BSDF importance sampling using a diffusion model

ZIYANG FU, University of California San Diego, United States of America
YASH BELHE, University of California San Diego, United States of America
HAOLIN LU, University of California San Diego, United States of America
LIWEN WU, University of California San Diego, United States of America
BING XU, University of California San Diego, United States of America
TZU-MAO LI, University of California San Diego, United States of America

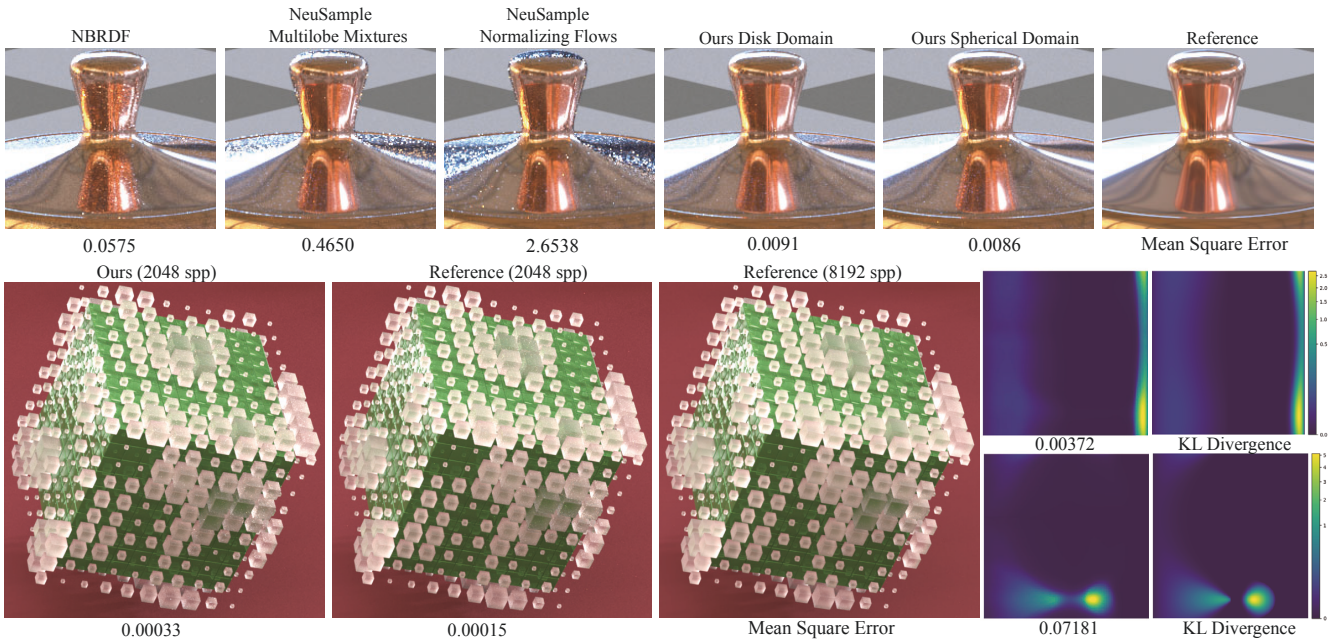


Fig. 1. Top Row: Equal-time comparison of our method against NBRDF [Sztrajman et al. 2021] and NeuSample [Xu et al. 2023] for sampling the specular metal material COPPER-SHEET, under global illumination using BRDF sampling only. Bottom Row: (left) Rendering results for a rough dielectric material [Walter et al. 2007] using our model and reference analytical sampling method at 2048 samples per pixel (spp). (right) PDF slices for a fixed incoming direction ω_i, that is close to normal (top) and approaching the grazing angle (bottom).

Previous neural sampling methods, primarily using analytical lobe mixtures and normalizing flows, often struggle with specular materials, particularly at grazing angles. Furthermore, they are limited to reflection, and do not handle transmission. Our key observation is that previous normalizing flows impose significant restriction in their network architecture for easy computation of the Jacobian. However, for low-dimensional BSDF sampling, the Jacobian computation is not the bottleneck. Therefore, we propose to use diffusion models to importance sample full BSDFs. Our method has two variants, one for most reflective materials that learns a distribution on a disk, and the other for extremely specular reflective materials and full BSDFs, which learns a distribution on a sphere. Our equal-time evaluations show that our method outperforms normalizing flows and significantly surpasses them in certain specular materials.

CCS Concepts: • **Computing methodologies** → **Ray tracing**.

Additional Key Words and Phrases: Rendering, importance sampling, neural sampling, diffusion models

ACM Reference Format:
Ziyang Fu, Yash Belhe, Haolin Lu, Liwen Wu, Bing Xu, and Tzu-Mao Li. 2024. BSDF importance sampling using a diffusion model. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24), December 03–06, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3680528.3687684

Authors' Contact Information: Ziyang Fu, University of California San Diego, United States of America, zifu@ucsd.edu; Yash Belhe, University of California San Diego, United States of America, ybelhe@ucsd.edu; Haolin Lu, University of California San Diego, United States of America, hal128@ucsd.edu; Liwen Wu, University of California San Diego, United States of America, liw026@ucsd.edu; Bing Xu, University of California San Diego, United States of America, b4xu@ucsd.edu; Tzu-Mao Li, University of California San Diego, United States of America, tzli@ucsd.edu.

SA Conference Papers '24, December 03–06, 2024, Tokyo, Japan

ACM ISBN 979-8-4007-1131-2/24/12
https://doi.org/10.1145/3680528.3687684

1 Introduction

In physically-based rendering, while analytical BSDFs (Bidirectional Scattering Distribution Functions) [Cook and Torrance 1982; Walter et al. 2007] often come with efficient sampling routines, they usually do not capture the nuanced behaviors of complex real-world materials [Dupuy and Jakob 2018; Matusik et al. 2003]. On the other hand, the tabular, measured BSDFs [Dupuy and Jakob 2018; Jakob et al. 2014] can be inefficient in terms of storage, consuming substantial GPU memory during rendering. Neural networks emerged as an appealing solution to compress BSDFs [Bi et al. 2020; Fan et al. 2022a,b; Guo et al. 2023; Sztrajman et al. 2021]. However, most neural BSDFs do not come with an efficient sampling routine for use in physically-based renderers. In this work, we demonstrate that deterministic diffusion models [Song et al. 2021] can be a useful option for importance sampling BSDFs.

Our work builds on the recent NeuSample [Xu et al. 2023] work that shares the same motivation. Xu et al. investigated several general purpose importance sampling strategies that can be used for reflective Neural BSDFs. Among these, one option is to use a neural network to predict a mixture of Gaussian lobes (Multilobe Mixtures in Figure 1), and another option is to use normalizing flows [Kobyzev et al. 2020] (Normalizing Flows in Figure 1). While normalizing flows are considered the most accurate sampling methods by Xu et al., the flows usually put significant constraints on the network architectures so that the Jacobian can be easily computed for probability density evaluation. Furthermore, existing methods focus on reflection and do not handle transmissive BSDFs.

Our key observation is that in a low-dimensional sampling problem like BSDF importance sampling, the Jacobian computation is usually *not* a bottleneck, unlike machine learning applications. Inspired by the recent progress in machine learning, we propose to use a deterministic diffusion model to map a base distribution to sampling directions. The deterministic diffusion model acts as an ordinary differential equation (ODE) integrator that performs the sampling transformation. This formulation results in a transformation that is expressive while also ensuring it is differentiable and bijective. Furthermore, recent work on distilling diffusion models [Liu et al. 2023] allows us to fit a faster ODE integrator that reaches the target with much fewer steps.

In this paper, we discuss the application of modern deterministic diffusion models for BSDF sampling. This requires addressing a few technical challenges, including defining a base distribution, training data generation, the output domain of the diffusion model and dealing with discontinuities and periodicity in the domains. We also show a real-time implementation that can apply the diffusion model BSDF sampling for 1024×1024 images at 60 frame per second for 4 samples per pixel on an RTX 4090.

Our main contributions are:

(1) Compared to previous methods using normalizing flows, we demonstrate that even when network size is small, the diffusion model exhibits greater expressive power at the same computational speed and achieves more accurate learning of complex distributions.
(2) By extending the learning of the diffusion model to a unit circle, we take into account the periodicity of the azimuth in spherical coordinates, achieving high-quality, full BSDF sampling for both reflection and transmission (Figure 1).
(3) We evaluate the advantages and disadvantages of sampling within the projected hemisphere domain and spherical domain, designing specialized methods for each.

Based on our evaluation results, when the BSDF material is diffuse or not extremely specular, we recommend sampling in the projected hemisphere domain using fewer sampling steps. When the BSDF material is a smooth mirror or metallic surface, we recommend sampling in the spherical domain to better capture grazing angles, while using more sampling steps to reduce fireflies.

2 Related Work

BSDF representation and compression. Due to the high memory footprint required by tabulated BSDF measurements [Matusik et al. 2003; Ngan et al. 2005], different models are fitted as a way of compression. Analytical BSDF models have been dominant in production rendering [Burley 2012; Cook and Torrance 1982; He et al. 1991; Heitz et al. 2015; Walter et al. 2007]. However, the extremely wide variety of surface properties, such as highly specular highlights, anisotropic features, layered structures, and iridiscence, make it difficult to propose a general model. Moreover, it is especially challenging to handle specific measurement angles, such as grazing angles. Neural representations for materials have lately emerged as a promising direction, offering more flexibility while achieving compactness. Among them, Sztrajman et al. [2021] encoded reflective BSDFs using a lightweight neural network, and Fan et al. [2022b] further incorporated layering operators using neural networks. Zheng et al. [2021] used neural processes to compactly represent reflective BSDFs while applying classifiers to tweak attributes and achieve certain editability. We demonstrate our sampling method on the RGL material dataset [Dupuy and Jakob 2018] which contains a wide variety of measure materials.

BSDF Importance sampling. Neural representations of BSDFs have recently gained popularity. In comparison, importance sampling these neural representations is less explored, which hinders their application in production. Earlier work importance sampled measured BSDFs by applying tabular solutions [Lawrence et al. 2004] but it requires large amounts of storage. Many previous works fitted parametric analytical models [Sun et al. 2018], which by construction provided closed-form importance sampling solutions. Sztrajman et al. [2021] fitted a parametric Blinn-Phong model to estimate the density for their learned neural encodings. Fan et al. [2022b] learned to fit a proxy distribution composed of one isotropic Gaussian lobe and one Lambertian lobe. Xu et al. [2023] proposed several importance samplers and achieved state-of-the-art performance for importance sampling neural materials. We take their work as the main baseline. Their histogram mixture method relies on training under direct supervision of a ground-truth probability density; therefore, we compare our method with their multi-lobe mixtures and normalizing flow methods. We further cover transmittance and are not limited to reflective BSDFs as in previous works.

Normalizing flows. [Rezende and Mohamed 2015] are popular models for density estimation and sampling (see surveys [Kobyzev

et al. 2020; Papamakarios et al. 2021] for an overview). The models specify a series of bijective transformations T to map a (usually simple) base distribution $p_z(z)$ to the target distribution $p_x(x;\theta)$ such that $x = T(z)$, where $z \sim p_z(z)$. The key property is that the transformation T must be invertible and both T and T^{-1} must be differentiable. Moreover, the Jacobian of T is triangular form to enable linear computational cost. Normalizing flows have been applied to various rendering problems [Müller et al. 2019; Zheng and Zwicker 2019]. Xu et al. [2023] used a lighter-weight flow to model the probability distribution of lighting directions for spatially varying neural materials, conditioning on UV coordinates and viewing directions. Their work did not handle transmission due to the projected hemisphere representation. Our work extends the domain to the sphere and makes it possible to generalize to materials with transmittance. More importantly, we show that diffusion models allow us to construct more expressive mappings for sampling under the same computation budget.

Diffusion model. Diffusion models have emerged as powerful deep generative models [Yang et al. 2023]. The initial diffusion model [Ho et al. 2020; Sohl-Dickstein et al. 2015] are probabilistic generative models that degrade data by injecting noise, then learn to reverse this process through Stochastic Differential Equations (SDEs). Later, Song et al. [2021] demonstrated the existence of an ordinary differential equation (ODE), also named the probability flow ODE, whose trajectories share the same marginal distributions as those of the reverse-time SDE. Unlike SDE, ODE solvers follow deterministic trajectories unaffected by stochastic fluctuations, typically converging much faster than stochastic counterparts, though with slighting lower sample quality [Yang et al. 2023]. A large body of works [Albergo et al. 2023; Heitz et al. 2023; Lipman et al. 2023; Liu 2022; Liu et al. 2023] on faster diffusion samplers are based on solving the probability flow ODE by constructing interpolations between two distributions. However, most of these applications operate in high-dimensional spaces and require large and complex networks, focusing solely on sampling without exact likelihood calculation. Our work demonstrates that for lower dimensional tasks requiring exact PDF values, the deterministic diffusion model provides better and more stable results compared to normalizing flows, using small network size, which is suitable for BSDF sampling.

3 preliminaries

BSDF Importance Sampling. Given a BSDF $f(\boldsymbol{\omega_o}, \boldsymbol{\omega_i})$, importance sampling aims to construct a probability distribution $p(\boldsymbol{\omega_o}|\boldsymbol{\omega_i}) \propto f(\boldsymbol{\omega_o}, \boldsymbol{\omega_i})$, where $\boldsymbol{\omega_o}$ and $\boldsymbol{\omega_i}$ are the outgoing and incoming direction respectively. To use p for importance sampling, we need to be able to draw samples from it and evaluate it for arbitrary incoming and outgoing directions.

Deterministic diffusion models. We model p using a linear interpolation based deterministic diffusion model [Heitz et al. 2023; Liu et al. 2023]. These methods transform samples from a base distribution $x_0 \sim p_0$ to a target distribution $x_1 \sim p_1$ through an ODE

$$dx_t = F(x_t, t)dt, \tag{1}$$

Here, F is a continuous function that determines how samples x_t are infinitesimally transformed via the diffusion process, and $x_t = tx_1 + (1-t)x_0$. The solution to this ODE [Heitz et al. 2023; Liu et al. 2023] is given by

$$F(x_t, t) = \mathbb{E}\left[x_1 - x_0 \mid x_t, t\right]. \tag{2}$$

The diffusion model $D_\theta(x_t, t, \boldsymbol{\omega_i})$ is trained to learn F. Once trained, D_θ can be used in place of F to transform samples from p_0 to p_1 by numerically integrating Equation (1).

In Section 1 of the supplementary material, we show an alternate derivation for the expression above by modeling the distributions at intermediate timesteps t as a convolution of the base p_0 density and target density p_1. The convolution perspective sheds light on an important property of diffusion models: while in theory they can model discontinuous distributions, but in practice, since we do not explicitly integrate the Dirac deltas at the boundary, it *will never* converge to the final distribution. In BSDF sampling, they struggle to handle discontinuities at domain boundaries, which is problematic near grazing angles–we fix this by changing the domain over which the BSDF is defined.

ODE integration and Reflow. To draw a sample $x_1 \sim p_1$, we start with a sample $x_0 \sim p_0$ and perform Euler integration which evolves samples as $x_{t+\Delta_t} = x_t + D_\theta(x_t, t, \boldsymbol{\omega_i})\Delta_t$, where Δ_t is the step size. This typically takes hundreds or thousands of integration steps, which can be a bottleneck for real-time applications. We deal with this by applying Reflow [Liu et al. 2023], a recent advance to accelerate diffusion model sampling. Reflow straightens the ODE trajectories *after training* D_θ without modifying the marginals p_0 and p_1; this is helpful since straighter trajectories require fewer integration steps. Practically, when applied to BSDF importance sampling, we have found that this reduces the integration steps from hundreds to just few steps, with minimal quality loss.

Deterministic diffusion and bijective mappings. Deterministic diffusion models form bijections between the source and target distributions [Liu et al. 2023; Song et al. 2021]. This property allows us to both draw samples from the target distribution, and importantly, evaluate the probability density of the sampling process by computing its Jacobian. Since BSDFs are low-dimensional, calculating the Jacobian of D_θ is not a computational bottleneck, enabling us to use diffusion models for efficient importance sampling.

4 Our Method

We first introduce our model architecture, its training, sampling, PDF evaluation and distillation using Reflow (Section 4.1). Next, we describe how our method handles importance sampling of reflective BSDFs (or Bidirectional Reflection Distribution Functions, BRDFs) by learning a distribution on a disk domain (Section 4.2). This works well for non-specular BSDFs, however, due to discontinuities as well as extremely high BSDF value at the boundary, distributions on a disk can cause problems at grazing angles for specular BSDFs. To further support specular BSDFs, we learn distributions on a sphere (Section 4.3); the change of domain removes boundary discontinuities and reduces the BSDF values at grazing angles via the cosine term, and also extends our sampling to full BSDFs with both reflectance and transmission.

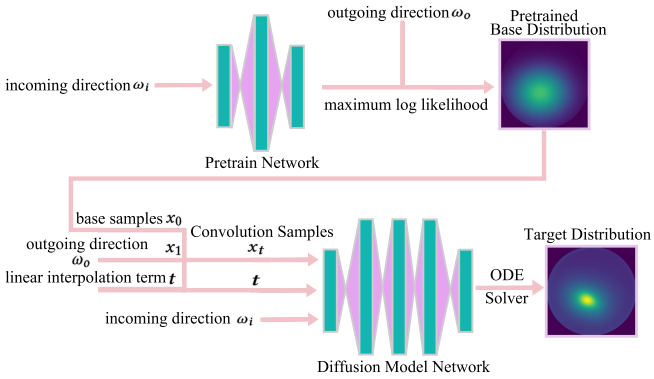


Fig. 2. Our diffusion model architecture for BSDF sampling. $\boldsymbol{\omega_o}$ and $\boldsymbol{\omega_i}$ are samples drawn from the BSDF $f(\boldsymbol{\omega_o}, \boldsymbol{\omega_i})$, and $\boldsymbol{t}$ is the linear interpolation term which follows the uniform distribution over the interval [0, 1].

4.1 Architecture, Training, Inference, and Distillation

We first discuss our architecture (Figure 2). Next, we specify the diffusion model loss function used for training. We then discuss the generation of the training data given a BSDF f, using a Markov chain Monte Carlo sampler [Foreman-Mackey et al. 2013]. Subsequently, we discuss sampling and probability density function (PDF) evaluation for both Monte Carlo integration and multiple importance sampling [Veach and Guibas 1995]. Finally, we discuss how we apply Reflow [Liu et al. 2023] to distill the model to be orders of magnitude faster, without losing much quality.

The overall architecture (Figure 2) of our model remains consistent whether applied to a BRDF or a full BSDF. Depending on the sampling domain, some variations in the inputs and the loss functions are used which are detailed in the Section 4.3.

Architecture and training. A diffusion model learns a mapping between two distributions. The base distribution should be a distribution easy to sample from and compute the PDF. Since the corresponding BSDF distributions can vary significantly in shape and variance, a simple fixed base distribution like a Gaussian with zero mean and unit variance is not an optimal choice.

We improve upon this by first training a small *pretrain* network (Figure 2, top) to capture the rough shape of the BSDF distribution by predicting the parameters for a parametric base distribution based on the input incident direction $\boldsymbol{\omega_i}$. For example, if a Gaussian distribution is used as the base distribution, then the network predicts the mean and covariance of the Gaussian. We specify the base distributions we use in the following subsections. We train this network by maximizing the log likelihood.

Next, the diffusion model maps the generated base distribution to our target (Figure 2, bottom). The model network D_θ is used to predict the slope F in Equation (2). Recall that the slope F is defined as an expectation. For a given incoming direction $\boldsymbol{\omega_i}$, we can compute this expectation by generating a large number of samples from three distributions, $x_0 \sim p_0, \boldsymbol{\omega_o} \sim p(\boldsymbol{\omega_o} \mid \boldsymbol{\omega_i})$, and $t \sim U(0,1)$, where p_0 is the base distribution obtained from the pretrain network, and $U(0,1)$ is the uniform distribution on $[0,1]$. We generate the samples x_t that follow the convolution distribution p_t using the relation $x_t = (1-t)x_0 + tx_1$, where $x_1 = \boldsymbol{\omega_o}$.

Our diffusion model D_θ takes x_t, t, and the incoming direction $\boldsymbol{\omega_i}$ for condition as inputs, and outputs the prediction for the ODE slope F. Following the approach by Heitz et al. [2023], the model can be trained by the loss:

$$\text{loss} = \|D_\theta(x_t, t, \boldsymbol{\omega_i}) - (x_1 - x_0)\|^2 \tag{3}$$

Training data generation through Markov chain Monte Carlo sampling. To generate the samples for training, previous methods [Xu et al. 2023] first randomly select values for $\boldsymbol{\omega_i}$, and then use them to sample online from the BSDF distribution $p(\boldsymbol{\omega_o} \mid \boldsymbol{\omega_i})$ for each training batch. This process, while feasible, is slow since it requires building a high-resolution histogram and perform inverse CDF sampling each time. Moreover, as we train multiple networks, reusing the generated samples would be ideal.

We propose a faster method for sampling $(\boldsymbol{\omega_o}, \boldsymbol{\omega_i})$ from precomputed sample pairs, which are generated through an MCMC sampler [Foreman-Mackey et al. 2013].

When the previous method randomly selected $\boldsymbol{\omega_i}$, it has inherently defined a distribution $p(\boldsymbol{\omega_i})$ based on the random distribution it used. The sample pairs $(\boldsymbol{\omega_o}, \boldsymbol{\omega_i})$ generated by the online sampling method will follow the joint distribution of $\boldsymbol{\omega_o}$ and $\boldsymbol{\omega_i}$, given that $p(\boldsymbol{\omega_o}, \boldsymbol{\omega_i}) = p(\boldsymbol{\omega_o} \mid \boldsymbol{\omega_i})\, p(\boldsymbol{\omega_i})$.

We can use an MCMC sampler to sample from the 4D joint distribution $p(\boldsymbol{\omega_o}, \boldsymbol{\omega_i})$, generating the sample pairs. We then draw samples for each training iteration from these pairs. To achieve this, we treat the BSDF $f(\boldsymbol{\omega_o}, \boldsymbol{\omega_i})$ as a 4D unnormalized joint distribution, proportional to $p(\boldsymbol{\omega_o}, \boldsymbol{\omega_i})$. We set $p(\boldsymbol{\omega_i})$ to be uniform. During training, we sample from these precomputed pairs by randomly selecting the index of each pair.

We replace the previous complex process [Xu et al. 2023] of building histograms and inverse CDF sampling with an offline MCMC sampling. During training, we require only random numbers generation each time, and the samples can be reused for all networks, significantly accelerating our model's training.

Sampling. Once we have the model D_θ, sampling is done in two steps. First, we generate the x_0 samples using the pretrain network given incoming direction $\boldsymbol{\omega_i}$. Then, we simulate the ODEs using Euler integration with a constant step size of $\Delta_t = 1/N$ for N steps. Specifically, we compute:

$$\hat{x}_{t+1/N} = \hat{x}_t + D_\theta(\hat{x}_t, t, \boldsymbol{\omega_i})/N \tag{4}$$

where $t \in \{0, \ldots, N-1\}/N$, with x_0 samples drawn from the pretrain base distribution p_0. The output x_1 from the ODE solver follow the distribution $p(\boldsymbol{\omega_o} \mid \boldsymbol{\omega_i})$.

PDF evaluation. As the ODE values are predicted using a neural network, their derivatives with respect to the input x_t are easily obtainable through automatic differentiation. Each discrete step in the ODE can be regarded as an invertible transform. Hence, we can compute the PDF by computing the Jacobian determinant through network gradient for each step, without an extra marginalization process. We accumulate Jacobian computed through the forward ODE trajectory and then multiply it by the PDF p_0 of the base samples x_0 to get the exact PDF value.

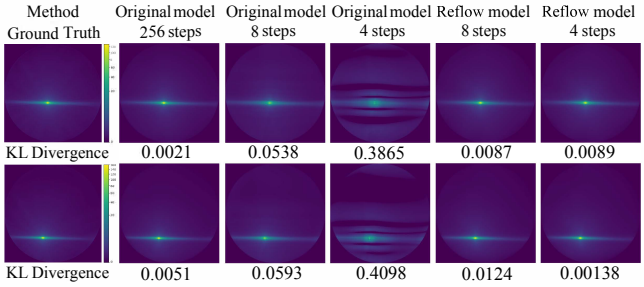


Fig. 3. Comparison on METAL-PAPER-COPPER material between orginal model and Reflow model with the same ODE sampling steps. The Reflow model, with fewer ODE steps, is significantly faster than the original model while accurately capturing the distribution.

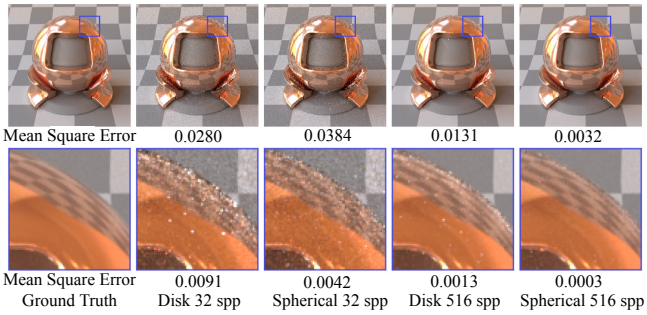


Fig. 4. Rendering results of COPPER-SHEET, an anisotropic conductor material, only using our diffusion model BSDF sampling methods in the unit disk domain and the spherical domain with 32 samples per pixel (spp) and 516 spp. Disk domain sampling suffers from fireflies at the grazing angles.

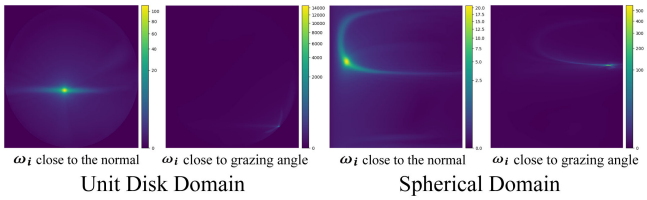


Fig. 5. PDF slices for a conductor material, comparing scenarios when the incoming direction $\boldsymbol{\omega_i}$ is near the normal (away from the grazing angle) and close to the grazing angle. In the unit disk domain, the PDF performs well outside the boundary but reaches extremely high values near the disk boundary (see the color bar on the right). In contrast, within the spherical domain, all PDF values remain within a reasonable range; however, the distribution's shape becomes considerably more complex.

To support multiple importance sampling [Veach and Guibas 1995], it is necessary to revert the trajectory from the BSDF $p(\boldsymbol{\omega_o} \mid \boldsymbol{\omega_i})$ back to the base p_0. We can obtain the reverse trajectory by integrating the ODE from $t = 1$ to $t = 0$. For specific PDF calculation formulae, please refer to our supplementary materials.

Fast and Accurate Reflow. Reflow [Liu et al. 2023] needs samples generated using a pretrained diffusion model for training. This involves performing hundreds or thousands of Euler steps/network evaluations to ensure near-convergence. Reflow then uses the samples generated from the diffusion model to straighten the ODE trajectory, ultimately reducing the Euler steps for solving the ODE to a single digit. We use TINYCUDANN [Müller 2021] to accelerate evaluation. While TINYCUDANN restricts precision to float16, this is acceptable since we mostly do basic addition, and we only generate samples without PDF calculation.

By reusing the samples generated from MCMC, we can quickly train several models. Consequently, we train two diffusion models on a single pretrained network: a small network for the base weights used for further training and a large network to better capture the distribution. The large network is used solely for network inference, generating ODE value predictions for online sampling. The small network uses samples generated from large network for the Reflow process. Figure 3 demonstrates the effectiveness of the Reflow method, significantly reducing the required sampling steps while achieving accurate distributions.

4.2 BRDF on Projected Hemisphere

We first consider sampling for a BRDF without transmission. Our goal is to sample proportional to the product of the BRDF defined on the unit hemisphere $\mathcal{H}$ and the cosine foreshortening term.

Typically, we transform $\boldsymbol{\omega} \in \mathcal{H}$ to another domain that is more convenient for sampling. An option is to project the unit hemisphere $\mathcal{H}$ onto the unit disk $\mathcal{H}_\perp$. This requires multiplying by the Jacobian, the inverse of the cosine term. This is equivalent to sampling the projection of the BRDF (without cosine) onto the unit disk $\mathcal{H}_\perp$.

To fit BRDF distributions of unit-disk projections $\boldsymbol{\omega}_\perp \in \mathcal{H}_\perp$, we select a 2D Gaussian distribution as our base distribution p_0. Our pretrain network outputs a 2D mean location and a 2D standard deviation for the two axes.

For most diffuse and (not extremely) specular materials, our experiments in Figure 11 show that learning on the unit disk domain is sufficient for BRDF sampling.

Problems with boundary discontinuities. We find that this approach fails near the boundary of the unit disk, making it difficult to accurately learn the BRDF at grazing angles. As we approach the grazing angle for $\boldsymbol{\omega_i}$, the energy over $\boldsymbol{\omega_o}$ tends to be concentrated at the boundary of the disk, see Figure 5. For materials like mirrors or conductors with very low roughness the grazing angle contains a significant amount of information. Thus, imperfect fitting leads to fireflies, see Figure 4. For a perfect fit, the diffusion model would need to model a discontinuous distribution that sharply transitions from the boundary values to zero at the disks' boundary, which it cannot do, see Section 3. In practice, it approximates the discontinuous jump a smooth transition, see the supplementary document for more details.

4.3 BSDF on Spherical Domain

In the previous subsection, we saw that learning distributions on a disk causes failures near the grazing angle. We fix these by learning a distribution on a sphere instead of a disk; this also extends our method's capability to handle full BSDF sampling including reflection and transmission.

We denote the differential solid angle in spherical coordinates as $d\omega = |\sin\theta|\, d\theta d\phi$, where ω is the solid angle, and $\theta \in [0, \pi], \phi \in [-\pi, \pi]$ are the elevation and azimuthal angles respectively. Thus, importance sampling in the spherical domain can be written as

$$p(\boldsymbol{\omega_o} \mid \boldsymbol{\omega_i}) \propto f(\boldsymbol{\omega_o}, \boldsymbol{\omega_i}) \cdot |\cos\theta| \cdot |\sin\theta|\,.$$

The inclusion of both $\cos\theta$ and $\sin\theta$ ensures that the PDF value of distribution is 0 at both normal incidence $\theta = 0$ (as well as $\theta = \pi$ for BSDF on whole sphere) and at grazing angles $\theta = \pi/2$ (for BRDF on hemisphere), which satisfies the convolution distribution property. The latter also ensures that extremely high values near grazing angles are removed since $\cos\theta$ approaches zero.

The azimuthal angle requires special care; as we approach its boundary, near $\phi = -\pi$ and $\phi = \pi$, p is non-zero. However, it is periodic, with a period of 2π. We exploit this by joining the two endpoints of the domain, transforming it into a unit circle, and learn p directly on this domain, thereby making the distribution continuous on ϕ.

Let $p(\phi)$ denote the marginal distribution of ϕ. It satisfies $p(\phi) = p(\phi + 2\pi)$ and is normalized $\int_{-\pi}^{\pi} p(\phi)d\phi = 1$. These make it a wrapped probability distribution on a unit 1-sphere. Its CDF, denoted as $c(z)$, is given by $c(z) = \int_{-\pi}^{z} p(\phi)d\phi$, where $z = (z \mod 2\pi) - \pi$. Consequently, since the definitions of PDF and CDF remain conceptually unchanged in the range $[-\pi, \pi]$, our derivations in supplementary materials for Equation 2 still hold within the periodic domain, and the ODE for ϕ is given by

$$\frac{dx_t}{dt} = \mathbb{E}\left[x_1 - x_0 \mid x_t, t\right],$$

where $x_t = (1-t)x_0 + t\phi$, and x_0 is sampled from a base distribution p_0, which also needs to be a wrapped probability distribution due to the constraints of the unit circle. Importantly, the term $x_1 - x_0$ is not measured using Euclidean distance, but rather using geodesic distance on the unit circle. The geodesic distance is straightforward to compute on a unit circle, and it is given by $d = (x_1 - x_0) \mod 2\pi - \pi$, which ensures $d \in [-\pi, \pi]$.

The architecture used for training is consistent with the one presented in Figure 2 albeit with a few modifications. For the pretrained base distribution, we opt for a combination of two independent 1D distributions. For the elevation θ, we use a Gaussian distribution. For the azimuth ϕ, due to its periodic nature, we use the von Mises distribution (also known as the circular normal distribution); for it the pretrained network predicts its mean μ and its concentration $\kappa \in [0, +\infty]$, which is analogous to the variance in a Gaussian.

When training the diffusion model, we modify the input to ensure that ϕ remains constrained on the unit 1-sphere. This is achieved through a simple positional encoding, setting $\phi_{input} = (\sin\phi, \cos\phi)$. Additionally, the loss function computes the geodesic distance for ϕ, and the treatment of θ remains unchanged.

5 Results

We implemented our method in PyTorch [Paszke et al. 2019] and integrated it into Mitsuba 3 [Jakob et al. 2022]. We also implemented a real-time megakernel path tracer with inline neural network inference in a custom renderer using the Vulkan API with hardware-accelerated ray tracing. Our training code is written in PyTorch and we use TINYCUDANN [Müller 2021] to accelerate training during Reflow. Automatic differentiation was used for Jacobian computation in PyTorch, while it was manually implemented in Vulkan.

Our model's training process is designed to be independent of the BSDF's nature, whether measured or as a neural representation. It solely requires the BSDF values, along with incident and outgoing directions, to directly learn the distribution.

Architecture details. For the incoming direction $\boldsymbol{\omega_i}$, which serves as the conditional vector, we apply positional encoding before inputting it to the network. For the pretraining network, we use the same network size for both the disk and spherical domains: a very small multilayer perceptron with one hidden layer containing 16 neurons. For the actual diffusion model network, we observed that learning in the spherical domain is more challenging than in the disk domain. Consequently, we use an MLP with 3 hidden layers and 32 neurons for the disk domain, and an MLP with 4 hidden layers and 32 neurons for the spherical domain.

Baselines. For the method using neural network sampling, we compare our method with approaches presented in NBRDF [Sztrajman et al. 2021] and NeuSample [Xu et al. 2023]. NBRDF fits a parametric Blinn-Phong model, which is isotropic. The first approach in NeuSample employs a combination of a Lambertian lobe and Gaussian mixtures to predict the target distribution. Network outputs the parameters for each Gaussian and the weights for each lobe. The second approach uses Normalizing Flows to map between a base distribution and the target distribution. This architecture is similar to ours, as it also involves pretraining the base distribution and then performing normalizing flow sampling. However, their implementation is limited to the unit disk domain and supports only BRDF. For comparison, we naïvely extended their output domain from the disk to the spherical domain without accounting for the periodicity of ϕ. Additionally, we compared our method with the sampling procedure provided by the RGL dataset [Dupuy and Jakob 2018], which uses a specially designed compressed lookup table.

Materials for BSDF sampling. For BRDF sampling, we evaluate on the RGL dataset [Dupuy and Jakob 2018] which includes a diverse collection of complex, real-world BRDFs. This dataset presents particularly challenging materials, especially at grazing angles. Notably, certain neural compressed BRDFs for RGL dataset [Sztrajman et al. 2021], demonstrate significant difficulties in accurately simulating materials like COPPER-SHEET at grazing angles, often failing entirely. To emphasize our model's enhanced ability to effectively handle grazing angles, and only for better comparison purposes, we train directly on these complex original BRDFs. This does not imply that our model is limited to measured BRDFs. On the contrary, by addressing situations more complex than neural BRDFs, we highlight our model's superior capability to accurately sample grazing angles and complex BRDFs, as Figure 9 shows.

For BSDFs, due to the lack of a measured BSDF dataset or neural BSDF work with much more complex distributions, and the absence of specifically designed sampling methods for BSDFs, we only test on rough dielectric [Walter et al. 2007] and Disney BSDFs [Burley 2015]. Figure 6 compares our method with the analytical sampling solution. We demonstrate our capability for accurate full BSDF sampling, and

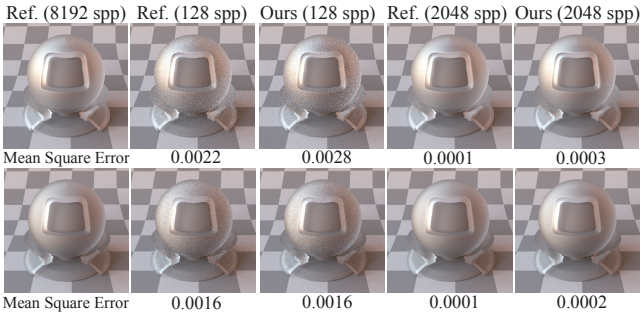


Fig. 6. The rendering results of two rough dielectric materials [Walter et al. 2007] with roughnesses 0.3 (top) and 0.5 (bottom) using our diffusion model with 8 sampling steps. We compare our method with the analytical sampling of rough dielectric in Mitsuba 3 (Ref.) and the results are comparable.

Table 1. Running times including generating samples and calculating their PDF of NeuSample and our method (4 sampling steps) on disk and spherical domains. The NeuSample methods are implemented purely using PyTorch. This table presents the sampling times (ms) for generating 1024x1024 resolution images with 1 sample per pixel.

Time (ms)	NeuSamp. Mixt. lobes	NeuSamp. Norm. Flows	Ours Torch Disk	Ours Torch Spher.	Ours Vulkan Disk	Ours Vulkan Spher.
Running	13.89	52.66	52.08	72.57	4.30	5.77

shows the potential for sampling more complex BSDFs without perfect analytical solution.

Rendering results and quantitative metrics for diverse materials (synthetic and real-world captured) under various lighting conditions (point and environmental) are presented in the supplementary materials. We refer readers to the supplementary materials for a comprehensive evaluation.

Time statistics. Table 1 shows the sampling time using our method and NeuSample on the same pytorch platform and our method solely on Vulkan using a single RTX 4090 GPU. Table 2 further shows the actual rendering FPS using Vulkan. Notably, our real-time implementation has achieved remarkable performance improvements. This makes it a viable option for real-time rendering of neural BSDF sampling. Training times per material averaged 3 hours for disk domain and 3.5 hours for spherical domain. Detailed timings for each step are provided in the supplementary material.

Better expressiveness and robustness. Compared to the two neural sampling methods in NeuSample, our model demonstrates greater robustness and expressiveness for most materials. Figure 11(a) shows the comparison between our method and the NeuSample baselines.

Our model is more flexible. Unlike mixture lobes and normalizing flows, having their expressive capabilities constrained by predefined parameters, our model does not impose such limitations during training. Expressiveness is adjustable by varying sampling steps.

In Figure 9, we selected four materials for detailed demonstration. Previous neural sampling methods struggled with these kinds of specular and anisotropic materials, but our approach shows significant improvement in these cases. Overall, the disk domain has the lowest Mean Square Error (MSE), indicating its easier-to-learn characteristics. However, at grazing angles, the spherical domain performs better, exhibiting fewer fireflies and demonstrating its ability to handle discontinuities.

Table 2. Comparison of FPS (frames per second) and storage size between RGL Tabular and our method using RTX 3070 and RTX 4090 GPUs on anisotropic materials. RGL Tabular is faster on the RTX 4090 compared to 3070, likely due to the greatly increased L2 cache size of the 4090 (4090 has 72 MB L2 cache while 3070 only has 4 MB). On RTX 4090, we tested on 4 samples per pixel because 1 samples per pixel will be CPU-bounded.

FPS/Size	RGL Tabular	Ours Disk	Ours Spher.
RTX 3070 (1 spp)	44.2	52.7	45.0
RTX 4090 (4 spp)	122.7	76.5	64.3
Size (float nums.)	282880	3328	4256

Figure 8 demonstrates our method's effectiveness in scenes with diverse materials. Results indicate the superior stability and quality of our methods in complex material compositions. While NBRDF struggles with highly anisotropic materials and NeuSample shows limitations for specular materials, our approach exhibits robust performance across all material types. Furthermore, we showcase near-identical results to ground truth sampling for transmissive BSDFs, underscoring the accuracy of our method.

In Figure 11(a), we present the graph of samples per pixel versus log(MSE). Theoretically, when we can perfectly importance sample the distribution, the result should be a straight line. For diffuse materials, our model shows little difference from the NeuSample methods. However, as the material becomes more complex and specular, the noise from learning failures increases, causing the graph to deviate. The NeuSample method exhibits instability with specular materials. When the material becomes highly specular, our method significantly outperforms theirs.

Better compression. We also compare our method with the tabular approach using the RGL dataset. Figure 7 shows that our sampling results are comparable to the tabular method, while achieving significant compression in storage space.

Table 2 shows the frames per second of our Vulkan renderer and the compression comparison between our method and the RGL tabular sampling for anisotropic materials. We separately calculated the number of floats used for the RGL tabular sampling and compared it with our method. Our model achieves a compression ratio of 85× for disk and 66.5× for spherical. If considering that our renderer uses float16, the compression ratio would be even higher.

Spherical vs. Disk. Figure 4 shows that the unit disk sampling exhibits noticeable fireflies at grazing angles, whereas the spherical sampling method effectively mitigates these artifacts. The results of Figures 9 and 11 show that, due to the more complex distribution on a spherical domain, the MSE is higher compared to the disk domain. Our experiments in Figure 10 show that projecting onto the unit disk makes the learning easier except the part near the disk boundary, since the general KL Divergence is smaller. One intuitive explanation is that most BRDFs are centered around an outgoing

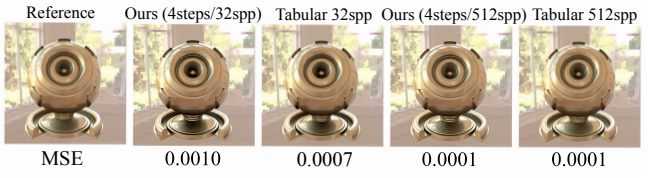


Fig. 7. Rendering results using our method (disk 4 steps) and RGL tabular methods under direct illumination with only BRDF sampling.

direction ω_o, with energy spreading relatively evenly around it. The results are more uniformly distributed in all directions compared to the spherical domain, where the curvature causes the distribution to take on a more complex shape, like Figure 5 shows.

Therefore, we recommend using the disk domain for most complex BRDFs. However, for learning BSDFs or materials in highly specular situations where one wants to avoid excessive fireflies, we suggest using the spherical domain.

6 CONCLUSION AND FUTURE WORK

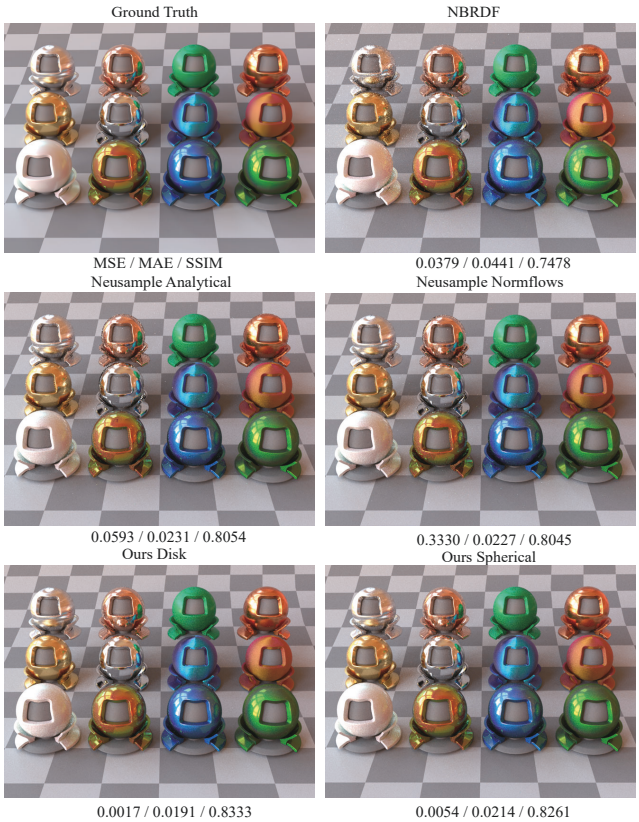
We propose a more expressive and robust method for BRDF importance sampling and extend our model to full BSDF by supporting diffusion model learning on the spherical domain.

Using the von Mises distribution in spherical domain can cause numerical issues when rendering highly specular materials like ANISO-MIRROR with float16 precision. Simpler and easily samplable base distributions on the unit sphere are worth to find.

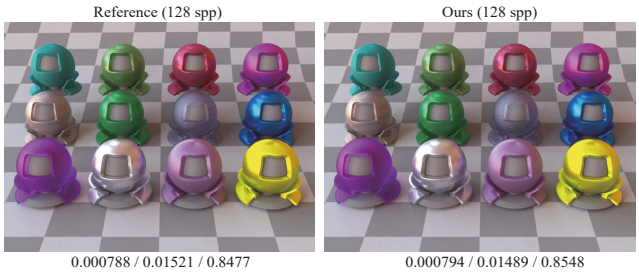
While our method performs robustly across a wide range of materials, it shows increased noise for near-perfect transmissive materials due to numerical instability. Extremely large PDF values in these cases degrade MCMC sample quality, and an efficient way to handle such corner cases are interesting to explore.

We only have tested and evaluated our model on 4D BSDFs; while theoretically straightforward, extending to **SVBRDF** is more challenging due to higher dimensions of conditions. Network capacity is the primary constraint; larger networks can effectively learn SVBRDFs, but rendering efficiency demands necessitate network compression, posing the main obstacle for SVBRDF extension. A possible approach involves constraining model flexibility by predefining sampling steps, learning complete diffusion model with larger network, storing specific steps using a more compact network.

Finally, we believe our method is not limited to BSDF sampling. Our approach may also provide insights for other importance sampling problems **in lower dimension**, such as path guiding, complex luminaire sampling, and portal sampling.



(a) Real-world captured materials scene results under environmental lighting using various BRDF sampling methods.



(b) Disney BSDFs scene results under environmental lighting using our BSDF sampling and analytical sampling methods.

Fig. 8. (a) Comparison of our method with NBRDF [Sztrajman et al. 2021], NeuSample [Xu et al. 2023], under the same number of samples per pixel for scenes with multiple materials. (b) Our method fitted to sample Disney BSDFs [Burley 2015], and compared to the analytical sampling routine. We measure numerical errors using Mean Square Error (MSE), Mean Absolute Error (MAE), and Structural Similarity Index Measure (SSIM)

Acknowledgments

This work was supported in part by NSF grants 2100237, 2120019, and 2238839, and gifts from Adobe and Google. We thank the anonymous reviewers for their feedback.

References

Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. 2023. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797* (2023).

Sai Bi, Zexiang Xu, Pratul Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. 2020. Neural reflectance fields for appearance acquisition. *arXiv preprint arXiv:2008.03824* (2020).

Brent Burley. 2012. Physically-based shading at Disney. In *SIGGRAPH Course Notes. Practical physically-based shading in film and game production.*, Vol. 2012. 1–7.

Brent Burley. 2015. Extending the Disney BRDF to a BSDF with integrated subsurface scattering. *SIGGRAPH Course: Physically Based Shading in Theory and Practice* 19, 7 (2015), 9.

R. L. Cook and K. E. Torrance. 1982. A Reflectance Model for Computer Graphics. *ACM Transactions on Graphics (TOG)* 1, 1 (1982), 7–24.

Jonathan Dupuy and Wenzel Jakob. 2018. An Adaptive Parameterization for Efficient Material Acquisition and Rendering. *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia)* 37, 6 (2018), 274:1–274:18.

Jiahui Fan, Beibei Wang, Milos Hasan, Jian Yang, and Ling-Qi Yan. 2022a. 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

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

Daniel Foreman-Mackey, David W Hogg, Dustin Lang, and Jonathan Goodman. 2013. emcee: the MCMC hammer. *Publications of the Astronomical Society of the Pacific* 125, 925 (2013), 306.

Jie Guo, Zeru Li, Xueyan He, Beibei Wang, Wenbin Li, Yanwen Guo, and Ling-Qi Yan. 2023. MetaLayer: A Meta-Learned BSDF Model for Layered Materials. *ACM Transactions on Graphics (TOG)* 42, 6 (2023), 1–15.

Xiao D. He, Kenneth E. Torrance, François X. Sillion, and Donald P. Greenberg. 1991. A Comprehensive Physical Model for Light Reflection. *SIGGRAPH Comput. Graph.* 25, 4 (1991), 175–186.

Eric Heitz, Laurent Belcour, and Thomas Chambon. 2023. Iterative α-(de) blending: A minimalist deterministic diffusion model. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–8.

Eric Heitz, Jonathan Dupuy, Cyril Crassin, and Carsten Dachsbacher. 2015. The SGGX microflake distribution. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 34, 4 (2015), 48.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

Wenzel Jakob, Miloš Hašan, Ling-Qi Yan, Jason Lawrence, Ravi Ramamoorthi, and Steve Marschner. 2014. Discrete stochastic microfacet models. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 33, 4 (2014), 1–10.

Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, and Delio Vicini. 2022. Dr. Jit: A just-in-time compiler for differentiable rendering. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 41, 4 (2022), 1–19.

Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. 2020. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence* 43, 11 (2020), 3964–3979.

Jason Lawrence, Szymon Rusinkiewicz, and Ravi Ramamoorthi. 2004. Efficient BRDF importance sampling using a factored representation. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 23, 3 (2004), 496–505.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. 2023. Flow Matching for Generative Modeling. In *International Conference on Learning Representations*.

Qiang Liu. 2022. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577* (2022).

Xingchao Liu, Chengyue Gong, and qiang liu. 2023. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *International Conference on Learning Representations*.

W. Matusik, H. Pfister, M. Brand, and L. McMillan. 2003. A Data-Driven Reflectance Model. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 22, 3 (2003), 759–769.

Thomas Müller. 2021. *tiny-cuda-nn*. https://github.com/NVlabs/tiny-cuda-nn

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.

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).

Danilo Rezende and Shakir Mohamed. 2015. Variational inference with normalizing flows. In *International conference on machine learning*. PMLR, 1530–1538.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*. PMLR, 2256–2265.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.

Tiancheng Sun, Henrik Wann Jensen, and Ravi Ramamoorthi. 2018. Connecting Measured BRDFs to Analytic BRDFs by Data-Driven Diffuse-Specular Separation. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 37, 6 (2018), 273.

Alejandro Sztrajman, Gilles Rainer, Tobias Ritschel, and Tim Weyrich. 2021. Neural BRDF representation and importance sampling. *Computer Graphics Forum* 40, 6 (2021), 332–346.

Eric Veach and Leonidas J. Guibas. 1995. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *SIGGRAPH*. 419–428.

Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. 2007. Microfacet models for refraction through rough surfaces. In *Rendering Techniques*. 195–206.

Bing Xu, Liwen Wu, Milos Hasan, Fujun Luan, Iliyan Georgiev, Zexiang Xu, and Ravi Ramamoorthi. 2023. NeuSample: Importance Sampling for Neural Materials. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–10.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. 2023. Diffusion models: A comprehensive survey of methods and applications. *Comput. Surveys* 56, 4 (2023), 1–39.

Chuankun 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.

Quan Zheng and Matthias Zwicker. 2019. Learning to importance sample in primary sample space. *Computer Graphics Forum (Proc. Eurographics)* 38, 2 (2019), 169–179.

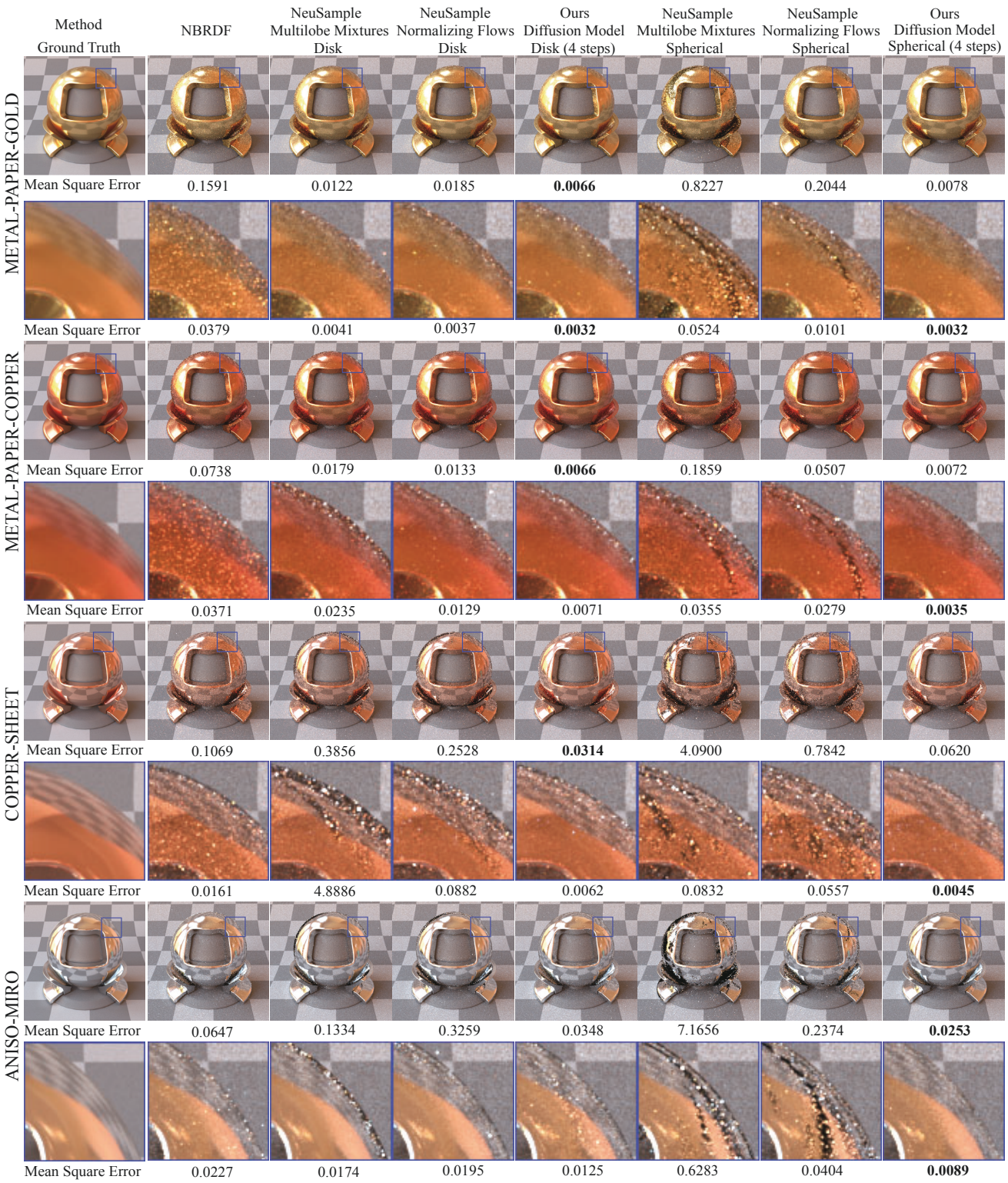


Fig. 9. An **equal time** Rendering results and **grazing angle slices** of four specular materials using various BRDF sampling methods. All renderings are performed using only BRDF sampling and global illumination. We set the time of our model on disk domain with 32 spp as baseline.

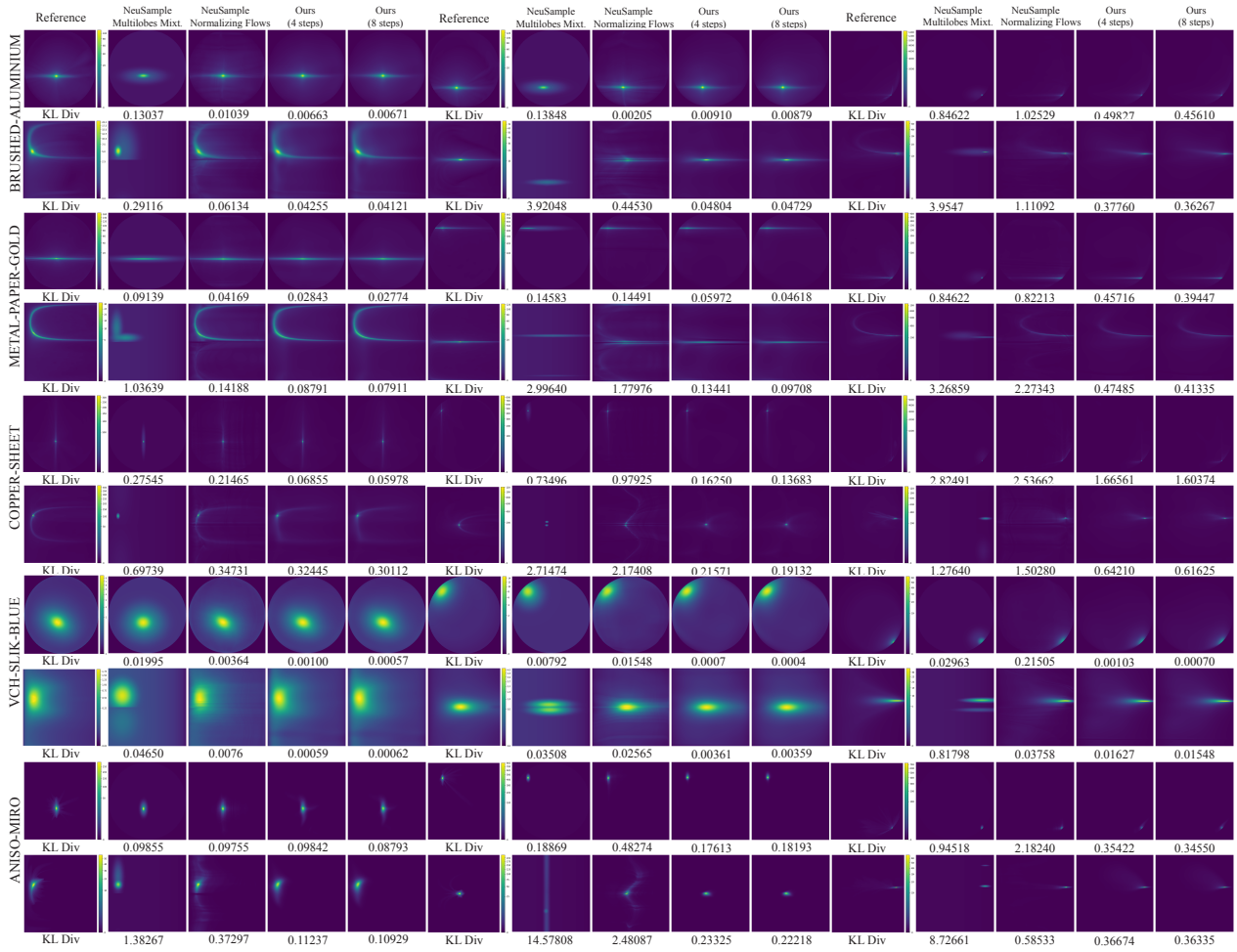
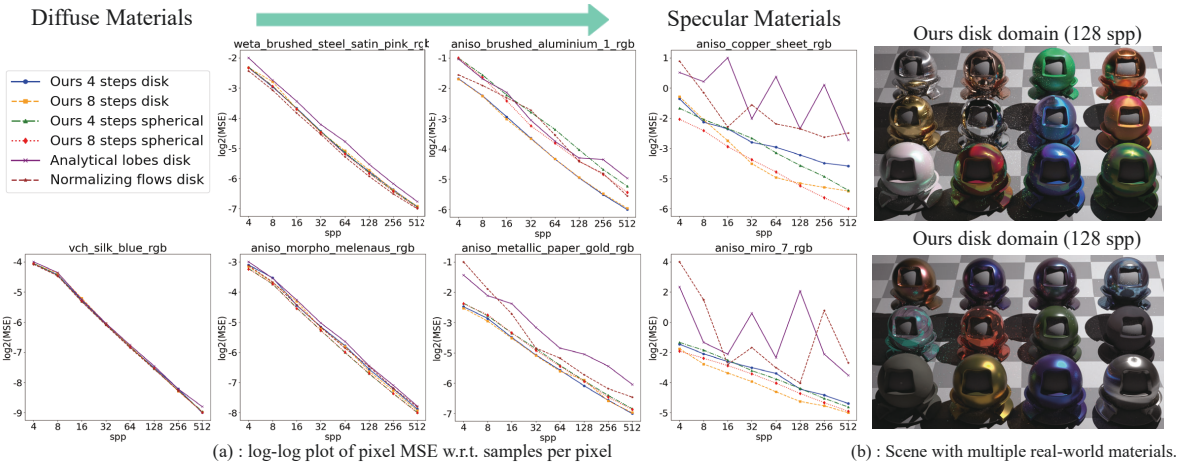


Fig. 10. PDF slices of five materials using various BRDF sampling methods. Each column represents the predicted PDF of one fixed outgoing direction, ranging from near the normal to near the grazing angle. For each material, the top row is the disk domain, and the bottom row is the spherical domain.



(a) : log-log plot of pixel MSE w.r.t. samples per pixel (b) : Scene with multiple real-world materials.

Fig. 11. (a) **Convergence graphs** of different types of materials, gradually transitioning from diffuse to specular. (b). Scenes with multiple real-world materials of our methods with same spp under point lighting.