

PhotoMat: A Material Generator Learned from Single Flash Photos

Xilong Zhou

Texas A&M University
Adobe Research
College Station, USA

Paul Guerrero

Adobe Research
London, UK

Miloš Hašan

Adobe Research
San Jose, USA

Valentin Deschaintre

Adobe Research
London, UK

Yannick Hold-Geoffroy

Adobe Research
San Jose, USA

Kalyan Sunkavalli

Adobe Research
San Jose, USA

Nima Khademi Kalantari

Texas A&M University
College Station, USA

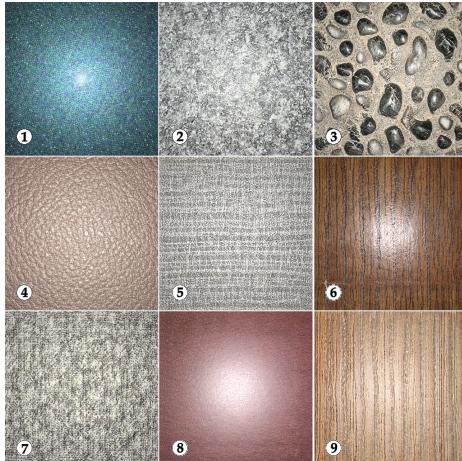


Figure 1: We present a material generative model trained exclusively on real cell-phone flash photographs. Our model produces plausible photorealistic materials (left), which can readily be applied to 3D scenes (right). All 18 materials (except glass) are from our generator. Geometry by Flavio Della Tommasa (CC-BY).

ABSTRACT

Authoring high-quality digital materials is key to realism in 3D rendering. Previous generative models for materials have been trained exclusively on synthetic data; such data is limited in availability and has a visual gap to real materials. We circumvent this limitation by proposing PhotoMat: the first material generator trained exclusively on *real photos* of material samples captured using a cell phone camera with flash. Supervision on individual material maps is not available in this setting. Instead, we train a generator for a neural material representation that is rendered with a learned relighting module to create arbitrarily lit RGB images; these are

compared against real photos using a discriminator. We train PhotoMat with a new dataset of 12,000 material photos captured with handheld phone cameras under flash lighting. We demonstrate that our generated materials have better visual quality than previous material generators trained on synthetic data. Moreover, we can fit analytical material models to closely match these generated neural materials, thus allowing for further editing and use in 3D rendering.

CCS CONCEPTS

- Computing methodologies → Reflectance modeling.

KEYWORDS

Materials, SVBRDF, generative models, GAN

ACM Reference Format:

Xilong Zhou, Miloš Hašan, Valentin Deschaintre, Paul Guerrero, Yannick Hold-Geoffroy, Kalyan Sunkavalli, and Nima Khademi Kalantari. 2023. PhotoMat: A Material Generator Learned from Single Flash Photos. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings (SIGGRAPH '23 Conference Proceedings)*, August 6–10, 2023, Los Angeles, CA, USA, <https://doi.org/10.1145/3588432.3591535>.

Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings (SIGGRAPH '23 Conference Proceedings), August 6–10, 2023, Los Angeles, CA, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3588432.3591535>

1 INTRODUCTION

Materials with accurate spatially-varying reflectance properties are among the most important factors determining the realism of rendered scenes. These materials are often represented as spatially varying parameters of analytic BRDFs, commonly termed SVBRDFs. Unfortunately, creating high-quality materials is challenging. Authoring materials from scratch usually requires significant time and experience with specialized tools like Substance Designer [Adobe 2023]. An alternative is to capture material reflectance using specialized hardware such as spherical gantries or domes. This also has found limited use due to the cost of the acquisition process.

In recent years, researchers have proposed data-driven methods to either capture or generate material reflectance without the need for specialized skills or hardware. For lightweight capture, many convolutional image-to-image translation models to map one or few RGB photos of materials to output material maps have been proposed [Deschaintre et al. 2018, 2020; Gao et al. 2019; Guo et al. 2021; Martin et al. 2022]. Other works have proposed methods for unconditional or conditional generation of materials [Guo et al. 2020; Zhou et al. 2022]. All these methods (both capture and generation) require explicit supervision on individual material maps, and as a result, are trained on synthetic material datasets. Because of the cost of authoring digital materials, these datasets are limited in number and diversity; for example, most of these methods are trained on the Deschaintre et al. [2018] material dataset that has only 1850 materials derived from 155 procedural materials in it. Moreover, even though they are carefully designed, these synthetic materials still have a substantial visual gap to real-world materials. As a consequence, the methods synthesize results that inherit the synthetic data distribution, giving them a certain digital look.

To address this challenge, we ask the question: is it possible to train a realistic material generator without using synthetic maps? This is a chicken-and-egg problem: to train a material generator we need a large dataset of real reflectance maps, while to produce such a dataset, we need methods to acquire realistic materials from one or few images. Our key idea to solve this problem is to perform both generation and acquisition within a *single framework*. In contrast to existing work, we present PhotoMat, a material generation model trained on a large dataset of easy to capture flash photographs, with known light source (flash) location. Our key contribution is a specifically designed system that ensures that PhotoMat learns real-world reflectance properties without directly observing ground truth material maps.

PhotoMat uses a high-dimensional “neural material” map generator (based on StyleGAN2 [Karras et al. 2019]) that produces relightable per-pixel features that encode the appearance of that point. These neural features are then fed into a per-pixel neural relighting multi-layer perceptron (MLP) to render images under a conditional light source (flash) location. A discriminator, also conditioned on the flash light location, takes these renderings as well as cropped samples from the real dataset as inputs. Through training, we learn both the generator that can produce a powerful

implicit neural representation of reflectance properties, as well as a relighting module that can simulate the rendering process given a chosen light location. Together, these two modules produce rendered images which follow the distribution of the real flash images.

We use our relightable neural material representation to explicitly estimate analytical material maps which, when rendered, closely match the relightable material. To achieve this, we train a material map estimator that takes the GAN-generated neural material as input and produces per-pixel material parameters that are then fed into an analytic differentiable renderer. The loss between the analytically and neurally rendered materials is back-propagated to train the material map estimator. We demonstrate that PhotoMat can learn a generic material generator for standard microfacet-based analytical material models, as well as specialized generators for specific materials (e.g. coated BRDFs for car paints).

Finally, to gather a large scale dataset of flash images with known light location, we propose a simple, but effective, data collection mechanism that eliminates the need for camera calibration. Each image is collected using a handheld phone camera with flash light on under weak ambient environment so that flash light dominates the material surface. We then apply a simple light detector to detect flash light position of each image. We use this casual capture process to collect and process a large dataset of 12,000 real photos. In summary, this paper makes the following contributions:

- PhotoMat, the first material generative model trained exclusively on real photos without relying on a specific analytic BRDF model or an existing SVBRDF dataset.
- A neural material representation that can be decoded into analytic SVBRDF parameters with no material map supervision.
- A data collection strategy (using a handheld phone with flash) can be easily scaled to large material datasets. We collect 12,000 material flash photos that will be publicly released.
- Material generation of high-quality SVBRDFs, across several material categories, that can be used in practical 3D rendering.

2 RELATED WORK

As an essential part of the rendering pipeline, material creation and editing have received significant attention from both researchers and industry practitioners. In the computer graphics industry, material creation is typically done using procedural node graph editing software [Adobe 2023], but material acquisition and generative models are growing in importance.

Material acquisition. Traditional acquisition approaches rely on extensively sampling both light and view directions using a gonioreflectometer [Guarnera et al. 2016; Matusik et al. 2003]. Recent methods have relied on data-driven approaches to retrieve material properties from a single image or a few images, with either flash or natural lighting. Methods using different architectures have been proposed [Deschaintre et al. 2018, 2019, 2020; Gao et al. 2019; Guo et al. 2021; Li et al. 2017, 2018; Martin et al. 2022; Zhou and Kalantari 2021, 2022]. All of these approaches rely on supervised training with synthetic data. Zhou et al. [2021] use an adversarial loss for training and complement the synthetic data with a small real dataset, and Henzler et al. [2021] rely on a small dataset of captured flash images for pre-training, but require a fine-tuning step to reproduce the input image appearance. As opposed to our

goal, these methods target material acquisition from a flash picture; neither of these approaches can generate new materials. In fact, many of the methods above ultimately rely on the *same* datasets, Substance Source and Substance Share, and the derived dataset by Deschaintre et al. [2018], highlighting the need for material research to move beyond limited synthetic data.

Material generation. MaterialGAN [Guo et al. 2020] trains an unconditional generative model for SVBRDFs. Their main goal is to optimize in the generator latent space to match the appearance of a real material given a few target pictures, thus the quality of the generated materials is only a secondary goal, since they are not used directly. TileGen [Zhou et al. 2022] extends MaterialGAN, focusing on a per class approach. This approach improves the architecture to enable tileable SVBRDF generation that can be conditioned on the input patterns. Hu et al. [2022] use a network similar to TileGen [Zhou et al. 2022] as a prior for material appearance transfer.

An alternative material representation heavily used in industry is procedural node graphs. Their creation typically relies on professional software and requires specialized skills. To simplify their creation, Guerrero et al. [2022] proposed an unconditional procedural graph generator, leveraging transformers to generate tokenized nodes, edges and parameters of material graphs. Recently, EG3D [Chan et al. 2021] proposed a GAN for 3D face generation supervised only with 2D images; this problem is related to ours, since it generates an asset type that is not directly observed in 2D photos but required in 3D scene authoring. We base our generative models on the StyleGAN 2 architecture [Karras et al. 2019], which is widely used for image synthesis in many domains. We modify the architecture to produce per-pixel neural feature maps, which are turned into final RGB values using a neural relighting MLP module.

Neural materials. PhotoMat produces an intermediate neural material representation that can be rendered into arbitrarily lit photographs through an additional MLP-based relighting module. This is akin to neural materials that combine feature textures with an MLP decoder and have been directly used for rendering [Fan et al. 2022; Kuznetsov et al. 2021; Rainer et al. 2020]. These works however require a large amount of data sampling to fit a specific material, and do not generate new materials.

3 THE PHOTOMAT METHOD

Our goal is to train a generative model producing material maps (such as albedo, roughness, normals, etc.) using a real dataset of flash material photos. The key problem is that the material maps are hidden variables that are not directly observed in the real RGB photos, so previous techniques supervised with ground-truth maps [Guo et al. 2020; Zhou et al. 2022] cannot be applied here.

3.1 Key idea

Our idea is to split the problem into two parts. First, we train a conditional *relightable* GAN for material images in the RGB domain: conditional on a desired flash highlight location, the generator is able to produce a rendered material image with the highlight in the requested place. This is done by generating an implicit neural material representation and using the conditional information with a neural relighting module in the last step to obtain a relit RGB

image. Second, this relightability property lets us generate the same material under many known lighting conditions. For common BRDF models this is sufficient to decode explicit material parameters per texel that fit the appearance of the texel under the specified lighting conditions. We train another network to do this decoding from an implicit neural material to an explicit analytical material.

Training this method requires a dataset with a variety of flash highlight locations. Fortunately, we do not have to explicitly capture the materials with separate varying light sources. We can use simple cropping of the original photos to obtain images with a variety of highlight locations. The uncropped square photos have flash highlights approximately in the center (we detect the exact location); the light (and camera) direction can therefore be computed for each pixel of each crop (Fig. 2).

Note that the photos, taken with a cell phone camera with flash, can be treated as having a *collocated* camera and point light, where the slight distance between the physical flash and camera is negligible. Therefore, whenever we refer to the *light location*, we imply this to be the same as the *camera location*. Similarly, *highlight location* means the point on the material sample directly below the light (camera). We can use all of these terms interchangeably, as they are trivially convertible to each other. Thus, by cropping photos in which the highlight is centered, we can achieve cropped material samples lit with a variety of highlight locations (and therefore a variety of light/camera locations with respect to the image center).

In summary, the PhotoMat solution consists of two steps: first, we propose a relightable generator to generate material images under conditional light source (highlight) locations (shown in the left part of the Fig. 3), trained on real images only. Second, we use a BRDF parameter estimator to decode analytical material reflectance parameters from the implicit neural material representation produced by the generator (shown in the right part of the Fig. 3). In the rest of this section, we provide more details.

3.2 Relightable generator

Generative adversarial networks (GANs) [Goodfellow et al. 2014] have been successful generative models in many areas (images, video, audio). We build on recent work on material generative models, TileGen [Zhou et al. 2022], which is built upon a tileable version of StyleGAN2 [Karras et al. 2019] to produce plausible tileable and controllable materials conditional on mask patterns. However, like all previous material generative models, TileGen is trained on synthetic data. Our architecture is also partially inspired by the EG3D

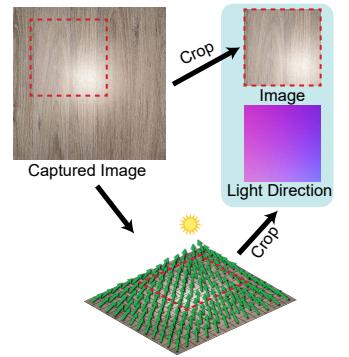


Figure 2: We randomly crop photos to $\frac{1}{2}$ size to create crops with random highlight locations; each texel of each crop has a known light direction.

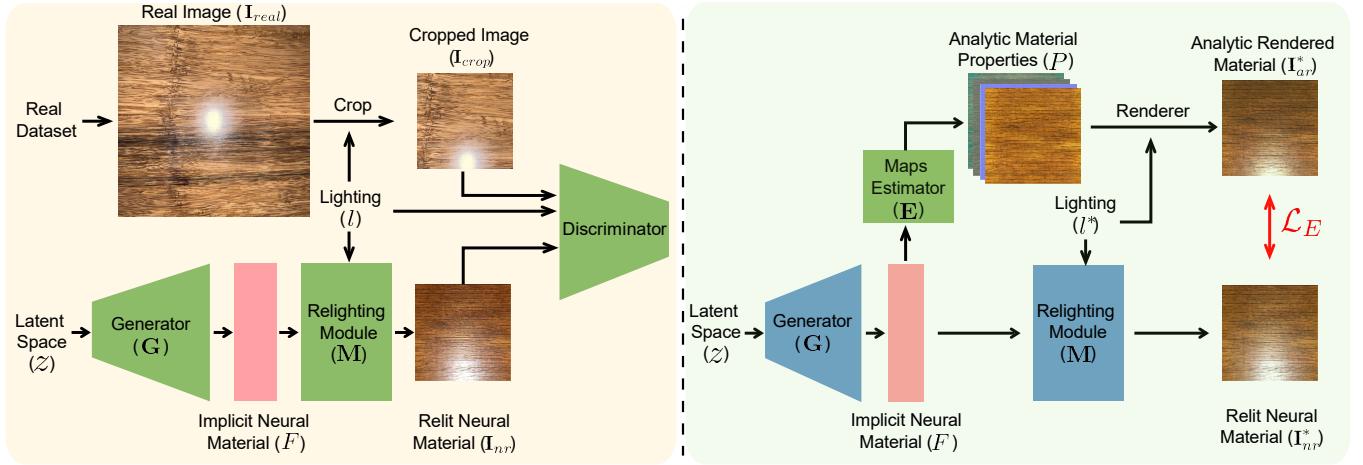


Figure 3: A visual illustration of our method. Left: In our relightable generator, a latent code z feeds into the feature generator G to produce an implicit neural material representation F , where every texel holds a 32-dimensional feature vector; optionally, F can be tileable. Next, F is fed to the conditional relighting module M to produce a relit neural material I_{nr} under highlight location l . During training, the discriminator takes either the relit neural material I_{nr} or a cropped real image I_{crop} , and attempts to distinguish them. Right: In the next step, we freeze the pretrained G and M , and train a convolutional decoder E to turn the neural material representation F into per-texel analytic reflectance properties P . The decoder is trained to minimize the loss \mathcal{L}_E between re-rendered analytic materials I_{ar}^* and relit neural materials I_{nr}^* . Once all networks are trained, we can generate new materials by randomly sampling z , producing implicit neural material representation $F = G(z)$ and decoding them into analytic material properties $P = E(F)$.

human face generator conditional on camera pose [Chan et al. 2021], where a StyleGAN-based generator produces a 3D feature field (triplane) of a human face, which is then passed through a neural renderer producing a 2D image of a face from a specific view.

The architecture of our relightable generator consists of a feature generator G and a conditional neural relighting module M . Given a latent code z , the feature generator $G(z)$ produces an implicit material representation F with channel number $C = 32$ and the same resolution as the final generated material. Next, F is fed into the relighting MLP module $M(F, l)$ conditioned on the highlight location l to produce a “neurally” relit RGB image I_{nr} :

$$I_{nr}(z, l) = M(G(z), l) = M(F, l). \quad (1)$$

Note that F can be seen as a neural representation for material reflectance. Similarly, M can be seen as a neural analogy to a classical local illumination renderer, where each texel of the material surface is independently lit based on the local light direction. We therefore convert the highlight location l into per-texel light direction ω and use an MLP to implement M with local light direction ω and local feature vector F for each individual texel as input. We show in Fig. 5 several examples of relit images produced by our generator.

To train this combination of neural material generator G and neural relighter M we use an adversarial loss with real flash images as training data. For each training step, we fetch a real example I_{real} from the real dataset, and crop a random region out of the original real materials to obtain the “relit real material” I_{crop} . For example, for training a 256^2 generator, we crop a random 256^2 region out of a real photo resized to 512^2 ; details about resolution are discussed in Sec. 4. We convert the highlight location in I_{real}

to the corresponding highlight location in I_{crop} , based on the crop boundaries, to account for this cropping operation.

To train the discriminator in the image domain with our generated features, we train the relighting module to simulate a collocated light and view rendering process given generated features and a light direction. The discriminator is conditioned on l and takes I_{crop} and the neural rendering I_{nr} for training. Once trained, our system can produce implicit neural material features that can be relit, using our relighting module, to produce renderings which follow the distribution of real flash photographs. We now describe how we use the generated features of our trained network, which contain all information to relight the represented material, as the input to a map estimator network.

In PhotoMat, G and M are trained in a way where the inherent ambiguities between lighting and SVBRDF can be avoided. The G and M models are trained with a GAN loss and varying lighting configurations, preventing baking highlights in neural materials F . Indeed, if the specular highlights were baked F , the discriminator would easily detect it due to the mismatch between the light condition and the observed highlight location.

3.3 Material map estimator

Once our neural material generator G and relighter M are trained, we aim to complete an end-to-end system to estimate analytical reflectance parameters P from the generated implicit representation. We train a material estimator (decoder) E , which takes the generated neural material F as input and outputs the corresponding per-texel reflectance parameters P . The goal is to produce P that, when rendered under different lighting conditions using an



Figure 4: Example flash photos from our in-the-wild dataset. We roughly center the highlight and avoid strong ambient lighting. Our pipeline is robust against small imperfections and does not require image calibration.

analytical differentiable renderer \mathcal{R} to produce images $\mathbf{I}_{ar}(P)$, approximates the image produced by the neural relighting module, $\mathbf{I}_{nr}(F)$. The loss \mathcal{L}_E between analytically rendered materials and neurally rendered materials is backpropagated to train \mathbf{E} . Each training iteration samples a random latent code z and a random lighting l , and our loss function is described as below:

$$\mathcal{L}_E = \mathcal{L}(\mathbf{I}_{ar}, \mathbf{I}_{nr}(z, l)), \quad (2)$$

where the analytically rendered image \mathbf{I}_{ar} is defined as:

$$\mathbf{I}_{ar} = \mathcal{R}(p, l) = \mathcal{R}(\mathbf{E}(\mathbf{G}(z)), l). \quad (3)$$

Here \mathcal{R} represents the differentiable analytic renderer for the appropriate material model (e.g. a standard GGX-based microfacet model [Walter et al. 2007], or a more specialized model). Note that l represents a randomly sampled lighting on a plane used for both the analytic and neural path, forcing consistency between \mathbf{I}_{ar} and \mathbf{I}_{nr} for a given highlight location. For the loss function \mathcal{L} , we use the distance between Gram matrices of VGG layers [Gatys et al. 2015] combined with L1 loss, similar to TileGen [Zhou et al. 2022]. The weight of gram matrix term is set as 1.0 and L1 term is 0.1. Note that \mathbf{E} takes the lighting-independent neural materials F as input and is forced to produce valid analytic BRDF parameter maps across different lighting configurations, avoiding baked lighting artifacts.

Light falloff. An idealized omni-directional point light would have a falloff given by the inverse squared distance. However, the camera flash is usually not omni-directional and may have an additional falloff term, compounded with vignetting effects of the lens. Moreover, camera hardware variety, processing and tonemapping, and the ambient lighting in photos taken in a naturally lit environment, further complicate the computation of the exact falloff. In fact, the latter two effects can differ between images taken with the same camera, so no single falloff term will be accurate. Trying to calibrate the data acquisition more precisely would however conflict with our goal of in-the-wild capture.

The imperfect falloff results in slight artifacts in the generated maps, where the estimator attempts to correct for the falloff by bending normals, and adjusting the intensity of other maps close

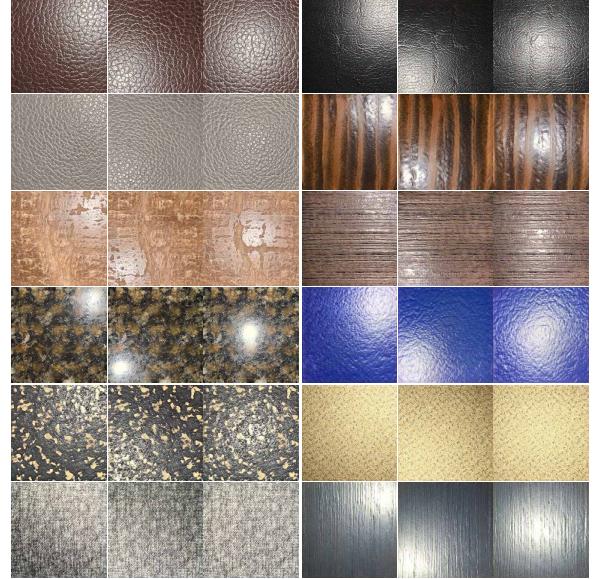


Figure 5: Example outputs of our neural material generator: each triplet is a single neural material, F , relit under 3 different light locations. This model generates 256^2 images and is trained on our small size (2.5k images) dataset.

to edges. We address this by finetuning the estimator with a different loss. In each iteration, we randomly shift the feature grid F horizontally and vertically (with wraparound) before passing it to the maps estimator. We use a Gram matrix loss between the shifted and unshifted neural renders, which compares their overall style without pixel alignment. This removes falloff effects from the results, because the shifting withholds any information about which texels are close to the edges from the map estimator. This however comes at the cost of a small reduction in normal maps contrast.

Summary. The material estimator \mathbf{E} is designed to decode explicit reflectance properties from the learned implicit neural material representation trained on real materials. Our end-to-end system is very fast, as it combines forward inference of a generator with a U-net estimator. For 256^2 , 512^2 and 1024^2 models, sampling rates are 900, 200 and 60 samples/minute respectively, tested on a single RTX2080 GPU. See Fig. 10 for the material maps estimated for the sampled materials from Fig. 5.

4 IMPLEMENTATION

In this section, we provide further details on the neural architectures, real dataset collection and processing, and training details.

Our neural material generator \mathbf{G} is built upon StyleGAN2 [Karras et al. 2019] modified to output a feature tensor F with the size of $H \times W \times 32$. We build a tileable version of \mathbf{G} , using the strategy proposed in TileGen [Zhou et al. 2022] to preserve tileable output. We do, however, observe that tileable generators are harder to train and have worse Fréchet inception distance (FID) [Heusel et al. 2017] and lower apparent visual quality than non-tileable generators; more research on high-quality tileable generation is needed.

Our relighting module \mathbf{M} is an eight-layer MLP applied per-texel, the input of which is the concatenated feature vector from the corresponding texel of F and a per-pixel light direction (a 3-dimensional unit vector) computed from the conditional light position l . Each inner layer of the MLP has 64 channels and we use LeakyReLU except at the last layer. The discriminator follows StyleGAN2 except that ours also takes per-pixel light directions concatenated to the RGB image input, for a total of 6-channel input.

When fetching real images from the dataset, we crop I_{crop} with resolution of $H \times W \times 3$ from a real image I_{real} with resolution of $2H \times 2W \times 3$. The crop coordinate is computed from conditional point light l under the assumption that the camera is always collocated with point light source, and always at fixed height relative to the surface; even if not always true in practice, we can assume this without loss of generality. Only the ratio of camera distance and sample size matters in our results, and remains constant. We assume the camera distance to the sample equals the size of the visible sample area; we observe this is close to the true camera field-of-view of typical cell phones.

For the maps estimator \mathbf{E} , we use a UNet [Ronneberger et al. 2015] with skip connections, which includes five downsampling and upsampling layers. This network takes F as input and outputs certain channels n of maps representing the material parameters ($32 \rightarrow 32 \rightarrow 64 \rightarrow 128 \rightarrow 256 \rightarrow 256 \rightarrow 256 \rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow n$). Typically the material parameters include diffuse albedo, specular albedo, roughness, and normal or height (we have tested both; in the latter case the UNet produces a height map, which is then explicitly converted to normal map for rendering). Our observations show that predicting height leads to better results; it could also be used for displacement mapping. Thus $n = 8$ for this model. We further demonstrate that our model can fit custom BRDF models. For metallic paint, we use a different analytic model including two specular lobes each with its own albedo and roughness and one color albedo. In this model n is set as 11.

Training. We follow the training strategy of StyleGAN2 to train \mathbf{G} and \mathbf{M} with a learning rate of 0.0025 with a batch size of 64. We train generation models at different image resolutions. We pick the \mathbf{G} and \mathbf{M} with best FID during training. For 256^2 , 512^2 and 1024^2 model, we pick checkpoint at iteration 48k, 12k, and 11k respectively. We then freeze the pretrained \mathbf{G} and \mathbf{M} , and use Adam [Kingma and Ba 2014] with a learning rate (lr) of 1×10^{-4} and batch size of 4 to train \mathbf{E} for 60k iterations. Finally, we finetune the \mathbf{E} by shifting F during training for another 60k iterations with $lr = 5 \times 10^{-5}$. During this fine-tuning process, we use a Gram matrix loss and a random shifting strategy, further encouraging material maps to be invariant to spatial location. We train \mathbf{G} and \mathbf{M} on 8 V100 GPUs and train \mathbf{E} on a single GPU.

Dataset collection. In this section, we describe the rules we used to collect and prepare our real dataset. First, we attempt to center the flash light as much as possible ; we further run a simple highlight detector to help estimate the position of flash light in case of misalignment. Second, we minimize ambient lighting to ensure that the flash light dominates. Third, we strive to collect a diverse dataset covering multiple material categories that are easily available in everyday environments.

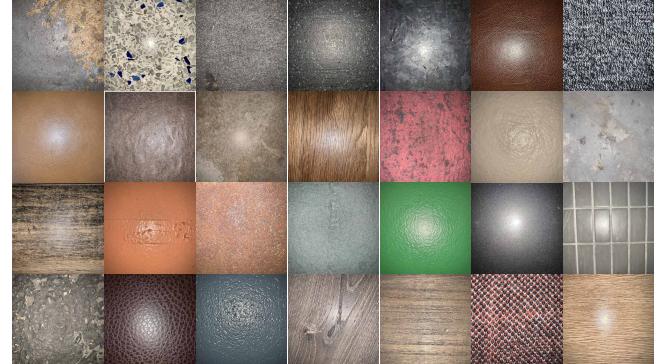


Figure 6: Example flash photos from our in-the-wild dataset. We roughly center the highlight and avoid strong ambient lighting. Our pipeline is robust against small imperfections and does not require image calibration.

Overall we present three datasets: a small *Glossy* dataset containing 2.5k images of materials (limited to materials with low roughness and obvious specular reflection), another small dataset of 300 images of *Car Paint* materials, which cannot be well reproduced with the standard GGX [Walter et al. 2007] model, and a larger “in-the-wild” dataset containing 9,000 images of materials covering many material categories (wood, stone, paint, leather, brick, metal, plastic, marble, fabric, paper, etc). We show examples of our “in-the-wild” dataset in Fig. 6. We combine all three datasets to build a larger 12,000-image *Diverse* dataset. The *Glossy* and *Car Paint* datasets are used to train our 256^2 and metallic paint models. The *Diverse* dataset is used to train 512^2 and 1024^2 models.

To compute the light position, we start by computing the intensity map as the minimum of RGB channels per pixel. We gamma correct (2.2) the intensity map and compute its mean. We finally obtain the flash light position by computing the weighted average of all pixels with intensity value greater than this mean intensity. This method works for most examples, for the few examples where the light position is detected incorrectly, we manually set light position as the center of the image. The accuracy of light detection is illustrated visually in the supplementary materials.

5 RESULTS

In this section, we show the outputs of the relightable generator and material maps estimator for 256^2 , 512^2 and 1024^2 resolutions. We then demonstrate that our method can be extended to other material reflectance models by fitting a special smaller car paint dataset. We then compare PhotoMat with TileGen to show our results are more visually realistic and perform an user study for comparison. Finally, we demonstrate that PhotoMat can be extended to a tileable version either using HexTiling [Mikkelsen 2022] or with the tiling strategy proposed by TileGen.

5.1 Neural Relit Materials and Analytical Material Maps

Low-resolution Results. We first show the results of a 256^2 model trained on the small 2.5k *Glossy* dataset. In Fig. 5, we show the relit neural materials, where each triple share the same latent space z

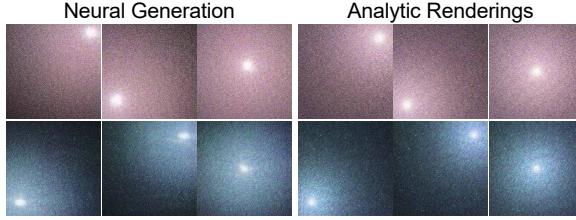


Figure 7: We collect a small car paint dataset with challenging appearance (micro-flakes). For this material, we fit a two specular lobes analytic model, each with its own albedo and roughness, but no normals. We show that this approach generates realistic relit materials (Neural Generation) and the estimated analytic model matches them closely (Analytic Renderings).

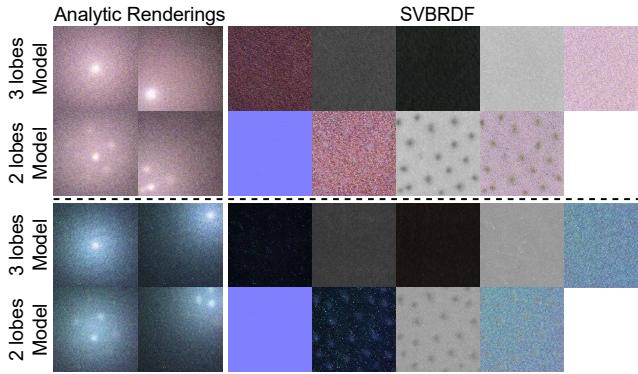


Figure 8: Comparison of our multi-lobe analytic BRDF model (3-lobes model) for car paint with the standard BRDF model (2-lobes model) used for other materials. The 3-lobes BRDF includes one diffuse and two specular lobes, and each specular lobe has its own albedo and roughness. We see that the 2-lobes model cannot correctly fit the learned neural generator, producing multiple highlights and holes, but 3-lobes model succeeds.

but with different conditional point light sources. As shown in the figure, the relit neural materials cover diverse material categories, and relighting results are realistic and consistent under different illuminations. The material parameter maps (normal, diffuse albedo, roughness and specular albedo) produced by our maps estimator \mathbf{E} and the analytic renderings are shown in Fig. 10. As can be seen, the analytic renderings using the estimated maps are comparable to the relighting results, demonstrating that the maps estimator can output material maps that reproduce the appearance of generated neural materials with minimal quality loss between relit neural materials and analytic renderings.

High-resolution Results. We also train models of size 512^2 and 1024^2 on our larger *Diverse* dataset, which covers more diverse material categories. In Fig. 13, we show side-by-side comparisons of generated relit neural materials and the corresponding renderings from estimated analytic BRDF parameters, along with the estimated parameter maps. For this larger dataset, we use normal, diffuse

albedo, roughness and specular albedo as material parameters. As is shown here, our higher resolution generator can also produce diverse and realistic relit materials, and the maps estimator can extract the reasonable corresponding materials maps even without supervision from ground truth material maps. We provide more visual results in supplementary materials.

5.2 Category-specific Model

Our approach is not limited to a standard BRDF model and can be extended to any specialized BRDF model that is able to reproduce the appearance of a specific dataset. In Fig. 7, we show results of model trained on the *Car Paint* dataset. More specifically, we train a relightable generator on car paint photographs, and adjust the maps estimator to fit a coated BRDF model with one diffuse lobe and two specular lobes, each with its own albedo and roughness. As is shown in the figure, PhotoMat generates realistic relit neural materials (left) and the analytic renderings generated using the estimated SVBRDF matches them closely (right).

We further compare our special analytic BRDF model to the standard BRDF model used for other materials in Fig. 8. We can observe that multiple holes are baked into the estimated maps for the standard model, since the standard BRDF model is not expressive enough to correctly fit the complex highlight falloff shape that our learned neural generator, trained on car paint dataset, captures. In comparison, our coated BRDF models produces high quality material maps that fit well to the relit neural materials. This illustrates that our approach is not tied to a specific BRDF model; neural generator easily extends to different material appearance and we can fit custom materials by training a corresponding maps estimator.

5.3 Comparison against TileGen

We compare PhotoMat against TileGen [Zhou et al. 2022] visually and by conducting an user study. Note that TileGen is trained per category (tiles, leather, stone and metal), while our model is general. So we compare results for two material categories: stone and leather. For PhotoMat, we use analytic renderings using estimated analytic material maps and for TileGen we sampled results from leather and stone pretrained model. The visual comparison in Fig 9 obviously show that PhotoMat trained on real images produces more realistic appearance compared to TileGen trained on a synthetic dataset. Please refer to figures in supplementary material for more visual comparison, where we add 30 materials per category for each method and show them side by side.

User study. To evaluate this claim of realism of our method against that of TileGen, we conduct a user study with 30 participants with varied backgrounds in graphics. We show 20 randomly selected pairs of materials, 10 leather and 10 stones, from both our method and TileGen, in the form of flash lit renderings. We use a 2AFC method asking the users to select, for each stimulus, "Which image looks more like a photograph of a real commonly found surface, as opposed to an image synthetically designed by an artist?". The stimuli are randomly ordered and named. On average, participants find our method's results more realistic 79.2% of the time. Further, out of 20 materials, 18 of ours are preferred and 1 shows equal preference. Out of 30 participants, 2 preferred TileGen results in the pairs (at 65% and 55%). The rest preferred

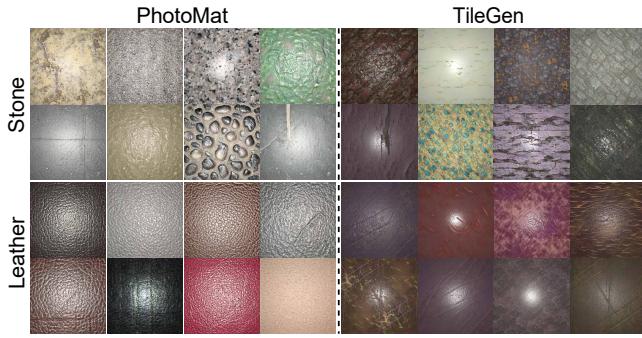


Figure 9: For stone and leather, we compare a set of generated materials by our models against the models from TileGen, which are based on synthetic data. A larger version of this comparison can be found in the supplementary materials. We also conducted a user study, which confirms that our models generate more realistic materials overall.

ours (between 60% and 100%). This study confirms that training on real data does indeed result in more realistic generated SVBRDFs.

5.4 Tileable and Extended Materials

To extend the spatial extent covered by our generated materials, we demonstrate that PhotoMat can be extended to a tileable version by either following the strategy proposed by TileGen or using HexTile [Mikkelsen 2022]. We train a tileable 256^2 model following the unconditional TileGen’s tiling technique. The original renderings and 2×2 tiled renderings of tileable PhotoMat are shown in Fig.11. Moreover, we also implement HexTile [Mikkelsen 2022], a previous method to smoothly blend hexagonal tiles extracted from the original material. The original renderings and tiled renderings are shown in Fig.12. Both strategies enable seamless renderings of PhotoMat materials, which can be passed to a rendering pipeline.

6 LIMITATIONS

Our dataset is naturally biased towards materials and scale ranges for which cell phone capture is convenient, and remains limited in scale. Scaling up our dataset size is likely to lead to the most dramatic improvements, much like with generative models in other domains [Ramesh et al. 2022]. We observed cases of mode collapse when training high-resolution and tileable versions of PhotoMat. As mode collapse is a known drawback of GANs, different generative architectures such as diffusion models may be interesting to explore.

7 CONCLUSION AND FUTURE WORK

In this paper, we show that it is possible to train a material generative model without any synthetic data, by using real flash-photographs dataset captured using a light-weight cell phone. Since the desired reflectance maps of real materials (albedo, normal, roughness, etc.) are never directly observed, we solve the problem in two steps.

First, by simple cropping of the original photos with centered flash highlights, we obtain real training images with a variety of highlight locations. We train a GAN producing similar cropped flash images conditional on the flash highlight position. From the implicit neural material representation produced by the GAN, we

train another neural network to extract per-texel analytic material parameters. We demonstrate a number of materials sampled from our generative model, applied in the context of a full rendered 3D scene with global illumination (see Fig. 1 and supplemental material).

As it only requires a cellphone, our data collection can be easily scaled to large material datasets; we provide such a dataset of 12,000 photos, and models trained on this dataset. While our trained models can be already used to generate material maps for practical use, we also believe that our approach opens the possibility of further extensions to different material models, other lighting conditions and larger real datasets. We believe this is a significant step in reducing the reliance of material research on synthetic data. This is especially important given the rapid progress in image generation demonstrated by diffusion models trained on extreme large-scale datasets. Extending these approaches to materials, for example to train conditional materials generators based on text or image prompts, would require approaches like ours to scale to these data requirements.

ACKNOWLEDGMENTS

This project was funded in part by the NSF CAREER Award #2238193 and a generous gift from Adobe.

REFERENCES

- Adobe. 2023. Substance 3D Designer.
- Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. 2021. Efficient Geometry-aware 3D Generative Adversarial Networks. In *arXiv*.
- Valentin Deschaintre, Miika Aittala, Frédéric Durand, George Drettakis, and Adrien Bousseau. 2018. Single-image SVBRDF Capture with a Rendering-aware Deep Network. *ACM Trans. Graph.* 37, 4 (2018), 128:1–128:15.
- Valentin Deschaintre, Miika Aittala, Frédéric Durand, George Drettakis, and Adrien Bousseau. 2019. Flexible SVBRDF Capture with a Multi-Image Deep Network. *Computer Graphics Forum* 38, 4 (2019).
- Valentin Deschaintre, George Drettakis, and Adrien Bousseau. 2020. Guided Fine-Tuning for Large-Scale Material Transfer. *Computer Graphics Forum (Proceedings of the Eurographics Symposium on Rendering)* 39, 4 (2020). <http://www-sop.inria.fr/reves/Basilic/2020/DDB20>
- Jiahui Fan, Beibei Wang, Milos Hasan, Jian Yang, and Ling-Qi Yan. 2022. Neural Layered BRDFs (*SIGGRAPH '22*). Article 4, 8 pages.
- Duan Gao, Xiao Li, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. 2019. Deep inverse rendering for high-resolution SVBRDF estimation from an arbitrary number of images. *ACM Trans. Graph.* 38, 4 (2019).
- Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2015. A Neural Algorithm of Artistic Style. *arXiv:1508.06576*
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems* 27. 2672–2680.
- Dar’ya Guarnera, Giuseppe Claudio Guarnera, Abhijeet Ghosh, Cornelia Denk, and Mashhuda Glencross. 2016. BRDF Representation and Acquisition. *Computer Graphics Forum* (2016).
- Paul Guerrero, Milos Hasan, Kalyan Sunkavalli, Radomir Mech, Tamy Boubekeur, and Niloy Mitra. 2022. MatFormer: A Generative Model for Procedural Materials. *ACM Trans. Graph.* 41, 4, Article 46 (2022). <https://doi.org/10.1145/3528223.3530173>
- Jie Guo, Shuichang Lai, Chengzhi Tao, Yuelong Cai, Lei Wang, Yanwen Guo, and Ling-Qi Yan. 2021. Highlight-Aware Two-Stream Network for Single-Image SVBRDF Acquisition. *ACM Trans. Graph.* 40, 4, Article 123 (jul 2021), 14 pages. <https://doi.org/10.1145/3450626.3459854>
- Yu Guo, Cameron Smith, Miloš Hašan, Kalyan Sunkavalli, and Shuang Zhao. 2020. MaterialGAN: Reflectance Capture using a Generative SVBRDF Model. *ACM Trans. Graph.* 39, 6 (2020), 254:1–254:13.
- Philipp Henzler, Valentin Deschaintre, Niloy J. Mitra, and Tobias Ritschel. 2021. Generative Modelling of BRDF Textures from Flash Images. *ACM Trans. Graph.* 40, 6, Article 284 (dec 2021), 13 pages.
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local

- nash equilibrium. *Advances in neural information processing systems* 30 (2017).
- Yiwei Hu, Miloš Hašan, Paul Guerrero, Holly Rushmeier, and Valentin Deschaintre. 2022. Controlling Material Appearance by Examples. *Computer Graphics Forum* (2022). <https://doi.org/10.1111/cgf.14591>
- Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2019. Analyzing and Improving the Image Quality of StyleGAN. arXiv:1912.04958
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- Alexandr Kuznetsov, Krishna Mulla, Zexiang Xu, Miloš Hašan, and Ravi Ramamoorthi. 2021. NeuMIP: multi-resolution neural materials. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–13.
- Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. 2017. Modeling Surface Appearance from a Single Photograph Using Self-Augmented Convolutional Neural Networks. *ACM Trans. Graph.* 36, 4 (2017), 45:1–45:11.
- Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. 2018. Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part III (Lecture Notes in Computer Science, Vol. 11207)*, 74–90.
- Rosalie Martin, Arthur Roullier, Romain Rouffet, Adrien Kaiser, and Tamy Boubekeur. 2022. MaterIA: Single Image High-Resolution Material Capture in the Wild. *Computer Graphics Forum* 41, 2 (2022), 163–177. <https://doi.org/10.1111/cgf.14466> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14466>
- Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. 2003. A Data-Driven Reflectance Model. *ACM Trans. Graph.* 22, 3 (2003), 759–769.
- Morten S Mikkelsen. 2022. Practical Real-Time Hex-Tiling. *Journal of Computer Graphics Techniques* Vol 11, 2 (2022).
- Gilles Rainer, Abhijeet Ghosh, Wenzel Jakob, and Tim Weyrich. 2020. Unified Neural Encoding of BTFs. *Computer Graphics Forum (Proceedings of Eurographics)* 39, 2 (2020), 167–178.
- Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.
- Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet Models for Refraction Through Rough Surfaces. *EGSR 2007* (2007), 195–206.
- Xilong Zhou, Milos Hasan, Valentin Deschaintre, Paul Guerrero, Kalyan Sunkavalli, and Nima Khademi Kalantari. 2022. TileGen: Tileable, Controllable Material Generation and Capture. In *SIGGRAPH Asia 2022 Conference Papers* (Daegu, Republic of Korea) (SA '22). Association for Computing Machinery, New York, NY, USA, Article 34, 9 pages. <https://doi.org/10.1145/3550469.3555403>
- Xilong Zhou and Nima Khademi Kalantari. 2021. Adversarial Single-Image SVBRDF Estimation with Hybrid Training. *Computer Graphics Forum* (2021). <https://doi.org/10.1111/cgf.142635>
- Xilong Zhou and Nima Khademi Kalantari. 2022. Look-Ahead Training with Learned Reflectance Loss for Single-Image SVBRDF Estimation. *ACM Transactions on Graphics (TOG)* 41, 6 (2022), 1–12.

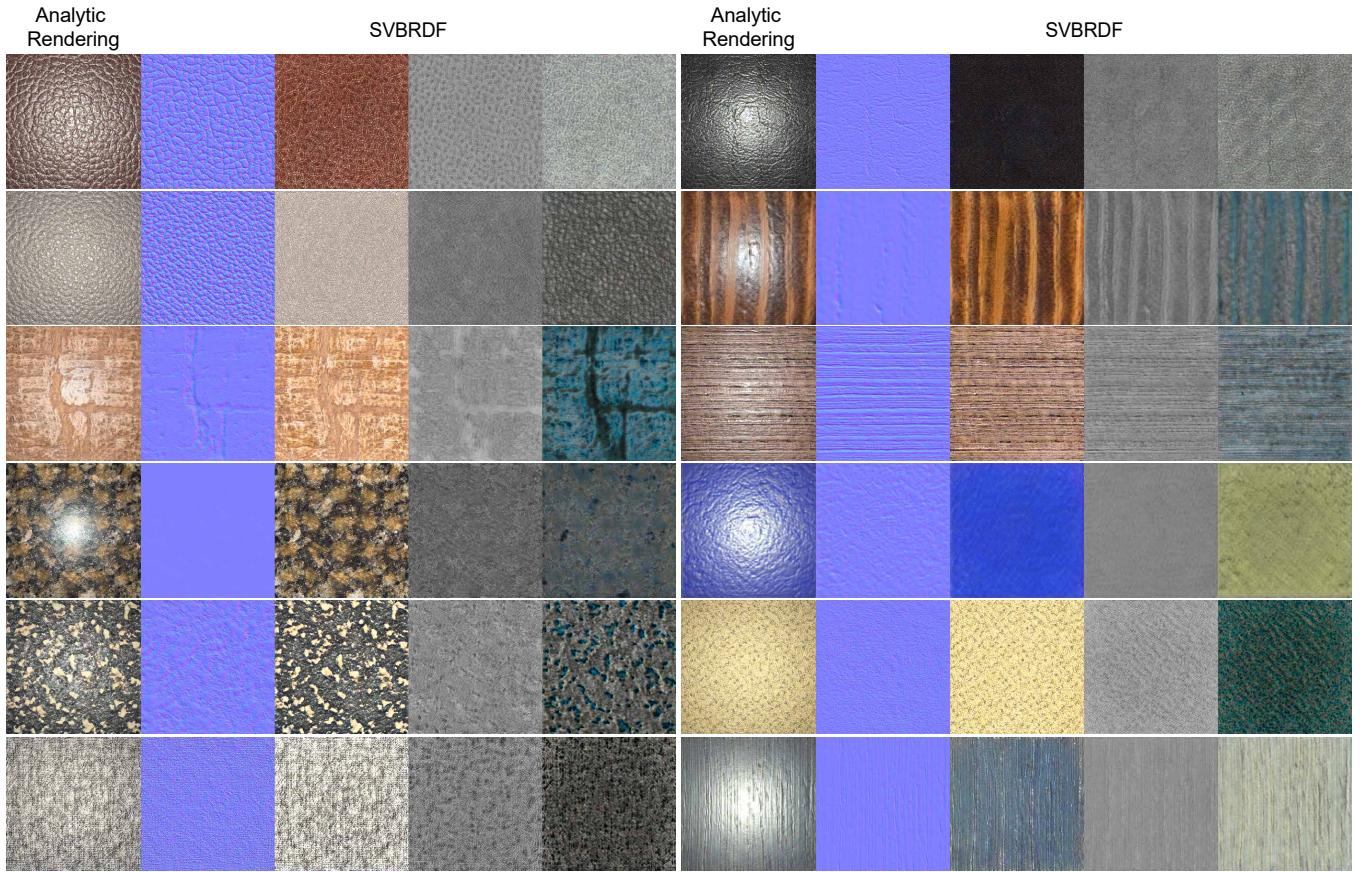


Figure 10: The material parameter maps produced by our maps estimator E for the examples shown in Fig. 5. The four maps are (left to right) normal, diffuse albedo, roughness and specular albedo. In practice, we estimate height and derive normal as its gradient. The occasional colored specular coefficient is due to imperfect white balance of the in-the-wild photos and can be easily corrected to grayscale if desired.

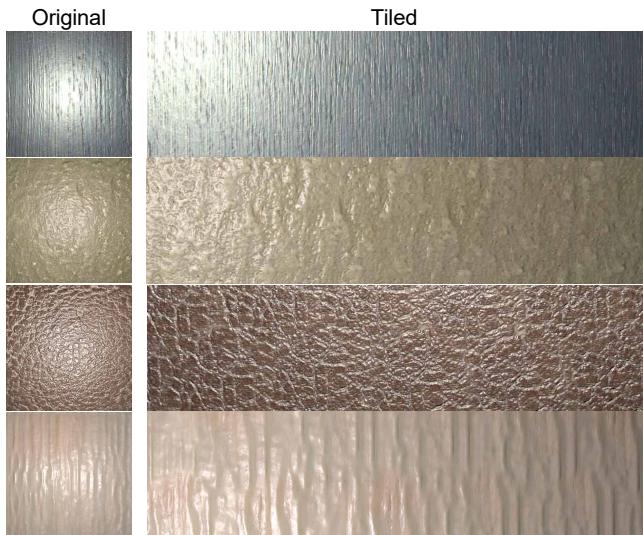


Figure 12: We apply the HexTile method [Mikkelsen 2022] to tile our estimated material maps. The original rendered materials (left) are compared with tiled rendered materials (right), demonstrating that our estimated maps can be smoothly tiled and applied to the rendering pipeline.

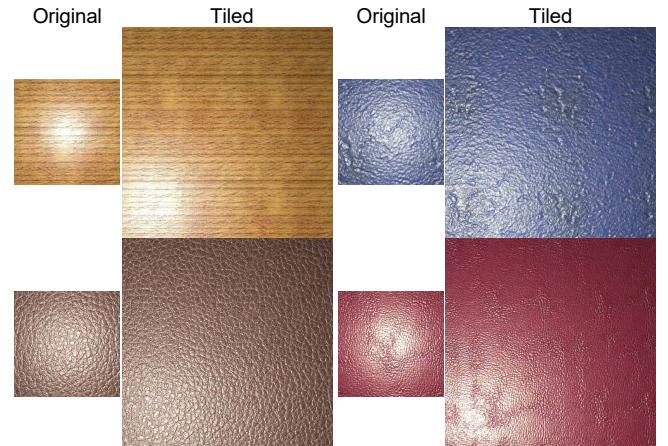


Figure 11: We trained a 256^2 model, tileable by construction, following the TileGen tiling strategy. This model produces natively tileable material maps; we show the renderings using original material maps alongside 2×2 tiled versions.

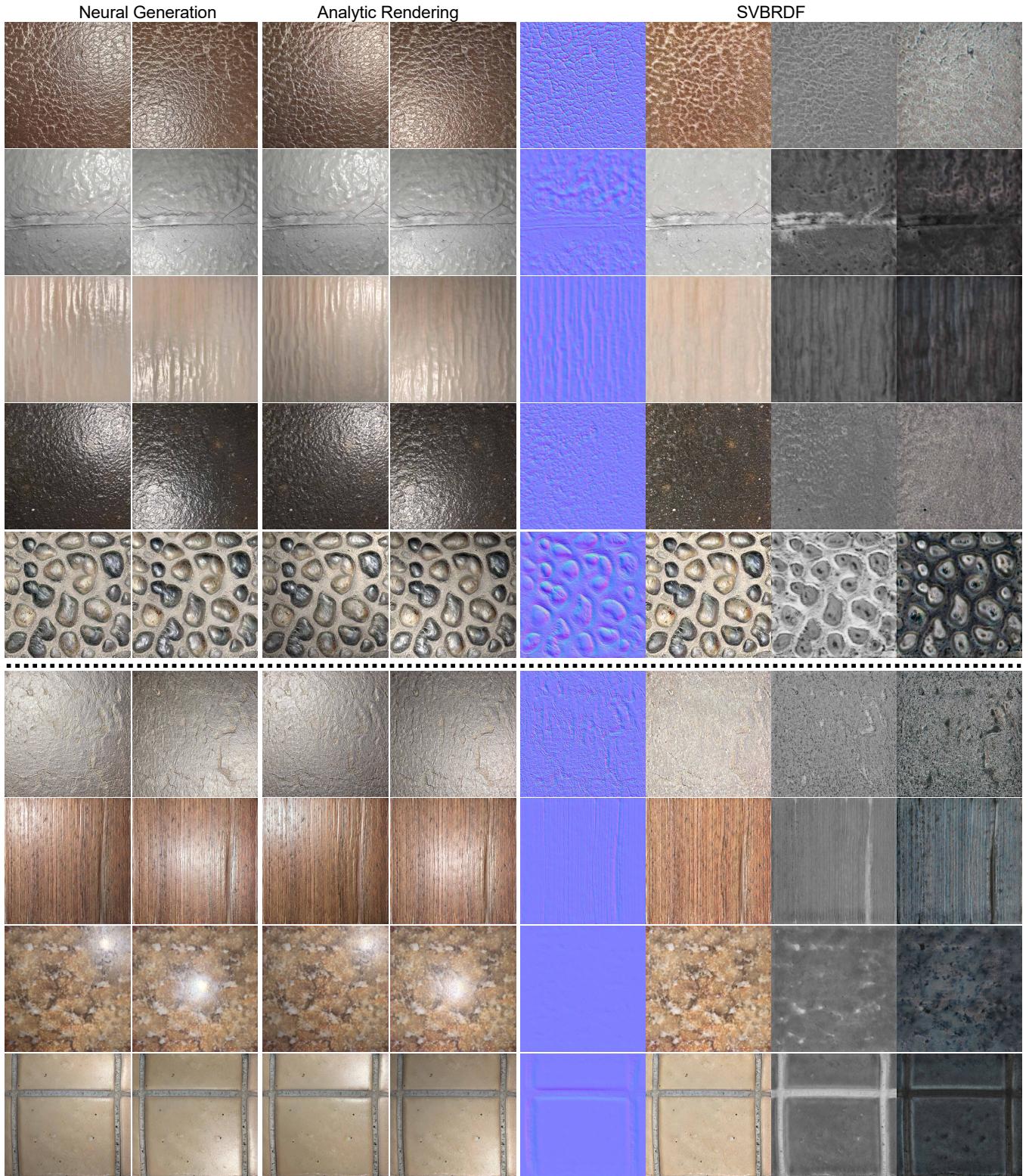


Figure 13: We train models of size 512^2 (top) and 1024^2 (bottom) on our largest *Diverse* dataset. Here we show generated relightable materials (Neural Generation), corresponding renderings from estimated analytic BRDF parameters (Analytic Rendering) and the material maps (SVBRDF), showing normal, diffuse albedo, roughness and specular albedo.