Neural Product Importance Sampling via Warp Composition

JOEY LITALIEN*, McGill University, Canada
MILOŠ HAŠAN, Adobe Research, USA
FUJUN LUAN, Adobe Research, USA
KRISHNA MULLIA, Adobe Research, USA
ILIYAN GEORGIEV, Adobe Research, UK

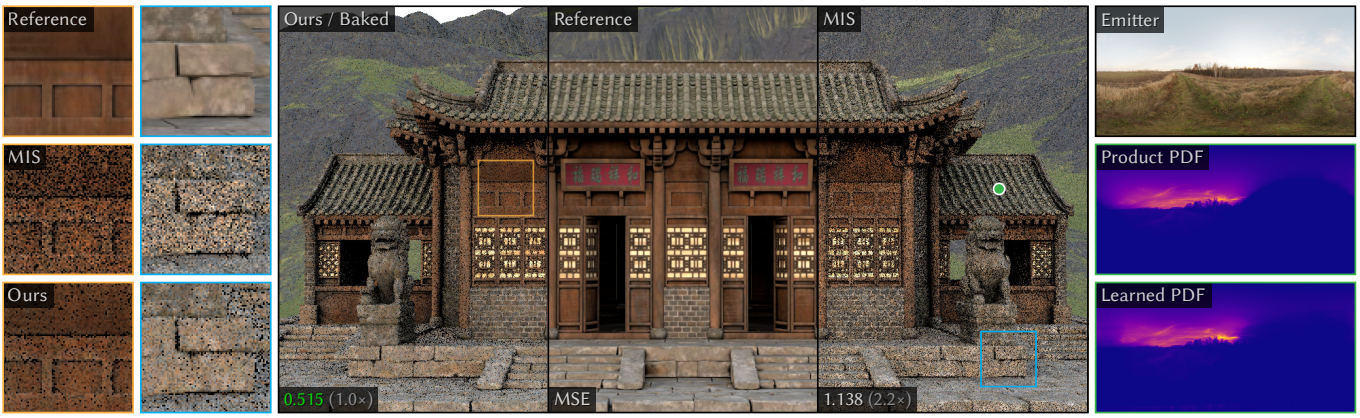


Fig. 1. Our method composes a neural spline flow *head warp* with an emitter *tail warp* to achieve approximate product importance sampling of environment lighting with other terms (cosine and BRDF). Applied to cosine-weighted environment sampling on the TEMPLE scene, we demonstrate significant variance reduction over multiple importance sampling (MIS) at equal rendering time (35 ms, 4 spp). We also visualize the conditional distribution learned by our model at the shading point marked in green. Our learned PDF closely matches the true (unshadowed) product. Our head warp does not have to learn the intricate details of the environment map already captured by the tail warp, and can be represented as a compact normalizing flow that can be baked for fast inference.

Achieving high efficiency in modern photorealistic rendering hinges on using Monte Carlo sampling distributions that closely approximate the illumination integral estimated for every pixel. Samples are typically generated from a set of simple distributions, each targeting a different factor in the integrand, which are combined via multiple importance sampling. The resulting mixture distribution can be far from the actual product of all factors, leading to sub-optimal variance even for direct-illumination estimation. We present a learning-based method that uses normalizing flows to efficiently importance sample illumination product integrals, e.g., the product of environment lighting and material terms. Our sampler composes a flow head warp with an emitter tail warp. The small conditional head warp is represented by a neural spline flow, while the large unconditional tail is discretized per environment map and its evaluation is instant. If the conditioning is low-dimensional, the head warp can be also discretized to achieve even better performance. We demonstrate variance reduction over prior methods on a range of applications comprising complex geometry, materials and illumination.

*Work partly done during an internship at Adobe Research, UK.

SA Conference Papers '24, December 3–6, 2024, Tokyo, Japan

ACM ISBN 979-8-4007-1131-2/24/12…$15.00
https://doi.org/10.1145/3680528.3687566

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

Additional Key Words and Phrases: Monte Carlo light transport, photorealistic rendering, importance sampling, neural rendering, normalizing flows

ACM Reference Format:
Joey Litalien, Miloš Hašan, Fujun Luan, Krishna Mullia, and Iliyan Georgiev. 2024. Neural Product Importance Sampling via Warp Composition. In *SIGGRAPH Asia 2024 Conference Papers (SA Conference Papers '24), December 3–6, 2024, Tokyo, Japan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3680528.3687566

1 INTRODUCTION

Rendering photorealistic images using Monte Carlo methods requires sampling product integrals with intricate shapes. The most common practical approach is to construct individual estimators that each focus on different factors of the integrand and combine them using multiple importance sampling (MIS) [Veach and Guibas 1995]. This combination yields an effective *mixture* sampling distribution; however, it is preferable to approximate the true *product* of the factors, especially in complex material and lighting configurations. Such full product sampling can greatly improve efficiency even the specific case of direct illumination estimation where the product includes the light source, the cosine term, and the material reflectance (BRDF) term.

The rendering community has recently begun investigating importance sampling methods based on neural probabilistic models. Along this axis, discrete normalizing flows (NFs) [Dinh et al. 2014;

Rezende and Mohamed 2015] provide an elegant generative framework for constructing flexible distributions by only requiring the specification of a (typically simple) base distribution and a series of bijective transformations, or *warps*. Seminal works [Müller et al. 2019; Zheng and Zwicker 2019] showed that NFs can be successfully applied to Monte Carlo rendering by learning on-the-fly from generated samples. However, despite their theoretical appeal and versatility, NF adoption in modern rendering engines has been mostly held back by their high computational cost. Recently, Xu et al. [2023] have shown that this may not longer hold true, proposing an NF-based framework with practical rendering speed-ups. The ever-growing demand for real-time photorealism has fostered the development of shading languages where neural networks are first-class citizens [Bangaru et al. 2023; Vaidyanathan et al. 2023], which is likely to further improve performance of NFs in future shaders.

We build on these findings and present a method to estimate product integrals using neural probabilistic models. Building atop NFs, our method learns to generate samples from the product distribution of a material model (e.g., a microfacet BRDF) and a distant emitter (an HDR environment map). Key to our method is the composition of a neural spline flow with a fast emitter warp. Our model learns to deform a uniform distribution into an intermediate one that is then transformed to the desired product PDF. We show that imbuing our model with a near-exact emitter warp is an effective inductive bias for neural product sampling. As our head network does not need to learn the fine details of the environment map, it can specialize to focus on the lower-dimensional conditional variations, which drastically simplifies the fitting task.

Our model is compact, shows competitive performance to traditional sampling methods, and integrates easily into an existing rendering pipeline. We implement it into the Mitsuba 3 renderer [Jakob et al. 2022b] and demonstrate reduced variance and improved visual quality on a variety of product sampling applications and scene configurations.

In summary, our key contributions are:

- A new compositional approach for product importance sampling based on normalizing flows, which combines a small but general head warp, represented by a neural spline flow, with a large tail warp, precomputed per environment map for fast evaluation;
- a novel neural architecture based on a circular variant of rational-quadratic splines; and
- integration into a practical rendering system showing improved performance over previous works in terms of equal sample count and equal time.

2 BACKGROUND AND RELATED WORK

Monte Carlo integration. We are interested in solving the surface reflection equation [Kajiya 1986] which states that the outgoing radiance L_o at a point $\mathbf{x}$ in direction $\boldsymbol{\omega}_\mathrm{o}$ is given by

$$L_\mathrm{o}(\mathbf{x}, \boldsymbol{\omega}_\mathrm{o}) = \int_\Omega f_\mathrm{r}^\perp(\mathbf{x}, \boldsymbol{\omega}_\mathrm{o}, \boldsymbol{\omega})\, L(\boldsymbol{\omega})\, V(\mathbf{x}, \boldsymbol{\omega})\, \mathrm{d}\boldsymbol{\omega}. \tag{1}$$

In this integral over directions $\boldsymbol{\omega}$ on the unit sphere Ω, $f_\mathrm{r}^\perp$ is the (cosine-weighted) bidirectional reflectance distribution function (BRDF), L denotes radiance emitted by a light source which we consider to be distant (i.e., an environment map), and the visibility indicator function V is zero when that radiance is blocked by the scene geometry.

We construct a Monte Carlo (MC) estimator for the above integral:

$$\langle L_\mathrm{o} \rangle_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f_\mathrm{r}^\perp(\mathbf{x}, \boldsymbol{\omega}_\mathrm{o}, \boldsymbol{\omega}_i)\, L(\boldsymbol{\omega}_i)\, V(\mathbf{x}, \boldsymbol{\omega}_i)}{p(\boldsymbol{\omega}_i \,|\, \mathbf{x}, \boldsymbol{\omega}_\mathrm{o})}, \quad \boldsymbol{\omega}_i \sim p, \tag{2}$$

which draws N directions $\boldsymbol{\omega}_i$ from a distribution with probability density function (PDF) p. The variance of $\langle L_\mathrm{o} \rangle_N$ becomes small when p is approximately proportional to the product $f_\mathrm{r}^\perp \cdot L \cdot V$. Finding such a PDF in practice is difficult, and in practice multiple importance sampling (MIS) [Veach and Guibas 1995] is used to combine separate estimators of the above form, each using a PDF that targets one of the factors. MIS is suboptimal as it is equivalent to using a mixture sampling distribution (rather than a product) and can be overly defensive [Karlík et al. 2019]. Devising distributions with closer proportionality to the actual product integrand remains the best way to achieve low estimation variance.

Product importance sampling. To estimate illumination on glossy surfaces from distant emitters, Clarberg et al. [2005] proposed a sparse wavelet representation along with a hierarchical sample warping scheme that approximates the product. Given the wavelet's large memory usage, Clarberg and Akenine-Möller [2008] instead suggested a sparse quad-tree built on-the-fly from BRDF samples to fit the distribution. Herholz et al. [2016] estimated indirect illumination via an adaptive Gaussian mixture model optimized via expectation-maximization. Conty Estevez and Lecocq [2018] used a proxy BRDF representation along with a spherical tabulation technique to perform approximate product importance sampling of multi-lobe materials. These approaches require multiple samples per shading point to amortize the precomputation. In contrast, our model is lightweight and remains efficient in low-sample regimes.

Xia et al. [2020] proposed a Gaussian representation for product importance sampling of multi-layered materials, while Villeneuve et al. [2021] devised analytical techniques for volumetric single scattering along rays. Unlike our framework, these methods are tailored to specific appearances and cannot be easily adapted to account for image-based lighting products.

Resampled importance sampling [Bitterli et al. 2020; Talbot 2005; Talbot et al. 2005] approximates a target (e.g., product) distribution by carefully picking a subset of candidate samples. The approximation quality is better the closer the candidate distribution is to the target. Resampling can be applied atop our neural sampler.

Closest to our work is the framework of Hart et al. [2020] who also recognized that sample warps can—in theory—be composed to achieve perfect product importance sampling. Their work specifically targets area light sources and uses simple analytic warps to approximately correct for the optimal transformation *after* sampling from a known strategy. Our method can be seen as complementary: we *start* with a more flexible, trainable neural warp and then correct with an environment emitter warp.

Normalizing flows. Normalizing flows (NFs) are a class of generative models that can construct arbitrary probability distributions [Chen et al. 2018; Dinh et al. 2014; Rezende and Mohamed 2015].

By design, these models achieve exact likelihood computation and efficient sampling [Papamakarios et al. 2021], two highly desirable properties for tasks such as density estimation, variational inference and, importantly for us, unbiased integral estimation.

At their core, NFs operate by learning a diffeomorphism (differentiable bijection) f from a simple base distribution $p_{\mathcal{Z}}$ to a more intricate distribution $p_{\mathcal{X}}$ via a series of warps that are parameterized by neural networks. Formally, if $f : \mathcal{Z} \to \mathcal{X}$ is an invertible transformation with tractable Jacobian determinant $\mathbf{J}_f(z;\boldsymbol{\theta}) \triangleq \det\left(\partial f(z;\boldsymbol{\theta})/\partial z^T\right)$ and $z \sim p_{\mathcal{Z}}$, we can express the post-warp density $p_{\mathcal{X}}$ via the change-of-variables formula:

$$p_{\mathcal{X}}(x;\boldsymbol{\theta}) \;=\; p_{\mathcal{Z}}(z)\left|\mathbf{J}_f(z;\boldsymbol{\theta})\right|^{-1}, \tag{3}$$

where $z = f^{-1}(x;\boldsymbol{\theta})$, and $\boldsymbol{\theta}$ represents the parameters of the flow model f. Under this formulation, efficient sampling of the resulting distribution *and* density evaluation can be simultaneously achieved by mapping $f(z;\boldsymbol{\theta}) = x$ forward and evaluating the determinant along the way. In practice, f is implemented as a composition of parameterized *coupling layers*, each designed to have a lower-triangular Jacobian matrix with easy to compute determinant. The parameters $\boldsymbol{\theta}$ are optimized to approximate a target measure $p^*_{\mathcal{X}}$ by minimizing a discrepancy metric such as Kullback–Leibler divergence. If samples $x \sim p^*_{\mathcal{X}}$ are available at training time, this is equivalent to fitting the model by maximum likelihood estimation.

While most NF literature focuses on high-dimensional problems such as image generation [Chen et al. 2019; Kingma and Dhariwal 2018], NFs shine in low-dimensional settings where remarkable density fits can be obtained with a fairly low parameter count. To improve robustness, neural spline flows [Durkan et al. 2019] introduce monotonic rational-quadratic splines to build invertible flow transformations. We adopt the circular variant of this framework [Rezende et al. 2020], with a uniform base distribution, and compose it with other warps to produce high-quality target samples in a stable manner.

Neural sampling for rendering. Normalizing flows have recently gained prominence in rendering for importance sampling [Müller et al. 2019; Zheng and Zwicker 2019], with extensions to control variates [Müller et al. 2020]. Xu et al. [2023] tackled the sampling of neural materials [Kuznetsov et al. 2021, 2022] by conditioning an NF model on pretrained neural feature descriptors. Zeltner et al. [2024] instead utilized a decoder network to extract the parameters of an analytic-lobe mixture for importance sampling complex materials; a similar approach has been explored in prior neural BRDF works [Fan et al. 2022; Sztrajman et al. 2021]. Tangentially, Dong et al. [2023] learned neural parametric mixtures for path guiding by implicitly encoding the spatially varying radiance distribution in a scene. Finally, Vicini et al. [2019] proposed a BSSRDF model based on a variational autoencoder that learns local geometry with low-order polynomial for subsurface scattering.

Neural samplers typically learn a full distribution from raw samples and do not leverage the existence of efficient sampling techniques for its individual factors. In contrast, our approach explicitly integrates exact emitter sampling as a form of inductive bias into the pipeline, which drastically simplifies the NF fitting task.

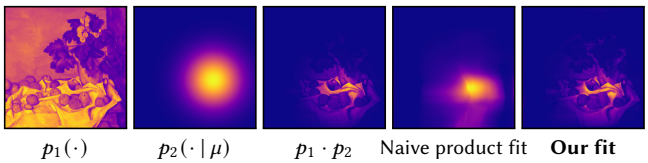


Fig. 2. Naively fitting a normalizing flow (NF) model to the product of a complex unconditioned density p_1 (image) and a simple conditioned density p_2 (Gaussian with parameterized mean μ) yields a poor result. The model is tasked with simultaneously learning the intricate shape of p_1 *and* the variations in μ. Instead, we apply a p_1 warp to the NF-model output, which drastically simplifies the shape of the distribution it needs to learn. The result is a near-perfect fit with an equal number of NF parameters.

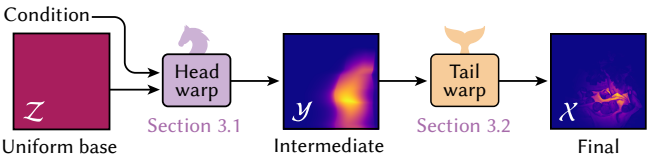


Fig. 3. Given a shading condition, our model maps uniform points through two warps to produce samples distributed approximately proportionally to a target product density. The shape of our intermediate density is coarse, similarly to a naive product fit (see Fig. 2), but leads to a precise fit when mapped through the tail warp.

3 NEURAL WARP COMPOSITION

Our goal is to generate samples proportionally to the product of two given densities, $p^*(x \mid \mathbf{c}) \propto p_1(x) \cdot p_2(x \mid \mathbf{c})$. We assume that the unconditional density p_1 has complex shape (e.g., an unshadowed environment map), while the conditional p_2 has simpler shape (e.g., BRDF parameterized by view direction, surface normal, roughness, etc). We do not explicitly model visibility as it is scene-dependent.

We could train one monolithic flow-based model to warp a base distribution into the target product p^*. Unfortunately, achieving high run-time performance requires a compact model which would not achieve a good fit because the product has complex shape that varies with the condition $\mathbf{c}$. We show an example in Fig. 2 on a product of an image-based density and a Gaussian with parameterized mean. With this naive strategy, a compact NF model has the capacity to learn only the rough shape of the target distribution.

Our key insight to addressing this problem is that we can handle the complexity and conditioning of the target separately via warp composition. Specifically, instead of using one complex conditional NF model to directly generate samples from the target product p^*, we use a *compact conditional head* NF model and feed its output to a *complex unconditional tail* warp derived from p_1. This separation drastically simplifies the learning space of the NF model by tasking it to fit a smoother intermediate distribution which is subsequently transformed into the final complex product by the tail warp. The latter can be constructed to efficiently handle the intricate shape of p_1 as we will discuss below. Note that our head-warp optimization is different from simply fitting to $p_2(\mathbf{x} \mid \mathbf{c})$ which would not yield the correct product distribution. Figure 3 shows a high-level overview of our method.

Formally, our model f is the composition of a conditional NF head warp $h : \mathcal{Z} \to \mathcal{Y}$, parameterized by $\boldsymbol{\theta}$, and an unconditional non-parameterized tail warp $t : \mathcal{Y} \to \mathcal{X}$:

$$f(z \mid \mathbf{c}; \boldsymbol{\theta}) \triangleq (t \circ h)(z \mid \mathbf{c}; \boldsymbol{\theta}) = t\big(h(z \mid \mathbf{c}; \boldsymbol{\theta})\big). \tag{4}$$

Provided that h, t, and their inverses are all continuous, $f : \mathcal{Z} \to \mathcal{X}$ is also a diffeomorphism and can be used to fit our target distribution. Hereinafter, we will use subscripts in density notations for disambiguation; our target product density is thus $p^*_{\mathcal{X}}(x \mid \mathbf{c})$ and the density learned by our model is $p_{\mathcal{X}}(x \mid \mathbf{c}; \boldsymbol{\theta})$.

Application to rendering. In our application, the target density is proportional to the product of unshadowed environment light (p_1) and cosine-weighted BRDF (p_2). The *shading condition* $\mathbf{c} \triangleq (\boldsymbol{\omega}_\mathrm{o}, \mathbf{n}, \boldsymbol{\rho})$ includes view direction $\boldsymbol{\omega}_\mathrm{o}$, surface normal $\mathbf{n}$, and a material-specific descriptor $\boldsymbol{\rho}$ (e.g., roughness or neural-material embedding). All domains coincide and are square: $\mathcal{Z} = \mathcal{Y} = \mathcal{X} = [0, 1]^2$. The tail-warp output samples x are finally transformed to the unit sphere via a lat-long warp, to obtain a sampled direction $\boldsymbol{\omega}$; we omit that warp here but plot distributions on the domain $\mathcal{X}$ as lat-long maps.

Next we detail the individual warps and describe how we train their composition via maximum likelihood estimation.

3.1 Neural-flow head warp

The goal of our head warp h is to transform a base distribution $p_{\mathcal{Z}}$ to an intermediate distribution $p_{\mathcal{Y}}$ that, when pushed through the tail warp t, yields the target distribution, i.e., $p_{\mathcal{X}} \approx p^*_{\mathcal{X}}$. To that end, we build h upon neural spline flows [Durkan et al. 2019] which utilize monotonic piecewise rational-quadratic splines as quantile functions (i.e., inverse CDFs) to warp samples. We use the circular variant of these flows [Rezende et al. 2020] to produce samples on the unit cylinder, since the azimuth angle in our final lat-long map wraps around. One important distinction over standard NF models is that we use a *uniform* base distribution on the unit square, $p_{\mathcal{Z}} = \mathcal{U}[0, 1]^2$, to enforce compact support. This is not directly feasible with a normal base distribution.

Figure 4 depicts our flow model. Given the shading condition $\mathbf{c}$, a fully connected encoder network first predicts a feature vector $\boldsymbol{\xi}$ that couples the condition's components in latent space. The concatenation $(\mathbf{c}, \boldsymbol{\xi})$ in turn conditions each of two consecutive coupling layers that transform sample coordinates. Each layer has its own network that infers spline parameters. To sample from the model, we draw $z \sim \mathcal{U}[0, 1]^2$ and feed it to the first coupling layer. The output of the second layer, $h(z) = y$, is then passed to the next warping stage.

Relation to NeuSample. Our model is inspired by NeuSample [Xu et al. 2023] but has some key differences. Apart from the already discussed architectural disparities, such as the warp composition and the uniform base, we use a *global* (cylindrical) equi-rectangular parameterization while NeuSample learns on the *local* shading-point hemisphere (projected onto a square). Our method thus guarantees that every point $y \in \mathcal{Y}$ has a pre-image in $\mathcal{Z}$ by design, which is not the case for NeuSample as it needs to learn the disk boundary and occasionally reject out-of-domain samples.

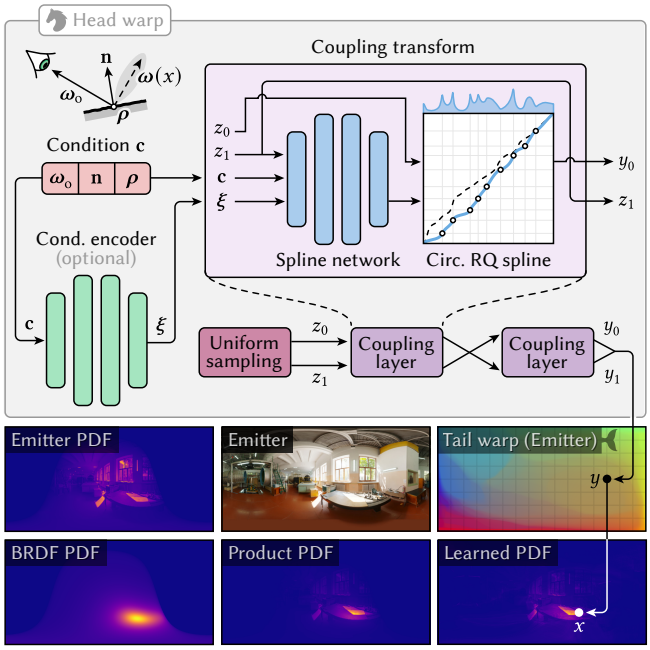


Fig. 4. Given a shading condition $\mathbf{c}$ (view direction $\boldsymbol{\omega}_\mathrm{o}$, surface normal $\mathbf{n}$ and material descriptor $\boldsymbol{\rho}$), a conditioner encoder first produces a latent vector $\boldsymbol{\xi}$. The vectors $\mathbf{c}$ and $\boldsymbol{\xi}$ condition two coupling layers, each warping samples via a circular piecewise rational quadratic (RQ) spline whose parameters (i.e., knot positions and derivatives) are inferred by a spline network. The output $y = (y_0, y_1)$ is then passed through our tail warp to produce the final sample x which is converted to a direction $\boldsymbol{\omega}$ via lat-long mapping.

3.2 Tail warp

The tail of our pipeline is an unconditional transformation of head-warp samples $y \in \mathcal{Y}$ to unit directions $\boldsymbol{\omega}$. The samples y are first mapped to unit-square points $x \in \mathcal{X}$ according to the density defined by a high-dynamic-range environment image (Fig. 4, bottom right), which are finally lat-long projected to the sphere. Several options for this tail warp are available, such as the common marginal row-column scheme or the hierarchical one of Clarberg et al. [2005]. Both can be constructed quickly but are discontinuous. This is usually not a major problem in practice, except that discontinuities may ruin the stratification of the input samples z. In our case a discontinuous tail warp may hinder the head-warp optimization, since small variations in $\mathcal{Y}$ may lead to abrupt changes in $\mathcal{X}$. A smooth emitter warp is thus desirable. But how does one construct such a warp?

We adopt a pragmatic approach and fit a large NF model to the emitter image. By construction, a spline-based flow guarantees a smooth map, and since this specific distribution is unconditional, it can be trained efficiently with samples generated using any of the two aforementioned schemes. Increasing the number of spline bins makes the added approximation error arbitrary small. We considered using an optimal-transport map [Feydy et al. 2019; Tong et al. 2024]. However, due to the regularization required to obtain practical OT solutions, it did not perform well, unless the target itself was smooth (which is far from the case in natural environment maps). Figure 5 shows how the tail-warp smoothness affects rendering quality.

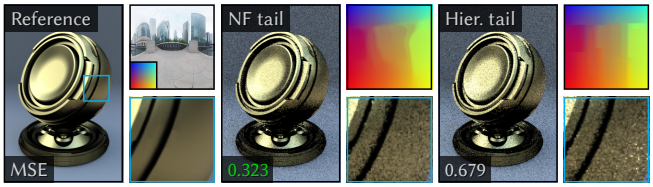


Fig. 5. Using a hierarchical tail warp exhibits discontinuities and hinders head-warp optimization. We do not show the standard marginal row-column warp as it performs even worse. In contrast, our NF-based tail warp makes for a smooth mapping and halves the MSE of images rendered with our full model. Colors encode the mapping from the unit square.

We train the tail NF model and discretize its forward and inverse warps, along with their respective densities, at high resolution. Sampling and PDF evaluation then reduce to simple bilinearly interpolated lookups. Note that this warp is constructed (in isolation) to transform *uniform* density to the emitter-image density. Next, we optimize the head warp to feed *nonuniform* input such that the output density is proportional to the target product.

3.3 Head-warp training

We optimize the head-warp parameters $\boldsymbol{\theta}$ by minimizing the *forward* Kullback–Leibler (KL) loss $\mathcal{L}_{\mathrm{KL}}(\boldsymbol{\theta})$ between the target density p^*_X and the fit p_X (below we omit the condition $\mathbf{c}$ for brevity):

$$\mathcal{L}_{\mathrm{KL}}(\boldsymbol{\theta}) = D_{\mathrm{KL}}\left[p^*_X(x) \,\middle\|\, p_X(x;\boldsymbol{\theta})\right] \tag{5}$$

$$= -\mathbb{E}_{x\sim p^*_X}\left[\log p_X(x;\boldsymbol{\theta})\right] + \text{const.} \tag{6}$$

$$= -\mathbb{E}_{x\sim p^*_X}\left[\log \left|\mathrm{J}_f(z;\boldsymbol{\theta})\right|^{-1}\right] + \text{const.}, \tag{7}$$

where we have expanded Eq. (3) on the last line, with $z = f^{-1}(x;\boldsymbol{\theta})$, and used the fact that our base distribution p_Z is uniform. This objective amounts to maximizing the log-likelihood w.r.t. p_X using data samples drawn from the target p^*_X. The sampling is done via on-the-fly tabulation and is detailed in Section 4.

Given that our model is the composition $f = t \circ h$, the target samples are mapped backward through the inverse composition $f^{-1} = h^{-1} \circ t^{-1}$; the corresponding Jacobians can be readily evaluated along the way. More precisely, we have

$$\log \left|\mathrm{J}_f(z;\boldsymbol{\theta})\right|^{-1} = \log \left|\mathrm{J}_t(h(z;\boldsymbol{\theta}))\,\mathrm{J}_h(z;\boldsymbol{\theta})\right|^{-1} \tag{8}$$

$$= \log \left|\mathrm{J}_t\big(h(z;\boldsymbol{\theta})\big)\right|^{-1} + \log \left|\mathrm{J}_h(z;\boldsymbol{\theta})\right|^{-1} \tag{9}$$

$$= \log \left|\mathrm{J}_{t^{-1}}(x)\right| + \log \left|\mathrm{J}_{h^{-1}}\big(t^{-1}(x);\boldsymbol{\theta}\big)\right|, \tag{10}$$

where in Eq. (10) we have applied the inverse function theorem.

Entropic regularization. KL-divergence optimization is susceptible to producing distributions that are overly aggressive in fitting high-density regions, leaving low density regions under-represented to the extent of producing fireflies in MC estimation. Theoretically χ^2 is a better loss that avoids fireflies, however it tends to be too conservative, producing overall higher noise levels, as also observed by Müller et al. [2019].

To improve robustness, we opt for a simpler approach that adds an entropic regularization term to our objective function (7):

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\mathrm{KL}}(\boldsymbol{\theta}) + \lambda \mathcal{L}_{\mathrm{H}}(\boldsymbol{\theta}), \qquad \text{where} \tag{11}$$

$$\mathcal{L}_{\mathrm{H}}(\boldsymbol{\theta}) = \mathbb{E}_{x\sim p^*_X}\left[p_X(x;\boldsymbol{\theta}) \log p_X(x;\boldsymbol{\theta})\right]. \tag{12}$$

This regularizer penalizes strong mismatched densities and mitigates degenerate cases that may lead to regions with too high variance responsible for fireflies. We set $\lambda = 0.0001$ for all experiments which we found strikes a good balance between outlier suppression and overall variance reduction.

Discussion. Our training scheme requires samples $x \sim p^*_X$ from the target. It is tempting to instead optimize the *reverse* KL divergence where the expectation is taken over *model-generated* samples $x \sim p_X$. That scheme does not necessitate discretizing the target p^*_X whose evaluation is required only up to a normalization constant [Papamakarios et al. 2021]. Unfortunately, we found that reverse KL optimization is generally too unstable for training with warp compositions. We hypothesize that better initialization schemes for the head flow may alleviate some of the numerical difficulties.

4 APPLICATIONS AND RESULTS

We implemented our method in PyTorch [Paszke et al. 2019], with fused neural networks in CUDA [Müller 2021], and integrated it into Mitsuba 3's [Jakob et al. 2022b,a] wavefront path-tracing renderer. In this section we evaluate its performance on several applications. We begin by describing our experimental setup.

Model hyperparameters. Our head warp comprises a small conditioner network followed by 2 coupling layers, which we found strikes a good balance between speed and sample quality. We use 32 spline bins. All networks have two layers with 64 neurons each with leaky ReLU activations, for a total of 36k parameters. Our emitter tail warp is a much larger flow network with 128 bins, 16 coupling layers with 256 hidden neurons, and two residual blocks. This network's forward/inverse maps and PDFs are discretized at 1k resolution which we found to be sufficient even for large environment maps.

Training. We train using the AdamW optimizer [Loshchilov and Hutter 2019] with a learning rate $\eta = 0.001$ and hyperparameters $\beta = (0.9, 0.999)$. Every training batch contains 256k samples: 1024 samples $x \sim p^*_X(\,\cdot \mid \mathbf{c})$ for each of 256 conditions $\mathbf{c}$. We train for 10k iterations on an NVIDIA A100 GPU; our most costly training with neural materials converges within 30 minutes, while the cosine-weighted and microfacet applications both take 20 minutes to train on average. The one-time pre-training of the tail warp is achieved in under 25 minutes.

For each condition $\mathbf{c} = (\boldsymbol{\omega}_{\mathrm{o}}, \mathbf{n}, \boldsymbol{\rho})$, the outgoing direction $\boldsymbol{\omega}_{\mathrm{o}}$ and normal $\mathbf{n}$ are sampled over the sphere. Since $\boldsymbol{\omega}_{\mathrm{o}} \cdot \mathbf{n} < 0$ cannot occur for opaque surfaces, we flip the normal accordingly. The material descriptor $\boldsymbol{\rho}$ is roughness for the Trowbridge–Reitz model or a NeuMIP feature vector. For Lambertian BRDFs, the condition is simply $\mathbf{c} = \mathbf{n}$. To generate samples for the condition, we evaluate the (unshadowed) integrand in Eq. (1) on a 512×1024 lat-long grid and tabulate the resulting luminance values into a CDF.

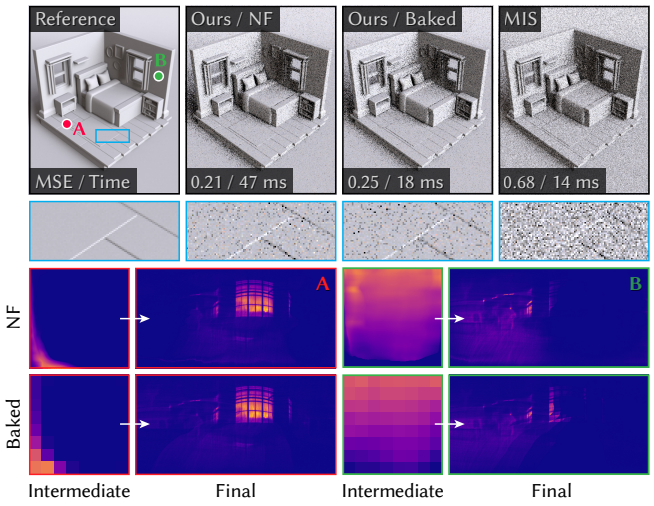


Fig. 6. Two intermediate distributions (at scene points A & B) learned by our head-warp NF model (top plots) and their low-resolution bakes (bottom plots) used for fast inference on the Dioramascene. Our baked scheme roughly matches the run time of the MIS baseline and achieves a 2.4× MSE reduction over it, with minimal noise increase over the slower NF model. Note that most of the remaining noise for our method comes from visibility.

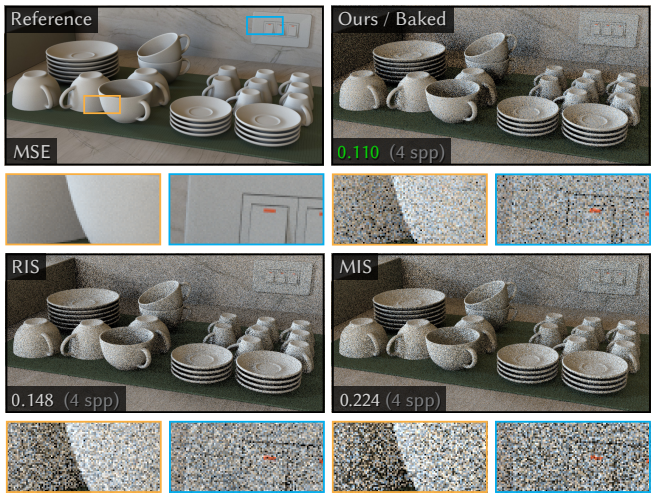


Fig. 7. Dishware scene (equal time). Our compact baked model for cosine-weighted emitter sampling achieves high performance and low variance.

Evaluation. We evaluate the performance of our model by measuring the mean square error (MSE) against several baselines at equal rendering time and equal sample count. We found MSE to be less sensitive to outliers than relative MSE in darker regions, which are more omnipresent in direct illumination renderings.

4.1 Cosine-weighted emitter sampling

We first demonstrate the benefit of our method to the case of emitter sampling. The popular hierarchical and row-column techniques do not account for surface orientation and generate zero-contribution directions below the local surface horizon, which is suboptimal. Our method can sample the upper hemisphere proportionally to the product of unshadowed radiance and cosine foreshortening by conditioning our model on the surface normal in global space.

Lightweight, baked head warp. Given the low dimensionality of this problem, we opt for a lighter head-warp architecture without the conditional encoder, directly feeding the normal to the flow network. We also reduce the number of spline bins to 4 and use a hierarchical tail warp as we empirically found that smoothness in the tail warp is not necessary to accurately learn this distribution. This model reduction makes our approach practical, as optimization and tail-warp baking can both be done in under five minutes.

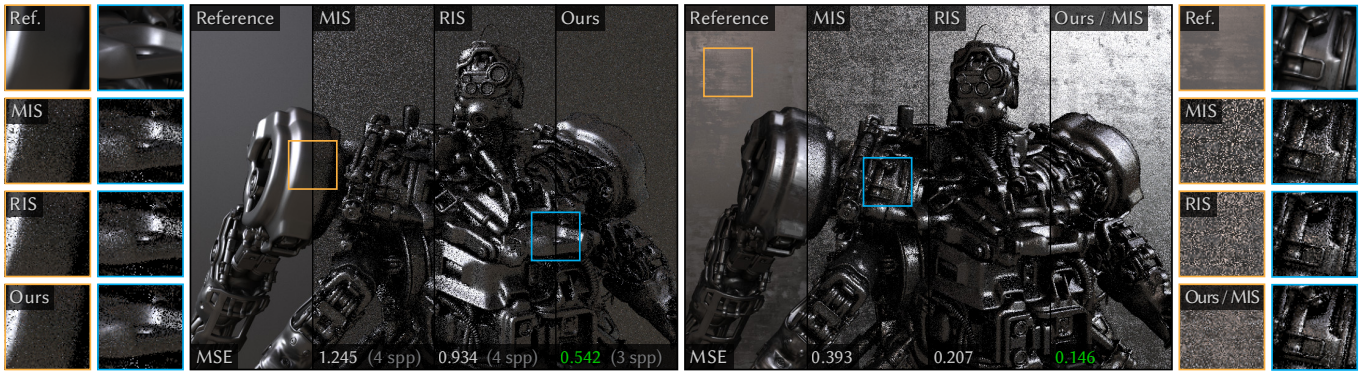
Even with these architectural changes, inference with our neural head warp still incurs a small performance overhead on scenes with simple geometry and materials. The use of MIS with BRDF sampling further exacerbates this problem as our network needs to run twice: once in each direction to compute the MIS weights. To alleviate this issue, we adopt a similar strategy as Xu et al. [2023] and bake our model into small histograms for fast sampling and PDF evaluation. A key observation here is that we only need to bake the simple *intermediate* distribution (Fig. 3) for a set of normal directions. This affords the use of a very low resolution and thus memory footprint. We compute 16×32 histograms with 8×8 resolution each (total of 132KB on disk), to achieve a substantial inference speed-up at only slightly increased variance. We label this variant as *Ours / Baked*.

Results. Figure 1 showcases our cosine-weighted emitter model applied to the Temple scene. Here, our method can well capture the variations in surface orientations, reducing the variance by 2.2× over MIS across the entire image. Our fast histogram variant achieves equal-time performance at equal sample count. In Fig. 6, we visualize intermediate densities of the native NF model and its baked counterpart at two conditions (i.e., surface points). Even at very low resolution, our histograms can yield faithful final product distributions at a fraction of the inference cost. In Fig. 7, we show additional results and compare against MIS and streaming RIS (4 candidates) at equal time, demonstrating once again superior quality over these baselines.

4.2 Microfacet materials

As a second application, we fit the product of a distant emitter with a microfacet-based BRDF with a Trowbridge–Reitz (GGX) distribution. The problem is now higher-dimensional as our model has to learn the complex 5D coupling dynamics between surface normal, view direction, and roughness—the conditions to our model.

Results. To assess how our model adapts to directional and scaling changes in the resulting product lobes, we train two variants: one with fixed roughness $r = 0.4$ and another with uniformly sampled roughness $r \in [0.2, 0.8]$. We benchmark on a scene with intricate geometry to challenge our model under a wide variety of conditions. To test different roughness values, we apply a texture on the material roughness. We compare against classical MIS and streaming RIS [Bitterli et al. 2020] with 4 candidates as they are the most practical existing strategies in this scenario.



Fixed roughness / Equal render time (1.0 s) Varying roughness / Equal sample count (2 spp)

Fig. 8. Droid scene. We apply our method to product importance sampling of microfacet BRDF and distant emitter. *Left*: When trained on a fixed roughness value, our model easily outperforms MIS and streaming RIS (4 candidates) at equal rendering time. *Right*: When roughness is an additional condition to our network, our head warp attains a worse fit, but an MIS combination with BRDF sampling achieves variance reduction at equal sample count.

Figure 8 shows the results. At fixed roughness, our model achieves excellent variance reduction at equal time. When roughness is also part of the condition, it still outperforms MIS at equal time but not RIS. We report MSE at equal sample count where we MIS-combine our model with BRDF sampling to further reduce noise. It is worth noting that we keep the capacity of our model to a minimum for performance reasons; as such it may not produce latent vectors $\boldsymbol{\xi}$ that are expressive enough to fully capture the input dependencies. Given the small size of our model, we posit that distilling it into lightweight, fixed-roughness sub-models [Hinton et al. 2015] could alleviate this problem; we leave such investigation as future work.

4.3 Neural materials

The application of our sampling framework to products of illumination and neural materials is straightforward. Similarly to NeuSample [Xu et al. 2023], we use the NeuMIP [Kuznetsov et al. 2021] feature vector at the shading uv-coordinates as the material descriptor $\boldsymbol{\rho}$.

Results. We compare our method against several baseline sampling techniques: cosine-weighted, emitter, and the flow-based variant of NeuSample which we implemented to the best of our ability. NeuSample employs a projected-disk parameterization, whereas our approach is defined in the equirectangular domain and does not suffer from out-of-domain samples—all points in our unit square map to a valid direction.

We benchmark our approach on three fabric materials with different glossiness. In Fig. 9, we render a high-roughness striped welt bed sheet and compare against emitter sampling, cosine sampling and NeuSample at equal rendering time. While the neural BRDF exhibits seemingly complex patterns, importance sampling it using NeuSample brings an only small improvement over cosine sampling. Emitter sampling performs favorably to these two but does not account for the full product. Our method performs best as it is informed about both the NeuMIP feature vector and the emitter. Figure 10 shows lower-roughness neural materials under directional lighting which prevents NeuSample and its MIS combination with emitter sampling from improving over the baselines; our product-sampling method remains the best by a significant margin.

4.4 Shadow-catcher compositing

Finally we show how our model can be employed for compositing virtual objects into photographs using the shadow catcher method. Compositing typically requires two shading passes: one including both the inserted object and the shadow catcher, and one of just the shadow catcher without the object [Debevec 1998]. We take the ratio of the luminance on the catcher with/without the object and use it as the visibility mask when alpha-composing the object and its shadow into the photograph. This desired transparency (assuming a Lambertian shadow catcher) can be written as a ratio of hemispherical integrals:

$$\tau = \frac{\int_\Omega V(\mathbf{x}, \boldsymbol{\omega}) L(\boldsymbol{\omega}) \langle \mathbf{n}, \boldsymbol{\omega} \rangle \, \mathrm{d}\boldsymbol{\omega}}{\int_\Omega L(\boldsymbol{\omega}) \langle \mathbf{n}, \boldsymbol{\omega} \rangle \, \mathrm{d}\boldsymbol{\omega}}, \tag{13}$$

where $L(\boldsymbol{\omega})$ is the direct luminance (grayscale radiance) from the environment emitter and $\langle \mathbf{n}, \boldsymbol{\omega} \rangle$ is the clamped cosine term. Computing this ratio requires either two rendering passes or significant re-engineering of the renderer to be able to produce these two integral estimates at once.

We note that τ is equivalent to the expectation of visibility $V(\mathbf{x}, \boldsymbol{\omega})$ with respect to a PDF proportional to the product $L(\boldsymbol{\omega}) \langle \mathbf{n}, \boldsymbol{\omega} \rangle$ (the denominator in Eq. (13) is the PDF's normalization). This density is precisely what our cosine product sampler learns. We can thus estimate the opacity mask in a *single pass* using the simple estimator

$$\langle \tau \rangle_N = \frac{1}{N} \sum_{i=1}^{N} V(\mathbf{x}, \boldsymbol{\omega}_i), \quad \boldsymbol{\omega}_i \sim p_\chi(\boldsymbol{\omega} \,|\, \mathbf{x}; \boldsymbol{\theta}). \tag{14}$$

The final composition then amounts to multiplying the visibility mask to the image and adding the masked object.

Results. We demonstrate this application by extracting a rectified image from an environment map to ensure almost exact lighting and use fSpy [Gantelius 2019] to roughly match the camera settings and model the shadow receiver plane. We show the resulting composition in Fig. 11. Thanks to our ratio estimator, we get photorealistic object insertion in one rendering pass.

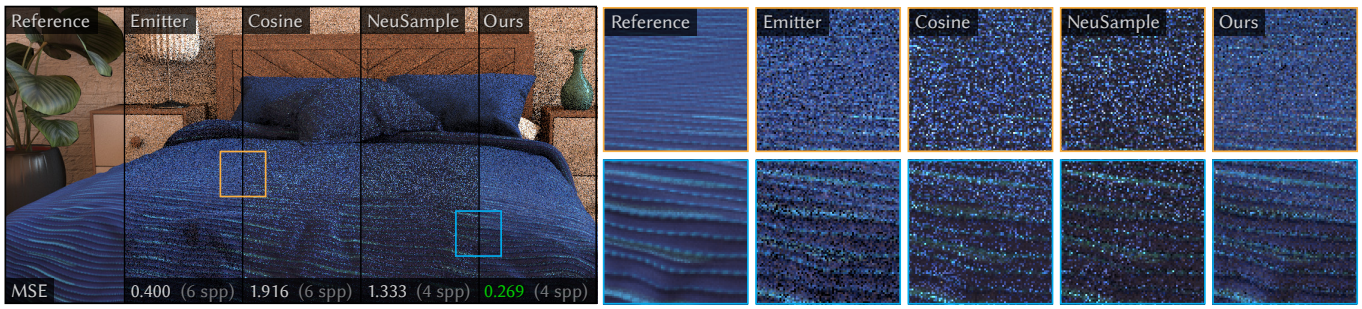


Fig. 9. Bedroom scene (equal time). For high-roughness neural materials like this striped welt fabric, cosine sampling and NeuSample [Xu et al. 2023] both produce high-variance results as they are oblivious to the non-uniform incident radiance. Emitter sampling does better but still samples only one factor in the product. Our model accounts for the product of material and lighting and performs best.

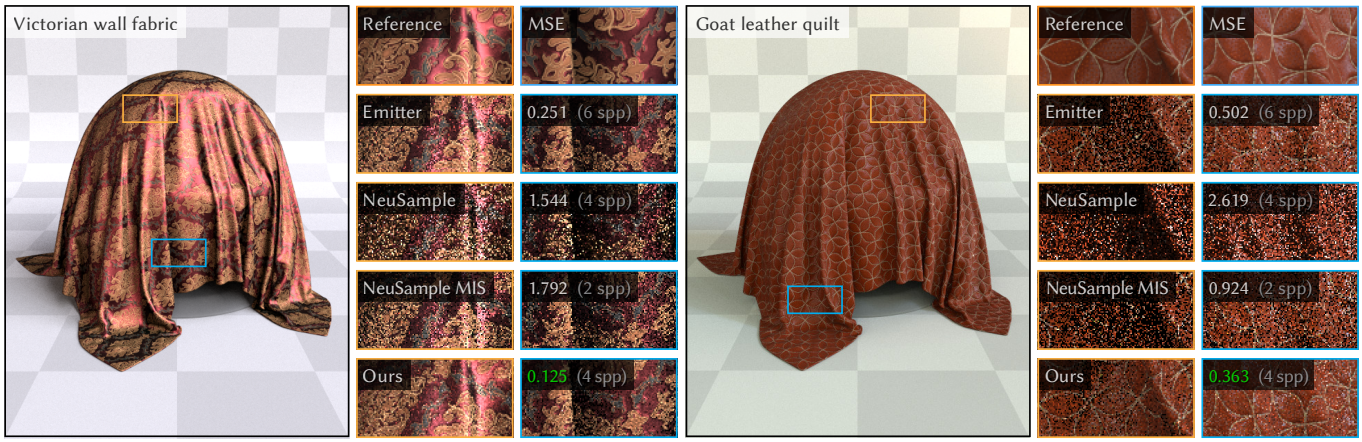


Fig. 10. Cloth scene (equal time). NeuSample yields excessive noise with lower-roughness neural materials illuminated non-uniformly, even when MIS-combined with emitter sampling. Emitter sampling alone does better but remains inferior to our product approach which generates clear results at equal time.

5 DISCUSSION

In Fig. 12 we plot the intermediate and final densities learned by our model on environment maps and materials used in our experiments. We compare to naively fitting the full product with a neural flow of the same capacity as our head warp. Unlike the naive fits, our final fits align well with their corresponding target densities, showcasing the benefit of separately handling detail and conditioning via warp composition.

Ablation study. To demonstrate the benefits of the individual components of our model, we conduct an ablation study and report results in Table 1. Unsurprisingly, naively fitting a flow to a product distribution performs much worse than our two compositional variants. Interestingly, the impact of using the smoother flow-based tail warp over a hierarchical one grows with BRDF complexity. Our conditional encoder and entropic regularization scheme further reduce error. We do not ablate the addition of an encoder for the cosine scheme as it is not part of the model.

Limitations. Our approach requires training per material model, with duration depending on conditioning dimension and target-distribution complexity. It provides benefit only when the training effort can be amortized in the subsequent rendering. To that end, the total preprocessing time can be cut in half when opting for a standard (e.g., hierarchical) emitter-sampling technique in lieu of optimizing a smooth tail warp. Our cosine-weighted model is particularly practical as it is optimized once per environment map, baked compactly, and used as an efficient drop-in replacement for traditional illumination sampling in any scene (also with MIS).

Our head warp may perform poorly when the product distribution is strongly dominated by the BRDF term (e.g., with very low roughness). We hypothesize that exposing an analytic parameterized BRDF warp to our model may mitigate this issue. Similarly, sun environment maps are not well supported: the tail warp collapses into a single point, which in turn causes numerical issues in the head-warp optimization. MIS can be applied in these scenarios but at the cost of an extra network pass for PDF evaluation. Future hardware acceleration might ease the implementation of fully fused flow kernels to better amortize such additional evaluations.

Future work. Our model is most effective with non-trivial scene complexity, i.e., when sampling is not the principal bottleneck in rendering. Applying our technique to more complex BRDFs, e.g., of layered materials [Guo et al. 2018], is thus a promising future avenue. Moreover, our current neural-material approach assumes the existence of a pre-trained (NeuMIP) material model; joint end-to-end

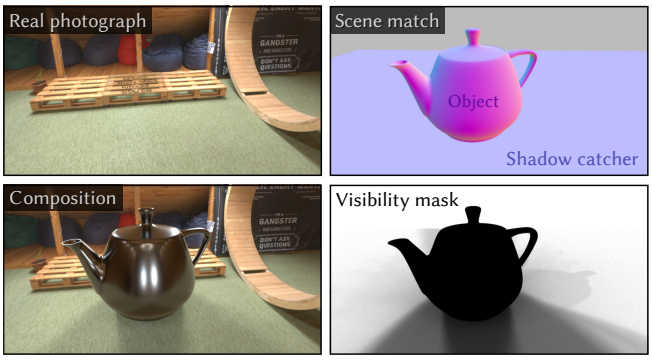


Fig. 11. Our cosine product sampler can be readily applied to shadow-catcher insertion of virtual objects into real photographs, in one render pass.

Table 1. Ablation of the main components of our model on three scenes, reporting rendering MSE. As expected, our full model performs best.

	Cosine Temple (Fig. 1)	**Microfacet** Droid (Fig. 8)	**Neural material** Bedroom (Fig. 9)
Naive neural-flow fit	0.715	1.411	1.320
Hierarchical tail warp	0.437	0.940	0.896
Neural-flow tail warp	0.406	0.901	0.739
+ Conditional encoder	—	0.580	0.323
+ Entropic regularization	**0.401**	**0.542**	**0.269**

training of material and sampling models would be preferable. Making our model MIS-aware could also help mitigate the drawbacks of poor conditional fits for near-unimodal products. For instance, using our model as the free strategy in MIS compensation [Karlík et al. 2019] could help further reduce variance. Finally, an application to non-axis-aligned portal-masked emitter sampling [Bitterli et al. 2015] seems within reach, where the flow network could be conditioned on a spatial feature grid to encode portal visibility.

6 CONCLUSION

Efficient product sampling of illumination and reflectance has been a long-standing problem in rendering. We introduced a novel compositional scheme that decouples the handling of distribution shape complexity and conditioning, to afford a compact model capable of achieving both tight fits to target product distributions and fast inference. We demonstrated the versatility of our approach through several practical applications, showing significant variance reduction on scenes of varying complexity. Our method excels when the shape of the product distribution does not degenerate into a singular island of density (e.g., very low roughness BRDFs or sky sun HDRIs). We hope our compositional approach stimulates renewed interest in flow-based models for light-transport simulation, which have been deemed impractical due to their high computational cost.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful feedback. All environment HDRIs were obtained from Poly Haven, while the 3D assets used to model the Temple, Droid, Dishware and Bedroom scenes were acquired from BlenderKit.

REFERENCES

Sai Praveen Bangaru, Lifan Wu, Tzu-Mao Li, Jacob Munkberg, Gilbert Bernstein, Jonathan Ragan-Kelley, Frédo Durand, Aaron Lefohn, and Yong He. 2023. SLANG.D: Fast, Modular and Differentiable Shader Programming. *ACM Transactions on Graphics* 42, 6, Article 264 (2023). https://doi.org/10.1145/3618353

Benedikt Bitterli, Jan Novák, and Wojciech Jarosz. 2015. Portal-Masked Environment Map Sampling. *Computer Graphics Forum (Proceedings of EGSR)* 34, 4 (June 2015). https://doi.org/10/f7mbx7

Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. 2020. Spatiotemporal Reservoir Resampling for Real-time Ray Tracing with Dynamic Direct Lighting. *ACM Transactions on Graphics* 39, 4 (2020). https://doi.org/10.1145/3386569.3392481

Ricky T. Q. Chen, Jens Behrmann, David Duvenaud, and Jörn-Henrik Jacobsen. 2019. Residual Flows for Invertible Generative Modeling. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. https://doi.org/10.5555/3454287.3455176

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. 2018. Neural Ordinary Differential Equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. https://doi.org/10.5555/3327757.3327764

Petrik Clarberg and Tomas Akenine-Möller. 2008. Practical Product Importance Sampling for Direct Illumination. *Computer Graphics Forum* 27, 2 (2008). https://doi.org/10.1111/j.1467-8659.2008.01166.x

Petrik Clarberg, Wojciech Jarosz, Tomas Akenine-Möller, and Henrik Wann Jensen. 2005. Wavelet Importance Sampling: Efficiently Evaluating Products of Complex Functions. *ACM Transactions on Graphics* 24, 3 (Aug. 2005), 1166–1175. https://doi.org/10.1145/1073204.1073328

Alejandro Conty Estevez and Pascal Lecocq. 2018. Fast Product Importance Sampling of Environment Maps. In *ACM SIGGRAPH 2018 Talks*. Article 69. https://doi.org/10.1145/3214745.3214760

Paul Debevec. 1998. Rendering Synthetic Objects Into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '98)*. 189–198. https://doi.org/10.1145/280814.280864

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2014. Density Estimation Using Real NVP. In *International Conference on Learning Representations*.

Honghao Dong, Guoping Wang, and Sheng Li. 2023. Neural Parametric Mixtures for Path Guiding. In *SIGGRAPH '23 Conference Proceedings*. https://doi.org/10.1145/3588432.3591533

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. 2019. Neural Spline Flows. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Article 675. https://doi.org/10.5555/3454287.3454962

Jiahui Fan, Beibei Wang, Miloš Hašan, Jian Yang, and Ling-Qi Yan. 2022. Neural Layered BRDFs. In *ACM SIGGRAPH 2022 Conference Proceedings*. https://doi.org/10.1145/3528233.3530732

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouve, and Gabriel Peyré. 2019. Interpolating between Optimal Transport and MMD using Sinkhorn Divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*. 2681–2690.

Per Gantelius. 2019. *fSpy*. https://github.com/stuffmatic/fSpy-Blender

Yu Guo, Miloš Hašan, and Shaung Zhao. 2018. Position-Free Monte Carlo Simulation for Arbitrary Layered BSDFs. *ACM Transactions on Graphics* 37, 6 (2018). https://doi.org/10.1145/3272127.3275053

David Hart, Matt Pharr, Thomas Müller, Ward Lopes, Morgan McGuire, and Peter Shirley. 2020. Practical Product Sampling by Fitting and Composing Warps. *Computer Graphics Forum* (2020). https://doi.org/10.1111/cgf.14060

Sebastian Herholz, Oskar Elek, Jiří Vorba, Hendrik Lensch, and Jaroslav Křivánek. 2016. Product Importance Sampling for Light Transport Path Guiding. *Computer Graphics Forum* 35, 4 (2016), 67–77. https://doi.org/10.1111/cgf.12950

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning and Representation Learning Workshop*.

Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. 2022b. *Mitsuba 3 Renderer*. https://mitsuba-renderer.org

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

James T. Kajiya. 1986. The Rendering Equation *(SIGGRAPH '86)*. 143–150. https://doi.org/10.1145/15886.15902

Ondřej Karlík, Martin Šik, Petr Vévoda, Tomáš Skřivan, and Jaroslav Křivánek. 2019. MIS Compensation: Optimizing Sampling Techniques in Multiple Importance Sampling. *ACM Transactions on Graphics* 38, 6, Article 151 (2019). https://doi.org/10.1145/3355089.3356565

Durk P. Kingma and Prafulla Dhariwal. 2018. Glow: Generative Flow with Invertible 1x1 Convolutions. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*.

Alexandr Kuznetsov, Krishna Mullia, Zexiang Xu, Miloš Hašan, and Ravi Ramamoorthi. 2021. NeuMIP: Multi-resolution Neural Materials. *ACM Transactions on Graphics* 40, 4 (2021). https://doi.org/10.1145/3450626.3459795

Alexandr Kuznetsov, Xuezheng Wang, Krishna Mullia, Fujun Luan, Zexiang Xu, Miloš Hašan, and Ravi Ramamoorthi. 2022. Rendering Neural materials on Curved Surfaces. In *ACM SIGGRAPH 2022 Conference Proceedings*. 1–9. https://doi.org/10.1145/3528233.3530721

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *7th International Conference on Learning Representations*.

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* 38, 5 (2019). https://doi.org/10.1145/3341156

Thomas Müller, Fabrice Rousselle, Alexander Keller, and Jan Novák. 2020. Neural Control Variates. *ACM Transactions on Graphics* 39, 6 (2020), 1–19. https://doi.org/10.1145/3414685.3417804

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. 2021. Normalizing Flows for Probabilistic Modeling and Inference. *Journal of Machine Learning Research* 22, 1, Article 57 (2021). https://doi.org/10.5555/3546258.3546315

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 8024–8035. https://doi.org/10.5555/3454287.3455008

Danilo Jimenez Rezende and Shakir Mohamed. 2015. Variational Inference with Normalizing Flows. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, Vol. 37. 1530–1538.

Danilo Jimenez Rezende, George Papamakarios, Sebastien Racaniere, Michael Albergo, Gurtej Kanwar, Phiala Shanahan, and Kyle Cranmer. 2020. Normalizing Flows on Tori and Spheres. In *Proceedings of the 37th International Conference on Machine Learning*. 8083–8092. https://doi.org/10.5555/3045118.3045281

Alejandro Sztrajman, Gilles Rainer, Tobias Ritschel, and Tim Weyrich. 2021. Neural BRDF Representation and Importance Sampling. In *Computer Graphics Forum*, Vol. 40. 332–346. https://doi.org/10.1111/cgf.14335

Justin F. Talbot. 2005. Importance Resampling for Global Illumination.

Justin F. Talbot, David Cline, and Parris Egbert. 2005. Importance Resampling for Global Illumination. In *Proceedings of the 16th Eurographics Conference on Rendering Techniques (EGSR '05)*. 139–146. https://doi.org/10.5555/2383654.2383674

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. 2024. Improving and Generalizing Flow-based Generative Models with Minibatch Optimal Transport. *Transactions on Machine Learning Research* (2024).

Karthik Vaidyanathan, Marco Salvi, Bartlomiej Wronski, Tomas Akenine-Möller, Pontus Ebelin, and Aaron Lefohn. 2023. Random-Access Neural Compression of Material Textures. *ACM Transactions on Graphics* 42, 4, Article 88 (July 2023). https://doi.org/10.1145/3592407

Eric Veach and Leonidas J. Guibas. 1995. Optimally Combining Sampling Techniques for Monte Carlo Rendering. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*. 419–428. https://doi.org/10.1145/218380.218498

Delio Vicini, Vladlen Koltun, and Wenzel Jakob. 2019. A Learned Shape-Adaptive Subsurface Scattering Model. *ACM Transactions on Graphics* 38, 4 (July 2019), 15. https://doi.org/10.1145/3306346.3322974

Keven Villeneuve, Adrien Gruson, Iliyan Georgiev, and Derek Nowrouzezahrai. 2021. Practical Product Sampling for Single Scattering in Media. In *Proceedings of EGSR*. https://doi.org/10.2312/sr.20211290

Mengqi (Mandy) Xia, Bruce Walter, Christophe Hery, and Steve Marschner. 2020. Gaussian Product Sampling for Rendering Layered Materials. *Computer Graphics Forum* 39, 1 (2020), 420–435. https://doi.org/10.1111/cgf.13883

Bing Xu, Liwen Wu, Miloš Hašan, Fujun Luan, Iliyan Georgiev, Zexiang Xu, and Ravi Ramamoorthi. 2023. NeuSample: Importance Sampling for Neural Materials. In *ACM SIGGRAPH 2023 Conference Proceedings*. https://doi.org/10.1145/3588432.3591524

Tizian Zeltner, Fabrice Rousselle, Andrea Weidlich, Petrik Clarberg, Jan Novák, Benedikt Bitterli, Alex Evans, Tomáš Davidovič, Simon Kallweit, and Aaron Lefohn. 2024. Real-Time Neural Appearance Models. *ACM Transactions on Graphics* (2024). https://doi.org/10.1145/3659577

Quan Zheng and Matthias Zwicker. 2019. Learning to Importance Sample in Primary Sample Space. In *Computer Graphics Forum*, Vol. 38. 169–179. https://doi.org/10.1111/cgf.13628

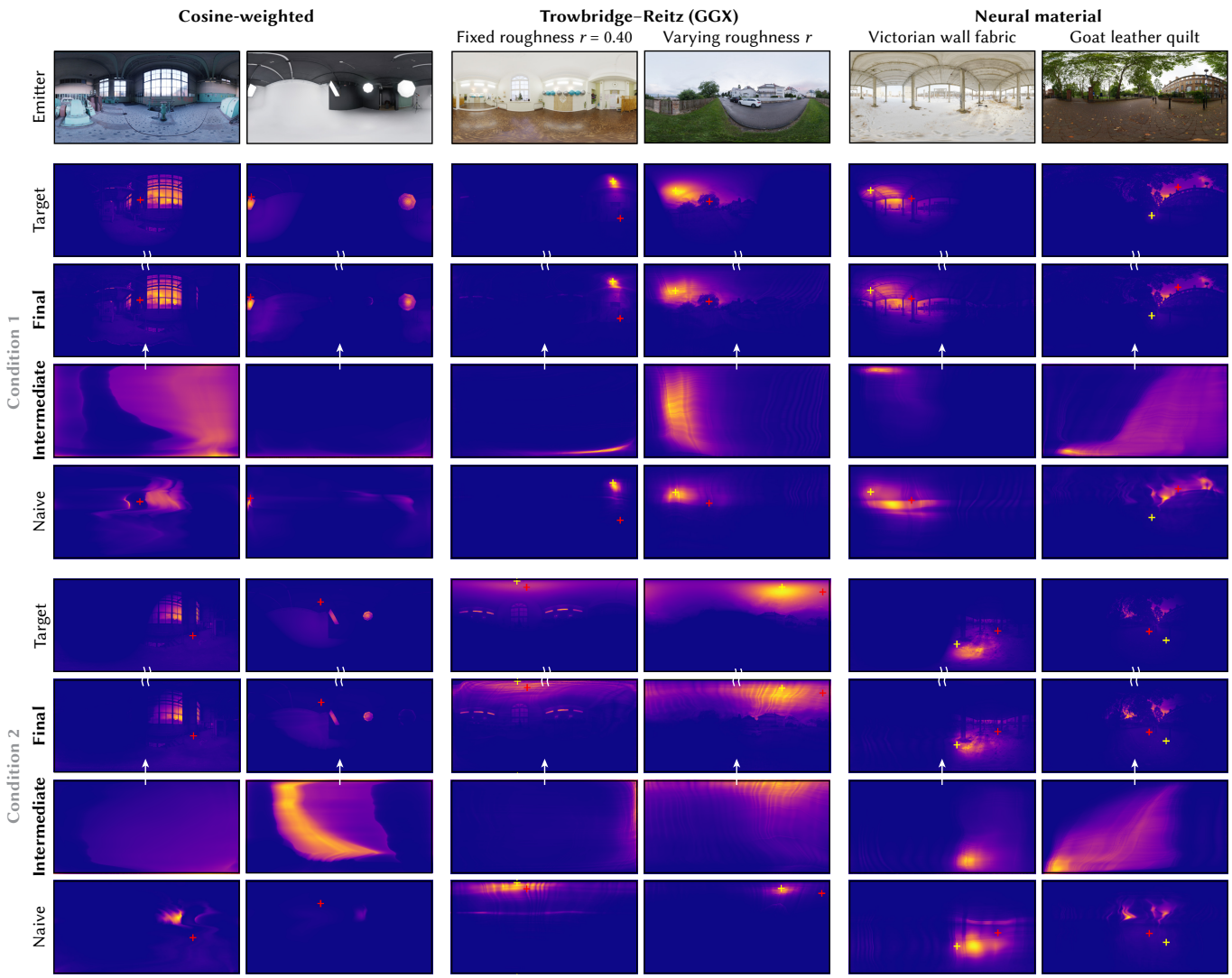


Fig. 12. For six environment maps, we plot fits of the product with one of three BRDFs used in our experiments. For each of the six configurations, we randomly sample two conditions and show the target product density and the **intermediate** and **final** densities learned by our model. For comparison we also visualize a naive neural-flow fit (i.e., without a tail warp). We mark the conditional (global) surface normal with a red cross and the reflected view direction with a yellow cross to localize the lobe in lat-long coordinates. All our learned final distributions align well with the target product, showcasing the benefit of warp composition for product importance sampling.