NeuSample: Importance Sampling for Neural Materials

Bing Xu
b4xu@ucsd.edu
University of California, San Diego
USA

Liwen Wu
liw026@ucsd.edu
University of California, San Diego
USA

Miloš Hašan
mihasan@adobe.com
Adobe Research
USA

Fujun Luan
fluan@adobe.com
Adobe Research
USA

Iliyan Georgiev
igeorgiev@adobe.com
Adobe Research
UK

Zexiang Xu
zexu@adobe.com
Adobe Research
USA

Ravi Ramamoorthi
ravir@cs.ucsd.edu
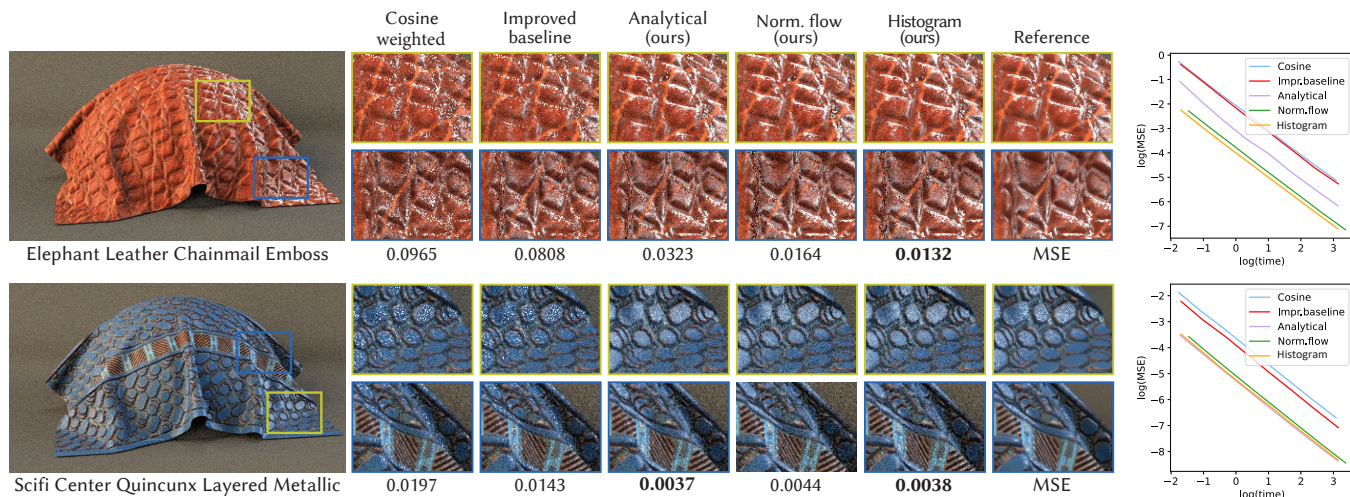University of California, San Diego
USA

Figure 1: Two neural materials rendered using five BRDF importance-sampling methods. To achieve equal render times, we adjust the number of samples per pixel (spp); from left to right: 33 spp, 32 spp, 32 spp, 25 spp, 32 spp.

ABSTRACT

Neural material representations have recently been proposed to augment the material appearance toolbox used in realistic rendering. These models are successful at tasks ranging from measured BTF compression, through efficient rendering of synthetic displaced materials with occlusions, to BSDF layering. However, importance sampling has been an after-thought in most neural material approaches, and has been handled by inefficient cosine-hemisphere sampling or mixing it with an additional simple analytic lobe. In this paper we fill that gap, by evaluating and comparing various pdf-learning approaches for sampling spatially varying neural materials, and proposing new variations of these approaches. We investigate three sampling approaches: analytic-lobe mixtures, normalizing flows, and histogram prediction. Within each type, we introduce improvements beyond previous work, and we extensively evaluate and compare these approaches in terms of sampling rate, wall-clock time, and final visual quality. Our versions of normalizing flows and histogram mixtures perform well and can be used in practical rendering systems, potentially facilitating the broader adoption of neural material models in production.

CCS CONCEPTS

• **Computing methodologies → Ray tracing**.

KEYWORDS

Rendering, neural materials, importance sampling, neural sampling, normalizing flows

1 INTRODUCTION

Rendering photo-realistic images of 3D scenes requires using physically plausible material models. Analytic bidirectional scattering distribution functions (BSDFs), typically based on microfacet theory, have been the dominant choice in the past decade. Combined with spatially varying parameters (diffuse and specular reflectance, roughness, normal and displacement maps), these models are capable of approximating a wide variety of appearances. However, not all real-world materials have reflectance functions that fit the analytic paradigm well. Furthermore, displacement mapping is expensive, and the cheaper normal mapping has only limited ability to realistically approximate surface mesostructure, ignoring occlusions, inter-reflections and parallax effects.

Recently, neural representations have been proposed to augment the material appearance toolbox [Rainer et al. 2019, 2020; Kuznetsov et al. 2021; Baatz et al. 2021; Sztrajman et al. 2021; Fan et al. 2022]. Unlike analytic models, neural models can theoretically represent any material, e.g., measured reflectance in the form of bidirectional texture function (BTF) data, or complex mesostructure defined using synthetic or real displacement data. The utility of these methods ranges from measured BTF compression, through efficient rendering of synthetic displaced materials with occlusions, to BSDF layering. The methods vary in their neural-network architectures, but they typically combine learned spatial-feature textures and fully connected blocks to approximate 6D reflectance functions over surface position, incoming and outgoing directions.

However, *importance sampling* has been an after-thought in most neural-material works. In NeuMIP [Kuznetsov et al. 2021] and its extension to curved-surface silhouettes [Kuznetsov et al. 2022], importance sampling is handled via simple Lambertian (i.e., cosine-hemisphere) sampling, which is ineffective for specular materials. Works on neural BRDF representation [Sztrajman et al. 2021] and BSDF layering [Fan et al. 2022] perform importance sampling via mixtures of Lambertian and Gaussian (or Blinn-Phong) lobes with network-predicted parameters; studying the efficiency of this approach has not been a core goal of these prior works. However, practical Monte Carlo rendering systems require all materials to provide three core interfaces at each surface shading point: evaluating the BSDF, sampling it, and evaluating the sampling probability density function (pdf); the pdf is expected to be approximately proportional to the BSDF. The latter two operations are needed for obtaining unbiased rendering as well as implementing multiple importance sampling (MIS) [Veach 1997] which is key to achieving robustness in complex production scenarios.

The objective of this paper is to fill this gap, by evaluating and comparing various pdf-learning approaches for sampling neural materials, as well as proposing new variations of these approaches. Specifically, we show our improved version of the *analytical mixture method* [Sztrajman et al. 2021; Fan et al. 2022], as well as improved

methods based on *normalizing flows* [Kobyzev et al. 2019] and a novel *histogram-mixture prediction* method inspired by Zhu et al. [2021] but significantly extended with new techniques. We find that both normalizing flows and histogram mixtures perform well across a selection of neural materials, while the improved analytic mixture is competitive for some but not all materials. In terms of wall-clock time, the histogram mixture method tends to perform best, while normalizing flows are also competitive, especially in terms of sampling rate (i.e., number of samples per pixel); performance depends on scene complexity and rendering cost. Our evaluation focuses on a NeuMIP-style architecture (Fig. 2), simplified to 2D feature textures instead of multi-resolution pyramids, but applies to any similar architectures parameterized by feature textures.

Our key contribution is a thorough evaluation of importance sampling methods for neural materials:

- We study three sampling approaches: analytic mixtures, normalizing flows, and histogram prediction (Section 3).
- Within each approach, we improve upon previous work and extensively evaluate and compare these approaches, in terms of sampling rate, wall-clock time, and final visual quality (Fig. 1 and Section 4).

Based on our evaluation results, we recommend our histogram mixture prediction as the overall best-performing method. However, our normalizing-flow variant can do better at equal sampling rate and could become overall best-performing in complex scenes. We believe our methods facilitate the broader practical adoption of neural material models in production rendering. They could also have impact beyond material importance sampling, such as in path guiding and complex luminaire sampling.

2 RELATED WORK

Neural materials. Learning to sample neural materials is an instance of the more general problem of learning a conditional probability distribution $p(\omega|\mathbf{x}, \omega_i)$, where $\mathbf{x}$ is a point in a 3D scene, ω_i is the direction of the incoming path and ω is an outgoing direction. This problem occurs repeatedly in 3D rendering: for example, path guiding is the problem of sampling, for a given path vertex $\mathbf{x}$, a direction ω that is likely to contribute significant incoming radiance $L(\mathbf{x}, \omega)$. A similar situation occurs in complex-luminaire rendering, where the goal is to sample a direction ω toward a luminaire point with strong emission. Therefore, methods from all of these areas are relevant to neural material sampling, and we will evaluate some of these ideas in our context (specifically, normalizing flows and histogram prediction methods), in addition to the simple analytical-mixture methods previously used for neural materials.

Neural-material representations have recently received much attention, as they show promise to overcome limitations of traditional analytic BSDFs with parameter textures and displacement maps, or measured and tabulated BTF data. Rainer et al. [2019, 2020] focused on BTF compression. [Kuznetsov et al. 2022] showed how to represent synthetic or real materials with significant parallax and occlusion effects in a neural form that is much more efficient to evaluate than true microgeometry. Sztrajman et al. [2021] and Fan et al. [2022] have approximated neural BSDFs with architectures allowing for high specularity and (in the latter work) layering operators. These approaches can theoretically represent any material data, be

it complex synthetic microgeometry with displacements, shadows and inter-reflections, or real measured reflectance functions. Neural materials are gaining in importance, but are still lacking a thorough exploration of importance sampling—a component that is crucial for broader practical adoption.

Analytic approximations. Cosine-hemisphere sampling of a neural material [Kuznetsov et al. 2021, 2022] is inefficient unless the material is very rough. To handle a wider variety of reflectance functions, some recent works fit analytic lobes to the local reflectance at a shading point, which have tractable pdf evaluation and sampling. Sztrajman et al. [2021] fit an entire Blinn-Phong model per spatial location, using a neural network that predicts two parameters (glossiness and diffuse/specular ratio) from a local feature vector φ. Fan et al. [2022] take a similar approach to predict a mixture of a Lambertian and Gaussian lobe (including mean and variance) given φ and ω_i. This approach is slightly more general, since the Gaussian lobe mean does not need to be on the reflected direction, allowing for additional effects like varying specular normal. While these analytic predictions can be sampled efficiently, they diminish the key advantage of neural materials, which is the ability to represent precisely the materials that do not fit well with analytic BRDF models. We propose and study an improved version of Fan et al.'s method with more Gaussian lobes which uses a similarly compact multi-layer perceptron (MLP) as Sztrajman et al. [2021] but extends to the complex spatially varying scenario. For training, Sztrajman et al. resort to supervised learning by using the fitted Blinn-Phong parameters of Ngan et al. [2005], while we directly maximize the likelihood of samples, avoiding the limitations of manually defined metrics.

Normalizing flows. A family of methods for fitting general probability distributions, called *normalizing flows*, has been proposed by the machine-learning community [Dinh et al. 2014; Kobyzev et al. 2019]. The key idea is to learn a bijective, invertible mapping $y = g(z; \theta)$ between the learned distribution $p(y)$ and a simple base distribution $p_z(z)$ (typically Gaussian or uniform). Moreover, the mapping g should have an easily computable Jacobian determinant. This allows for efficient sampling of the resulting pdf (by sampling the base distribution and passing the sample z through g) as well as efficient pdf evaluation (by mapping a point y through the efficient inverse $f = g^{-1}$ and computing the base pdf and the Jacobian determinant). The function g is sometimes called a *pushforward* while f^{-1} is called a *normalizing flow*, since it maps (flows) the complex target distribution onto a much simpler (often Gaussian), normalized distribution. Many architectures have been proposed to implement f and g. Please refer to the survey of Kobyzev et al. [2019] for an overview of the design decisions.

A general application of normalizing flows to sampling problems in graphics, such as path guiding, was presented by Müller et al. [2019]. They use a large U-shaped neural network with fully connected layers to predict parameters for the piecewise polynomial coupling transforms that warp the initial uniform distribution to the target. Müller et al. showed that the learned pdfs outperform state-of-the-art sampling methods for equal sample counts but not in terms of wall-clock time. They acknowledged that the practicality of their method is not immediate as its cost is higher than the

(a) Neural materials: glossy, anisotropic, bumpy, layered, etc

(b) NeuMIP network architecture

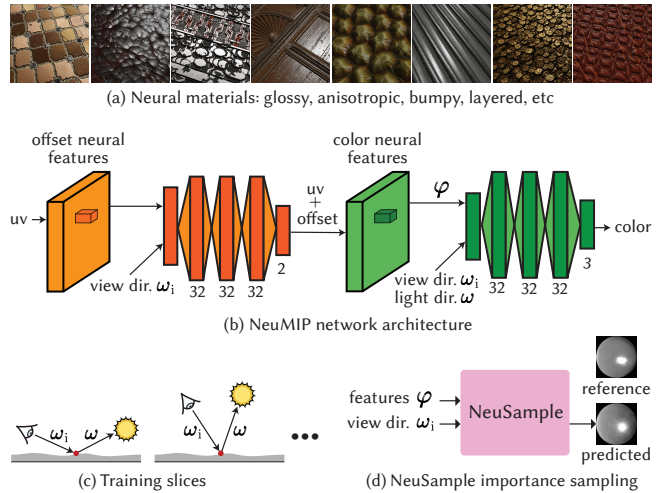(c) Training slices   (d) NeuSample importance sampling

Figure 2: Neural materials can (a) encode various appearance effects: specular/glossy reflectance, displacement, anisotropy, and multi-layered BRDFs. (b) The NeuMIP architecture uses two feature textures and two MLPs. Only the last step depends on the light direction, so it is the only part that requires importance sampling. (c) The model is trained on synthetically rendered slices of the 6D reflectance function with varying camera and lighting directions. (d) Our methods learn to importance sample neural materials, conditioned on the pretrained NeuMIP features.

gains from the better importance sampling achieved. They did not study importance sampling for materials.

For materials, Xie et al. [2019] used normalizing flows to approximate microfacet BRDFs with multiple scattering by fitting data simulated using geometric optics on actual heightfield microgeometry. Their method employs efficient affine transformations including conditional scaling and translation. While this exact architecture could be used for neural-material sampling, we found the fitted pdfs were not of sufficiently high quality, and we propose a modified architecture. The comparison can be seen in Table 2.

We build upon a more advanced invertible transformation (typically neural spline flows [Durkan et al. 2019]) introduced in the normalizing flow literature that we found to maximize expressiveness. We apply additional techniques to keep the architecture efficient enough to achieve a gain in wall-clock time.

Histogram prediction. Instead of learning invertible transformations or lobe mixtures, we can predict a *histogram* (a discretized pdf approximation) given the conditional information. This approach is only feasible in low dimensions but is effective for 2-dimensional material importance sampling. To our knowledge, the only related targets importance sampling complex luminaires [Zhu et al. 2021]. It learns an MLP to predict a small image of the luminaire from a given view, which is then used for evaluation and sampling. Our histogram-mixture method significantly extends this idea.

3 METHODS

In this section, we introduce key concepts, followed by the three sampling method types evaluated in this paper.

Preliminaries. We would like to find an approximation $p(\omega|\varphi,\omega_i)$ to a target pdf $p^*(\omega|\varphi,\omega_i)$ over outgoing directions ω, conditioned on an incoming direction ω_i and a local feature vector φ that encodes the material properties at a given location. The target pdf p^* is the luminance of the BRDF lobe at that location and direction, normalized to integrate to 1 over the hemisphere: this is the perfect importance-sampling distribution. We study different methods to approximate p^*: all are able to evaluate the approximation p for a given ω_i as well as sample ω from it efficiently. Note that the terms *incoming* and *outgoing* refer to the direction of simulation. We consider BRDFs, which are non-zero in the top hemisphere; the technical task of extending to full BSDFs we leave for future work.

Projected hemisphere. Our goal is to sample proportionally to the product of BRDF and cosine foreshortening on the unit hemisphere $\mathcal{H}$. This is equivalent to sampling the projection of the BRDF (without cosine) onto the unit disk $\mathcal{H}_\perp$. Such a pdf can be transformed into a hemispherical one via multiplication with the cosine term (division in the opposite direction). We choose to fit distributions of unit-disk projections $\omega_\perp \in \mathcal{H}_\perp$. For most methods, the planar unit disk is a more convenient domain for defining the pdfs.

Application to NeuMIP. Our approach could work for most neural-material representations, but we specifically learn sampling for a variant of NeuMIP [Kuznetsov et al. 2021], where the BRDF depends only on a learnt 8-dimensional feature vector φ that is retrieved from a given UV coordinate. In NeuMIP, the UV coordinate is corrected by a separate offset module to handle parallax effects; the offsetting happens before BRDF importance sampling and is orthogonal to our method. We do not consider the multi-resolution version of NeuMIP (trilinearly-interpolated pyramid of feature textures), nor its silhouette extension [Kuznetsov et al. 2022]. These methods are also orthogonal to BRDF importance sampling, so they are likely to work with our approach as well.

Three sampling methods. We consider three neural sampling approaches. The first one predicts a mixture of a few simple analytical lobes (Lambertian and Gaussian). The second one is based on normalizing flows, and the third predicts discretized histograms. Within each category we present a method that improves over previous work for our application. In the following subsections we describe these methods, and in Section 4 we analyze their performance in terms of variance reduction and computational cost.

Fitting. We train neural networks that fit the pdf p to the ground truth p^* by minimizing the KL divergence $D_{KL}(p^*\|p)$ between the two. This is equivalent to maximizing the log-likelihood (with respect to p) of directions sampled from the ground truth distribution p^*. In practice, in every training batch we sample from the ground-truth distribution $p^*(\omega_\perp|\varphi,\omega_i)$ (discretized to a high-resolution grid of directions $\omega_\perp$ for a randomly chosen value of the condition φ,ω_i), evaluate the log pdf of the trained model p, and back-propagate to update the neural-network parameters.

3.1 Mixture of analytical lobes

Improved baseline analytical mixture. A baseline method [Fan et al. 2022] approximates the desired pdf $p^*(\omega|\varphi,\omega_i)$ by predicting a combination of a Lambertian lobe and an isotropic 2D Gaussian lobe, given φ and ω_i. The predicted parameters are the (scalar) standard deviation σ of the Gaussian lobe and the relative weight w between the two lobes. However, this approach is too limited to represent even materials with local shading normals which are very common in neural BTFs including NeuMIP. To that end, we improve the baseline by also predicting the mean μ of the Gaussian lobe:

$$p(\omega_\perp|\varphi,\omega_i) = w\frac{1}{\pi} + (1-w)G(\omega_\perp;\mu,\sigma), \tag{1}$$

where $G(\omega_\perp;\mu,\sigma)$ is a normalized 2D Gaussian with mean μ and standard deviation σ, evaluated at the projected direction $\omega_\perp$. The inputs to the MLP are the feature vector φ and the direction ω_i in the local shading frame, while the outputs are the mixture parameters: μ, σ, and w. Note that the Lambertian pdf on the projected hemisphere is a constant $1/\pi$, i.e., uniform sampling on the projected hemisphere is equivalent to cosine sampling on the hemisphere.

Our analytical mixture. To better capture multi-modal highlights or highlights with non-Gaussian or asymmetric falloff, we further choose a mixture of one Lambertian lobe and two axis-aligned anisotropic Gaussian lobes with diagonal covariance matrices:

$$p(\omega_\perp|\varphi,\omega_i) = w_1\frac{1}{\pi} + w_2\,G_d(\omega_\perp;\mu_1,\sigma_1) + w_3\,G_d(\omega_\perp;\mu_2,\sigma_2), \tag{2}$$

where $G_d(\omega_\perp;\mu,\sigma)$ is a 2D Gaussian with mean μ and standard deviations $\sigma = (\sigma_x, \sigma_y)$ in the x and y axes. The three weights are positive and sum to 1, and are predicted by a simple MLP with one hidden layer, along with the corresponding Gaussian means and standard deviations. Similarly, the inputs to the MLP are a feature vector φ and direction ω_i in the local shading frame, and the outputs are the parameters of the mixture: $\mu_1, \sigma_1, \mu_2, \sigma_2, w_1, w_2, w_3$ (see Fig. 3). More components or full anisotropy can potentially capture more complex distributions but increase the fitting difficulty and the computational workload for common cases with only one highlight. We compromise with two diagonal Gaussian lobes.

Invalid samples. The support of the above two pdf mixtures is wider than the unit disk, i.e., outside the BRDF support, meaning that in practice some samples will be invalid (i.e., off the hemisphere). The resulting Monte Carlo estimates thus have zero values, as also happens for importance sampling analytic BRDFs like microfacet models [Walter et al. 2007], without posing practical issues.

3.2 Normalizing flows

Neural materials have arbitrary reflection profiles, embedded local normals, texture, and potentially complex layered (anisotropic) behaviors. A mixture of simple lobes with a few trainable parameters is not expressive enough to always provide accurate importance sampling, also considering the difficulty in determining the number of mixture components required for all materials. Normalizing flows provide a compelling alternative for our application, as they support both sampling and density evaluation, and have been shown to be able to fit complex, multi-modal distributions.
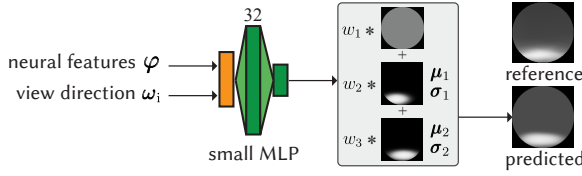
Figure 3: Our analytical-lobe mixture method takes as input the neural-material features at the UV surface location and the viewing direction to infer the parameters of three lobes, one Lambertian and two anisotropic Gaussians, to capture diffuse reflection and (potentially multi-modal) highlights. High performance is ensured by using an MLP with one hidden layer.

Normalizing flows learn a bijective mapping between a simple base distribution and (an approximation of) a complex target distribution, and can be used to generate samples from the latter. A carefully designed architecture with an autoregressive property ensures the mapping's Jacobian is triangular, with an easy-to-compute determinant, which is needed for pdf evaluation. The architecture also ensures tractable invertibility of the mapping [Dinh et al. 2016; Papamakarios et al. 2017], useful for multiple importance sampling.

As a base distribution p_z we use a 2D Gaussian. We learn a bijective function g to map samples z from p_z to samples $y = g(z|\boldsymbol{\varphi}, \boldsymbol{\omega}_i)$ that approximately follow the target $p^*(\cdot|\boldsymbol{\varphi}, \boldsymbol{\omega}_i)$. If the inverse $f = g^{-1}$ and its Jacobian determinant are efficiently computable, we can evaluate the learned distribution p at a point (direction) $\boldsymbol{\omega}_\perp$ as

$$p(\boldsymbol{\omega}_\perp|\boldsymbol{\varphi}, \boldsymbol{\omega}_i) = p_z(z) \cdot \left|\frac{\partial z}{\partial \boldsymbol{\omega}_\perp}\right|, \quad \text{where} \quad z = f(\boldsymbol{\omega}_\perp|\boldsymbol{\varphi}, \boldsymbol{\omega}_i). \quad (3)$$

This framework satisfies our goal of fitting pdfs that can be evaluated and sampled from efficiently, provided we can find a neural network to represent f (and g) with the desired properties. A detailed exploration of the neural architectures used for normalizing flows is out of our scope, and we refer the reader to the surveys of Kobyzev et al. [2019] and Papamakarios et al. [2021].

We did ablations on representative state-of-the-art transformations, from simple affine transformations [Dinh et al. 2016] to free-form monotonic neural networks [Wehenkel and Louppe 2019; Huang et al. 2018]. The former has limited flexibility due to the simple transformation while the latter lacks an analytical inverse. We concluded that monotonic piecewise polynomials [Müller et al. 2019; Durkan et al. 2019; Dolatabadi et al. 2020] achieve the best balance, providing similar expression power while being efficient for computing the inverse. We build upon monotonic piecewise rational quadratic splines [Durkan et al. 2019] which provide more flexibility compared to RealNVP [Dinh et al. 2016] and piecewise polynomials [Müller et al. 2019]. On the other hand, since we are working with distributions of only two dimensions, there is no significant difference in performance between coupling layers and autoregressive architectures. We choose coupling transforms due to their simplicity. Namely, each invertible transformation is applied onto only one dimension of $\boldsymbol{\omega}_\perp$ and we couple the two transformations together to fuse the dimensions.

Below we detail the monotonic piecewise rational quadratic (RQ) spline and our corresponding application. We split the square
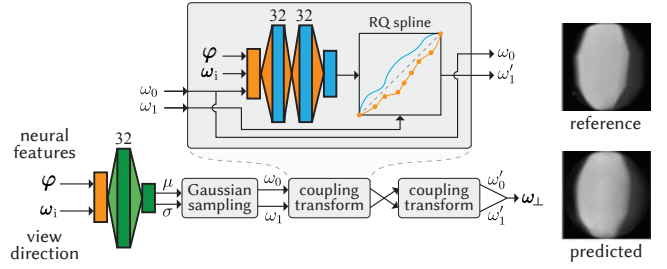

Figure 4: Our lightweight normalizing-flow model transforms input samples (ω_0, ω_1) to a (projected) direction $\boldsymbol{\omega}_\perp = (\omega_0', \omega_1')$. The input samples are generated from a base Gaussian distribution with MLP-inferred mean and standard deviation. Each is then warped by an analytically invertible piecewise rational quadratic (RQ) spline. The spline parameters (bin widths, bin heights, derivatives) are inferred by another small MLP.

region $[-1, 1]$ of the initial z space and target $\boldsymbol{\omega}_\perp$ space into several intervals. Then within each interval, we learn a monotonically increasing rational-quadratic function (we omit the interval index for brevity):

$$z = f(\boldsymbol{\omega}_\perp|\boldsymbol{\varphi}, \boldsymbol{\omega}_i) = z_0 + \frac{(z_1 - z_0)[s\xi^2 + \delta_0\xi(1-\xi)]}{s + [\delta_1 + \delta_0 - 2s]\xi(1-\xi)}, \quad (4)$$

where ω is just one dimension (x or y) of $\boldsymbol{\omega}_\perp$, $\xi(z) = \frac{\omega - \omega_0}{\omega_1 - \omega_0}$, $s = \frac{z_1 - z_0}{\omega_1 - \omega_0}$, and $\omega_{0(1)}$, $z_{0(1)}$ and $\delta_{0(1)}$ are the locations and derivatives at the left (right) boundary of each interval. Those are predicted by a 3-layer MLP taking $(\boldsymbol{\varphi}, \boldsymbol{\omega}_i)$ and the other dimension of $\boldsymbol{\omega}_\perp$ as input. We use 20 intervals to capture the pdf variation. Solving and selecting the correct root of a quadratic equation gives the inverse of the transformation. We refer the reader to Durkan et al. [2019] for Jacobian computation and further details. Since we have two dimensions in the target pdf variable $\boldsymbol{\omega}_\perp$, the 3-layer MLP inference and the inverse of the above transformation need to run twice during sampling. Note that the predicted splines can be shared between the pdf query and sample routine for multiple importance sampling at each query point $(\boldsymbol{\varphi}, \boldsymbol{\omega}_i)$. Figure 4 illustrates our architecture.

The rich information encoded in the conditional neural feature vector $\boldsymbol{\varphi}$ helps us to greatly simplify our importance sampling network of invertible transformations to capture the complex spatially varying pdfs. Unlike previous work [Müller et al. 2019; Xie et al. 2019], we further use a conditional Gaussian base distribution that depends on $(\boldsymbol{\varphi}, \boldsymbol{\omega}_i)$, instead of using a uniform distribution or a Gaussian distribution with fixed zero mean and unit variance. Conditional normalizing flows have been studied for super-resolution and image segmentation tasks [Winkler et al. 2019]. As we show in Section 4 below, this conditional distribution helps us to further reduce the total architecture size (one 3-layer MLPs used in the flow, and one 2-layer MLP to predict the base-Gaussian parameters). Moreover, the conditional probability is guaranteed to be normalized by construction. The flat log-log convergence plots in Fig. 8 have the expected slope, which confirms the unbiasedness.

3.3 Histogram prediction

Another approach for importance sampling is to directly predict an easy-to-sample distribution. We opt for a piecewise constant 2D distribution, i.e., a histogram, represented as a regular grid of bin values. Following previous work which uses a low-resolution predicted rendering of a complex luminaire as a sampling distribution [Zhu et al. 2021], we predict a histogram approximating $p^*(\omega_\perp|\varphi, \omega_i)$ through an MLP taking the condition (φ, ω_i) as input.

We found that a direct application of this idea does not perform well in our setting (called 'histogram (baseline)' in Table 2). The reason is likely that while a luminaire can be always centered in the predicted histogram, our pdf lobes can have very different centers and shapes as the condition (φ, ω_i) varies. To this end, we present a variant that fits our distributions better and is also computationally efficient. Instead of directly predicting a histogram for each query, we assume these histograms can be decomposed into histogram mixtures, i.e., weighted combinations of shared-basis histograms:

$$p(\omega_\perp|\varphi, \omega_i) = \sum_{k=1}^{K} w_k(\varphi, \omega_i)\, H_k(\omega_\perp\,|\,\omega_k, t_k(\varphi, \omega_i)), \quad (5)$$

$$\omega_k = R(\omega_i, \phi_k(\varphi, \omega_i)), \quad (6)$$

where H_k are K basis histograms, globally shared across the neural material, w_k are the corresponding mixture weights; R represents azimuthal rotation, $\phi_k \in [0, 2\pi]$ are rotation angles applied to ω_i, and $t_k \in [0, 1]$ is a scalar latent code. The rationale behind the 2D (ϕ_k, t_k) parameterization is to efficiently encode continuous changes in the BRDF lobe as the incoming direction ω_i varies. As can be seen in Fig. 6, both the lobe's position and shape change. The code t_k captures the shape change and part of the rotation. This design reduces the number of mixture components needed for fitting the pdf compared to simpler designs.

Figure 5 illustrates our histogram-mixture architecture; w_k, ϕ_k, t_k are encoded into a small MLP that takes the condition (φ, ω_i) as input, and H_k are implicitly encoded into a (ω, t_k)-dependent MLP. After training, basis histograms are baked into a $K \times T \times N \times N$ tensor for fast query (K basis histograms with $N \times N$ resolution, T latent-code discretization levels). This scheme allows us to use a larger MLP during training to better fit the pdf, without hurting the speed of the inference which performs fast tensor look-ups.

Training. We use $K = 10$ basis histograms for each neural material and train the network using L_2 loss between histogram and ground truth pdf response for randomly sampled tuples $(\varphi, \omega_i, \omega)$:

$$l = \sum_{\omega, \varphi, \omega_i} \left\| p(\omega|\varphi, \omega_i) - p^*(\omega|\varphi, \omega_i) \right\|_2^2. \quad (7)$$

We have found this loss to perform better than KL divergence. After training, we tabulate histograms of resolution $N \times N = 64 \times 64$ and discretize the latent code t into $T = 100$ equi-spaced values.

Since t_k is learned implicitly, it is not guaranteed to be distributed uniformly in $[0, 1]$. To encourage such distribution, and to reduce post-training discretization error, we add a quantization term to the above loss: $l_q = ||t_k - \hat{t}_k||_2^2$, where $\hat{t}_k$ is the quantization of t_k.
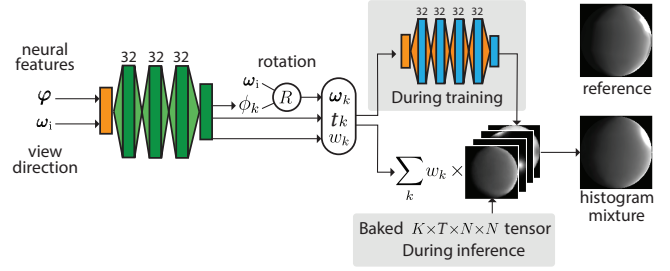

Figure 5: Our histogram-mixture model sends the neural-feature vector φ and incoming direction ω_i to a small MLP to infer a weight, azimuth angle, and latent code for each basis histogram. These are then combined to produce the output histogram. The histogram is encoded by another MLP during training, and is then baked into a 4D tensor for efficient sampling during inference. A more basic approach would use the initial (green) MLP to directly predict a 2D sampling histogram from φ and ω_i. While fast and simple, that approach cannot accurately model the wide variety of lobe shapes in the large space of the condition (φ, ω_i) for every material.
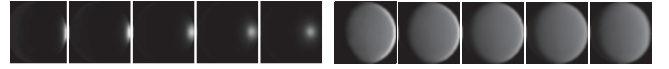

Figure 6: Our histogram approach can efficiently model continuous changes in the pdf lobe shape based on the incoming angle (columns) using the learned rotation and scalar latent code. The change in lobe shape cannot be explained by rotation alone, and the latent code t helps significantly. Different basis histograms (left and right) are necessary to represent different modes of the pdf lobes and/or different spatial locations on the material.

4 RESULTS

We now present an evaluation of our three proposed methods. The supplemental document discusses ablations around design choices.

We implemented all our sampling techniques in PyTorch [Paszke et al. 2019] and integrated them into Mitsuba 3 [Jakob et al. 2022]. Every sampling and pdf-evaluation call for a specific material is run on a wavefront of rays. All results are produced using a single NVIDIA RTX 3090 GPU. Our implementations are renderer-agnostic and could be easily integrated into other systems. We will make our code and data publicly available.

Stratification and MIS. For simplicity, and to ensure correctness, all our results use independent pseudo-random sampling, showing slopes of -1 on log-log plots due to the linear variance reduction with increasing sample count. One can also utilize (low-discrepancy) stratification; we have verified this helps convergence slightly and does not change the relative ordering of the methods.

Most of our results utilize only BRDF sampling as it is our main focus. Emitter sampling in combination with multiple importance sampling (MIS) works as expected with our methods, since we support all required sub-routines for pdf evaluation and sampling. We utilize MIS in Fig. 11 as well as in the *Cowhide* and *Metal* scenes in Fig. 9, which are all rendered with global illumination.

Table 1: Timing breakdown for GPU rendering of NeuMIP materials on the scene from Fig. 1. The neural-material evaluation itself takes 0.025 s/spp and is included in the total rendering time. This simple scene is an almost worst-case scenario for the overhead of (our) advanced importance sampling. In practical scenes this overhead diminishes quickly.

Time (s/spp)	Cosine weighted	Improved baseline	Analytical (ours)	Norm. flow (ours)	Histogram (ours)
Rendering	0.0412	0.0441	0.0435	0.0553	0.0443
Sampling	0.0001	0.0018	0.0012	0.0065	0.0018

Table 2: Importance-sampling speed-up factors, in terms of the relative runtime the improved baseline (Section 3.1) needs to achieve equal quality. We also include normalizing-flow [Xie et al. 2019] and histogram [Zhu et al. 2021] baselines. For each material we highlight the best improvement factor.

Material	Cosine weight.	Improved baseline	Analyt. (ours)	Norm. flow (baseline)	Norm. flow (ours)	Histogram (baseline)	Histogram (ours)
Egyptian	0.05×	1×	1.87×	0.82×	2.36×	0.03×	1.62×
Cowhide	0.66×	1×	1.06×	1.02×	1.48×	0.46×	1.56×
Elephant	0.84×	1×	2.50×	1.57×	4.92×	0.40×	6.12×
Giallo	0.88×	1×	0.74×	0.76×	1.28×	1.14×	1.27×
Moroccan	0.30×	1×	3.39×	0.89×	6.78×	0.13×	3.03×
Scifi	0.17×	1×	3.06×	1.56×	5.17×	0.03×	1.45×
Sheep	0.07×	1×	1.17×	0.28×	1.31×	0.01×	1.21×
Quincunx	0.73×	1×	3.86×	1.46×	3.25×	0.96×	3.76×
Melted	0.12×	1×	1.29×	0.90×	1.44×	0.85×	1.39×
Tortoise	0.83×	1×	1.02×	2.78×	3.32×	1.42×	2.88×
Victorian	0.35×	1×	1.06×	0.91×	1.17×	1.07×	1.34×
Wagon	0.23×	1×	1.16×	0.30×	1.38×	0.01×	1.07×

Performance. Table 1 provides a breakdown of the sampling and total rendering time for direct lighting in the scene in Fig. 1. The overhead of our importance-sampling techniques ranges from 2% to 12% of the total rendering time. Our improved analytical and histogram-mixture methods each use one shallow MLP, two- and four-layer respectively, resulting in a small 2–4% overhead. Our normalizing-flow sampler is slower as it performs three shallow-MLP evaluations (one two-layer and two three-layer) and some sequential batch operations, e.g., calculating the CDFs along the polynomial intervals necessary for inverse-inference root-finding. That sampler can generate 0.78× number of samples to render in equal time (see Fig. 1). In more complex practical scenes, the overhead of our neural sampling becomes even more negligible.

Pdf prediction. In Fig. 7 we plot pdfs predicted by all sampling methods on four materials at randomly chosen conditions (φ, ω_i). The improved, normal-mapping-aware baseline is able to find the isotropic highlight when the pdf varies little over the surface (*Goat Leather Clover Quilt*) but fails to capture anisotropy (*Stylized Melted Metal*). Our analytical method improves on this shortcoming but still suffers from the limited expressiveness of the simple analytical lobes to cover the entire 4D conditioning space (UV and incoming direction). Our normalizing-flow and histogram alternatives greatly improve the fitting accuracy, leading to more efficient sampling.

Rendering results. Figures 1, 9 and 10 show equal-time comparisons of all methods on eleven neural materials. The *Turtle Shell* and *Wagon Fine Wood Panelling* materials are uniformly lit, and for the others we use a more complex environment map to show specular effects. *Cowhide Leather* and *Stylized Melted Metal* also show global illumination. The top three materials in Fig. 9 have nearly isotropic highlights everywhere; *Victorian Fabric* has multiple lobes, while the bottom three and Fig. 10 have anisotropy. The improvement factors for our methods are shown in Table 2 where we also include two additional baselines: practical normalizing-flow sampling [Xie et al. 2019] and naive histogram sampling [Zhu et al. 2021].

Our proposed methods consistently outperform the cosine and baseline methods used in previous works. Please note that the "baseline" method we provide is actually significantly improved by us. The original baseline [Sztrajman et al. 2021; Fan et al. 2022] would fail to beat even cosine sampling, since it cannot learn normal mapping. We also observed that suboptimal solutions can perform well at some (φ, ω_i) conditions but create problems in others, resulting in overall worse performance (see *Giallo Marble*, where the analytical method is worse than baseline). Instead, the two alternatives we recommend are robust and can handle these complex variations. Normalizing flows perform best for some materials, and are especially good at capturing anisotropy (Fig. 10 and bottom three materials in Fig. 9). Histogram mixtures perform best in some scenarios, and are especially good at multi-coating materials with complex variation (*Victorian Fabric*) while bringing just a tiny overhead. These two methods outperform the analytical one in most complex cases with difficult spatial BRDF variations, while remaining robust on simpler materials (*Cowhide Leather*, *Sheep Leather*). Overall, our samplers significantly reduce noise at low sampling rates, cutting the rendering cost when more neural materials are applied in a practical application. Table 2 shows that the benefits range from 1.28× (*Giallo Marble*) to 6.78× (*Moroccan Tiles*), with a mean of 2.58× corresponding to a time benefit of 158%.

The "tabulated" column in Fig. 10 employs sampling from an impractical, near-optimal pdf that is a brute-force 2048×2048 discretization of the ground-truth BRDF distribution at each shading point. Those results still show some small error due to using the BRDF luminance to importance sample all three color channels. Our practical methods achieve similar performance in comparison.

We further plot error convergence in Fig. 8 to quantify the benefit of using our neural samplers. The normalizing flow mostly performs on par with the histogram mixture, better for anisotropic materials (equal-spp plots in first row), but the latter does best in most common cases thanks to its much smaller overhead (equal-time plots in second row). When the geometry is not too complex, we recommend our histogram mixture, to benefit from the advanced importance sampling with nearly no sacrifice in computational cost. The normalizing flow is a good choice for anisotropic materials.

Fig. 11 shows a scene with multiple neural materials, rendered with our normalizing-flow sampler with 128 spp and 12 light bounces. Our sampler handles global illumination well and can be used for practical rendering needs. The four stools are textured with the *Elephant Leather* material, the table is made of *Stylized Melted Metal* and *Giallo Antico Marble*, creating a realistic embossing effect. The bottle is textured with the *Stylized Light Bulb Screw* material.

Failure cases and limitations. As seen in some zoom-ins in Fig. 9, it is not guaranteed that our proposed methods outperform the improved baseline at every condition (φ, ω_i). Our analytical method

has almost no advantage on simple isotropic materials with a single highlight, though it helps in more challenging cases. Normalizing flows have longer inference time (Table 1), thus not showing gains on simpler scenes and materials. We also observe that in rare extreme cases, histogram mixtures can fail to predict the pdf lobe when its shape at a neighboring condition (φ, ω_i) changes rapidly.

Summary and recommendations. Based on our experiments, one should use our analytical mixture method only when it is known beforehand that the material has simple lobes with only a few highlights. The other alternatives provide a more general solution without bringing extra rendering overhead. In terms of wall-clock time, in our experiments the histogram method achieves the best variance reduction, which makes it our default recommendation. The method is also relatively easy to use at inference time thanks to the precomputed discretized tensor replacing a large MLP. However, wall-clock times are heavily dependent on scenes and the renderer. The normalizing flow may sometimes be the winning option, especially with anisotropy or in more complex settings with expensive ray tracing or material evaluation. Future code optimization for the MLP and RQ-spline evaluation may further speed up these methods; it is possible that hardware support for these operations on the GPU will become available, further increasing the gains from accurate sampling. Our improved analytic mixture is an additional option if easy implementation and very fast sampling are priorities.

5 CONCLUSION AND FUTURE WORK

In this paper, we evaluated and compared several importance sampling approaches for neural material representations. We studied three types of methods: analytic mixtures, normalizing flows, and histogram prediction. While simple analytic mixtures of Lambertian and Gaussian lobes have been used before, our version outperforms them as they lack the ability to handle dramatic normal variations, anisotropy, and layered materials. Moreover, we introduced variations of normalizing flows and histogram mixtures based on novel design ideas that perform well across the board, and we can recommend either approach for practical neural material implementations. These approaches provide a complete toolbox to enable the use of spatially varying neural materials in production rendering.

Our work opens several future directions. Our sampling models, like many neural material models themselves, are not universal and need to be trained and precomputed for every material separately; finding a universal evaluator and sampler architecture, where only the feature textures change, would be valuable. Another direction would be to jointly train sampling and evaluation models; currently the sampling model fits the evaluation model including its approximation error. Finally, our solution is not limited to neural materials, and could be applicable to other importance-sampling tasks where directions are sampled conditionally on scene position, such as path guiding, complex luminaire sampling, and portal sampling.

ACKNOWLEDGMENTS

REFERENCES

Hendrik Baatz, Jonathan Granskog, Marios Papas, Fabrice Rousselle, and Jan Novák. 2021. NeRF-Tex: Neural Reflectance Field Textures. In *Eurographics Symposium on Rendering - DL-only Track*, Adrien Bousseau and Morgan McGuire (Eds.). The Eurographics Association.

Laurent Dinh, David Krueger, and Yoshua Bengio. 2014. NICE: Non-linear Independent Components Estimation. arXiv:1410.8516 [cs.LG]

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803* (2016).

Hadi Mohaghegh Dolatabadi, Sarah Erfani, and Christopher Leckie. 2020. Invertible generative modeling using linear rational splines. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 4236–4246.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. 2019. Neural spline flows. *Advances in neural information processing systems* 32 (2019).

Jiahui Fan, Beibei Wang, Milos Hasan, Jian Yang, and Ling-Qi Yan. 2022. Neural Layered BRDFs. In *ACM SIGGRAPH 2022 Conference Proceedings* (Vancouver, BC, Canada) *(SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.1145/3528233.3530732

Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. 2018. Neural autoregressive flows. In *International Conference on Machine Learning*. PMLR, 2078–2087.

Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, and Delio Vicini. 2022. Dr.Jit: A Just-In-Time Compiler for Differentiable Rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)* 41, 4 (July 2022). https://doi.org/10.1145/3528223.3530099

Ivan Kobyzev, Simon J. D. Prince, and Marcus A. Brubaker. 2019. Normalizing Flows: An Introduction and Review of Current Methods. arXiv:1908.09257 [stat.ML]

Alexandr Kuznetsov, Krishna Mullia, Zexiang Xu, Miloš Hašan, and Ravi Ramamoorthi. 2021. NeuMIP: multi-resolution neural materials. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–13.

Alexandr Kuznetsov, Xuezheng Wang, Krishna Mullia, Fujun Luan, Zexiang Xu, Miloš Hašan, and Ravi Ramamoorthi. 2022. Rendering Neural Materials on Curved Surfaces. *SIGGRAPH '22 Conference Proceedings* (2022), 9 pages. https://doi.org/10.1145/3528233.3530721

Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. 2019. Neural Importance Sampling. *ACM Transactions on Graphics (TOG)* 38, 5, Article 145 (2019), 19 pages.

Addy Ngan, Frédo Durand, and Wojciech Matusik. 2005. Experimental Analysis of BRDF Models. *Rendering Techniques* 2005, 16th (2005), 2.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. 2021. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research* 22, 1 (2021), 2617–2680.

George Papamakarios, Theo Pavlakou, and Iain Murray. 2017. Masked autoregressive flow for density estimation. *Advances in neural information processing systems* 30 (2017).

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

Gilles Rainer, Abhijeet Ghosh, Wenzel Jakob, and Tim Weyrich. 2020. Unified Neural Encoding of BTFs. *Computer Graphics Forum (Proceedings of Eurographics)* 39, 2 (2020), 167–178.

Gilles Rainer, Wenzel Jakob, Abhijeet Ghosh, and Tim Weyrich. 2019. Neural BTF Compression and Interpolation. *Computer Graphics Forum* (2019).

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

Eric Veach. 1997. *Robust Monte Carlo Methods for Light Transport Simulation*. Ph. D. Dissertation. Stanford University.

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

Antoine Wehenkel and Gilles Louppe. 2019. Unconstrained monotonic neural networks. *Advances in neural information processing systems* 32 (2019).

Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. 2019. Learning likelihoods with conditional normalizing flows. (2019). arXiv:1912.00042

Feng Xie, Anton Kaplanyan, Warren Hunt, and Pat Hanrahan. 2019. Multiple Scattering Using Machine Learning. In *ACM SIGGRAPH 2019 Talks* (Los Angeles, California) *(SIGGRAPH '19)*. Association for Computing Machinery, New York, NY, USA, Article 70, 2 pages. https://doi.org/10.1145/3306307.3328144

Junqiu Zhu, Yaoyi Bai, Zilin Xu, Steve Bako, Edgar Velázquez-Armendáriz, Lu Wang, Pradeep Sen, Miloš Hašan, and Ling-Qi Yan. 2021. Neural Complex Luminaires: Representation and Rendering. *ACM Trans. Graph.* 40, 4, Article 57 (jul 2021), 12 pages. https://doi.org/10.1145/3450626.3459798
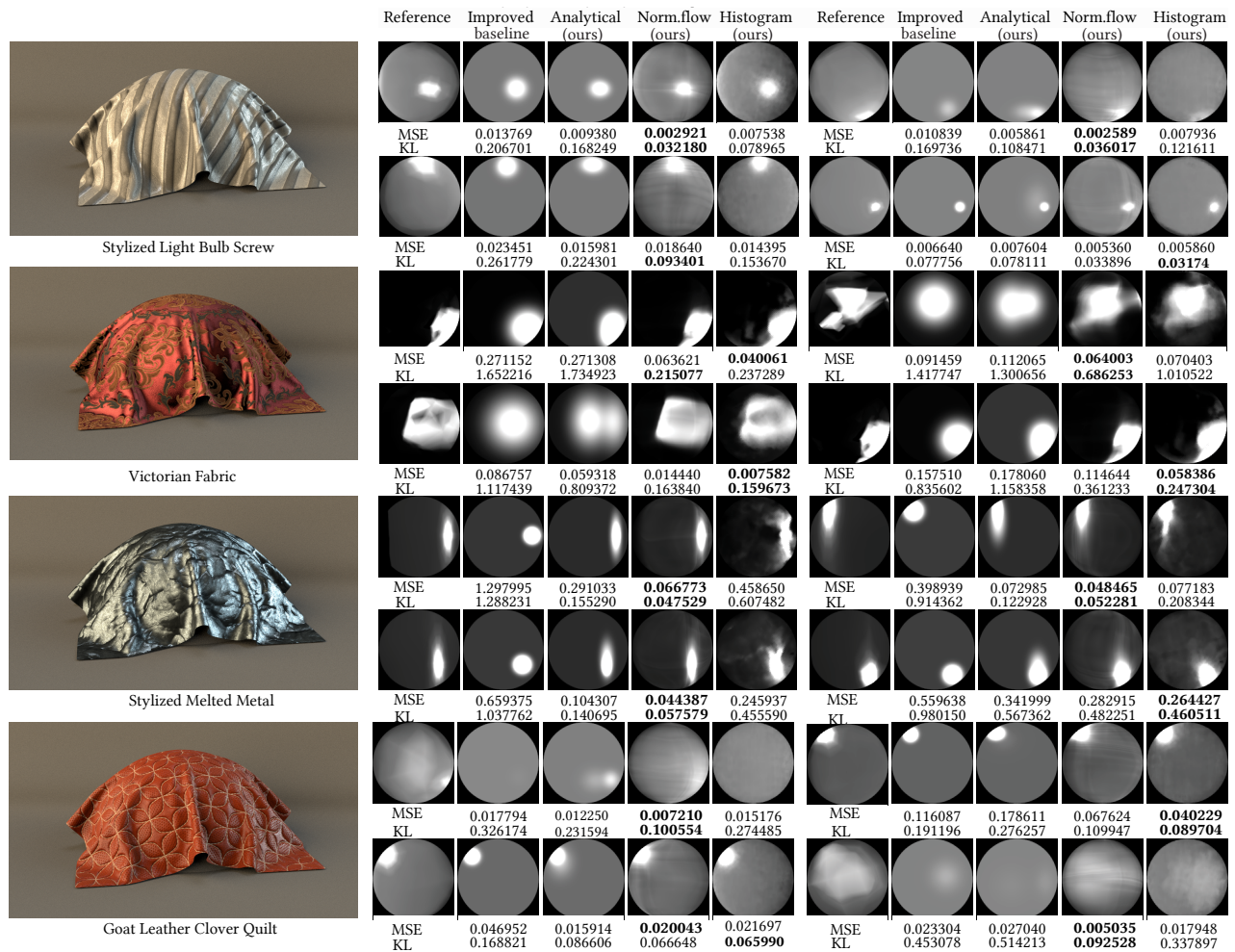
Figure 7: Comparison of pdf prediction of various importance-sampling methods on four neural materials. Each column represents the predicted pdf of the outgoing direction at a random pair of UV coordinates and incoming direction. We show the corresponding mean squared error (MSE) and KL divergence numbers below each result w.r.t. the reference pdf plotted image.
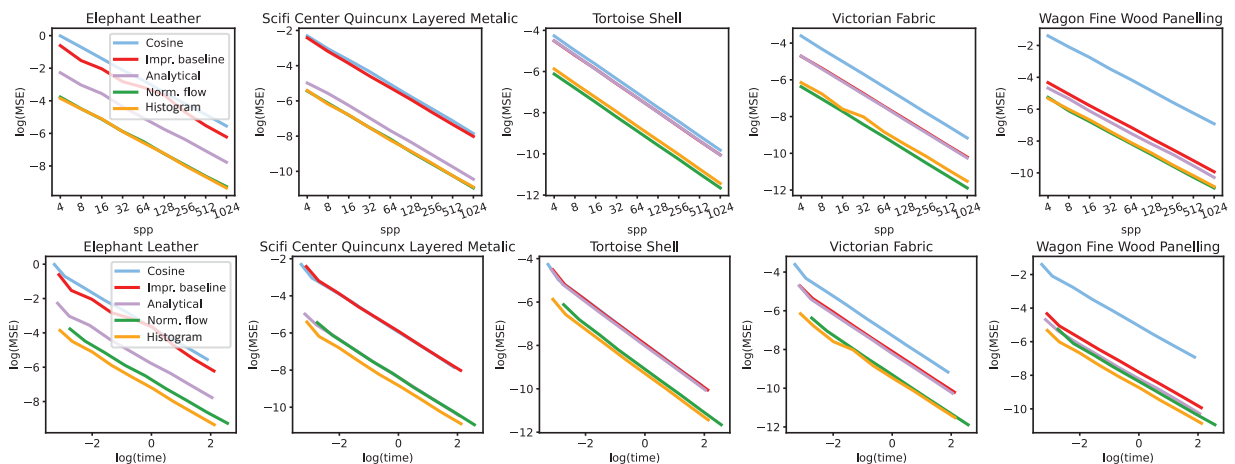

Figure 8: Convergence graphs. Top row: log-log plot of pixel MSE w.r.t. samples per pixel. Bottom row: log-log plots of rendering MSE w.r.t. rendering time. Please refer to the supplementary material for graphs on more neural materials.
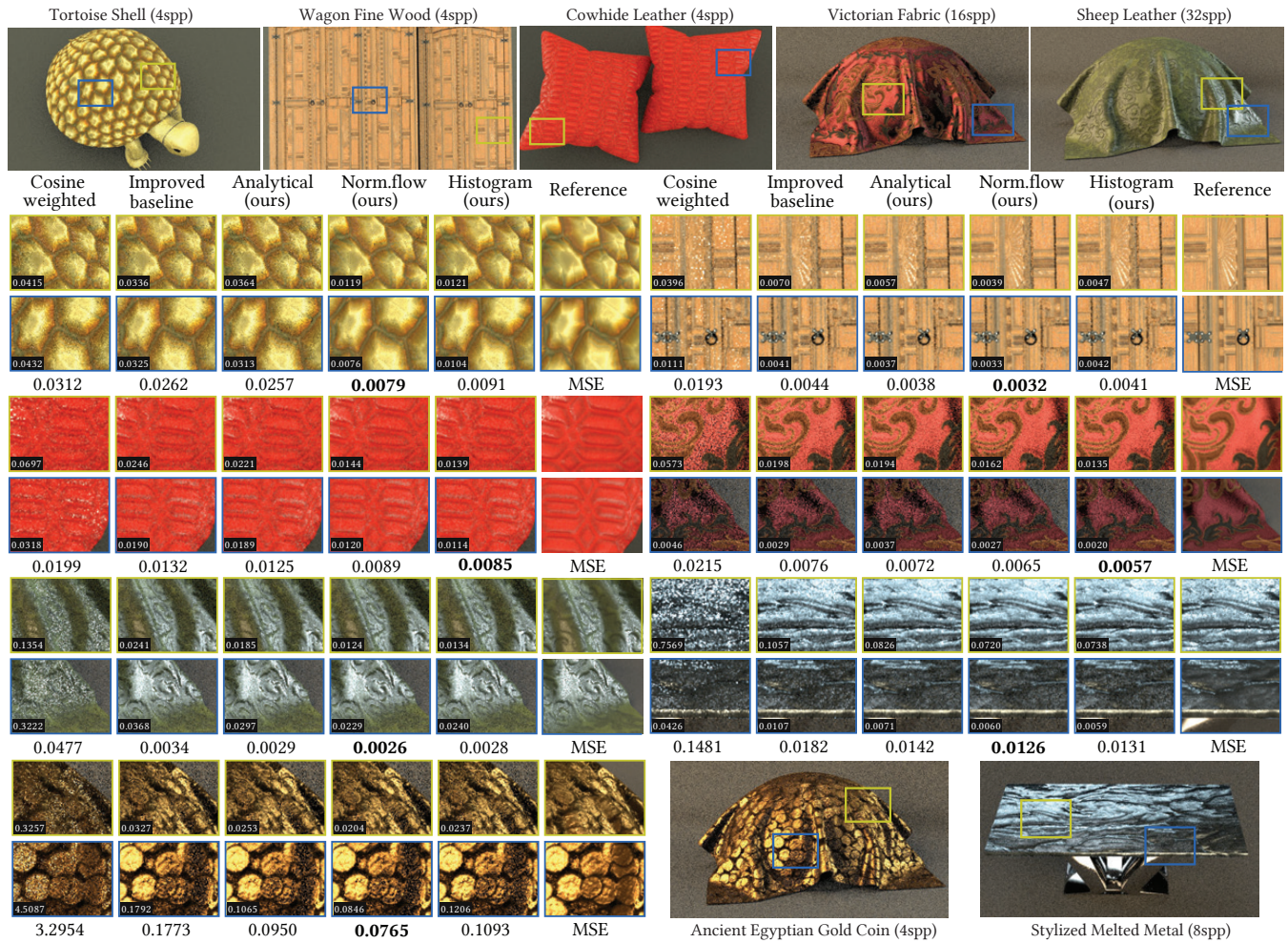
Figure 9: Equal-time rendering comparison of various neural materials under constant (top row) and environment-map (remaining rows) lighting. *Cowhide Leather* and *Stylized Melted Metal* are rendered with global illumination. First three materials have one isotropic highlight each; *Victorian Fabric* has multiple lobes; bottom three have anisotropic highlights. As in Fig. 1, spp are adjusted for cosine weighted (1.03×) and normalizing flow (0.78×) on simple scenes to achieve equal render time.
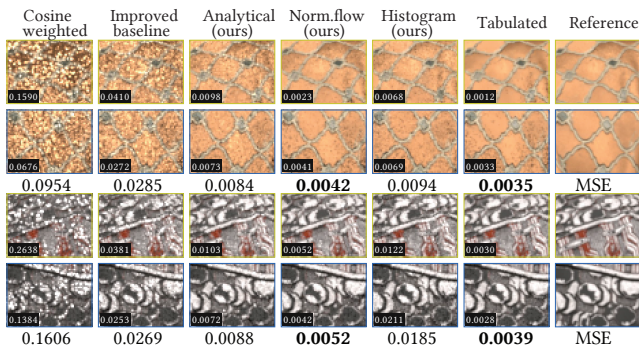

Figure 10: Equal-time rendering comparison of two complex anisotropic materials. The "tabulated" column samples from a near-optimal discretized ground-truth pdf; it takes hours to render and is only for comparison to the reference.
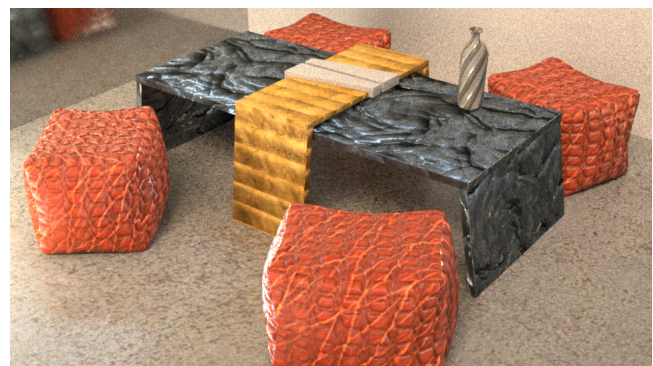

Figure 11: A more complex scene containing multiple neural materials, rendered with global illumination using our normalizing-flow sampler.