

A Hybrid Neural-Microfacet BRDF Model for Real-Time Rendering

Louis De Oliveira^{1,2}, Anastasia Karpova¹, Georges Nader¹, Antoine Houdard¹, Pierre Mézières², Damien Rioux-Lavoie¹, Romain Pacanowski²

¹Ubisoft La Forge, ²Inria

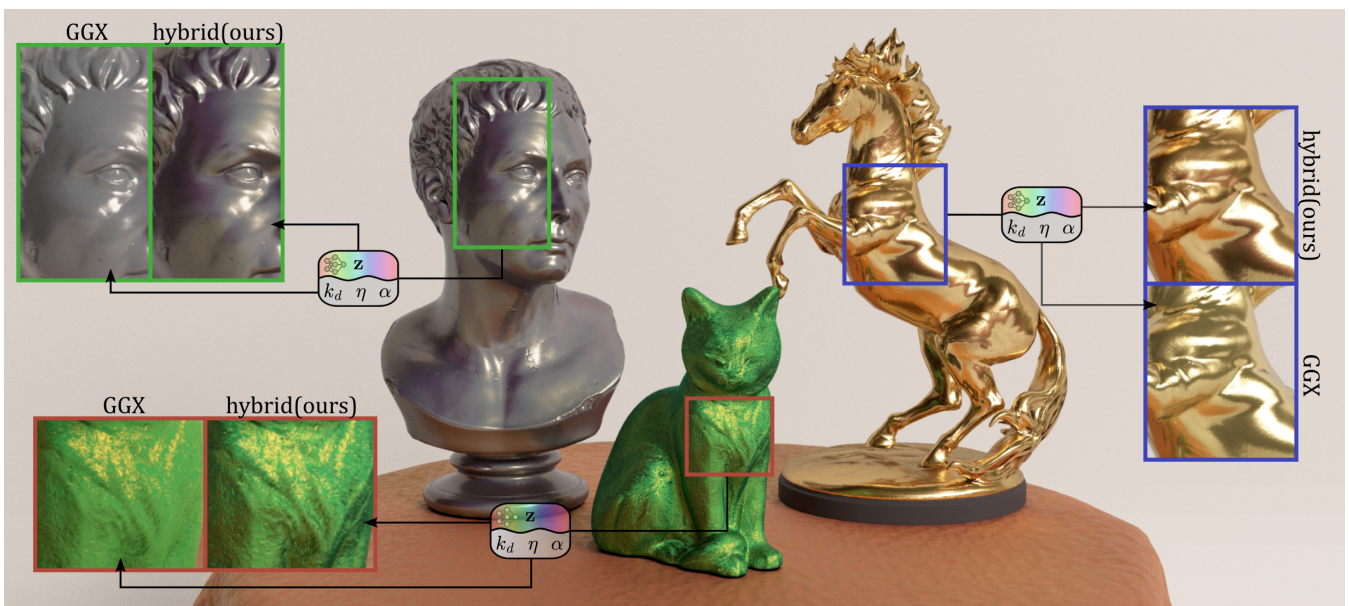


Figure 1: Our hybrid model is fitted from measured BRDFs and represented as a compact set of microfacet parameters (k_d, η, α) and a low-dimensional latent code \mathbf{z} shared across a single neural network. For each material, the neural correction improves upon the microfacet model alone, capturing subtle appearance effects, while the analytical component remains a faithful and useful approximation. From left to right irid_flake_paint2 on the bust, cc_green_malachite on the cat and aniso_metallic_paper_gold on the horse from the RGL database [DJ18a].

Abstract

Over the past decade, microfacet-based BRDF models have formed the foundation of real-time rendering pipelines. Despite their widespread use, they often fail to reproduce subtle appearance effects arising from complex light–surface interactions, which have led to the emergence of specialized physics-based models for specific optical phenomena (e.g., diffraction, iridescence, multilayers). Although more accurate, these models lose versatility and lack performance for real-time rendering. Recently introduced, neural models have demonstrated their ability to approximate BRDF reference data coming from measurements, simulations, or even complex shading networks. However, most current neural models require relatively large networks, making them costly for real-time rendering. In this paper, we introduce a hybrid model that combines a GGX-type microfacet model and a neural model to leverage the best features of both representations. The neural component corrects the appearance approximated by the microfacet component, allowing much smaller network than in existing neural models. We show that, at identical memory cost, our model approximates measurements better than state-of-the-art neural models for a low evaluation overhead compared to a microfacet-based model. Furthermore, our hybrid model remains easily editable by artists and benefits from an important sampling scheme, making it attractive for both offline and real-time rendering.

CCS Concepts

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

1. Motivation

The appearance of an object emerges from the interaction between the light and its surface. Modeling it accurately is at the core of computer graphics and remains an active research topic. The reflection of the light by the surface is described by a physical quantity known as the Bidirectional Reflectance Distribution Function (BRDF). In computer graphics, especially in a real-time context, an ideal BRDF representation should satisfy the following properties:

- (i) *expressiveness*, to represent a large number of materials,
- (ii) *robustness* when fitting measurements or simulated data,
- (iii) *fast evaluation* with importance sampling support,
- (iv) *low memory impact* for stored parameters,
- (v) *editability* through meaningful parameters.

Over the past decade, microfacet-based BRDF models have formed the foundation of real-time rendering pipelines [AMHH*18]. Despite their widespread use, they often fail to reproduce subtle appearance effects arising from complex light–surface interactions. This has motivated specialized physics-based models for specific optical phenomena (e.g., diffraction, iridescence, multilayers). Although more accurate, these models lose versatility and generally lack performance for real-time rendering. Recently, neural models have shown strong ability to approximate BRDF reference data coming from measurements, simulations, or complex shading networks. However, most current neural models require relatively large networks, making them costly for real-time rendering. Furthermore, parameters of a neural model are not meaningful in terms of appearance control, which makes them less suitable for authoring compared to analytical BRDF formulations.

In this paper, we introduce a hybrid model that combines an analytical microfacet-based model and a neural model to leverage the best features of each part. We then demonstrate that such a model has good characteristics regarding the above-mentioned requirements. The neural component acts as a corrective term, refining the output of the analytic BRDF to capture appearance effects that are otherwise difficult to reproduce. To keep memory tight (iv), we enforce the learning of a common neural network for a whole collection of measured BRDFs, and only differentiate the correction term through a low-dimensional latent space as input of the network. Furthermore, we propose a joint training strategy that learns the parameters of both the analytic and neural components simultaneously on the whole BRDF collection (ii), and demonstrate high-quality reconstruction (i) of many measured BRDFs from three databases (MERL [MPBM03a], UTIA [FV14], RGL [DJ18b]). We also enforce the reconstruction of the microfacet component to be close to the reference data during the training, implying that the network only learns residual effects rather than duplicating the base model. Restricting the network to predict residuals allows us to use a shallow multilayer perceptron (MLP) that remains small and computationally efficient (iii), enabling high visual fidelity with minimal computational overhead, while still allowing the parameters of the analytic BRDF to be changed (v). The reconstruction accuracy achieved by our model matches existing neural approaches while relying on a significantly smaller neural network, and our model can even outperform them when operat-

ing at comparable network sizes. As a result, it delivers clear improvements over standard analytical BRDF models while preserving real-time performance.

In the following, we review the relevant previous work on BRDF representations (Section 2), then introduce our hybrid model and the joint fitting procedure (Section 3). We then demonstrate its expressiveness capabilities by approximating a large number of real-world measurements (Section 4). More precisely, we evaluate our model in terms of BRDF reconstruction error and rendering error, comparing it to the neural state-of-the-art approach [ZRW*24] (Sections 4.1 and 4.3), and we perform ablation studies to validate our formulation (Section 4.2). Finally, we discuss the advantages of our hybrid formulation both for real-time rendering performance and intuitive authoring in Section 5 and conclude by discussing limitations and future directions (Section 6).

2. Related Work

In the context of real-time rendering, building a representation that simultaneously satisfies the five criteria defined in the previous section remains particularly challenging. In the following, we review the main families of BRDF representations: tabulated approaches, analytic representations, non-parametric models, and neural ones.

Tabulated BRDFs are obtained thanks to dense measurements (e.g., [MPBM03b]) or long light transport simulation (e.g., [GHZ18, GGN20, YXW*23]) and permit very high quality reconstruction and fast evaluation. However, they are generally impractical in production rendering due to prohibitive memory requirements. A more practical approach regarding memory cost is to model the BRDF with a parametric function.

Single-Lobe BRDF Models are most commonly formulated using the microfacet framework, introduced to the Computer Graphics community by Cook and Torrance [CT82]. This framework considers that the reflectance of a surface can be modeled as a collection of small Fresnel mirrors (i.e., microfacets) whose orientations are statistically defined by a distribution term (cf. [WMLT07]). Most of the time, the microfacet representation is supplemented by a Lambertian term. Although very efficient in terms of evaluation for real-time rendering [LDR14] and easy to manipulate through its 7 or 8 parameters, the representation remains inherently limited in terms of expressiveness (cf. [NDM05]). To overcome these limitations, different types of extensions have been proposed, such as correction for multiple-scattering between the microfacets (cf. [HHdD16, KC17, XH18]) or to encapsulating wave phenomena (e.g., [BB17, FBP25, HP17]). Although these approaches significantly enrich the appearance of materials, they are usually effect-specific and increase both the number of parameters and the overall complexity of handling different materials. Another approach to augment expressiveness is to consider multiple-lobe parametric models.

Multiple-lobe BRDF Models are either obtained empirically (e.g., [Bur12, LFTG97]) or by modeling the surface as a stack of parallel interfaces or media layers to obtain so-called layered BRDFs. The Weidlich and Wilkie [WW07] model, optimized later by Elek [EP10], was one of the first to provide an evaluation of the BRDF resulting from multiple layers. These models, which do

not take fully into account scattering effects, are improved by the recent introduction of statistical models [dB22, Bel18, YYTM19, WB20, RCL21]. Despite these advances, the use of layered BRDF models remains limited since they require high memory demands (each layer requires its own set of parameters). In practice, to balance quality, performance, and storage costs, real-time systems typically restrict the number and type of layers that can be represented. Moreover, although these models significantly increase the range of achievable appearances, the large number of interdependent parameters and the indirect relationship between model parameters and perceived appearance make intuitive control difficult. In particular, layered BRDFs require an explicit material decomposition that is hard to define and even harder to adjust a posteriori, as modifications to one layer propagate non-linearly through the entire model. Furthermore, as shown by Ngan [NDM05], multiple-lobe models tend to be numerically unstable during the fitting process. For this reason, the aforementioned layered BRDF models are rarely tested against real-world BRDF measurements. Aside from parametric representations, non-parametric models have also been studied to represent BRDFs.

Non-parametric Models overcome physical constraints and seek and consider the BRDF as a signal defined on the hemisphere that can be projected into basis functions (e.g., Spherical harmonics [KSS02, SBN15], wavelets [SS95, CBP07], Fourier basis [JdJM14, ZJ18]) or reconstructed after a PCA on a database (cf. [MPBM03b]). Other approximating approaches use radial basis functions [ZERB05], Spherical Gaussians [WRG*09, XSD*13], Zernike polynomials [KDS96], rational functions [PSCS*12], Gaussian Mixtures [CBP21], tensor product [BOK11] or dictionaries [TUGM22]. These methods can represent all types of materials very precisely, as long as dense measurements or simulations are provided as input. Their main limitations are their memory cost, because it grows quadratically with the specularity [MTR08] of the materials, and their lack of editing capabilities.

Neural methods have emerged as a powerful alternative to analytic BRDF models, predicting reflectance directly using neural networks. Such methods are effective in modeling even complex and layered materials [ZRW*24, FWH*22] and can also work as compression for measured materials [HGC*20, SRRW21]. The use of autoencoders and hypernetworks [GSZ*24] allows for exploration of learned manifolds of BRDFs, enabling interpolation between data points or edition of the BRDF model through the latent variables [ZZW*22]. Neural BRDF models do have some drawbacks compared to more physically based approaches: like most data-driven methods, they lack inductive bias for this task, thus requiring a large amount of data to train. Furthermore, when dealing with deep neural networks, the inference cost can quickly become prohibitive for real-time rendering. However, it has been shown that the use of shallow MLPs can be evaluated in real-time during the rendering stage [WDOHN24, ZRW*24, BB25]. Recent methods, such as [ZRW*24, DZJ*24], explore the use of embeddings for real-time inference, reducing the size of the neural network but increasing the memory cost of the method. Other neural BRDFs methods tackle the issue of inference speed more directly. Xu et al. [XCL*25] apply Int8 quantization-aware training to achieve an order-of-magnitude speedup, while Xu et al. [XZL*] combine a coarse-to-fine network architecture with texture-space

shading and spatiotemporal amortization to reach real-time frame rates on mobile VR devices. Regardless of the inference strategy, efficient importance sampling for neural BRDFs remains an open problem [XCL*25]. Existing solutions either require a dedicated network evaluated at every ray bounce [BWZ*, WBX*25], or rely on an expensive conversion to an analytical proxy [ZRW*24]

Across these families of BRDF representations, none simultaneously satisfies the ideal BRDF representation for real-time applications. Our work aims to combine the best of both the analytical and neural worlds: by making the analytical component an integral part of the model, we retain the physical structure and importance sampling scheme of microfacet models while leveraging the expressiveness of neural networks to correct their limitations.

3. Hybrid BRDF Model

In this section, we introduce a hybrid BRDF model that combines a standard analytical reflectance formulation with learned neural components, leveraging the strengths of both physically based and neural approaches. The idea is to let the analytical component model the dominant reflectance lobes and capture the primary perceptual characteristics of the material, while more complex and subtle effects are delegated to a neural network. We jointly optimize both the parameters of the analytical BRDF and the neural network weights in an end-to-end training process, allowing the model to automatically distribute representational capacity between the analytical and learned components.

3.1. General Formulation

Our hybrid formulation models the target BRDF f_t as the combination of an analytical component f_a modulated by a gating function f_g and corrected by an additive residual term f_c :

$$f_t(\omega_i, \omega_o) = f_c(\omega_i, \omega_o, \mathbf{z}) + f_g(\omega_i, \omega_o, \mathbf{z}) \times f_a(\omega_i, \omega_o, \mathbf{p}), \quad (1)$$

where ω_i and ω_o are respectively the incident light and viewing directions, $\mathbf{z} \in \mathbb{R}^l$ denotes a per-BRDF latent code and $\mathbf{p} \in \mathbb{R}^m$ the per-BRDF parameters of the analytical model. The neural component of the model consists of both the gating function f_g and residual term f_c , which are jointly predicted by a single neural network Ψ_w :

$$(f_c, f_g) = \Psi_w(\omega_i, \omega_o; \mathbf{z}), \quad (2)$$

where w denotes the trainable parameters of the neural network. Figure 2 provides an overview of our approach and illustrates the structure of the proposed hybrid BRDF representation. We additionally enforce f_c to be positive, and constrain $f_g \in [0, 1]$ using an exponential activation for the residual term and a sigmoid activation for the gating term. These constraints ensure that the final BRDF remains positive even for combinations of latent codes and analytical parameters not seen during training. This structural property enables intuitive material editing as further detailed in Section 5.1.

In this hybrid formulation, each term plays a distinct role. The analytical term f_a models the primary reflectance behavior and ensures that the model remains grounded to physically-based parameters. The gating function f_g provides a direction-dependent modulation that adjusts the influence of the analytical component. Fi-

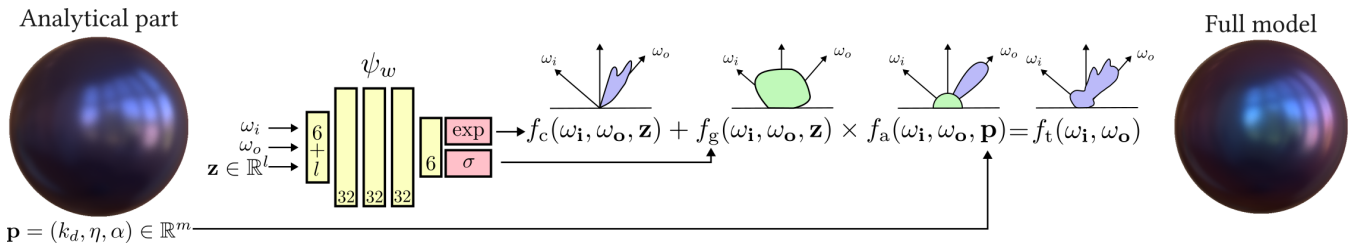


Figure 2: Overview of our method. We jointly optimize a latent code \mathbf{z} and analytical parameters $\mathbf{p} = (k_d, \eta, \alpha)$ for each BRDF and a shared neural network. The output of the neural network (here, a 32×3 MLP) is split into two parts: the correction $f_c(\omega_i, \omega_o, \mathbf{z}) \in \mathbb{R}_+^3$ with an exponential activation and the gate $f_g(\omega_i, \omega_o, \mathbf{z}) \in [0, 1]^3$ with a sigmoid activation. These outputs are combined with the evaluation of the analytical model to produce our improved BRDF.

nally, the additive correction term f_c enables the model to reproduce complex reflectance effects that the analytic model cannot express.

3.2. Model Architecture

In practice, our hybrid model formulation requires defining both a neural network architecture ψ and an analytical model f_a . In this work, we restrict ourselves to a single-lobe analytical model together with a lightweight MLP for the network to meet real-time rendering requirements.

Analytical Model. Inspired by widely used models in real-time scenarios, we use an analytical component f_a composed of a Lambertian lobe and a GGX [WMLT07] microfacet-based specular lobe:

$$f_a(\omega_i, \omega_o; p) = \frac{k_d}{\pi} + \frac{F(\omega_i, \omega_o, \eta)G(\omega_i, \omega_o, \alpha)D(\omega_i, \omega_o, \alpha)}{4(\omega_i \cdot n)(\omega_o \cdot n)}, \quad (3)$$

where the set of parameters $p = \{k_d, \eta, \alpha\}$ is to be fit per material, with $k_d \in [0, 1]^3$ being the diffuse albedo of the Lambertian term, $\eta \in [1, 10]^3$ corresponds to a colored index of refraction and model how the light is reflected with the Fresnel term F , and G and D are respectively the geometric attenuation and normal distribution terms of the microfacet model that depend on a roughness parameter $\alpha \in [0, 1]^2$ for anisotropic materials.

We choose this analytical formulation for its simplicity and its widespread use in real-time rendering. The GGX model is a good fallback for our hybrid model because it is a good trade-off between quality and performance. As shown in Table 4, it is faster to evaluate than any MLP and can also represent a wide variety of materials. In practice, we observe that across the 312 BRDFs fitted in our experiments, this single-lobe model combined with our neural correction is sufficient to reproduce the vast majority of target appearances accurately.

Network Architecture. As illustrated in Figure 2, we model both f_c and f_g with one shallow MLP ψ_w . In practice, we use an MLP with a few hidden layers (1-3) and with limited hidden dimensions (16-32) because such architectures can be evaluated in a fragment shader during the rendering stage for real-time applications [WDOHN24], and optimized leveraging cooperative vec-

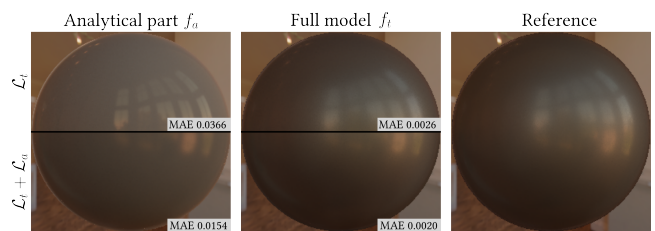


Figure 3: Impact of the analytical loss \mathcal{L}_a term. **Top-part:** with full loss (4), f_a remains close to the target reflectance. **Bottom-part:** without \mathcal{L}_a , f_a is allowed to deviate from the target. This impacts the overall quality and weakens the editability, as the analytic parameters p no longer match the final appearance. Tested Material: *alumn-bronze* from MERL.

tors [BB25]. The inputs of ψ_w are the incoming and outgoing directions ω_i, ω_o stored in Cartesian coordinates and the latent code \mathbf{z} for the selected BRDF. The last linear layer is split into two parts to which we apply an exponential activation for the residual term f_c and a sigmoid activation for the gating term f_g .

3.3. Fitting Method

We now describe how the parameters of our hybrid BRDF model are fitted to measured BRDF data. The parameters describing our models consist of the latent code $\mathbf{z} \in \mathbb{R}^l$, the analytical parameters p and the neural network weights $w \in \mathbb{R}^m$. Although it would be possible to fit all these parameters independently for each target BRDF, this would not be practical in a real-time rendering context, as it would require storing a dedicated set of network weights w for every material. To avoid this prohibitive cost, our fitting strategy relies on sharing the network weights w across a collection of BRDFs $\{f_t^i\}_{i=1}^N$.

In practice, we rely on a mini-batch stochastic gradient descent to jointly optimize the common neural network weights w together with the per-BRDF analytical parameters $\mathbf{p} = \{p_i\}$ and the per-BRDF latent codes $\mathbf{z} = \{z_i\}$. At each training iteration, we randomly sample a batch $\mathcal{B} = \{(\omega_i, \omega_o)\}$ of 1024 incident and outgoing direction pairs drawn from a cosine-weighted distribution over the hemisphere. Each BRDF f_t^i in the dataset is evaluated at these

directions, and we minimize the following objective:

$$\sum_{i=1}^N \sum_{\omega \in \mathcal{B}} \mathcal{L}_a(f^i(\omega), f_a(\omega; p_i)) + \mathcal{L}_t(f^i(\omega), f_t(\omega; p_i, z_i, w)), \quad (4)$$

where f^i , f_a , and f_t denote the reference, analytical, and hybrid BRDFs, respectively. The analytical loss term \mathcal{L}_a encourages the analytical component to approximate the target BRDF as accurately as possible on its own, whereas the hybrid loss term \mathcal{L}_t supervises the full model. Including both terms in the objective is essential to obtain the desired behavior: the analytical model captures the primary perceptual characteristics of the material, such as the dominant diffuse and specular lobes, whereas the neural network focuses on modeling complex residual effects that cannot be well approximated by the analytical formulation alone. Figure 3 demonstrates this behavior on an example material with the result obtained with the combined loss (4) compared to the result obtained with solely the hybrid loss term \mathcal{L}_t . Both loss terms use the same error metric:

$$\mathcal{L}_{a,t}(f_1, f_2) = \|\log(1 + \cos(\theta_i) f_1) - \log(1 + \cos(\theta_i) f_2)\|^2, \quad (5)$$

where θ_i denotes the incident elevation angle. The logarithmic compression reduces the dynamic rangewhile the cosine elevation downweights high-energy grazing-angle samples, which are known to dominate the loss and destabilize training [LKYU12].

In our implementation, the gradients are then computed using auto-differentiation from PyTorch [PGM*19], and optimization is done with the AdamW optimizer [KB14, LH17]. In practice, we optimize the parameters w of the neural network ψ and the per BRDF parameters z, p starting with a learning rate of 0.005 and a cosine decay for 200k steps. We find that clipping gradient norms over 0.01 helps with stability. A full training for 100 BRDFs (such as the MERL dataset) takes about 10 minutes on an RTX 5080.

3.4. Rendering Implementation

Integrating our hybrid model into a rendering system is straightforward. At runtime, the analytical component is handled exactly as in a conventional renderer, with the only addition being the inclusion of the latent vector in the material definition. The neural network is exported as a shader function that is evaluated after the analytical BRDF computation and outputs the gating and corrective terms, which are then used to update the analytical BRDF value according to Eq. (1). As a result, the memory overhead of the hybrid model is limited to the storage of the latent vector, while the size of the neural network (14 kB for a 32×3 MLP shared between all BRDFs of the dataset) determines the performance overhead. Both can be adjusted to meet the performance and quality requirements of the target rendering application. Moreover, our hybrid model can be efficiently sampled by taking advantage of its analytical component f_a .

4. Results

This section provides numerical results that demonstrate our model’s ability to fit complex measured BRDFs (section 4.1). Additionally, in section 4.2, we conduct ablation experiments that validate our hybrid architecture choice together with its capability of

representing a wide range of BRDFs. Then we evaluate the raw rendering performance cost in Section 4.3.

We evaluate our method on the following datasets: 100 Isotropic BRDFs from MERL [MPBM03a], 51 Isotropic and 11 anisotropic BRDFs from the RGL [DJ18b], and 150 anisotropic BRDFs from the UTIA BRDF dataset [FV14]. These three datasets provide a diverse collection of measured real-world materials covering a broad range of appearances, such as fabrics, metals, plastics, paints, etc.

In the experiments presented in this section, we trained our model on each dataset, using a single network shared by all BRDFs within that dataset. This highlights the ability of our hybrid model to represent a large variety of BRDF behaviors using a common network.

4.1. Quality Analysis

To assess the quality of our model, we use two complementary metrics: one measuring the BRDF reconstruction error in function space and the other measuring the rendering error against a reference scene. We then compare these values to the ones obtained with our implementation of the fully neural BRDF architecture from [ZRW*24], and we refer to this method as *neural* in the following. This architecture consists of a shallow MLP and a frame transform module that outputs 3 shading frames to transform the input directions. To be comparable, we replace the encoder part with a trainable latent code with a dimension that matches the dimension of the concatenation of our latent code z and the parameters of the analytical model p . We therefore consider a latent space of dimension 12 (resp. 16) to compare with our approach, with a latent code of dimensions 4 (resp. 8) since we also require 7 (8 for anisotropic) parameters for the analytical model.

BRDF Reconstruction Quality. The BRDF-space reconstruction error is computed with the Symmetric Mean Absolute Percentage Error (SMAPE) by sampling both the target and the estimated BRDF with 10^6 directions $\mathcal{B} = \{(\omega_i, \omega_o)\}$.

$$SMAPE(f, \hat{f}) = \frac{2}{10^6} \sum_{\omega \in \mathcal{B}} \frac{|f(\omega) - \hat{f}(\omega)|}{|f(\omega)| + |\hat{f}(\omega)|}. \quad (6)$$

Figure 4 and 5 show results using this metric. In particular, figure 4 presents the average SMAPE value obtained across all BRDFs of each tested dataset for various MLP sizes, both for our hybrid approach and the neural approach [ZRW*24]. This clearly shows that our hybrid approach provides better BRDF reconstruction, especially for smaller models. In practice, this means that our model can match the quality of fully neural models at much lower computational cost, resulting in a higher frame-rate in real-time applications or shorter rendering times in path-tracing. As the network size increases, the gap between the two approaches decreases. This is expected since larger networks imply better expressiveness and can approximate the BRDF sufficiently well on their own. The relative contribution of the analytical part is therefore less important. Note that this also explains the smaller difference observed on the UTIA dataset, where even very compact networks already achieve low SMAPE values (roughly $10 \times$ smaller than on the other dataset).

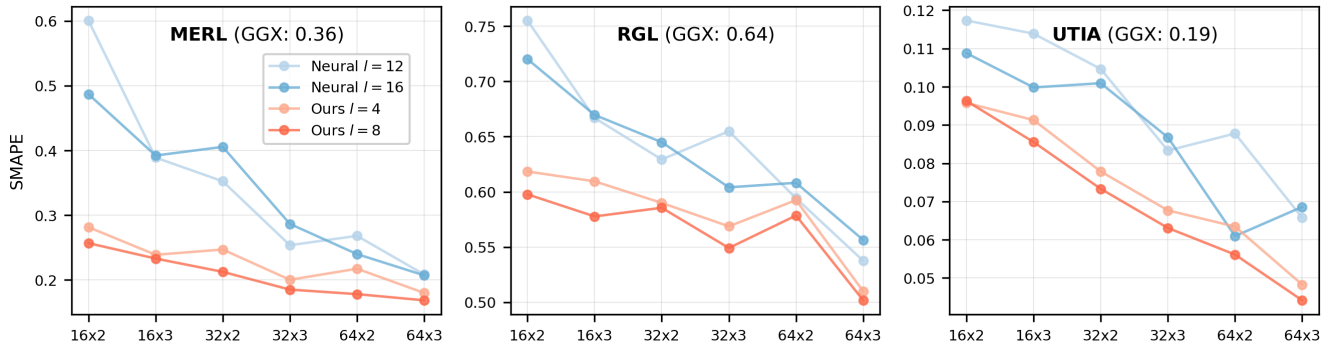


Figure 4: Average SMAPE on BRDF reconstruction for the 3 datasets for multiple MLP size. Since the analytical part of our hybrid model requires 7/8 parameters, we compare it to a fully neural model with 8 additional latent dimensions to provide a comparison with the same number of parameters (per BRDF) for both methods. The metric next to the dataset name is the SMAPE metric of the analytical part of the hybrid model on the particular dataset. For the MERL and RGL datasets, our approach, for small-sized MLPs, outperforms the Neural model or the GGX microfacet-based model. The improvement is not as significant in the case of the UTIA dataset because its materials are very similar, and the anisotropic GGX model alone already provides a good approximation.

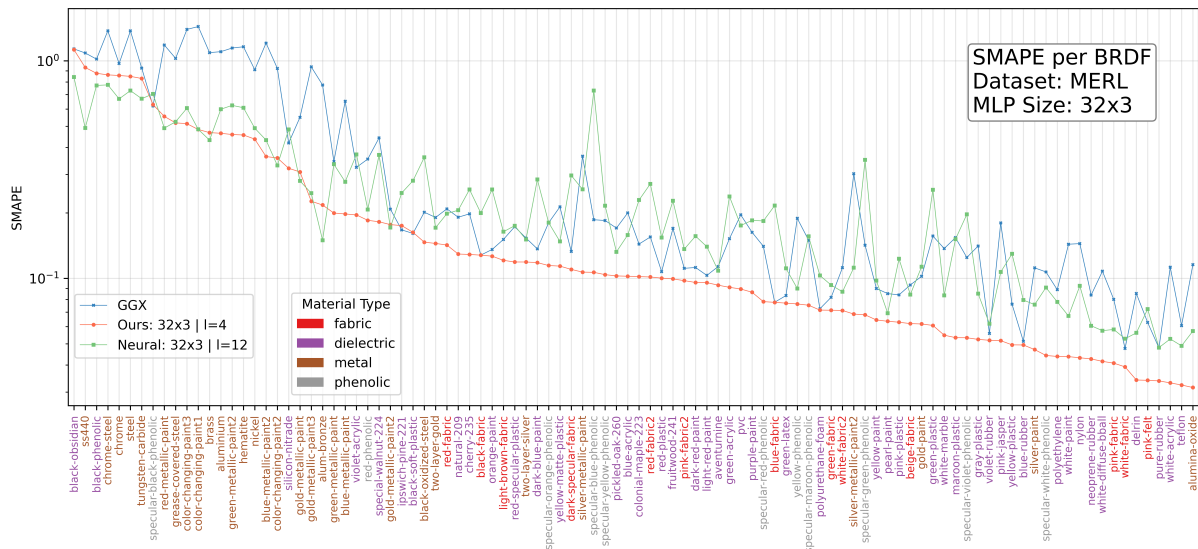


Figure 5: Per BRDF SMAPE on the MERL database, which consists of 100 measured materials. Our model with the 32×3 MLP and a 4D latent space is compared to the Neural model [ZRW*24] with a 32×3 MLP and a 12D latent space, as well as a microfacet-based GGX model. Of the 100 materials, our hybrid model outperforms the Neural model for 88 materials of them. The supplemental material provides the same analysis for the RGL and UTIA databases.

Rendering Quality. Based on the Mean Absolute Error (MAE), we evaluate our model by comparing its renderings with those obtained with only its analytical component, and the neural model from [ZRW*24]. These renderings are computed with 1024 samples per pixel (spp) and compared against a reference image computed with the measured BRDF. Figure 6 shows results for one material of each of the considered datasets in a simple test scene consisting of a sphere lit by the Uffizi environment map. The number of parameters (12) and the MLP size (32×3) are the same for the neural model and ours. As shown in the Figure 6, our model

produces a color-averaged MAE lower than the ones from the Neural or GGX models. Table 1, which gathers the results obtained for the 300 materials tested as well as different MLP sizes and parameters, confirms the performance of our model in the vast majority of cases, especially for small MLPs. In summary, with small MLPs (16×3 and 32×3), our model consistently outperforms the neural model with the same network sizes, for a comparable rendering cost and identical memory usage. In practice, we find that the 32×3 network offers a good compromise between expressiveness

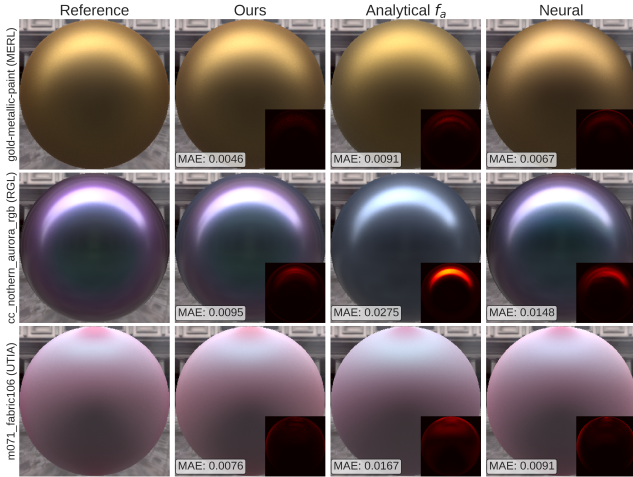


Figure 6: Per-material rendering comparisons between our model, its underlying analytical component, and the neural model. The insets show the per-pixel absolute error as well as the color-averaged MAE for the whole image. The number of parameters (12) and the MLP size (32×3) are the same for the neural model and ours.

and evaluation time, and therefore adopt this configuration for the major part of our rendering experiments.

4.2. Ablation Studies

Model Architecture Variants. To validate our choice of the split additive and multiplicative terms for the model architecture (1), we ran ablations to compare it to three variants:

- *Additive* term only: the MLP has a single output with an identity activation that is directly added to the analytical model.
- *Log Multiplicative*: the MLP has a single output with an identity activation that is multiplied by the analytical model in logarithmic scale, and the output is then exponentiated. This approach ensures that the MLP’s output range can match the range of the analytical BRDF values, making the comparison fairer.
- *Analytical Input*: the same architecture as our model is used, but the MLP takes additionally the analytical model in logarithmic scale as input.

The results in Table 2 show that using both additive and multiplicative terms together yields better performance. Note that using only an additive model significantly degrades the results. Furthermore, we do not observe any notable difference when adding the analytical input. For simplicity, our model omits the analytical input from our model.

Sparse Training. It is possible to add new materials after training on a certain number of BRDFs without needing to update the weights of the MLP ψ_w . We train on half the materials from the MERL Dataset, stratified by material type (metallic, fabric, dielectric, phenolic), and then optimize z and p only for the remaining materials. We compare in Table 3 the performance between a full training and a sparse training, differentiating between materials that

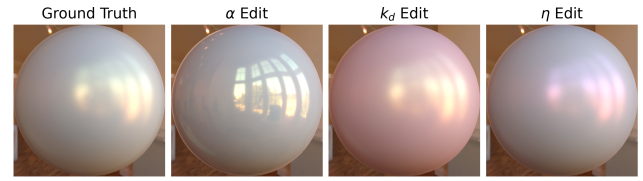


Figure 7: Editing the *satin-gold* BRDF from the RGL dataset by modifying the fitted parameters (roughness α , albedo k_d , IOR η) of the analytical component of our model.

ψ were trained on and the others. When training on only 50% of the MERL BRDFs, we observe a moderate degradation in BRDF-space metrics compared to training on the full dataset. Interestingly, the impact on image space metrics remains limited, with the MAE increasing only slightly from 0.0087 to 0.0099. When evaluating separately on the training subset, the performance remains nearly unchanged between full and sparse training, confirming that optimizing only the latent parameters (z, p) is sufficient to represent materials accurately already seen during training. On the held-out subset, the degradation is more noticeable in BRDF-space, but remains moderate in render-space. This suggests that the network ψ_w captures a material prior that generalizes well to unseen BRDFs, even when trained on only half of the dataset. Overall, these results indicate that new materials can be incorporated after training without updating the weights of ψ_w , while maintaining high rendering fidelity. Additional results and renders are provided in the supplemental material.

4.3. Raw Rendering Performance Evaluation

We measure the impact of evaluating an MLP on the BRDF evaluation in BRDFExplorer [Dis12]. Table 4 reports the frame rendering time when using the analytical BRDF alone, our hybrid BRDF, and the neural BRDF baseline for various MLP sizes. The evaluation time of our approach is comparable to that of the Neural model from [ZRW*24] at equal network size. However, as shown in the previous section, our model achieves significantly better reconstruction quality for the same MLP configurations. Moreover, as we show in the following section, our hybrid formulation enables efficient fallback mechanisms and importance sampling, making it particularly well-suited for real-time rendering applications.

5. Advantages of Our Hybrid Formulation

This section discusses the advantages of our proposed hybrid formulation. The analytical fallback naturally provides intuitive post-training authoring through analytical parameters (Section 5.1), as well as an importance sampling strategy that improves convergence (Section 5.2). Also, our formulation allows for a selective evaluation of the neural network, which is beneficial in many real-time rendering scenarios (Section 5.3).

5.1. Material Exploration

Since the neural component acts as a correction over the analytical model, modifying the analytical parameters produces consistent new appearances. Figure 7 shows examples of varying diffuse

Table 1: Mean Absolute Error (MAE) in terms of rendering. For the three datasets, we compare rendered images (same configuration as the ones from Fig. 6) between a GGX microfacet-based model, our hybrid model, and the Neural one for the same number of parameters and for different sizes of MLP. For a given dataset, the MAE is averaged over all renderings. Except for the RGL dataset and an MLP size of 64×3 or 16×3 , our hybrid model exhibits better results.

MLP	MERL (GGX MAE: 0.0099)				RGL (GGX MAE: 0.0333)				UTIA (GGX MAE: 0.0201)			
	12 params		16 params		12 params		16 params		12 params		16 params	
	Ours $l=4$	Neural $l=12$	Ours $l=8$	Neural $l=16$	Ours $l=4$	Neural $l=12$	Ours $l=8$	Neural $l=16$	Ours $l=4$	Neural $l=12$	Ours $l=8$	Neural $l=16$
16×3	0.0059	0.0086	0.0056	0.0084	0.0267	0.0256	0.0245	0.0288	0.0109	0.0123	0.0106	0.0119
32×3	0.0051	0.0062	0.0052	0.0067	0.0203	0.0238	0.0203	0.0208	0.0101	0.0106	0.0098	0.0113
64×3	0.0046	0.0049	0.0044	0.0048	0.0149	0.0143	0.0160	0.0153	0.0094	0.0098	0.0093	0.0100

Table 2: BRDF-space SMAPE metric for architecture variations of our model. The values are computed for a 32×3 MLP.

Variant	MERL	RGL	UTIA
Analytical Input	0.2015	0.5493	0.0621
Additive	0.4231	0.7148	0.0842
Log Multiplicative	0.2186	0.5907	0.0708
Ours	0.1974	0.5446	0.0644

Table 3: Results when using 50% of the MERL BRDFs on training. For fairness, 50% of BRDFs are not selected randomly from the entire dataset, but from within each material type (Figure 5). The values are computed for a 32×3 MLP.

BRDF in training	Subset	BRDF-space		Render-space	
		MAE	SMAPE	MAE	PSNR
100%	All	0.0031	0.1759	0.0087	37.594
50%		0.0041	0.2104	0.0099	36.659
100%	Train subset	0.0031	0.2057	0.0086	37.698
50%		0.0032	0.2049	0.0088	37.530
100%	Held-out subset	0.0031	0.1461	0.0893	37.491
50%		0.0050	0.2160	0.0110	35.788

color, refractive index, and roughness on a fitted material. The results remain coherent throughout, which we attribute to the joint training strategy: the analytical loss ensures the GGX component encodes the dominant reflectance behavior, so the network only captures residual effects. As a consequence, the analytical parameters retain their physical meaning after fitting, and the network’s correction adapts to moderate parameter changes. This same reasoning explains why linear interpolation between two fitted materials produces plausible intermediate appearances (Figure 8): blending both the analytical parameters and latent codes amounts to interpolating between two physically grounded baselines, with the network filling in consistent residuals throughout. We note that this is not guaranteed for large excursions in parameter space, as disentanglement between the analytical parameters and the latent code is not enforced. Nevertheless, within a reasonable range, the hybrid formulation provides a natural and stable basis for material exploration and spatially varying BRDF maps, where per-textel parameters and latent code maps can drive smooth appearance variation across a surface after training.

Table 4: Evaluation time (ms) of our hybrid model and the neural baseline for different MLP sizes, implemented in a GLSL shader in BRDFExplorer under environment map lighting, averaged over 100 spp, without importance sampling. Timings were recorded using NVIDIA Nsight on a RTX 5080.

MLP Size	16×2	16×3	32×2	32×3	64×2	64×3	GGX
Hybrid (ms)	0.10	0.13	0.22	0.37	0.78	1.40	0.03
Neural (ms)	0.16	0.17	0.30	0.45	0.98	1.59	

5.2. Importance Sampling

Efficient importance sampling remains an open challenge for neural BRDFs [XCL*25]. Existing solutions rely on evaluating a dedicated network at every ray bounce, either to directly predict sampling distributions [BWZ*] or to construct an analytical proxy [ZRW*24], introducing additional inference overhead and preprocessing cost. Our hybrid formulation provides a natural and overhead-free alternative: since the analytical component is jointly trained to remain close to the target BRDF, it serves directly as a sampling proxy. In practice, we apply multiple importance sampling (MIS) between cosine sampling and GGX lobe sampling without requiring any additional neural evaluation. We compare our approach against a tabulated reference method [LRR04] sampled at 1-degree angular resolution (5.62 MB per material), which provides an upper bound on sampling efficiency. As shown in Figures 9 and 10, our method consistently outperforms cosine sampling and approaches the tabulated reference, at a fraction of the memory cost. This shows that the analytical component carries most of the BRDF energy, and it is therefore sound to use it as the primary source for importance sampling.

5.3. Selective Neural Evaluation

Our hybrid model improves visual fidelity over the analytical GGX model but introduces an inference overhead at every shading point. Several strategies can mitigate this cost: leveraging hardware accelerators via the cooperative vectors extension [ZRW*24], exploiting low-bit integer arithmetic [XCL*25], or amortizing inference temporally across frames [XZL*]. These are all orthogonal to our method and directly applicable to our compact shared MLP. Beyond these, our hybrid formulation enables an additional and uniquely effective strategy: evaluating the network only when it meaningfully contributes to the final appearance, and falling back

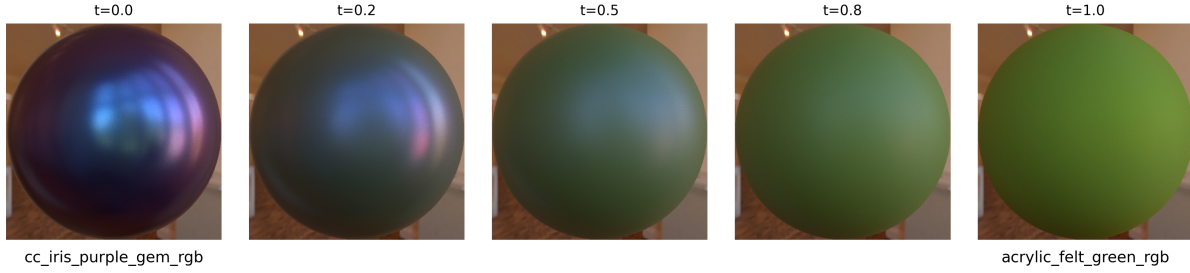


Figure 8: BRDF interpolation example. **From left to right**, we linearly interpolate (parameter t) between two materials from the RGL dataset, the latent vector, and the analytical parameters of our hybrid model, showing very plausible results.

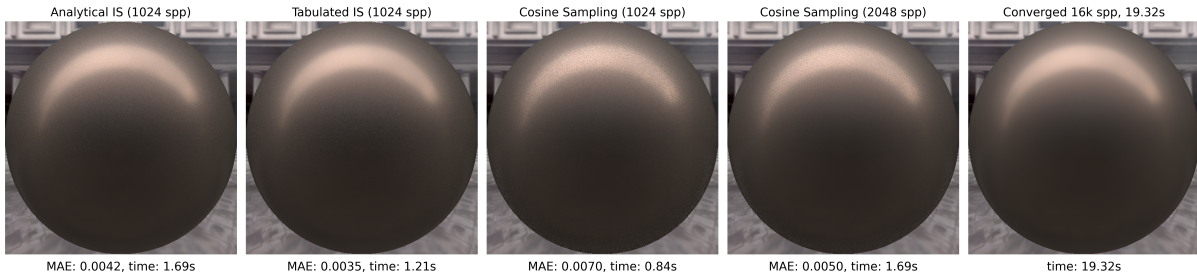


Figure 9: Importance Sampling (IS) comparisons in image space for the red-phenolic material (MERL). **From left to right:** IS based on the GGX distribution and a cosine lobe distribution, tabulated IS (similarly to [LRR04]), with a memory footprint of 5.62 MB, the cosine lobe IS, and a converged result using tabulated IS. For all images, the BRDF values are taken from our fitted hybrid model. As shown by the two images on the left, using an importance sampling based on the microfacet model is not as optimal as the tabulated form but provides clear visual improvements over cosine lobe importance sampling. Additional comparisons are provided in the supplemental material.

to the analytical component otherwise. To demonstrate this, we render a 1080p frame using a GPU path tracer with seven bounces, progressively disabling the neural component of our hybrid model after increasing bounce depths. Table 5 reports rendering times with and without cooperative vectors and Figure 11 shows the corresponding visual results. The neural correction has negligible visual impact beyond the first two bounces, and evaluating the network only at the first hit is often sufficient. Yet the computational gains are substantial: with cooperative vectors, rendering time drops from 2.94 ms to 2.35 ms and 1.6 ms; without hardware acceleration, from 5.57 ms to 3.6 ms and 1.9 ms, approaching the 1.1 ms analytical baseline. Crucially, selective inference and hardware acceleration are complementary. The gains from reducing the amount of neural evaluations are consistent regardless of the inference back end. This makes selective inference an effective and hardware-agnostic strategy, particularly valuable on platforms lacking dedicated neural inference support. More advanced heuristics based on surface roughness, screen-space contribution, or level of detail could also be envisioned to further reduce the rendering overhead.

6. Limitations and Future Work

Model Limitations. The analytical part of the model is intrinsically interpretable from the microfacet theory and provides artistic control and editability (Section 5.1). However, the latent space that drives the corrective layer is not itself interpretable, and editing the latent vector after the optimization process is not guaranteed

Table 5: Rendering time (ms) for different numbers of path tracing bounces in Figure 11. Neural inference, for our hybrid model, is progressively enabled per bounce. Results are shown with inlined and hardware-accelerated (CoopVec) inference. In this rendering scenario, CoopVec are 1.74 times faster on average.

GGX until	Neural inference enabled until	inlined	CoopVec	Speed up
7 bounces	0 bounces	1.10 ms	—	—
	1 bounces	1.90 ms	1.60 ms	× 1.19
	2 bounces	3.60 ms	2.35 ms	× 1.53
	3 bounces	5.20 ms	2.75 ms	× 1.89
	4 bounces	5.40 ms	2.85 ms	× 1.89
	5 bounces	5.50 ms	2.90 ms	× 1.90
	6 bounces	5.55 ms	2.93 ms	× 1.89
7 bounces	5.57 ms	2.94 ms	× 1.89	

to preserve BRDF-like behavior. Moreover, our optimization process does not enforce any physical constraints such as energy conservation or Helmholtz reciprocity. However, we empirically observe that our learned BRDFs remain close to reciprocal in most cases (i.e., absolute error $< 1e-4$), and that their integrated energy tends to be similar or slightly lower than the reference. These limitations suggest interesting future directions: for instance, by adding physically based priors into the model or introducing regularization terms to better structure the latent space.

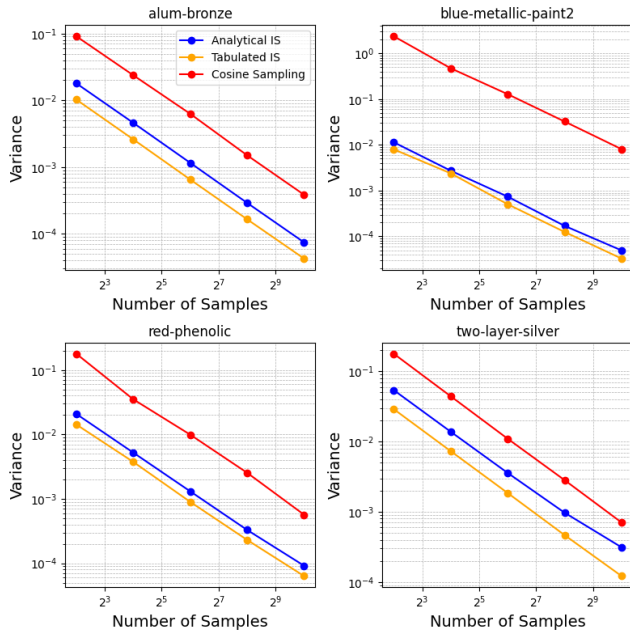


Figure 10: Color-averaged variance of the absolute difference between an image rendered with importance sampling and a reference image computed with 16k samples. Analytical Importance Sampling corresponds to applying either cosine sampling or GGX sampling using a MIS procedure for our model. As shown above, using our IS method approximates the true PDF quite accurately compared to the reference solution provided by tabulated Importance Sampling.

SVBRDFs and Appearance Acquisition. Neural methods are known to perform well when fitting noisy data due to the stochastic nature of the training process, while analytical models perform well on very sparse data due to their built-in inductive bias. Our Hybrid model would therefore be particularly well suited for SVBRDF acquisition using real data from multi-view and multi-light setups, as such data is often noisy and sparse. In practice the spatial variations would be encoded in texture maps of z and p in a similar manner to Zeltner et al. [ZRW*24]. The maps could include aggressive quantization and clusterization of the analytical parameters, leveraging latent space to apply the best possible correction.

Iridescent Materials and Multi Lobe BRDFs. The current implementation of our model may struggle to accurately represent complex effects such as iridescence (Figure 12). More generally, it tends to underperform on BRDFs that fall far outside the single-lobe framework with smooth angular variation. This is due to the strong single-lobe bias imposed by the GGX model used in the analytical component (Section 3.2). Capturing more complex effects would require increasing the expressive power of the model, either by enriching the analytical model or by relying on a more expressive neural component. Richer analytic models, such as multi-lobe formulations, are significantly harder to optimize and introduce additional hyperparameters, while a larger neural component would allow to recover more complex appearances, at a substantial performance cost. In other words, finding the right balance between

analytical priors and neural expressiveness remains an open challenge.

7. Conclusion

We introduced a hybrid reflectance model that combines an analytical microfacet-based GGX model with a lightweight neural network. The latter precisely and automatically fine-tunes the analytical model to reproduce appearances that are not representable by the analytical model alone. We have shown that a shallow neural network, coupled with an analytical model, can reproduce complex appearances from various measured databases, while incurring only a slight increase in computational cost. Importantly, the analytical component remains a close approximation of the target appearance, keeping it meaningful to author or to use as a proxy. Beyond reconstruction quality, our proposed hybrid formulation offers numerous practical advantages for real-world rendering workflows, including intuitive material authoring, efficient importance sampling without additional overhead and flexible performance optimizations through selective neural evaluation. Overall, this results in a new practical reflectance model that jointly offers high expressiveness, stable behavior, fast evaluation, low memory footprint, and some editability, making it well-suited for production use.

References

- [AMHH*18] AKENINE-MÖLLER T., HAINES E., HOFFMAN N., PESCE A., IWANICKI M., HILLAIRE S.: *Real-Time Rendering 4th Edition*. A K Peters/CRC Press, Boca Raton, FL, USA, 2018. 2
- [BB17] BELCOUR L., BARLA P.: A practical extension to microfacet theory for the modeling of varying iridescence. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–14. 2
- [BB25] BELCOUR L., BENYOUB A.: Hardware accelerated neural block texture compression with cooperative vectors. *ArXiv abs/2506.06040* (2025). URL: <https://api.semanticscholar.org/CorpusID:279243696>. 3, 4
- [Bel18] BELCOUR L.: Efficient rendering of layered materials using an atomic decomposition with statistical operators. *ACM Transactions on Graphics* 37, 4 (2018), 1. 3
- [BOK11] BILGILI A., ÖZTÜRK A., KURT M.: A general brdf representation based on tensor decomposition. *Computer Graphics Forum* 30, 8 (2011), 2427–2439. doi:<https://doi.org/10.1111/j.1467-8659.2011.02072.x>. 3
- [Bur12] BURLEY B.: Physically-Based Shading at Disney. 2
- [BWZ*] BAI Y., WU S., ZENG Z., YAN L., WANG B.: Brdf importance baking: A lightweight neural solution to importance sampling general parametric brdfs. *Computer Graphics Forum n/a, n/a*, e70286. doi:<https://doi.org/10.1111/cgf.70286>. 3, 8
- [CBP07] CLAUSTRES L., BARTHE L., PAULIN M.: Wavelet encoding of brdfs for real-time rendering. In *Proceedings of Graphics Interface 2007* (2007), GI '07, Association for Computing Machinery, p. 169–176. doi:[10.1145/1268517.1268546](https://doi.org/10.1145/1268517.1268546). 3
- [CBP21] COOPER V. L., BIERON J. C., PEERS P.: Estimating homogeneous data-driven brdf parameters from a reflectance map under known natural lighting. *IEEE Transactions on Visualization and Computer Graphics* 28, 12 (2021), 4289–4303. 3
- [CT82] COOK R. L., TORRANCE K. E.: A Reflectance Model for Computer Graphics. *ACM Transactions on Graphics* 1, 1 (Jan. 1982), 7–24. doi:[10.1145/357290.357293](https://doi.org/10.1145/357290.357293). 2

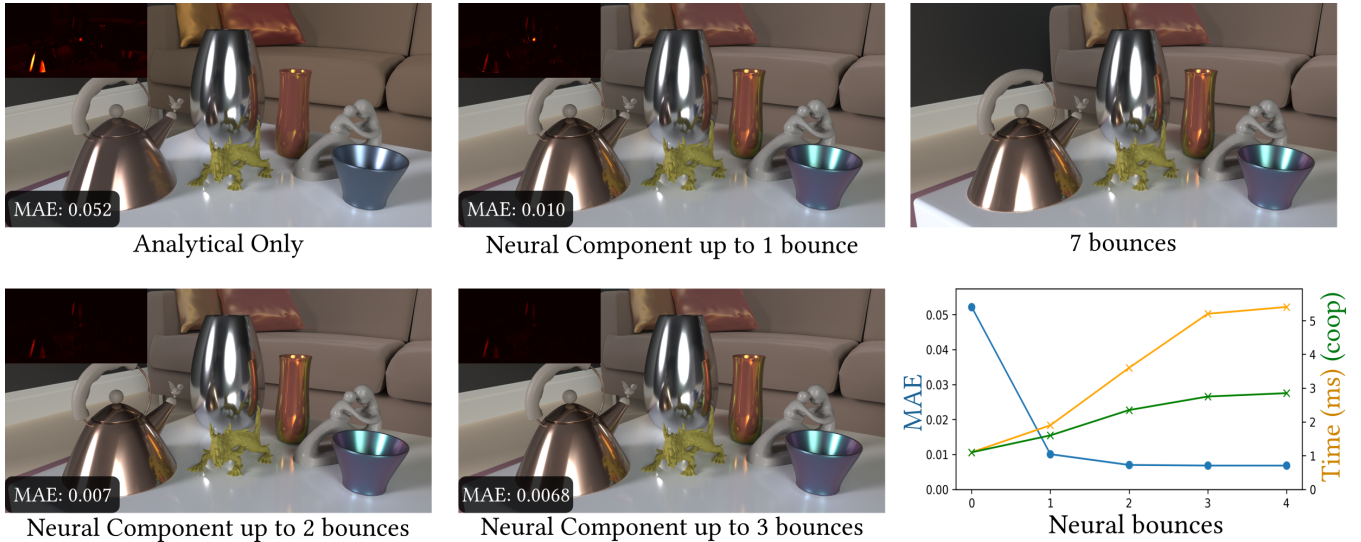


Figure 11: Path-Tracing performance evaluation. All materials of the scene’s objects are rendered using our hybrid microfacet-neural model, which is fitted from three datasets. We show the effect in terms of MAE and rendering time (green and yellow curves show the time per sample with and without cooperative vector acceleration respectively), of falling back on the analytical component model after a certain number of bounces. The different materials used for this scene are listed in the supplementary material.



Figure 12: Our model struggles to reproduce complex phenomena such as iridescence (from *cc_amber_citrine* from RGL) accurately without using a large neural network. At the bottom of each image is displayed, (Left) the color-averaged MAE and (Right) a map of the absolute error.

- [dB22] DE DINECHIN H., BELCOUR L.: Rendering Layered Materials with Diffuse Interfaces. *Proc. ACM Comput. Graph. Interact. Tech.* 5, 1 (May 2022), 13:1–13:12. doi:10.1145/3522620.3
- [Dis12] DISNEY ANIMATION STUDIOS: Brdf explorer. <https://github.com/wdas/brdf>, 2012. Physically-based reflectance visualization and analysis tool. 7
- [DJ18a] DUPUY J., JAKOB W.: An adaptive parameterization for efficient material acquisition and rendering. *ACM Transactions on Graphics* 37, 6 (Dec. 2018), 1–14. doi:10.1145/3272127.3275059. 1
- [DJ18b] DUPUY J., JAKOB W.: An adaptive parameterization for efficient material acquisition and rendering. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 37, 6 (Nov. 2018), 274:1–274:18. doi:10.1145/3272127.3275059. 2, 5
- [DJZ*24] DOU Y., ZHENG Z., JIN Q., NI B., CHEN Y., KE J.: Real-Time Neural BRDF with Spherically Distributed Primitives. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Seattle, WA, USA, June 2024), IEEE, pp. 4337–4346. doi:10.1109/cvpr52733.2024.00415. 3

- [EP10] ELEK O., PRAGUE M.: Layered materials in real-time rendering. In *14th Central European Seminar on Computer Graphics* (2010), vol. 27. 2
- [FBP25] FOURNEAU G., BARLA P., PACANOWSKI R.: Efficient Modeling and Rendering of Iridescence from Cholesteric Liquid Crystals. In *EGSR 2025 - 36th Eurographics Symposium on Rendering* (Copenhagen, Denmark, June 2025), Eurographics Association. 2
- [FV14] FILIP J., VÁVRA R.: Template-based sampling of anisotropic BRDFs. *Computer Graphics Forum* 33, 7 (October 2014), 91–99. doi:10.1111/cgf.12477. 2, 5
- [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. doi:10.1145/3528233.3530732. 3
- [GGN20] GAMBOA L. E., GRUSON A., NOWROUZSAHRAI D.: An Efficient Transport Estimator for Complex Layered Materials. *Computer Graphics Forum* 39, 2 (2020), 363–371. doi:10.1111/cgf.13936. 2
- [GHZ18] GUO Y., HAŠAN M., ZHAO S.: Position-free monte carlo simulation for arbitrary layered BSDFs. *ACM Trans. Graph.* 37, 6 (Dec. 2018), 279:1–279:14. doi:10.1145/3272127.3275053. 2
- [GSZ*24] GOKBUDAK F., SZTRAJMAN A., ZHOU C., ZHONG F., MANTIUK R., OZTIRELI C.: Hypernetworks for Generalizable BRDF Representation. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LXXVI* (Oct. 2024), Springer-Verlag, pp. 73–89. doi:10.1007/978-3-031-73116-7_5. 3
- [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 (2020), 157–166. doi:10.1111/cgf.13920. 3
- [HHdD16] HEITZ E., HANIKA J., D’EON E., DACHSBACHER C.: Multiple-scattering microfacet bsdfs with the smith model. *ACM Trans. Graph.* 35, 4 (July 2016). doi:10.1145/2897824.2925943. 2
- [HP17] HOLZSCHUCH N., PACANOWSKI R.: A two-scale microfacet

- reflectance model combining reflection and diffraction. *ACM Trans. Graph.* 36, 4 (July 2017). doi:10.1145/3072959.3073621. 2
- [JdJM14] JAKOB W., D'EON E., JAKOB O., MARSCHNER S.: A comprehensive framework for rendering layered materials. *ACM Transactions on Graphics (ToG)* 33, 4 (2014), 1–14. 3
- [KB14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. *CoRR abs/1412.6980* (2014). URL: <https://api.semanticscholar.org/CorpusID:6628106>. 5
- [KC17] KULLA C., CONTY A.: Revisiting physically based shading at imageworks. *SIGGRAPH Course, Physically Based Shading 2*, 3 (2017). 2
- [KDS96] KOENDERINK J. J., DOORN A. J. v., STAVRIDIS M.: Bidirectional reflection distribution function expressed in terms of surface scattering modes. In *Proceedings of the 4th European Conference on Computer Vision-Volume II - Volume II* (Berlin, Heidelberg, 1996), ECCV '96, Springer-Verlag, p. 28–39. 3
- [KSS02] KAUTZ J., SNYDER J., SLOAN P.-P. J.: Fast arbitrary brdf shading for low-frequency lighting using spherical harmonics. *Rendering Techniques 2*, 291–296 (2002), 1. 3
- [LDR14] LAGARDE S., DE ROUSIERS C.: Moving frostbite to physically based rendering 3.0. *SIGGRAPH Course: Physically Based Shading in Theory and Practice 3* (2014). 2
- [LFTG97] LAFORTUNE E. P. F., FOO S.-C., TORRANCE K. E., GREENBERG D. P.: Non-linear approximation of reflectance functions. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (USA, 1997)*, SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., p. 117–126. doi:10.1145/258734.258801. 2
- [LH17] LOSHCHELOV I., HUTTER F.: Decoupled weight decay regularization. In *International Conference on Learning Representations* (2017). URL: <https://api.semanticscholar.org/CorpusID:53592270>. 5
- [LKYU12] LÖW J., KRONANDER J., YNNERMAN A., UNGER J.: BRDF models for accurate and efficient rendering of glossy surfaces. *ACM Transactions on Graphics 31*, 1 (Jan. 2012), 1–14. doi:10.1145/2077341.2077350. 5
- [LRR04] LAWRENCE J., RUSINKIEWICZ S., RAMAMOORTHI R.: Efficient brdf importance sampling using a factored representation. In *ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), SIGGRAPH '04, Association for Computing Machinery, p. 496–505. doi:10.1145/1186562.1015751. 8, 9
- [MPBM03a] MATUSIK W., PFISTER H., BRAND M., MCMILLAN L.: A data-driven reflectance model. *ACM Trans. Graph.* 22, 3 (July 2003), 759–769. doi:10.1145/882262.882343. 2, 5
- [MPBM03b] MATUSIK W., PFISTER H., BRAND M., MCMILLAN L.: A data-driven reflectance model. In *ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), SIGGRAPH '03, Association for Computing Machinery, p. 759–769. URL: <https://doi.org/10.1145/1201775.882343>, doi:10.1145/1201775.882343. 2, 3
- [MTR08] MAHAJAN D., TSENG Y.-T., RAMAMOORTHI R.: An analysis of the in-out brdf factorization for view-dependent relighting. *Computer Graphics Forum 27*, 4 (2008), 1137–1145. doi:https://doi.org/10.1111/j.1467-8659.2008.01251.x. 3
- [NDM05] NGAN A., DURAND F., MATUSIK W.: Experimental analysis of brdf models. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques* (Goslar, DEU, 2005), EGSR '05, Eurographics Association, p. 117–126. 2, 3
- [PGM*19] PASZKE A., GROSS S., MASSA F., LERER A., BRADBURY J., CHANAN G., KILLEEN T., LIN Z., GIMELSHEIN N., ANTIGA L., DESMAISON A., KÖPF A., YANG E., DEVITO Z., RAISON M., TEJANI A., CHILAMKURTHY S., STEINER B., FANG L., BAI J., CHINTALA S.: Pytorch: An imperative style, high-performance deep learning library, 2019. URL: <https://arxiv.org/abs/1912.01703>, arXiv:1912.01703. 5
- [PSCS*12] PACANOWSKI R., SALAZAR CELIS O., SCHLICK C., GRANIER X., POULIN P., CUYT A.: Rational brdf. *IEEE Transactions on Visualization and Computer Graphics 18*, 11 (2012), 1824–1835. doi:10.1109/TVCG.2012.73. 3
- [RCL21] RANDRIANANDRASANA J., CALLET P., LUCAS L.: Transfer matrix based layered materials rendering. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–16. 3
- [SBN15] SOLER C., BAGHER M. M., NOWROUZEZAHRAI D.: Efficient and accurate spherical kernel integrals using isotropic decomposition. *ACM Transactions on Graphics (TOG)* 34, 5 (2015), 1–14. 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. doi:10.1111/cgf.14335. 3
- [SS95] SCHRÖDER P., SWELDENS W.: Spherical wavelets: efficiently representing functions on the sphere. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1995), SIGGRAPH '95, Association for Computing Machinery, p. 161–172. URL: <https://doi.org/10.1145/218380.218439>, doi:10.1145/218380.218439. 3
- [TUGM22] TONGBUASIRILAI T., UNGER J., GUILLEMOT C., MIANDJI E.: A sparse non-parametric brdf model. *ACM Trans. Graph.* 41, 5 (Oct. 2022). URL: <https://doi.org/10.1145/3533427>, doi:10.1145/3533427. 3
- [WB20] WEIER P., BELCOUR L.: Rendering layered materials with anisotropic interfaces. *Journal of Computer Graphics Techniques (JCGT)* 9, 2 (2020), 20. 3
- [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 the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers* (New York, NY, USA, 2025), SIGGRAPH Conference Papers '25, Association for Computing Machinery. doi:10.1145/3721238.3730679. 3
- [WDOHN24] WEINREICH C., DE OLIVEIRA L., HOUDARD A., NADER G.: Real-time neural materials using block-compressed features. In *Computer Graphics Forum* (2024), vol. 43, Wiley Online Library, p. e15013. 3, 4
- [WMLT07] WALTER B., MARSCHNER S. R., LI H., TORRANCE K. E.: Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques* (Goslar, DEU, June 2007), EGSR '07, Eurographics Association, pp. 195–206. 2, 4
- [WRG*09] WANG J., REN P., GONG M., SNYDER J., GUO B.: All-frequency rendering of dynamic, spatially-varying reflectance. *ACM Trans. Graph.* 28, 5 (Dec. 2009), 1–10. doi:10.1145/1618452.1618479. 3
- [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* (Dec. 2007), GRAPHITE '07, Association for Computing Machinery, pp. 171–178. doi:10.1145/1321261.1321292. 2
- [XCL*25] XU Z., CHEN X., LIU C., WANG B., WANG L., MONTAZERI Z., YAN L.-Q.: Towards comprehensive neural materials: Dynamic structure-preserving synthesis with accurate silhouette at instant inference speed. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers* (New York, NY, USA, 2025), SIGGRAPH Conference Papers '25, Association for Computing Machinery. doi:10.1145/3721238.3730626. 3, 8
- [XH18] XIE F., HANRAHAN P.: Multiple scattering from distributions of specular v-grooves. *ACM Trans. Graph.* 37, 6 (Dec. 2018). doi:10.1145/3272127.3275078. 2
- [XSD*13] XU K., SUN W.-L., DONG Z., ZHAO D.-Y., WU R.-D., HU

- S.-M.: Anisotropic spherical gaussians. *ACM Trans. Graph.* 32, 6 (Nov. 2013). doi:10.1145/2508363.2508386. 3
- [XZL*] XU Z., ZHOU Y., LITMAN Y., CHIANG M. J.-Y., YAN L., MICHELS A.: Real-time neural materials on mobile vr. *Computer Graphics Forum* n/a, n/a, e70318. doi:https://doi.org/10.1111/cgf.70318. 3, 8
- [YXW*23] YU Y., XIA M., WALTER B., MICHELSEN E., MARSCHNER S.: A full-wave reference simulator for computing surface reflectance. *ACM Trans. Graph.* 42, 4 (July 2023). URL: https://doi.org/10.1145/3592414, doi:10.1145/3592414. 2
- [YYTM19] YAMAGUCHI T., YATAGAWA T., TOKUYOSHI Y., MORISHIMA S.: Real-time rendering of layered materials with anisotropic normal distributions. In *SIGGRAPH Asia 2019 Technical Briefs*. 2019, pp. 87–90. 3
- [ZERB05] ZICKLER T., ENRIQUE S., RAMAMOORTHI R., BELHUMEUR P.: Reflectance sharing: image-based rendering from a sparse set of images. EGSR '05, Eurographics Association, p. 253–264. 3
- [ZJ18] ZELTNER T., JAKOB W.: The layer laboratory: a calculus for additive and subtractive composition of anisotropic surface reflectance. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–14. 3
- [ZRW*24] ZELTNER T., ROUSSELLE F., WEIDLICH A., CLARBERG P., NOVÁK J., BITTERLI B., EVANS A., DAVIDOVIĆ T., KALLWEIT S., LEFOHN A.: Real-time Neural Appearance Models. *ACM Transactions on Graphics* 43, 3 (June 2024), 1–17. doi:10.1145/3659577. 2, 3, 5, 6, 7, 8, 10
- [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. doi:10.1145/3490385. 3