

# Neural Enhancement of Analytical Appearance Models

Xuanzhe Shen, Xiaohe Ma, Kun Zhou *Fellow, IEEE* and Hongzhi Wu *Member, IEEE*

**Abstract**—Traditional analytical reflectance models, while compact and interpretable, lack the capacity to accurately represent physical measurements. Recent neural models, which closely fit input data, are less generalizable and often more expensive to store and evaluate. To combine the strengths and overcome the limitations of these two classes of models, we present neural enhancement, a novel framework to boost an input analytical appearance model, by identifying and replacing its key computational nodes/operators with small-scale multi-layer perceptrons. This allows us to leverage the computational graph structure of the original model, while improving its expressiveness at a modest cost. To make the enhancement computationally tractable, we propose a hypercube-based search to automatically and efficiently identify the node(s) and/or operator(s) to be replaced towards maximal gain in a differentiable fashion. We enhance a number of common analytical BRDF models. The results are, at once accurate, compact and efficient, and compare favorably with state-of-the-art work on fitting measured reflectance. Finally, our models are fully compatible with standard rasterization or ray-tracing pipeline.

**Index Terms**—BRDF, reflectance, appearance modeling

## 1 INTRODUCTION

MODELING the visual appearance of objects is a fundamental problem in computer graphics [1]. For reflective appearance, Bidirectional Reflectance Distribution Function (BRDF) is a 4D function that completely describes the reflectance distribution of a material for any lighting and view directions [2]. To represent a BRDF, a straightforward tabulation would impose a heavy burden on storage, editing, and rendering, due to the sheer dimensionality of this function [3]. Therefore, seeking an optimal appearance representation has been an important on-going quest over the past 50 years.

Starting with the seminal work of [4], researchers leverage the knowledge from fields including physics and statistics, to derive a series of highly compact analytical models for representing a BRDF. They range from early phenomenological models [5], [6] to first principles ones [7], [8], [9], many of which are still actively used in academia and industry today. Although analytical models are compact, efficient, and interpretable/editable, they lack the capacity to fit to measured data with high precision [10].

On the other hand, neural implicit models have gained increasing popularity in the last decade. Since neural networks with sufficient capacities are essentially universal approximators, these models precisely fit measured data with a BRDF-specific network [11], or a large BRDF-independent network in conjunction with optimizeable latent parameters [12]. However, as a data-driven approach, neural models are not easily generalizable. Moreover, the lack of semantics in an end-to-end neural approach makes interpretation/editing challenging.

Our goal is to combine the strengths of both analytical and neural models, by finding a reflectance model that is at once, accurate, compact, and efficient. We are inspired by the historical development of analytical BRDF models: expert researchers often build new models upon existing ones; they first identify under-performing term(s) (e.g., the microfacet normal distribution function, or the Fresnel term), and then replace with a new, carefully hand-crafted version, to improve the overall performance. We would like to harness the power of modern machine learning tools to perform this task in an automatic and systematic manner.

In this paper, we present neural enhancement, a novel framework to enhance an input analytical appearance model, which identifies and replaces bottleneck computational node(s) and/or operator(s) using multilayer perceptrons (MLPs), toward maximum gain in fitting accuracy. This allows us to leverage the existing computational graph structure of the original model, while improving its expressiveness at a modest cost. As a straightforward algorithm would result in a combinatorial explosion, we propose a hypercube-based search to test and gradually update only a small subset of all replacement options, making the process computationally tractable.

The effectiveness of neural enhancement is demonstrated on the standard anisotropic Cook-Torrance BRDF model with GGX distribution [13] (parameter# = 12). Using the training data from multiple appearance datasets, our enhanced model (parameter# = 39) is a unified representation for accurate fitting of complex measured BRDFs [14], whose quality is comparable or better than state-of-the-art BRDF models. Due to the simple structure of MLP, our model can be directly translated into shader code for standard rasterization or ray-tracing pipeline. The performance of an unoptimized implementation of the enhanced model is comparable to a state-of-the-art neural method with the smallest network size [11]. We also apply our idea to improve other

• K. Zhou and H. Wu are the corresponding authors. X. Shen and X. Ma contributed equally. All authors are with State Key Lab of CAD & CG, Zhejiang University, Hangzhou, China, 310058. K. Zhou is also affiliated with Hangzhou Research Institute of AI and Holographic Technology. E-mail: {kunzhou,hwu}@acm.org

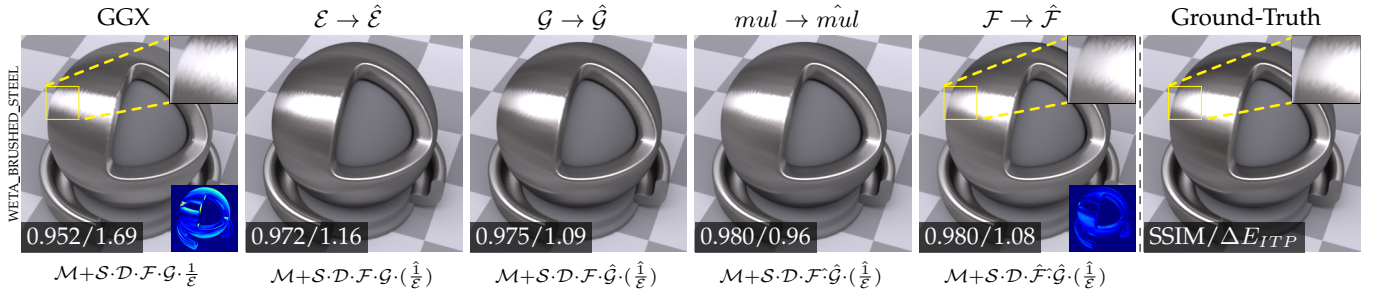


Fig. 1: We enhance the accuracy of an input analytical appearance model, by automatically identifying and replacing original computational nodes/operators with small multi-layer perceptrons (indicated with hats). The 1st to 5th images visualize the boosting process for GGX BRDF model, resulting in a new, hybrid model that more closely matches the measured ground-truth (last image). Quantitative errors in SSIM and  $\Delta E_{ITP}$  ( $\times 10^3$ ) rae reported in the bottom-left of each related image. We show the error maps of both the original GGX BRDF and the enhanced GGX BRDF, along with zoomed-in results, on the right side of the corresponding images. The vanilla GGX cannot accurately fit grazing-angle anisotropic reflections, unlike  $\epsilon \rightarrow \hat{\epsilon}$  model. Please refer to Fig. 2 for the definition of each term in the analytical model.

analytical models [5], [7], [15], test on bidirectional texture functions (BTFs) and evaluate the impact of different factors on the final result. We hope that our hybrid approach will inspire interesting future work along this direction.

## 2 RELATED WORK

We review related work on bidirectional appearance functions defined at a *single* location. No spatial coherence is exploited, unlike in, e.g., [16], [17]. While our focus is on BRDF, we also list some BTF-related techniques, due to the similarity in handling a BRDF and a BTF texel: both are parameterized over a pair of lighting and view directions. Another line of work [18], [19] also involves node graphs for appearance, but is orthogonal to ours as their focus is on the spatial domain. Interested readers are referred to excellent recent surveys for a broader view of the topic [20], [21], [22], [23], [24].

### 2.1 Analytical BRDF Models

This class of work can be roughly divided into phenomenological and first principles ones. Phenomenological models rely on careful observations/expertise to develop compact analytical expressions for appearance modeling. For example, the seminal work of [4] proposes a simple model for computing specular highlights. Ward [5] introduces a similar model for anisotropic reflections based on a distribution suggested by Beckmann and Spizzichino [25]. Ashikhmin and Shirley [9] further adopt an explicit Fresnel term and a non-constant diffuse one to improve expressiveness.

First principles models are based on manual derivations directly from or inspired by physics. Torrance and Sparrow establish the microfacet theory [26], which is later introduced to graphics by Blinn [27], and finally leads to the widely used Cook-Torrance BRDF [7]. Schlick [28] develops efficient approximations for the Fresnel term and the shadowing term of [29]. The shadowing function is derived from the numeric integration of arbitrary microfacet distributions in [30]. Walter et al. [13] rediscover [31] as the GGX (Trowbridge-Reitz) distribution, and this model becomes the de-facto standard for physically-based rendering (PBR)

nowadays. For brevity, we refer to it as the GGX BRDF model or GGX model in this paper.

Recently, Wu et al. [32] search for a sparse mixture of analytical BRDF models to represent a BTF texel, which can be viewed as an apparent BRDF (ABRDF). Brady et al. [15] employ genetic programming in deriving mathematical expressions to characterize complex BRDFs via a large-scale optimization.

Despite considerable continuing research efforts [33], [34], [35], it is fundamentally challenging for an analytical model to fit measured data with high accuracy [10], due to the insufficient capacity of its compact expression. By neural enhancement of an analytical model, our approach increases its expressiveness by replacing it with a few small MLPs, while inheriting many of its strengths, such as compactness, efficiency, and generalizability.

### 2.2 Neural Appearance Models

These models can be categorized into **material-specific** and **material-independent** ones. The first line of work trains a specific network for each material, essentially trading generalization for accuracy. Rainer et al. [36] introduce a neural encoding of BTF texels with a material-specific autoencoder. Sztrajman et al. [11] improve highlight reconstruction with lightweight networks using adaptive angular sampling. Trunz et al. [37] propose a Shapley-value-based method for latent space pruning. This method does not require retraining and can preserve angular appearance fidelity while achieving compact reflectance representations. In a follow-up paper [38], Gokbudak et al. generate the network weights through a hyper-network, enabling reconstruction even under sparse sampling conditions. Zeltner et al. [39] introduce a real-time neural appearance model that encodes high-resolution Spatially Varying Bidirectional Reflectance Distribution Function (SVBRDF) parameters into a hierarchical latent texture, decoded by two lightweight MLPs for BRDF evaluation and importance sampling.

Another line of work proposes a general network whose weights are fixed after training. It can represent different materials with optimizable per-material neural parameters.

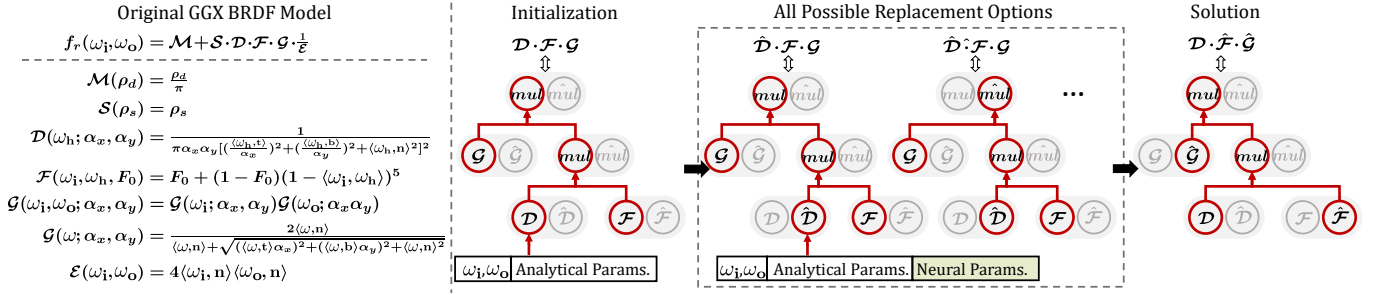


Fig. 2: Our enhanced model. Here the GGX BRDF model is an input example, defined in the left part of the figure. We start with converting the analytical model into a computational graph of computational nodes and arithmetic operators. Note that only a subgraph is displayed here, due to limited space. Each node/operator can be replaced with a neural module (indicated with a hat), with analytical parameters replaced by the concatenation of analytical and neural ones. Ideally, we would like to pick the model with the lowest loss among all possible replacement options, as the final solution. Params. = parameters.

Hu et al. [40] design an autoencoder to discover a low-dimensional latent space from a sequence of BRDF slices. Rainer et al. [41] build upon [36] to train a unified network architecture, which projects reflectance measurements to a shared latent parameter space. Kuznetsov et al. [42] propose a neural framework for multi-scale appearance, which is improved in [43] to better handle complex shadows and specular highlights. Fischer and Ritschel [44] propose a meta-learning approach for BRDF and SVBRDF estimation that achieves near-overfit quality in a few gradient steps using a learned initialization and step size. Zheng et al. [45] work in the BRDF function spaces, and exploit neural processes to obtain a compact latent representation. Fan et al. [12] perform a learned layering operation on BRDF latent codes via a large network to recover the original function. A biplane representation for BTF is proposed in [16], which learns to encode high-frequency details in both spatial and angular domains. A generalized normal distribution function is introduced to learn high-frequency directional patterns [46]. Shi et al. [47] train a universal network to predict meso-scale geometry and micro-scale BRDFs based on a large-scale BRDF. Conceptually, our approach can be viewed as a generalized version of the above two, as all nodes/operators can be neural enhanced.

As a data-driven approach, neural models with sufficient capacities can precisely (over)fit measurements. However, the network size is often large in order to implement a complex mapping, resulting in expensive training and rendering. The generalization ability is also limited, due to the lack of first principles often used in the design of analytical models. Although our model belongs to the class of **material-independent**, general neural models, it inherits the majority of the computational graph structure of an input analytical model, and only employs a few small MLPs. These two factors considerably alleviate the aforementioned issues.

### 2.3 Symbolic Regression

Symbolic regression aims to discover a symbolic expression  $f$  from a number of  $(x, f(x))$  pairs [48]. Recently, Petersen et al. [49] use a recurrent neural network to generate a distribution over tractable mathematical expressions, and

train the network with a novel risk-seeking policy gradient to improve expression fitting. A method is proposed in [50] to identify generalized symmetries in computational graphs using neural network gradients, extending symbolic regression to probability distributions with normalizing flows. Biggio et al. [51] use large-scale pre-training algorithms to procedurally generate an infinite set of equations, and train a transformer to predict symbolic equations from input-output pairs. In comparison, we do not derive pure symbolic expressions from scratch, due to its limited capacity.

## 3 OUR MODEL

Our enhanced model can be viewed as a modified version of an analytical one. Given an input analytical model, we first convert it into a computational graph with terminal computational nodes and internal arithmetic operators, as illustrated in Fig. 2. Next, some of the nodes and/or operators will be identified and replaced (detailed in Sec. 4) with small MLPs, which we call neural modules, to improve the overall accuracy.

The nodes represent semantic terms defined in the original model. For example, the nodes of the GGX BRDF model are Lambertian reflectance (denoted as  $\mathcal{M}$ ), specular albedo ( $\mathcal{S}$ ), microfacet distribution function ( $\mathcal{D}$ ), Fresnel approximation ( $\mathcal{F}$ ), geometric factor ( $\mathcal{G}$ ), and the reciprocal normalization term ( $\frac{1}{\mathcal{E}}$ ), as shown in Fig. 2. We use the reciprocal term to avoid division as an operator, which tends to be less stable in optimization. The original input are the lighting/view direction  $(\omega_i/\omega_o)$ , as well as BRDF parameters (diffuse/specular albedo:  $\rho_d/\rho_s$ , roughnesses:  $\alpha_x/\alpha_y$ , the specular reflectance at normal incidence:  $F_0$ , normal:  $\mathbf{n}$ , and tangent expressed as a rotation angle  $\mathbf{t}_\theta$ ).

For an operator, it takes the results from two children nodes/neural modules as input, and outputs the result of the arithmetic computation it represents. In this paper, addition/multiplication are used as operators.

The neural module is simply a small MLP. Its output dimensionality is designed to match that of the node or operator it replaces. The input configuration of the MLP is determined by the specific node being replaced. If the replaced node takes any input BRDF parameters, these are

substituted with a concatenation of all analytical parameters and a newly introduced set of neural parameters. The inclusion of  $\omega_i, \omega_o$  in this concatenation depends on whether they serve as inputs to the original node. For example, we are enhancing the GGX BRDF model, and setting the number of neural parameters to 27; a node takes  $\alpha_x, \alpha_y$  and  $\omega_i, \omega_o$  as input. In this case, the input dimension to the MLP would be  $12(\text{original analytical parameters}) + 27(\text{neural parameters}) + 6(\text{for } \omega_i, \omega_o) = 45$ . If an operator is replaced, the input remains as two operands from two children. In this paper, the MLP is a 4-layer fully connected network with 16, 32, 16 neurons per hidden layer respectively (Fig. 3). Each FC layer before the last one is followed by a leaky ReLU activation layer.

Note that our approach is not limited to the above configurations. We experiment with various settings and find they perform well in practice. Please refer to Sec. 7.2 for the impact of various factors over the final result, including MLP architectures, sizes, and the number of neural parameters. It is also worth mentioning that there are multiple ways to build the computational graph. A node could group together multiple semantic terms as well as their mutual operators. However, doing so draws closer to an end-to-end neural network. On the other hand, a node can be as small as a single input parameter/constant. This, however, significantly increases the computational cost, and makes it difficult to find a suitable neural module for such a simple node. The current granularity of nodes mimics the manual development of analytical models. Additionally, our method is not limited to assigning each node to either a neural module or a single analytical expression. In principle, candidate components from multiple analytical models and the neural module could jointly compete at each node. However, to avoid a rapid increase in complexity, we currently enhance one analytical model at a time. We leave it to future work to automatically determine the optimal granularity for each node and to enable more comprehensive enhancement that incorporates multiple analytical models.

## 4 NEURAL ENHANCEMENT

In this section, we describe how to identify and replace nodes/operators with neural modules (Sec. 3) in the computational graph of an input analytical model, towards maximum gain in fitting quality. A naïve straightforward approach would enumerate all  $2^N$  different combinations of neural replacement options, train an enhanced model for each configuration, and pick the one that produces the lowest error. Here  $N$  is the total number of nodes and operators in the graph. To avoid this combinatorial explosion, we propose a hypercube-based search algorithm at the cost of landing on a local minimum.

### 4.1 Enhancement State

Before describing our algorithm, we first introduce a related concept. For a node/operator, we use a bit to indicate whether it is the original one (=0) or replaced with a neural module (=1). Therefore, the enhancement state of all nodes and operators in a computational graph can be represented as an N-bit vector. The collection of all possible states forms

an N-dimensional hypercube [52]. Generally speaking, if there are  $m - 1$  different neural modules to choose from for each node/operator, we can extend the enhancement state to an N-dimensional vector of  $m$ -ary digits.

### 4.2 Algorithm

We denote the current enhancement state as  $B_c$ , which is initialized to be an all-zero vector corresponding to the input analytical model. To efficiently reduce computational complexity, instead of training all possible models, each time we only consider a model (whose state is denoted as  $B$ ) that differs from the current model in at most one node/operator. This corresponds to a Hamming distance of less or equal to 1 between  $B$  and  $B_c$  (i.e., a node/operator may be replaced by a neural module; conversely, a module may be switched back to the original node/operator). There are  $N+1$  such models. Each  $B$  is independently built using the computational graph and optimized BRDF/neural parameters of  $B_c$ , then a neural module is added or reverted, or the computational graph remains unchanged, based on the Hamming distance between  $B_c$  and  $B$ . We train each of these models by jointly optimizing the weights in all neural modules, neural parameters, and analytical BRDF parameters of  $B_c$  and all  $B$ . After a user-specified number of epochs (30 in main experiments), we pick the model with the lowest error as the current one, and repeat the above process until the enhancement state does not change. Fig. 4 illustrates the correspondence between the state and the computational graph, along with the relationship between  $B_c$  and  $B$ .

Note that we are not limited to using a Hamming distance threshold of 1. With more computational resources, one can certainly increase the number of models being evaluated simultaneously with a larger threshold (e.g., a threshold of 2 corresponds to  $1 + N + \frac{N(N-1)}{2}$  models). Moreover, our hypercube-based search is flexible to include additional constraints. For example, the user might specify if certain nodes/operators remain analytical or neural, which correspond to fixed bits in the enhancement state vector. One can also impose an upper limit on the number of neural modules, which can be computed by counting the number of 1's in the state.

### 4.3 Training

To train a model, our loss function measures the L1 difference between the log-transformed ground-truth and predicted BRDF value as follows, similar to previous work [11], [45]:

$$L = \sum_{\mathbf{k}} \left| \log \left( 1 + \tilde{f}_r(\omega_i^{\mathbf{k}}, \omega_o^{\mathbf{k}})(\tilde{\mathbf{n}}, \omega_i^{\mathbf{k}}) \right) - \log \left( 1 + f_r(\omega_i^{\mathbf{k}}, \omega_o^{\mathbf{k}})(\tilde{\mathbf{n}}, \omega_i^{\mathbf{k}}) \right) \right|, \quad (1)$$

where  $\tilde{f}_r/f_r$  is the ground-truth/predicted BRDF value with the current model, respectively.  $\tilde{\mathbf{n}}$  is the ground-truth normal, and  $\omega_i^{\mathbf{k}}/\omega_o^{\mathbf{k}}$  is a sampled lighting/view direction, respectively.

We use the RMSProp optimizer with an alpha of 0.9 and the learning rate of  $10^{-3}$ , as it provides more stable

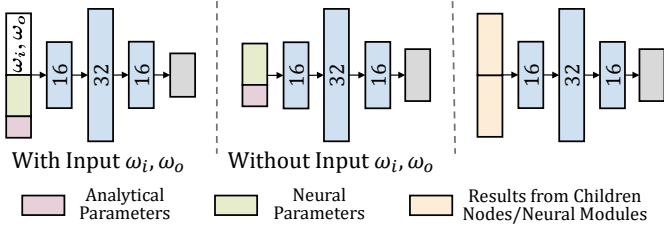


Fig. 3: Visualization of neural module architectures. Each module is a 4-layer fully connected network, and each FC layer before the last one is followed by a leaky ReLU activation. For replacement to a node, if  $\omega_i, \omega_o$  are part of the original input, we concatenate them with analytical BRDF parameters and neural parameters as the input to the module (left); otherwise, the concatenation of analytical BRDF parameters and neural parameters are fed to the network directly (center). For the replacement of an operator, the neural module takes the results from two children nodes/modules as input (right). The output of a module is the same as the node/operator being replaced.

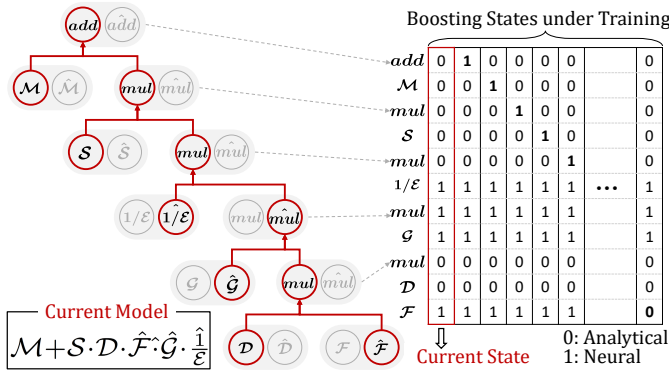


Fig. 4: Correspondence between enhancement state and computational graph. Here we take GGX BRDF as an example. Its computational graph has 11 nodes/operators in total, resulting in an 11-bit vector that represents the enhancement state: for each bit, 0 indicates that the original node/operator is used, and 1 for a neural module. The table on the right part shows all states being evaluated at the moment, whose Hamming distance is less than or equal to 1 with the current state (the first column marked in red).

convergence and slightly better reconstruction quality than Adam in our preliminary tests. The batch size is 100K. Xavier initialization is applied to all weights in neural modules. Similar to [12], all neural parameters are initialized to 0.5. For an analytical BRDF parameter, it is initialized with the mean value of the valid range in OpenSVBRDF statistics [53].

## 5 RUNTIME

### 5.1 Fitting

To use the neural enhanced model to fit a BRDF or a BTF texel, we fix the weights in all neural modules, and optimize the neural parameters as well as analytical BRDF parameters, with respect to measurements at different view and lighting directions. The initialization of these parameters is

the same as in Sec. 4.3. The fitting process is performed with a learning rate of  $10^{-3}$  over 1,000 epochs.

## 5.2 Rendering

For rendering with our model, we start with the existing shader code of the original analytical model. Next, according to the final enhancement state, we modify the parts of the code with simple implementations of small MLPs, which mainly involve vector addition and multiplication operations. The modified shader takes neural parameters and analytical parameters as input, and outputs a BRDF value for a given pair of  $(\omega_i, \omega_o)$ . For importance sampling, we directly re-fit the data samples to the original analytical model based on the code of [54], and use the corresponding importance sampling method, similar to existing work [11], [45]. It will be interesting future work to explore neural importance sampling (i.e., NeuSample [55]) without refitting: one possible approach is to learn to predict analytical lobe mixtures for our enhanced model. We render a standard, public-domain material ball of [56], and evaluate the reconstruction quality using SSIM and  $\Delta E_{ITP}$  [57].

## 6 DATA PREPARATION

We train on a variety of synthetic and captured data sources. The total number of training samples is over 28 million, consisting of the following three parts.

**Isotropic BRDFs.** We use 100 measured BRDFs in the MERL database [3]. We draw  $10^5$  random directional pairs according to [58], and use them to query the measurement values from MERL. The total number of training samples is  $100 \times 100,000 = 10.0M$ .

**Anisotropic BRDFs.** We sample two types of anisotropic BRDF from OpenSVBRDF [53]. First, we randomly sample 100 parametric GGX BRDFs based on statistics across the dataset. Then we render  $10^5$  values per BRDF as virtual measurements using the same sampling method mentioned above. Similar to [59], each measurement is perturbed with a multiplicative Gaussian noise ( $\mu = 1, \sigma = 0.1$ ) to simulate the acquisition noise.

Second, we randomly sample 300 neural representations from the dataset. We recover the lumitexel parameterized by their device from the neural representation, which contains 12,288  $\omega_i$  and one fixed  $\omega_o$ . We further remove the device-specific form-factor from the lumitexel to obtain the BRDF values as training samples. The total number of training samples from OpenSVBRDF is  $100 \times 10^5 + 300 \times 12,288 = 13.7M$ .

**BTF.** We randomly sample 200 ABRDFs from the first 11 samples of each category in the UBO BTF dataset [60], with  $151 \times 151$  pairs of  $(\omega_i, \omega_o)$  for each ABRDF. The total number of training samples is  $200 \times 151 \times 151 = 4.5M$ .

## 7 RESULTS & DISCUSSIONS

All experiments are conducted on a workstation with dual Intel Xeon 4210 CPUs, 256GB DDR4 memory, and a single NVIDIA GeForce RTX 3090 GPU. It takes about 20 hours to learn our enhanced model with 210 epochs. We render the images with Mitsuba [61], using BRDF importance sampling with 4,096 samples per pixel.



Fig. 5: Comparisons with state-of-the-art BRDF representations on measured BRDFs from EPFL [14] (top 3 rows) and UTIA [62] (bottom 2 rows) datasets. From the first column to last, the ground-truths, the results of our enhanced GGX, the original GGX BRDF [13], the FULL model in [15], Löw et al. [35], NBRDF [11], and NLBRDF [12]. Quantitative errors in SSIM and  $\Delta E_{ITP}$  ( $\times 10^3$ ) are reported at the bottom-left of each related image, while the error map is visualized at the bottom-right.

Our main result is the enhanced GGX model defined as  $\mathcal{M} + \mathcal{S} \cdot \mathcal{D} \cdot \hat{\mathcal{F}} \cdot \hat{\mathcal{G}} \cdot (\frac{1}{\mathcal{E}})$ . Here a term with a hat indicates a replacement with a neural module. Altogether 3 computational nodes and 1 operator are replaced with neural modules, resulting in approximately 7K trainable weights. The parameters of the new model are original 12 analytical BRDF parameters ( $\mathbf{n}$ ,  $\mathbf{t}_\theta$ ,  $\rho_d$ ,  $\rho_s$ ,  $\alpha_x$ ,  $\alpha_y$ ,  $F_0$ ) and 27 neural parameters, the total number of which is 39. In addition, the network size of the enhanced GGX model is 26.45KB.

Performance-wise, it takes 27.3s to fit our enhanced GGX model with respect to  $10^5$  measurements at different view and lighting directions. It takes 34.2s for fitting the same data with the original GGX model. The rendering speed in Mitsuba is  $13.68 \times 10^6 / 21.83 \times 10^6$  rays/s with our enhanced model and the original one, respectively. While optimizing rendering performance is not the goal of this paper, we can add an extra term (which approximates runtime efficiency) to the loss function to account for it in future work.

### 7.1 Comparisons

We compare our enhanced GGX model with state-of-the-art techniques using high-quality measured materials, including 51 isotropic and 11 anisotropic BRDFs from the EPFL dataset [14] (a total of 62 materials) and the materials from the UTIA dataset [46], as our validation data. Such data are never seen by any of the models for fairness. Moreover, while many of the existing methods are specialized in handling a certain type of BRDF or BTF, our model is **trained once for all**. We do not fine-tune the model for different tasks. In Fig. 16 and the supplemental material, we quantitatively and qualitatively show the fitting quality of various models across all isotropic and anisotropic validation materials from [14]. Please also refer to the accompanying video for animated results.

**Isotropic BRDFs.** In Fig. 5, we compare the reconstruction quality of isotropic BRDFs against state-of-the-art analytical models [13], [15], [35] and neural ones [11], [12]. We collect  $10^5$  BRDF values at different pairs of  $\omega_i, \omega_o$ , according to the sampling method described in Sec. 6, and use the same data as input to each model. As shown in the figure,

even the latest analytical models cannot always produce accurate fitting results. The network of NLBRDF [12] with over  $10^9$  weights deviates considerably from the ground-truth in certain cases. This is due to the inherent spectral bias and lack of positional encoding in NLBRDF, making it challenging to reconstruct high-frequency highlights [63], [64]. Consequently, mere network scaling yields smooth local minima, whereas our analytical backbone serves as a robust inductive bias for capturing sharp details in the angular domain. The top performers are our model and a state-of-the-art dedicated network, NBRDF [11]. While NBRDF trains a different network for each material, our approach optimizes the parameters for a fixed network. Our enhanced GGX substantially improves the original GGX, producing a more accurate fit to the ground-truth appearance.

**Anisotropic BRDFs.** In Fig. 6, we compare the reconstruction quality of anisotropic BRDFs against NBRDF [11]. Note that we no longer compare with most of the methods in Fig. 5, because they are limited to isotropic BRDFs only. Since our model preserves some of the prior knowledge from its input analytical model, our generalization ability is stronger than a purely data-driven approach: we achieve higher-quality fitting results than theirs, when using the same number of input measurements.

We perform further comparisons with NBRDF [11] in Fig. 7. We find that their approach is sensitive to the sampling strategy. Using anisotropic sampling angles  $(\theta_h, \phi_h, \theta_d, \phi_d)$  for isotropic data leads to poor quality in their results. In addition, fitting the retro-reflection directional samples from EPFL dataset [14] leads to the results clearly deviating from the ground-truth (e.g., mismatching highlight shape). In both cases, we produce fitting results closer to the ground-truth, demonstrating our robustness to different sampling strategies.

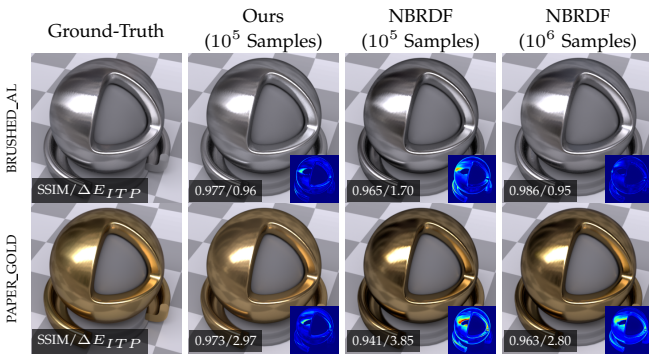


Fig. 6: Comparisons with NBRDF [11] on anisotropic BRDFs from EPFL dataset [14]. From the first column to last, ground-truths, reconstruction results of our enhanced GGX using  $10^5$  input measurements, results of NBRDF using  $10^5$  measurements, and using  $10^6$  measurements, respectively. Quantitative errors in SSIM and  $\Delta E_{ITP}$  ( $\times 10^3$ ) are reported at the bottom-left of each related image, and the error map is shown at the bottom-right.

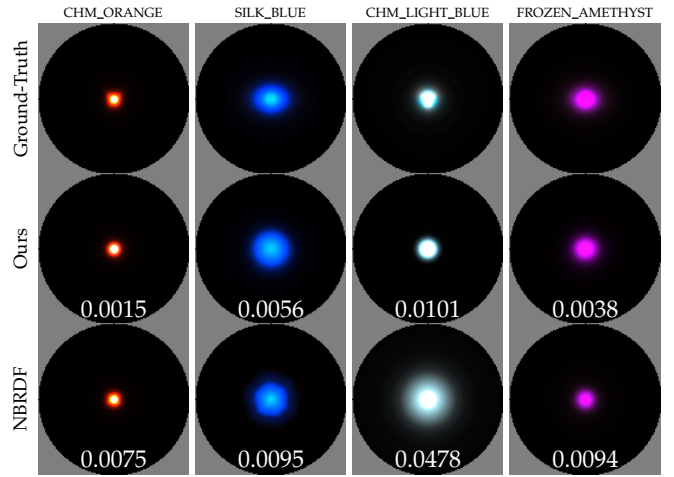


Fig. 7: Comparisons with NBRDF [11] on different sampling strategies. Each image is a BRDF slice with a fixed  $w_o$ . The first row are ground-truths, followed by the results of our enhanced GGX and NBRDF. The first two columns use anisotropic sampling angles  $(\theta_h, \phi_h, \theta_d, \phi_d)$  as input for isotropic BRDF. The last two columns show results using the retro-reflection directional samples as input. Numerical error is reported at the bottom of each related image, computed as the mean absolute value of the logarithmic loss.

## 7.2 Evaluations

**Enhancement Process.** In Fig. 1, we further visualize the process of neural enhancement of the GGX model. The order of enhanced terms is  $\mathcal{E}$ ,  $\mathcal{G}$ ,  $\times$ (of  $\mathcal{F} \times \mathcal{G}$ ), and  $\mathcal{F}$ . This progression shares similarities with the historical development of BRDF models, such as the improvement of  $\mathcal{F}$ ,  $\mathcal{G}$ , and  $\mathcal{E}$  in papers like [28] and [65]. Our framework automates this previously manual process of identifying and improving various terms in an existing analytical model.

**Enhancing Other Models.** In theory, our framework can improve any analytical BRDF model. We show the increased quality of the enhanced versions of Cook-Torrance/Ward BRDF and one model proposed in GenBRDF [15] in Fig. 15. The average SSIM for all BRDFs from the EPFL dataset [14] across three BRDF models and their enhanced versions is also shown in Fig. 17.

**Residual Networks.** A straightforward alternative to enhancing an analytical model is to append a residual neural network at its output. As shown in Fig. 8, we compare our method against two such residual MLPs: one with a similar size as our enhanced model (architecture:  $\{32, 64, 64, 16\}$ ) and another five times larger ( $\{32, 64, 128, 128, 64\}$ ). Both residual networks take as input the concatenation of the lighting-view direction pair, analytical BRDF parameters, and neural BRDF parameters, similar to the first input configuration shown in Fig. 3. The analytical and neural BRDF parameters, along with the residual network, are jointly optimized to minimize the loss between the output and the ground truth. Despite the larger capacity, neither residual network matches the accuracy of our method. This demonstrates the value of our framework: the rear-end of an analytical model might not be the best place to plug in neural modules; our framework automatically explores and

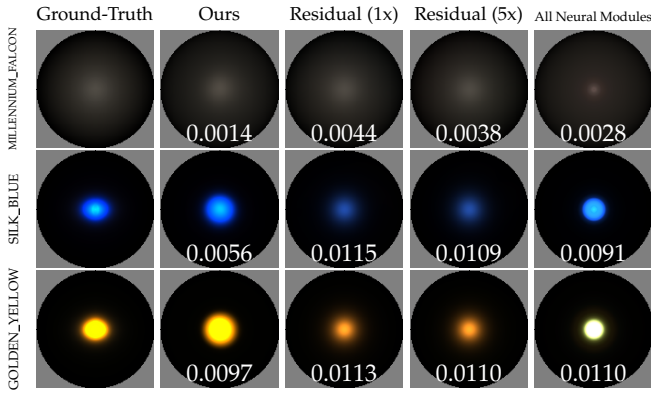


Fig. 8: Comparisons with residual networks and all neural modules. From the left column to right, the ground-truths, results of our enhanced GGX, GGX model appended with a residual network roughly the size of our model ( $\{32,64,64,16\}$ ), another residual network with about 5 times the size ( $\{32,64,128,128,64\}$ ) and a model with all nodes/operators replaced by neural modules. Quantitative error is reported at the bottom of each related image, computed as the mean absolute value of the logarithmic loss.

finds suitable locations for neural replacements.

**All Neural Modules.** We compare with a model where all nodes and operators are replaced with neural modules. In Fig. 8, despite its larger network size, our enhanced model outperforms this all-neural-node model in all materials. Note that such a model also loses the semantics of parameters of the original analytic model.

**Neural Module Size.** We evaluate the impact of neural module size over reconstruction quality both quantitatively (Fig. 9) and qualitatively (Fig. 10). We test four 4-layer MLPs with different sizes/architectures:  $\{45, 8, 16, 8\}$ ,  $\{45, 16, 16, 16\}$ ,  $\{45, 16, 32, 16\}$ , and  $\{45, 32, 32, 32\}$ . Our current choice strikes a good balance between fitting accuracy and network size.

**Neural Parameter Number.** We further investigate the impact of the number of neural parameters over reconstruction quality, qualitatively (Fig. 13) and quantitatively (Fig. 9). We test 3 different neural BRDF parameters. Our current choice is made after balancing reconstruction quality and the footprint of neural parameters.

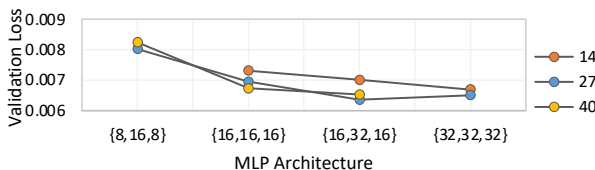


Fig. 9: Validation losses of different variants of our model. The horizontal axis displays different neural module architectures, ordered in increasing size. We also validate different parameter numbers.

**Neural Module Granularity.** In Fig. 11, we show two enhanced models with different module granularity: one model allows both nodes and operators to be enhanced,

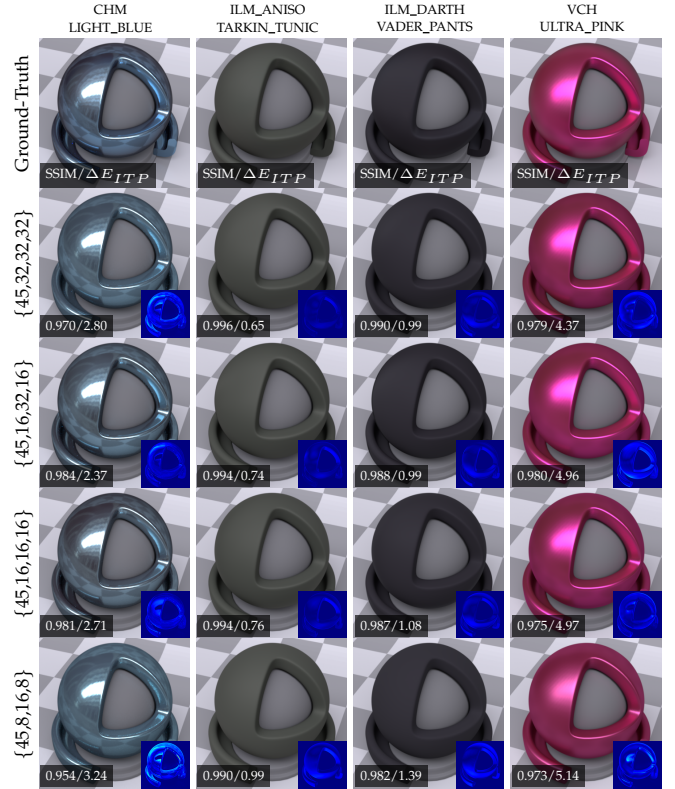


Fig. 10: Impact of neural module size over reconstruction quality. From the top row to the bottom, the ground-truths, and reconstruction results from our enhanced GGX with decreasing neural module size. The architecture of each neural module is indicated on the left of each row. Quantitative errors in SSIM and  $\Delta E_{ITP}$  ( $\times 10^3$ ) are reported at the bottom-left of each related image, and the error map at the bottom-right.

while the other only nodes replaceable with neural modules. The former shows a higher reconstruction quality due to its smaller granularity.

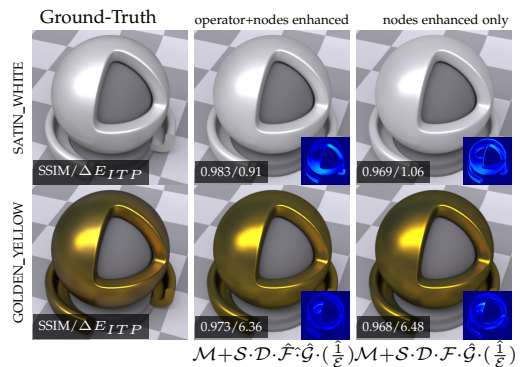


Fig. 11: Impact of module granularity over reconstruction. From the first column to last, the ground-truths, GGX model with nodes and operators enhanced, GGX model with nodes enhanced only. Quantitative errors in SSIM and  $\Delta E_{ITP}$  ( $\times 10^3$ ) are reported at the bottom-left of each related image, and the error map is shown at the bottom-right.

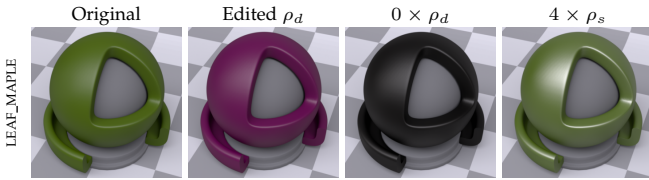


Fig. 12: Editing of our enhanced GGX model. From the first image to last, original, changing  $\rho_d$  to a different color, setting  $\rho_d$  to 0, and increasing  $\rho_s$  by  $4\times$ , respectively.

**Number of Epochs between Enhancement State Changes.** We evaluate the impact of the number of epochs between consecutive state changes (Sec. 4.2). When we set the number to 10, the order of boosted terms is  $\mathcal{E}, \mathcal{G}, \mathcal{F}, \times$  (of  $\mathcal{D} \times \mathcal{F}$ ); when 30 is set, the order is  $\mathcal{E}, \mathcal{G}, \times$  (of  $\mathcal{F} \times \mathcal{G}$ ),  $\mathcal{F}$ ; when 50 is set, the order is  $\mathcal{E}, \mathcal{G}, \mathcal{F}, \times$  (of  $\mathcal{D} \times \mathcal{F}$ ). Note that the enhanced models of the three cases are almost the same, despite the differences in the number of epochs between state changes.

**Material Editing.** In Fig. 12, we show material editing results of our enhanced GGX model. Since both diffuse and specular albedos remain in the model after our enhancement, we can directly change them for editing purposes. Note that no analytical refitting is needed for importance sampling the edited materials in the figure.

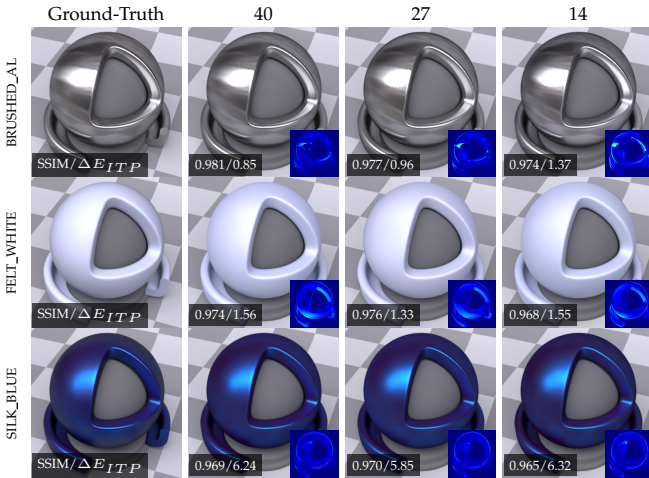


Fig. 13: Impact of neural parameter number over reconstruction quality. The first column are ground-truths, and the results of our enhanced GGX with different neural parameter numbers. Quantitative errors in SSIM and  $\Delta E_{ITP}$  ( $\times 10^3$ ) are reported at the bottom-left of each related image, and the error map is shown at the bottom-right.

## 8 LIMITATIONS & FUTURE WORK

Our work is subject to a number of limitations. First, a neural enhanced model cannot generalize to cases that (1) cannot be represented by the input analytical model, and (2) are substantially different from the training data, as shown in examples in Fig. 14 and the supplemental material. Next, our framework takes as input an analytical model, and does

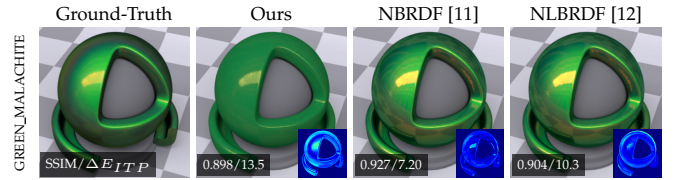


Fig. 14: Limitation. We cannot accurately fit a sample, which cannot be well represented by the input analytical model, and substantially deviates from the training data, such as the color-changing BRDF here. While NBRDF [11] can produce a good fitting, its network is dedicated to a particular BRDF only.

not propose new computation graphs. It will be interesting to combine existing generative approaches (e.g., [15]) with ours, to further pursue more efficient models. In addition, unlike the analytical parameters, the neural ones do not have immediate semantics. We are interested in applying related techniques [45], [66] to support material editing with neural parameters. Finally, while we obtain high-quality visual results with no explicit constraint on energy conservation or reciprocity, it might be useful to strictly enforce the physical correctness.

In the future, we hope that our key idea of neural enhancement can offer a novel, alternative approach other than pure analytical or neural methods, which may bring unique benefits to other fields (e.g., procedural modeling), by making more use of domain knowledge embedded in existing models. Moreover, we are intrigued in automatically determining the granularity of computation nodes from an input analytical model, as well as applying new tools (e.g., [67]) to further improve the efficiency of neural modules. Finally, it is a promising direction to combine our idea with neural architecture search.

## ACKNOWLEDGMENTS

This work is partially supported by NSF China (62332015, 62227806 & 62421003), the Fundamental Research Funds for the Central Universities (226-2023-00145), a gift from Adobe, the XPLOER PRIZE, Information Technology Center, State Key Lab of CAD&CG, Zhejiang University, the Central Government-Guided Fund for Local Science and Technology Development (2025ZY01034) and Provincial Key Laboratory Program (2025E10048).

## REFERENCES

- [1] J. Dorsey, H. Rushmeier, and F. Sillion, *Digital Modeling of Material Appearance*. Elsevier, 2010.
- [2] F. E. Nicodemus, "Directional reflectance and emissivity of an opaque surface," *Applied optics*, vol. 4, no. 7, pp. 767–775, 1965.
- [3] W. Matusik, H. Pfister, M. Brand, and L. McMillan, "A data-driven reflectance model," *ACM Trans. Graph.*, vol. 22, p. 759–769, jul 2003.
- [4] B. T. Phong, "Illumination for computer generated pictures," *Commun. ACM*, vol. 18, p. 311–317, jun 1975.
- [5] G. J. Ward, "Measuring and modeling anisotropic reflection," in *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pp. 265–272, 1992.

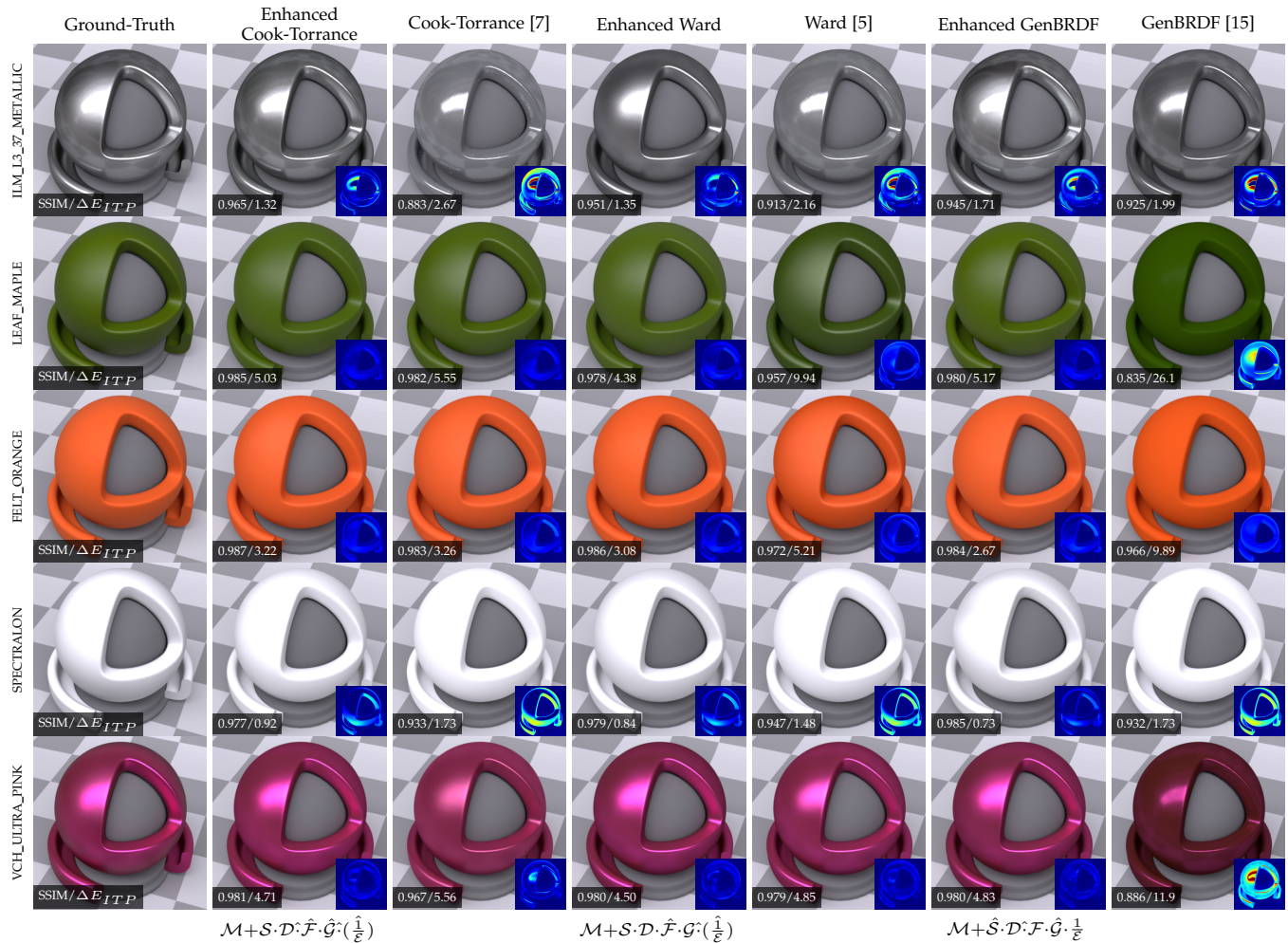


Fig. 15: Neural enhancement of other analytical BRDF models on measured BRDFs from EPFL dataset [14]. From the first column to last, the ground-truths, the results of enhanced/original Cook-Torrance model, enhanced/original Ward model, and enhanced/original FULL model in GenBRDF [15], respectively. Quantitative errors in SSIM and  $\Delta E_{ITP}$  ( $\times 10^3$ ) are reported at the bottom-left of each related image, and the error map is shown at the bottom-right. The enhanced model for each analytical one is provided at the bottom of the corresponding column.

[6] E. P. Lafortune, S.-C. Foo, K. E. Torrance, and D. P. Greenberg, "Non-linear approximation of reflectance functions," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 117–126, 1997.

[7] R. L. Cook and K. E. Torrance, "A reflectance model for computer graphics," *ACM Transactions on Graphics (ToG)*, vol. 1, no. 1, pp. 7–24, 1982.

[8] M. Oren and S. K. Nayar, "Generalization of lambert's reflectance model," in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pp. 239–246, 1994.

[9] M. Ashikhmin and P. Shirley, "An anisotropic phong brdf model," *Journal of graphics tools*, vol. 5, no. 2, pp. 25–32, 2000.

[10] A. Ngan, F. Durand, and W. Matusik, "Experimental analysis of brdf models," *Rendering Techniques*, vol. 2005, no. 16th, p. 2, 2005.

[11] A. Sztrajman, G. Rainer, T. Ritschel, and T. Weyrich, "Neural brdf representation and importance sampling," in *Computer Graphics Forum*, vol. 40, pp. 332–346, Wiley Online Library, 2021.

[12] J. Fan, B. Wang, M. Hasan, J. Yang, and L.-Q. Yan, "Neural layered brdfs," in *ACM SIGGRAPH 2022 Conference Proceedings, SIGGRAPH '22*, (New York, NY, USA), Association for Computing Machinery, 2022.

[13] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance, "Microfacet models for refraction through rough surfaces," in *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pp. 195–206, 2007.

[14] J. Dupuy and W. Jakob, "An adaptive parameterization for efficient material acquisition and rendering," *ACM Transactions on graphics (TOG)*, vol. 37, no. 6, pp. 1–14, 2018.

[15] A. Brady, J. Lawrence, P. Peers, and W. Weimer, "genbrdf: Discovering new analytic brdfs with genetic programming," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, pp. 1–11, 2014.

[16] J. Fan, B. Wang, M. Hasan, J. Yang, and L.-Q. Yan, "Neural biplane representation for btf rendering and acquisition," in *ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH '23*, (New York, NY, USA), Association for Computing Machinery, 2023.

[17] D. Gao, X. Li, Y. Dong, P. Peers, K. Xu, and X. Tong, "Deep inverse rendering for high-resolution svbrdf estimation from an arbitrary number of images," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 134–1, 2019.

[18] L. Shi, B. Li, M. Hašan, K. Sunkavalli, T. Boubekur, R. Mech, and W. Matusik, "Match: differentiable material graphs for procedural material capture," *ACM Trans. Graph.*, vol. 39, Nov. 2020.

[19] P. Guerrero, M. Hašan, K. Sunkavalli, R. Měch, T. Boubekur, and N. J. Mitra, "Matformer: a generative model for procedural materials," *ACM Trans. Graph.*, vol. 41, July 2022.

[20] M. Weinmann and R. Klein, "Advances in geometry and reflectance acquisition," in *SIGGRAPH Asia Courses*, pp. 1:1–1:71, 2015.

[21] T. Weyrich, J. Lawrence, H. P. A. Lensch, S. Rusinkiewicz, and T. Zickler, "Principles of appearance acquisition and representation," *Found. Trends. Comput. Graph. Vis.*, vol. 4, no. 2, pp. 75–

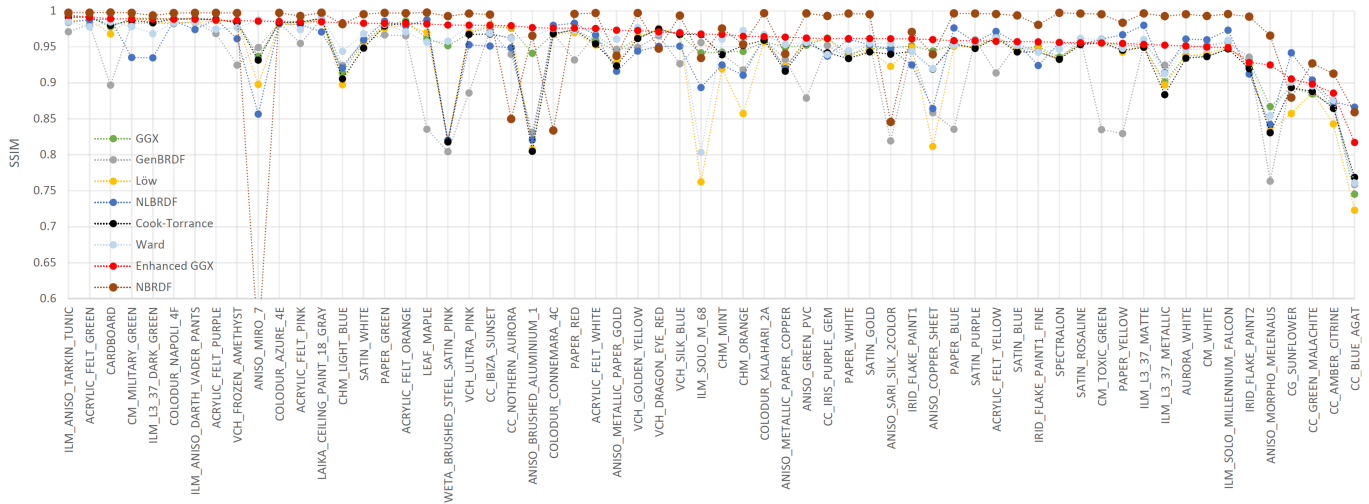


Fig. 16: We computed the SSIM of our enhanced GGX, the original GGX, GenBRDF [15], the original Cook-Torrance [7], Löw [35], Ward BRDF [5] models, NLBRDF [12] and NBRDF [11], across validation materials in [14]. Our enhanced GGX model outperforms analytical BRDFs in most cases. It also achieves better results compared to the material-independent NLBRDF, which has over  $10^9$  network parameters. Please refer to the supplemental material for more details including rendered results.

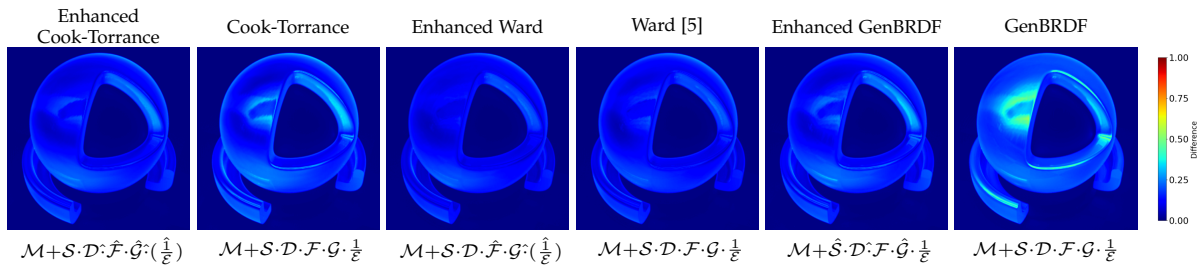


Fig. 17: Average error map for all BRDFs from the EPFL dataset [14] across three analytical BRDF models and their enhanced versions. From the first column to the last, the results correspond to the boosted/original Cook-Torrance model, the enhanced/original Ward model, and the enhanced/original FULL model in GenBRDF [15], respectively.

191, 2009.

[22] D. Guarnera, G. C. Guarnera, A. Ghosh, C. Denk, and M. Glencross, “Brdf representation and acquisition,” in *Computer Graphics Forum*, vol. 35, pp. 625–650, Wiley Online Library, 2016.

[23] Y. Dong, “Deep appearance modeling: A survey,” *Visual Informatics*, vol. 3, no. 2, pp. 59–68, 2019.

[24] B. Kavosoghafi, S. Hajisharif, E. Miandji, G. Baravdish, W. Cao, and J. Unger, “Deep svbrdf acquisition and modelling: A survey,” in *Computer Graphics Forum*, vol. 43, p. e15199, Wiley Online Library, 2024.

[25] P. Beckmann and A. Spizzichino, *The Scattering of Electromagnetic Waves from Rough Surfaces*. Goford: Pergamon Press, 1963. [distributed in the Western Hemisphere by Macmillan, New York].

[26] K. E. Torrance and E. M. Sparrow, “Theory for off-specular reflection from roughened surfaces,” *Journal of the Optical Society of America*, vol. 57, no. 9, pp. 1105–1114, 1967.

[27] J. F. Blinn, “Models of light reflection for computer synthesized pictures,” in *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, pp. 192–198, 1977.

[28] C. Schlick, “An inexpensive brdf model for physically-based rendering,” in *Computer graphics forum*, vol. 13, pp. 233–246, Wiley Online Library, 1994.

[29] B. Smith, “Geometrical shadowing of a random rough surface,” *IEEE transactions on antennas and propagation*, vol. 15, no. 5, pp. 668–671, 1967.

[30] M. Ashikhmin, “A microfacet-based brdf generator,” *Proc. ACM SIGGRAPH 2000*, New Orleans, USA, July, 2000.

[31] T. Trowbridge and K. P. Reitz, “Average irregularity representation of a rough surface for ray reflection,” *Journal of the Optical Society of America*, vol. 65, no. 5, pp. 531–536, 1975.

[32] H. Wu, J. Dorsey, and H. Rushmeier, “A sparse parametric mixture model for btf compression, editing and rendering,” in *Computer Graphics Forum*, vol. 30, pp. 465–473, Wiley Online Library, 2011.

[33] M. Kurt, L. Szirmay-Kalos, and J. Krivánek, “An anisotropic brdf model for fitting and monte carlo rendering,” *ACM SIGGRAPH Computer Graphics*, vol. 44, no. 1, pp. 1–15, 2010.

[34] M. M. Bagher, C. Soler, and N. Holzschuch, “Accurate fitting of measured reflectances using a shifted gamma micro-facet distribution,” in *Computer Graphics Forum*, vol. 31, pp. 1509–1518, Wiley Online Library, 2012.

[35] J. Löw, J. Kronander, A. Ynnerman, and J. Unger, “Brdf models for accurate and efficient rendering of glossy surfaces,” *ACM Transactions on Graphics (TOG)*, vol. 31, no. 1, pp. 1–14, 2012.

[36] G. Rainer, W. Jakob, A. Ghosh, and T. Weyrich, “Neural btf compression and interpolation,” in *Computer Graphics Forum*, vol. 38, pp. 235–244, Wiley Online Library, 2019.

[37] E. Trunz, M. Weinmann, S. Merzbach, and R. Klein, “Efficient structuring of the latent space for controllable data reconstruction and compression,” *Graphics and Visual Computing*, vol. 7, p. 200059, 2022.

[38] F. Gokbudak, A. Sztrajman, C. Zhou, F. Zhong, R. Mantiuk, and C. Oztireli, “Hypernetworks for generalizable brdf representation,” in *European Conference on Computer Vision*, pp. 73–89, Springer, 2024.

[39] T. Zeltner\*, F. Rousselet\*, A. Weidlich\*, P. Clarberg\*, J. Novák\*, B. Bitterli\*, A. Evans, T. Davidovič, S. Kallweit, and A. Lefohn, “Real-time neural appearance models,” *ACM Transactions on*

- Graphics, vol. 43, no. 3, pp. 1–17, 2024.
- [40] B. Hu, J. Guo, Y. Chen, M. Li, and Y. Guo, “Deepbrdf: A deep representation for manipulating measured brdf,” in *Computer Graphics Forum*, vol. 39, pp. 157–166, Wiley Online Library, 2020.
- [41] G. Rainer, A. Ghosh, W. Jakob, and T. Weyrich, “Unified neural encoding of btfs,” in *Computer Graphics Forum*, vol. 39, pp. 167–178, Wiley Online Library, 2020.
- [42] A. Kuznetsov, “Neumip: Multi-resolution neural materials,” *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, 2021.
- [43] B. Xue, S. Zhao, H. W. Jensen, and Z. Montazeri, “A hierarchical architecture for neural materials,” *Computer Graphics Forum*, vol. 43, no. 6, p. e15116, 2024.
- [44] M. Fischer and T. Ritschel, “Metappearance: Meta-learning for visual appearance reproduction,” *ACM Transactions on Graphics (TOG)*, vol. 41, no. 6, pp. 1–13, 2022.
- [45] C. Zheng, R. Zheng, R. Wang, S. Zhao, and H. Bao, “A compact representation of measured brdfs using neural processes,” *ACM Trans. Graph.*, vol. 41, nov 2021.
- [46] A. Kuznetsov, M. Hašan, Z. Xu, L.-Q. Yan, B. Walter, N. K. Kalantari, S. Marschner, and R. Ramamoorthi, “Learning generative models for rendering specular microgeometry,” *ACM Trans. Graph.*, vol. 38, Nov. 2019.
- [47] W. Shi, J. Dorsey, and H. Rushmeier, “Learning-based inverse bi-scale material fitting from tabular brdfs,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 4, pp. 1810–1823, 2022.
- [48] M. Schmidt and H. Lipson, “Distilling free-form natural laws from experimental data,” *science*, vol. 324, no. 5923, pp. 81–85, 2009.
- [49] B. K. Petersen, M. Landajuela, T. N. Mundhenk, C. P. Santiago, S. K. Kim, and J. T. Kim, “Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients,” *arXiv preprint arXiv:1912.04871*, 2019.
- [50] S.-M. Udrescu, A. Tan, J. Feng, O. Neto, T. Wu, and M. Tegmark, “Ai feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 4860–4871, 2020.
- [51] L. Biggio, T. Bendinelli, A. Neitz, A. Lucchi, and G. Parascandolo, “Neural symbolic regression that scales,” in *International Conference on Machine Learning*, pp. 936–945, Pmlr, 2021.
- [52] T. Koshy, *Discrete mathematics with applications*. Elsevier.
- [53] X. Ma, X. Xu, L. Zhang, K. Zhou, and H. Wu, “Opensvbrdf: A database of measured spatially-varying reflectance,” *ACM Transactions on Graphics (TOG)*, vol. 42, no. 6, pp. 1–14, 2023.
- [54] J. Dupuy, “Photorealistic surface rendering with microfacet theory,” 2015.
- [55] B. Xu, L. Wu, M. Hasan, F. Luan, I. Georgiev, Z. Xu, and R. Ramamoorthi, “Neusample: Importance sampling for neural materials,” in *ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH ’23*, (New York, NY, USA), Association for Computing Machinery, 2023.
- [56] A. Mazzone and C. Rydalch, “Standard shader ball: A modern and feature-rich render test scene,” in *SIGGRAPH Asia 2023 Technical Communications, SA ’23*, (New York, NY, USA), Association for Computing Machinery, 2023.
- [57] B. Kavosoghafi, R. K. Mantiuk, S. Hajisharif, E. Miandji, and J. Unger, “Perceived Quality of BRDF Models,” *Computer Graphics Forum*, 2025.
- [58] S. M. Rusinkiewicz, “A new change of variables for efficient brdf representation,” in *Rendering Techniques’ 98: Proceedings of the Eurographics Workshop in Vienna, Austria, June 29–July 1, 1998*, pp. 11–22, Springer, 1998.
- [59] X. Ma, Y. Yu, H. Wu, and K. Zhou, “Efficient reflectance capture with a deep gated mixture-of-experts,” *IEEE TVCG*, pp. 1–12, 2023.
- [60] M. Weinmann, J. Gall, and R. Klein, “Material classification based on training data synthesized using a bf database,” in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part III*, 13, pp. 156–171, Springer, 2014.
- [61] W. Jakob, S. Speierer, N. Roussel, M. Nimier-David, D. Vicini, T. Zeltner, B. Nicolet, M. Crespo, V. Leroy, and Z. Zhang, “Mitsuba 3 renderer,” 2022. <https://mitsuba-renderer.org>.
- [62] J. Filip and R. Vávra, “Template-based sampling of anisotropic brdfs,” in *Computer Graphics Forum*, vol. 33, pp. 91–99, Wiley Online Library, 2014.
- [63] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, “On the spectral bias of neu-

ral networks,” in *International conference on machine learning*, pp. 5301–5310, PMLR, 2019.

- [64] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [65] M. Ashikmin, S. Premoze, and P. Shirley, “A microfacet-based brdf generator,” in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 65–74, 2000.
- [66] Y. Guo, C. Smith, M. Hašan, K. Sunkavalli, and S. Zhao, “Materialgan: Reflectance capture using a generative svbrdf model,” *arXiv preprint arXiv:2010.00114*, 2020.
- [67] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, “Kan: Kolmogorov-arnold networks,” *arXiv preprint arXiv:2404.19756*, 2024.



**Xuanzhe Shen** is a master student in the State Key Lab of CAD & CG, Zhejiang University. He received his B.Eng. degree from the same university in 2024. His research interests include appearance reconstruction and rendering.



**Xiaohe Ma** is currently a Senior Research Scientist at Meshy AI. She received her Ph.D. degree in Computer Science from the State Key Lab of CAD & CG, Zhejiang University, in 2025, and her B.Eng. degree from the School of Data and Computer Science, Sun Yat-sen University, in 2019. Her research interests include texture generation and 3D/4D generation with learning-based techniques.



**Kun Zhou** is a Cheung Kong Professor of Computer Science at Zhejiang University, the Director of the State Key Lab of CAD&CG and Hangzhou Research Institute of Holographic and AI Technology. He received his Ph.D. degree from Zhejiang University, and then spent six years with Microsoft Research Asia, serving as a lead researcher of the graphics group before returning to Zhejiang University. He was named one of the world’s top 35 young innovators by MIT Technology Review (2011), and received an Asigraphics Outstanding Technical Contributions Award (2022) and an ACM SIGGRAPH Test-of-Time Award (2024). He is a Fellow of IEEE and ACM.



**Hongzhi Wu** is a professor in the State Key Lab of CAD & CG, Zhejiang University. He received B.Sc. in computer science from Fudan University, and Ph.D. in computer science from Yale University. His current research interests include high-density illumination multiplexing devices and differentiable acquisition. Hongzhi is a recipient of Excellent Young Scholars, NSF China. He is on the editorial board of IEEE TVCG.