

BSDF Importance Baking: A Lightweight Neural Solution to Importance Sampling General Parametric BSDFs

YAOYI BAI, University of California, Santa Barbara, USA
 SONGYIN WU, University of California, Santa Barbara, USA
 ZHENG ZENG, University of California, Santa Barbara, USA
 BEIBEI WANG, Nankai University, China
 LING-QI YAN, University of California, Santa Barbara, USA



Fig. 1. We demonstrate the effectiveness of our work (BSDF sampling network, together with BSDF evaluation and PDF query networks) on *multiple-bounce microfacet conductors* (top) and *layered BxDF* (bottom). (Top) We compare our method with Heitz et al. [2016] on various materials (roughness and colors) with equal sampling weight. Our method shows lower noise levels, despite requiring a longer time for network inference. (Bottom) Comparing with Guo et al. [2018] on several layered materials (different isotropic roughness, index of refraction, attenuation coefficient, and albedo colors) with equal time, our model achieves both higher performance and less noise level.

Parametric Bidirectional Scattering Distribution Functions (BSDFs) are pervasively used because of their flexibility to represent a large variety of material appearances by simply tuning the parameters. While efficient evaluation of parametric BSDFs has been well-studied, high-quality importance sampling techniques for parametric BSDFs are still scarce. Existing sampling strategies either heavily rely on approximations, resulting in high variance, or solely perform sampling on a portion of the whole BSDF slice. Moreover, many of the sampling approaches are specifically paired with certain types of BSDFs. In this paper, we seek an efficient and general way for importance sampling parametric BSDFs. We notice that the nature of importance sampling is the mapping between a uniform distribution and the target distribution. Specifically, when BSDF parameters are given, the mapping that performs

importance sampling on a BSDF slice can be simply recorded as a 2D image that we name as importance map. Following this observation, we accurately precompute the importance maps using a mathematical tool named optimal transport. Then we propose a lightweight neural network to efficiently compress the precomputed importance maps. In this way, we have brought parametric BSDF important sampling to the precomputation stage, avoiding heavy runtime computation. Since this process is similar to light baking where a set of images are precomputed, we name our method importance baking. Together with a BSDF evaluation network and a PDF (probability density function) query network, our method enables full multiple importance sampling (MIS) without any revision to the rendering pipeline. Our method essentially performs perfect importance sampling. Compared with previous methods, we demonstrate reduced noise levels on rendering results with a rich set of appearances, from multiple-bounce microfacet conductors with anisotropic roughness, to layered materials and Disney principled materials.

Authors' addresses: Yaoyi Bai, University of California, Santa Barbara, Santa Barbara, CA, USA, 93106, yaoyibai@cs.ucsb.edu; Songyin Wu, University of California, Santa Barbara, Santa Barbara, CA, USA, 93106, s_wu975@ucsb.edu; Zheng Zeng, University of California, Santa Barbara, Santa Barbara, CA, USA, 93106, zhengzeng@ucsb.edu; Beibei Wang, Nankai University, Tianjin, China, beibei.wang@nankai.edu.cn; Ling-Qi Yan, University of California, Santa Barbara, Santa Barbara, CA, USA, 93106, lingqi@cs.ucsb.edu.

CCS Concepts: • Computing methodologies → Rendering.

Additional Key Words and Phrases: physically based rendering, importance sampling, neural rendering

1 INTRODUCTION

Bidirectional Scattering Distribution Functions (BSDFs) are key to realistic appearances. Among all BSDFs, parametric BSDFs are pervasively used because of their flexibility to represent a large variety of materials' optical properties using a few parameters.

In the modern Monte Carlo rendering framework, BSDFs need not only be evaluated, but also importance sampled. The core of BSDF importance sampling is to choose an incident direction according to a probability density function (PDF) that closely resembles the 2D BSDF slice jointly determined by BSDF parameters and an outgoing direction. However, parametric BSDFs barely have perfect importance sampling strategies. Some BSDFs do not support analytical sampling, e.g., the Disney Principled BRDFs using Generalized-Trowbridge-Reitz (GTR) [Burley 2012; McAuley et al. 2012]. Some solutions tend to sample part of the BSDF slice, e.g., visible normal distribution functions (VNDFs) [Heitz and d'Eon 2014] without taking the Fresnel term into account. Some other solutions use random walk to sample, e.g., position-free layered material [Guo et al. 2018], resulting in perfect importance sampling in theory but also high computational cost and variance.

Recently, neural approaches have been proposed to represent measured/synthetic BRDFs [Fan et al. 2021; Sztrajman et al. 2021; Zheng et al. 2021], which are mostly tied with compression of BSDFs and seldomly study the problem of accurate importance sampling. Recent works that use neural methods for importance sampling do exist, but few work well with the naturally high-dimensional parametric BSDFs, especially those utilizing normalizing flow [Müller et al. 2019; Xie et al. 2019].

In this work, we focus on accurately solving the importance sampling problem for general parametric BSDFs. Our insight is that, importance sampling, regardless of the specific methods (e.g., marginalized inverse transform sampling) to achieve it, is in essence a mapping between a uniform distribution and a target distribution. Specifically, importance sampling a parametric BSDF slice is a 2D to 2D mapping which can be simply recorded as a 2D image that we name as *importance map*. Suppose we can precompute and compress all the importance maps for all combinations of parameters and incident directions, we will be able to bring accurate parametric BSDF importance sampling to the precomputation stage, avoiding heavy runtime computation.

Therefore, the core of general and accurate BSDF importance sampling is to generate high-quality importance maps. The commonly used 2D importance sampling methods, e.g., the marginalized inverse transform sampling or hierarchical sample warping [Clarberg et al. 2005], produce discontinuous importance maps, raising difficulties for both compression and interpolation. To overcome these issues, we introduce the Optimal Transport (OT) theory to this task, which is by nature suitable to provide a smooth mapping between distributions to generate importance maps efficiently. To alleviate the expensive storage cost of importance maps, we propose a lightweight neural network for compression and query. Together with a BSDF evaluation network and a PDF query network, we provide a

complete neural solution for general parametric BSDFs, supporting full multiple importance sampling (MIS) without any revision to the rest of the rendering pipeline.

Since the process of computing and compressing importance maps is similar to light baking, in the sense that a set of images are precomputed and queried during runtime, we name our method *BSDF importance baking*. It essentially performs perfect importance sampling, and can be used for any general parametric BSDFs, even those lacking analytic importance sampling solutions. Compared with previous methods, we demonstrate reduced noise levels on rendering results with a rich set of appearances, spanning a wide range of parametric BSDFs from multiple-bounce microfacet conductors with anisotropic roughness, to layered materials and Disney principled materials.

In summary, our contributions include:

- a novel, accurate and efficient BSDF importance sampling solution for general parametric BSDFs,
- a new theory that connects optimal transport (OT) and BSDF importance sampling to compute *importance maps*, which maps 2D uniform distribution to 2D BSDF slices.
- an application of lightweight neural networks to compress precomputed *importance maps*, as well as optional BSDF evaluation and PDF query networks for comparatively complex BSDFs for a full MIS solution, and
- a database of *importance maps* that collects parametric BSDF importance sampling data for public use, including multiple-bounce microfacet conductors, layered BSDF, as well as Disney principled materials.

2 RELATED WORK

Parametric BSDFs. Parametric BSDFs represent different materials with explicit parameters. Two major groups of parametric BSDFs include empirical models (e.g., [Ashikhmin and Shirley 2001; Phong 1975]) and physically-based models, for example, microfacet models [Blinn 1977] with different normal distribution functions (NDFs), including Beckmann [Beckmann and Spizzichino 1963], GGX [Walter et al. 2007] and Generalized-Trowbridge-Reitz (GTR) [Burley 2012], and their multiple-bounce extensions [Heitz et al. 2016; Wang et al. 2022; Xie and Hanrahan 2018]. Apart from multiple-bounce microfacet models, position-free layered BSDFs [Guo et al. 2018] contain more parameters, such as the number of layers and the properties of the medium between layers. The parameters can also be artist-driven. For example, Disney principled materials [Burley 2012; McAuley et al. 2012] are defined by few intuitive parameters, such as sheen and metallic. In this paper, we focus on providing a general solution to importance sampling parametric BSDFs.

BSDF Importance Sampling. Under the Monte Carlo rendering framework, the BSDF sampling strategy is crucial to variance reduction. However, sampling all the components within the BSDF is non-trivial. In the microfacet model, sampling the NDF gives a good approximation but can produce significant variance when the outgoing direction is from the grazing angle. Heitz et al. [2017; 2018; 2014] reduced sampling variance by sampling the distribution of visible normals (VNDF). However, VNDF sampling only works for NDF that is stretch-invariant. Thus, some NDFs (e.g., GTR) do not

allow for accurate VNDF sampling and only use an approximation. Even worse, when considering multiple scattering in the microfacet model [Heitz et al. 2016], the importance sampling becomes a random walk due to the absence of the closed-form formulation, leading to low performance. BSDF slices can also be approximated by simple lobes, such as two Gaussians [Fan et al. 2022] or Blinn-Phong models [Sztrajman et al. 2021], then BSDF importance sampling can be evaluated by sampling these known models. However, these simple lobes are unlikely to fit complex BSDFs and result in high variance, especially BSDFs with multiple lobes. However, our method computes the actual PDF values, producing less variance.¹

Hierarchical Sample Warping. Clarberg et al. [2005] proposed an efficient and high-quality hierarchical warping technique that maps a uniform distribution to a hierarchical distribution. They applied it to sample environment maps, BSDFs, and their products on-the-fly without evaluating the full integral. However, hierarchical structures break the continuity of distributions and increase the precomputation difficulty. Additionally, dynamic sampling requires that all BSDF slices are known and represented as wavelets. It means that all possible BSDFs need to be tabulated. However, tabulating all data is impossible for our layered BSDFs, which include ten dimensions. We will explain this issue in Sec. 3.3.

Sampling Specific Types of BSDFs. Lawrence et al. [2004] reparameterize BRDFs and decompose them into factored representation to achieve simple and compact importance sampling of analytical and measured BRDFs. However, the decomposition structure can only handle simple BSDFs with mild glossiness or a single lobe. Otherwise, the results are far from accurate. Moreover, the parametric BSDFs investigated in this paper contain tremendously higher dimensions than 4D. Then decomposing these BSDFs becomes entirely not feasible.

Unlike these methods, our method considers 2D slices of the entire BSDF and allows perfect importance sampling. Our model does not require that the BSDF has a closed-form formulation or contains only one lobe, thus can be used for arbitrary parametric BSDF models, including multiple scattering models, layered BSDFs, and Disney principled BSDFs.

Optimal transport in computer graphics. Optimal transport (OT) is a mathematical framework to manipulate distributions [Kantorovich 1942; Monge 1781]. Recent works in computer graphics have applied optimal transport to various fields, such as shape interpolation [Bonnel et al. 2016, 2011; Solomon et al. 2015]. OT is capable of providing good distribution mapping and natural interpolation, but the fact that OT calculates slowly constrains its computer graphics applications. Solutions with approximations enable faster OT calculation, such as Sinkhorn distances [Cuturi 2013], convolutional Wasserstein distances [Solomon et al. 2015], geomloss [Feydy et al. 2019], sliced optimal transport (SOT) [Paulin et al. 2020] and sliced partial optimal transport (SPOT) [Bonnel and Coeurjolly 2019]. However, they are still prohibitively slow in rendering, because computational

¹Note that BSDF is a 3-channel value, but PDF is a single-channel value. In this case, BSDF sampling cannot be perfect since PDF cannot always be the same with R, G and B values. Therefore, given some PDF values (for example, grayscale) and performing importance sampling accordingly, it should be considered the optimal solution for BSDF importance sampling.

resources have already been significantly diluted by the massive number of shading computations running in parallel. To avoid extensive runtime distribution mapping computation, we take advantage of the smooth mapping acquired using OT while entirely steering away from its disadvantage of slow computation by applying it only in the precomputation stage.

Neural network for BSDF representation. Neural networks (NN) have been recently used for BRDF representation on measured materials by representing one spatially-varying BRDF (SVBRDF) or one Bidirectional Texture Function (BTF) per network [Kuznetsov et al. 2021; Rainer et al. 2019], all BRDFs within a unified network [Hu et al. 2020; Rainer et al. 2020; Zheng et al. 2021], or each BRDF as a standalone decoder network [Sztrajman et al. 2021]. Besides compressing the measured BRDFs, Sztrajman et al. [Sztrajman et al. 2021] also support importance sampling by mapping a measured BRDF to an approximate parametric BRDF, but the differences between them result in imperfect importance sampling. Zheng et al. [2021] applied two-layer NICE [Dinh et al. 2014] to sample measured BRDFs. Xie et al. [2019] proposed to use a RealNVP [Dinh et al. 2016] network to learn multiple scattering equivalent NDFs in the slope space, enabling importance sampling. However, their method cannot support the high-dimensional parameter space for parametric BSDFs (e.g., spatially-varying Fresnel and anisotropic roughness) because of the notoriously bulky structure of normalizing flow structures and is considered too slow [Müller et al. 2019] for practical use.

Our method focuses on importance sampling general parametric BSDFs. By completely decoupling the computation of the importance maps to the precomputation stage, we only require using a lightweight neural network to perform compression. Furthermore, since our importance maps are generally smooth thanks to optimal transport, they are naturally suitable for a neural network to compress.

3 BACKGROUND AND ANALYSIS

3.1 Parametric BSDF Rendering

Parametric BSDFs. Parametric BSDFs are explicitly controlled by material property parameters, such as roughness, refractive index, attenuation coefficient, and etc. For example, single-bounce Smith microfacet BRDFs can be analytically written as:

$$f_s(\omega_i, \omega_o) = \frac{F(\omega_i)G(\omega_i, \omega_o)D(\omega_h)}{4|\omega_i \cdot \mathbf{n}||\omega_o \cdot \mathbf{n}|}, \quad (1)$$

where F is the Fresnel term, often approximated with a Schlick reflectance R_0 and an incident direction ω_i . G is the shadowing-masking term accounting for self-occlusion, and D is the normal distribution function (NDF), parameterized on the half vector $\omega_h = \frac{\omega_i + \omega_o}{|\omega_i + \omega_o|}$, as well as roughness α_x and α_y . Given these material parameters and an outgoing direction ω_o , 2D BSDF slices can be easily generated where each pixel records the BSDF radiance value of an incident direction ω_i .

For multiple-bounce Smith microfacet BSDFs where the light may bounce multiple times before exiting the surface as shown in Fig. 2, there is still no analytical expression. Both the BSDF evaluation and importance sampling need heavy random walk simulations.

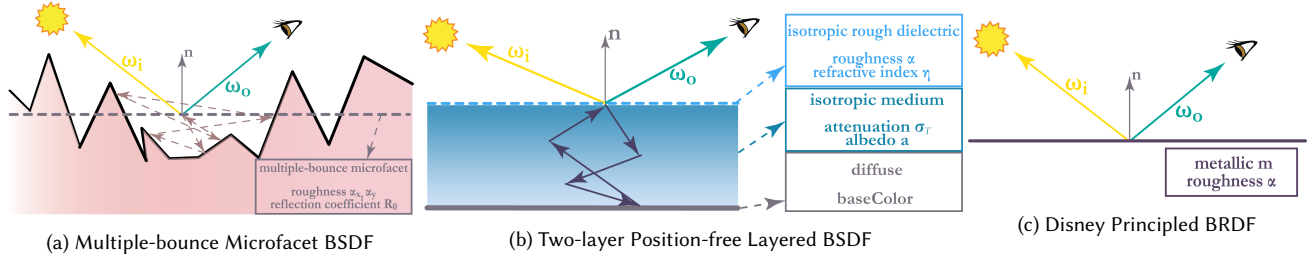


Fig. 2. Several parametric BSDF models investigated in our paper. Note that the incident direction ω_i points towards light sources, and the outgoing direction ω_o points towards the camera.

In this work, we mainly focus on multiple-bounce BSDFs rather than single-bounce BSDFs, since the former are more challenging. Without loss of generality, we focus on GGX NDFs and describe conductors in the main text.

We also study layered BSDFs [Guo et al. 2018], as illustrated in Fig. 2. The configuration for layered BRDFs includes a top layer using a rough dielectric (with roughness α and refractive index η), a bottom layer using a diffuse BRDF and a homogeneous participating media (with attenuation coefficient σ_T and albedo (a_R, a_G, a_B)) in the middle. Also note that there are no constraints about the number of layers or the types of BSDFs in each layer in our method. We merely initiate from a simple layered BSDF model to prove the generalization of our method.

Additionally, we study the Disney principled BSDF [Burley 2012; McAuley et al. 2012]. This artists-friendly model was summarized to be few zero to one intuitive parameters, such as metallic, roughness, specular, anisotropic, sheen, and etc. In this paper, we mainly focus on manipulating metallic m and roughness α values also shown in Fig. 2.

Monte Carlo (MC) integration and importance sampling. The MC method provides an unbiased estimator to any definite integral in the domain Ω with an arbitrary integrand $f(x)$

$$\int_{\Omega} f(x) dx \approx \frac{1}{N} \sum_{k=1}^N \frac{f(X_k)}{p(X_k)} \quad X_k \sim p(x). \quad (2)$$

The MC estimator approximates the integral by drawing samples according to a probability density function (PDF) $p(x)$. The $f(X_k)/p(X_k)$ is known as the *sampling weight*. To minimize the estimation variance, *importance sampling* is desired – the closer between the “shapes” of $f(x)$ and $p(x)$, the lower variance does the MC estimator have.

BSDF importance sampling. MC integration is the core of the modern rendering pipeline, solving the *rendering equation* at each shading point:

$$L(\omega_o) = \int L(\omega_i) f_s(\omega_i, \omega_o) \langle \omega_i, n \rangle d\omega_i. \quad (3)$$

Given the BSDF parameters ϵ and the viewing direction ω_o , the BSDF becomes a 2D slice, and BSDF importance sampling seeks on a good PDF similar to this 2D BSDF slice, together with a sampling technique that generates samples according to this PDF.

Though BSDF importance sampling can be easily calculated if the whole BSDF values at all dimensions are known, it would be

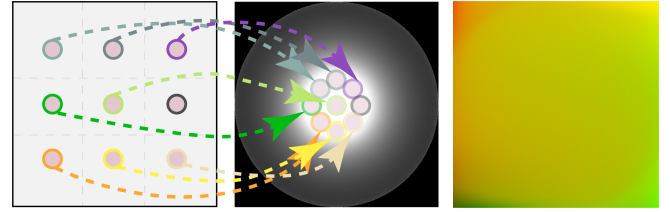


Fig. 3. A visualization of the process to generate an importance map. We start from generating sample points $(\xi_i^{(0)}, \xi_i^{(1)})$ on a uniform grid in the unit square $[0, 1]^2$ (left). The importance sampling process takes a 2D sample $(\xi_i^{(0)}, \xi_i^{(1)})$, and maps it to position (u_i, v_i) on a 2D BSDF slice (middle). This mapping can be stored as an image (right), where the position of each pixel is $(\xi_i^{(0)}, \xi_i^{(1)})$, with its color set as $(r = u_i, g = v_i, b = 0)$.

extraordinarily difficult for parametric BSDFs because of storage constraints. At the same time, the efficiency of BSDF importance sampling is crucial, since it will be evaluated at every ray bounce for every sample ray from the camera. Massive amount of heavy BSDF importance sampling calculation will dramatically slow down the rendering calculation. Therefore, we propose to separate BSDF importance sampling calculation into the precomputation and compression of sampling data, as well as efficient runtime value lookup.

3.2 BSDF Importance Baking

In this section, we analyze the essence of BSDF importance sampling, and present insight and motivation for our importance baking scheme.

We start with two issues from current BSDF importance sampling solutions.

- (1) Not all parametric BSDFs can be analytically sampled, such as multiple-bounce microfacet BSDFs. This includes certain difficult-to-sample NDFs such as GTR, certain sampling methods such as visible NDF (VNDF) sampling only working with specific NDFs, as well as multiple-bounce BSDFs in general.
- (2) Even analytic sampling methods are not perfect. In order to achieve the best quality, BSDF sampling requires a reasonable pdf $p(x)$ close to $f_s(\omega_i, n)$, but most analytical methods only sample the VNDF (ignoring the Fresnel term) or even just NDF D (ignoring everything else).

To deal with these issues and achieve perfect importance sampling, we propose our understanding in the essence of the sampling process – drawing samples according to a specific PDF is actually mapping from a uniform distribution to that PDF, viz., $f : (\xi^{(0)}, \xi^{(1)}) \mapsto (u, v)$ (in 2D). As an example, we visualize such a mapping acquired on a 2D BSDF slice in Fig. 3. In this visualization, we assume the pixels represent a uniform grid on the unit square $[0, 1]^2$, and each pixel i at position $(\xi_i^{(0)}, \xi_i^{(1)})$ stores its mapped position (u_i, v_i) as red and green. In this way, we are able to generate an image for any 2D mapping, which we name the *importance map*.

With this understanding, we immediately come up with the following insights.

- (1) For a 2D BSDF slice, it is not crucial whether it can be sampled analytically or not because analytic sampling only corresponds to a quick lookup on the importance map. Instead, being able to acquire and query a high-quality importance map is the actual key to perfect importance sampling. Fortunately, we demonstrate that the importance map *can be precomputed* with the help of optimal transport (Sec. 4.1).
- (2) Suppose one importance map can be obtained from a 2D BSDF slice defined with BSDF parameters ϵ and the outgoing direction ω_o . In order for the full parametric BSDF to be importance sampled, we have to collect all importance maps of all combinations of ϵ and ω_o . This requires heavy storage, and therefore *compression is needed*. Fortunately, we demonstrate that compression is not only possible but also efficient with the help of a lightweight neural network (Sec. 4.2).

Since we propose to precompute and compress the importance maps, the entire process is similar to the concept of light baking in real-time rendering. Therefore, we name it *BSDF importance baking*.

3.3 Existing Importance Sampling Strategies Analysis

Importance sampling estimates the properties of a distribution, which takes discrete values from a different function to reconstruct a new function that displays similarities with the target distribution. Usually from an easy-to-implement distribution, e.g., $U[0, 1]^d$ to an arbitrarily complex function. Importance sampling is not special, and it just requires that the target distribution is a normalized (constantly scaled) version of the target function.

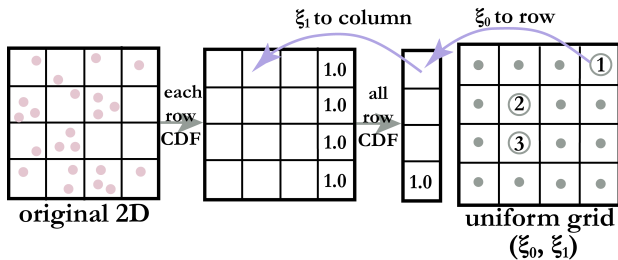


Fig. 4. In the 2D marginalized inverse transform sampling, two random variables are used to firstly choose a row and then choose a column, forming a sampled location at the target distribution. Therefore, this sampling strategy acts as a mapping from the uniform distribution to the target distribution.

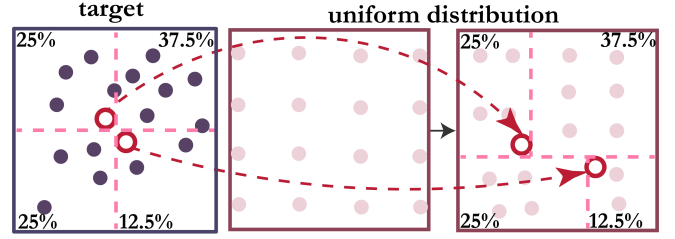


Fig. 5. The hierarchical sample warping strategy performs sampling by hierarchically choosing one of the quarters with its probability (i.e., the sample distribution in the target). However, this strategy can not guarantee continuity between each quarter. For example, two adjacent samples (red circles) in the target distribution are mapped from two samples at a distance, leading to the discontinuities in the binning result, as shown in Fig. 6.

In 1D, a typically used method is called the marginalized inverse transform sampling. It generates sample numbers using uniform random number $\xi \sim U[0, 1]$ between 0 and 1 for any probability distribution given its cumulative distribution function (CDF). Comparing ξ and CDF values will map the random number to a certain value within the distribution domain. For discrete 2D situations, marginalized inverse transform sampling can be fulfilled by using two uniform random variables $(\xi_0, \xi_1) \sim U[0, 1]^2$ as shown in Fig. 4. By calculating the CDF of all elements in one dimension (eg. all columns) and the overall CDF of the other dimension (eg. all rows), each pair of (ξ_0, ξ_1) can be used to attain a pair of 2D sample (u_i, v_i) on the target distribution similar to 1D situation. The mapping can be written as a function f of elements between distributions $f : (\xi_0, \xi_1) \mapsto (u_i, v_i)$.

Another typical sampling strategy is hierarchical sample warping [Clarberg et al. 2005], which constructs levels of sample point mapping from coarse to detail between uniform distributed samples and target distribution, as shown in Fig. 5. It efficiently generates high-quality sample mapping without evaluating the whole target distribution. However, we observe that hierarchical sample warping allows two neighboring samples on the target to be mapped from two far-away samples in the uniform distribution (see Fig. 5).

In Fig. 6, we compare different sampling strategies by showing their *importance maps*, as well as the *sampling binning results* by applying bilinear interpolation using each importance map. Both marginalized inverse transform sampling and hierarchical sample warping show discontinuous importance maps, leading to an incorrect filamentous connections or grid artifacts in the binning results.

In summary, the existing solutions can be exploited for importance sampling, but they generate discontinuous importance maps, which raises difficulties for precomputation, compression, and interpolation. Therefore, we seek for a better solution to generate importance maps.

4 OUR METHOD

4.1 Optimal Transport for Precomputed Importance

In this subsection, we focus on attaining the importance maps. We start from an important fact that is often ignored. That is, potential

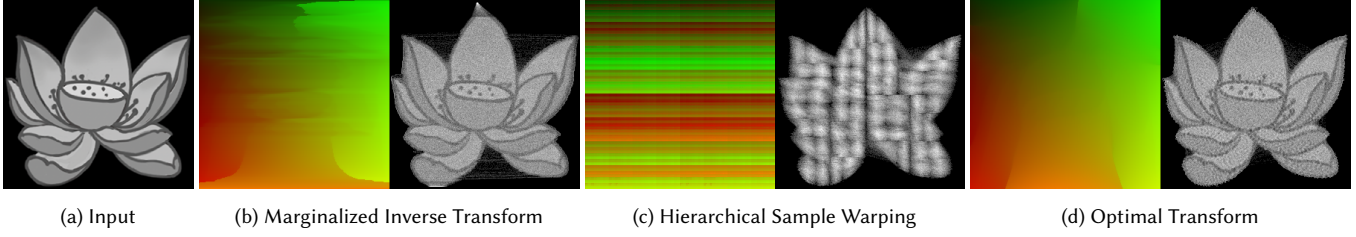
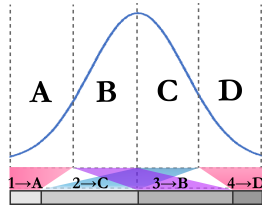


Fig. 6. Let the input lotus image (a) be the target distribution. We use three importance sampling strategies to generate the importance map and then interpolate and reconstruct images from the importance map via binning. Here, *binning* means sampling a location from the importance map using the same uniform sample sequence and accumulating it into the corresponding pixel, while applying bilinear interpolation, to reconstruct the images. Both marginalized inverse transform sampling and hierarchical sample warping generate importance maps with severe color discontinuity and high-frequency color variation, which leads to incorrect filamentous connections or grid artifacts in the binning result and raises difficulties for compression and further interpolation. On the contrary, the importance map by optimal transport has a smoother transition and lower frequency, allowing easier compression.

mappings that produce *the same importance sampled PDF* are not unique. Consider a toy example of sampling a truncated, and normalized 1D Gaussian defined on $[0, 1]$ shown on the right. Suppose we uniformly subdivide $[0, 1]$ into four segments A, B, C and D, each integrates to a probability of 0.1, 0.4, 0.4 and 0.1. Then we subdivide the uniform $[0, 1]$ into four

segments with lengths 0.1, 0.4, 0.4 and 0.1, and name them 1, 2, 3 and 4, respectively. Then the mapping $1 \mapsto A$, $2 \mapsto B$, $3 \mapsto C$, $4 \mapsto D$ is a valid mapping that importance samples the Gaussian, but the mapping $1 \mapsto A$, $2 \mapsto C$ and $3 \mapsto B$, $4 \mapsto D$ is also perfect importance sampling.



Moreover, from this example, one can immediately tell which sampling strategy is better: the first one is much smoother, and the second one suffers from discontinuity. This conclusion is never trivial because it directly proves that one pervasively used solution to obtain the importance maps – the marginalized inverse transform sampling, a.k.a. row-column sampling – is not suitable for generating good importance maps (Fig. 6). This is because a small perturbation in ξ_0 may result in a different row with a different 1D distribution, then even for similar ξ_1 , the resulting column can be far away.

Therefore, we prefer a strategy that provides smooth mapping that a small perturbation makes the mapped sample move moderately nearby. To satisfy this requirement, we refer to optimal transport (OT) – specifically, discrete optimal transport from the Lagrangian view [Feydy et al. 2019], which is able to find an optimal one-to-one mapping between two *point distributions* with the same number of points. In our case, this is to map from the unit square to the 2D BSDF slice for each combination of BSDF parameters ϵ and outgoing directions ω_o .

To conduct optimal transport, we first discretize both distributions into ordered point sets

$$\alpha = \cup_{i=1}^n \delta(\xi_i^{(0)}, \xi_i^{(1)}), \quad \beta = \cup_{j=1}^n \delta(u_j, v_j), \quad (4)$$

where n is the total number of points, and the $\delta(\cdot)$ is Dirac delta impulse at different positions. We weigh each point the same, which

immediately indicates that it is the local densities of those points that represent the values of the original continuous distributions. In other words, the continuous-to-discrete conversion itself is exactly importance sampling. We manually convert the unit square into a regular grid α (pixels) and use row-column sampling to convert the 2D BSDF slice into β .

Then we conduct optimal transport, giving a 1-to-1 correspondence between any two point distributions α and β , minimizing the Euclidean distance between them:

$$\operatorname{argmin}_{\psi} \sum_{i=1}^n \|\alpha(i) - \beta(\psi(i))\|, \quad (5)$$

where ψ is a permutation of the sequence $1, 2, \dots, n$, computed by the optimal transport process. After this, we record the positions of each $\beta(\psi(i))$ into each pixel's red and green channel, which completes the computation of an importance map.

Note that during the conversion of β , we used row-column sampling. However, this is in essence different from using that to find the mapping – we only use row-column sampling to discretize an image into points, and the mapping is found by optimal transport. More conversion tools can be explored in the research area of image stippling [Kim et al. 2008].

Also, note specifically that we focus on the *black box* usage of optimal transport as a general mathematical tool. However, we do not intend to compare or improve specific optimal transport solvers. We also do not extend further discussion on specific accurate/approximate distance metrics (e.g., earth mover's distance, Wasserstein distance, Kullback–Leibler divergence, Sinkhorn distance, etc.). In Sec. 5.1, we provide our choices for implementation.

In summary, for each of the three kinds of BSDFs and each possible combinations of parameters, we compute the BSDF slices, and calculate OT mapping to attain importance maps. With all these calculations, we have a database of importance maps and BSDF slices. Details of the whole generation process can be found in Sec. 5.1.

4.2 Lightweight Neural Networks for Importance Baking

Now that the OT already provides a reliable sampling scheme with these computed *importance maps*, then we should focus on data storage and compression. We introduce a lightweight neural network for compression and storage because networks provide a higher

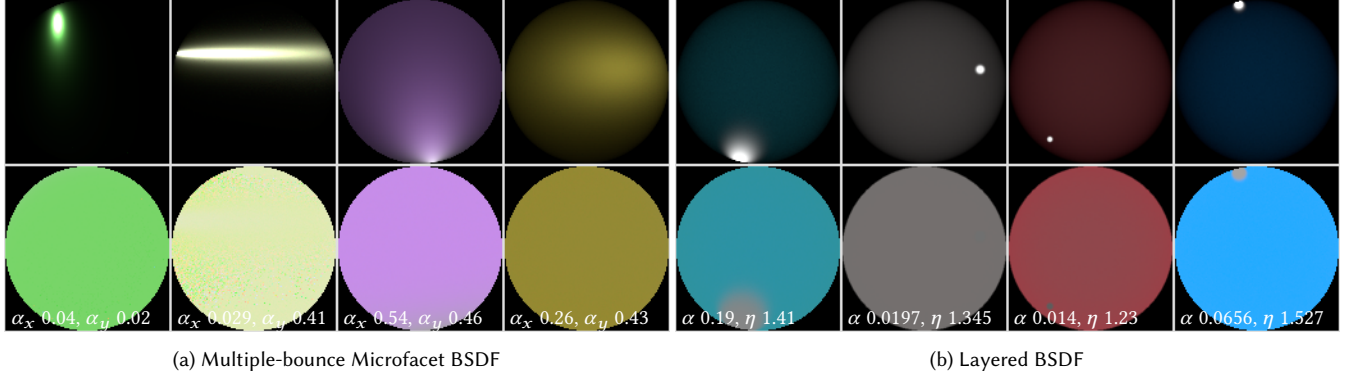


Fig. 7. We visualize BSDF slices (top row) and their corresponding sampling weight slices (bottom row). (a) For multiple-bounce microfacet BSDFs, no matter what the BSDF looks like, the sampling weight is a low-frequency function whether the roughness values are low or high, or the BSDF is isotropic or anisotropic. Therefore, we can output sampling weight in the BSDF sampling network without increasing its complexity. Note that the noise of the sampling weight comes from the original BSDF slice. (b) For layered materials, the situation becomes more complex. In the two-layer model we are studying, the reflection on the top dielectric surface creates a white highlight circle. Therefore, the sampling weight images of layered BSDF are not in uniform color. However, the sampling weight slices at the bottom row still show that the sampling weights are relatively low-frequency values.

compression rate and better expressiveness than traditional methods.

Before we proceed, we would like to note, as one could already tell immediately, the difference between our method and the normalizing flow methods. We treat the neural network as a general compression tool to save the data we have already computed for importance sampling. This is in essence different to letting neural networks figure out how to perform importance sampling. Therefore, our solution makes it much easier for networks to “learn” the sample mapping, and dramatically reduces the complexity of our networks, resulting in significantly better results and performance. We will also discuss more on the differences between our method and normalizing flow in Sec. 4.3.

We start with a few design principles from a series of observations.

- (1) We notice a significant amount of similarities between the importance maps when the BSDF parameters and incident directions change. The smooth change of these importance maps inspire us to use a neural network to compress them.

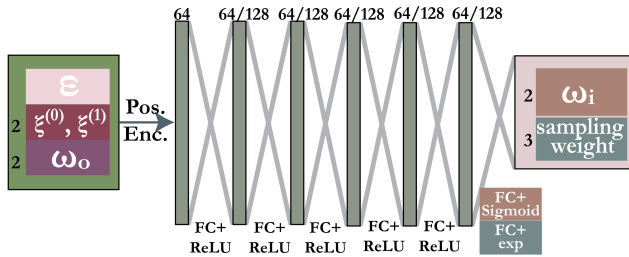


Fig. 8. The structure of our BSDF importance sampling network. The network takes the following as inputs: material parameters ϵ , outgoing ray direction ω_o and a pair of uniformly distributed random numbers ξ_0 and ξ_1 . Then we process the inputs with a positional encoding, feed to an MLP (six layers of 64/128 hidden units), and finally output the sampled direction ω_i and the sampling weight. Note that we increase the MLP size from 64 to 128 for layered BSDFs, since the sampling weight slices are more complex than other BSDFs as shown in Fig. 7. More details are shown in Sec.5.2.

- (2) As mentioned in Sec. 2, since runtime performance is crucial to core rendering, the highest level design of our neural network is to keep it as lightweight as possible, thus allowing for fast inference during rendering.
- (3) During BSDF importance sampling, only one incident direction needs to be sampled at a time. Therefore, the importance map should be *point queried* instead of being output on the whole. This also further reduces the complexity of our neural network.
- (4) For Monte Carlo estimation, the BSDF sampling process is expected to output not only an incident direction but also its *sampling weight*, which is the BSDF value divided by the PDF value. For multiple-bounce microfacet materials and Disney principled materials, the sampling weight is a 3-channel value and is close to constant since we design our PDFs to have the same shapes with the BSDF slices converted to grayscale. For layered materials, the sampling weight data are more complex but still relatively low-frequency. Fig. 7 shows more details about sampling weight data.

Based on these design principles, we propose a lightweight neural network for our BSDF importance sampling.

BSDF Sampling Network. Aside from the BSDF parameters ϵ and the outgoing direction ω_o , our importance sampling network takes two random numbers and outputs the sampled incident direction together with its sampling weight:

$$\mathcal{I}(\epsilon, \omega_o, \xi_0, \xi_1) = (\omega_i, sw_R, sw_G, sw_B), \quad (6)$$

where sw is the 3-channel sampling weight. The network structure is shown in Fig. 8. Note that since we have full control of the sampling process, the training data, especially the sampling weight to our BSDF importance sampling network, is guaranteed to be correct (unbiased), albeit some noise can remain. Therefore, the source of bias in our results can only originate from the learning process of the network itself, as will be analyzed in the next subsection.

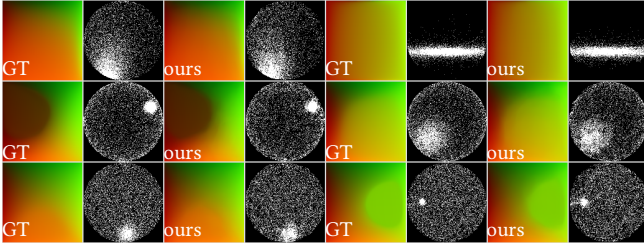


Fig. 9. We visualize pairs of importance maps and their corresponding binning results (overexposed to better visualize the difference) for different materials, including multiple-bounce microfacet conductor materials (top row), layered materials (middle row) and Disney principled materials (bottom row), learned by our BSDF sampling network. By comparing with the ground truth, our model can accurately learn the importance maps and reproduce the BSDF lobes for various parametric BSDFs.

To validate the functionality of our BSDF sampling network, we compare the importance maps and the binning results between the ground truth and our network’s prediction, as shown in Fig. 9. Thanks to our use of optimal transport that results in smooth importance maps, our network is able to learn to compress them well.

So far, our solution to general parametric BSDF importance sampling has already completed. However, modern rendering pipeline usually supports multiple importance sampling (MIS). In the MIS framework, a renderer should implement not only BSDF sampling, but also BSDF evaluation (returning the BSDF value given the incident and outgoing directions) and PDF query (returning the PDF value similarly). For some BSDFs, these two tasks can be elegantly computed in closed form, e.g., the Disney Principled BSDFs. But for some other BSDFs, it is still difficult to evaluate the BSDF values, e.g., the layered BSDFs.

Therefore, *for completeness*, we also compress the BSDFs and PDFs, using separate neural networks.

BSDF evaluation network. For BSDF evaluation, our evaluation network \mathcal{E} takes an additional incident direction ω_i and outputs the BSDF value as a 3-channel RGB value:

$$\mathcal{E}(\epsilon, \omega_o, \omega_i) = f_s(\omega_o, \omega_i) \langle \omega_i, n \rangle. \quad (7)$$

Note again that the BSDF evaluation network is *optional* and should only be used when there is no analytical BSDF evaluation scheme. We choose the simple network structure because it fulfills our need already, producing correct BSDF values, as validated in Fig. 1 and Fig. 15. We are aware of other existing neural BSDF/BTF compression methods [Fan et al. 2022; Rainer et al. 2020, 2019; Sztrajman et al. 2021; Zheng et al. 2021]. And we also believe that their success can further strengthen our importance sampling scheme when combined together, replacing our simple evaluation network. But we do not extend analysis on them any further, since BSDF evaluation is not our main contribution.

PDF query network. Our PDF query network \mathcal{P} has a very similar definition compared to the BSDF evaluation network. It also takes in the combinations of BSDF parameters as well as the incident directions and returns the PDF value of sampling that direction

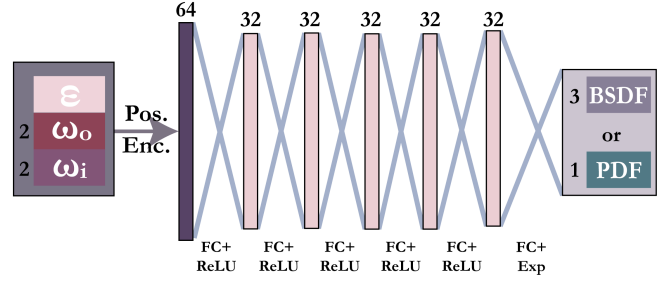


Fig. 10. The structure of our BSDF evaluation and PDF query networks. These two networks take material parameters ϵ , outgoing direction ω_o , and incident direction ω_i as inputs and output the BSDF value or PDF, respectively. More details are shown in Sec. 5.2.

under solid angle measurement:

$$\mathcal{P}(\epsilon, \omega_o, \omega_i) = \text{PDF}(f_s(\omega_o, \omega_i) \langle \omega_i, n \rangle), \quad (8)$$

For readers unfamiliar with rendering, we make a special note here that our PDF query networks only serve as providing the PDF values for the computation of MIS weights (not sampling weights). A physically correct PDF is certainly welcomed – as it will reduce the variance of the MIS combined results. However, even completely wrong PDFs in this step only leads to higher variance rather than any bias. We elaborate on this more in the next section with a simple but convincing experiment.

Our BSDF evaluation and PDF query networks share a similar lightweight structure, as illustrated in Fig. 10. Note specifically that the emphasis on lightweight neural network design in core rendering [Fan et al. 2022; Zhu et al. 2022] is different from that in deep learning. We use neural networks only as a general tool for efficiently compressing and querying high-dimensional data.

4.3 Properties and analysis

Source of Bias. Bias is not desired, but there is no guarantee that neural networks produce fully unbiased results. In our case, the BSDF sampling network learns the sampling weight values, which is equivalent to $\frac{f_s \langle \omega_i, n \rangle}{\text{PDF}}$. However, this PDF is the correct PDF (rather than the output of our PDF query network) as part of the unbiased training data. And it will not be exposed to any other stages but only together as the sampling weight. Therefore, the only source of bias originates from the learning process of the network itself, and is on sampling weight only, which is usually low frequency and easy to learn as shown in Fig. 7.

For our PDF query network, we do not include any sum-to-one (and other normalization) constraints. As mentioned earlier, this is because the PDF used in MIS can be any value, and the resulting MIS weights may even be negative [Kondapaneni et al. 2019]. To further prove this statement, we show the MIS rendering results using a ball status scene with negative uniform MIS PDF, tenth power PDF, and cos-weighted PDF as reference in Fig. 11.

Relationship to other BSDF sampling methods. Our neural BSDF importance sampling solution has significant differences from other

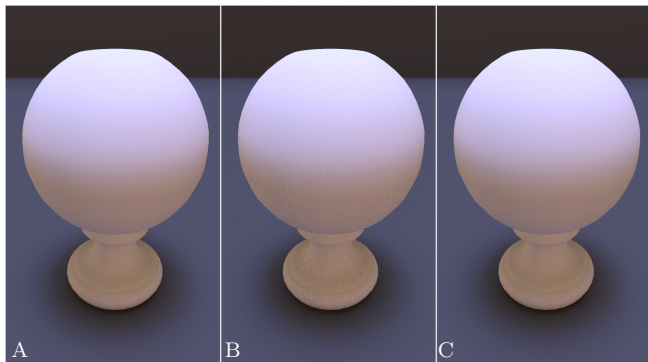


Fig. 11. Rendered results of a ball status scene with a diffuse BSDF using different MIS PDFs: (a) -10, (b) tenth power of cos-weighted hemisphere PDF value, and (c) cos-weighted hemisphere PDF treated as the ground truth. All results are rendered with 16,384 samples for convergence. These three results are identical, which confirms that the MIS PDF value does not require any sum to one constraint, and the choice of MIS PDF does not introduce any bias.

neural BSDF methods. For example, Fan et al. [2022] sample different layered BRDF parameters by learning a Gaussian lobe and a Lambertian lobe. Similarly, Sztrajman et al. [2021] apply a shallow network to learn the mapping between autoencoded BRDFs and parameters of fitted analytical Blinn-Phong importance sampling parameters. Though Sztrajman et al. [2021] also use a neural network to calculate BRDF importance sampling, their network only provides fitting parameters to the analytical Blinn-Phong model.

Therefore, the core of these importance sampling methods is still the analytical model fitting of BSDF slices. The insight of these strategies relies on a rough prior knowledge of the shape of the BSDF slices. When the prior breaks, the efficiency of sampling will be abysmal. For example, the Blinn-Phong importance sampling model serves as a good sample function for BSDFs with isotropic single specular lobe. However, anisotropic materials or complex multiple-lobe materials cannot be accurately described with only one specular highlight. We show the Blinn-Phong importance sampling fitting results of anisotropic multiple-bounce microfacet BSDF in Fig. 12 for validation.

We have differentiated our approach against normalizing flow-based importance sampling methods at the beginning of Sec. 4.2, and more comparisons can be found in Fig. 13. For other sampling methods, such as Dupuy et al. [2018], they perform targeted design for specific kind of BSDFs. We consider them too far from our general parametric BSDF sampling, and do not extend the further discussion.

5 IMPLEMENTATION DETAILS

5.1 Data Generation

BSDF slice generation. We use an AMD 64-core 3995WX CPU machine for training data generation. The material parameters ϵ are sampled uniformly in their own space except the roughness is sampled in squared space. For outgoing directions, we sample them uniformly in square root of the polar angle to emphasize the grazing angle cases. Each pair of $(\epsilon, \omega_{ox}, \omega_{oy})$ will be used for generating a 2D BSDF slice with a resolution of 128×128 , where each pixel

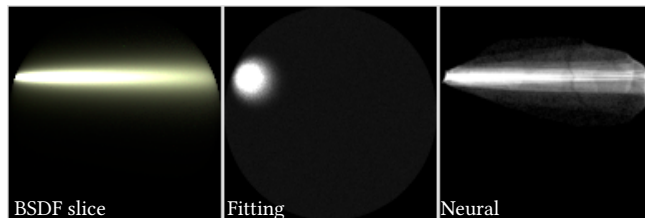


Fig. 12. Binning result comparison between the fitted analytical Blinn-Phong model (middle) and our neural solution (right) on an anisotropic multiple-bounce microfacet BSDF slice. Our solution shows a better match than the fitted Blinn-Phong model to the reference BSDF slice.

represents an incoming direction $(\omega_{ix}, \omega_{iy})$ and stores the three-channel $f_s(\omega_i, n)$ value. In our experiments, we generate 32,768 slices for our training on each material.

The dataset for all the networks are generated from these 2D BSDF slices. These 2D BSDF slices can be directly used as training data for BSDF evaluation networks. For the PDF query network’s dataset, we calculate the luminance of three-channel BSDFs to get single-channel data and then get the PDF by normalizing the luminance with the summed solid angle measurement. Finally, for the BSDF sampling network, we reparameterize the PDF from the hemisphere to a unit disk and then perform marginalized inverse transform sampling on the unit disk. This ensures that we are sampling on a correct distribution domain. The importance sampled sample points function as the target distributions of OT. Additionally, we also store the three-channel $\frac{f_s(\omega_i, n)}{\text{PDF}}$ values as the sampling weight. Note that the sampling and PDF calculation are all based on solid angle-measured BSDF radiance values, and there is no parameter space transformation. Therefore, there is no need for Jacobian calculation. In summary, all training data originate from BSDF evaluation.

Optimal transport. In the next step, we use Geomloss [Feydy et al. 2019] to provide an initial mapping state, and then we use SOT [Paulin et al. 2020] for further optimization. We choose to use two OT solutions because combining two methods will combine the advantages and avoid the limitations of these two methods. SOT computes swiftly, but we notice apparent crevices on BSDF slices even after tens of thousands of iterations, especially for small roughness situations. The networks can grasp these crevices and produce incorrect dark areas in the rendering results. Geomloss runs slower than SOT, but it provides an outstanding initial state for SOT to optimize further. Note again, Geomloss and SOT are just ways of giving the mapping for the next learning step of our method. Other specific implementations of OT are all available choices.

5.2 Network and Training Details

Our networks are trained on a server with an Intel 16-core i9-7960X CPU and a NVIDIA 3090, and then the weights are saved into binary files for each kind of parametric BSDFs for renderer integration.

Input Encoding. Input encoding has been proved to provide sharper and more accurate results in several works [Mildenhall et al. 2020; Müller et al. 2021]. Therefore, we adopt the input encoding for our query parameters similar to [Müller et al. 2021]. The frequency

Material	Parameter(Symbol)	Encoding Method
Multiple Bounce Material	Roughness ($\alpha \in \mathbb{R}^2$)	freq(α) $\in \mathbb{R}^{2 \times 4}$
	R_0 ($a \in \mathbb{R}^3$)	freq(a) $\in \mathbb{R}^{3 \times 4}$
	Camera Dir ($\omega_o \in \mathbb{R}^2$)	freq(ω_o) $\in \mathbb{R}^{2 \times 12}$
	Light Dir ($\omega_i \in \mathbb{R}^2$)	freq(ω_i) $\in \mathbb{R}^{2 \times 12}$
	Sample Input ($\xi \in \mathbb{R}^2$)	freq(ξ) $\in \mathbb{R}^{2 \times 12}$
Layered Material	Refractive Index ($\eta \in \mathbb{R}$)	ob(η) $\in \mathbb{R}^4$
	Roughness ($\alpha \in \mathbb{R}$)	ob(nl($1 - e^{-\alpha}$)) $\in \mathbb{R}^4$
	Attenuation Coeff ($\sigma_T \in \mathbb{R}$)	ob(σ_T) $\in \mathbb{R}^5$
	Albedo ($a \in \mathbb{R}^3$)	id(a) $\in \mathbb{R}^3$
	Camera Dir ($\omega_o \in \mathbb{R}^2$)	ob(ω_o) $\in \mathbb{R}^{2 \times 12}$
	Light Dir ($\omega_i \in \mathbb{R}^2$)	ob(ω_i) $\in \mathbb{R}^{2 \times 12}$
	Sample Input ($\xi \in \mathbb{R}^2$)	ob(ξ) $\in \mathbb{R}^{2 \times 12}$
Disney Material	Metallic ($m \in \mathbb{R}$)	ob(m) $\in \mathbb{R}^8$
	Roughness ($\alpha \in \mathbb{R}$)	ob(nl($1 - e^{-\alpha}$)) $\in \mathbb{R}^8$
	Camera Dir ($\omega_o \in \mathbb{R}^2$)	ob(ω_o) $\in \mathbb{R}^{2 \times 12}$
	Light Dir ($\omega_i \in \mathbb{R}^2$)	ob(ω_i) $\in \mathbb{R}^{2 \times 12}$
	Sample Input ($\xi \in \mathbb{R}^2$)	ob(ξ) $\in \mathbb{R}^{2 \times 12}$

Table 1. Parameters Encoding. For encoding method, freq(\cdot) means frequency encoding [Mildenhall et al. 2020] with both sine and cosine functions, ob(\cdot) means one-blob encoding [Müller et al. 2019], id(\cdot) stands for identity function, and nl(\cdot) means normalize the value’s range to $[0, 1]$.

encoding consists of $2k$ sine and cosine functions, with k different frequencies from 2^0 to 2^{k-1} , and the one blob encoding [Müller et al. 2019] has l bins for the discrete Gaussian values of the encoded numbers. Details of each encoding for different types of materials are shown in Table 1.

Loss functions. For BSDF evaluation networks, in order to capture both highlight area’s intensity and non-highlight area’s color, we use symmetric mean absolute percentage error (SMAPE) on BSDF values to optimize the networks as following:

$$\mathcal{L}_{\text{eval}} = \frac{\|\text{pred} - \text{gt}\|_1}{\text{sg}(\|\text{pred}\|_1) + \text{sg}(\|\text{gt}\|_1) + \varepsilon}$$

where ε is set to 0.01 and $\text{sg}(\cdot)$ means there is no gradient back propagation here.

For PDF query networks, the predicted values are transformed into log space and then we apply ℓ_1 loss as follows:

$$\mathcal{L}_{\text{pdf}} = \|\log(1 + \text{pred}) - \log(1 + \text{gt})\|_1.$$

For BSDF sampling network, we directly apply ℓ_1 loss on network output, i.e. the incident direction. Additionally, to reduce the number of queries for networks in rendering, our network also returns the sampling weight. It is a low-frequency function (its visualization is shown in Fig. 7) that is averaged from the corresponding three-channel BSDFs. We use ℓ_1 loss for the sampling weight outputs in log space. Thus, the full loss of our sample network shows below:

$$\mathcal{L}_{\text{sample}} = \|\hat{\omega} - \omega_{\text{gt}}\|_1 + \lambda \|\log(1 + \hat{\text{sw}}) - \log(1 + \text{sw}_{\text{gt}})\|_1,$$

where $\hat{\omega}$ indicates the predicted incident direction, and the $\hat{\text{sw}}$ is predicted sampling weight which is a 3 channel value. ω_{gt} and sw_{gt} are the corresponding ground truth of them. We set λ to 0.4 in our training.

Training details. We use the ADAM [Kingma and Ba 2014] optimizer with default parameters and cosine annealing scheduler to optimize our BSDF sampling networks, and Ranger [Wright 2019] optimizer to optimize the BSDF evaluation and PDF query networks. The batch size of training is set to 1048576 and learning rate is set to 0.0001. Each networks are trained with 500 epochs in total. It takes about 48 hours to train BSDF sampling networks on each material, and 12 hours to train BSDF evaluation networks and PDF query networks separately on each material.

5.3 Renderer Integration

We integrate our network in Mitsuba renderer [Jakob 2010] with minimum revisions to only BSDF classes. The network inferences are integrated with C++ and Mitsuba using Eigen [Guennebaud et al. 2010]. The fully-connected layers are interpreted as matrix multiplications implemented in three different classes. Inside neural BSDF classes, the evaluation, sample, and PDF functions will only need to call the network inference functions with corresponding parameters. At render time, the ray directions ω_i and ω_o for each shading point, as well as BSDF parameters ϵ serve as inputs of both BSDF evaluation and PDF query networks. The BSDF sampling network requires BSDF parameters ϵ , outgoing direction ω_o and a pair of random numbers (ξ_0, ξ_1) . Then the sampling network returns an incident direction ω_i together with the sampling weight. In this case, we do not need to refer to the BSDF evaluation and PDF query networks inside the sample function again. Using sampling weight relieves us from performing three network inferences and thus optimize the calculation efficiency. At the same time, it further decreases the bias brought about by the differences between BSDF sampling and PDF query networks. In sum, we have not made any revisions to the integrator. All the modifications we have made are only inside a neural BSDF class.

6 RESULTS AND COMPARISON

We integrate our network in Mitsuba renderer [Jakob 2010], and network inferences are integrated using C++ and Eigen [Guennebaud et al. 2010]. All the rendering performance is measured on an Intel 8-core i9-9900K machine. Now we only perform CPU network inference.

In this section, we validate our results on multiple-bounce rough conductors with the GGX NDF, position-free layered BSDFs, and Disney principled materials. Also, we use relative mean square error (relMSE) to measure the difference with the ground truth. Since our method and the other methods might converge to different ground truths, we use their own converged results as the ground truth (GT).

6.1 Multiple-bounce Microfacet BRDFs

For multiple-bounce microfacet conductors, we compare our rendering results with Heitz et al. [2016] and use their method to generate all ground truth (GT) results. We start with comparing the efficiency and qualities of BSDF sampling only rendering results between our neural network and RealNVP used by Xie et al. [2019]. We also perform comparisons for low-roughness conductors, as well as MIS rendering results.

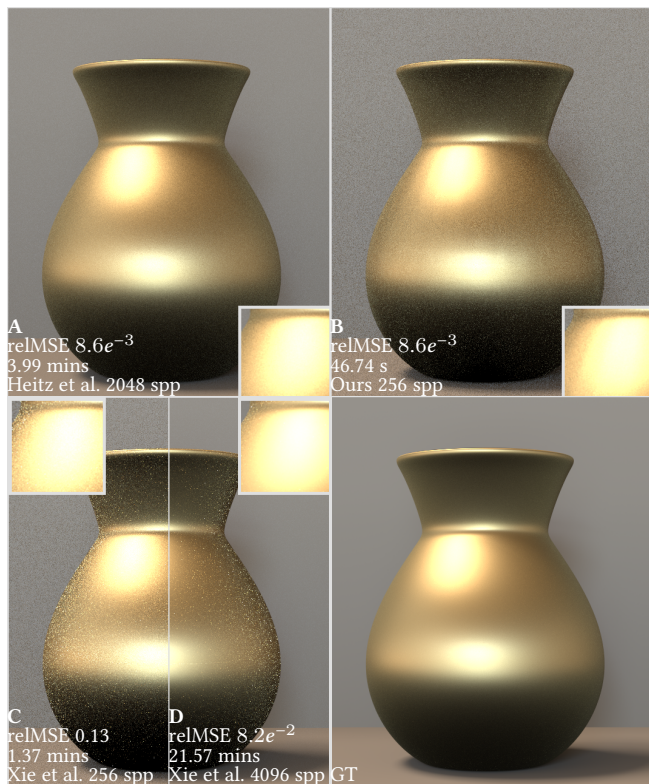


Fig. 13. We perform equal quality comparisons between (A) Heitz et al. [2016] and (B) our method (BSDF sampling network) on an anisotropic multiple-bounce microfacet conductor BSDF. Also, we perform equal-spp comparisons between (B) our method and (C) Xie et al. [2019]. Our method achieves less noise with better performance, while (D) Xie et al. [2019] produce a noisy result with 16 times samples. Our method achieves the correct appearance and with less amount of time.

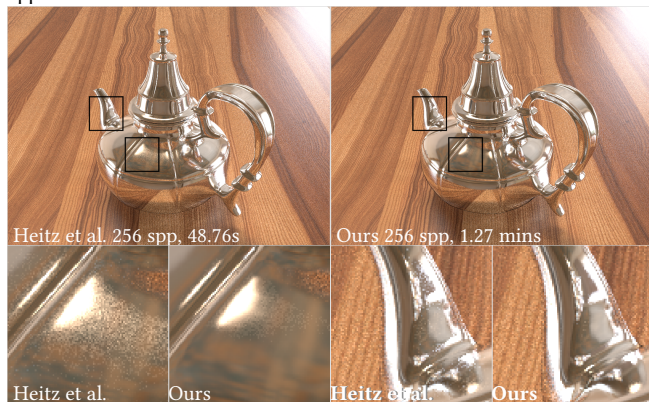


Fig. 14. Comparison between our method (BSDF sampling network) and Heitz et al. [2016] on low-roughness ($\alpha = 0.01$) multiple-bounce microfacet conductor BSDFs with equal sampling rate. Our method shows correct highlights, while preserving a lower noise level than Heitz et al. [2016].

Vase scene. In Fig. 13, we show an anisotropic conductor vase with $\alpha_x = 0.3$ and $\alpha_y = 0.1$ under an environment light, considering direct lighting only. We use our BSDF sampling network trained on the multiple-bounce dataset and compare it against Heitz et

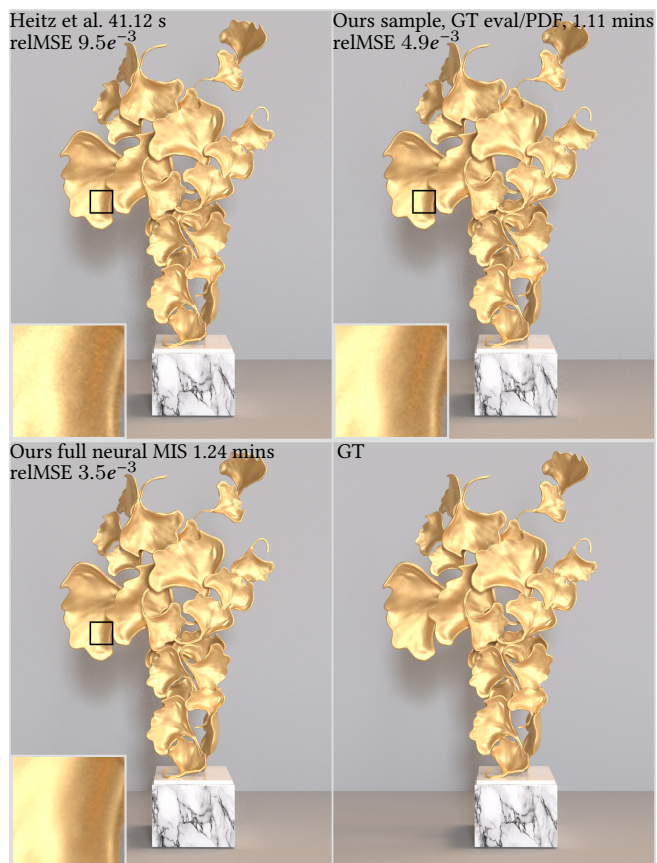


Fig. 15. Comparison between our method (BSDF sampling network + GT evaluation and PDF), our method (three networks), Heitz et al. [2016] and the ground truth rendered with 256 spp. The roughness is set as 0.3. By introducing our BSDF sampling network only, our model is already able to produce results with lower variance, while the other components (evaluation and PDF query network) further reduce the noise level, at the cost of a longer time.

al. [2016] and Xie et al. [2019]. For fairness, we set the hidden dimension of the RealNVP network by Xie et al. [2019] as 32, keeping a similar parameter size as ours, and train their model at the same time as ours. Since their model only provides the sampled direction and its PDF, we call the GT evaluation function and divide it by their PDF as the sampling weights. For all the methods, we perform BSDF sampling only. Our method use less time to achieve equal quality than Heitz et al. [2016]. At the same time, the result by Xie et al. [2019] has higher variance with 2× time cost.

Low roughness teapot scene. BRDF sampling is also tricky for multiple-bounce microfacet conductors with low roughness, due to the high frequency of the BRDF lobes. To show the effectiveness of our BSDF sampling network on such cases, we compare our method against Heitz et al. [2016] on the *teapot scene* (with $\alpha = 0.01$) with equal-spp in Fig. 14. In this comparison, we perform BSDF sampling only. Our result preserves the correct highlights and shows much less variance than Heitz et al. [2016], with only 0.5× extra cost.

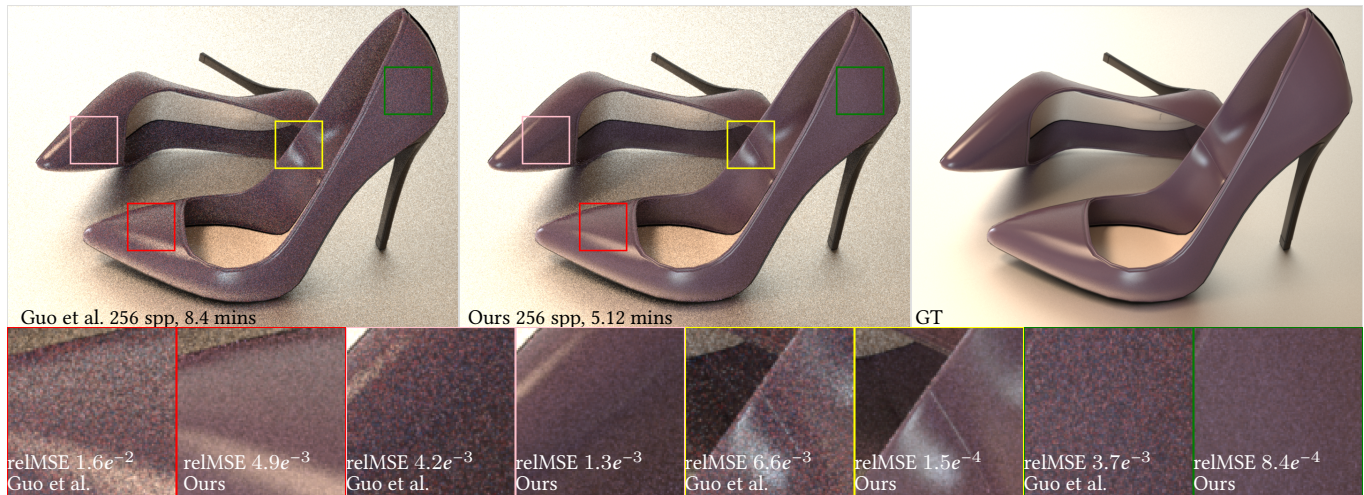


Fig. 16. Equal-spp comparison between our model (BSDF sampling network) and Guo et al. [2018] on a layered material. Our method not only shows less variance but also has higher efficiency. The layered material consists of a dense media in the middle, leading to an expensive random walk for Guo et al. [2018].

Ginkgo ornament scene. Besides the BSDF sampling network, we also introduce BSDF evaluation and PDF query networks, which are essential for complex BSDFs with heavy evaluations, like the multiple-bounce Smith microfacet BSDFs. Our full solution enables MIS for Monte Carlo rendering. In Fig. 15, we compare our methods (sample network only and full solution), and Heitz et al. [2016]. For our method (sample network only), we use the BSDF evaluation and PDF from Heitz et al. [2016] to enable MIS. By comparison, we find that our full solution outperforms the others in terms of rendering quality, with only a slight time overhead.

Conductor Kitchen Shelf. In Fig. 1 (top), we compare our full solution (three networks) with Heitz et al. [2016] on a variety of objects with different colors and roughness lit by an area light. The roughness of the four objects (from left to right) are set as $\alpha = 0.2$, $\alpha = 0.6$, $\alpha = 0.03$ and $\alpha_x = 0.08$ and $\alpha_y = 0.3$, covering from low roughness to high roughness, from isotropic to anisotropic materials. By comparison, our result shows higher quality than Heitz et al. [2016] both visually and quantitatively. However, the time cost of our method is about $2\times$ slower, due to the network inference. Since our current network inference is a simple CPU implementation, we believe that further optimization can decrease our time cost significantly.

6.2 Position-free Layered BSDFs

In this section, we demonstrate effectiveness of our model on layered materials by comparing with Guo et al. [2018] and use this method to generate ground truth (GT). Again, we only show results on two-layer BSDFs for simplicity, and more layers can be handled in the same way.

Layered Shoes. In Fig. 16, we compare our BSDF sampling model (trained on layered materials) with Guo et al. [2018] with equal spp on a Shoe scene lit by an environment map. The shoes have a dielectric material with roughness $\alpha = 0.08$ and refractive index $\eta = 1.5$ as the top layer, a diffuse with color as the substrate, together with a medium ($\sigma_T = 0.8$). Both methods perform BSDF sampling only, without MIS. With an equal sampling rate, the result by Guo

et al. [2018] is much noisier while having a much longer time than ours. The expensive time cost of Guo et al. [2018] is due to the random walk in the dense medium.

Layered Kitchen Shelf. Then, we validate our full solution (three networks) on layered BSDFs, by comparing against Guo et al. [2018] with equal time in Fig. 1 (bottom). In this scene, we show several objects, with different roughness α , refractive index η , attenuation coefficient σ_T , and albedo color lit by an environment map, considering both direct lighting and indirect lighting. Both methods use MIS to render this scene. To achieve equal time, we use 1024 spp for our method and 850 spp for Guo et al. [2018]. Our result has a lower variance and higher computational efficiency than theirs.

6.3 Disney Principled BSDFs

Next, we show some results on Disney Principled BSDFs. We choose to vary two parameters *metallic* and *roughness* while fixing the remaining parameters. As a comparison, we implement full Disney principled model in Mitsuba renderer to generate ground truth (GT) results. In Fig. 17, we show an equal-spp comparison between our model (BSDF sampling model) and Disney principled BSDF on the coffee cup scene with varying parameters (metallic and roughness), lit by an environment map. Both our model and Disney principled BSDF use BSDF sampling only. Under all these settings, our method achieves less noise level than Disney Principled BSDFs, especially for materials with low metallic and roughness, at the cost of a longer rendering time ($1.5\times$) due to the network inference. It proves our method provides a better BSDF importance sampling strategy than Disney Principled BSDFs.

7 DISCUSSION AND LIMITATIONS

Unified representation of BSDF evaluation and sampling. In our BSDF evaluation network, when the BSDF parameters are specified, one outgoing direction ω_o will produce a corresponding 2D BSDF slice, where each pixel is a BSDF value. This BSDF slice is queried using the 2D incident direction ω_i . Similarly, given the BSDF parameters and an outgoing direction, a 2D importance map is established,

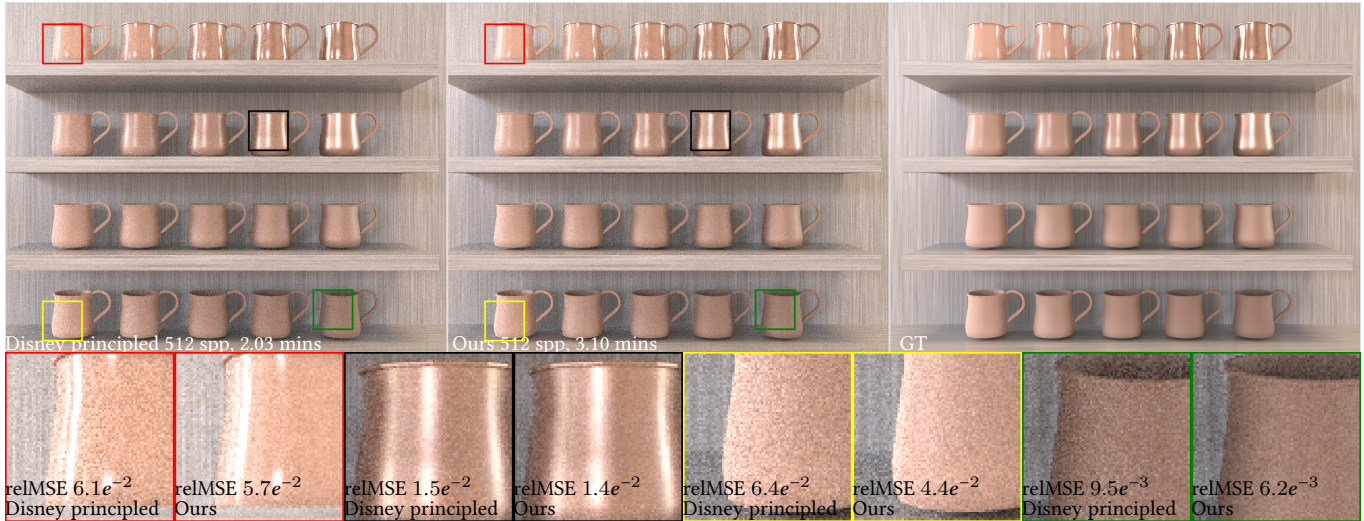


Fig. 17. Equal-spp comparison between our model (BSDF sampling network) and Disney Principled BSDF on a broad range of materials, with metallic m , varies from 0.1, 0.3, 0.5, 0.7, to 0.9, and roughness α varies from 0.15, 0.35, 0.65, to 0.85. Our method shows a lower noise level than Disney Principled BSDFs, proving that our method provides a better BSDF importance sampling strategy than theirs.

where a pixel represents the mapped sample position and the sampling weight. This importance map is queried with a 2D random number (ξ_0, ξ_1) . With this similarity identified, we have the evidence to claim that BSDF evaluation and sampling (and PDF query) are in essence very similar. Therefore, training a neural network for sampling should not be more complex than training for evaluation. This observation also shows that prohibitively expensive neural network structures, such as RealNVP and NICE, can be avoided.

Non-parametric/measured BSDFs. Intuitively, non-parametric BSDFs are often large data blocks. However, they are in fact of much lower dimensions than the parametric BSDFs. For example, a measured bidirectional texture function (BTF) has a much lower-dimensional parameter space (6D) than any of our examples (see Table. 1). Besides, in our work, we focus on BSDF importance sampling rather than BSDF evaluation, and we have already analyzed commonly used sampling schemes for measured BSDFs, such as fitting and normalizing flow.

Performance. Our results correctly reconstruct the parametric BSDFs' appearance with comparatively less noise, but the performance influence cannot be neglected even though our networks are small. Since we simply use inline CPU integration to fully integrate our neural networks into the renderer with minimum revision to

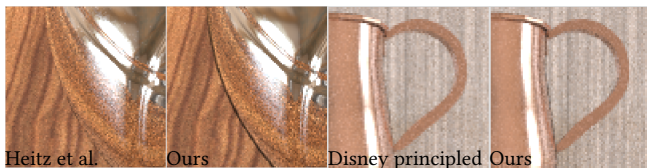


Fig. 18. Failure case. Our method introduces bias at grazing angles for highly specular materials. For example, the edges of the teapot and the coffee cup have an apparent black border.

the rendering pipeline, the inference of our neural networks is far from optimized. There are many ways to further improve inference efficiency, such as using a GPU inference framework, e.g., TensorRT, and/or devoting to considerable engineering optimization [Müller et al. 2021].

Bias. Neural network prediction will inevitably introduce bias even though the original training data are unbiased. Strict applications such as white furnace test will expose the bias issue immediately. Our method does have visible bias, as show in Fig. 18, but we do not see apparent problems in these results from our practical applications. Nevertheless, we believe it is still meaningful to study other methods that enable unbiased importance map compression.

8 CONCLUSION AND FUTURE WORK

We have introduced BSDF importance baking, a lightweight neural solution to perform perfect importance sampling of parametric BSDFs. We start from the observation that the mapping that performs importance sampling on a BSDF slice can be simply recorded as a 2D importance map. Following this observation, we propose to use optimal transport to precompute the importance maps accurately; then, we use a lightweight neural network to compress them efficiently. Together with an optional BSDF evaluation network and PDF query network, our method enables full multiple importance sampling (MIS) without any revision to the rendering pipeline. Compared with previous methods, we demonstrate reduced noise levels on rendering results with a rich set of appearances, including conductors with anisotropic roughness, layered BSDF, and Disney principled materials.

We believe that we have brought about novel contributions: our method is the first to utilize optimal transport in rendering applications that are not affected by its heavy computation during runtime; and our method is the first complete neural alternative with the potential to fully replace parametric BSDFs with MIS.

In the future, an immediate research direction is to improve the quality of our lightweight neural networks and further improve their performance. Sampling other forms of appearance representation could also be interesting, for example, using our importance baking scheme to investigate the sampling problem of 4D light fields data or 5D neural radiance fields (NeRF) data. Apart from neural compression, other data compression strategies could be explored as well, until a more efficient or unbiased method is found.

REFERENCES

- Michael Ashikhmin and Peter Shirley. 2001. An Anisotropic Phong Light Reflection Model. *Journal of Graphics Tools* 5 (01 2001).
- P. Beckmann and A. Spizzichino. 1963. *The scattering of electromagnetic waves from rough surfaces*. Pergamon Press.
- James F Blinn. 1977. Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*. 192–198.
- Nicolas Bonneel and David Coeurjolly. 2019. SPOT: Sliced Partial Optimal Transport. *ACM Transactions on Graphics (SIGGRAPH)* 38, 4 (2019).
- Nicolas Bonneel, Gabriel Peyré, and Marco Cuturi. 2016. Wasserstein barycentric coordinates: histogram regression using optimal transport. *ACM Trans. Graph.* 35, 4 (2016), 71–1.
- Nicolas Bonneel, Michiel Van De Panne, Sylvain Paris, and Wolfgang Heidrich. 2011. Displacement interpolation using Lagrangian mass transport. In *Proceedings of the 2011 SIGGRAPH Asia conference*. 1–12.
- Brent Burley. 2012. Physically-based shading at disney. In *ACM SIGGRAPH*, Vol. 2012. vol. 2012, 1–7.
- Petrik Clarberg, Wojciech Jarosz, Tomas Akenine-Möller, and Henrik Wann Jensen. 2005. Wavelet importance sampling: efficiently evaluating products of complex functions. In *ACM SIGGRAPH 2005 Papers*. 1166–1175.
- Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems* 26 (2013).
- Laurent Dinh, David Krueger, and Yoshua Bengio. 2014. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516* (2014).
- Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803* (2016).
- Jonathan Dupuy and Wenzel Jakob. 2018. An adaptive parameterization for efficient material acquisition and rendering. *ACM Transactions on graphics (TOG)* 37, 6 (2018), 1–14.
- Jiahui Fan, Beibei Wang, Miloš Hašan, Jian Yang, and Ling-Qi Yan. 2021. Neural BRDFs: Representation and Operations. *arXiv preprint arXiv:2111.03797* (2021).
- Jiahui Fan, Beibei Wang, Miloš Hašan, Jian Yang, and Ling-Qi Yan. 2022. Neural Layered BRDFs. In *Proceedings of SIGGRAPH 2022*.
- Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trounev, and Gabriel Peyré. 2019. Interpolating between Optimal Transport and MMD using Sinkhorn Divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*. 2681–2690.
- Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. <http://eigen.tuxfamily.org>.
- Yu Guo, Miloš Hašan, and Shuang Zhao. 2018. Position-free Monte Carlo simulation for arbitrary layered BSDFs. *ACM Transactions on Graphics (ToG)* 37, 6 (2018), 1–14.
- Eric Heitz. 2017. *A Simpler and Exact Sampling Routine for the GGX Distribution of Visible Normals*. Ph. D. Dissertation. Unity Technologies.
- Eric Heitz. 2018. Sampling the GGX distribution of visible normals. *Journal of Computer Graphics Techniques* 7, 4 (2018), 1–13.
- Eric Heitz and Eugene d'Eon. 2014. Importance sampling microfacet-based BSDFs using the distribution of visible normals. In *Computer Graphics Forum*, Vol. 33. Wiley Online Library, 103–112.
- Eric Heitz, Johannes Hanika, Eugene d'Eon, and Carsten Dachsbacher. 2016. Multiple-scattering microfacet BSDFs with the Smith model. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–14.
- Bingyang Hu, Jie Guo, Yanjun Chen, Mengtian Li, and Yanwen Guo. 2020. DeepBRDF: A Deep Representation for Manipulating Measured BRDF. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 157–166.
- Wenzel Jakob. 2010. Mitsuba renderer.
- Leonid V Kantorovich. 1942. On the translocation of masses. In *Dokl. Akad. Nauk. USSR (NS)*, Vol. 37. 199–201.
- Dongyeon Kim, Minjung Son, Yunjin Lee, Henry Kang, and Seungyong Lee. 2008. Feature-guided image stippling. In *Computer Graphics Forum*, Vol. 27. Wiley Online Library, 1209–1216.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- Ivo Kondapaneni, Petr Vévoda, Pascal Grittmann, Tomáš Skřivan, Philipp Slusallek, and Jaroslav Krivánek. 2019. Optimal multiple importance sampling. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–14.
- Alexandr Kuznetsov, Krishna Mullia, Zexiang Xu, Miloš Hašan, and Ravi Ramamoorthi. 2021. NeuMIP: Multi-Resolution Neural Materials. *Transactions on Graphics (Proceedings of SIGGRAPH)* 40, 4, Article 175 (July 2021), 13 pages.
- Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. 2004. Efficient BRDF importance sampling using a factored representation. *ACM Transactions on Graphics (ToG)* 23, 3 (2004), 496–505.
- Stephen McAuley, Stephen Hill, Naty Hoffman, Yoshiharu Gotanda, Brian Smits, Brent Burley, and Adam Martinez. 2012. Practical physically-based shading in film and game production. In *ACM SIGGRAPH 2012 Courses*. 1–7.
- Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.
- Gaspard Monge. 1781. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris* (1781).
- Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. 2019. Neural importance sampling. *ACM Transactions on Graphics* 38, 5 (2019).
- Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. 2019. Neural Importance Sampling. *ACM Trans. Graph.* 38, 5, Article 145 (Oct. 2019), 19 pages. <https://doi.org/10.1145/3341156>
- Thomas Müller, Fabrice Rousselle, Jan Novák, and Alexander Keller. 2021. Real-time neural radiance caching for path tracing. *arXiv preprint arXiv:2106.12372* (2021).
- Lois Paulin, Nicolas Bonneel, David Coeurjolly, Jean-Claude Iehl, Antoine Webanck, Mathieu Desbrun, and Victor Ostromoukhov. 2020. Sliced optimal transport sampling. *ACM Trans. Graph.* 39, 4 (2020), 99.
- Bui Tuong Phong. 1975. Illumination for Computer Generated Pictures. *Commun. ACM* 18, 6 (jun 1975), 311–317. <https://doi.org/10.1145/360825.360839>
- 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>
- Gilles Rainer, Wenzel Jakob, Abhijeet Ghosh, and Tim Weyrich. 2019. Neural BTF Compression and Interpolation. *Computer Graphics Forum (Proceedings of Eurographics)* 38, 2 (March 2019).
- Justin Solomon, Fernando De Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. 2015. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (ToG)* 34, 4 (2015), 1–11.
- Alejandro Sotrajman, Gilles Rainer, Tobias Ritschel, and Tim Weyrich. 2021. Neural BRDF Representation and Importance Sampling. In *Computer Graphics Forum*, Vol. 40. Wiley Online Library, 332–346.
- Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. 2007. Microfacet Models for Refraction through Rough Surfaces. *Rendering techniques 2007* (2007), 18th.
- Beibei Wang, Wenhua Jin, Jiahui Fan, Jian Yang, Nicolas Holzschuch, and Ling-Qi Yan. 2022. Position-free multiple-bounce computations for smith microfacet BSDFs. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–14.
- Less Wright. 2019. Ranger - a synergistic optimizer. <https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer>.
- Feng Xie and Pat Hanrahan. 2018. Multiple Scattering from Distributions of Specular V-Grooves. *ACM Trans. Graph.* 37, 6, Article 276 (2018), 14 pages.
- Feng Xie, Anton Kaplanyan, Warren Hunt, and Pat Hanrahan. 2019. Multiple scattering using machine learning. In *ACM SIGGRAPH 2019 Talks*. 1–2.
- Chuanjun Zheng, Ruzhang Zheng, Rui Wang, Shuang Zhao, and Hujun Bao. 2021. A Compact Representation of Measured BRDFs Using Neural Processes. *ACM Transactions on Graphics (TOG)* 41, 2 (2021), 1–15.
- Junqiu Zhu, Sizhe Zhao, Lu Wang, Yanning Xu, and Yan Ling-Qi. 2022. Practical Level-of-Detail Aggregation of Fur Appearance. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2022)* 41, 4 (2022).