Eurographics Symposium on Rendering 2026
I. Gkioulekas and A. Jarabo
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 45 (2026), Number 4

Neural Material Adapter: Transforming Complex Materials into Efficient Analytic BRDFs

Rajesh Sharma[1,2], Tiziano Portenier[2], Sebastian Weiss[2] , Markus Gross[1,2], Marios Papas[2]

[1]ETH Zurich, Switzerland [2]DisneyResearch|Studios, Switzerland

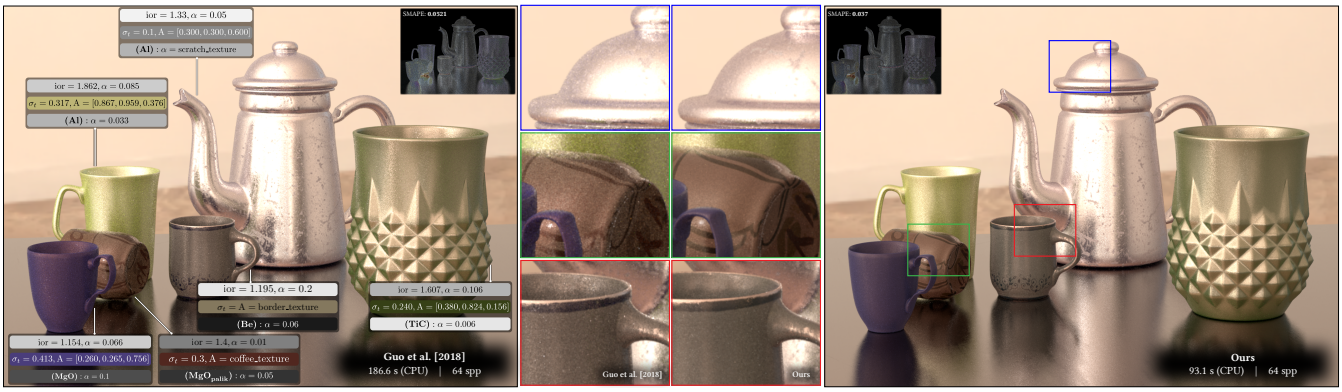

Figure 1: **Neural Material Adapter** uses the Principled BRDF augmented with additional lobes and with directionally varying parameters as a learnable representation to accurately represent complex multi-layered materials. We show a direct comparison of path-traced renders at 64 samples per pixel between the reference ([GHZ18] left, 186.6s, SMAPE 0.052) and **Neural Material Adapter** (right, 93.1s, SMAPE 0.037) with Mitsuba 0.5 [JSR*]. Our method accurately approximates complex materials and has lower variance at equal spp while being significantly faster to evaluate per sample. All renders on the CPU.

Abstract

Despite their superior visual fidelity, analytical multi-layered material models remain difficult to integrate into production pipelines due to heavy computational overhead and relative noise. We present the Neural Material Adapter (NMA), a framework that bridges the gap between high-fidelity material data and efficient rendering by mapping complex appearances into the parameter space of differentiable analytic BRDFs. Our approach is founded on the insight that directionally invariant parameters are fundamentally insufficient to capture the directional complexity of real-world light transport. We demonstrate that by allowing these parameters to vary freely with incident direction, we create a representation that defines an appearance gamut that not only overlaps with layered materials but also opens up regions physically unattainable by standard layered models. We use this expanded representation to fit high-fidelity isotropic appearances to a parametric Principled BRDF with directional parameter variation. By leveraging the strong inductive bias of differentiable analytic priors, we train a lightweight neural adapter that enables remarkably stable learning from sparse, noisy reference data. The resulting model is compact enough for efficient CPU-based inference without precomputation and exhibits powerful zero-shot generalization: once trained, NMA predicts view-dependent parameters for unseen layered material configurations and high-resolution textures. By decoupling complex directional appearance from heavy simulation, NMA enables a new class of expressive, high-fidelity materials that remain fully compatible with existing industry-standard rendering pipelines.

CCS Concepts

• ***Computing methodologies*** → ***Reflectance modeling; Ray tracing;***

1. Introduction

Realistic material appearance plays a critical role in computer graphics, impacting both realism and artistic control. Material-light interaction is typically modeled using analytical Bidirectional Scattering Distribution Functions (BSDFs), measured data-driven BSDFs, or specialized layered material models. Analytical physically-based BSDF models, such as the Principled BSDF by Burley [BS12], provide intuitive parameter controls but lack the flexibility to capture the full complexity of real-world appearances. Conversely, measured datasets, such as MERL [MPBM03] and RGL [DJ18], capture real-world appearances faithfully but offer limited editability.

A significant amount of research has aimed to bridge this gap by modeling the layered structure of materials. Notable examples include the analytical models of Weidlich and Wilkie [WW07] and Jakob et al. [JdJM14], as well as the statistical approaches by Belcour et al. [Bel18]. A particularly influential parametric model is the Position-Free Monte Carlo (PFMC) model proposed by Guo et al. [GHZ18], which simulates light transport through arbitrary material layers using a simplified path tracer. PFMC achieves high accuracy and supports the texturing of any layer parameter without precomputation. However, its numerical nature incurs a significant cost at render time.

Recent neural methods have sought to accelerate these high-fidelity approximations by learning from complex appearance data and, with sufficient network capacity, can generalize across a wide range of material appearances. However, because these methods learn the BSDF's directional function from scratch, they require large networks and extensive reference data, making both training and inference during rendering expensive. The inference cost can be alleviated by GPU acceleration [FWH*22, ZRW*24, KWM*22], but that is often impractical for CPU-based production renderers. Alternatively, expensive computation can be offloaded into preprocessing that encodes material appearance into baked per-texel representations decoded by a smaller network at render time [GLH*23, ZRW*24].

While latent textures reduce rendering inference costs, they require reprocessing whenever the source parameters or textures change, limiting editability and interactivity. Additionally, neural scattering functions are not invertible, so importance sampling is approximated [FWH*22, SRRW21, ZZW*22, XWH*23], leading to reduced sampling efficiency compared to analytic models.

Given these constraints, an alternative hybrid approach uses a lightweight neural network solely to translate complex appearance model parameters into the parameter space of a fast, physically based analytic BSDF [WJHY22, YSJR17]. By leveraging the inductive bias of the physically-based analytic target model to represent the directional distribution, these adapters only require lightweight networks that learn the parameter mapping rather than the complex directional distribution itself, substantially reducing both training and inference costs. Because evaluation and sampling are both performed by the analytic model, importance sampling is accurate by construction. However, existing neural material adapters have demonstrated success only on narrow material classes: hair and fur rendering [YSJR17] and interface-free microflake volume stacks [WJHY22].

Among the preceding methods, those that generalize across material classes either require per-texel precomputation for spatially-varying inputs [GLH*23] or sacrifice the exact importance sampling that production renderers depend on [FWH*22, ZRW*24], while the only adapters that achieve online parameter estimation remain limited to narrow material classes [YSJR17, WJHY22]. A fundamental gap remains: no existing method simultaneously delivers high directional accuracy, exact importance sampling, generalization across broad material classes, and zero-shot SVBRDF support via online parameter estimation, all at a computational cost practical for CPU-based rendering.

In this work, we introduce the Neural Material Adapter (NMA), a hybrid solution that learns a mapping from a broad range of material classes to an analytical target model. Unlike previous hybrid approaches, our model supports general multi-layer material classes, such as configurations consisting of media with specular interfaces [GHZ18] as well as measurements of complex real-world materials [MPBM03, DJ18]. We leverage the Principled BRDF [BS12] as a building block for our target model. While it provides a strong, physics-based prior, we show that, unsurprisingly, it is not expressive enough to model complex layered appearances. Our key insight is that its limitation in capturing such complexity is not due to the underlying microfacet theory, but rather the directional invariance of its parameter space and insufficient non-scalar lobes.

To address these limitations, we propose relaxing and augmenting the Principled BRDF to form a powerful analytical target model. First, we demonstrate that modeling the analytical parameters of the Principled BRDF as a function of incident direction yields a representation that encompasses a vast region of the layered appearance gamut, unattainable by the base model. This relaxation enables capturing effects like directional color shifts and complex highlight distributions observed in multi-layer materials. Second, we show that augmenting this relaxed Principled BRDF with additional colored lobes further expands the appearance gamut of our target model. Note that despite the aforementioned extensions, our model inherits energy conservation and accurate importance sampling from the underlying Principled BRDF.

At the core of our method is a lightweight Multi-Layer Perceptron (MLP) that acts as a translator between complex material model parameters and our expanded analytical target model. We show that our highly-efficient CPU-based inference implementation enables online parameter translation, which avoids any offline baking step commonly required in fully neural approaches. This simplifies the authoring workflow significantly since authoring happens directly in the parameter space of the input appearance model. In addition, texture interpolation is naturally handled by our nonlinear model, which avoids artifacts commonly occurring with latent-space linear interpolation methods. In particular, after training our model on a source material class (e.g. a 3-layer PFMC topology), NMA requires neither per-material optimizations nor texel preprocessing. During shading, our MLP directly translates (potentially textured) source appearance parameters. This is in contrast to existing neural approaches like MetaLayer [GLH*23], which requires offline per-texel preprocessing, Neural PFMC [FWH*22], which requires per-material latent optimization, or per-material training techniques [ZRW*24].

To train our model efficiently, we further propose AvA, a novel Average-vs-Average supervision scheme. While previous neural or hybrid techniques typically rely on point sample supervision, we supervise training to match the mean energy within tabulated bins over the directional domain, effectively decoupling reconstruction accuracy from sampling density. Our experiments show that this approach significantly reduces the amount of required training data and yields superior results on highly glossy materials. Together with the strong inductive bias from our Principled-BRDF-based target model, we achieve remarkably fast and stable training from sparse, often noisy training samples.

In summary, our contributions are:

- Neural Material Adapter, a novel hybrid material model that supports an isotropic 3-layer analytical material class as well as measured MERL data.
- An energy-conserving, importance-sampleable, directionally-varying parametric target model based on the Principled BRDF.
- A novel Average-vs-Average supervision scheme that improves results on highly glossy materials and enables sparse training data sampling.
- Efficient zero-shot online spatially-varying parameter translation compatible with any CPU or GPU-based path tracer.

2. Related Work

Non-neural Material Representations. Traditionally, materials are either represented analytically or obtained from real-world measurements.

Parametric models [AS00, CT82] simulate light-material interactions using physical properties of surfaces, often approximating them with microfacet models. These methods are typically fast when modeling simple materials. However, they can become expensive and require additional parameters when simulating more complex materials such as multilayered surfaces or those with intricate interactions [WW07, JdJM14, GHZ18]. A key advantage of parametric models is that they offer intuitive editing via physically meaningful parameters and exact importance sampling by construction, properties that fully neural approaches sacrifice. The computational cost of layered simulation has motivated dedicated acceleration techniques such as the Gaussian product sampling of Xia et al. [XWHM20], underscoring that the variance and expense of methods like PFMC remain an active concern. Belcour [Bel18] proposes a statistical framework that decomposes layered transport into atomic operators acting on directional moments, yielding compact BSDF lobes without per-material precomputation.

Another approach involves measuring real-world materials. Datasets such as MERL [MPBM03] and RGL [DJ18] contain a wide variety of captured real-world BSDFs. While they provide high realism, importance sampling of such tabulated data often requires significant memory and offers limited flexibility for editing or controlling material properties. Early work has explored fitting parametric models to measured BSDFs through optimization [MPBM03] or data-driven factorizations [TUKK20, SJR18], but these decompositions do not map to the parameter spaces of standard shading models used in production workflows.

Neural BSDF Representations. Neural methods offer high representation capacity, enabling close approximation of complex scattering data. However, this capacity comes at a cost: these methods replace the renderer's native evaluation and sampling routines with neural network inference, requiring custom evaluation plugins or modified rendering pipelines.

Existing neural material representations [HGC*20, KWM*22, ZZW*22] train neural networks to evaluate BSDF values rather than parameters of an analytic model, either from parametric models or measured data [FWH*22, GLH*23]. These models do not target an underlying analytic BSDF and therefore cannot leverage its built-in importance sampling or energy conservation guarantees. Neural methods could in principle generalize across materials given sufficient data and network capacity, but this generalization is expensive: either the evaluation network itself must be large enough to encompass all materials (requiring GPU inference), or a separate encoding network must process each new material before rendering. For instance, MetaLayer [GLH*23] uses a large encoder to predict the weights of a smaller evaluation network; the generalization capacity resides in the encoder rather than the evaluator.

Many neural methods also require preprocessing before rendering can begin. Fan et al. [FWH*22] optimize a 32-dimensional latent vector for each material (10–45 s on an RTX 2080 Ti). MetaLayer [GLH*23] reduces the per-material cost to under one second via its MetaNet encoder, but spatially-varying materials require per-texel precomputation (27–53 s per texture map, 1.7 GB LUT, scaling linearly with resolution).

Importance sampling presents a fundamental challenge for neural BSDF representations. Because these methods predict raw BSDF values rather than parameters of a known analytic distribution, the sampling distribution must be constructed separately and never perfectly matches the evaluation function. Existing solutions range from learned proxy distributions [FWH*22, ZZW*22] to dedicated sampling architectures [XWH*23], all requiring additional neural components on top of the base material model.

More recent work explores alternative neural architectures for this sampling problem: learned reparameterization [WBX*25] and diffusion-based sampling [FBL*24], where the learned sampling distribution remains an approximation of the separate neural evaluation function. Li et al. [LWY*25] avoid the evaluation–sampling split by using a single flow matching network for both sampling and PDF evaluation, though ODE discretization introduces residual inconsistency. Their method also requires per-material training and multi-step ODE integration with no analytic parameter output. Analytic-parameter adapters sidestep this complexity entirely.

Sztrajman et al. [SRRW21] propose a lightweight MLP (2 hidden layers of 21 neurons, 675 weights) per measured BRDF, with importance sampling handled via a learned analytic proxy distribution. Zhou et al. [ZSR*24] extend this with reciprocity enforced architecturally and energy passivity encouraged through a penalty loss.

Zeltner et al. [ZRW*24] present a neural appearance model with view-dependent processing, the most architecturally similar prior work to ours. Their method employs two jointly trained decoders: an evaluation decoder that outputs raw BRDF values and a sam-

pling decoder that outputs analytic lobe parameters. Because these are separate models with different training objectives, the importance sampling distribution only approximates the evaluation function. Their method also requires per-material baking (∼4–5 h on an RTX 4090) with no generalization to unseen materials and a modified shader compiler for integration. Zeltner et al. demonstrate that view-dependent processing is beneficial for material appearance, a motivation we share. However, where their view-dependent evaluator outputs raw neural values that cannot be importance-sampled directly, we output view-dependent analytic parameters that serve both evaluation *and* sampling within a single model, ensuring tight consistency between the two.

BSDF Parameter Adapters. An alternative to learning the full scattering function is to learn a mapping from complex appearance model parameters to the parameters of an analytic BSDF. Because both evaluation and sampling are performed by the analytic target model, these hybrid methods inherit its built-in importance sampling, which closely matches the evaluation function by construction. Furthermore, the strong inductive bias of the physically-based target model (encoding microfacet theory, Fresnel reflectance, energy conservation) enables remarkably compact network architectures, since the MLP only learns the parameter mapping rather than the scattering physics itself.

Wang et al. [WJHY22] (SpongeCake) demonstrate this approach for layered microflake volumes using an MLP (3 hidden layers of 128 neurons) to map layer configuration parameters to analytic single-scattering and Lambertian lobe parameters that approximate multi-scattering. Single scattering is computed analytically, which is possible because their model is restricted to volumetric layers without reflecting or refracting interfaces. The resulting representation inherits importance sampling from the SGGX lobes and requires only one-off class training. However, for spatially-varying materials, the 3×128 network must be evaluated per texel as a precomputation step; at render time, only the analytic lobes are evaluated. Similarly, Yan et al. [YSJR17] use a small MLP (2 hidden layers of 10 neurons, ∼200 parameters) to map fiber scattering parameters to dipole BSSRDF parameters for hair and fur rendering, inheriting sampling from the dipole model. Notably, their MLP evaluates online at each shading point, enabling native support for spatially-varying fiber parameters without precomputation.

While these methods validate the hybrid approach, they are each specialized to a narrow material class: interface-free microflake volumes for SpongeCake and fiber scattering for Yan et al. Neither supports general layered materials with specular interfaces, coated conductors, or arbitrary layer configurations. Our Neural Material Adapter generalizes this hybrid concept to a broad range of material classes, including PFMC configurations with stacked dielectric and conductor layers separated by scattering media, as well as measured materials from the MERL [MPBM03] and RGL [DJ18] datasets, while retaining zero-shot SVBRDF support without precomputation.

We target the production-standard Principled BRDF with view-dependent parameters and colored lobe augmentation, enabling a richer appearance gamut than domain-specific targets while retaining native importance sampling, energy conservation, and compatibility with any renderer that supports the Principled BRDF. After one-off training on the source appearance model, our method performs online parameter estimation at each shading point, accepting any source parameters and textures within the training distribution without precomputation.

Our view dependent parameters do not attempt to represent true Birectional Texture Functions (BTF) like those in [KMX*21] or [RJGW19] but are used to augment the expressiveness of Principled BRDF.

We also show that target-space editing is possible by enabling view-dependent parameter curves or textures as a novel editing modality. Please see the supplemental material for a prototype.

3. Methodology

We now introduce our Neural Material Adapter (NMA) framework. After a brief background, we detail the design of our target model in Section 3.1. In Section 3.2, we describe our proposed AvA training objective. Section 3.3 discusses training data generation from the PFMC model [GHZ18], followed by training details in Section 3.4.

Background A bidirectional scattering distribution function (BSDF) $f_r(\omega_i, \omega_o)$ describes how a surface scatters incident light as a function of the light direction ω_o and the viewing direction ω_i (we follow the Mitsuba [JSR*] convention):

$$f_r(\omega_i, \omega_o) = \frac{dL_o(\omega_i)}{dE_i(\omega_o)}, \tag{1}$$

where L_o is outgoing radiance, and E_i is incident irradiance. Analytic BSDF models introduce a parameter vector $\vec{p}$ controlling material appearance:

$$f_r(\omega_i, \omega_o) = f_{\text{base}}(\omega_i, \omega_o; \vec{p}), \tag{2}$$

This enables spatially varying appearance using a single, principled appearance model.

A powerful instance of such a BSDF model is the PFMC model [GHZ18]. PFMC represents appearance as a physical stack of layers, where light scattering is modeled using numerical Monte-Carlo simulation through multiple layers of participating media, separated by dielectric interfaces. PFMC can model arbitrary layer configurations and features effects beyond reflection such as transmission and anisotropy. In addition, it is fully authorable by the parameter vector $\vec{p}$, which enables texturing of any layer parameter. However, its flexibility and accuracy come at a cost: the Monte-Carlo simulation is prohibitively expensive, especially when dense media layers are involved.

3.1. Target Model Design

We choose the Principled BRDF [BS12], a highly flexible BRDF model targeting production needs, as the building block for our target model. Our choice of the target model is motivated by the widespread use of Principled BRDF-like physically plausible BRDFs in production renderers and the availability of a differentiable implementation of Principled BRDF in Mitsuba. When overfitting the Principled BRDF to a complex layered PFMC material configuration using gradient descent, we observe that it is not able to match the source appearance, see Figure 2 (*single*). This is unsurprising

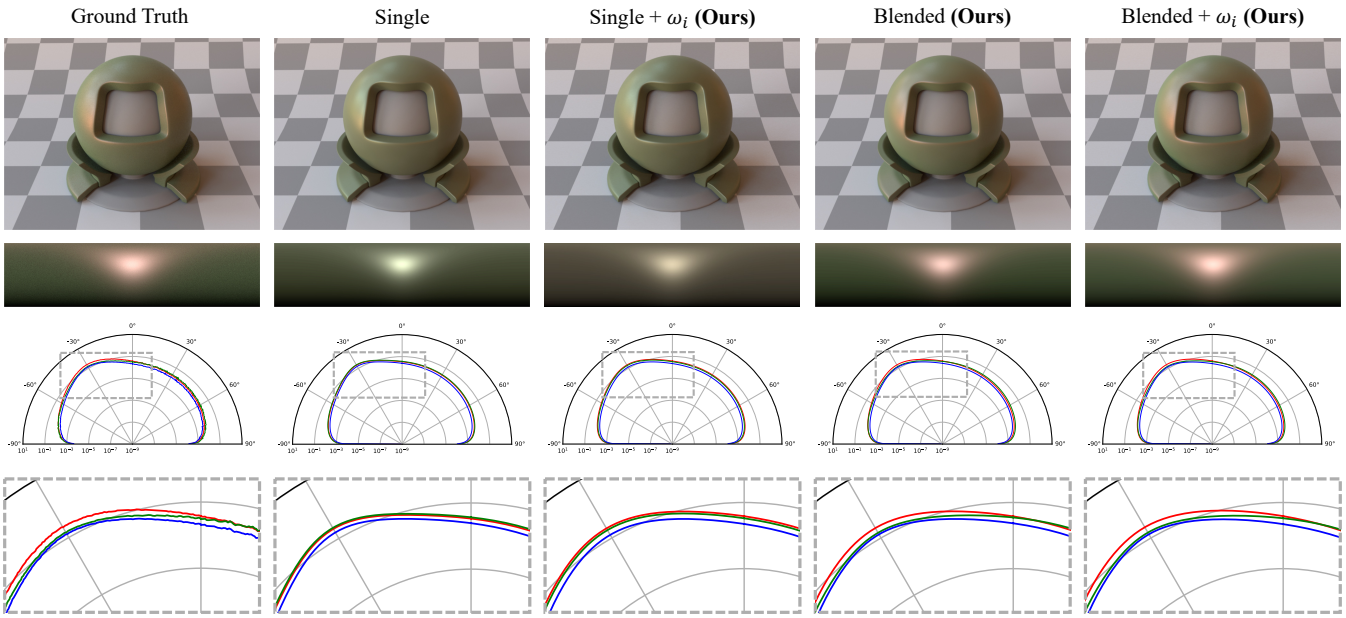

Figure 2: A multi-layered material from Guo et al. model [GHZ18] fitted with our proposed target model. As shown in the lobe plots, our target model can capture the complex view-dependent colored lobe structures of layered materials that would be otherwise impossible to express with the base Principled BRDF.

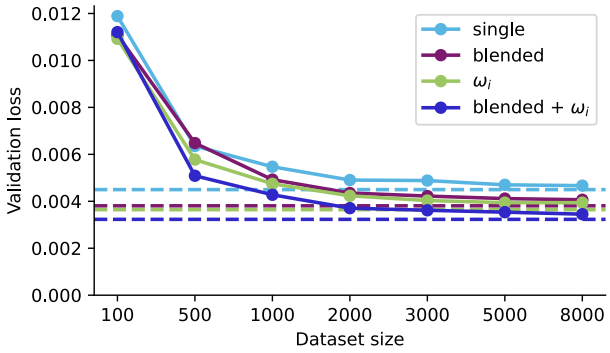

Figure 3: **Quantitative evaluation** of the validation loss (Equation 7) as a function of dataset size computed on our validation set with 89 materials. The dashed lines correspond to the upper bound (overfit on the validation set). Solid lines correspond to models that were trained on datasets with increasing size that do not overlap with the validation set. Our proposed target model (blended $+\omega_i$) fits the data significantly more accurately than the baselines. Also, as the dataset size increases, we observe that each variant approaches its overfitting loss, demonstrating good generalization capability.

given that the Principled BRDF only has a single non-scalar parameter in $\vec{p}$. The reference in Figure 2 (*Ground Truth*) motivates that such layered configurations need at least two colored lobes.

Given this observation, we extend our target model to a blend of *two* Principled BRDF instances, with a learned blend weight α:

$$\alpha\, f_r(\omega_i, \omega_o; \vec{p}_1) + (1-\alpha)\, f_r(\omega_i, \omega_o; \vec{p}_2) = RGB, \tag{3}$$

where $\vec{p}_1$ and $\vec{p}_2$ are Principled BRDF parameters.

While this greatly expands the gamut of our target model, a quantitative evaluation on 89 randomly sampled PFMC configurations reveals that this is still insufficient to match complex multi-layer appearances, especially at grazing angles around silhouettes.

We therefore expand our target model further. In particular, we train a small Multi-layer Perceptron (MLP) to model $(\vec{p}_1, \vec{p}_2, \alpha)$ as a function of ω_i, i.e.,

$$f_r(\omega_i, \omega_o) = f_{\text{base}}\big(\omega_i, \omega_o; \vec{p}_1(\omega_i), \vec{p}_2(\omega_i), \alpha(\omega_i)\big) \tag{4}$$

Depending on the application, we use three to five hidden layers with 32 to 64 neurons each. (3×32) is sufficient for single material overfits and small material libraries, such as MERL. To generalize to thousands of complex multi-layer materials, we train larger models with (5×64) hidden neurons. Figure 4 shows an overview of the model architecture. In Figure 3 we show that this expansion closely matches randomly sampled PFMC configurations quantitatively, and we use this target model throughout the rest of the paper. It is worth noting that our target model still inherits the energy conservation of the base model, i.e.:

$$\forall \omega_i, \int_\Omega f_r(\omega_i, \omega_o) \cos\theta_r d\omega_o \leq 1. \tag{5}$$

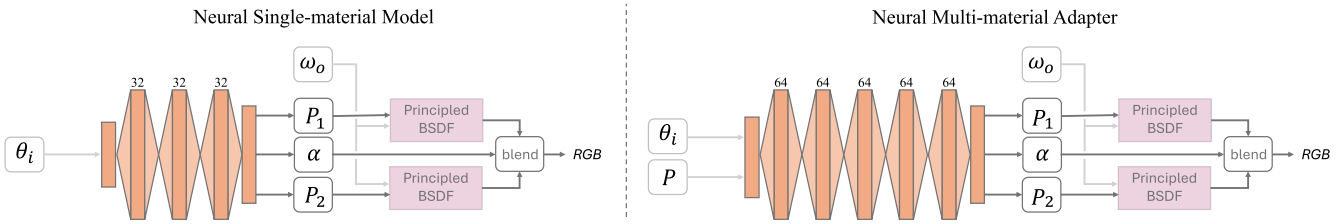

Figure 4: **Neural Material Adapter overview**. We represent complex materials through a reparameterization using a learned blended, directionally dependent, Principled BRDF target model. For a single material overfit (left), a small network is used to predict the parameters p_1 and p_2 of two Principled BRDFs and their blending coefficient α, given the input direction ω_i. On the right, we show how we extend our network to encode multiple materials of the same class by additionally taking the source model parameters P as input and increasing the network capacity.

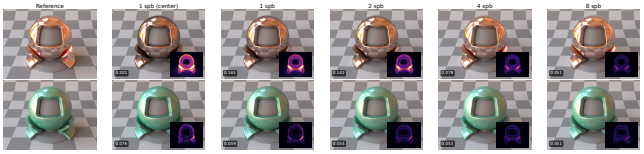

Figure 5: Motivation of our AvA training objective. Training on a uniform grid of point samples (1spb center) results in poor fits even for simple materials such as a rough conductor with low roughness (top row). Complex multi-layered materials (bottom row) face similar problems even with moderately glossy materials due to their complex multi-scattering behavior. Sampling the bin area (columns 3 to 6) improves the fits significantly, whereas more samples per bin increase accuracy up to the point of diminishing return.

However, reciprocity ($f_r(\omega_i,\omega_o) = f_r(\omega_o,\omega_i)$) is violated (but the Principled BRDF itself violates this), and using our model in a bidirectional path tracer will introduce bias.

3.2. AvA: Average-vs-Average Objective

So far, we have discussed our target model and justified its design based on overfitting experiments. For it to be applicable in practice for editing and texturing, our model needs to efficiently and accurately represent any combination of parameters of the source appearance model. For this purpose, we directly feed PFMC parameters P in addition to ω_i to the network (see Figure 4 *right*).

We train our network on an extensive set of randomly sampled PFMC material configurations and supervise on evaluated BRDF values. Since the training data is in a discrete, tabulated form, we found that supervising on point samples results in poor fits (see Figure 5 2nd and 3rd columns). Instead, we propose to sample multiple points inside a 4D bin and average their contributions before computing the training loss. This significantly improves fit accuracy on specular materials and, in particular, multi-layered configurations (see Figure 5). In particular, let $\tilde{g}_b$ be the ground truth bin value for bin b in the 4D table. We then compute the loss function $\mathcal{L}(\tilde{g}_b,\tilde{f}_b)$ with

$$\tilde{f}_b = \frac{1}{N_b}\sum_{k=1}^{N_b} f(\omega_i,\omega_o), \tag{6}$$

where N_b is the number of samples per bin and f is our target model. During training, we randomly sample N_b directions inside the bin b uniformly with respect to solid angle for each sample in the minibatch. Note that we only do one forward pass through the neural network per sample, we only evaluate our target model f multiple times per sample.

Similar to previous work ([SRRW21], [KMH*25]), we compute the loss in the log domain, i.e.

$$\mathcal{L}(y,\hat{y}) = |\log(1+y) - \log(1+\hat{y})|. \tag{7}$$

3.3. PFMC Training Data Generation

We represent our training data using a 4-dimensional tabulation that maps (ω_i,ω_o) pairs to *RGB* values. While response measurements of real-world materials such as MERL [MPBM03] and RGL [DJ18] are represented in tabulated form, simulated models such as PFMC must be sampled explicitly.

We discretize the PFMC BRDF into a 4D table, where each table entry, which we name a *bin*, is indexed by a pair of directions (ω_i,ω_o). We tabulate the 4D domain uniformly with respect to solid angle, which guarantees uniform sample density over the domain. For analytical models, we can simply sample pairs of (ω_i,ω_o) with respect to solid angle to populate the 4D table. PFMC, however, traces rays through the layers and scatters in the layer media. Formally, it is defined as $g(\omega_i) = (\omega_o, t(\omega_i,\omega_o))$, where

$$t(\omega_i,\omega_o) = \frac{f(\omega_i,\omega_o)}{p(\omega_o|\omega_i)} \tag{8}$$

is the local throughput. Note that neither $f(\omega_i,\omega_o)$ nor $p(\omega_o|\omega_i)$ is available in analytical form, since $t(\omega_i,\omega_o)$ is Monte-Carlo estimated during the random walk.

To populate the 4D table with PFMC samples, we sample N random directions ω_i uniformly with respect to solid angle over the hemisphere. For each sample, we evaluate $g(\omega_i)$ and compute the bin average $\tilde{g}_b$ of bin b with

$$\tilde{g}_b \approx \frac{2\pi}{\Omega_b N}\sum_{k=1}^{N_b} t(\omega_i,\omega_o), \tag{9}$$

where Ω_b is the solid angle of bin b and N_b is the number of samples that land in bin b.

Table 1: Sampling distributions for the layered material parameters used in our dataset adapted from [GLH*23]. In contrast to the original setup, we sample the medium extinction coefficient continuously over $[0,1]$. For conductors, we use the complex refractive index parameters (η,κ) sampled over the corpus range, rather than the Schlick approximation. Parameters marked with * are RGB while the rest are scalar values.

Parameter	Symbol	Sampling
Roughness for top interface	α_1	$10^{U(-3,-0.5)}$
Roughness for bottom interface	α_2	$10^{U(-3,0)}$
IOR for dielectric interface	η_1,η_2	$U(1.05,2)$
Conductive interface*	η,κ	corpus range
Single-scattering albedo for medium*	ρ	$1-U(0,1)^2$
Extinction coefficient for medium	σ_t	$U(0,1)$

Our proposed data generation strategy has multiple advantages. Firstly, uniformly sampling the tabulated bin areas denoises the PFMC estimates, making training significantly more stable. Secondly, and more importantly, thanks to the target model prior and our AvA training objective, this data aggregation scheme enables fitting very high-frequency functions using a coarse tabulation. Thirdly, to accurately fit PFMC, fitting on point samples is literally impossible since neither $f(\omega_i,\omega_o)$ nor $p(\omega_o|\omega_i)$ is computable in explicit form. Note that previous work fitting PFMC, e.g. [FWH*22], generated training samples by leveraging the approximations $\tilde{f}(\omega_i,\omega_o)$ and $\tilde{p}(\omega_o|\omega_i)$ provided by [GHZ18], which enables fitting on point samples at the cost of bias. These approximations are mainly provided as variance reduction techniques for next event estimation with multiple importance sampling, but $\tilde{f}$ cannot accurately simulate multiple scattering through the layer stack [GHZ18].

To generate our PFMC dataset, we randomly sample 8000 3-layer parameter configurations similar to [GLH*23]; see Table 1. The topology consists of a rough conductor layer with a homogeneous medium on top, coated with a rough dielectric layer. For each such 3-layer material, we sample 819'200 random directions ω_i to form the 4D table $\{\tilde{g}_i\}$, and we use a table resolution of $(\theta=40,\phi=80)$ per material. We restrict the dataset to isotropic materials, which reduces the dataset size by one dimension to 3D. We refer readers to the supplemental materials for the full derivation and further details about our tabulation and averaging process.

3.4. Training Details

We train our model in PyTorch and implement the target model in Mitsuba 3 [VJ22]. We use the Adam optimizer [KB14] with a learning rate of 1×10^{-5}. We use a batch size of 1000 and sample $N_b=20$ direction pairs per bin for AvA computation. For overfitting a single material, we use a ReduceLROnPlateau learning rate scheduler and the model converges in around 5 minutes on a single RTX 3090. For training our model on the dataset of 8000 PFMC materials, we leverage a OneCycle learning rate scheduler and train the model for two hours on a single RTX 3090.

4. Results

In this section, we validate our method by ablating the design choices that we introduce and provide comparisons with related work.

Experimental Setup Unless stated otherwise, all reported results use a *5×64* network (five hidden layers of 64 neurons each) and are evaluated on our validation set of 89 unseen materials drawn from the PFMC benchmark. The metrics used in our tables are defined as follows:

- **AE** – Absolute Error.
- **MAPE** – Mean Absolute Percentage Error.
- **SMAPE** – Symmetric Mean Absolute Percentage Error.
- **MRSE** – Mean Relative Squared Error.
- **PSNR** – Peak Signal-to-Noise Ratio (dB).
- **SSIM** – Structural Similarity Index.
- **FLIP** – perceptual image-difference metric.

Ablations. We run a quantitative ablation of our Multi-Material Neural Adapter on a validation set of 89 materials generated with PFMC by Guo et al. [GHZ18]. In Figure 3, we compare the quality of SINGLE, BLENDED, ω_i and our complete method BLENDED + ω_i. SINGLE represents a variant where the MLP predicts the parameters for a single Principled BRDF model for each input PFMC parameter set, irrespective of the incident angle ω_i. In BLENDED, the MLP predicts the parameters of two Principled BRDF models and a blend factor, irrespective of the incident angle ω_i. In ω_i, the MLP receives as input the incident direction and, based on this angle, adjusts the parameters of a single Principled BRDF model. Finally, our BLENDED + ω_i model (Ours) combines both proposals, where the MLP receives the incident direction and, based on that, can adjust the parameters of both Principled BRDF models along with their blend factor.

With solid lines, we plot our validation loss in Figure 3 as the number of randomly generated PFMC materials in the training set increases. Please note that the training set of these variants does not include the materials from the validation set. The ablation demonstrates that, compared to the SINGLE multi-material variant, both of our extensions, when used in isolation and combined, yield lower validation error. As all variants use an MLP with the same number of layers and neurons, we believe that the improvements in validation error are due to the proposed extensions of the target model. Furthermore, even with dataset sizes of 3000 or 5000 materials, we can achieve good generalization and approach the quality of variants that were exclusively trained on the validation set (denoted with dotted lines).

We additionally perform qualitative and quantitative evaluations on rendered shader balls from our validation set. In Figure 6, we provide visual comparisons between the four variants of our model and the PFMC ground truth. Overall, we consistently observe that the SINGLE variant underperforms all other variants, whereas the BLENDED + ω_i is consistently on top. This is further supported by the aggregated metrics presented in Table 2.

The main differences between a layered material and the Principled BRDF arise from (a) extra multiscattering and (b) view-dependent effects, where the bottom layer shows through the

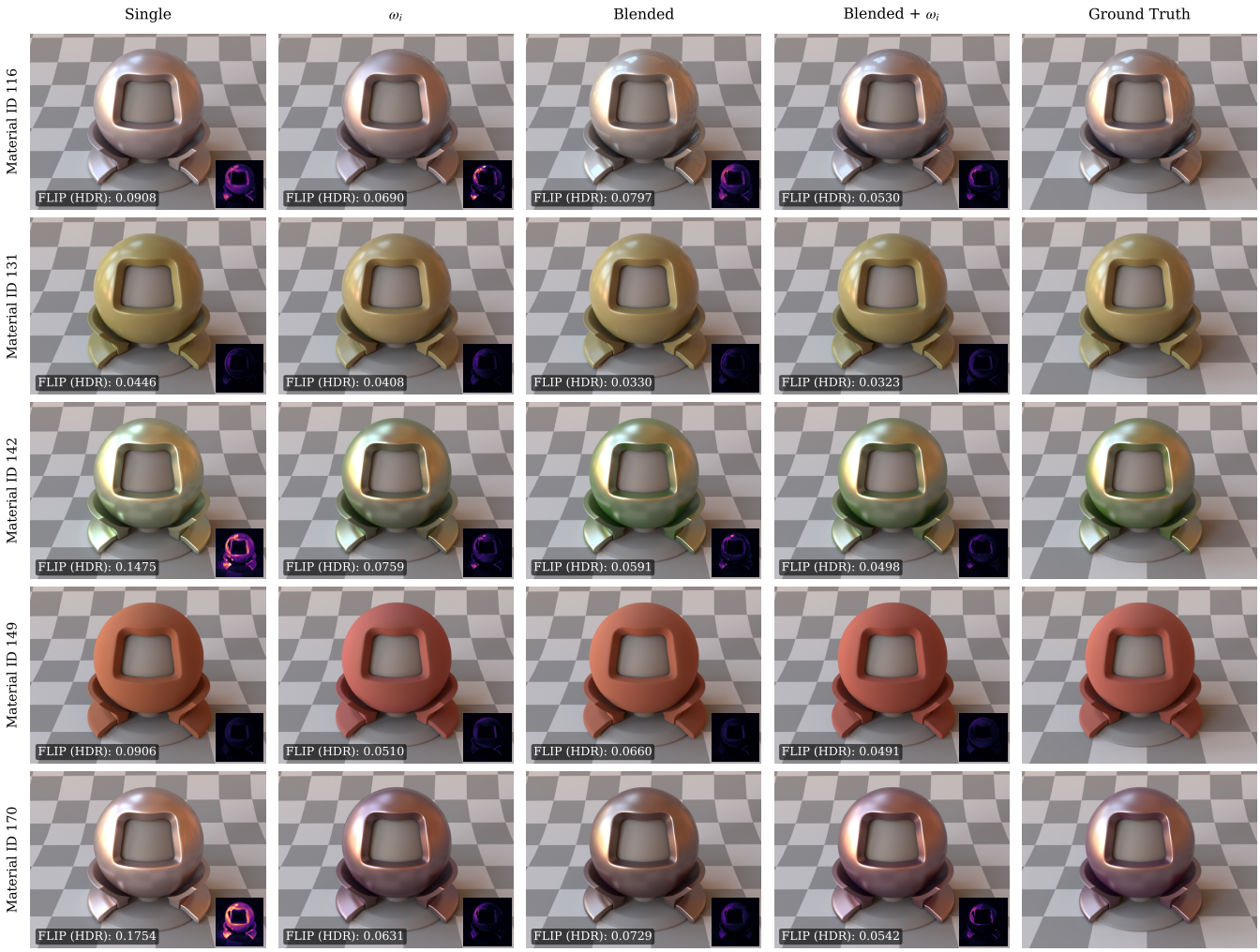

Figure 6: **Qualitative evaluation**. We provide material samples from PFMC [GHZ18] as ground truth and compare against different multi-material variants of our approach. The full set of validation results are included in the supplemental.

Table 2: Comparison of image reconstruction metrics on the material validation set. Lower is better for all metrics except SSIM (higher is better). Best results are in bold.

Method	L1 ↓	L2 ↓	MAPE ↓	SMAPE ↓	MRSE ↓	SSIM ↑	FLIP (HDR) ↓	FLIP (LDR) ↓
Single (5x64)	0.0055	0.0003	2.6285	2.5006	0.0124	0.9954	0.0484	0.0387
ω_i (5x64)	0.0051	0.0002	2.5037	2.3927	0.0106	0.9961	0.0453	0.0367
Blended (5x64)	0.0053	0.0003	2.4593	2.3516	0.0114	0.9957	0.0454	0.0368
Blended + ω_i (5x64)	**0.0050**	**0.0002**	**2.3876**	**2.2867**	**0.0103**	**0.9963**	**0.0432**	**0.0352**

medium and top dielectric layers. The second BRDF allows for approximating multiple colors and multiple scattering, and the θ_i dependency makes them view dependent. This is visible in Figure 6 for Material ID 170 where the underlying copper is visible near normal incidence through the purple-ish medium. A natural question arises as to whether the directional blending of more Principled BRDFs may yield better results. Our ablation (Table 3) indicates that beyond 2 instances, the gains are minimal at the cost of additional complexity and inference time.

Textured Materials Our model trained on 8000 materials naturally supports spatially varying materials with online CPU inference. The learned mapping is smooth in the input space, and the network's per-pixel evaluations are independent: there is no shared cross-pixel state that would alias under sub-pixel parameter variation. The renderer's multi-sample anti-aliasing handles geometric aliasing as usual, but

(a) Constant parameters (b) +textured conductor roughness (c) +textured medium extinction (d) +textured medium albedo

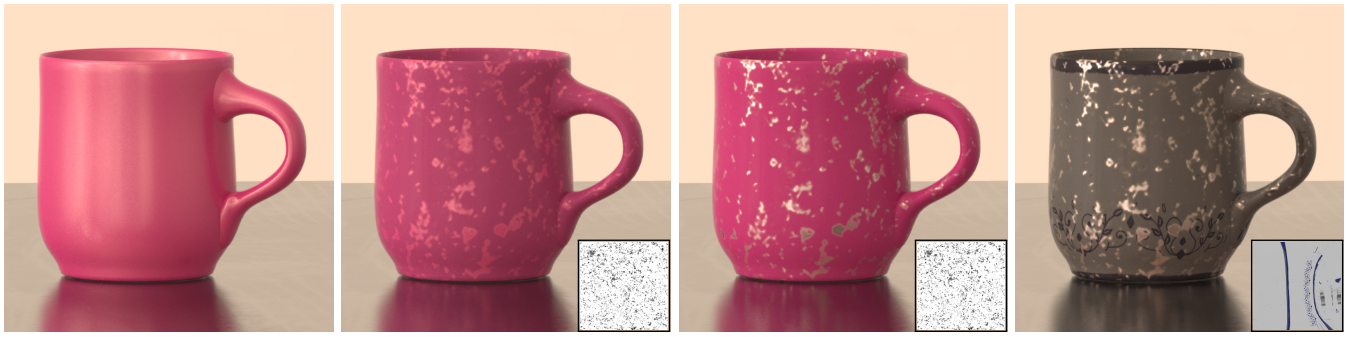

Figure 7: **Layered Textures.** Our model generalizes to spatially varying parameters despite never having been specially trained for it. (a) A basic 3-layered material with a base metal, a colored medium, and a rough dielectric on top. (b) Spatially varying conductor roughness begins to reveal the metal underneath. (c) Adding spatially varying extinction coefficient brings out the metal further. (d) A textured medium albedo on top shows the full layered effect with three input textures working together.

Table 3: This figure shows the diminishing gains from blending more than two Principled BRDFs. We trained three additional variants of our model that directionally blend 3, 4, and 5 Principled BRDFs. Aggregate metrics on 92 unseen validation materials, rendered at 256 spp. Best value in each column is **bold**.

N	$L_1 \downarrow$	$L_2 \downarrow$	MAPE $\downarrow$	SMAPE $\downarrow$	MRSE $\downarrow$	SSIM $\uparrow$	$\text{FLIP}_{\text{HDR}} \downarrow$	$\text{FLIP}_{\text{LDR}} \downarrow$
1	0.01072	0.000478	4.921 %	4.696 %	0.0059	0.9587	0.0707	0.0505
2	0.01007	**0.000341**	**4.674 %**	**4.455 %**	**0.0050**	0.9593	**0.0657**	0.0467
3	0.01101	0.000436	5.192 %	4.852 %	0.0068	**0.9610**	0.0712	0.0502
4	**0.01004**	0.000384	4.723 %	4.486 %	0.0056	0.9586	0.0666	**0.0465**

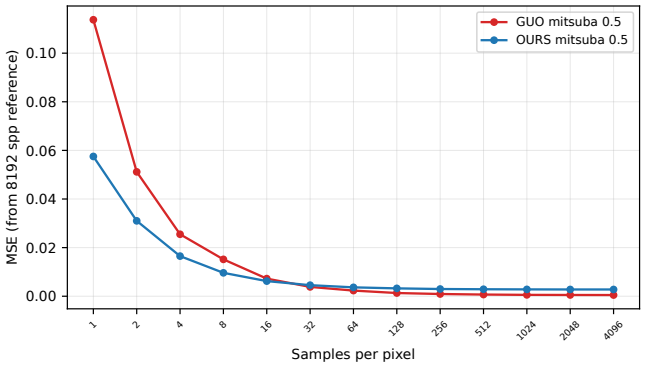

Figure 8: Variance at various SPP for the teaser scene computed vs the 8192 spp reference render. Our method converges faster. Also noticeable is the representation gap where the variance remains slightly higher than the reference at the tails.

no special handling is required for the BRDF parameters themselves. Any input parameter can be textured, and during inference, we evaluate a per-texel directionally blended BRDF. We show some examples of textured materials in Figure 7 and also in Figure 1.

Efficiency Comparison with the Ground Truth Source Model We compare the performance and efficiency of our method in comparison to the source model from [GHZ18] that we used to train our

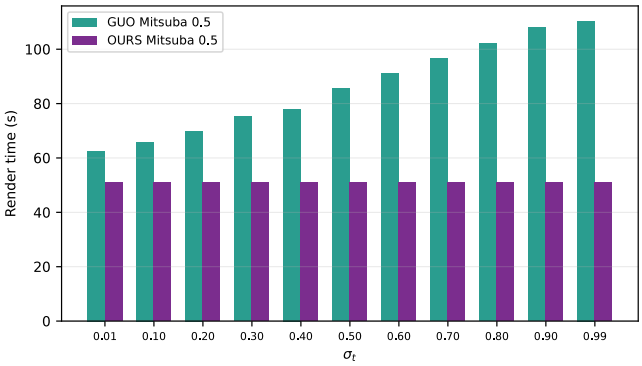

Figure 9: The layered analytical model incurs a penalty in rendering time based on σ_T whereas our target model needs the same inference and render time at same spp.

model. While we use PyTorch and Mitsuba 3 for training, we additionally implement efficient CPU-only implementations for both network inference and target model evaluation in Mitsuba 0.5, which enables a fair comparison to the CPU-only reference implementation of the ground truth [GHZ18]. We used fused Eigen kernels for efficient CPU network inference. The teaser figure and the timings in Figure 9 and Table 4 are based on our CPU implementation. Our method is able to reproduce the 3-layered materials well with less ($\approx 1.2\times$) noise and faster ($\approx 1.72\times$) rendering times per sample. In addition, the layered analytical model incurs a penalty in rendering time based on σ_T, whereas our target model performance is constant with respect to the medium density, as shown in Figure 9.

Comparison with Fan et al. Fan et al. [FWH*22] use a custom CUDA-based pipeline for rendering from their network, along with a custom importance sampling framework. Due to these custom components, direct comparisons on rendered scenes are not straightforward. However, since their decoder weights and training code are publicly available, we were able to evaluate their inference time and errors on BSDF tabulations of our validation set.

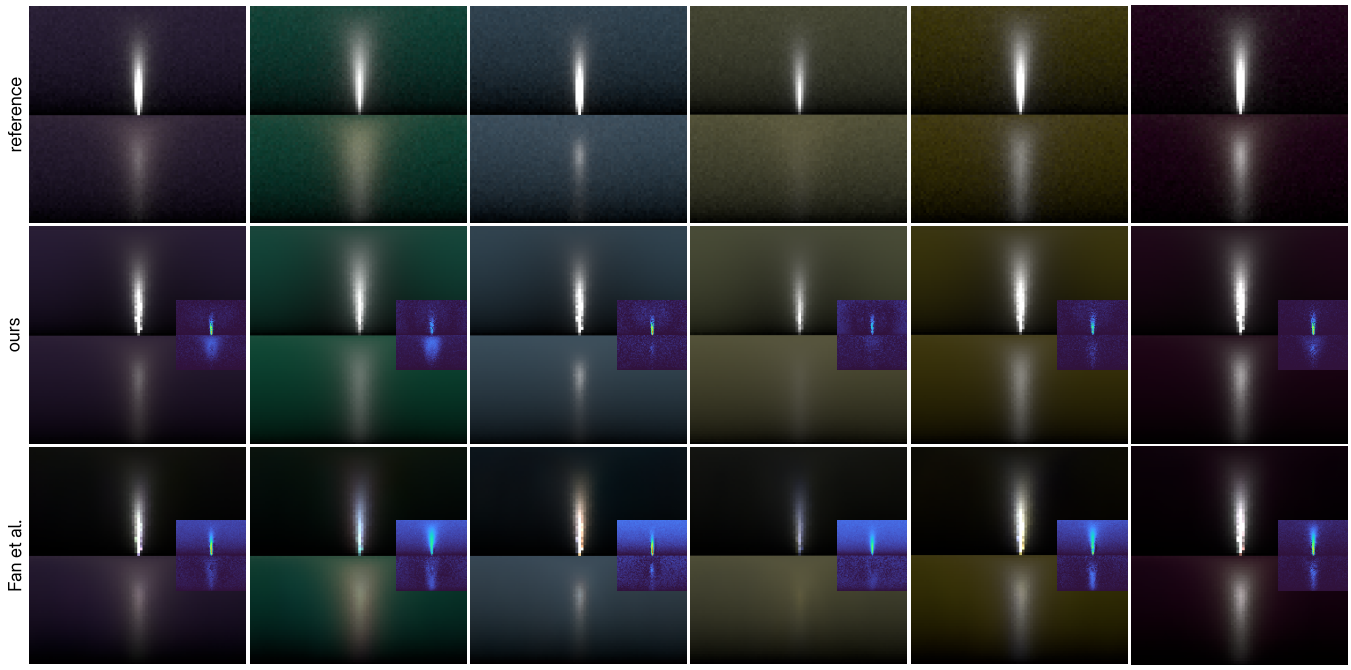

Figure 10: We show two BSDF slices (bottom:normal incidence, top: 70° incidence) for 6 materials from the validation set. Insets show absolute error. Our method (center row) consistently demonstrates significantly lower error, especially at 70° incidence.

Table 4: CPU-only mean render-time and variance comparison on 92 unseen validation materials at 32 spp. Variance is mean squared error against a 2048-spp reference. Efficiency $E = 1/(\text{var} \cdot t)$ (higher is better) is the standard Monte Carlo rendering-efficiency metric.

	Time (s) ↓	Variance ↓	E ↑
[GHZ18] model @ 32 spp	9.90	1.49×10^{-3}	94.5
Our model (5x64) @ 32 spp	**5.75**	$\mathbf{1.19} \times 10^{-3}$	**158.1**
Ratio (ours / [GHZ18])	0.58×	0.80×	**1.67**×

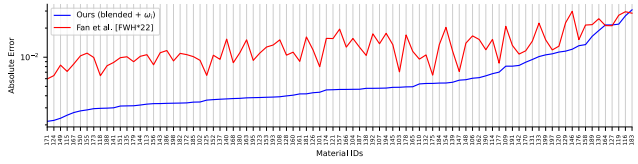

Figure 11: **Absolute error on individual validation materials**. Our model generalizes well and consistently outperforms Fan et al. on our validation set.

In Table 5 we compare our method (BLENDED + ω_i) against the Neural Layered Model of Fan et al. [FWH*22]. We directly optimize latents for each material in our PFMC validation set using the official implementation of [FWH*22]. We evaluate various errors at a resolution of $40 \times 80 \times 40 \times 80$ angular bins, and we uniformly sample 10 locations within each bin. Our model demonstrates significantly higher accuracy for all metrics while also being more than

Table 5: Comparison to Fan et al. on the validation set consisting of 89 materials. We report average numbers evaluated on an RTX 4090 at a resolution of $40 \times 80 \times 40 \times 80$, and we compute 10 samples per bin.

	AE	**PSNR**	**SMAPE**	**Val. Loss**	**Time**
Ours	0.0065	45.5	0.100	0.0037	3.76s
Fan et al.	0.0130	39.7	0.257	0.0100	99.78s

25 times faster when evaluated on the same GPU. This significant speedup is due to our compact architecture: we use a lightweight MLP with (5×64) hidden neurons (see Figure 4), whereas the decoder in [FWH*22] contains ≈ 1 million parameters. Note that this comparison does not include the benefits from our built-in importance sampling, since [FWH*22] do not provide the code and weights of their importance sampling model.

In Figure 11, we expand the aggregated values from Table 5 with a plot of the absolute error for each material in our validation set. We consistently outperform [FWH*22] throughout our validation set with the exception of a handful of cases involving challenging layer configurations, where we achieve a similar error rate.

In Figure 10, we show a qualitative comparison on rendered slices from 6 materials in our validation set. We observe good visual matches between our method and the reference, especially at grazing angles, where the Neural Layered model struggles to capture color variation outside of highlights, while also producing color artifacts within them.

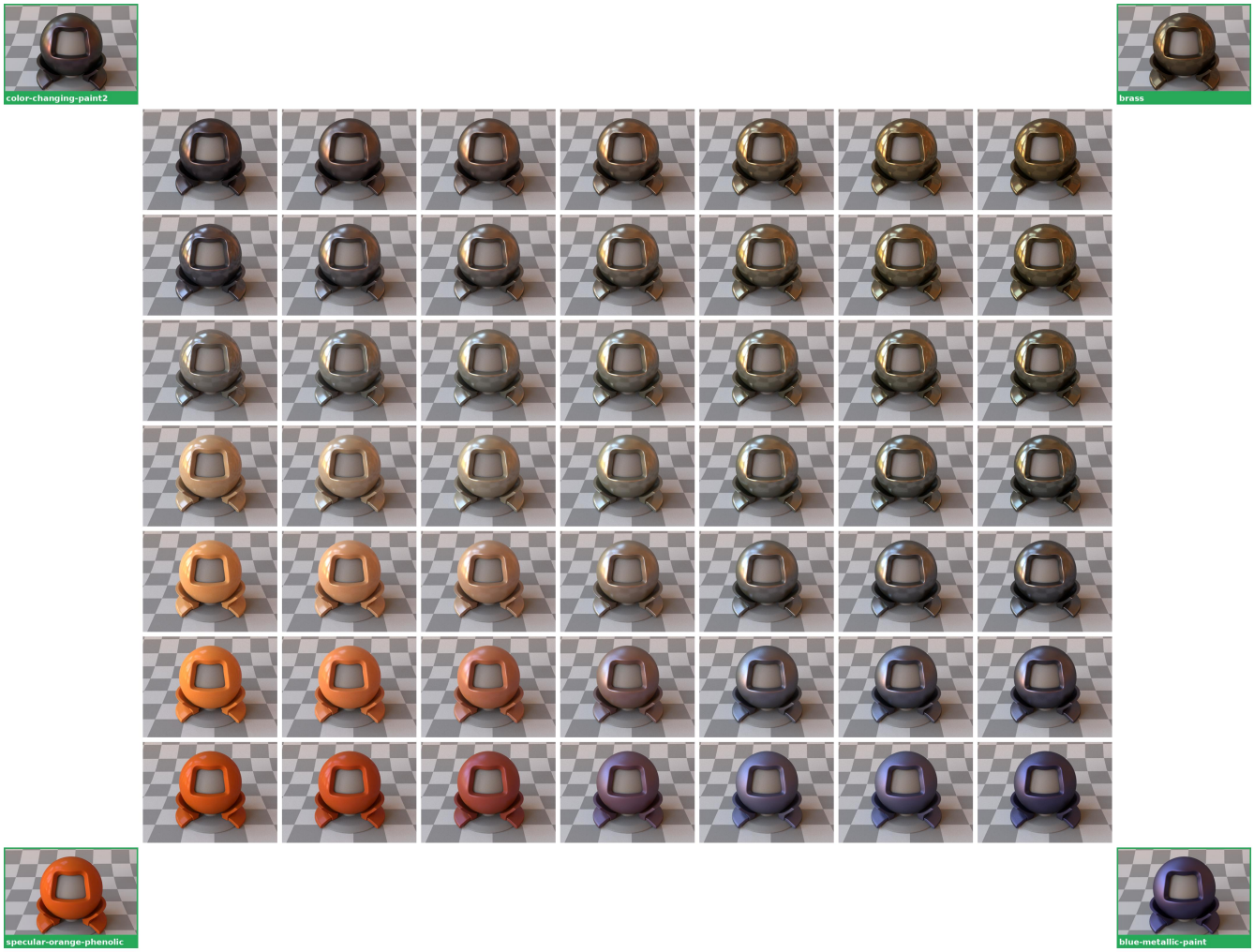

Figure 12: MERL Material interpolation in the learned material latent space. Corner cells (green-outlined, labelled with the MERL material names) are ground-truth renders; the 7×7 interior is produced by bilinearly interpolating the four corner embedding vectors in our compact 3×32 multi-material MLP (7,159 parameters) and decoding them to a wi-dependent blended principled BRDF. Smooth, plausible intermediate appearances emerge across the entire grid. The 4 corner renders are rendered from tabulated data as references, nearest corner is the fit to the references.

Evaluation on MERL To show that our method is not restricted to the PFMC source model, we trained a smaller, (3×32) parameter BLENDED + ω_i model on the 100-material MERL corpus [MPBM03]. As Figure 12 shows, our model is able to fit complex-to-represent MERL materials like *color* − *changing* − *paint*2. In addition, our method learns a smooth latent space that enables the creation of new plausible materials through latent space interpolation. We use PyTorch's `nn.Embedding` for learning the latent space.

Discussion We limit our experiments and evaluations on PFMC to isotropic, reflection-only materials in a fixed 3-layer topology. Note that this is not a fundamental limitation of the approach, and single-material overfits on anisotropic materials demonstrate that our target model is capable of representing such effects (Figure 13). Additionally, we demonstrate the generalization of our approach to other source materials, such as the measured MERL dataset. The directional dependency of our model does violate reciprocity. For applications that require reciprocity, the blended-only variant remains reciprocal and is included as an ablation/design point. For production forward path tracing, exact reciprocity is typically less critical than positivity, energy control, and robust sampling.

5. Conclusion

We presented the Neural Material Adapter, a hybrid approach that uses a lightweight MLP to perform online parameter translation from complex source appearance models to a production-standard Principled BRDF. By allowing the analytic parameters to vary with view direction and augmenting the target with colored lobes, NMA

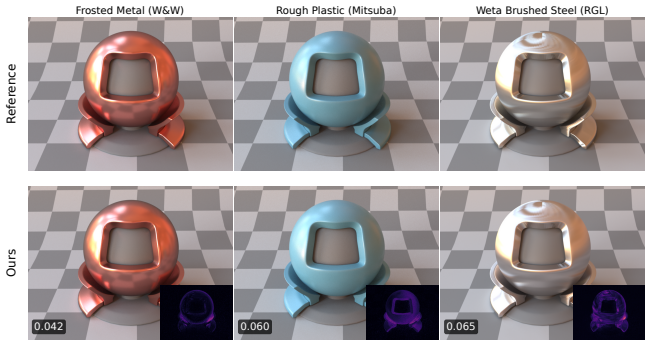

Figure 13: **Examples**. Our model is capable of representing materials from a variety of sources. Here we show overfits for analytical models for coating, a limited layered model rough plastic and a measured material depicting anisotropy along with the FLIP metric.

expands the appearance gamut of the Principled BRDF to capture layered effects inaccessible to constant-parameter assignments. Because evaluation and sampling are both performed by the analytic target model, NMA inherits built-in importance sampling and energy conservation by construction. After one-off class training, the adapter performs zero-shot online parameter estimation at each shading point, requiring no per-material optimization, no latent codes, and no texture baking, while authoring remains in the source parameter space. Our Average-vs-Average supervision scheme further enables robust training from noisy Monte Carlo sources.

Scaling to deeper layer stacks could potentially be addressed by adopting a recursive layer composition strategy analogous to Fan et al. [FWH*22] and adding lobes to the target model to accommodate the richer scattering of deeper configurations, which we consider for future work.

More broadly, NMA demonstrates that the inductive bias of physically-based analytic models provides sufficient structure for compact MLPs to learn the parameter mapping rather than the scattering physics itself. Unlike prior neural material adapters such as SpongeCake [WJHY22] and Yan et al. [YSJR17] that target narrow material classes, NMA generalizes across layered configurations and measured materials while integrating directly into production workflows through the popular Principled BSDF interface. Where fully neural representations must reconstruct both the directional distribution and a compatible sampling scheme from scratch, requiring large networks, GPU inference, and approximate importance sampling, NMA offloads all of this to the analytic target model for free. This generality, combined with native importance sampling and efficient CPU inference, suggests that leveraging analytic priors is a promising direction for practical neural material representations.

Acknowledgements

We thank the anonymous reviewers for their constructive feedback. We are grateful to Brent Burley and André Mazzone for valuable discussions on the Principled BRDF and layered materials, and to Violaine Fayolle for providing 3D models and textures.

References

[AS00] ASHIKHMIN M., SHIRLEY P.: An Anisotropic Phong BRDF Model. *Journal of Graphics Tools 5*, 2 (Jan. 2000), 25–32. URL: http://www.tandfonline.com/doi/abs/10.1080/10867651.2000.10487522, doi:10.1080/10867651.2000.10487522. 3

[Bel18] BELCOUR L.: Efficient rendering of layered materials using an atomic decomposition with statistical operators. *ACM Transactions on Graphics 37*, 4 (Aug. 2018), 1–15. URL: https://dl.acm.org/doi/10.1145/3197517.3201289, doi:10.1145/3197517.3201289. 2, 3

[BS12] BURLEY B., STUDIOS W. D. A.: Physically-based shading at disney. In *Acm siggraph* (2012), vol. 2012, vol. 2012, pp. 1–7. 2, 4

[CT82] COOK R. L., TORRANCE K. E.: A Reflectance Model for Computer Graphics. *ACM Transactions on Graphics 1*, 1 (Jan. 1982), 7–24. URL: https://dl.acm.org/doi/10.1145/357290.357293, doi:10.1145/357290.357293. 3

[DJ18] DUPUY J., JAKOB W.: An adaptive parameterization for efficient material acquisition and rendering. *ACM Transactions on Graphics 37*, 6 (Dec. 2018), 1–14. URL: https://dl.acm.org/doi/10.1145/3272127.3275059, doi:10.1145/3272127.3275059. 2, 3, 4, 6

[FBL*24] FU Z., BELHE Y., LU H., WU L., XU B., LI T.-M.: BSDF importance sampling using a diffusion model. In *SIGGRAPH Asia 2024 Conference Papers* (New York, NY, USA, 2024), SA '24, Association for Computing Machinery. URL: https://doi.org/10.1145/3680528.3687684, doi:10.1145/3680528.3687684. 3

[FWH*22] FAN J., WANG B., HASAN M., YANG J., YAN L.-Q.: Neural Layered BRDFs. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings* (Vancouver BC Canada, Aug. 2022), ACM, pp. 1–8. URL: https://dl.acm.org/doi/10.1145/3528233.3530732, doi:10.1145/3528233.3530732. 2, 3, 7, 9, 10, 12

[GHZ18] GUO Y., HAŠAN M., ZHAO S.: Position-free monte carlo simulation for arbitrary layered BSDFs. *ACM Transactions on Graphics 37*, 6 (Dec. 2018), 1–14. URL: https://dl.acm.org/doi/10.1145/3272127.3275053, doi:10.1145/3272127.3275053. 1, 2, 3, 4, 5, 7, 8, 9, 10

[GLH*23] GUO J., LI Z., HE X., WANG B., LI W., GUO Y., YAN L.-Q.: MetaLayer: A Meta-Learned BSDF Model for Layered Materials. *ACM Transactions on Graphics 42*, 6 (Dec. 2023), 1–15. URL: https://dl.acm.org/doi/10.1145/3618365, doi:10.1145/3618365. 2, 3, 7

[HGC*20] HU B., GUO J., CHEN Y., LI M., GUO Y.: DeepBRDF: A Deep Representation for Manipulating Measured BRDF. *Computer Graphics Forum 39*, 2 (May 2020), 157–166. URL: https://onlinelibrary.wiley.com/doi/10.1111/cgf.13920, doi:10.1111/cgf.13920. 3

[JdJM14] JAKOB W., D'EON E., JAKOB O., MARSCHNER S.: A comprehensive framework for rendering layered materials. *ACM Transactions on Graphics 33*, 4 (July 2014), 1–14. URL: https://dl.acm.org/doi/10.1145/2601097.2601139, doi:10.1145/2601097.2601139. 2, 3

[JSR*] JAKOB W., SPEIERER S., ROUSSEL N., MERLIN NIMIER-DAVID, VICINI D., ZELTNER T., NICOLET B., CRESPO M., LEROY V., ZHANG Z.: Mitsuba 3 - a retargetable forward and inverse renderer. URL: https://www.mitsuba-renderer.org/. 1, 4

[KB14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 7

[KMH*25] KAVOOSIGHAFI B., MANTIUK R., HAJISHARIF S., MIANDJI E., UNGER J.: Perceived quality of brdf models. *Computer Graphics Forum 44* (07 2025). doi:10.1111/cgf.70162. 6

[KMX*21] KUZNETSOV A., MULLIA K., XU Z., HAŠAN M., RAMAMOORTHI R.: Neumip: Multi-resolution neural materials. *Transactions on Graphics (Proceedings of SIGGRAPH) 40*, 4 (July 2021). 4

[KWM*22] KUZNETSOV A., WANG X., MULLIA K., LUAN F., XU Z., HASAN M., RAMAMOORTHI R.: Rendering Neural Materials on Curved Surfaces. In *ACM SIGGRAPH 2022 Conference Proceedings* (New York, NY, USA, July 2022), SIGGRAPH '22, Association for Computing Machinery, pp. 1–9. URL: https://dl.acm.org/doi/10.1145/3528233.3530721, doi:10.1145/3528233.3530721. 2, 3

[LWY*25] LI Z., WANG Z., YANG J., HAŠAN M., WANG B.: PureSample: Neural materials learned by sampling microgeometry, 2025. URL: https://arxiv.org/abs/2508.07240, arXiv:2508.07240. 3

[MPBM03] MATUSIK W., PFISTER H., BRAND M., MCMILLAN L.: A data-driven reflectance model. *ACM Transactions on Graphics 22*, 3 (July 2003), 759–769. URL: https://dl.acm.org/doi/10.1145/882262.882343, doi:10.1145/882262.882343. 2, 3, 4, 6, 11

[RJGW19] RAINER G., JAKOB W., GHOSH A., WEYRICH T.: Neural btf compression and interpolation. *Computer Graphics Forum (Proceedings of Eurographics) 38*, 2 (Mar. 2019). 4

[SJR18] SUN T., JENSEN H. W., RAMAMOORTHI R.: Connecting measured BRDFs to analytic BRDFs by data-driven diffuse-specular separation. *ACM Transactions on Graphics 37*, 6 (Dec. 2018), 1–15. URL: https://dl.acm.org/doi/10.1145/3272127.3275026, doi:10.1145/3272127.3275026. 3

[SRRW21] SZTRAJMAN A., RAINER G., RITSCHEL T., WEYRICH T.: Neural BRDF Representation and Importance Sampling. *Computer Graphics Forum 40*, 6 (Sept. 2021), 332–346. URL: https://onlinelibrary.wiley.com/doi/10.1111/cgf.14335, doi:10.1111/cgf.14335. 2, 3, 6

[TUKK20] TONGBUASIRILAI T., UNGER J., KRONANDER J., KURT M.: Compact and intuitive data-driven BRDF models. *The Visual Computer 36*, 4 (Apr. 2020), 855–872. URL: https://doi.org/10.1007/s00371-019-01664-z, doi:10.1007/s00371-019-01664-z. 3

[VJ22] VICINI D., JAKOB W.: Mitsuba 3: A retargetable forward and inverse renderer. https://mitsuba.readthedocs.io/, 2022. https://github.com. 7

[WBX*25] WU L., BI S., XU Z., TAN H., ZHANG K., LUAN F., LU H., RAMAMOORTHI R.: Neural BRDF importance sampling by reparameterization. In *Proceedings of SIGGRAPH 2025* (New York, NY, USA, 2025), SIGGRAPH Conference Papers '25, Association for Computing Machinery. URL: https://doi.org/10.1145/3721238.3730679, doi:10.1145/3721238.3730679. 3

[WJHY22] WANG B., JIN W., HAŠAN M., YAN L.-Q.: Spongecake: A layered microflake surface appearance model. *ACM Transactions on Graphics (TOG) 42*, 1 (2022), 1–16. 2, 4, 12

[WW07] WEIDLICH A., WILKIE A.: Arbitrarily layered micro-facet surfaces. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia* (Perth Australia, Dec. 2007), ACM, pp. 171–178. URL: https://dl.acm.org/doi/10.1145/1321261.1321292, doi:10.1145/1321261.1321292. 2, 3

[XWH*23] XU B., WU L., HASAN M., LUAN F., GEORGIEV I., XU Z., RAMAMOORTHI R.: NeuSample: Importance Sampling for Neural Materials. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings* (Los Angeles CA USA, July 2023), ACM, pp. 1–10. URL: https://dl.acm.org/doi/10.1145/3588432.3591524, doi:10.1145/3588432.3591524. 2, 3

[XWHM20] XIA M. M., WALTER B., HERY C., MARSCHNER S.: Gaussian product sampling for rendering layered materials. *Computer Graphics Forum 39*, 1 (2020), 420–435. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13883, doi:10.1111/cgf.13883. 3

[YSJR17] YAN L.-Q., SUN W., JENSEN H. W., RAMAMOORTHI R.: A bssrdf model for efficient rendering of fur with global illumination. *ACM Transactions on Graphics (TOG) 36*, 6 (2017), 1–13. 2, 4, 12

[ZRW*24] ZELTNER T., ROUSSELLE F., WEIDLICH A., CLARBERG P., NOVÁK J., BITTERLI B., EVANS A., DAVIDOVIČ T., KALLWEIT S., LEFOHN A.: Real-Time Neural Appearance Models. *ACM Transactions on Graphics 43*, 3 (June 2024), 1–17. URL: http://arxiv.org/abs/2305.02678, doi:10.1145/3659577. 2, 3

[ZSR*24] ZHOU C., SZTRAJMAN A., RAINER G., ZHONG F., GOKBUDAK F., GUO Z., XIA W., MANTIUK R., OZTIRELI C.: Physically based neural bidirectional reflectance distribution function, 2024. URL: https://arxiv.org/abs/2411.02347, arXiv:2411.02347. 3

[ZZW*22] ZHENG C., ZHENG R., WANG R., ZHAO S., BAO H.: A Compact Representation of Measured BRDFs Using Neural Processes. *ACM Transactions on Graphics 41*, 2 (Apr. 2022), 1–15. URL: https://dl.acm.org/doi/10.1145/3490385, doi:10.1145/3490385. 2, 3