arXiv:2411.12015v2 [cs.GR] 10 Nov 2025

M^{3}ashy: Multi-Modal Material Synthesis via Hyperdiffusion

Chenliang Zhou, Zheyuan Hu, Alejandro Sztrajman, Yancheng Cai, Yaru Liu, Cengiz Oztireli

Department of Computer Science and Technology, University of Cambridge
chenliang.zhou@cst.cam.ac.uk



Figure 1: 3D models and scenes rendered with our synthesized neural materials demonstrate visually rich results.

Abstract

High-quality material synthesis is essential for replicating complex surface properties to create realistic scenes. Despite advances in the generation of material appearance based on analytic models, the synthesis of real-world measured BRDFs remains largely unexplored. To address this challenge, we propose M^{3}ashy, a novel **m**ulti-**m**odal **ma**terial **s**ynthesis framework based on **hy**perdiffusion. M^{3}ashy enables high-quality reconstruction of complex real-world materials by leveraging neural fields as a compact continuous representation of BRDFs. Furthermore, our multi-modal conditional hyperdiffusion model allows for flexible material synthesis conditioned on material type, natural language descriptions, or reference images, providing greater user control over material generation. To support future research, we contribute two new material datasets and introduce two BRDF distributional metrics for more rigorous evaluation. We demonstrate the effectiveness of M^{3}ashy through extensive experiments, including a novel statistics-based constrained synthesis, which enables the generation of materials of desired categories.

1 Introduction

Material synthesis plays a crucial role in visual computing, enabling the creation of realistic material appearances for applications in scene understanding (Gupta, Arbelaez, and Malik 2013), material recognition (Bell et al. 2015), intrinsic image decomposition (Bousseau, Paris, and Durand 2009), generative image synthesis (Karras, Laine, and Aila 2019), and physics-based vision for simulation (Wu et al. 2017). Material appearance is commonly modeled by the *bidirectional reflectance distribution function (BRDF)*. Although there has been significant progress (*e.g.*, Gatys, Ecker, and Bethge 2015; Zhou et al. 2018) in generative modeling of analytic BRDFs (Phong 1975; Walter et al. 2007), there is a lack of work focusing on that of measured ones. For the analytic BRDFs, the actual per-point reflectance model is typically relatively simple and low-dimensional (Ngan, Durand, and Matusik 2005; Guarnera et al. 2016). In contrast, a measured BRDF is tabulated from real-world capture and can be substantially higher-dimensional (*e.g.*, Matusik et al. 2003), with the ability to represent complex and irregular scattering behaviors that exceed the expressiveness of analytic models (Ngan, Durand, and Matusik 2005). However, the high-dimensionality often hinders the performance of learning-based methods. To address this gap, we propose M^{3}ashy, a framework for realistic material generation that leverages neural fields (Sztrajman et al. 2021) as an alternative low-dimensional continuous representation for material appearance that combines high-quality reconstruction with memory efficiency. This simplifies the learning process, enabling the model to capture the underlying material distribution more efficiently.

An additional challenge in material synthesis is the lack of robust quantitative metrics for evaluating synthesis quality, making it difficult to assess and compare different ap-

proaches, unlike generative models in other fields (Theis, van den Oord, and Bethge 2016; Betzalel et al. 2022). A final limitation is the absence of multi-modal conditioning, which would enable users to guide the synthesis process using diverse inputs, such as material type, text descriptions or reference images. This limitation reduces the flexibility and control available to artists and designers. To address these limitations, we propose a set of novel BRDF distributional metrics and leverage a multi-modal conditional hyperdiffusion model to support flexible user input.

The main contributions of this work are as follows:

- A novel material synthesis pipeline using a multi-modal conditional hyperdiffusion model that supports user-specified material generation via material type, natural language descriptions, or image references.
- A thorough evaluation of M^{3}ashy's effectiveness, including a set of novel BRDF distributional metrics and a novel constrained synthesis experiment to synthesize materials of desired categories.
- Two new datasets: AugMERL, an enhanced collection of tabulated BRDF values, and NeuMERL, a dataset of materials represented through INRs.

2 Related Work

Material modeling Material appearance has been widely modeled by the *bidirectional reflectance distribution function (BRDF)* (Nicodemus et al. 1977; Guarnera et al. 2016; Montes and Ureña 2012; Ngan, Durand, and Matusik 2005; Westin, Li, and Torrance 2004). While analytic BRDF models (Phong 1975; Cook and Torrance 1982; Walter et al. 2007; Burley 2012) offer efficient reconstruction and editing, their simplified assumptions limit the representation of complex real-world materials (Ngan, Durand, and Matusik 2005; Guarnera et al. 2016; Sztrajman et al. 2017, 2019). Data-driven approaches offer higher realism (Matusik et al. 2003), although they are often hard to manipulate and require large storage. Dimensionality reduction techniques can alleviate this issue, but at the expense of compromising material quality (Lawrence, Rusinkiewicz, and Ramamoorthi 2004; Nielsen, Jensen, and Ramamoorthi 2015). Recently, deep learning methods provide efficient, low-dimensional representations (Hu et al. 2020; Zheng et al. 2021; Sztrajman et al. 2021; Gokbudak et al. 2023; Fan et al. 2022; Guo et al. 2023). Our work leverages a neural field architecture (Sztrajman et al. 2021) for efficient, realistic material modeling. Please see Sec. A for additional related work on material acquisition and databases.

Material synthesis Material synthesis based on analytic models has been widely explored (Zhang et al. 2024; Memery, Cedron, and Subr 2023; Chen et al. 2023b,a; Hu et al. 2023; Tchapmi et al. 2022; Xu et al. 2023; Henzler et al. 2021). For data-driven representations, previous works have resourced to various strategies, including dimensionality reduction (Matusik et al. 2003; Abdi and Williams 2010), perceptual mappings (Nielsen, Jensen, and Ramamoorthi 2015; Serrano et al. 2016; Sun, Jensen, and Ramamoorthi 2018) and deep learning (Hu et al. 2020; Gokbudak et al. 2023), but these approaches do not offer a generative modeling of materials. Hu et al. (2020) use a convolutional autoencoder to learn a low-dimensional manifold from a measured BRDF database, enabling material editing. However, their BRDF representation is constrained to a fixed resolution with high storage requirements. Gokbudak et al. (2023) leverage a hypernetwork architecture to predict the weights of a neural fields representation of material appearance. Nevertheless, their approach requires sample measurements of BRDF data as input and is thus limited to sparse reconstruction. There also exist methods for text- (Xu et al. 2023; Memery, Cedron, and Subr 2023) and image-conditioned synthesis (Hu et al. 2020; Henzler et al. 2021), but these either focus on analytic materials or do not provide a generative modeling approach.

In this work, we introduce a generative approach for measured real-world material based on a multi-modal hyperdiffusion architecture. Our method generates continuous, low-dimensional representations of materials, and can be conditioned on material type, natural language, and images. In Tab. 1, we compare various state-of-the-art material modeling methods: Our approach is the first generative pipeline to support unconditional, multi-modal, and constrained synthesis of measured real-world materials while also contributing novel datasets and introducing quantitative metrics for material synthesis evaluation.

3 Methods

An overview of our material synthesis pipeline, M^{3}ashy, is shown in Fig. 2, consisting of three main stages. First, we augment the MERL dataset through RGB permutation and PCA interpolation, generating the *Augmented MERL (AugMERL)* dataset (Sec. 3.1). Next, we adopt neural fields as a low-dimensional, continuous representation for materials, fitting them to individual materials in AugMERL to create a new dataset of neural material representations, *Neural MERL (NeuMERL)*. Finally, we train a transformer-based, multi-modal hyperdiffusion model on NeuMERL to capture the complex distribution of neural materials, enabling high-fidelity and diverse synthesis through unconditional, multi-modal conditional, and constrained generation (Sec. 4).

3.1 Data Augmentation

We utilize the MERL dataset (Matusik et al. 2003), which includes 100 materials, each represented by $D_{\text{MERL}} = 90 \times 90 \times 180$ densely sampled BRDF values.

Through experimentation, we determined that 100 samples are insufficient for effective hyperdiffusion training. To address this, we augment the MERL dataset using RGB permutation and PCA interpolation. First, we permute the three color channels (RGB) of each MERL sample, yielding an expanded dataset of $100 \times 3! = 600$ samples. An example of RGB permutation is illustrated in Fig. 3.

After applying RGB permutation, we perform principal component analysis (PCA) (Abdi and Williams 2010) to reduce the dimensionality of the BRDF data from D_{MERL} to 300. In this lower-dimensional space, we perform linear interpolation to further augment the dataset, expanding it to 2400 materials. Compared to direct linear interpolation

BRDF modeling method	Measured	Generative	Type	Text	Image	CS	Datasets	Metrics
DeepBRDF (Hu et al. 2020)	✓	✗	✗	✗	✓	✗	✗	✗
Henzler et al. (2021)	✗	✓	✗	✗	✓	✗	✗	✓
MATLABER (Xu et al. 2023)	✗	✓	✗	✓	✗	✗	✗	✗
Memery, Cedron, and Subr (2023)	✗	✓	✗	✓	✗	✗	✗	✗
Gokbudak et al. (2023)	✓	✗	✗	✗	✓	✗	✗	✗
M^{3}ashy (ours)	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison of material modeling methods. Our M^{3}ashy is the first generative pipeline for measured real-world materials that supports both unconditional and multi-modal conditional synthesis guided by type, text, or image. It also enables a statistics-based constrained synthesis (CS) and introduces novel datasets and material distributional metrics.

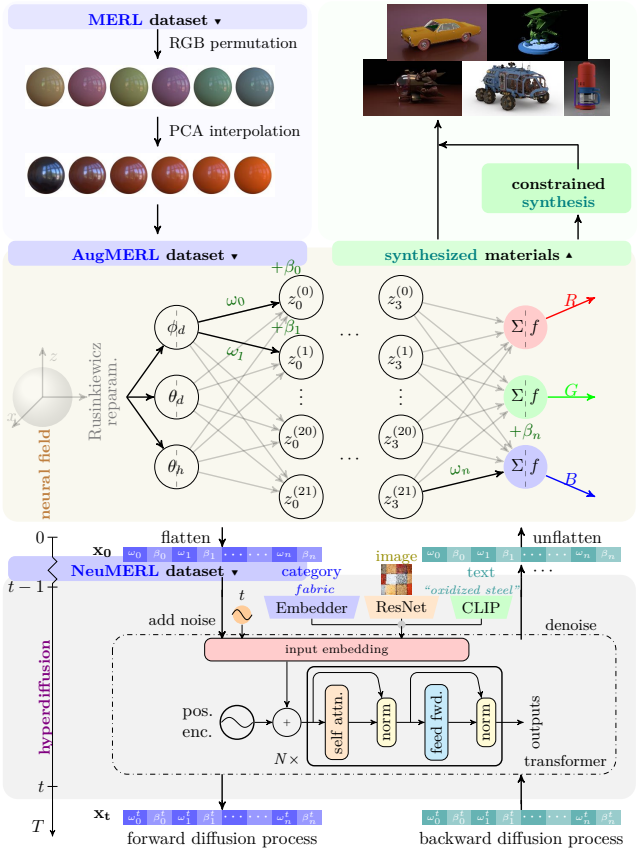


Figure 2: An overview of M^{3}ashy, our novel neural material synthesis framework, consisting of three main stages. 1 (top left): Data augmentation using RGB permutation and PCA interpolation to create an expanded dataset, *AugMERL*; 2 (middle): Neural field fitted to individual materials, resulting in *NeuMERL*, a dataset of neural material representations; and 3 (bottom): Training a multi-modal conditional hyperdiffusion on NeuMERL to enable conditional synthesis of high-quality, diverse materials guided by inputs such as material type, text descriptions, or reference images. We further propose a novel statistics-based constrained synthesis method to generate materials of a specified type (top right).

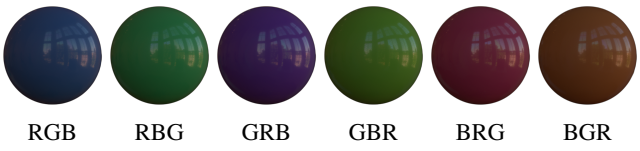


Figure 3: Six RGB permutations of the MERL material *blue acrylic*. (a) represents the original material. This permutation strategy expands the dataset by a factor of 6.

in the high-dimensional BRDF space, interpolation in PCA space is more effective in capturing the underlying structure of the BRDF data and yields perceptually accurate results (Matusik et al. 2003; Lawrence, Rusinkiewicz, and Ramamoorthi 2004; Romeiro and Zickler 2010). An example of materials generated through PCA interpolation is shown in Fig. 4. We refer to this augmented dataset as *Augmented MERL (AugMERL)*. For additional details on PCA, please see Sec. B.1 in the supplementary.

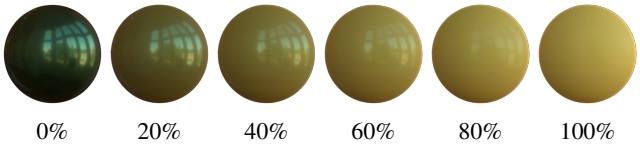


Figure 4: Linear interpolation of two MERL materials, (a) *green metallic paint* and (f) *yellow plastic*, in the PCA space.

3.2 Neural Field Fitting

Neural fields provide a low-dimensional, continuous representation for material data. Following prior work (Sztrajman et al. 2021), we overfit a compact neural field f_r^{ξ} parameterized by ξ, to each material in AugMERL. Once fitted, we treat the flattened weights of f_r^{ξ} as the material's neural representation. With 2400 materials in AugMERL, this process yields a dataset of 2400 neural material representations, which we refer to as *Neural MERL (NeuMERL)*.

Representing materials as flattened 1D vectors enables a flexible framework for modeling complex distributions, abstracting away the underlying data's dimensionality. This approach makes our pipeline adaptable to diverse data formats. In the following section, we detail three key techniques employed in fitting the neural fields.

Rusinkiewicz reparametrization In our preliminary experiments and other studies (*e.g.*, Sztrajman et al. 2021; Zhou et al. 2024), it is observed that directly using the conventional BRDF input format – namely the incident and outgoing directions $\boldsymbol{\omega}_i, \boldsymbol{\omega}_o \in \mathbb{R}^3$ – can complicate the fitting process for certain materials and occasionally introduce undesirable artifacts possibly due to the high dimensionality of the input. To address this, we employ the Rusinkiewicz reparametrization (Rusinkiewicz 1998), which defines the half and difference vectors $\mathbf{h}$ and $\mathbf{d}$ as follows:

$$\mathbf{h} \coloneqq \frac{\boldsymbol{\omega}_i + \boldsymbol{\omega}_o}{\|\boldsymbol{\omega}_i + \boldsymbol{\omega}_o\|};\ \mathbf{d} \coloneqq R_{\hat{\mathbf{b}},-\theta_\mathbf{h}} R_{\hat{\mathbf{n}},-\varphi_\mathbf{h}} \boldsymbol{\omega}_i, \tag{1}$$

where $R_{\mathbf{v},\alpha}$ denotes a rotation around the vector $\mathbf{v}$ by the angle α, $\hat{\mathbf{n}}$ is the surface normal, and $\hat{\mathbf{b}}$ is the surface binormal. This reparametrization helps improve the robustness of the neural field fitting process by addressing reciprocity constraints more directly (Sztrajman et al. 2021; Zhou et al. 2024).

We then proceed by adopting the spherical coordinates of the half and difference vectors $\mathbf{h}$ and $\mathbf{d}$, specifically $\theta_\mathbf{h}, \varphi_\mathbf{h}, \theta_\mathbf{d}, \varphi_\mathbf{d}$, as inputs to our neural fields. A further advantage of using the Rusinkiewicz reparametrization is that, since our materials are isotropic, the BRDF remains invariant with respect to $\varphi_\mathbf{h}$. Consequently, we can omit this parameter, reducing the input complexity from $(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \in \mathbb{R}^6$ to $(\theta_\mathbf{h}, \theta_\mathbf{d}, \varphi_\mathbf{d}) \in [0, \frac{\pi}{2}]^2 \times [0, \pi)$. This reparametrization enhances the efficiency of our neural field representation without sacrificing accuracy.

Mean absolute logarithmic loss The high dynamic range of BRDF values makes fitting reflectance data particularly sensitive to error distribution. For low reflectance values, even minor fitting errors can have a large impact on the loss, causing shifts in perceived "hue" in rendered images, which leads to unrealistic colors and reduced visual fidelity. To address these issues, we employ a mean absolute logarithmic loss for BRDF values (Sztrajman et al. 2021):

$$\mathcal{L}_{\text{NF}}(\boldsymbol{\xi}) \coloneqq \mathop{\mathbb{E}}_{\theta_\mathbf{h}, \theta_\mathbf{d}, \varphi_\mathbf{d}} \left[\left| \log\left(1 + f_r \cos\theta_i\right) - \log\left(1 + f_r^{\boldsymbol{\xi}} \cos\theta_i\right) \right| \right] \tag{2}$$

where f_r denotes the ground-truth BRDF, and θ_i is the polar angle of the incident direction. This loss is computed per color channel, offering a balanced approach that stabilizes training across samples with both low and high values. Consequently, it enhances the model's capability to manage dynamic reflectance variations, leading to more realistic color reproduction and improved visual fidelity.

Weight initialization Ideally, the neural materials in NeuMERL should originate from a consistent distribution. However, due to a phenomenon known as *weight symmetry* (Liao, Leibo, and Poggio 2016), we observe that different weights can yield the same neural field. For instance, swapping weights between two neurons in a hidden layer, or flipping the signs of both input and output weights for a neuron before an odd, linear, or piecewise-linear activation function like ReLU (Nair and Hinton 2010), results in an identical neural field. To address this, we propose using the optimized weights from the first fitted neural field as the initialization for subsequent neural field fittings. This approach helps align the weights across all fitted fields, promoting consistency within NeuMERL and facilitating smoother training of the hyperdiffusion model in the next stage.

3.3 Multi-Modal Conditional Hyperdiffusion

To model the complex distribution within the NeuMERL dataset, we utilize a diffusion process (Ho, Jain, and Abbeel 2020; Peebles et al. 2022; Erkoç et al. 2023). Specifically, we employ a transformer-based denoising network (Vaswani 2017), leveraging its demonstrated efficacy (Peebles et al. 2022) and its attention mechanisms allowing for an effective focus on relevant information, enhancing the network's ability to capture intricate dependencies in the data.

Our hyperdiffusion supports conditioning across three modalities: material type (represented as integers), text description, and reference images. We utilize a categorical encoding for material type, an augmented CLIP text embedding (Radford et al. 2021; Zhou, Zhong, and Oztireli 2023) for text, and a ResNet (He et al. 2016) for images. This multi-modal conditioning approach enables material synthesis to be guided by different user inputs, enhancing workflow intuitiveness and accessibility and allowing for a more smooth and accurate translation of creative vision into generated materials. For conditional sampling, we employ classifier-free guidance (CFG) (Ho and Salimans 2022).

Please refer to Secs. B.2, B.3, C.2 and D.2 in the supplementary for further details on the diffusion model, transformer, attention mechanism, and CFG, respectively.

4 Experiments

In this section, we present extensive experiments on M^{3}ashy. For additional details, please refer to the appendix: model and experiment specifics in Secs. C and D, further results in Sec. F, and more experiments in Sec. G.

4.1 Dataset

We fit neural fields to individual materials in the AugMERL dataset (Sec. 3.1), which is derived from the MERL BRDF dataset (Matusik et al. 2003). Our hyperdiffusion model is trained on the NeuMERL dataset (Sec. 3.2), which consists of neural material representations. The training-validation split is 80%-20% for each individual fitting on AugMERL and 95%-5% for NeuMERL. All samples derived from a given material remain within one split. In the constrained synthesis experiments in Sec. 4.5, statistical information is gathered from the MERL dataset (Matusik et al. 2003).

4.2 Material Distributional Metrics

We use Fréchet Inception Distance (FID) (Heusel et al. 2017) as an image-based metric to assess the quality of rendered single-view images.

To the best of our knowledge, effective metrics directly comparing material distributions are still lacking. Drawing inspiration from the metrics for point clouds (Yang et al. 2019), we introduce three novel material distributional metrics – minimum matching distance (MMD), coverage

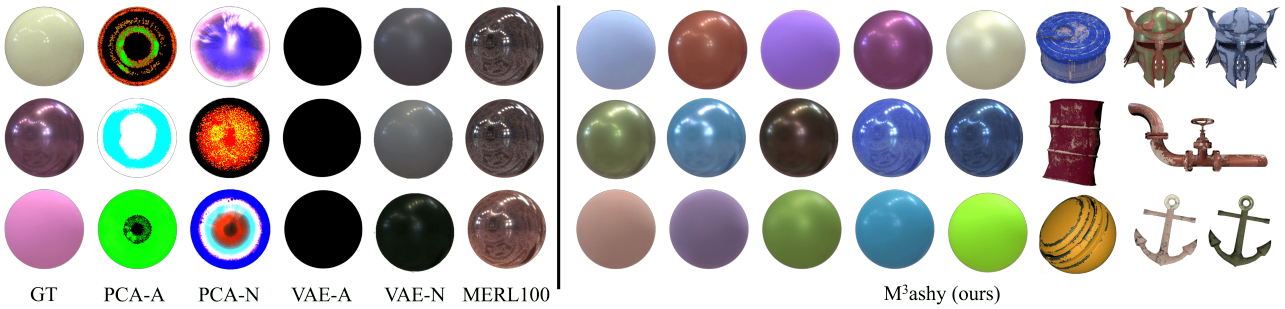


Figure 5: Material synthesis. Baseline models fail to capture the underlying distribution effectively, resulting in homogeneous outputs or severe artifacts. In contrast, M³ashy successfully captures the complex neural material distribution, achieving significantly better fidelity and diversity. Our materials also support spatially varying rendering configurations (last three columns).

(COV), and 1-nearest neighbor (1-NNA) – to evaluate the fidelity and diversity of synthesized BRDF sets $\mathcal{S}$ relative to a reference set $\mathcal{R}$. Each metric is based on an underlying distance measure $d(f_r, f_r')$ between two BRDFs.

Minimum matching distance (MMD) MMD measures the average distance from each reference BRDF to its nearest synthesized counterpart:

$$\mathcal{L}^d_{\text{MMD}}(\mathcal{R}, \mathcal{S}) \coloneqq \frac{1}{|\mathcal{R}|} \sum_{f_r \in \mathcal{R}} \min_{f_r' \in \mathcal{S}} d(f_r, f_r') \tag{3}$$

MMD evaluates the fidelity of the synthesized set relative to the reference, with a lower score indicating higher fidelity.

Coverage (COV) COV calculates the proportion of reference BRDFs that are "covered" by the synthesized set. A reference BRDF is considered covered if it is the closest neighbor to at least one synthesized BRDF:

$$\mathcal{L}^d_{\text{COV}}(\mathcal{R}, \mathcal{S}) = \frac{\left| \left\{ \operatorname*{argmin}_{f_r \in \mathcal{R}} d(f_r, f_r') \mid f_r' \in \mathcal{S} \right\} \right|}{|\mathcal{R}|} \tag{4}$$

COV assesses the diversity of the synthesized set, with a higher score reflecting better coverage.

1-nearest neighbor (1-NNA) 1-NNA is a leave-one-out metric that measures the similarity between the reference and synthesized BRDF distributions, capturing both diversity and fidelity:

$$\mathcal{L}^d_{\text{1-NNA}}(\mathcal{R}, \mathcal{S}) \coloneqq \frac{\sum_{f_r \in \mathcal{R}} \mathbb{I}\left[N_{f_r} \in \mathcal{R}\right] + \sum_{f_r' \in \mathcal{S}} \mathbb{I}[N_{f_r'} \in \mathcal{S}]}{|\mathcal{S}| + |\mathcal{R}|}, \tag{5}$$

where $\mathbb{I}[\cdot]$ is the indicator function and N_{f_r} denotes the nearest neighbor of f_r in $(\mathcal{R} \cup \mathcal{S}) - \{f_r\}$. In this metric, each sample is classified as belonging to either the reference set $\mathcal{R}$ or the synthesized set $\mathcal{S}$ based on the membership of its nearest neighbor. If $\mathcal{R}$ and $\mathcal{S}$ are drawn from the same underlying distribution, the classifier's accuracy will approach 50% with a large sample size.

Each metric can be computed using an underlying distance d. Potential options include rendering-based metrics such as root mean squared error (RMSE), peak signal-to-noise ratio (PSNR), and structural similarity index measure (SSIM) (Wang et al. 2004). Since higher values in PSNR and SSIM correspond to greater similarity, we negate these values – resulting in NegPSNR and NegSSIM, respectively – to make them plausible distance functions To directly assess the distance between two BRDFs without relying on renderings, we also introduce the following BRDF L1 distance:

$$d_{\text{BRDF-L1}} \coloneqq \mathop{\mathbb{E}}_{\theta_\mathbf{h}, \theta_\mathbf{d}, \varphi_\mathbf{d}} \Big[\left| f_r - f_r' \right| \Big]; \tag{6}$$

For further background on the image-based metrics and validation of the proposed material distributional metrics, please refer to Sec. E in the supplementary.

4.3 Unconditional Synthesis

We begin by presenting the results of unconditional synthesis of neural materials using M³ashy. We compare our approach with the PCA-based method of Nielsen, Jensen, and Ramamoorthi (2015) and with the sparse reconstruction model of Gokbudak et al. (2023), which uses a hypernetwork to model measured tabular BRDF data. As Gokbudak et al.'s method is not generative, we extend it with a variational autoencoder (VAE) to enable comparison. Both methods are applied to the AugMERL (-A) and NeuMERL (-N) datasets, resulting in four baselines: VAE-A, VAE-N, PCA-A, and PCA-N. We also develop another baseline, MERL100, that represents our method trained on the original MERL dataset to demonstrate the effectiveness of our data augmentation.

Table 2 provides a detailed comparison of baselines with our proposed metrics, while Figs. 1 and 5 showcase renderings of these materials across various geometries and scenes. To achieve more complex visual effects, our synthesized materials support rendering with bump or normal maps, as well as spatially varying configurations (Jakob et al. 2022) (additional results in Sec. F). The quantitative and qualitative results indicate that M³ashy consistently outperforms all baselines across metrics, producing diverse, high-quality, visually appealing, and perceptually realistic renderings. This demonstrates the effectiveness of M³ashy for neural material synthesis. Notably, materials synthesized by some baselines exhibit significant artifacts, likely due to the limitations of these simpler models in capturing the complex distribution of measured materials.

Metric		Training set	PCA-A	PCA-N	VAE-A	VAE-N	MERL100	M^{3}ashy (ours)
FID (↓)		0.187	10.9	23.8	26.1	10.0	7.56	**0.440**
MMD (↓)	BRDF-L1$\times 10^{-3}$	2.51	9.05	9.22	9.09	5.83	4.30	**4.02**
	RMSE$\times 10^{2}$	7.54	33.3	30.2	63.7	15.5	13.4	**9.34**
	NegPSNR	−28.7	−13.9	−14.8	−8.30	−20.9	−22.6	**−25.6**
	NegSSIM$\times 10$	−9.55	−6.74	−6.29	−2.68	−6.86	−8.27	**−9.40**
COV (%) (↑)	BRDF-L1	60.8	2.50	30	0.833	20.8	28.3	**50.8**
	RMSE	55.8	18.3	28.3	0.833	16.7	25.0	**50.0**
	NegPSNR	56.7	18.3	28.3	0.833	18.3	25.0	**50.0**
	NegSSIM	59.2	23.3	16.7	0.833	17.5	22.5	**51.7**
1-NNA (%) (↓)	BRDF-L1	58.8	100	95.4	100	96.7	92.5	**80.0**
	RMSE	55.4	96.3	93.4	100	93.3	84.6	**60.0**
	NegPSNR	55.0	94.2	90.0	100	93.3	84.6	**60.4**
	NegSSIM	57.5	96.3	96.7	100	93.8	86.3	**61.7**

Table 2: Qualitative evaluation of unconditional synthesis with metrics assessing generation fidelity and diversity. M^{3}ashy significantly outperforms all baseline models across these metrics, underscoring its effectiveness in neural material synthesis.

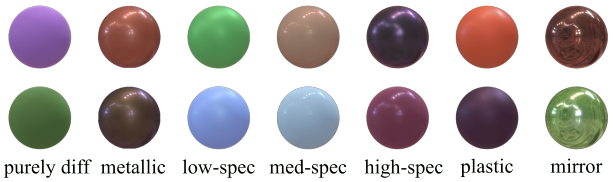


Figure 6: Synthesized materials of seven distinct categories using our novel constrained synthesis. Grounded in BRDF statistical analysis, this approach provides enhanced explainability and interpretability compared to standard conditional synthesis methods.

4.4 Multi-Modal Conditional Synthesis

To further evaluate the effectiveness of our pipeline, we perform multi-modal conditional synthesis by conditioning our model on various modalities of input: material type, text description, or material images.

For material type conditioning, we represent each of the 48 material types in the MERL dataset (Matusik et al. 2003) (*e.g.*, *acrylic*, *metallic*, *plastic*, *etc.*) using integers. The full list of material types is available in Sec. D.3 in the supplementary; For text conditioning, we use descriptions derived from the MERL dataset (Matusik et al. 2003). For additional materials in the AugMERL dataset, descriptions are assigned as follows: For RGB-permuted materials, we retain the original description but omit color-specific words (*e.g.*, *"red metallic paint"* becomes *"metallic paint"*); For PCA-interpolated materials, we generate descriptions in the format *"a mixture of* t_A *and* t_B*"*, where t_A and t_B are the descriptions of the interpolated materials A and B, respectively; For image-based conditioning, we use cropped single-view renderings of materials from AugMERL as input. We encode input texts and images using CLIP encoders (Radford et al. 2021).

Figures 7 to 9 present the results for type-, text-, and image-conditioned synthesis, respectively. Across all conditioning modes, the synthesized materials demonstrate realism, diversity, and a close alignment with the input conditions. Notably, in text- and image-conditioned synthesis, M^{3}ashy effectively generalizes to unseen texts (*e.g.*, *"green metal"*, *"red plastic"*, and *"highly specular material"*) and real-world images, producing materials that are perceptually consistent with these previously unseen inputs.

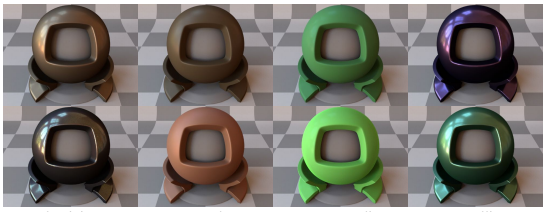


Figure 7: Type-conditioned synthesis. The synthesized materials are diverse and closely align with the input type.

4.5 Constrained Synthesis

We classify materials into seven categories based on their reflective properties: *diffuse*, *metallic*, *low-specular*, *medium-specular*, *high-specular*, *plastic*, and *mirror*. To enable the synthesis of materials within a specified category, we introduce a novel approach called *constrained synthesis*. This statistics-based method complements our conditional pipeline by enforcing constraints on unconditionally synthesized samples, allowing for targeted material generation according to desired reflective characteristics.

We derive the theoretical upper limit for a diffuse reflectance value, f_{diffuse}, for a constant (*i.e.*, purely diffuse) BRDF $f_r(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \equiv f_{\text{diffuse}}$ in each color channel. For a physically valid BRDF that adheres to energy passivity (Zhou et al. 2024), the reflected energy must not exceed



Figure 8: Text-conditioned synthesis. Synthesized materials align with the texts and generalize to unseen inputs: *"green metal"*, *"red plastic"*, and *"highly specular material"*.



Figure 9: Image-conditioned synthesis. Each of the eight pairs consists of the input (left) and the synthesized (right). M^{3}ashy effectively generates realistic materials that closely align with the conditioning images and generalizes to unseen, real-world images (last column).

the incident energy in each channel. Thus, we have:

$$1 \geq \int_{H^2} f_r(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o) \cos\theta_o \, \mathrm{d}\boldsymbol{\omega}_o \tag{7}$$

$$= f_{\text{diffuse}} \int_{H^2} \cos\theta_o \, \mathrm{d}\boldsymbol{\omega}_o = \pi f_{\text{diffuse}} \Rightarrow f_{\text{diffuse}} \leq \frac{1}{\pi}. \tag{8}$$

Building on this observation and a statistical analysis of the MERL's mean and maximum reflectance values across color channels and material types (Sec. G.1 in the supplementary), we propose a set of rules for categorizing materials into seven types. These rules enable the selection of synthesized materials based on desired material characteristics. Unlike many black-box machine learning approaches, this method is rooted in BRDF analysis, offering inherent explainability and interpretability. Below, we outline two of these rules, with the full set detailed in Sec. D.4.

- Purely diffuse materials: The reflectance values in all directions do not exceed the diffuse threshold f_{diffuse}, allowing for only $e := 8 \times 10^4$ exceptions:

$$|\{(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) \in \mathbb{R}^6 \mid \|f_r(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i)\|_\infty > f_{\text{diffuse}}\}| < e, \tag{9}$$

where $\|\cdot\|_\infty$ denotes the maximum reflectance value among the three color channels.

Metric	Sparse reconstruction		Compression	
	MERL	AugMERL	MERL	AugMERL
PSNR (↑)	32.2	**36.3**	45.2	**48.3**
Delta E (↓)	2.1	**1.8**	0.693	**0.623**
SSIM (↑)	0.972	**0.983**	**0.994**	**0.994**

Table 3: Quantitative comparison of training on MERL versus AugMERL in the sparse BRDF reconstruction and BRDF compression experiments. The results demonstrate that training on AugMERL consistently enhances performance across all metrics.

- Metallic materials: The reflectance values in all directions exceed the diffuse threshold f_{diffuse}:

$$\forall(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) \in \mathbb{R}^6, \|f_r(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i)\|_\infty > f_{\text{diffuse}} \tag{10}$$

Figure 6 shows materials generated through our constrained synthesis, incorporating the proposed filtering rules to ensure that the synthesized outputs match the specified material categories. The results demonstrate that the synthesized materials effectively exhibit the characteristics of the desired categories.

4.6 Ablation Study

To assess the impact of our augmented material dataset AugMERL, we further train our model on the original MERL dataset in the unconditional synthesis task. We report both quantitative and qualitative results for this model, presented in the "MERL100" columns of Tab. 2 and Fig. 5, respectively. The results indicate that the model trained on AugMERL exhibits higher quality and diversity compared to the one trained on MERL, demonstrating the effectiveness of our augmented dataset in enhancing the synthesis pipeline.

Additionally, we conduct sparse BRDF reconstruction and BRDF compression experiments following a previous method (Gokbudak et al. 2023). For sparse reconstruction, we set the sample size to $N = 4000$, while for compression, we use a latent dimension of 40. In both experiments, we train the model on either the original MERL dataset or the AugMERL dataset. The results, summarized in Tab. 3, demonstrate that training on AugMERL consistently enhances material quality across all evaluated metrics, further validating the effectiveness of our augmented dataset.

5 Conclusion and Future Work

We introduced M^{3}ashy, a **m**ulti-**m**odal **m**aterial **s**ynthesis approach with **h**yperdiffusion. Using neural fields as the core representation, we trained hyperdiffusion on their weights, demonstrating its ability to generate high-quality, diverse materials. Additionally, we contribute two material datasets and three BRDF metrics for future research. At this stage, our method does not account for physical correctness. Promising directions for future work include developing physically accurate neural representations of BRDFs and extending the approach to support more complex materials.

References

Abdi, H.; and Williams, L. J. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4): 433–459.

Bangay, S.; and Radloff, J. D. 2004. Kaleidoscope configurations for reflectance measurement. In *Proceedings of the 3rd International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa*, AFRIGRAPH '04, 161–170. New York, NY, USA: Association for Computing Machinery. ISBN 1581138636.

Bell, S.; Upchurch, P.; Snavely, N.; and Bala, K. 2015. Material recognition in the wild with the materials in context database. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3479–3487.

Betzalel, E.; Penso, C.; Navon, A.; and Fetaya, E. 2022. A Study on the Evaluation of Generative Models. arXiv:2206.10935.

Bousseau, A.; Paris, S.; and Durand, F. 2009. User-assisted intrinsic images. *ACM Transactions on Graphics (TOG)*, 28(5): 1–10.

Burley, B. 2012. Physically-Based Shading at Disney. *SIGGRAPH Comput. Graph.*

Chen, D. Z.; Siddiqui, Y.; Lee, H.-Y.; Tulyakov, S.; and Nießner, M. 2023a. Text2Tex: Text-driven Texture Synthesis via Diffusion Models. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 18512–18522.

Chen, R.; Chen, Y.; Jiao, N.; and Jia, K. 2023b. Fantasia3D: Disentangling Geometry and Appearance for High-quality Text-to-3D Content Creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 22246–22256.

Cook, R. L.; and Torrance, K. E. 1982. A Reflectance Model for Computer Graphics. *ACM Trans. Graph.*, 1(1): 7–24.

Csiszár, I. 1975. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, 146–158.

Dana, K. J.; van Ginneken, B.; Nayar, S. K.; and Koenderink, J. J. 1999. Reflectance and texture of real-world surfaces. *ACM Trans. Graph.*, 18(1): 1–34.

Devlin, J. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dupuy, J.; and Jakob, W. 2018a. An adaptive parameterization for efficient material acquisition and rendering. *ACM Transactions on graphics (TOG)*, 37(6): 1–14.

Dupuy, J.; and Jakob, W. 2018b. An Adaptive Parameterization for Efficient Material Acquisition and Rendering. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 37(6): 274:1–274:18.

Erkoç, Z.; Ma, F.; Shan, Q.; Nießner, M.; and Dai, A. 2023. HyperDiffusion: Generating Implicit Neural Fields with Weight-Space Diffusion. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 14254–14264.

Fan, J.; Wang, B.; Hašan, M.; Yang, J.; and Yan, L.-Q. 2022. Neural Layered BRDFs. In *Proceedings of SIGGRAPH 2022*.

Filip, J.; and Vávra, R. 2014. Template-based sampling of anisotropic BRDFs. *Computer Graphics Forum*, 33(7): 91–99.

Gatys, L.; Ecker, A. S.; and Bethge, M. 2015. Texture synthesis using convolutional neural networks. *Advances in neural information processing systems*, 28.

Gokbudak, F.; Sztrajman, A.; Zhou, C.; Zhong, F.; Mantiuk, R.; and Oztireli, C. 2023. Hypernetworks for Generalizable BRDF Estimation. *arXiv e-prints*, arXiv–2311.

Guarnera, D.; Guarnera, G.; Ghosh, A.; Denk, C.; and Glencross, M. 2016. BRDF Representation and Acquisition. *Computer Graphics Forum*, 35(2): 625–650.

Guo, J.; Li, Z.; He, X.; Wang, B.; Li, W.; Guo, Y.; and Yan, L.-Q. 2023. MetaLayer: A Meta-Learned BSDF Model for Layered Materials. *ACM Trans. Graph.*, 42(6).

Gupta, S.; Arbelaez, P.; and Malik, J. 2013. Perceptual organization and recognition of indoor scenes from RGB-D images. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 564–571.

Haindl, M.; and Filip, J. 2013. *Visual texture: Accurate material appearance measurement, representation and modeling*. Springer Science & Business Media.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Henzler, P.; Deschaintre, V.; Mitra, N. J.; and Ritschel, T. 2021. Generative modelling of BRDF textures from flash images. *ACM Trans. Graph.*, 40(6).

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. arXiv:2006.11239.

Ho, J.; and Salimans, T. 2022. Classifier-Free Diffusion Guidance. arXiv:2207.12598.

Hu, B.; Guo, J.; Chen, Y.; Li, M.; and Guo, Y. 2020. DeepBRDF: A Deep Representation for Manipulating Measured BRDF. *Computer Graphics Forum*, 39(2): 157–166.

Hu, Y.; Guerrero, P.; Hasan, M.; Rushmeier, H.; and Deschaintre, V. 2023. Generating Procedural Materials from Text or Image Prompts. In *ACM SIGGRAPH 2023 Conference Proceedings*, 1–11.

Jakob, W.; Speierer, S.; Roussel, N.; and Vicini, D. 2022. Dr.Jit: A Just-In-Time Compiler for Differentiable Rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)*, 41(4).

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4401–4410.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Lawrence, J.; Rusinkiewicz, S.; and Ramamoorthi, R. 2004. Efficient BRDF importance sampling using a factored representation. *ACM Transactions on Graphics (ToG)*, 23(3): 496–505.

Li, H.; Foo, S.-C.; Torrance, K. E.; and Westin, S. H. 2006. Automated three-axis gonioreflectometer for computer graphics applications. *Optical Engineering*, 45(4): 043605.

Liao, Q.; Leibo, J.; and Poggio, T. 2016. How important is weight symmetry in backpropagation? *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 1.

Marschner, S. R.; Westin, S. H.; Lafortune, E. P. F.; Torrance, K. E.; and Greenberg, D. P. 1999. Image-Based BRDF Measurement Including Human Skin. In Lischinski, D.; and Larson, G. W., eds., *Eurographics Workshop on Rendering*. The Eurographics Association. ISBN 3-211-83382-X.

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

Memery, S.; Cedron, O.; and Subr, K. 2023. Generating Parametric BRDFs from Natural Language Descriptions . *Computer Graphics Forum*.

Montes, R.; and Ureña, C. 2012. An overview of BRDF models. *University of Grenada, Technical Report LSI-2012-001*.

Nair, V.; and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814.

Ngan, A.; Durand, F.; and Matusik, W. 2005. Experimental Analysis of BRDF Models. *Rendering Techniques*, 2005(16th): 2.

Nicodemus, F. E.; Richmond, J. C.; Hsia, J. J.; Ginsberg, I. W.; Limperis, T.; Harman, S.; and Baruch, J. J. 1977. *Geometrical considerations and nomenclature for reflectance*. NATIONAL BUREAU OF STANDARDS, USA.

Nielsen, J. B.; Jensen, H. W.; and Ramamoorthi, R. 2015. On Optimal, Minimal BRDF Sampling for Reflectance Acquisition. *ACM Transactions on Graphics (TOG)*, 34(6): 186:1–186:11.

Peebles, W.; Radosavovic, I.; Brooks, T.; Efros, A. A.; and Malik, J. 2022. Learning to learn with generative models of neural network checkpoints. *arXiv preprint arXiv:2209.12892*.

Phong, B. T. 1975. Illumination for computer generated pictures. *Commun. ACM*, 18(6): 311–317.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Romeiro, F.; and Zickler, T. 2010. Blind reflectometry. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11*, 45–58. Springer.

Rusinkiewicz, S. 1998. A New Change of Variables for Efficient BRDF Representation. In *Eurographics Workshop on Rendering*. Vienna, Austria.

Serrano, A.; Gutierrez, D.; Myszkowski, K.; Seidel, H.-P.; and Masia, B. 2016. An Intuitive Control Space for Material Appearance. *SIGGRAPH ASIA*, 35(6): 186:1–186:12.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising diffusion implicit models. *ICLR 2021*.

Sun, T.; Jensen, H. W.; and Ramamoorthi, R. 2018. Connecting measured brdfs to analytic brdfs by data-driven diffuse-specular separation. *ACM Transactions on Graphics (TOG)*, 37(6): 1–15.

Sztrajman, A.; Křivánek, J.; Wilkie, A.; and Weyrich, T. 2017. Image-based Remapping of Material Appearance. In *Proc. 5th Workshop on Material Appearance Modeling*, 5–8.

Sztrajman, A.; Křivánek, J.; Wilkie, A.; and Weyrich, T. 2019. Image-based Remapping of Spatially-Varying Material Appearance. *Journal of Computer Graphics Techniques (JCGT)*, 8(4): 1–30.

Sztrajman, A.; Rainer, G.; Ritschel, T.; and Weyrich, T. 2021. Neural BRDF Representation and Importance Sampling. *Computer Graphics Forum*.

Tchapmi, L. P.; Ray, T.; Tchapmi, M.; Shen, B.; Martin-Martin, R.; and Savarese, S. 2022. Generating procedural 3D materials from images using neural networks. In *Proceedings of the 2022 4th International Conference on Image, Video and Signal Processing*, 32–40.

Theis, L.; van den Oord, A.; and Bethge, M. 2016. A note on the evaluation of generative models. arXiv:1511.01844.

Vaswani, A. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Walter, B.; Marschner, S. R.; Li, H.; and Torrance, K. E. 2007. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, 195–206. Goslar, DEU: Eurographics Association. ISBN 9783905673524.

Wang, Z.; Bovik, A.; Sheikh, H.; and Simoncelli, E. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612.

Weinmann, M.; Gall, J.; and Klein, R. 2014. Material Classification Based on Training Data Synthesized Using a BTF Database. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III*, 156–171. Springer International Publishing.

Weinmann, M.; and Klein, R. 2015a. Advances in geometry and reflectance acquisition (course notes). In *SIGGRAPH Asia 2015 Courses*, SA '15. New York, NY, USA: Association for Computing Machinery. ISBN 9781450339247.

Weinmann, M.; and Klein, R. 2015b. Material recognition for efficient acquisition of geometry and reflectance. In *Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part III 13*, 321–333. Springer.

Westin, S. H.; Li, H.; and Torrance, K. E. 2004. A comparison of four BRDF models. In *Eurographics Symposium on Rendering*, 1–10.

White, D. R.; Saunders, P.; Bonsey, S. J.; van de Ven, J.; and Edgar, H. 1998. Reflectometer for measuring the bidirectional reflectance of rough surfaces. *Appl. Opt.*, 37(16): 3450–3454.

Wu, J.; Zhang, C.; Xue, T.; Freeman, W. T.; and Tenenbaum, J. B. 2017. Marrnet: 3d shape reconstruction via 2.5 d sketches. *Advances in Neural Information Processing Systems*, 30.

Xu, X.; Lyu, Z.; Pan, X.; and Dai, B. 2023. MATLABER: Material-Aware Text-to-3D via LAtent BRDF auto-EncodeR. arXiv:2308.09278.

Yang, G.; Huang, X.; Hao, Z.; Liu, M.-Y.; Belongie, S.; and Hariharan, B. 2019. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4541–4550.

Zhang, S.; Peng, S.; Xu, T.; Yang, Y.; Chen, T.; Xue, N.; Shen, Y.; Bao, H.; Hu, R.; and Zhou, X. 2024. MaPa: Text-driven Photorealistic Material Painting for 3D Shapes. arXiv:2404.17569.

Zheng, C.; Zheng, R.; Wang, R.; Zhao, S.; and Bao, H. 2021. A Compact Representation of Measured BRDFs Using Neural Processes. *ACM Trans. Graph.*, 41(2).

Zhou, C.; Sztrajman, A.; Rainer, G.; Zhong, F.; Gokbudak, F.; Guo, Z.; Xia, W.; Mantiuk, R.; and Oztireli, C. 2024. Physically Based Neural Bidirectional Reflectance Distribution Function. arXiv:2411.02347.

Zhou, C.; Zhong, F.; and Oztireli, C. 2023. CLIP-PAE: Projection-Augmentation Embedding to Extract Relevant Features for a Disentangled, Interpretable and Controllable Text-Guided Face Manipulation. In *Proceedings of ACM SIGGRAPH 2023*, 1–9.

Zhou, Y.; Zhu, Z.; Bai, X.; Lischinski, D.; Cohen-Or, D.; and Huang, H. 2018. Non-stationary texture synthesis by adversarial expansion. *arXiv preprint arXiv:1805.04487*.

A Related Work on Material Acquisition and Databases

Capturing real-world material appearance requires lengthy acquisition times and substantial storage. Traditional capture uses four-axis gonioreflectometers (Li et al. 2006; White et al. 1998) with later advancements enhancing angular coverage, wavelength resolution, efficiency, and accuracy (White et al. 1998). Systems range from basic sensor setups, like camera arrays (Weinmann and Klein 2015a), varied illumination sources (Haindl and Filip 2013), and specific geometries (Marschner et al. 1999; Bangay and Radloff 2004; Weinmann and Klein 2015b), to advanced configurations for capturing full material patches (Dana et al. 1999; Weinmann, Gall, and Klein 2014). Filip and Vávra contributed a database of 150 materials, many exhibiting anisotropic properties (Filip and Vávra 2014). These material databases are continuously being augmented by later methods (Ngan, Durand, and Matusik 2005; Dupuy and Jakob 2018a). These improvements have enabled the creation of extensive BRDF material databases (Matusik et al. 2003; Dupuy and Jakob 2018b; Nielsen, Jensen, and Ramamoorthi 2015), which underpin the datasets in our work.

B Additional Background

B.1 Principal Component Analysis

Principal component analysis (PCA) (Abdi and Williams 2010), also known as Karhunen-Loève transform or Hotelling transform, is a linear dimensionality reduction technique. We utilize PCA for data augmentation in Sec. B.1 and baseline models in Sec. 4.3.

Given a dataset of n samples in a high-dimensional space $\mathcal{D} \in \mathbb{R}^{n \times d}$, PCA seeks to linearly transform the data onto a lower-dimensional space $\mathbb{R}^k$ spanned by k *principal components* capturing the primary variance of the data. In this reduced space, synthetic data points are generated by sampling from a Gaussian distribution with the same mean and variance as the original data in each principal component direction. These newly sampled points are then mapped back to the original feature space using the inverse PCA transformation. This method allows the creation of new data samples that maintain the underlying structure and variance characteristics of the original dataset, which can be useful for data augmentation and improving model robustness.

B.2 Diffusion Model

The forward and backward processes in hyperdiffusion are modeled as Markov chains with a total timestep T and learnable parameter $\boldsymbol{\eta}$:

$$q(\mathbf{x}_1, \ldots, \mathbf{x}_T|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \tag{11}$$

$$p_{\boldsymbol{\eta}}(\mathbf{x}_0, \ldots, \mathbf{x}_T) = p_{\boldsymbol{\eta}}(\mathbf{x}_t) \prod_{t=1}^{T} p_{\boldsymbol{\eta}}(\mathbf{x}_T|\mathbf{x}_{t-1}). \tag{12}$$

In the forward process, starting with the original data $\mathbf{x} = \mathbf{x}_0$, we iteratively add Gaussian noise at each step:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t I), \tag{13}$$

where $\{\beta_t\}_{t=1}^T$ defines the variance schedule. Each noisy vector, paired with the sinusoidal embedding of the timestep, is passed through a linear projection layer. The output projections are then combined with a learnable positional encoding vector. As the forward process progresses, $p(\mathbf{x}_T)$ converges towards a standard Gaussian distribution, $\mathcal{N}(0, I)$.

In the backward process, the transformer takes these inputs and produces denoised tokens, which are passed through a final output projection layer to generate the predicted noise. To train a learnable model $\boldsymbol{\epsilon}_{\boldsymbol{\eta}}(\mathbf{x}_t, t)$ parameterized by $\boldsymbol{\eta}$, we minimize the score matching objective:

$$\mathcal{L}_{\text{HD}}(\boldsymbol{\eta}) := \mathbb{E}_{\mathbf{x}_0, t\sim\mathcal{U}(1,T), \boldsymbol{\epsilon}\sim\mathcal{N}(0,I)} \left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\eta}}(\mathbf{x}_t, t)\|_2^2\right], \tag{14}$$

where $\mathcal{U}(1, T)$ represents the uniform distribution over $\{1, 2, \ldots, T\}$. This objective encourages the model to accurately predict the noise $\boldsymbol{\epsilon}$, effectively guiding the denoising process.

During inference, the network enables sampling via an iterative process (Song, Meng, and Ermon 2021), leveraging the factorization of the learned distribution as

$$p_{\boldsymbol{\eta}}(\mathbf{x}) = p(\mathbf{x}_T) p_{\boldsymbol{\eta}}(\mathbf{x}_0|\mathbf{x}_T) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\eta}}(\mathbf{x}_{t-1}|\mathbf{x}_t) \tag{15}$$

for $p(\mathbf{x}_T) := \mathcal{N}(0, I)$. For conditional sampling, we employ classifier-free guidance (CFG) (Ho and Salimans 2022) (please refer to Sec. D.2 for the algorithm).

B.3 Attention Mechanism

The attention mechanism (Vaswani 2017) allows models to focus on specific parts of input data, dynamically assigning different levels of "attention" or importance to different elements. The attention mechanism enables models to learn which parts of the input sequence are most relevant to predicting each output token. This process improves performance by allowing models to prioritize relevant information and ignore irrelevant details, especially in long sequences.

In modern applications, attention modules are widely used in natural language processing (NLP), computer vision, and beyond. The transformer architecture (Vaswani 2017), which relies heavily on the self-attention mechanism, has become foundational in NLP models, including BERT (Devlin 2018) and GPT (Mann et al. 2020). Attention allows these models to capture complex dependencies between words in a sentence, regardless of their distance from each other, leading to significant advancements in tasks like language translation, sentiment analysis, and image processing.

The attention mechanism computes a weighted combination of values based on the relevance of each value to a given query. In the context of self-attention (or scaled dot-product attention) in transformer models (Vaswani 2017), this process involves three main components: queries ($\mathbf{q}$), keys ($\mathbf{k}$), and values ($\mathbf{v}$). In summary, the attention mechanism dynamically focuses on relevant parts of the input by computing similarity scores between queries and keys, normalizing these scores, and using them to weight the values. This

allows models to capture dependencies across elements in a sequence, making attention a powerful tool for handling long-range dependencies in data.

Given an input sequence of embeddings $\mathbf{X}$ (*e.g.*, word embeddings in NLP or patch embeddings in vision), we first transform it into three different linear projections:

$$\mathbf{q} = \mathbf{X}W_{\mathbf{q}}; \tag{16}$$
$$\mathbf{k} = \mathbf{X}W_{\mathbf{k}}; \tag{17}$$
$$\mathbf{v} = \mathbf{X}W_{\mathbf{v}}, \tag{18}$$

where $W_{\mathbf{q}}$, $W_{\mathbf{k}}$, and $W_{\mathbf{v}}$ are learnable weight matrices. These projections represent the queries, keys, and values, respectively. The attention score between each query and key is then computed as the dot product $\mathbf{q} \cdot \mathbf{t}$. This results in a matrix of scores that represents the similarity between each element in the sequence. To stabilize gradients and prevent large values in the dot-product computation, the scores are scaled by the square root of the dimensionality of the queries/keys $\sqrt{d_{\mathbf{k}}}$. The scaled scores are $\frac{\mathbf{q}\cdot\mathbf{t}}{\sqrt{d_k}}$.

The scaled scores are passed through a softmax function, producing attention weights that sum to 1. This step converts the scores into probabilities, indicating the relevance of each value with respect to each query. These attention weights are used to compute a weighted sum of the values. Specifically, the output of the attention mechanism is

$$f_{\text{attention}}(\mathbf{q}, \mathbf{k}, \mathbf{v}) = f_{\text{softmax}}\left(\frac{\mathbf{q} \cdot \mathbf{k}}{\sqrt{d_k}}\right)\mathbf{v}. \tag{19}$$

This produces a context vector for each query that incorporates information from all other elements in the sequence, weighted by their relevance.

In practice, multiple attention heads are used in parallel. Each head learns different aspects of the input by using different projections $W_{\mathbf{q}}, W_{\mathbf{k}}$, and $W_{\mathbf{v}}$. The outputs from each head are then concatenated and linearly transformed to produce the final output.

B.4 Variational Autoencoder

We adopt variational autoencoders (VAEs) (Kingma and Welling 2013) as one of the baseline models in Sec. 4.3. VAEs are probabilistic generative models designed to capture the underlying probability distribution of a given dataset $\mathcal{D}$.

A VAE consists of a parameterized encoder $q_{\boldsymbol{\psi}}(\mathbf{z}|\mathbf{x})$ and decoder $p_{\boldsymbol{\zeta}}(\mathbf{x}|\mathbf{z})$, defined by parameters $\boldsymbol{\psi}$ and $\boldsymbol{\zeta}$, respectively. Assuming that the latent variables $\mathbf{z} \in \mathbb{R}^{D_{\mathbf{z}}}$ follow a prior distribution $p(\mathbf{z})$, both the encoder and decoder are jointly optimized to maximize the *evidence lower bound (ELBO)* on the likelihood of the data:

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}(\boldsymbol{\psi}, \boldsymbol{\zeta}; \mathbf{x}) := &\mathbb{E}_{q_{\boldsymbol{\psi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\zeta}}(\mathbf{x}|\mathbf{z})\right] \\ &- \mathcal{D}_{\text{KL}}\left(q_{\boldsymbol{\psi}}(\mathbf{z}|\mathbf{x}), p(\mathbf{z})\right),\end{aligned} \tag{20}$$

where $\mathcal{D}_{\text{KL}}$ represents the Kullback-Leibler divergence between the approximate posterior and the prior distribution (Csiszár 1975). This probabilistic framework enables VAEs to effectively approximate the true data distribution, facilitating robust generative modeling of complex data.

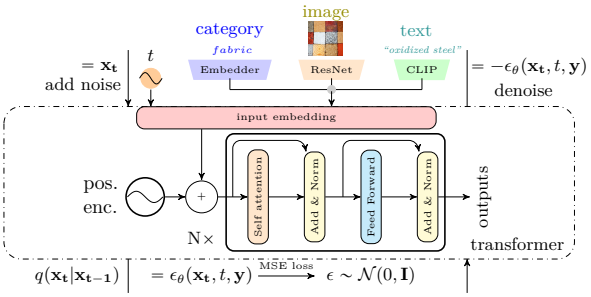


Figure 10: Transformer as hyperdiffusion backbone.

B.5 K-Means Clustering

We adopt K-means clustering in constrained synthesis (Sec. 4.5), where unconditional materials are filtered out by statistical information rather than a neural network. K-means clustering is a commonly used unsupervised method to partition n data samples $\mathbf{X} \in \mathbb{R}^{n\times p}$ into k clusters $\mathcal{S} = \{S_1, ..., S_k\}$, minimizing the distance between each sample and its center

$$\mu_i = \frac{1}{|S_i|}\sum_{\mathbf{x_i}\in S_i}\mathbf{x_i}. \tag{21}$$

The optimal partition can be computed via the following objective

$$\mathcal{S}^* = \arg\min_{\mathcal{S}}\sum_{i=0}^{k-1}\sum_{\mathbf{x_i}\in S_i}||\mathbf{x_i} - \mu_i||^2. \tag{22}$$

Given the assigned clusters, we can obtain the classification decision boundary directly.

C Model Implementation Details

C.1 Neural Field

The dimensionality of the flattened weights for our neural field $f_r^{\boldsymbol{\xi}}$ is $D_{\text{NF}} = 675$.

C.2 Transformer Backbone in Hyperdiffusion

The input and output tokens are mapped to vectors with learnable embedders. Sinusoidal positional encoding is also used. The feed-forward networks are single layer MLPs with ReLU activation functions.

The encoder network contains multiple identical layers, each with a feed-forward sublayer after multi-head attention. A residual connection is employed for each sublayer, followed by the layer normalization. The decoder is similar, but with an extra per-layer multi-head attention at the end, receiving the the encoder output.

Fig. 10 illustrates the transformer architecture.

C.3 PCA-Based Baselines

For PCA-based baselines introduced in the unconditional synthesis in Sec. 4.3, the dimensionalities of the reduced spaces for PCA-A and PCA-N are 300 and 100, respectively.

C.4 VAE-Based Baselines

In this section, we detail the model hyperparameters for VAE-based baselines introduced in the unconditional synthesis in Sec. 4.3.

For VAE-A, to reduce the input complexity, we downsample it from $90 \times 90 \times 180$ to $45 \times 45 \times 90$ and upsample the synthetic results using nearest neighbor algorithm. The hidden dimension is 256 and the latent space dimension is 300.

For VAE-N, the VAE architecture contains MLP-based encoder and decoder, each with 4 layers. The input dimension is D_{NF}, and the hidden and the latent space dimensions are both 300. The likelihood of the reconstructed data is measured by the mean squared error (MSE).

D Experiment Details

D.1 Training Details

In the neural field fitting, we set the batch size to 512, the number of epochs to 100, and the learning rate to 5×10^{-3}.

When training the hyperdiffusion model, we set the total timestep T to 100, the batch size to 512, the number of epochs to 700 for unconditional synthesis and an additional 200 for conditional synthesis, and the learning rate to adaptive from 5×10^{-4} to 5×10^{-6}. The experiments are run on an NVIDIA GeForce RTX 4090 GPU.

D.2 Conditional Sampling with Classifier-Free Guidance

We employ classifier-free guidance (CFG) (Ho and Salimans 2022) for conditional sampling on the hyperdiffusion. We present the algorithm in Alg. 1. Notice that we require $\omega > -1$ for conditional synthesis. Otherwise, the model downgrades to unconditional synthesis when $\omega = -1$.

Algorithm 1: Conditional sampling with classifier-free guidance (CFG)

Require: total timestep T
Require: guidance scale $\omega \geq -1$
Require: conditional context $\mathbf{y}$
Require: variance schedule $\{\beta_t\}_{t=1}^{T}$
  $\mathbf{x}_T \sim \mathcal{N}(0, I_{D_{\text{NF}}})$ ▷ sample $\mathbf{x}_T$ from prior
  for $t = T$ to 1 **do**
    $\boldsymbol{\epsilon}_{\boldsymbol{\eta}}^{\text{CFG}}(\mathbf{x}_t, \mathbf{y}, t) = (1+\omega)\boldsymbol{\epsilon}_{\boldsymbol{\eta}}(\mathbf{x}_t, \mathbf{y}, t) - \omega\boldsymbol{\epsilon}_{\boldsymbol{\eta}}(\mathbf{x}_t, \emptyset, t)$
    $\alpha_t = \prod_{i=1}^{t} \sqrt{1-\beta_i}$
    $\gamma_t = \prod_{t=1}^{t} \alpha_t$
    $\mathbf{x}_t^{\text{CFG}} = \frac{1}{\sqrt{\gamma_t}}(\mathbf{x}_t - \sqrt{1-\gamma_t}\boldsymbol{\epsilon}_{\boldsymbol{\eta}}^{\text{CFG}}(\mathbf{x}_t, \mathbf{y}, t))$
    $\mathbf{x}_{t-1} = \sqrt{\gamma_{t-1}}\mathbf{x}_t^{\text{CFG}} + \sqrt{1-\gamma_{t-1}}\boldsymbol{\epsilon}_{\boldsymbol{\eta}}^{\text{CFG}}(\mathbf{x}_t, \mathbf{y}, t)$
  end for
  return $\mathbf{x}_0$

D.3 Full Material Type List

We include the full list of 48 material types used in the type-conditional synthesis in Sec. 4.4: *acrylic*, *alum-bronze*, *alumina-oxide*, *aluminium*, *aventurnine*, *brass*, *wood*, *chrome-steel*, *chrome*, *colonial-maple*, *color-changing-paint*, *delrin*, *diffuse-ball*, *fabric*, *felt*, *fruitwood*, *grease-covered-steel*, *hematite*, *ipswich-pine*, *jasper*, *latex*, *marble*, *metallic-paint*, *natural*, *neoprene-rubber*, *nickel*, *nylon*, *obsidian*, *oxidized-steel*, *paint*, *phenolic*, *pickled-oak*, *plastic*, *polyethylene*, *polyurethane-foam*, *pvc*, *rubber*, *silicon-nitrade*, *soft-plastic*, *special-walnut*, *specular-fabric*, *specular-phenolic*, *specular-plastic*, *stainless-steel*, *steel*, *teflon*, *tungsten-carbide*, *two-layer*.

D.4 Full Set of Constrained Synthesis Rules

We include the full set of seven rules used in constrained synthesis:

- Purely diffuse materials: the reflectance values from all directions do not exceed the diffuse threshold f_{diffuse}, with only $e := 8 \times 10^4$ exceptions:

$$|\{(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) \in \mathbb{R}^6 \mid \|f_r(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i)\|_\infty > f_{\text{diffuse}}\}| < e, \tag{23}$$

  where $\|\cdot\|$ is the ℓ^∞-norm that selects the maximum component from the three color channels of reflectance values.
- Metallic materials: the reflectance values from all directions exceeds the diffuse threshold f_{diffuse}:

$$\forall(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i) \in \mathbb{R}^6, \|f_r(\boldsymbol{\omega}_o, \boldsymbol{\omega}_i)\|_\infty > f_{\text{diffuse}} \tag{24}$$

- Specular materials: Through our statistical analysis, we identify two specular thresholds $f_{\text{specular}}^{(1)} := 100$ and $f_{\text{specular}}^{(2)} := 600$. The materials can be classified as low-, mid-, or high-specular if the maximum reflectance value fall in the range $[f_{\text{diffuse}}, f_{\text{specular}}^{(1)})$, $[f_{\text{specular}}^{(1)}, f_{\text{specular}}^{(2)})$, and $[f_{\text{specular}}^{(2)}, \infty)$, respectively.
- Plastic materials: materials whose specular part is white. In other words, if the reflectance value exceeds f_{diffuse} for some direction, the difference between any two color channels should be smaller than a tolerance δ_{plastic}, which we set to be 5% of the maximum reflectance value.
- Mirror-like materials: the specular lobe in the polar diagram is narrow. In order to quantify this, we observe that under the Rusinkiewicz reparametrization (Sec. 3.2), the evaluation at $f_r(\theta_{\mathbf{h}} = 0, \theta_{\mathbf{d}} = 0, \varphi_{\mathbf{d}} = 0)$ is the peak of the lobe. Incrementing $\theta_{\mathbf{h}}$ decreases the BRDF. The width w of the lobe can then be defined as the value of $\theta_{\mathbf{h}}$ for which the reflectance value drops to half of the the peak $\frac{f_r(0,0,0)}{2}$. Empirically, we found that if the width $w < w_{\text{mirror}} := 0.349$, the material exhibits mirror-like behavior.

E Metric Details

E.1 Rendering-Based Metrics

Given two rendered images $I_1, I_2 \colon \mathbb{R}^w \times \mathbb{R}^h \to [0,1]^c$, where w, h are the width and height of the images, respectively, and C is the number of channels, we propose the following rendering-based metrics assessing the similarity and reconstruction quality between the two images.

Root mean squared error (RMSE) RMSE checks if pixel values at the same coordinates match.

$$\mathcal{L}_{\text{RMSE}}(I_1, I_2) := \sqrt{\frac{1}{wh}\sum_{i=1}^{w}\sum_{j=1}^{h}\left(I_1(i,j) - I_2(i,j)\right)^2}.$$

Note that RMSE depends strongly on the image intensity scaling. RMSE aims for lower value for better performance.

Peak signal-to-noise ratio (PSNR) is the scaled mean squared error (MSE). Given the maximum pixel value p, *i.e.*, peak signal, PSNR is defined as

$$\mathcal{L}_{\text{PSNR}}(I_1, I_2) = 10\log_{10}\frac{p^2}{\mathcal{L}^2_{\text{RMSE}}(I_1, I_2)}.$$

PSNR measures the image reconstruction quality and aims for higher values.

Structural similarity index measure (SSIM) SSIM (Wang et al. 2004) is a perception-based metric that measures the perceptual similarity of the two images. The computation of SSIM is based on three comparison measurements between the two images: luminance (l), contrast (c), and structure (s) defined as

$$l(I_1, I_2) := \frac{2\mu_1\mu_2 + c_1}{\mu_1^2 + \mu_2^2 + c_1}; \tag{25}$$

$$c(I_1, I_2) := \frac{2\sigma_{I_1}\sigma_{I_2} + c_2}{\sigma_{I_1}^2 + \sigma_{I_2}^2 + c_2}; \tag{26}$$

$$s(I_1, I_2) := \frac{\sigma_{I_1 I_2} + c_3}{\sigma_{I_1}\sigma_{I_2} + c_3}, \tag{27}$$

respectively, where μ_I, σ_I, and $\sigma_{I_1 I_2}$ are the mean, standard deviation, and variance of the images:

$$\mu_I := \frac{1}{w \cdot h}\sum_{i=1}^{w}\sum_{j=1}^{h} I(i,j); \tag{28}$$

$$\sigma_I := \sqrt{\frac{1}{w \cdot h - 1}\sum_{i=1}^{w}\sum_{j=1}^{h}\left(I(i,j) - \mu_I\right)^2}; \tag{29}$$

$$\sigma_{I_1 I_2} := \frac{1}{w \cdot h - 1}\sum_{i=1}^{w}\sum_{j=1}^{h}\left(I_1(i,j) - \mu_{I_1}\right)\left(I_2(i,j) - \mu_{I_2}\right) \tag{30}$$

and

$$c_1 := (k_1 L)^2; \tag{31}$$

$$c_2 := (k_2 L)^2; \tag{32}$$

$$c_3 := \frac{c_2}{2} \tag{33}$$

are the variables to stabilize the division with weak denominator where k_1 and k_2 are coefficients defaulted to 0.01 and 0.03, respectively, and L is the dynamic range of the pixel-values typically chosen to be $2^l - 1$ where l is the number of bits per pixel.

SSIM is then defined as a weighted combination of the above comparative measures with exponential weights $a, b, c > 0$:

$$\mathcal{L}_{\text{SSIM}}(I_1, I_2) := l^a(I_1, I_2)c^b(I_1, I_2)s^c(I_1, I_2). \tag{34}$$

SSIM aims for higher values for better performance.

E.2 Distributional Metrics Validation

In this section, we explore the effectiveness of the proposed novel material distributional metrics (Sec. 4.2) by examining the nearest neighbor materials under certain distances.

In Fig. 11, we demonstrate the nearest reference BRDF to each synthetic sample under the distance

$$d_{\text{BRDF-L1-Log}} := \mathop{\mathbb{E}}_{\theta_{\mathbf{h}},\theta_{\mathbf{d}},\varphi_{\mathbf{d}}}\Big[\,\big|\log\left(1 + f_r\right) - \log\left(1 + f_r'\right)\big|\,\Big], \tag{35}$$

which is used to compute the distributional metrics MMD, COV, and 1-NNA. While there exist mismatches due to the limited size of the reference set, we see that proposed BRDF distance is capable of matching most BRDFs in terms of reflective behaviors, leading to plausible BRDF distributional metrics.

In Fig. 12, we further visualize the nearest neighbor information by plotting the heatmap of pairwise mean squared logarithmic distance defined as

$$d_{\text{MSL}} := \mathop{\mathbb{E}}_{\theta_{\mathbf{h}},\theta_{\mathbf{d}},\varphi_{\mathbf{d}}}\Big[\left(\log\left(1 + f_r\cos\theta_i\right) - \log\left(1 + f_r'\cos\theta_i\right)\right)^2\Big]. \tag{36}$$

From the Figure we can see that the nearest neighbor is captured effectively with the underlying distance, validating the design choice of our BRDF distributional metrics.

F Further Synthesis Results

For the unconditional synthesis experiment in Sec. 4.3, further results are presented in Figs. 13 to 17 for M^{3}ashy (our method), PCA-A, PCA-N, VAE-A, and VAE-N baselines, respectively. In addition, Fig. 18 presents the complex geometries rendered with our synthesized materials. To support richer visual effects, our synthesized materials can also be rendered with normal maps (Fig. 19), bump maps (Fig. 20) and spatially varying configurations (Jakob et al. 2022) (Figs. 21 and 22).

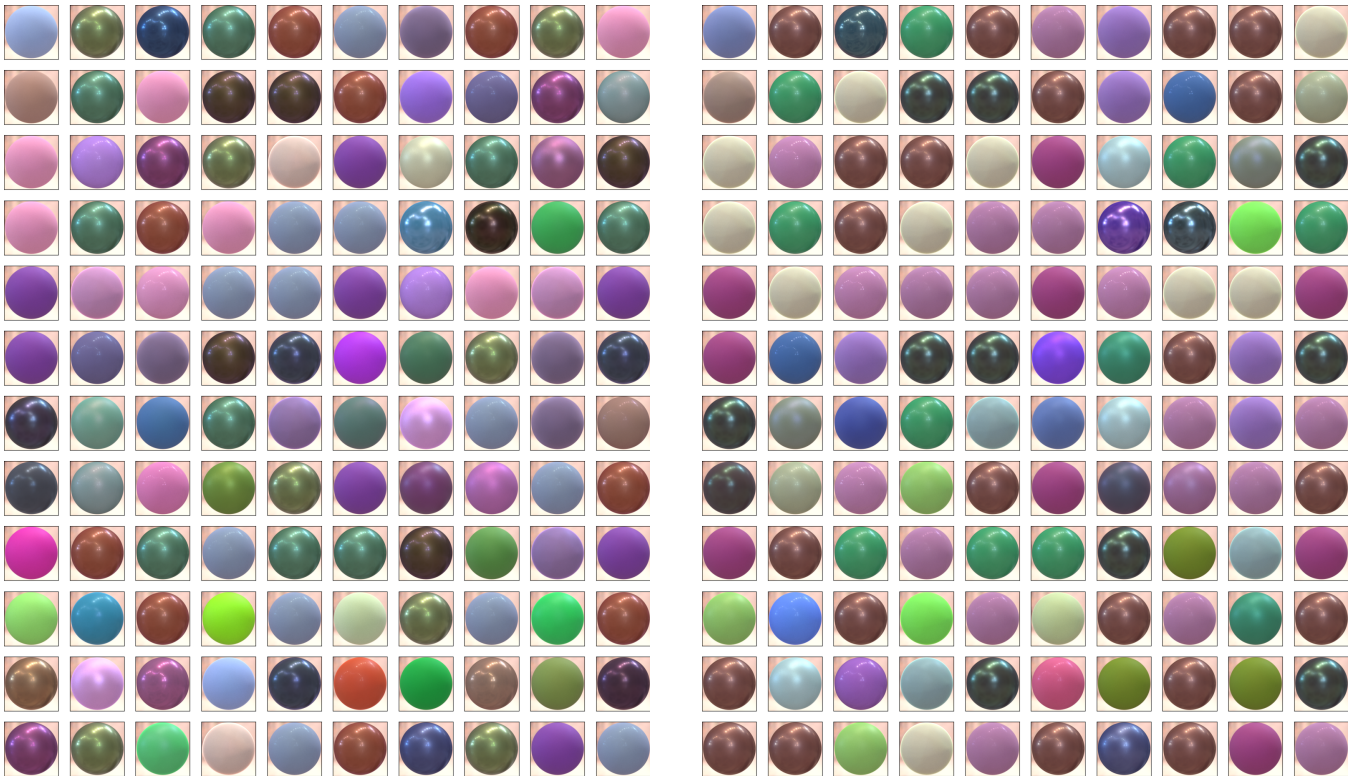
G Additional Experiments

G.1 Statistical Analysis on MERL Dataset

We perform the statistical analysis on the MERL dataset (Matusik et al. 2003). We collect mean and maximum reflectance values in different color channels across all materials. Figure 23 demonstrates the statistical summary averaged across each material type while Fig. 24 details the statistics in each individual material grouped by type. In addition, to identify the classification boundaries between different material types, in Fig. 25, we apply K-means clustering (for background see Sec. B.5) on the maximum reflectance of red channel.

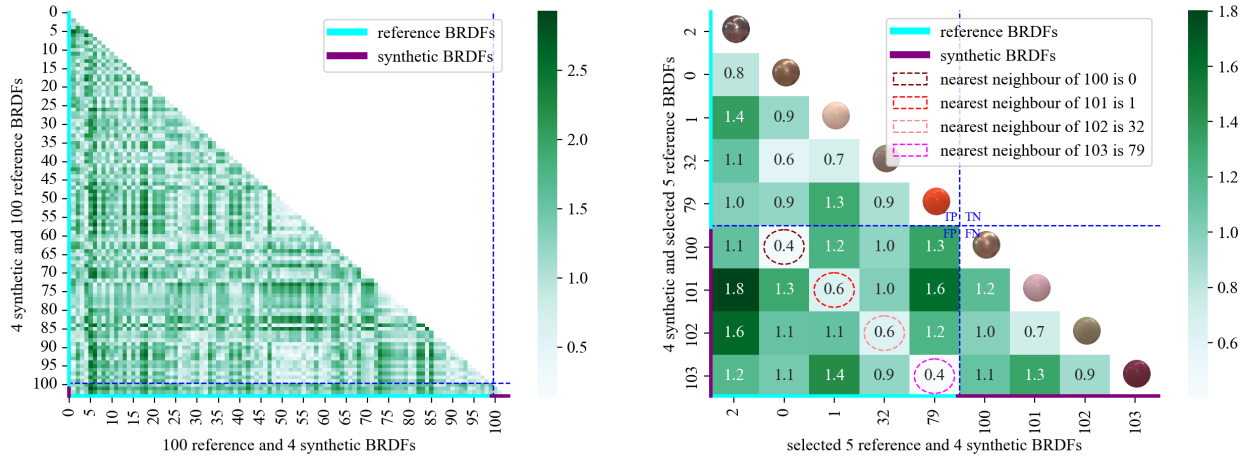
G.2 Superresolution from Low-Density BRDF

We present the experiment on the BRDF reconstruction for low-density input using NBRDF (Sztrajman et al. 2021), showing its capability of superresolution. In the experiment, when fitting the NBRDF to a MERL material (Matusik et al. 2003), we downsample the input $\theta_h, \theta_d, \phi_d$ by a factor of



(a) Synthetic set (b) Nearest neighbor reference

Figure 11: Nearest reference under $d_{\text{BRDF-L1-Log}}$ (Eq. (35)): for each synthetic material in (a), we identify its nearest neighbor in the reference set under $d_{\text{BRDF-L1-Log}}$ presented at the same grid position in (b). The nearest neighbor metric is capable of matching most BRDFs in terms of reflective behaviors.



(a) Pairwise distance of synthetic and all reference BRDFs (b) Pairwise distance of synthetic and selected reference BRDFs

Figure 12: Nearest neighbor information illustrated by pairwise mean squared logarithmic distance. We can see that the nearest neighbor is captured effectively with the underlying distance.

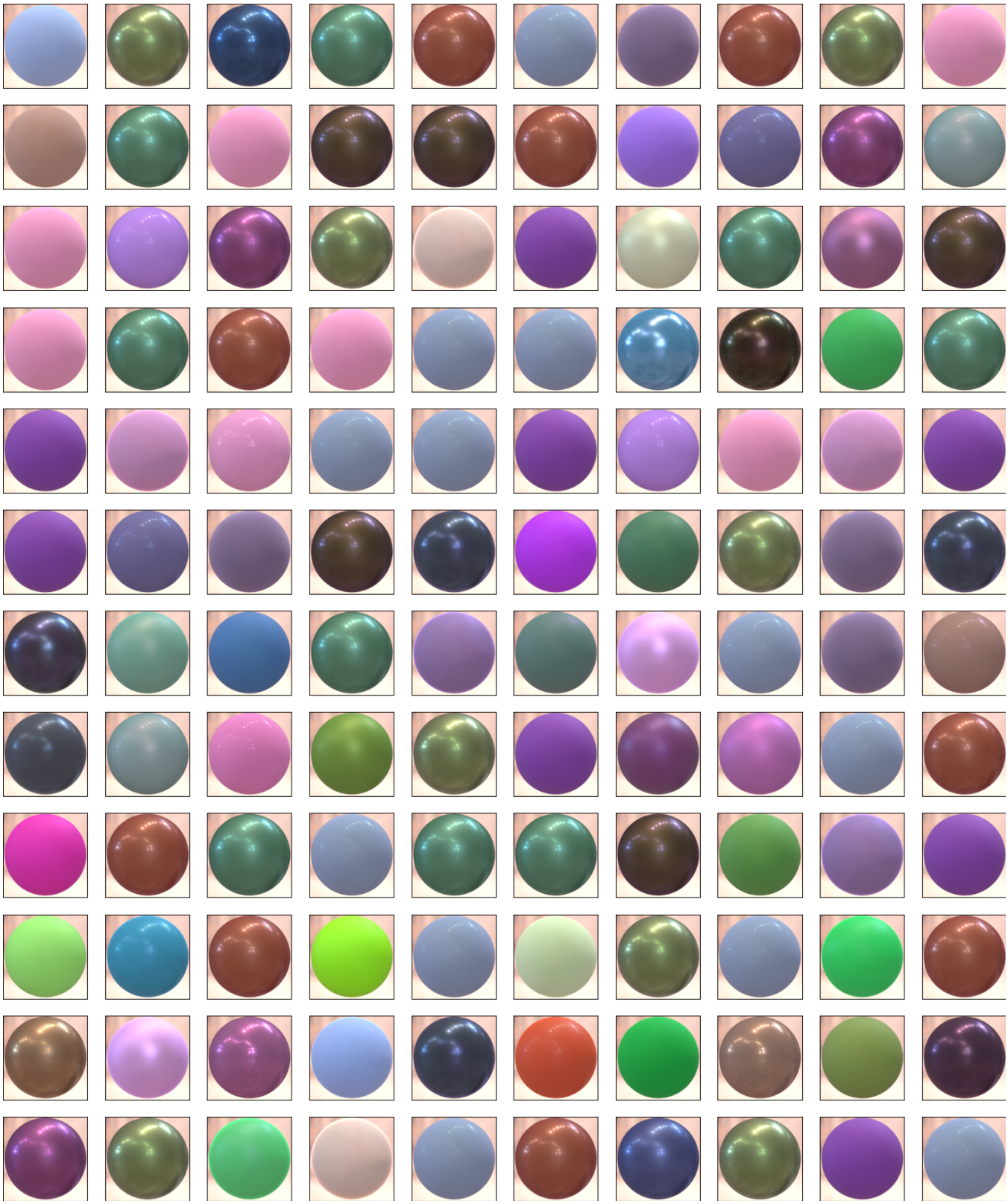


Figure 13: Synthesized materials by M^{3}ashy (our method).

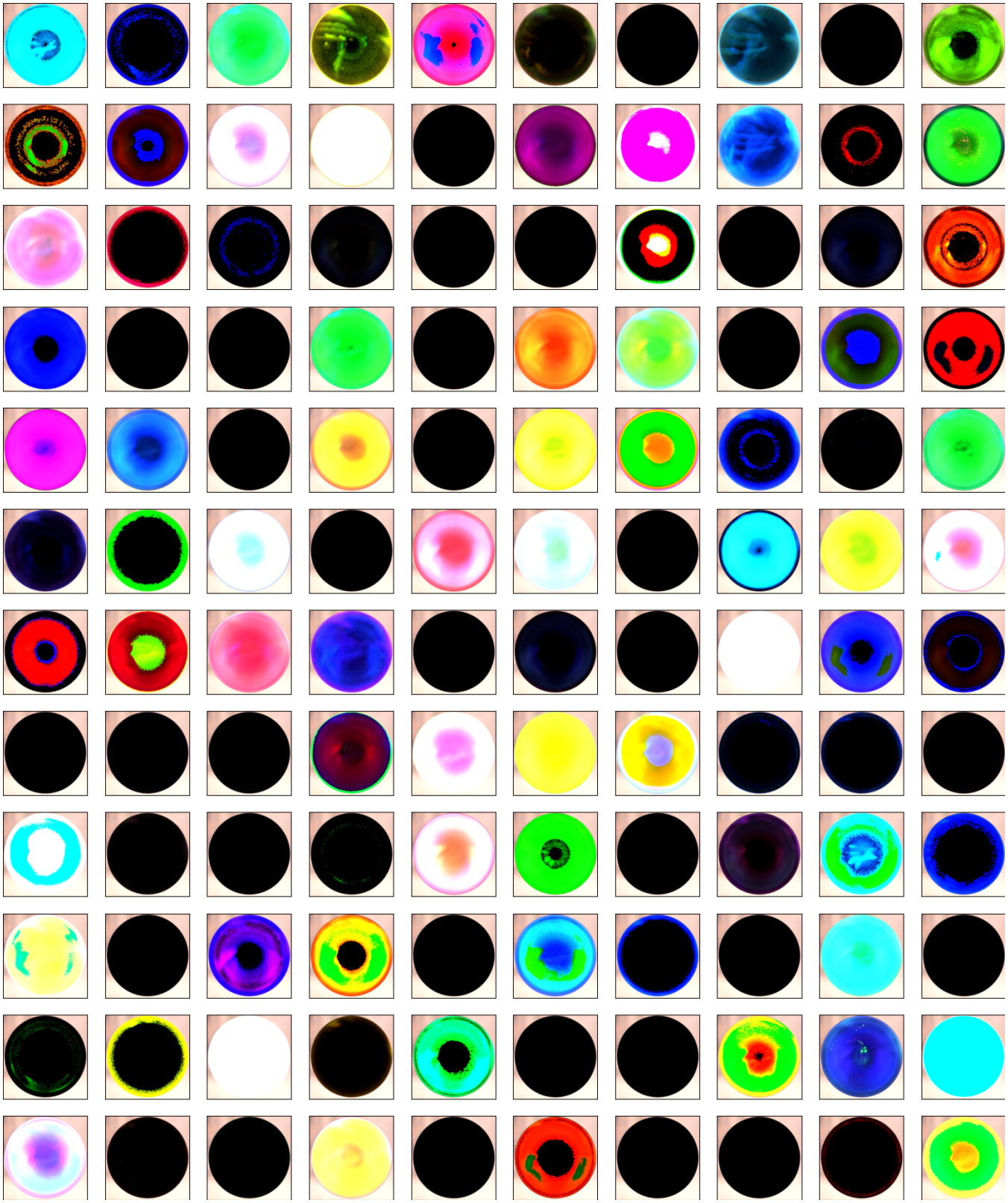


Figure 14: Synthesized materials by PCA-A.

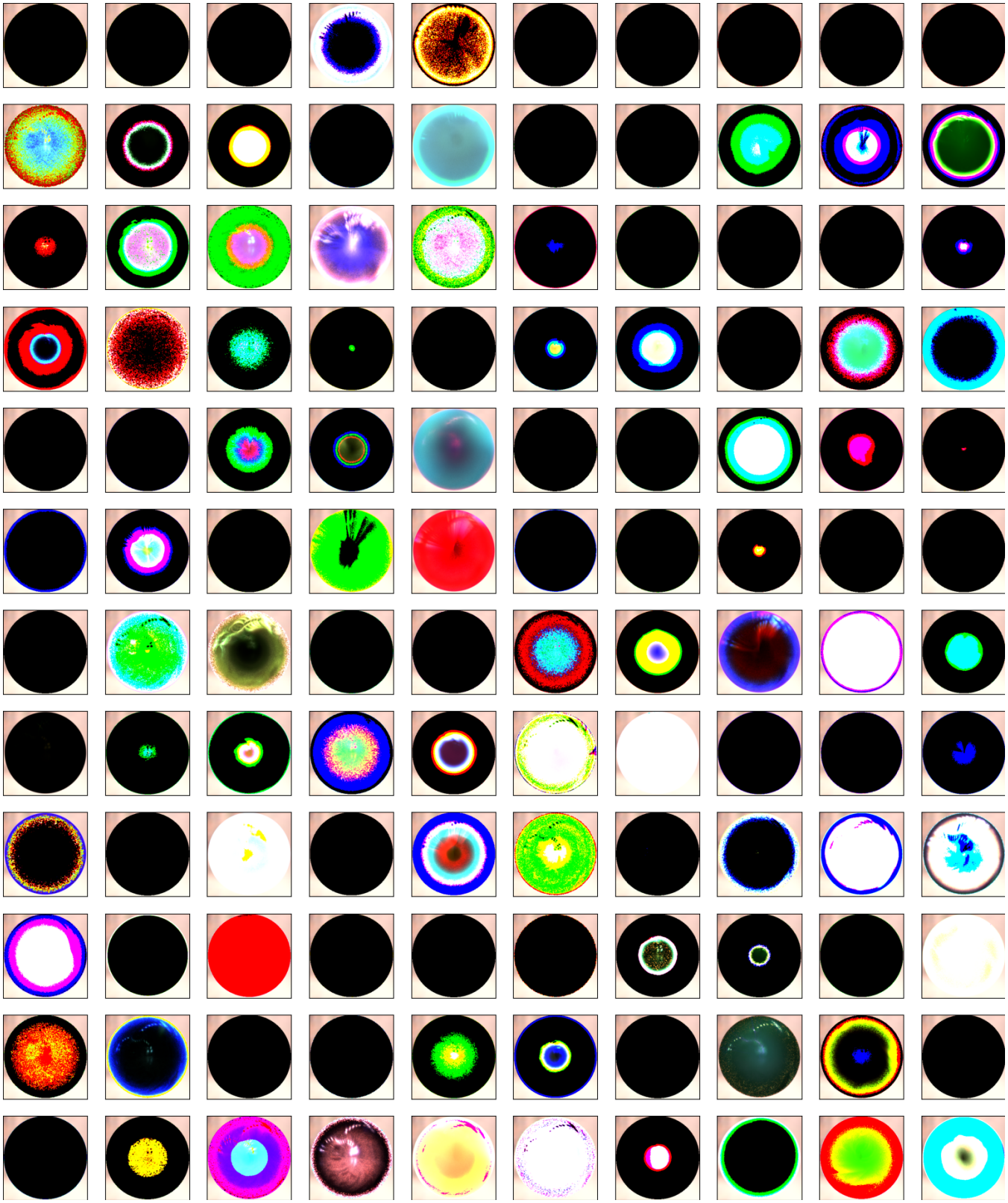


Figure 15: Synthesized materials by PCA-N.

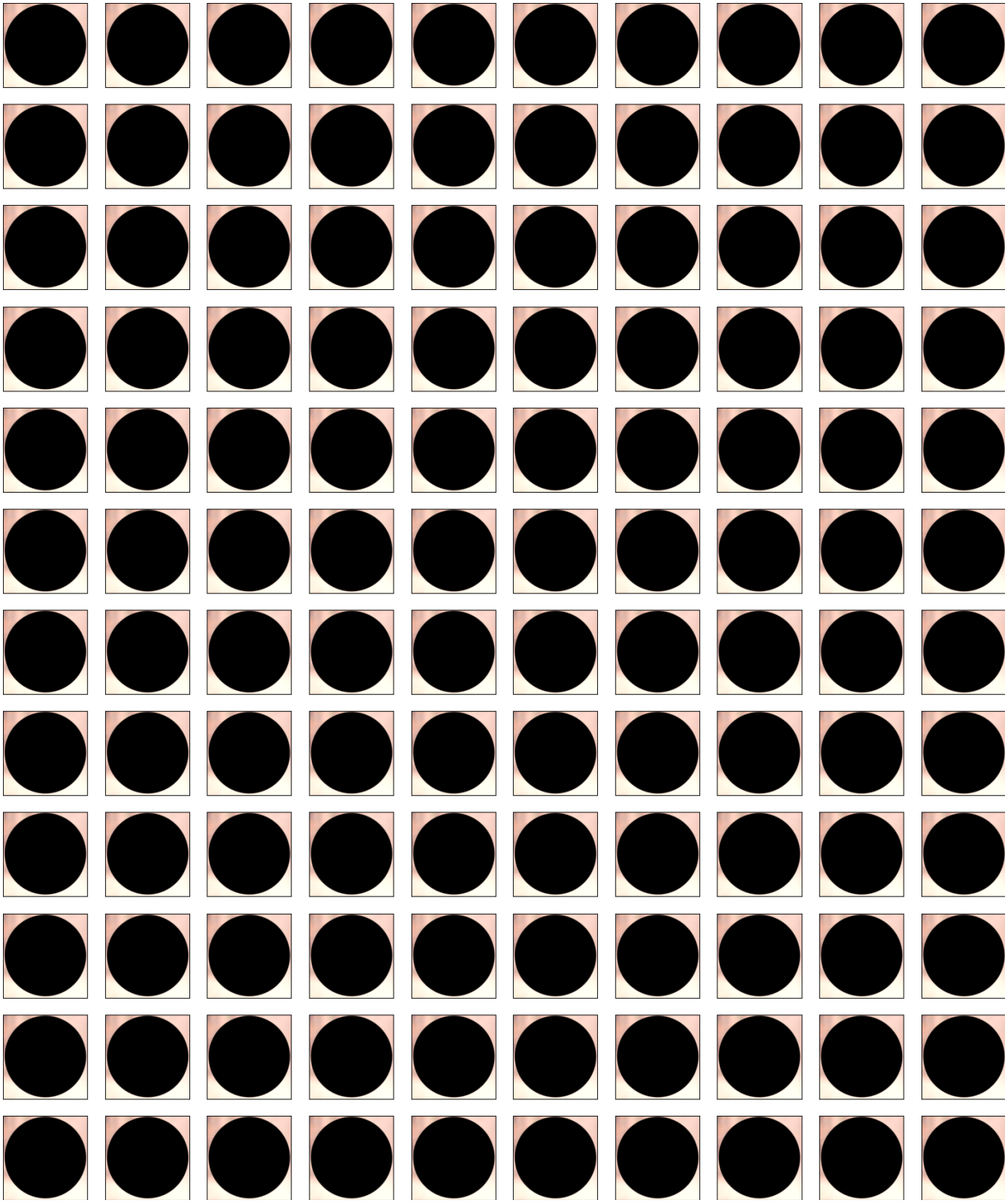


Figure 16: Synthesized materials by VAE-A.

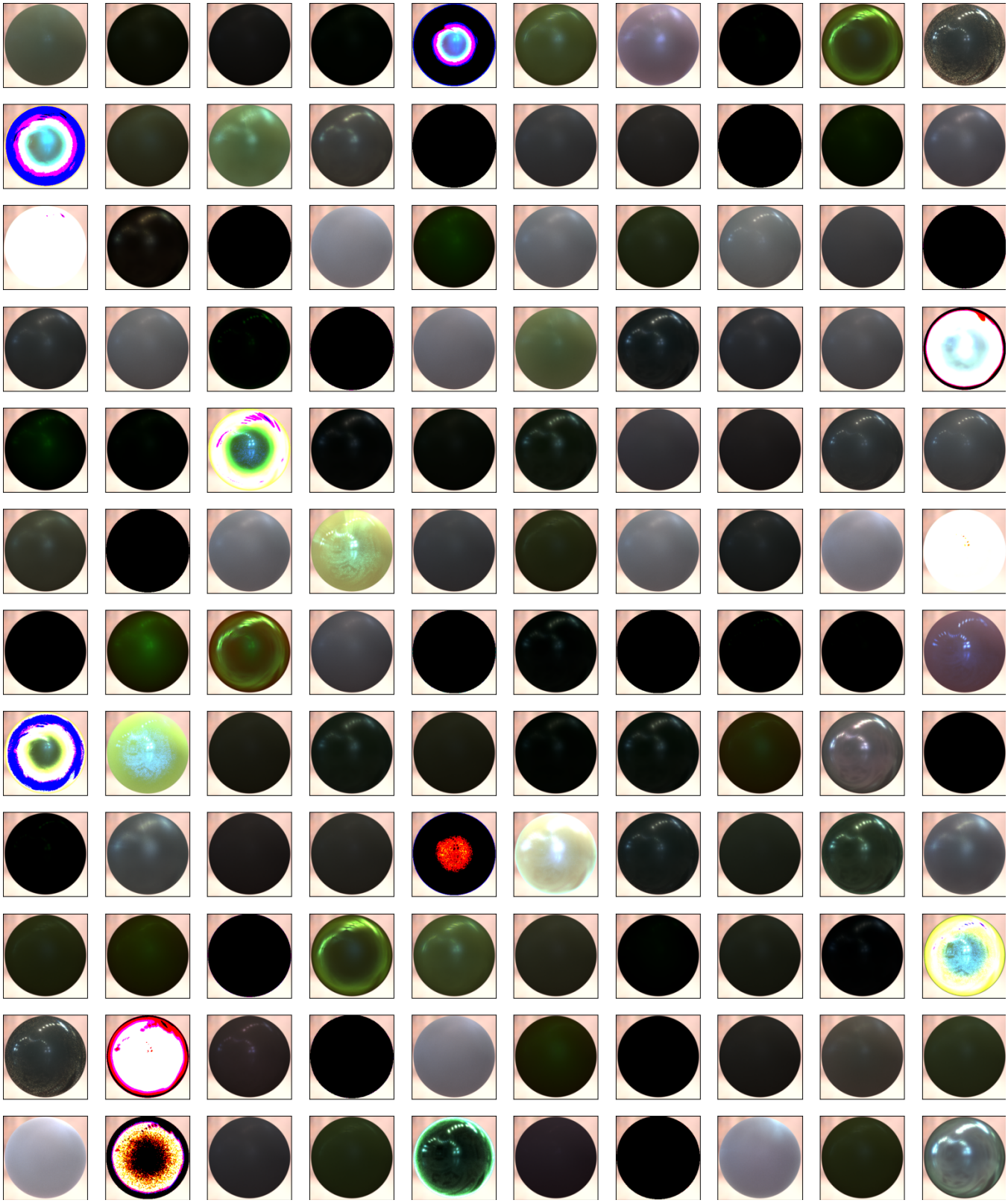


Figure 17: Synthesized materials by VAE-N.



Figure 18: Renderings of different 3D models using our synthesized neural materials, highlighting the quality and diversity.

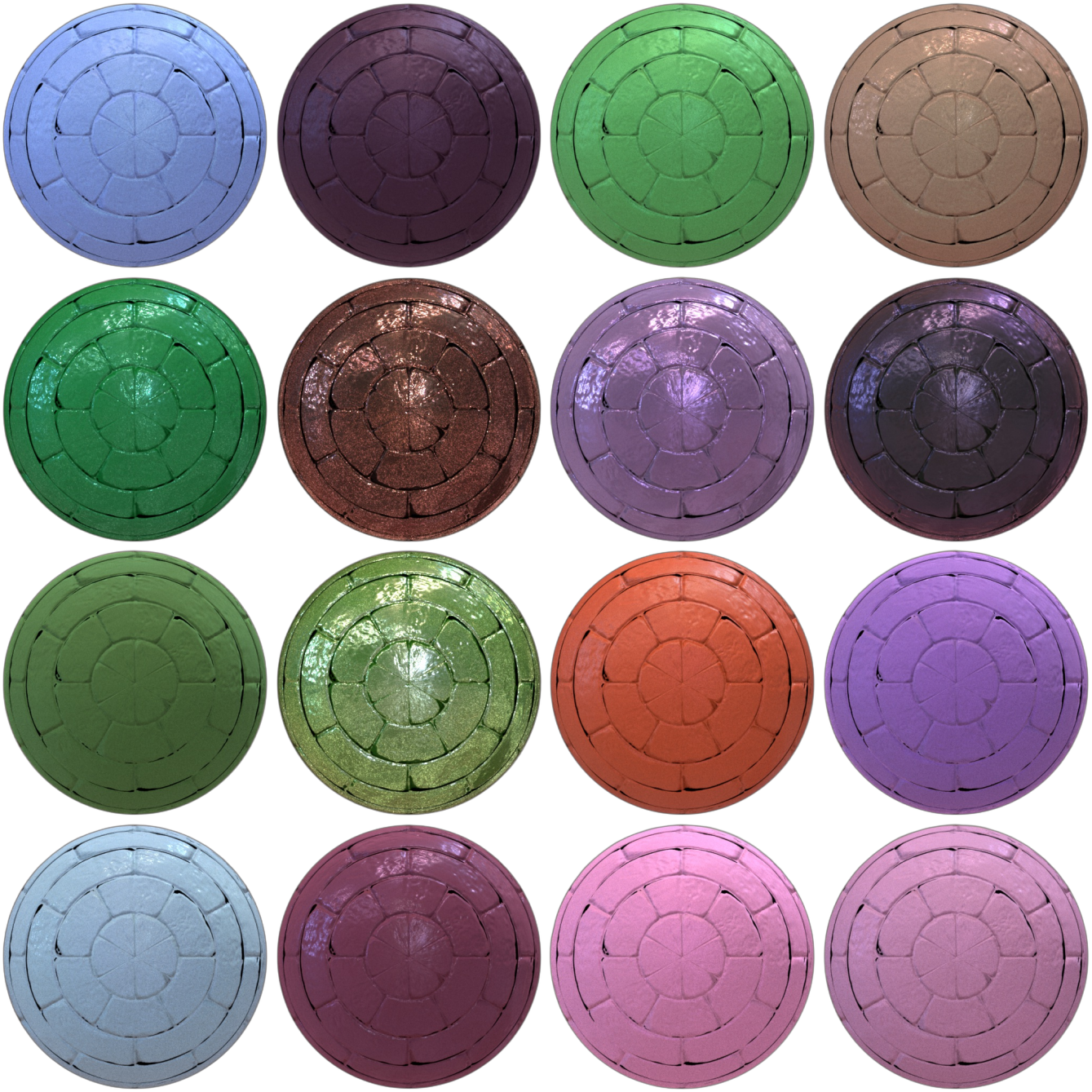


Figure 19: Our synthesized neural materials rendered with normal maps.

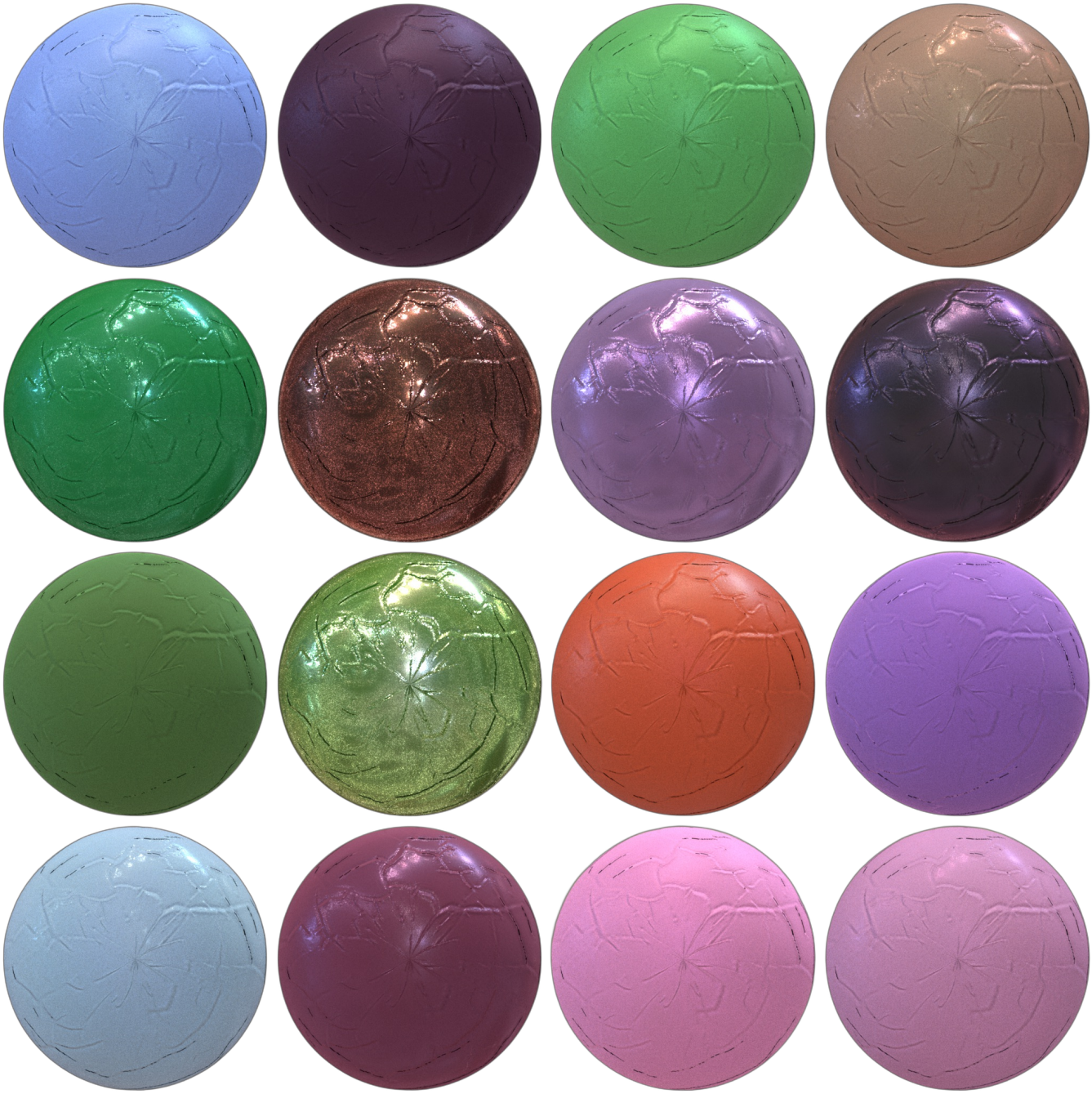


Figure 20: Our synthesized neural materials rendered with bump maps.

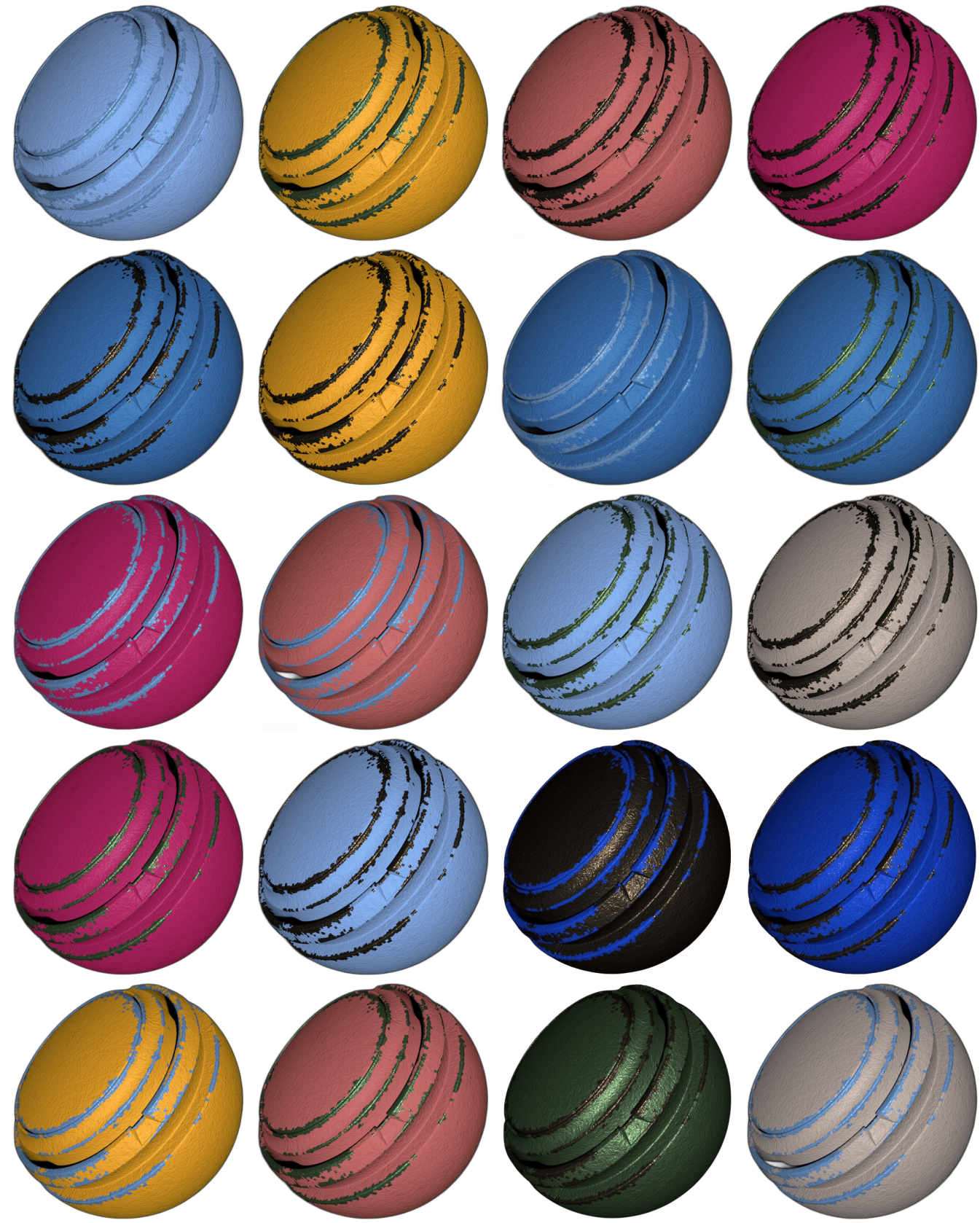


Figure 21: Our synthesized neural materials rendered in a spatially varying.



Figure 22: Renderings of different 3D models using our synthesized neural materials, highlighting the quality and diversity.

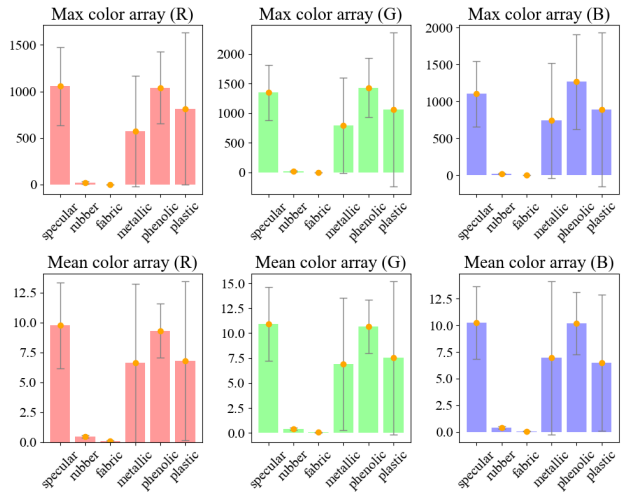


Figure 23: Per-type averaged mean and maximum reflectance.

# samples	x	baseline	NBRDF
$90^2 \times 180$	1x	**1 ± 0**	0.9987 ± 7e-6
$23^2 \times 45$	4x	**0.996 ± 9.3e-5**	**0.996** ± 0.00012
$12^2 \times 23$	8x	0.984 ± 0.001	**0.9934 ± 1.475e-4**
$6^2 \times 12$	16x	0.915 ± 0.024	**0.962 ± 0.0067**
$4^2 \times 8$	24x	0.823 ± 0.0527	**0.896 ± 0.02614**
$3^2 \times 6$	32x	0.745 ± 0.07467	**0.831 ± 0.04942**

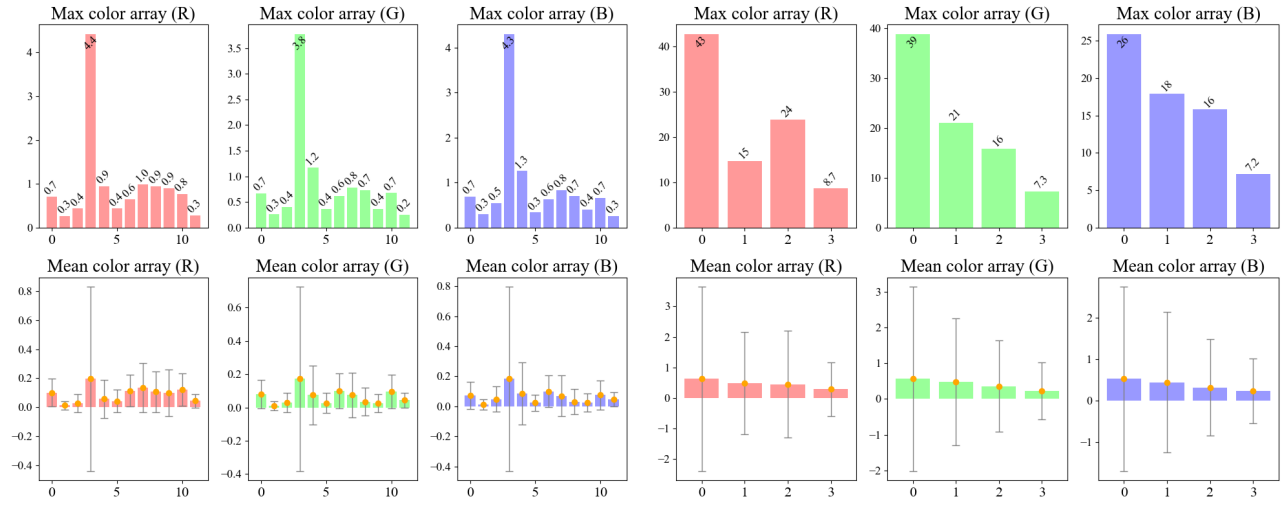
Table 4: Low-density reconstruction comparison (SSIM)

$x = 1, 2, 4, 8, 16$ in all three angles. Since the original sampling density is $90 \times 90 \times 180$, after downsampling, the number of input samples is

$$(1 + \lfloor \frac{89}{x} \rfloor) \times (1 + \lfloor \frac{89}{x} \rfloor) \times (1 + \lfloor \frac{179}{x} \rfloor).$$

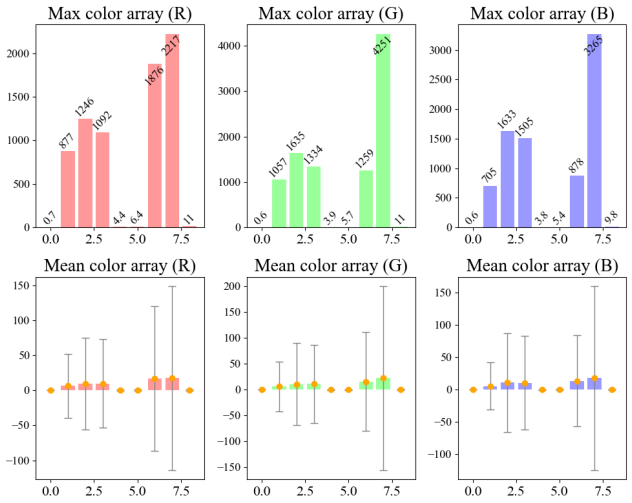
For comparison we develop a baseline model, which just evaluates the BRDF according to the nearest neighbor. On the other hand, NBRDF model is trained over this downsampled data. Under the same scene setting, we compare SSIM of rendered images using the full-density groundtruth, the baseline, and the NBRDF model in Tab. 4.

From the Table, we see that the NBRDF model exhibits significantly better reconstruction capability over the MERL dataset. This capability might be useful in the scenarios where high-resolution BRDF collection is not available.

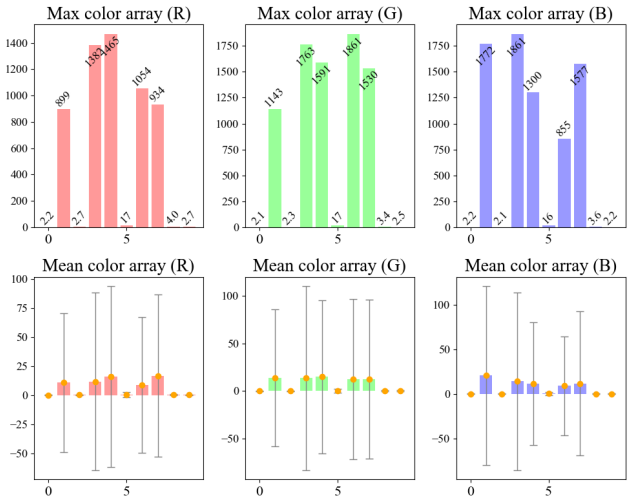


(a) Fabric BRDF statistics

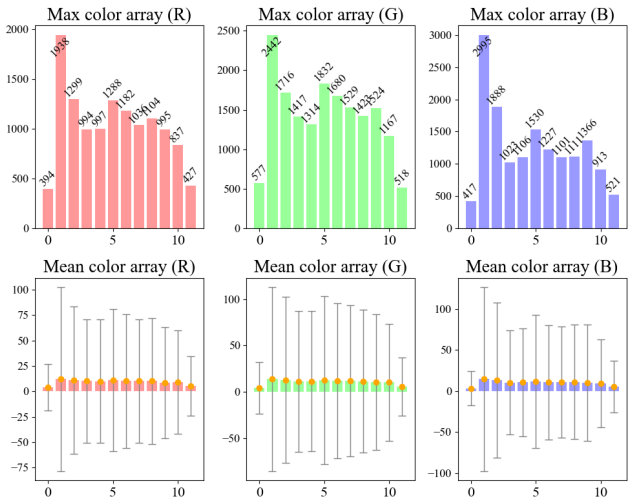
(b) Rubber BRDF statistics



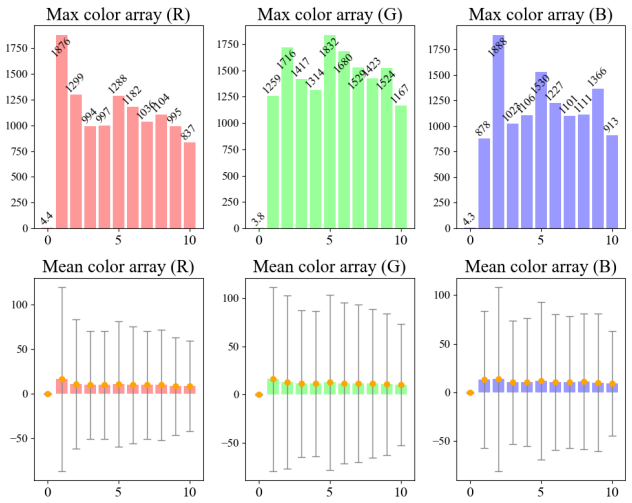
(c) Plastic BRDF statistics



(d) Metallic BRDF statistics



(e) Phenolic BRDF statistics



(f) Specular BRDF statistics

Figure 24: Full MERL BRDF (Matusik et al. 2003) statistical analysis grouped by material type.

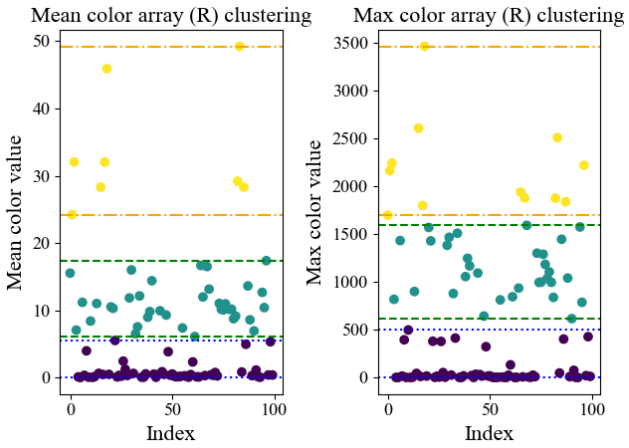


Figure 25: K-means clustering on mean (left) and maximum (right) reflectance of red channel.