Toward Richer Material Generation via Procedural Data Enhancement

YUNCHEN YU, Cornell University & NVIDIA, USA
JACOB MUNKBERG, NVIDIA, Sweden
JON HASSELGREN, NVIDIA, Sweden
CHRIS CUMMINGS, NVIDIA, United Kingdom
STEVE MARSCHNER, Cornell University & NVIDIA, USA
ANDREA WEIDLICH, NVIDIA, Canada

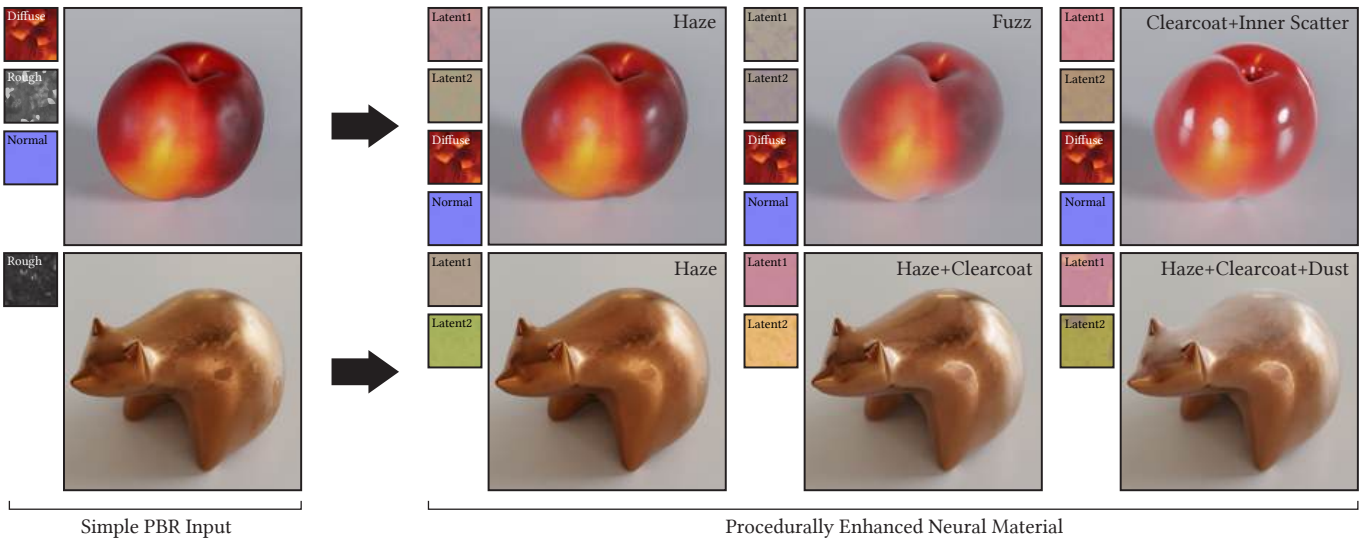

Fig. 1. Creating complex materials at scale is challenging. We introduce a material enhancement procedure that uplifts PBR materials into visually rich appearances that can be compactly represented as neural materials. By adding surface effects, we can make an object appear waxy, peach-fuzzed, or sugar-coated (top row). Similarly, we can introduce subtle spatial specular breakup, as well as large-scale changes like clearcoat or dust layers (bottom row). The small insets show the textures used by each model. Normal maps are applied unchanged to the neural material.

Generative models for material creation are fundamentally limited by the quality and expressivity of available training data. Simple physically based rendering (PBR) materials, which combine a diffuse term with a single-lobe specular component, are commonly used for training but are insufficient to capture many important visual effects present in real materials.

We present a method that enhances such simple PBR materials to more expressive ones, by augmenting the single GGX specular lobe into a layered model that captures a broader range of non-diffuse effects. Starting from a simple material, we procedurally construct a corresponding multi-lobe non-diffuse component guided by physical priors, enabling effects such as dust, clearcoat, and layered scattering. To provide a compact representation for downstream applications, we encode this non-diffuse component as a neural material with a shared 6D latent space, where each material instance is represented by two latent textures and decoded by a pretrained universal MLP. We further regularize the latent space to support material generation.

The resulting neural material dataset enables training generative models for richer material creation. To demonstrate this application, we finetune a video diffusion model to produce neural latent textures that encode our multi-lobe material, and present generative results as proof of feasibility. Our procedural data enhancement approach is an important step toward improving expressivity in material generation.

CCS Concepts: • **Computing methodologies** → **Reflectance modeling**; **Artificial intelligence**.

Additional Key Words and Phrases: procedural modeling, data augmentation, neural materials, video diffusion models, material generation

ACM Reference Format:
Yunchen Yu, Jacob Munkberg, Jon Hasselgren, Chris Cummings, Steve Marschner, and Andrea Weidlich. 2026. Toward Richer Material Generation via Procedural Data Enhancement. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '26), July 19–23, 2026, Los Angeles, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3799902.3811210

Authors' Contact Information: Yunchen Yu, yy735@cornell.edu, Cornell University & NVIDIA, USA; Jacob Munkberg, jmunkberg@nvidia.com, NVIDIA, Sweden; Jon Hasselgren, jhasselgren@nvidia.com, NVIDIA, Sweden; Chris Cummings, ccummings@nvidia.com, NVIDIA, United Kingdom; Steve Marschner, srm@cs.cornell.edu, Cornell University & NVIDIA, USA; Andrea Weidlich, aweidlich@nvidia.com, NVIDIA, Canada.

SIGGRAPH Conference Papers '26, Los Angeles, CA, USA

ACM ISBN 979-8-4007-2554-8/2026/07
https://doi.org/10.1145/3799902.3811210

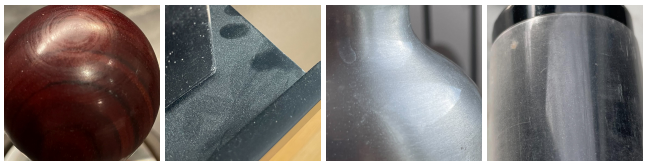

Fig. 2. Simple PBR material models cannot reproduce complex surface effects of real materials such as (from left to right) double highlights (haze), dust, clearcoat, or thin translucency.

1 Introduction

Generative models recently emerged as a new way of authoring materials. However, the quality of these models is inherently bound to the quality and coverage of input training data. While large datasets exist for simplified appearance models using the basic combination of a diffuse and a GGX specular BSDF [Walter et al. 2007], these models fail to encompass the visual richness of real-world surfaces.

As can be seen in Fig. 2, real-world materials show effects like clearcoat, dust and other imperfections, which are necessary to achieve realistic appearances but require more expressive models. Such models include OpenPBR [Andersson et al. 2025], the Disney Principled BSDF [Burley 2012], or even custom BSDFs and layering solutions like MaterialX [Smythe and Stone 2017], and are commonly used in VFX. However, large-scale datasets for these high-fidelity models do not exist. Since manually authoring complex materials is prohibitively expensive, the field lacks the data necessary to train the next generation of expressive material synthesizers.

Unlike prior work that focuses on enhancing texture resolution [Gauthier et al. 2024] or detail while retaining simplified shading models, leaving the BSDF representation untouched [Hadadan et al. 2025], our goal is to expand the appearance space of the BSDF itself. To achieve this, we propose a procedural dataset augmentation pipeline that uses simple, widely available low-dimensional material data as input and uplifts it into a more expressive layered BSDF representation, guided by physical principles and domain priors. We further encode our uplifted data into a compact, 6D universal neural representation, with a regularized latent space suitable for downstream generative applications.

The contribution of this work is a pipeline (Fig. 3) for constructing a large dataset of expressive neural materials from existing PBR datasets: we describe our per-material procedural enhancement process, explaining design decisions (Section 3), and discuss our neural compression of the enhanced materials (Section 4). As an example demonstration, we use our resulting dataset to train a generative model that can produce materials with improved visual richness compared to simple PBR materials.

2 Related Work

Our work builds on prior advances in procedural material modeling, neural materials, and generative models, as reviewed in this section.

Procedural Material Enhancement. Computer graphics has long relied on specialized models to capture distinct surface appearances. Dedicated models exist for e.g. fuzz [Zeltner et al. 2022], dust [Lucas et al. 2024], haze [Barla et al. 2018], and clearcoat [Weidlich and Wilkie 2007], but these effects are often underrepresented in training data for generative material models. Structural enhancements on analytical materials are typically restricted to microscale effects like the addition of flakes [Atanasov and Koylazov 2016; Chermain et al. 2020; Xing et al. 2024], scratches [Raymond et al. 2016; Werner et al. 2017] and other imperfections [Dong et al. 2016; Kuznetsov et al. 2019]. To expand the appearance space, materials can be synthesized from measured data [Zhou et al. 2026], but such data lacks spatial variation. We consider texture resolution improvements or techniques that purely operate on PBR textures like [Chen et al. 2025] orthogonal to our work; these textures are inputs to our uplifting.

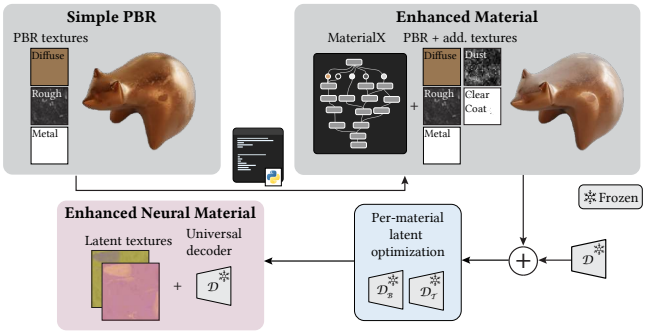

Fig. 3. Our uplifting pipeline. Procedural rules create an enhanced material as MaterialX graph while reusing existing textures. We use a universal decoder $\mathcal{D}$ with frozen weights and optimize only the latent textures.

Our uplifted BRDF is a deterministic, analytic model that avoids precomputation [Jakob et al. 2014; Zeltner and Jakob 2018], making training and evaluation efficient relative to position-free layering [Guo et al. 2018]. Neural layering is another option [Fan et al. 2022; Guo et al. 2023], but our analytic form is simpler and easier to extend to new materials. Volumetric methods [Wang et al. 2022] are less useful, as input data is lacking. Unlike fixed-structure models such as OpenPBR, our model is easy to extend through adding BSDFs.

Neural Materials. Neural materials [Kuznetsov et al. 2021; Zeltner et al. 2024] replace analytical BSDFs and textures with small neural networks and corresponding latent textures. They provide a powerful way of representing self-shadowing, fibers, and complex micro-structures. Recently, there is also a *generative* neural material approach [Raghavan et al. 2025] for 2D material patches, which combines a universal MLP defining a basis for neural materials and a conditional diffusion model for generating neural material latents from text or images. In this work, we extend the generative neural material approach to texture-mapped 3D objects, creating object-specific materials with enhanced expressivity.

Diffusion Models. Image diffusion models add noise to an image through a sequence of diffusion steps. They are trained to reverse this process, enabling sample generation by iterative denoising starting from Gaussian noise. Many generative models have been developed based on similar principles [Dhariwal and Nichol 2021; Ho et al. 2020; Sohl-Dickstein et al. 2015]. Video diffusion models [Blattmann et al. 2023a,b; Hong et al. 2023; NVIDIA 2025; Wan

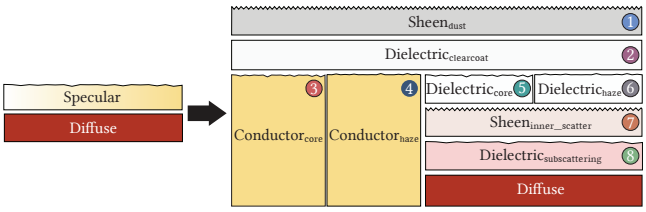

Fig. 4. A schematic illustration of our enhanced BRDF. The specular component includes eight lobes, while the diffuse component is unmodified. Lobes 3, 4, 5 and 6 are the basic core and haze reflection lobes, whereas lobe 1, 2, 7 and 8 are what we call "decorator lobes."

2025; Yang et al. 2024b] extend image-based diffusion to the temporal domain, enabling video generation from text or images. In our application of generative neural materials, we build upon the Cosmos [NVIDIA 2025] video diffusion model.

Material Extraction using Diffusion. Various approaches combine image diffusion models with inpainting, or texture refinement [Chen et al. 2023; Richardson et al. 2023; Yeh et al. 2024; Zeng et al. 2024]. By finetuning image diffusion models, conditioning on geometry and lighting [Deng et al. 2024; Zhang et al. 2024a], light and material disentanglement are improved.

Another line of research leverages databases of PBR materials, and learns to project the input (image or text) onto the known representation [Ceylan et al. 2024; Fang et al. 2024; Zhang et al. 2024b]. These methods are limited by the expressiveness of their material databases, but benefit from improved regularization.

By finetuning multi-view image diffusion models [Shi et al. 2023] conditioned on normal maps, per-object materials can be generated from text or image inputs. These approaches [Boss et al. 2025; Zhang et al. 2024c] generate a set of views of PBR texture maps, which are projected into texture space. Recent methods [Chen et al. 2024; Engelhardt et al. 2025; Feng et al. 2025; He et al. 2025; Seed 2025; Shao et al. 2025; Yang et al. 2025] extend this approach with additional conditioning (e.g. depth and/or world space positions).

A common limitation is lack of view consistency, which may result in blurring in the extracted textures. By finetuning *video models* for object rotation, view consistency is improved [Hasselgren et al. 2026; Munkberg et al. 2025; Voleti et al. 2024; Yang et al. 2024a]. The view consistency problem can be avoided by having the diffusion model operate directly in 3D space [Xiang et al. 2025, 2024; Yu et al. 2024b], but these approaches require large-scale supervision with high quality 3D models.

3 Procedural Enhancement of Simple PBR Materials

Our goal is to build a dataset of physically plausible materials that spans a wide range of appearances. We represent these materials using a multi-lobe BRDF capable of expressing diverse visual phenomena (Section 3.1), and construct them by procedurally enhancing simple PBR materials from existing datasets (Section 3.2).

3.1 Generalized Non-Diffuse Reflection Model

A simple PBR material consists of a Lambertian diffuse lobe and a GGX specular lobe; here, we consider PBR materials parameterized by base color, roughness, and metalness maps [Karis 2013; Khronos 2025]. In our proposed BRDF model, shown in Fig. 4, we replace the single GGX specular lobe with a multi-lobe non-diffuse component while preserving the Lambertian diffuse term. Maintaining separate diffuse and non-diffuse components in our design enables flexible editing of each part (see Section 5.3), and makes rendering tasks such as denoising (requiring albedo) remain straightforward.

Our non-diffuse component consists of eight lobes combined via coating and mixing, organized into conductive and dielectric branches. The conductive branch models metallic reflection with two lobes of different roughness, denoted *core* and *haze*; the dielectric branch similarly contains two lobes layered above the original diffuse component, with optional scattering lobes in between. The two branches are combined through mixing, and additional decorator lobes may be layered on top. The formulation and parameters used for each lobe are guided by domain knowledge from material science and VFX production materials [Emrose et al. 2022; Kulla and Conty 2017; Matusik et al. 2003; Ngan et al. 2005; Polyanskiy 2025].

Core and Haze Lobes. 3 4 5 6 In the specular component of a simple PBR model, base color and metalness are typically used together to define an effective Fresnel reflectance (e.g. [Karis 2013]); moreover, a single surface roughness is used. In contrast, every material we construct includes two conductive GGX lobes and two dielectric ones. We model metallic and non-metallic reflection separately using conductive and dielectric GGX lobes to preserve their respective Fresnel behavior. In addition, both conductive and dielectric branches adopt two correlated roughness scales (see Section 3.2): a lower-roughness *core* lobe that produces a sharp highlight, and a higher-roughness *haze* lobe that captures blurrier reflection. Together, these lobes produce characteristic double-highlight appearances that often make materials more realistic, as shown in Fig. 1 (Haze) and Fig. 2 (leftmost).

Decorator Lobes. To expand the range of achievable appearances, we also include four optional decorator lobes, each modeling a distinct type of surface or volume scattering, in thin media above or below the main specular reflection.

Dust/Fuzz 1 Our top-layer decorator is a sheen lobe [Kulla and Conty 2017] that captures strongly view-dependent effects coming from dust or fuzz (short hair). The strength and angular dependence of this lobe are controlled by a weight and a thickness parameter. We use the term "dust" if the weight of this lobe is modulated by the object's surface normal, and use "fuzz" if the weight is uniform.

Clearcoat 2 Clearcoat is modeled as a low-roughness GGX lobe under the dust/fuzz lobe. Coated upon both the conductive and dielectric GGX lobes, it models a thin, glossy coating like glazing. The lobe is controlled by a binary weight (1 if present) and an IOR.

Inner Scattering 7 We use another sheen lobe beneath the dielectric GGX lobes to model view-dependent scattering within thin layers. It primarily affects grazing angle appearances by introducing soft, retro-reflective scattering. The lobe is controlled by a weight, a thickness parameter, and a scatter color.

Subcutaneous Scattering 8 This is a rough, tinted GGX lobe under the inner scatter lobe, modeling subtle, low-order scattering from the otherwise opaque-looking bulk material and producing a soft,

Fig. 5. Enhancement effects used in our dataset as compared to the simple PBR model. Our haze enhancement (Type 0) is applied to all materials, while Types 1—5 correspond to different combinations of decorator lobes.

gently colored highlight. It is controlled by a binary weight, a high roughness, and a scatter color loosely coupled to the base color.

Details on all lobes, their parameter ranges and constraints, and isolated effects of each lobe are in our supplemental material (Sec1).

3.2 Procedural Enhancement of Simple PBR Materials

We now describe how a specific simple PBR material instance is procedurally enhanced into a multi-lobe material from Section 3.1.

In order to more accurately represent specular reflection from both conductors and dielectrics and improve upon the limited appearance of single-roughness highlights, our baseline enhancement—on all materials—replaces the single GGX lobe with four specular lobes, mapping the original base color, roughness, and metalness textures to the parameters of our core and haze lobes.

First, we use metalness to mix the conductive and dielectric branches without further modification, preserving its intended meaning as an indicator of whether a material is metallic or dielectric. While this may allow "half-metal" materials (e.g. metalness 0.5), in practice, most maps in PBR datasets are near-binary. The complex-valued IOR of the conductive GGX lobes are derived from the simple PBR base color, following [Gulbrandsen 2014], while the IOR of the dielectric GGX lobes is sampled separately from a bounded range.

Moreover, we use the single roughness texture from the simple PBR material to compute two correlated roughness scales for our core and haze lobes. Let $r \in (0, 1)$ be the input roughness, typically converted to GGX roughness α via $\alpha = r^2$. We instead construct

$$\alpha_{\text{core}} = r^p, \qquad \alpha_{\text{haze}} = s \cdot r^q, \tag{1}$$

where $p > 2$, $q < 2$, and $s > 1$ are per-material scalars sampled within their ranges (see supplemental Sec2). In our design, the same $(\alpha_{\text{core}}, \alpha_{\text{haze}})$ pair is shared by both dielectric and conductive GGX lobes to preserve the artistic intention of the input roughness data.

This process defines our *haze-only* enhancement, denoted as *Type 0*. We include five other appearance types by enabling additional decorator lobes (Fig. 5), with the last two types (4–5) applied only to dielectric materials. These types span a range of additional appearances but do not exhaust all possible combinations. Parameters of decorator lobes (e.g. dust thickness, clearcoat IOR) are sampled from physically meaningful ranges to ensure realistic results, with some (e.g. subcutaneous scatter color) loosely tied to the input PBR parameters to maintain consistency with the bulk material.

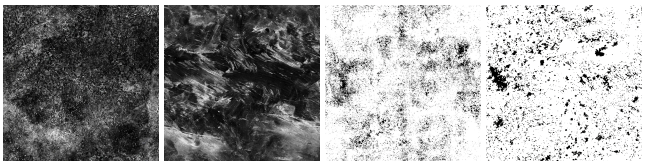

Fig. 6. Example masks that modulate spatial variations in dust (left images) and clearcoat (right images), simulating fingerprints or ripple-like patterns on dust layers and dents or scratches on shiny coatings.

Type 1: Dust (1) We enable the dust/fuzz lobe by setting its weight nonzero. To simulate dust accumulation, this lobe weight is driven by the world-space normal vectors of the object, favoring upward-facing surfaces, and further shaped by spatially varying masks that introduce features such as fingerprints on dust applied through triplanar mapping. Examples of dust variation masks are in Fig. 6.

Type 2: Clearcoat (2) We enable the clearcoat lobe by setting its weight to 1 across the material, and optionally modulate the weight texture with binary masks, to simulate small damages on the coating.

Type 3: Dust + Clearcoat (1)(2) Combining Type 1 and Type 2, producing surfaces that have a glossy coating and a dust layer.

Type 4: Fuzz + Inner- and Sub-Scatter (1)(7)(8) We apply the dust/fuzz lobe with uniform weight over dielectric objects to produce a fuzzy look (see Fig. 1, peach fuzz). To further soften the appearance, we also enable the inner and subcutaneous scatter lobes in the dielectric branch. The scatter color, shared by both lobes, is derived from the base color via gentle desaturation.

Type 5: Fuzz + Clearcoat + Inner- and Sub-Scatter (1)(2)(7)(8) All lobes are enabled, with the dust/fuzz lobe applied uniformly. The inner- and sub-scattering lobes soften the appearance at grazing angles. If the fuzz weight is high, the fuzz reduces the sharpness of the clearcoat, yielding a milky or gel-covered appearance.

Details of each enhancement type are in the supplemental material (Sec2), along with comparisons to simple PBR appearances. Currently, we allow any enhancement type on any input material, which might produce uncommon appearances. A natural extension is to guide our type selection using object/material recognition.

4 Neural Representation of Enhanced BRDFs

Our enhanced BRDF contains a Lambertian diffuse lobe and a multi-lobe non-diffuse component. The non-diffuse component requires 22 parameters (supp-Sec1) and is impractical to use directly for generative material applications. Instead, we represent the non-diffuse component as a neural material with a compact 6D latent space—compressing the 22 parameter textures into two RGB latent textures—and combine it with the diffuse lobe to form the full BRDF.

4.1 Neural Material Formulation

As in prior work on neural materials [Zeltner et al. 2024], we train a single encoder–decoder MLP pair to represent the space of non-diffuse materials expressible by our multi-lobe model. After training, the network weights are frozen [Raghavan et al. 2025], defining a

Fig. 7. Comparison of informed vs. random training data and their ꟻLIP error images [Andersson et al. 2020]. We reconstruct the same reference material with both approaches. The network built with informed data captures the reference material significantly better than the one with random data.

latent space shared across material instances. Individual materials are then represented by distinct latent textures in this 6D space.

Our universal decoder MLP consists of four hidden layers with 64 neurons each, and uses an exponential activation function. Given a 6D latent vector, which represents a non-diffuse material, along with a pair of queried directions (ω_i, ω_o), the decoder outputs a BRDF value $f_{\text{neu}}(\omega_i, \omega_o)$, as well as a transmission albedo $T_{\text{neu}}(\omega_i)$ and a luminance reflectance value $R_{\text{neu}}(\omega_i)$.

In particular, $T_{\text{neu}}(\omega_i)$ represents the fraction of incident energy transmitted through the non-diffuse material f_{neu}; it is used to enforce energy conservation when layering f_{neu} above a Lambertian diffuse lobe to build the full, enhanced BRDF. Specifically, using c to denote the PBR base color, our combined BRDF is

$$f(\omega_i, \omega_o) = f_{\text{neu}}(\omega_i, \omega_o) + T_{\text{neu}}(\omega_i)\,\frac{c}{\pi}, \tag{2}$$

Moreover, $R_{\text{neu}}(\omega_i)$ is a scalar measure of reflected energy in f_{neu}, used for importance sampling *between* this non-diffuse component and the diffuse lobe during rendering. In addition to the evaluation decoder, we also train a sampling decoder following [Zeltner et al. 2024], used for importance sampling *within* the non-diffuse BRDF.

4.2 Universal MLP Training

We next describe our data used for training the universal neural material, as well as practical challenges encountered in the training.

Training Data. Training the universal neural material does not require a large curated material database, as our goal is to learn the directional behavior of BRDFs represented by our multi-lobe model rather than how material parameters may vary across a textured surface. Thus, we generate training materials procedurally, matching the statistical distributions of the materials we hope to represent.

Our training data comprises five material classes. Four are *themed* classes targeting specific appearance effects: *haze*, *dust*, *clearcoat*, and *inner/subcutaneous scattering*. These classes enable corresponding decorator lobes alongside optional additional lobes, allowing the network to learn both isolated and combined effects. Each themed class contains an 8192×8192 grid of material data points, generated from a total of 4096 distinct base materials per theme. For each base material, all parameters (except specular color) are held constant while roughness is densely sampled, yielding 16384 (128×128) roughness variations per base material. To ensure physical plausibility and prevent the network from wasting capacity on unlikely combinations, base material parameters are sparsely sampled from a predefined data distribution. In Fig. 7, we compare training with our physically informed, biased data distribution and with purely random material data, where both setups use the same training time, network size, and number of material samples.

In addition, we include a fifth *infill* class to populate the latent space more uniformly (see "latent space regularization"). In this class, each cell in the 8192×8192 parameter grid corresponds to a material whose parameters are sampled independently and uniformly in their ranges. This gives rise to another 67M randomly parameterized materials. Visualization of all material classes, physically informed training data preparation, and training details are in the supplemental document.

Latent Space Regularization. We aim to obtain a neural material with a latent space suitable for downstream generative material creation. However, our appearance space is substantially larger than that of prior work (e.g. [Raghavan et al. 2025]) due to our material uplifting. For our generative process we need a low-dimensional latent space, but compressing 22 parameters into six channels leads to an entangled mapping from original parameters to latent channels, making it challenging to build a well-behaved latent space.

Early on, as we finetuned diffusion models (Section 5.2) on small datasets, diffusion-generated latent codes sometimes decoded to invalid materials (producing Infs or other artifacts), spoiling up to 50% of our outputs. This suggested that the neural material latent space had useful materials packed closely against invalid regions, making small errors in generation result in unusable outputs. To resolve this, we made two training adjustments to help smooth the mapping between materials and latent codes.

First, adding the already mentioned *infill* class containing random input materials helped populate the latent space more uniformly. Since this class is only 20% of our training data, the reconstruction quality was not significantly affected by including random, possibly implausible materials. Second, we injected small uniform noise in the range of ±0.005 into latent values when evaluating the reconstruction loss. This requires the decoder to accurately reconstruct BRDFs from slightly perturbed latent codes and made our neural model less sensitive to perturbations in latents and more robust for generative use. The bound of 0.005 was chosen experimentally based on our training data, as the largest value we could adopt without leading to a large increase in training loss. While we did preliminary experiments with a learned jittering noise range instead of a fixed one, we did not observe improvements over our simple jittering strategy. Overall, we found that combining random infilling and latent jittering largely eliminated artifacts in material generation.

4.3 Per-Material Optimization

With a pretrained universal decoder, we can then convert analytical materials enhanced from simple PBR inputs, initially described by many parameter maps, into neural materials specified with 6D latent codes—only two RGB textures. For each individual material, we optimize its latent textures against the frozen universal decoder [1].

[1]Latent codes directly computed with the analytical parameters and the universal encoder are not sufficiently accurate for materials not present in the training data.

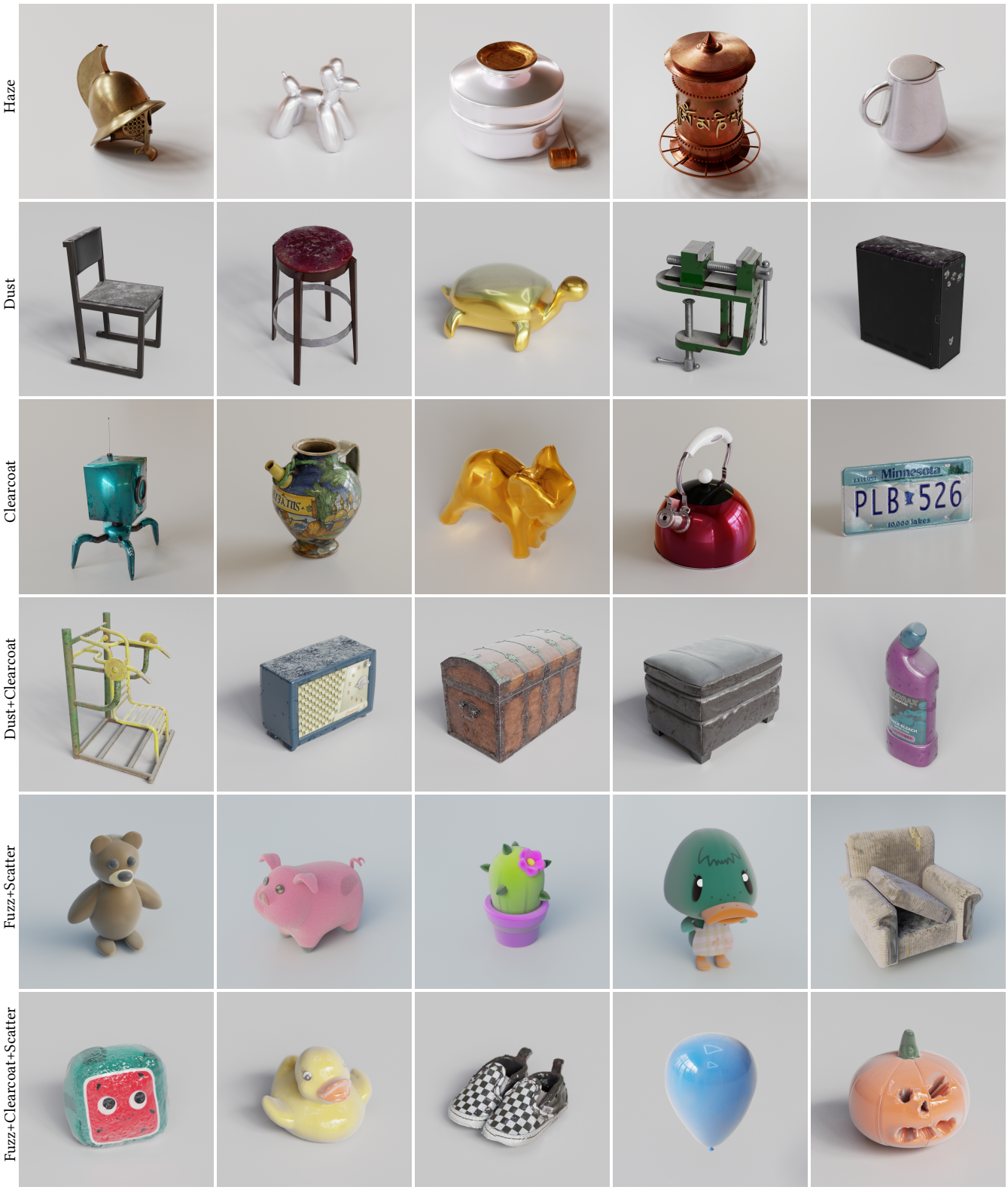

Fig. 8. 3D Models from BlenderVault [Litman et al. 2025] and Objaverse [Deitke et al. 2023] with materials uplifted with our procedural data enhancement. The materials have been encoded into a compact neural material representation suitable for real-time rendering and large-scale data generation.

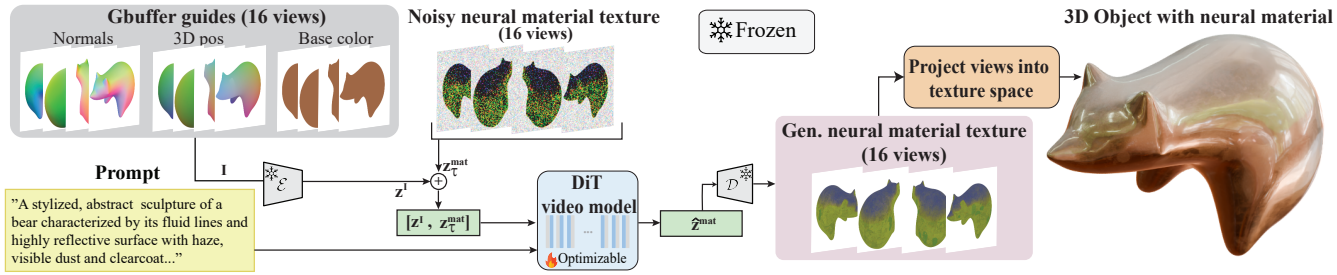

Fig. 9. Our generative pipeline starts from a known 3D model and a text prompt. We first render views of normals, world space positions, and base color. These conditions are encoded using a frozen VAE encoder, $\mathcal{E}$, to produce latent conditions, $\mathbf{z}^{\mathbf{I}}$. These are concatenated with noisy latents, $\mathbf{z}_\tau^{\mathbf{mat}}$, representing neural material textures. The latents and text prompt are passed to a finetuned video model, which generates a denoised latent, $\hat{\mathbf{z}}^{\mathbf{mat}}$, which is decoded into views of neural material textures, using a frozen VAE decoder $\mathcal{D}$. Finally, we project the generated views into texture space to extract a neural material.

Table 1. Best, medium, and worst ꟻLIP error for each type.

Type	Best	Medium	Worst
Type 0	0.0487	0.0920	0.1871
Type 1	0.0312	0.0727	0.1670
Type 2	0.0649	0.0858	0.1192
Type 3	0.0554	0.0877	0.1664
Type 4	0.0276	0.0643	0.1705
Type 5	0.0454	0.0807	0.1754

Given analytical parameter textures that describe an enhanced non-diffuse BRDF, we need to minimize the errors between the decoded outputs from the latents and their analytically computed references. Specifically, we use an ℓ_1 loss that combines three weighted error terms to encourage accurate reconstruction of the BRDF, transmission albedo, and luminance reflectance, across all texels and over all incident–outgoing direction pairs:

$$\mathcal{L} = \lambda_f \left\| f_{\text{neu}} - f_{\text{ref}} \right\|_1 + \lambda_t \left\| T_{\text{neu}} - T_{\text{ref}} \right\|_1 + \lambda_r \left\| R_{\text{neu}} - R_{\text{ref}} \right\|_1, \quad (3)$$

In practice, we found that the BRDF term must dominate the loss, as errors in the BRDF directly lead to appearance discrepancies. When the BRDF is well fitted, the remaining quantities also tend to be reconstructed accurately, as they are strongly correlated with the BRDF. In particular, luminance reflectance is derived from integrating the BRDF and therefore requires only a small weight, while transmission albedo is less coupled to the BRDF and benefits from a slightly higher weight. These observations led to our choice of $\lambda_f = 0.95$, $\lambda_t = 0.04$, and $\lambda_r = 0.01$. To evaluate the fitting quality, we randomly selected 50 examples from each enhancement type, rendered reference and reconstructed materials on preview balls, and generated ꟻLIP error images. We computed the mean ꟻLIP value in each error image and report the best, medium, and worst errors in Table 1 (more visualizations in supplemental material).

The latent textures optimized according to Eq. 3 fully specify a non-diffuse BRDF as described in Section 4.1, and importance sampling of f_{neu} is handled by the sampling decoder[2], as in [Zeltner et al. 2024]. Notably, because f_{neu} and T_{neu} in Eq. 2 are neural approximations, we might still violate energy conservation due to fitting errors (e.g. overestimated T_{neu}). Thus, we filter combined materials with a (weakened) white furnace test: we render objects under a unit-valued environment map and discard materials that exhibit energy gain (pixel values > 1), tolerating small violations due to Monte Carlo noise in rendering. In practice, we find that around 90% of our materials pass this white furnace test, and these valid materials form our dataset of expressive neural materials.

[2]During per-material optimization, we did not include sampling-related terms in the loss. The latent textures still produce effective sampling parameters when decoded, thanks to the correlation between BRDF shapes and importance sampling schemes.

5 Experimental Downstream Application

We now demonstrate an example application of the dataset built with our proposed pipeline, using the data to finetune a video diffusion model to generate neural material latents on 3D objects.

5.1 A Dataset of 3D Objects with Neural Materials

With our enhancement and neural fitting processes discussed earlier, we constructed a dataset of neural materials on 10,640 3D models from Objaverse [Deitke et al. 2023] and BlenderVault [Litman et al. 2025]. Our latent optimizations, combined, took 22 hours on a multi-user cluster with a total of 80 L40 and L40S GPUs. For each asset, we create a Lambertian lobe using the base color texture, scale it with T_{neu}, and replace the GGX specular lobe with an enhanced non-diffuse component, represented by two RGB latent textures. Each asset adopts one of the enhancement types in Section 3.2, with approximate proportions of 10%, 20%, 20%, 20%, 10% and 20% for Types 0–5. Fig. 8 shows selected objects with uplifted materials.

5.2 Generative Neural Materials

Inspired by Munkberg et al. [2025], we synthesize neural materials on 3D shapes by leveraging a recent video diffusion model architecture [Hasselgren et al. 2026] to produce the neural material textures.

As shown in Fig. 9, our pipeline takes a 3D model with a valid texture parameterization and a text prompt describing the material as input. We assume the 3D model has a base color texture; if missing, it can be generated from the prompt using any previous method [Munkberg et al. 2025; Xiang et al. 2025]. The diffusion model generates multiple views of material intrinsics: G-buffers of neural material textures, conditioned on corresponding input

Fig. 10. Examples of our generated neural materials for a selection of 3D models from our test set. Intended enhancement types of these assets' reference neural materials are dust (a, b), clearcoat (c, d), dust + clearcoat (e), fuzz + scatter (f, g, h), all lobes (i, j). Text prompts are provided in the supplemental material. The material in (b) contains some artifacts from projecting views of latents into the texture space, and (e) is a near-failure case as the dust is not pronounced.

geometry (views of surface normals and world space positions). Finally, we project the intrinsic views into texture space using texture splatting, with bilinear filtering, view averaging, and inpainting for uncovered regions. This allows us to bake down a single set of neural textures that, when combined with the existing base color map, defines a full, enhanced material.

Dataset. For each object in our dataset, we render a video with 16 frames at 512×512 resolution, using a path tracer with three bounces and Blender AgX tonemapping. We use black backgrounds and smooth camera orbits, and adopt the "Boiler Room" light probe from Poly Haven [Zaal and et al. 2024] for all objects. We only use the shaded videos to generate captions using Qwen2.5-VL-7B [Qwen 2024], and the fixed light probe avoids prompt noise due to lighting variation; these prompts are then augmented with keywords describing enhancement types. We also render intrinsic maps (normals, world space positions, base colors and neural material textures[3]). We use these data to finetune a recent Diffusion Transformer (DiT) video model, Cosmos [NVIDIA 2025], for neural material generation.

Video Diffusion Model Architecture. Here we use `Cosmos-1.0-Diffusion-7BVideo2World`[4], which supports text and image guided video generation at a resolution of 1280×704 pixels and 121 frames. The base model leverages the pretrained `Cosmos-1.0-Tokenizer-CV8x8x8` to encode and decode RGB videos to and from latent space.

Finetuning. The model comprises the pretrained VAE encoder-decoder pair (the Cosmos Tokenizer mentioned above), $(\mathcal{E}, \mathcal{D})$, and a transformer-based denoising function, $\mathbf{f}_\theta$. Given an input video $\mathbf{I}$ consisting of normals, world space positions, and base colors for N views of a 3D model, our goal is to train $\mathbf{f}_\theta$ to denoise views of neural material textures conditioned on $\mathbf{I}$. We finetune the embedding layer (extended from the base model to support our input conditions) and all DiT layers on 64 A100 GPUs for 20K iterations, using the AdamW optimizer and a learning rate of $5e{-}5$.

We use $\mathcal{E}$ to encode the input conditions, $\mathbf{I}$, into a latent tensor, $\mathbf{z}^{\mathbf{I}}$. The target latent variable, $\mathbf{z}_0^{\mathbf{mat}}$, for our dataset is constructed by encoding the neural material texture, $\mathbf{y}$, which has six channels, using $\mathcal{E}$. This encoder expects RGB input, so we encode three channels at a time and concatenate the encoded results in the frame dimension:

$$\mathbf{z}_0^{\mathbf{mat}} = [\mathcal{E}(\mathbf{y}_{0,1,2}), \mathcal{E}(\mathbf{y}_{3,4,5})]. \tag{4}$$

Noise, ϵ, is introduced to $\mathbf{z}_0^{\mathbf{mat}}$, representing the neural material texture, to produce $\mathbf{z}_\tau^{\mathbf{mat}}$. The parameters, θ, of the diffusion model $\mathbf{f}_\theta$ are optimized by minimizing the objective function:

$$\begin{aligned} \hat{\mathbf{z}}^{\mathbf{mat}}(\theta) &= \mathbf{f}_\theta([\mathbf{z}_\tau^{\mathbf{mat}}, \mathbf{z}^{\mathbf{I}}]; \mathbf{c}_{\text{prompt}}, \tau) \\ \mathcal{L}(\theta) &= \mathbb{E}_{\mathbf{z}_0^{\mathbf{mat}} \sim p_{\text{data}}, \epsilon \sim \mathcal{N}(0,\sigma^2 I)} \left\| \hat{\mathbf{z}}^{\mathbf{mat}}(\theta) - \mathbf{z}_0^{\mathbf{mat}} \right\|_2^2 \end{aligned} \tag{5}$$

where $[\,,\,]$ denotes concatenation in the channel dimension, $\mathbf{c}_{\text{prompt}}$ is the encoded text prompt (encoded by T5-XXL [Raffel et al. 2023]) and τ is the timestep. We increase the input feature count of the input embedding layer of $\mathbf{f}_\theta$ to account for our additional input conditions, $\mathbf{z}^{\mathbf{I}}$. We use the denoising score matching loss with uncertainty-based weighting from Cosmos [NVIDIA 2025], applied to the predicted latent $\hat{\mathbf{z}}^{\mathbf{mat}}(\theta)$ and the corresponding target latent $\mathbf{z}_0^{\mathbf{mat}}$.

Evaluation. We evaluate our generative pipeline on a test set of 16 3D assets with hand-crafted neural materials. We first render images of test objects with their reference neural materials, and annotate these images with captions from Qwen2.5-VL-7B and enhancement type keywords. We then generate neural materials using these text prompts, repeating with five random seeds and subjectively picking the best results, and render 16 views of each asset with the generated material. On our $16 \times 16 = 256$ images, we report the CLIP-based Fréchet Inception Distance (CLIP-FID) [Kynkäänniemi et al. 2023],

[3]We mean neural latents, but try to avoid confusion with the diffusion latent space.
[4]https://github.com/NVIDIA/Cosmos

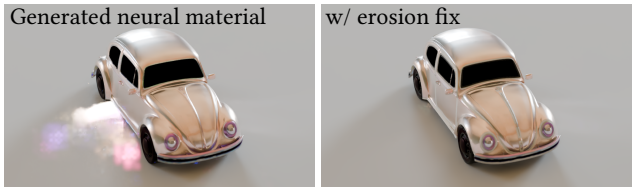

Fig. 11. We occasionally observe artifacts in generated neural materials, likely because the video model only approximates the manifold of valid latent textures. Assets with many UV islands can also introduce errors when projecting generated frames to texture space, especially in poorly covered silhouette regions. As shown above, we mitigate these artifacts by eroding uncovered regions inward by four pixels and using a diffuse material there.

Fig. 12. Generated neural materials using three different prompts applied to the same 3D asset. Full text prompts are in the supplemental document.

CLIP Maximum-Mean Discrepancy (CMMD) [Jayasumana et al. 2024], Learned Perceptual Image Patch Similarity (LPIPS) [Zhang et al. 2018], and the Peak Signal-to-Noise Ratio (PSNR) in the last row of Table 2 (Our_{ref}). Some examples of generated neural materials are shown in Fig. 10, and Fig. 11 shows a failure case. Moreover, by editing the enhancement keyword in the prompt, we can create different neural material variants on an object, as in Fig. 12.

We compare against TRELLIS.2 [Xiang et al. 2025], a recent image-guided PBR material generation method. We use their PBR texture generation pipeline with known geometry, and for each test set example, input a shaded image rendered with our reference neural material, and use TRELLIS.2 to generate PBR material maps. TRELLIS.2 produces base color, roughness, and metalness textures, and our quantitative evaluation on these PBR materials are reported in Table 2, the TRELLIS.2 row. Since a major difference is that we assume a known base color and generate the non-diffuse components, when comparing with TRELLIS.2, we also compose full materials from our generation in another version, by combining TRELLIS.2 generated base colors and our generated neural latents. The evaluation of this material variant is in Table 2, the $\text{Our}_{\text{trellis}}$ row; note that the Our_{ref} variant mentioned earlier instead combines our generated neural latents with known base colors. See the supplemental material for additional details and visual comparisons.

5.3 Additional Microscale Enhancement

Besides uplifting non-diffuse BRDFs, materials can be enriched via other procedural enhancements. For instance, microscale effects such as wave-optics-induced glints can introduce fine visual breakups that improve realism for some materials. However, these small details are difficult to represent in textures and synthesize

Table 2. Quantitative metrics for material generation, computed over 16 views for each of the 16 objects in our test set. We compare against TRELLIS.2 [2025]: $\text{Our}_{\text{trellis}}$ represents our uplifted materials with the base color generated by TRELLIS.2; Our_{ref} uses the reference base color.

	CLIP-FID (↓)	CMMD (↓)	LPIPS (↓)	PSNR (↑)
TRELLIS.2	8.642	0.132	0.0510	24.54
$\text{Our}_{\text{trellis}}$	6.527	0.058	0.0444	28.28
Our_{ref}	3.907	0.020	0.0215	32.53

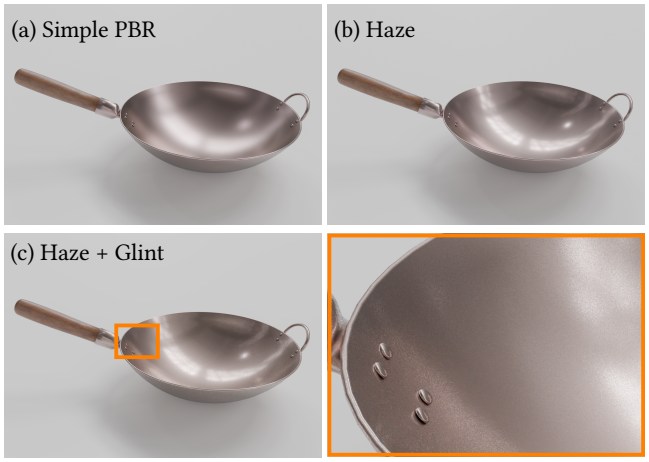

Fig. 13. Procedural enhancement of a metallic material. (a) Simple PBR model (b) Our haze enhancement, producing both sharper highlights and an overall hazy appearance. (c) Haze and glint enhancement (zoom in to compare with "Haze"), with closeup of colored glints on the right.

with diffusion models. Many microscale effects primarily associate with specular reflection, and as we intentionally exclude the diffuse term from our neural material representation, we can model these fine-scale effects ad-hoc, by modulating our non-diffuse BRDF.

As an example, we introduce colorful glints on some objects by deriving a procedural noise model using Simplex noise, informed by wave optics surface scattering as inspired by [Yu et al. 2025, 2024a]. As shown in Fig. 13 and our supplemental video, a glinty appearance can be added to our haze enhancement by modulating our neural BRDF with distinct noise instances $\mathcal{N}^{(i)}$ at different surface points:

$$f_{\text{neu}}^{(i)}(\lambda, \omega_i, \omega_o) = f_{\text{neu}}(\omega_i, \omega_o) \cdot \mathcal{N}^{(i)}(\lambda, \omega_i, \omega_o) \tag{6}$$

This shows that procedural enhancements at different scales can be composed, and fine-scale enhancement can reintroduce detailed structures difficult for generative models to produce.

6 Discussion and Future Work

In this last section, we discuss some directions for future work.

Visual Breakup Texture Selection. To enhance spatial variations in our uplifted materials, we modulated some decorator lobe weights using pre-defined masks (recall Fig. 6). This can lead to appropriate looking imperfections (e.g. scratches, fingerprints) as on hand-crafted materials, but the resulting appearance still depends on how

mask frequency relates to object scale. In practice, meshes may not reflect real-world sizes, and a more robust solution needs to incorporate object size estimation. A preferable alternative is to automate mask creation and placement using a video-to-video material transfer approach similar to [Hadadan et al. 2025].

Resolution Limit. The quality of our generated materials is limited by the video resolution supported in the diffusion stage. While our neural material data can capture the aforementioned high-frequency spatial variations, the generative pipeline may smooth out these fine-scale details, especially for dust variations (see Fig. 12). This constraint, owing to the current resolution limit in video models, will likely be reduced as higher resolution models become available.

Multi-Lobe Model Extension. Our multi-lobe material model was designed to produce the visual effects explored in this work. The BRDF lobes are lightweight and easy to adjust, but they only cover our selected subset of phenomena. Effects such as iridescence or layered absorption would require additional decorator lobes, which are easy to add thanks to our BRDF's modular structure. However, increasing the number of appearance modes will likely amplify limitations related to our neural representation, requiring higher dimensional latent spaces and improved latent space regularization.

Acknowledgments

We thank everyone who worked on the neural appearance project, in particular, Tizian Zeltner, Fabrice Rousselle, and Craig Kolb. The material test object in Fig. 5 and 7 was created by Robin Marin and released under CC (https://creativecommons.org/licenses/by/3.0/).

References

Pontus Andersson, Jim Nilsson, Tomas Akenine-Möller, Magnus Oskarsson, Kalle Åström, and Mark D. Fairchild. 2020. ꟻLIP: A Difference Evaluator for Alternating Images. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 3, 2 (2020), 15:1–15:23. doi:10.1145/3406183

Zap Andersson, Paul Edmondson, Julien Guertault, Adrien Herubel, Peter Kutz, Andréa Machizaud, Jamie Portsmouth, Frédéric Servant, and Jonathan Stone. 2025. OpenPBR Surface: An open shading model for physically based materials. In *Proceedings of the Digital Production Symposium (DigiPro '25)*. Association for Computing Machinery, New York, NY, USA, Article 1, 12 pages. doi:10.1145/3744199.3744632

Asen Atanasov and Vladimir Koylazov. 2016. A practical stochastic algorithm for rendering mirror-like flakes. In *ACM SIGGRAPH 2016 Talks* (Anaheim, California) *(SIGGRAPH '16)*. Association for Computing Machinery, New York, NY, USA, Article 67, 2 pages. doi:10.1145/2897839.2927391

P. Barla, R. Pacanowski, and P. Vangorp. 2018. A Composite BRDF Model for Hazy Gloss. *Computer Graphics Forum* 37, 4 (2018), 55–66. doi:10.1111/cgf.13475

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. 2023a. Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets. *arXiv:2311.15127* (2023).

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. 2023b. Align your Latents: High-Resolution Video Synthesis with Latent Diffusion Models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mark Boss, Zixuan Huang, Aaryaman Vasishta, and Varun Jampani. 2025. SF3D: Stable Fast 3D Mesh Reconstruction with UV-unwrapping and Illumination Disentanglement. *Conference on Computer Vision and Pattern Recognition (CVPR)* (2025).

Brent Burley. 2012. Physically Based Shading at Disney. In *SIGGRAPH Courses: Practical Physically Based Shading in Film and Game Production.*

Duygu Ceylan, Valentin Deschaintre, Thibault Groueix, Rosalie Martin, Chun-Hao Huang, Romain Rouffet, Vladimir Kim, and Gaëtan Lassagne. 2024. MatAtlas: Text-driven Consistent Geometry Texturing and Material Assignment. arXiv:2404.02899 https://arxiv.org/abs/2404.02899

Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. 2023. Text2tex: Text-driven texture synthesis via diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*. 18558–18568.

Yujin Chen, Yinyu Nie, Benjamin Ummenhofer, Reiner Birkl, Michael Paulitsch, and Matthias Nießner. 2025. PBR-SR: Mesh PBR Texture Super Resolution from 2D Image Priors. In *NeurIPS*.

Zhaoxi Chen, Jiaxiang Tang, Yuhao Dong, Ziang Cao, Fangzhou Hong, Yushi Lan, Tengfei Wang, Haozhe Xie, Tong Wu, Shunsuke Saito, Liang Pan, Dahua Lin, and Ziwei Liu. 2024. 3DTopia-XL: High-Quality 3D PBR Asset Generation via Primitive Diffusion. *arXiv preprint arXiv:2409.12957* (2024).

Xavier Chermain, Basile Sauvage, Jean-Michel Dischler, and Carsten Dachsbacher. 2020. Procedural Physically-based BRDF for Real-Time Rendering of Glints. *Computer Graphics Forum (Proceedings of Pacific Graphics)* 39, 7 (2020), 243–253. doi:10.1111/cgf.14141

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. 2023. Objaverse: A Universe of Annotated 3D Objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13142–13153.

Kangle Deng, Timothy Omernick, Alexander Weiss, Deva Ramanan, Jun-Yan Zhu, Tinghui Zhou, and Maneesh Agrawala. 2024. FlashTex: Fast Relightable Mesh Texturing with LightControlNet. In *European Conference on Computer Vision (ECCV)*.

Prafulla Dhariwal and Alexander Quinn Nichol. 2021. Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems*.

Zhao Dong, Bruce Walter, Steve Marschner, and Donald P. Greenberg. 2016. Predicting Appearance from Measured Microgeometry of Metal Surfaces. *ACM Trans. Graph.* 35, 1, Article 9 (Dec. 2016), 13 pages. doi:10.1145/2815618

Luke Emrose, Curtis Black, and Emanuel Schrade. 2022. ASH - A Case For Layered Shading. In *Proceedings of the 2022 Digital Production Symposium* (Vancouver, BC, Canada) *(DigiPro '22)*. Association for Computing Machinery, New York, NY, USA, Article 7, 15 pages. doi:10.1145/3543664.3543675

Andreas Engelhardt, Mark Boss, Vikram Voleti, Chun-Han Yao, Hendrik P. Lensch, and Varun Jampani. 2025. SViM3D: Stable Video Material Diffusion for Single Image 3D Generation. *International Conference on Computer Vision* (2025).

Jiahui Fan, Beibei Wang, Milos Hasan, Jian Yang, and Ling-Qi Yan. 2022. Neural Layered BRDFs. In *ACM SIGGRAPH 2022 Conference Proceedings* (Vancouver, BC, Canada) *(SIGGRAPH '22)*. Association for Computing Machinery, New York, NY, USA, Article 4, 8 pages. doi:10.1145/3528233.3530732

Ye Fang, Zeyi Sun, Tong Wu, Jiaqi Wang, Ziwei Liu, Gordon Wetzstein, and Dahua Lin. 2024. Make-it-Real: Unleashing Large Multimodal Model for Painting 3D Objects with Realistic Materials. arXiv:2404.16829 [cs.CV]

Yifei Feng, Mingxin Yang, Shuhui Yang, Sheng Zhang, Jiaao Yu, Zibo Zhao, Yuhong Liu, Jie Jiang, and Chunchao Guo. 2025. RomanTex: Decoupling 3D-aware Rotary Positional Embedded Multi-Attention Network for Texture Synthesis. *arXiv preprint arXiv:2503.19011* (2025).

Alban Gauthier, Bernhard Kerbl, Jérémy Levallois, Robin Faury, Jean-Marc Thiery, and Tamy Boubekeur. 2024. MatUp: Repurposing Image Upsamplers for SVBRDFs. *Computer Graphics Forum* 43, 4 (2024).

Ole Gulbrandsen. 2014. Artist Friendly Metallic Fresnel. *Journal of Computer Graphics Techniques (JCGT)* 3, 4 (9 December 2014), 64–72. http://jcgt.org/published/0003/04/03/

Jie Guo, Zeru Li, Xueyan He, Beibei Wang, Wenbin Li, Yanwen Guo, and Ling-Qi Yan. 2023. MetaLayer: A Meta-Learned BSDF Model for Layered Materials. 42, 6, Article 222 (Dec. 2023), 15 pages. doi:10.1145/3618365

Yu Guo, Miloš Hašan, and Shuang Zhao. 2018. Position-free monte carlo simulation for arbitrary layered BSDFs. *ACM Trans. Graph.* 37, 6, Article 279 (Dec. 2018), 14 pages. doi:10.1145/3272127.3275053

Saeed Hadadan, Benedikt Bitterli, Tizian Zeltner, Jan Novák, Fabrice Rousselle, Jacob Munkberg, Jon Hasselgren, Bartlomiej Wronski, and Matthias Zwicker. 2025. Generative detail enhancement for physically based materials. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25)*. Association for Computing Machinery, New York, NY, USA, Article 164, 11 pages. doi:10.1145/3721238.3730751

Jon Hasselgren, Zheng Zeng, Milos Hasan, and Jacob Munkberg. 2026. VideoMatGen: PBR Materials through Joint Generative Modeling. *arXiv preprint arXiv:2603.16566* (2026).

Zebin He, Mingxin Yang, Shuhui Yang, Yixuan Tang, Tao Wang, Kaihao Zhang, Guanying Chen, Yuhong Liu, Jie Jiang, Chunchao Guo, and Wenhan Luo. 2025. MaterialMVP: Illumination-Invariant Material Generation via Multi-view PBR Diffusion. *arXiv preprint arXiv:2503.10289* (2025).

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. 2023. CogVideo: Large-scale Pretraining for Text-to-Video Generation via Transformers. In *The Eleventh International Conference on Learning Representations*.

Wenzel Jakob, Eugene d'Eon, Otto Jakob, and Steve Marschner. 2014. A comprehensive framework for rendering layered materials. *ACM Trans. Graph.* 33, 4, Article 118 (July 2014), 14 pages. doi:10.1145/2601097.2601139

Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. 2024. Rethinking FID: Towards a Better Evaluation Metric for Image Generation. arXiv:2401.09603

Brian Karis. 2013. Real shading in Unreal Engine 4. *ACM SIGGRAPH Course on Physically Based Shading Theory and Practice* 4, 3 (2013), 1.

Khronos. 2025. *glTF 2.0 Specification.* https://registry.khronos.org/glTF/specs/2.0/glTF-2.0.html#appendix-b-brdf-implementation

Christopher Kulla and Alejandro Conty. 2017. Revisiting Physically Based Shading at Imageworks. In *SIGGRAPH 2017 Course: Physically Based Shading in Theory and Practice.*

Alexandr Kuznetsov, Miloš Hašan, Zexiang Xu, Ling-Qi Yan, Bruce Walter, Nima Khademi Kalantari, Steve Marschner, and Ravi Ramamoorthi. 2019. Learning generative models for rendering specular microgeometry. *ACM Trans. Graph.* 38, 6, Article 225 (Nov. 2019), 14 pages. doi:10.1145/3355089.3356525

Alexandr Kuznetsov, Krishna Mullia, Zexiang Xu, Miloš Hašan, and Ravi Ramamoorthi. 2021. NeuMIP: Multi-Resolution Neural Materials. *Transactions on Graphics (Proceedings of SIGGRAPH)* 40, 4, Article 175 (July 2021).

Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. 2023. The Role of ImageNet Classes in Fréchet Inception Distance. In *Proc. ICLR.*

Yehonathan Litman, Or Patashnik, Kangle Deng, Aviral Agrawal, Rushikesh Zawar, Fernando De la Torre, and Shubham Tulsiani. 2025. MaterialFusion: Enhancing Inverse Rendering with Material Diffusion Priors. In *3DV.*

Simon Lucas, Mickaël Ribardière, Romain Pacanowski, and Pascal Barla. 2024. A Fully-correlated Anisotropic Micrograin BSDF Model. *ACM Trans. Graph.* 43, 4, Article 111 (July 2024), 14 pages. doi:10.1145/3658224

Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. 2003. A Data-Driven Reflectance Model. *ACM Transactions on Graphics* 22, 3 (July 2003), 759–769.

Jacob Munkberg, Zian Wang, Ruofan Liang, Tianchang Shen, and Jon Hasselgren. 2025. VideoMat: Extracting PBR Materials from Video Diffusion Models. In *Eurographics Symposium on Rendering - CGF Track.*

Addy Ngan, Frédo Durand, and Wojciech Matusik. 2005. Experimental analysis of BRDF models. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques* (Konstanz, Germany) *(EGSR '05).* Eurographics Association, Goslar, DEU, 117–126.

NVIDIA. 2025. Cosmos World Foundation Model Platform for Physical AI. *arXiv preprint arXiv:2501.03575* (2025).

Michail N. Polyanskiy. 2025. *Refractive index database.* https://refractiveindex.info

Qwen. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115* (2024).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. arXiv:1910.10683 [cs.LG] https://arxiv.org/abs/1910.10683

Nithin Raghavan, Krishna Mullia, Alexander Trevithick, Fujun Luan, Miloš Hašan, and Ravi Ramamoorthi. 2025. Generative Neural Materials. 43, Article 162 (2025), 11 pages. doi:10.1145/3721238.3730746

Boris Raymond, Gaël Guennebaud, and Pascal Barla. 2016. Multi-scale rendering of scratched materials using a structured SV-BRDF model. *ACM Trans. Graph.* 35, 4, Article 57 (July 2016), 11 pages. doi:10.1145/2897824.2925945

Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. 2023. TEXTure: Text-guided texturing of 3d shapes. In *ACM SIGGRAPH 2023 conference proceedings.* 1–11.

ByteDance Seed. 2025. *Seed3D 1.0: From Images to High-Fidelity Simulation-Ready 3D Assets.* https://seed3d.github.io/Seed3D/report.pdf

Mingqi Shao, Feng Xiong, Zhaoxu Sun, and Mu Xu. 2025. MVPainter: Accurate and Detailed 3D Texture Generation via Multi-View Diffusion with Geometric Control. *arXiv preprint arXiv:2505.12635* (2025). https://arxiv.org/abs/2505.12635

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. 2023. MVDream: Multi-view Diffusion for 3D Generation. *arXiv:2308.16512* (2023).

Doug Smythe and Jonathan Stone. 2017. MaterialX: An Open Standard for Network-Based CG Object Looks. SIGGRAPH 2017 Birds of a Feather.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep Unsupervised Learning Using Nonequilibrium Thermodynamics. In *International Conference on Machine Learning.*

Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitrii Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. 2024. SV3D: Novel Multi-view Synthesis and 3D Generation from a Single Image using Latent Video Diffusion. In *European Conference on Computer Vision (ECCV).*

Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques* (Grenoble, France) *(EGSR'07).* Eurographics Association, Goslar, DEU, 195–206.

Team Wan. 2025. Wan: Open and Advanced Large-Scale Video Generative Models. *arXiv preprint arXiv:2503.20314* (2025).

Beibei Wang, Wenhua Jin, Miloš Hašan, and Ling-Qi Yan. 2022. SpongeCake: A Layered Microflake Surface Appearance Model. *ACM Trans. Graph.* 42, 1, Article 8 (Sept. 2022), 16 pages. doi:10.1145/3546940

Andrea Weidlich and Alexander Wilkie. 2007. Arbitrarily layered micro-facet surfaces. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia* (Perth, Australia) *(GRAPHITE '07).* Association for Computing Machinery, New York, NY, USA, 171–178. doi:10.1145/1321261.1321292

Sebastian Werner, Zdravko Velinov, Wenzel Jakob, and Matthias B. Hullin. 2017. Scratch iridescence: wave-optical rendering of diffractive surface structure. *ACM Trans. Graph.* 36, 6, Article 207 (Nov. 2017), 14 pages. doi:10.1145/3130800.3130840

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. 2025. Native and Compact Structured Latents for 3D Generation. *Tech report* (2025).

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. 2024. Structured 3D Latents for Scalable and Versatile 3D Generation. *arXiv preprint arXiv:2412.01506* (2024).

You-Xin Xing, Hao-Wen Tan, Yan-Ning Xu, and Lu Wang. 2024. A Tiny Example Based Procedural Model for Real-Time Glinty Appearance Rendering. *J. Comput. Sci. Technol.* 39, 4 (Sept. 2024), 771–784. doi:10.1007/s11390-024-4123-3

Haibo Yang, Yang Chen, Yingwei Pan, Ting Yao, Zhineng Chen, Chong-Wah Ngo, and Tao Mei. 2024a. Hi3D: Pursuing High-Resolution Image-to-3D Generation with Video Diffusion Models. In *ACM MM.*

Jiayu Yang, Taizhang Shang, Weixuan Sun, Xibin Song, Ziang Chen, Senbo Wang, Shenzhou Chen, Weizhe Liu, Hongdong Li, and Pan Ji. 2025. Pandora3D: A Comprehensive Framework for High-Quality 3D Shape and Texture Generation. *arXiv preprint arXiv:2502.14247* (2025).

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, Yuxuan Zhang, Weihan Wang, Yean Cheng, Ting Liu, Bin Xu, Yuxiao Dong, and Jie Tang. 2024b. CogVideoX: Text-to-Video Diffusion Models with An Expert Transformer.

Yu-Ying Yeh, Jia-Bin Huang, Changil Kim, Lei Xiao, Thu Nguyen-Phuoc, Numair Khan, Cheng Zhang, Manmohan Chandraker, Carl S Marshall, Zhao Dong, et al. 2024. TextureDreamer: Image-guided Texture Synthesis through Geometry-aware Diffusion. *arXiv preprint arXiv:2401.09416* (2024).

Xin Yu, Ze Yuan, Yuan-Chen Guo, Ying-Tian Liu, Jianhui Liu, Yangguang Li, Yan-Pei Cao, Ding Liang, and Xiaojuan Qi. 2024b. TEXGen: a Generative Diffusion Model for Mesh Textures. *ACM Trans. Graph.* 43, 6, Article 213 (2024).

Yunchen Yu, Bruce Walter, Steve Marschner, and Andrea Weidlich. 2025. Realistic Cloth Rendering with a Ray-Wave Hybrid Shading Model. *ACM Transactions on Graphics (TOG)* 44, 6 (2025), 1–17.

Yunchen Yu, Andrea Weidlich, Bruce Walter, Eugene d'Eon, and Steve Marschner. 2024a. Appearance Modeling of Iridescent Feathers with Diverse Nanostructures. *ACM Transactions on Graphics (TOG)* 43, 6 (2024), 1–18.

Greg Zaal and et al. 2024. *Poly Haven - The Public 3D Asset Library.* https://polyhaven.com

Tizian Zeltner, Brent Burley, and Matt Jen-Yuan Chiang. 2022. Practical Multiple-Scattering Sheen Using Linearly Transformed Cosines. In *ACM SIGGRAPH 2022 Talks* (Vancouver, BC, Canada) *(SIGGRAPH '22).* Association for Computing Machinery, New York, NY, USA, Article 7, 2 pages. doi:10.1145/3532836.3536240

Tizian Zeltner and Wenzel Jakob. 2018. The layer laboratory: a calculus for additive and subtractive composition of anisotropic surface reflectance. *ACM Trans. Graph.* 37, 4, Article 74 (July 2018), 14 pages. doi:10.1145/3197517.3201321

Tizian Zeltner, Fabrice Rousselle, Andrea Weidlich, Petrik Clarberg, Jan Novák, Benedikt Bitterli, Alex Evans, Tomáš Davidovič, Simon Kallweit, and Aaron Lefohn. 2024. Real-time Neural Appearance Models. *ACM Trans. Graph.* 43, 3, Article 33 (June 2024).

Xianfang Zeng, Xin Chen, Zhongqi Qi, Wen Liu, Zibo Zhao, Zhibin Wang, Bin Fu, Yong Liu, and Gang Yu. 2024. Paint3d: Paint anything 3d with lighting-less texture diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 4252–4262.

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. 2024c. CLAY: A Controllable Large-scale Generative Model for Creating High-quality 3D Assets. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–20.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR.*

Shangzhan Zhang, Sida Peng, Tao Xu, Yuanbo Yang, Tianrun Chen, Nan Xue, Yujun Shen, Hujun Bao, Ruizhen Hu, and Xiaowei Zhou. 2024b. MaPa: Text-driven Photorealistic Material Painting for 3D Shapes. In *ACM SIGGRAPH 2024 Conference Papers.* Article 4, 12 pages.

Yuqing Zhang, Yuan Liu, Zhiyu Xie, Lei Yang, Zhongyuan Liu, Mengzhou Yang, Runze Zhang, Qilong Kou, Cheng Lin, Wenping Wang, and Xiaogang Jin. 2024a. DreamMat: High-quality PBR Material Generation with Geometry- and Light-aware Diffusion Models. *ACM Trans. Graph.* 43, 4, Article 39 (2024).

Chenliang Zhou, Zheyuan Hu, Alejandro Sztrajman, Yancheng Cai, Yaru Liu, and Cengiz Oztireli. 2026. M^3ashy: Multi-Modal Material Synthesis via Hyperdiffusion. In *Proceedings of the 40th AAAI Conference on Artificial Intelligence* (Singapore).