs of real-time BRDF remains to be discussed. Here we explore one structure that is appropriate for BRDF.

3.2.1 Principles and Choices

Discretizing the sphere surface is much more complicated than that of a cube volume used in geometry representation, in terms of uniform pixelation and fast pixel indexing, both of which are naturally achieved by the cube volume with regular discretization but pose challenges to the sphere surface. To achieve the post pruning for sparse measurements, the property of hierarchy is required. That is, unsupervised pixels should be replaced with their parent (macro) pixel, resulting in a multi-resolution sparse grid at runtime, which is important for isotropic reflectance and sparse measurements. Together, the sphere discretization should meet the following main requirements:

- **Uniform Pixelation.** The sphere pixelation should be agnostic to materials and sampling strategies.
- **Fast Indexing.** Real-time rendering requires fast indexing of sphere pixels in arbitrary direction.
- **Hierarchy.** For sparse measurements, a post grid pruning is necessary. The intrinsic hierarchy is required to support multi-resolution sparse grids.

There are several choices for sphere discretization, such as longitude-latitude lattice, Layered Sukharev grid [32], Fibonacci lattice [9], and Hierarchical Equal Area isoLatitude Pixelation (HEALPix) [10]. Figure 3 illustrates the comparison of these choices. The simplest longitude-latitude lattice is suitable for indexing which is directly achieved by rounding up and down, but it arranges too many grid points on the polar cap. The Sukharev grid also fails to achieve uniform pixelation. Although the Fibonacci lattice can evenly distribute points on a sphere surface, it cannot locate the pixel for a given direction analytically and requires an expensive nearest neighbor search, which prevents it from being used in real-time rendering. In contrast, HEALPix satisfies all the aforementioned requirements; summarized in Tab. 1. We therefore employ HEALPix as the underlying sphere grid of our neural BRDF framework.

3.2.2 HEALPix-Based Data Structure

HEALPix tessellates a sphere surface into equal area curvilinear quadrilaterals. The base resolution comprises 12 pixels in three rings around the poles and equator. The resolution of the grid is determined by only one hyperparameter N_{side}, which defines the number of divisions along the side of a base-resolution pixel. As a result, a HEALPix map has $12N_{side}^2$ pixels and $12N_{side}^2+2$ grid points, where the pixels have the same area.

Pixelation	Uniform	Indexing	Hierarchy	Iso-Lat
Long-Lat	No	< 0.5ms	No	Yes
Sukharev	No	~ 5ms	No	No
Fibonacci	Yes	~ 5ms	No	No
HEALPix	Yes	< 0.5ms	Yes	Yes

Table 1. Comparison of pixelation methods. **Indexing**: time consumption of obtaining the corresponding pixel p given bidirection $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_o$ for a 1920×1080 frame. **Iso-Lat**itude: an essential factor for consistent modeling of the Fresnel term of reflectance at various azimuth angles.

Unlike prior work that used HEALPix for cosmological applications [30], we arrange features to points at grid corner instead of the pixel center; Figure 4 shows the comparison. This design reduces the computation cost of pixel indexing for four nearest neighboring features. Although arranging features at the grid center has a modest indexing cost for nine nearest neighbors, the number of memory accesses doubles due to the excessive number of features.

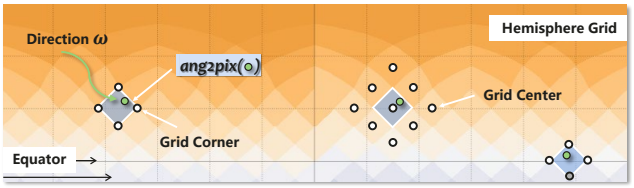


Figure 4. **Illustration of our hemisphere grid in Cartesian coordinate.** Arranging features at grid corner ease the query of nearest neighboring features. A hemisphere in our design includes pixels covering the equator.

Consider a hemisphere covering equator line (*i.e.* a ring adjacent to the equator line in another hemisphere is included). The number of the hemisphere grid points is $n = 6N_{side}^2 + 6N_{side} + 1$ [1]. We denote a hemispherical feature-grid as $\mathcal{H}_g \in \mathbb{R}^{n\times k}$ with feature dimension k.

Given a query direction $\boldsymbol{\omega} = (\theta, \phi)$, where $\theta \in [0, \pi]$ is the colatitude in radians measured from the north pole and ϕ is the longitude in radians measured eastward. ***(i)*** We first find the curvilinear quadrilateral p to which the direction $\boldsymbol{\omega}$ belongs, through the fast analytic indexing $ang2pix$. We refer the readers to the reference [10] for more details about HEALPix. ***(ii)*** Then we extract the features $\boldsymbol{z}_p^t \in \mathcal{H}_g$ at four corners of the quadrilateral pixel p, where $t \in \{1, 2, 3, 4\}$ is the local indices. Meanwhile, we fetch the corresponding spherical coordinates $\boldsymbol{\omega}_p^t$ to compute the interpolation weights w_p^t during training, where the weights are assigned with $1/4$ at runtime for efficiency:

$$w_p^t = \begin{cases} w_p^t / \sum_{j=1}^4 w_p^j, & \text{where } w_p^t = 1/d_p^t, \quad \text{Training}, \\ 1/4, & \text{Runtime}, \end{cases}$$

where d_p^t is the great-circle distance between the spherical coordinates of $\boldsymbol{\omega}_p^t$ and $\boldsymbol{\omega}$. The query for a hemispherical

[1] A ring in equatorial belt comprises $4N_{side}$ points [10]. The hemisphere excluding equator has $6N_{side}^2 - 2N_{side} + 1$ points.

DATABASE	TYPE	SAMPLES	SOURCE
MERL	BRDF$^{\bullet}$	Dense	Measured
RGL	BRDF$^{\bullet\circ}$	Dense	Analytic
LayeredBRDF	SVBRDF$^{\bullet\circ}$	Dense	Simulated
Special Coatings	BRDF$^{\bullet\circ}$	448/576	Measured
UBO2014	BTF$^{\circ}$	22801	Measured
UBOFAB19	SVBRDF$^{\circ}$	100	Measured

Table 2. Materials used in our experiments. **TYPE:** BRDF, BTF and SVBRDF are included. **SAMPLES:** Sparse measurements are involved to assess the generalization ability of NeuBRDF. **SOURCE:** Both the reflectance measured from real-world, simulated by Monte Carlo methods, and evaluated over analytic models are considered. ($^{\bullet}$ isotropic, $^{\circ}$ anisotropic.)

feature-grid $\mathcal{H}_g$ by $\boldsymbol{\omega}$ can be expressed as:

$$\psi(\boldsymbol{\omega}; \mathcal{H}_g) = \sum_{t=1}^{4} w_p^t \boldsymbol{z}_p^t, \quad \text{where} \quad p = ang2pix(\boldsymbol{\omega}; \mathcal{H}_g).$$

(iii) Recall that this process takes place independently for incident $\boldsymbol{\omega}_i$ and outgoing $\boldsymbol{\omega}_o$ and therefore two hemispherical feature-grids are required for representing these two factors. In practice, we merge the two hemispheres into a sphere that comprises three duplicate rings, the equator and its two nearest adjacent rings. The number of the sphere grid points is $2n$, *i.e.* $12N_{side}^2 + 12N_{side} + 2$. We denote a spherical feature-grid as $\mathcal{S}_g \in \mathbb{R}^{2n \times k}$. Eventually, querying a spherical grid $\mathcal{S}_g$ by $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_o$ is:

$$\hat{\boldsymbol{\omega}}_o = (\pi - \boldsymbol{\omega}_o.\theta, \boldsymbol{\omega}_o.\phi),$$
$$\psi(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o; \mathcal{S}_g) = [\psi(\boldsymbol{\omega}_i; \mathcal{S}_g^+), \psi(\hat{\boldsymbol{\omega}}_o; \mathcal{S}_g^-)], \quad (1)$$

where $[\cdot, \cdot]$ denotes concatenation. The $\mathcal{S}_g^+$ and $\mathcal{S}_g^-$ denote the north and south hemispherical feature-grid, both of which include the equator ring and a nearest adjacent ring to the equator in another hemisphere.

3.2.3 Decoder

In order to lift the partial low-dimensional factors to the original 4D space, we employ a multilayer perceptron (MLP) as a non-linear decoder, where the input to MLP is the result of spherical feature-grid query $\psi(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o; \mathcal{S}_g)$. Then, the BRDF $f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$ of our method is described as:

$$f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \approx m(\psi(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o; \mathcal{S}_g); \Phi), \quad (2)$$

where $m(\cdot; \Phi)$ is the MLP with parameters Φ (including weights and biases). Interestingly, the architecture designed from the perspective of factorizing the 4D BRDF resembles that from the parametric encoding [20, 26, 29, 34]. The additional trainable parameters (beyond weights and biases) allow the use of a tiny MLP without sacrificing representation quality and yield immediate *benefits*: (1) NeuBRDF can be trained to convergence much faster than the method that consists entirely of MLPs; (2) It provides feasibility for applying neural BRDF to real-time rendering applications.

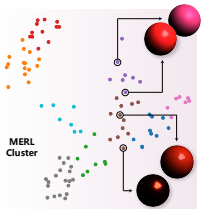
3.3. Neural Reflectance Primitives

Although our factorization solution can effectively realize dimensionality reduction, a spherical feature-grid $\mathcal{S}_g$ with our default setting still requires about 0.8 million parameters ($N_{side} = 64, k = 16$). Following [35], we further compress the storage by incorporating vector-quantization into our framework. Concretely, the $\mathcal{S}_g \in \mathbb{R}^{2n \times k}$ is compressed into an integer vector $\mathcal{S}_g \in \mathbb{Z}^{2n}$ with the range $[0, 2^b - 1]$. The integers are used as *indices* into a codebook matrix $\mathcal{D} \in \mathbb{R}^{2^b \times k}$, where b is the bitwidth to store such an integer. We thus now name $\mathcal{S}_g$ the spherical index grid. The BRDF $f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o)$ is then computed as:

$$f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \approx m(\psi(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o; \mathcal{D}[\mathcal{S}_g]); \Phi), \quad (3)$$

where $[\cdot]$ is the indexing operation and $\mathcal{D}[\mathcal{S}_g]$ denotes lookup. (The training of these indices is detailed in suppmat.) For *fp16* storage, this gives us a compression ratio of $16 \cdot 2n \cdot k / (2n \cdot b + 16 \cdot k \cdot 2^b)$, which can be orders of magnitude when b is small and n is large.

Notably, we provide an option to maximize the quality-cost tradeoff for deploying NeuBRDF into memory constrained devices. Consider that both the aforementioned codebook and spherical index grid are material-specific. Here, our aim is to learn a set of *neural reflectance primitives* that are shared across multiple materials with similar characteristics. Specifically, we first employ a convolutional auto-encoder with bottleneck architecture, of which the objective is to reconstruct the raw reflectance data. The information bottleneck prompts the clustering of similar materials. Thus, a clustering algorithm such as K-Means is then applied to the learned bottleneck codes. For a cluster with t materials, we instantiate an MLP, a codebook storing the primitives, and t spherical index grids $\mathcal{S}_g$, *i.e.* only the low-cost integer indices are exclusive. Still, both of these modules are trained jointly.



3.4. Bidirectional Texture Function

A bidirectional texture function (BTF) is a function that takes an extra 2D location coordinate $\boldsymbol{u} \in \mathbb{R}^2$ as input, in addition to the incident and outgoing directions. To represent the spatially-varying effects, we propose using the neural texture [37] as a *plug-in* for our NeuBRDF. Specifically, we define the neural texture as $\mathcal{T} \in \mathbb{R}^{h \times w \times c}$, where c is the texture channel. The neural texture lookup $\varphi(\boldsymbol{u}; \mathcal{T})$ is achieved by grid sampling, where $\boldsymbol{u}$ is the UV coordinate at the shading point. The resulting texture feature, together with the reflectance primitives from codebook, are fed to the MLP $m(\cdot, \Phi)$, which predicts the BTF value:

$$f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{u}) \approx m([\psi(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o; \mathcal{D}[\mathcal{S}_g]), \varphi(\boldsymbol{u}; \mathcal{T})]; \Phi). \quad (4)$$

Note that a mipmap neural texture [18] is also applicable with our NeuBRDF.

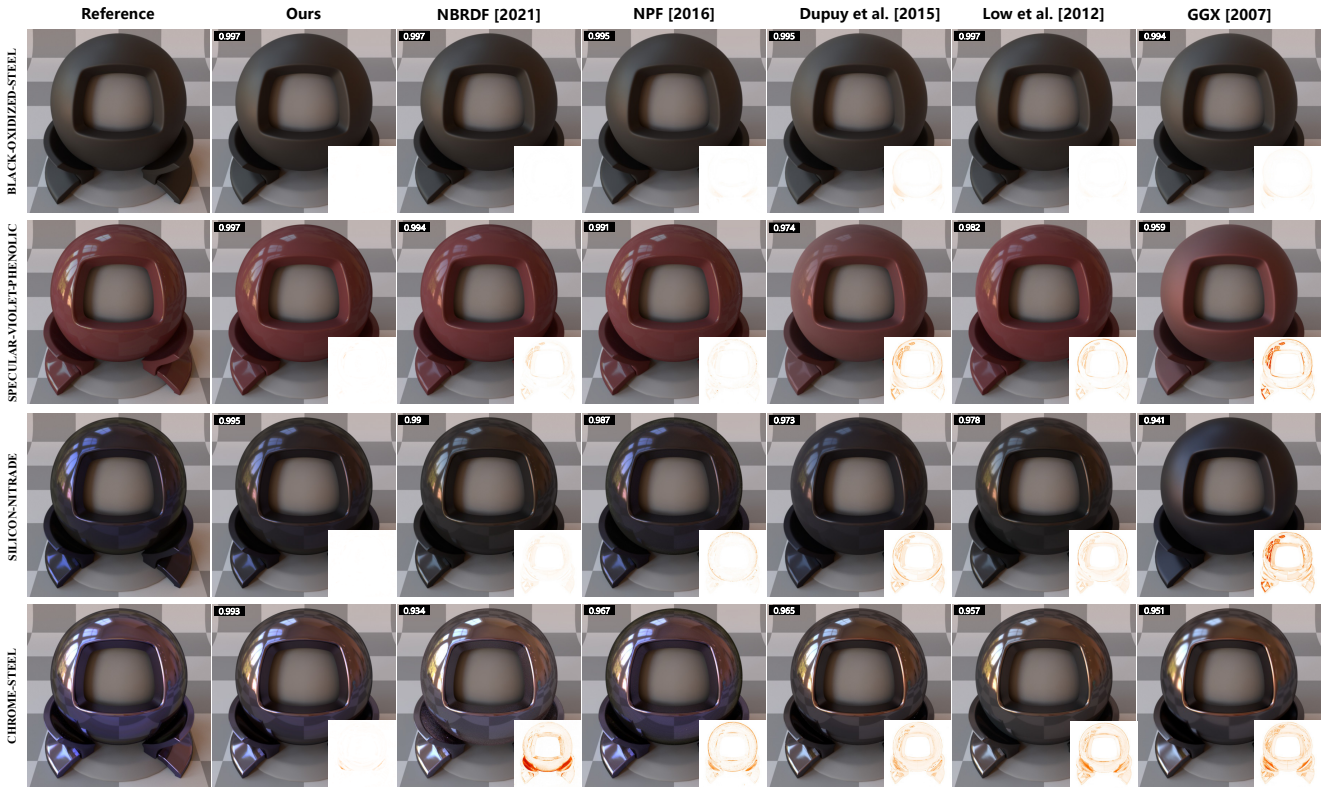


Figure 5. **Quality comparison in fitting reflectance models to MERL [23] measurements.** The analytical microfacet reflectance models, including GGX [39] and Low *et al*. [21], struggle in representing the highly frequency changes (SILICON-NITRADE) and often lead to color shift (CHROME-STEEL). Compared with the previous factorization-based methods, NPF [1] and Dupuy *et al*. [4], where the NPF demonstrates fair visual effect, our method shows better qualitative results. We show the SSIM value in top-left corner and SSIM per-pixel value in bottom-right (the redder the color, the smaller the SSIM), following NBRDF [33] for fair comparison.

4. Experiments

We implement the NeuBRDF training in PyTorch [28], and deploy the runtime version in Falcor [16] for real-time testing (Fig. 1) and in Mitsuba [14] for fair comparison with some methods implemented in this framework. We use the following model configuration unless otherwise specified: $N_{side} = 64, k = 16, b = 9$ and the MLP has 2 hidden layers with 64 neurons. More implementation details are shown in suppmat.

Datasets. To demonstrate the versatility of NeuBRDF, we extensively experiment on various materials. The dataset used in our experiments are listed in Tab. 2, including isotropic BRDFs, anisotropic BRDFs, and spatially varying BRDFs, where the data sources are chosen deliberately to span measured, analytic, and simulated material. We also select some very sparse measurement (such as the Special Coating [7]) to demonstrate the generalizability of NeuBRDF and the effectiveness of sphere grid pruning.

Baselines. We compare our method to several strong BRDF baselines: ***(i)*** the usual parametrized microfacet model GGX [39] and the state-of-the-art one proposed by Low *et al*. [21]; ***(ii)*** the recent non-parametric methods relied on factorization [1, 4]; ***(iii)*** neural approaches for BRDF [33], Layered BRDF [5], and BTF [6]. Furthermore, to demonstrate the application of accelerating physically-based simulation, we employ layered material simulator [12] and volume rendering using microflake phase function to generate realistic and complex appearances.

Methods	MAE ↓	RMSE ↓	SSIM ↑
GGX [39]	0.0189	0.0206	0.969
Low *et al*. [21]	0.0080	0.0088	0.986
Dupuy *et al*. [4]	0.0174	0.0190	0.976
NPF [1]	0.0056	0.0062	0.990
NBRDF [33]	0.0028	0.0033	0.995
Ours	**0.0017**	**0.0031**	**0.996**

Table 3. **Average image-based losses** of representation methods from Fig. 5 over all MERL materials.

4.1. Quality Validation

It's a standard practice to evaluate BRDF models or tune parameters of analytical models on measured reflectance data. The fitting results on MERL measurements are demonstrated in Fig. 5. Our method achieves the most faithful rendering among both diffuse, glossy, and specular materials. The quantitative results on the whole MERL measure-

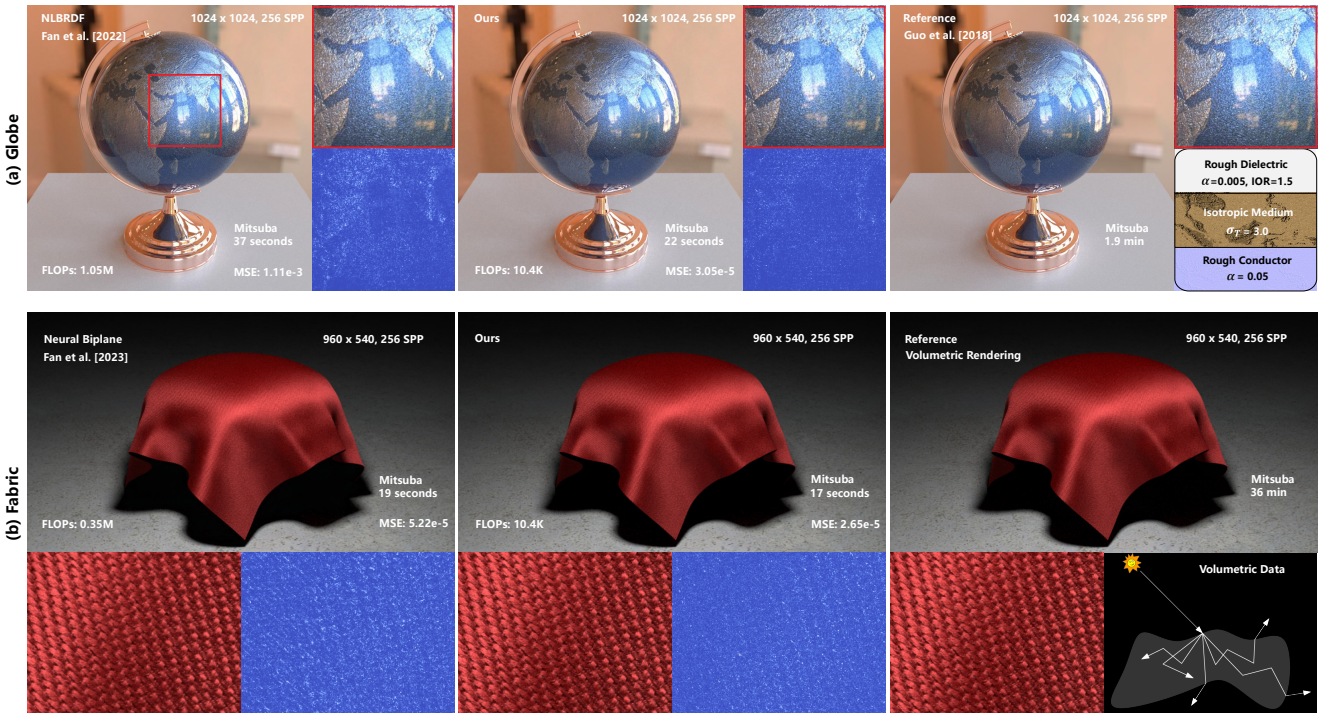


Figure 6. **Accelerating physically based simulation.** We demonstrate the comparison with the recent state-of-the-art neural approach, including NLBRDF [5] and Neural Biplane [6]. **(a) Simulation from Layered BRDF [12].** Our method requires a quarter of the time consumption compared to the simulation. **(b) Simulation from volume rendering.** The fabric is rendered from micro-CT volumetric data, using the microflake phase function. Our method requires almost one percent of the time compared to the simulation. Additionally, our method demonstrates superior results and performances compared with SOTA neural approaches in both scenes. We implement our method in Mitsuba (an offline renderer) [14] for fair comparison.

ment are shown in Tab. 3. We report the mean absolute error (MAE), root mean squared error (RMSE), and structural similarity index measure (SSIM) on the rendered image, following NBRDF [33]. Results on anisotropic materials, sparse measurements, and multiple instances sharing one codebook are shown in suppmat.

4.2. Extension Applications

Accelerating Physically Based Simulation. Reflectance models designed for photorealistic rendering are often layered [11, 12, 15]. By constructing various interface layers and internal media, these physical methods can cover a wide range of appearances. Evaluating such models is typically achieved through light transport between layers, simulated by Monte Carlo methods. Since our method is agnostic to the internal light-layer interactions, it can be directly used for accelerating these physical-based methods.

In many complex scenes, these layered BRDF methods often use albedo texture, normal map, or spatially varying scattering/absorption coefficient in internal layers. As a compatible solution, we still use the plug-and-play neural texture $\mathcal{T}$. The neural texture can be considered as a blend of multiple internal textures. With the expense of building the reflectance data (hours) and training ($<$ 1 hour), our method can drastically reduce the rendering cost; see Fig. 6 for an illustration. Our approach performs surprisingly well in such a complicated scene. We use the default base framework together with a $800 \times 400 \times 8$ neural texture to capture the spatially varying effects. These results demonstrate that our approach generalizes well to complex materials and can produce highly realistic results in a much smaller computation resource compared with either the physical simulations or the neural network approaches.

BTF Compression. Tabulating an isotropic BRDF consumes tens of megabytes memory and becomes prohibitive for spatially varying material, impeding the network transfer and runtime loading. Our framework can be easily extended for the compression purpose; the BTF compression results are illustrated in Fig. 7. The convincing results of NeuBRDF with a naive neural texture plugin suggest the feasibility of decoupling spatially varying effect and bidirectional $\omega_{i,o}$, which is also proven in neural biplane [6]. Similarly, neural texture compression (NTC) [38] is proposed to compress the texture using implicit neural functions. However, they only operate on the image space and rely on an extra reflectance model to consume the decoded image textures. Instead, we consider the reflectance model itself. Likewise, the highly specific neural texture proposed in NTC is also compatible with our framework.

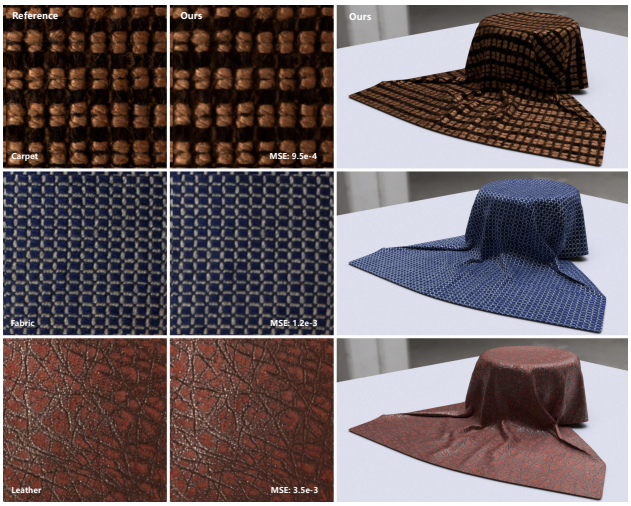


Figure 7. **BTF compression on UBO2014.** A BTF contains $400 \times 400 \times 22801$ RGB measurements, which occupancy $\sim$ 500 MB storage (compressed). Simply with a $128 \times 128 \times 4$ neural texture ($\sim$ 131KB), NeuBRDF ($\sim$ 95KB) can capture these spatially varying effects, resulting in a compression ratio of thousands of times. NeuBRDF and neural texture $\mathcal{T}$ use *fp*16 by default. (The scene in the right column is from [22].)

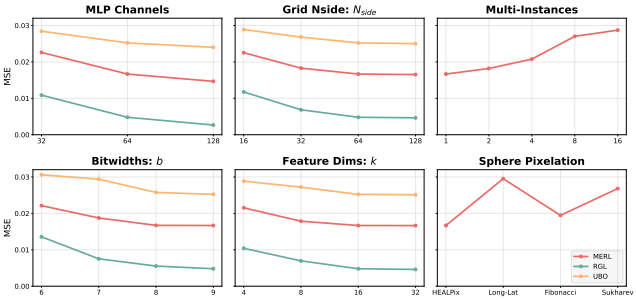


Figure 8. **Ablation studies on different model configurations.** MSE is computed on the raw data. Multiple material instances are selected from a same cluster. Sphere pixelations have similar number of grid points for fair comparison.

4.3. Ablation Studies

Model Configuration. We investigate the impact of MLP channels, feature-grid resolution (N_{side}), codebook bitwidth, and codebook feature dimension; shown in Fig. 8 (left two columns). In order to fully validate the model at different configurations, we report the MSE on the raw reflectance data rather than the rendered image that not yields full coverage of the bidirection space.

Quantization. The primitive indexing is exactly the same as vector quantization in the perspective of quantization. To this end, we conduct ablation study on ***(i)*** the model without codebook, where the sphere grid allocates bulky features instead of lightweight integer indices, and ***(ii)*** the model with low-bit quantization. The MSE results on MERL raw data and the required parameters are shown in Tab. 4. Our choice consumes less memory and performs better than low-bit quantization. Additionally, the use of codebook can alleviate the overfitting drawback of feature-grid and allows sharing primitives across instances. We implement low-bit quantization by simulating quantization along the line of neural image compression [2, 38].

	MSE ↓	Memory Cost ↓	Ratio ↓
w/o Quantization	**0.0160**	1.60MB	1
w Low-bit	0.0189	0.40MB	0.25
w CodeBook (ours)	0.0167	**0.07MB**	**0.045**

Table 4. **Quantization studies.** The codebook is also a quantization technique, for which we compare it with low-bit (4b) quantization. Memory costs are computed in *fp*16 except the low-bit quantization, and the MLP is not taken into account.

Sphere Pixelation. The design of sphere pixelation is critical to the success of NeuBRDF. We demonstrate representing quality in Fig. 8 (bottom right), where the grid resolutions are managed as close as possible for each pixelation. Both of the other pixelations are manually tailored for BRDF representation, which are similar with that of HEALPix described in Sec. 3.2.2. Regardless of the exclusive merit of hierarchy of HEALPix, the model with HEALPix achieves best representation results. Although the Fibonacci pixelation is comparable in terms of MSE, the indexing time of Fibonacci is about 5ms in contrast to the 0.5ms of HEALPix. We anticipate that the HEALPix, with our adaptation for BRDFs, can serve as a fundamental data structure in material representation.

Runtime Time Breakdown. We implement our NeuBRDF in Falcor [16] for real-time testing. Figure 1 shows the evaluation time breakdown. Evaluating our reflectance models takes about 4.5ms in a 5-maxdepth path tracer at full resolution (1 SPP, 1920×1080), which is agnostic to materials.

5. Conclusions

In this article, we propose a neural reflectance model that is a choice for achieving realistic appearance in real-time rendering applications. The proposed method is also a unified framework that can represent a variety of materials, which can be used for measured data compression, fitting sparse measurements, physical-based methods acceleration and so on. The key components to the effectiveness consist of the delicately designed sphere data structure and neural reflectance primitives. In the future, we will exploit more material application based on such architecture, *i.e.* spherically distributed primitives with a tiny-MLP decoder, such as material editing and material acquisition.

6. Acknowledgement

This work was supported by National Science Foundation of China (U20B2072, 61976137). This work was also partly supported by SJTU Medical Engineering Cross Research Grant YG2021ZD18.

References

[1] Mahdi M Bagher, John Snyder, and Derek Nowrouzezahrai. A non-parametric factor microfacet model for isotropic brdfs. *ACM Transactions on Graphics (TOG)*, 35(5):1–16, 2016. 2, 6

[2] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016. 8

[3] Brent Burley and Walt Disney Animation Studios. Physically-based shading at disney. In *ACM SIGGRAPH*, pages 1–7. vol. 2012, 2012. 1

[4] Jonathan Dupuy, Eric Heitz, Jean-Claude Iehl, Pierre Poulin, and Victor Ostromoukhov. Extracting microfacet-based brdf parameters from arbitrary materials with power iterations. In *Computer Graphics Forum*, pages 21–30. Wiley Online Library, 2015. 6

[5] 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. 2, 6, 7

[6] 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. 2, 6, 7

[7] 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. 6

[8] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5501–5510, 2022. 3

[9] Álvaro González. Measurement of areas on a sphere using fibonacci and latitude–longitude lattices. *Mathematical Geosciences*, 42(1):49–64, 2010. 2, 4

[10] Krzysztof M Gorski, Eric Hivon, Anthony J Banday, Benjamin D Wandelt, Frode K Hansen, Mstvos Reinecke, and Matthia Bartelmann. Healpix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere. *The Astrophysical Journal*, 622(2):759, 2005. 2, 4

[11] Ibón Guillén, Julio Marco, Diego Gutierrez, Wenzel Jakob, and Adrian Jarabo. A general framework for pearlescent materials. *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020. 7

[12] Yu Guo, Miloš Hašan, and Shuang Zhao. Position-free monte carlo simulation for arbitrary layered bsdfs. *ACM Transactions on Graphics (ToG)*, 37(6):1–14, 2018. 6, 7

[13] Bingyang Hu, Jie Guo, Yanjun Chen, Mengtian Li, and Yanwen Guo. Deepbrdf: A deep representation for manipulating measured brdf. In *Computer Graphics Forum*, pages 157–166. Wiley Online Library, 2020. 2

[14] Wenzel Jakob. Mitsuba renderer, 2010. https://mitsuba-renderer.org. 6, 7

[15] Wenzel Jakob, Eugene D"Eon, Otto Jakob, and Steve Marschner. A comprehensive framework for rendering layered materials. *ACM Transactions on Graphics*, 33(4):1–14, 2014. 7

[16] Simon Kallweit, Petrik Clarberg, Craig Kolb, Tom'aš Davidovič, Kai-Hwa Yao, Theresa Foley, Yong He, Lifan Wu, Lucy Chen, Tomas Akenine-Möller, Chris Wyman, Cyril Crassin, and Nir Benty. The Falcor rendering framework, 2022. https://github.com/NVIDIAGameWorks/Falcor. 6, 8

[17] Jan J Koenderink, Andrea J Van Doorn, and Marigo Stavridi. Bidirectional reflection distribution function expressed in terms of surface scattering modes. In *Computer Vision—ECCV'96: 4th European Conference on Computer Vision Cambridge, UK, April 15–18, 1996 Proceedings Volume II 4*, pages 28–39. Springer, 1996. 2

[18] Alexandr Kuznetsov. Neumip: Multi-resolution neural materials. *ACM Transactions on Graphics (TOG)*, 40(4), 2021. 2, 5

[19] Lutz Latta and Andreas Kolb. Homomorphic factorization of brdf-based lighting computation. *ACM Transactions on Graphics (TOG)*, 21(3):509–516, 2002. 2

[20] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 5

[21] Joakim Löw, Joel Kronander, Anders Ynnerman, and Jonas Unger. Brdf models for accurate and efficient rendering of glossy surfaces. *ACM Transactions on Graphics (TOG)*, 31 (1):1–14, 2012. 6

[22] Ryota Maeda. Custom plugin in python for bidirectional texture function rendering with mitsuba 2, 2021. https://github.com/elerac/btf-rendering. 8

[23] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Transactions on Graphics*, 22(3):759–769, 2003. 6

[24] Michael D McCool, Jason Ang, and Anis Ahmad. Homomorphic factorization of brdfs for high-performance rendering. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 171–178, 2001. 2

[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. *Communications of the ACM*, 65(1):99–106, 2021. 3

[26] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022. 1, 2, 3, 5

[27] 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. 1

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Ad-*

vances in neural information processing systems, 32, 2019. 6

[29] 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. 5

[30] Nathanaël Perraudin, Michaël Defferrard, Tomasz Kacprzak, and Raphael Sgier. Deepsphere: Efficient spherical convolutional neural network with healpix sampling for cosmological applications. *Astronomy and Computing*, 27:130–146, 2019. 2, 4

[31] Gilles Rainer, Abhijeet Ghosh, Wenzel Jakob, and Tim Weyrich. Unified neural encoding of btfs. In *Computer Graphics Forum*, pages 167–178. Wiley Online Library, 2020. 2

[32] Aleksandr G Sukharev. Optimal strategies of the search for an extremum. *USSR Computational Mathematics and Mathematical Physics*, 11(4):119–137, 1971. 2, 4

[33] 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. 2, 6, 7

[34] 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, 5

[35] 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, 2, 3, 5

[36] Towaki Takikawa, Thomas Müller, Merlin Nimier-David, Alex Evans, Sanja Fidler, Alec Jacobson, and Alexander Keller. Compact neural graphics primitives with learned hash probing. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–10, 2023. 2

[37] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 5

[38] Karthik Vaidyanathan, Marco Salvi, Bartlomiej Wronski, Tomas Akenine-Möller, Pontus Ebelin, and Aaron Lefohn. Random-access neural compression of material textures. *arXiv preprint arXiv:2305.17105*, 2023. 7, 8

[39] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 195–206, 2007. 1, 6

[40] Stephen H Westin, James R Arvo, and Kenneth E Torrance. Predicting reflectance functions from complex surfaces. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 255–264, 1992. 2

[41] Chuankun Zheng, Ruzhang Zheng, Rui Wang, Shuang Zhao, and Hujun Bao. A compact representation of measured brdfs using neural processes. *ACM Transactions on Graphics (TOG)*, 41(2):1–15, 2021. 2