MaterialGAN: Reflectance Capture using a Generative SVBRDF Model

YU GUO, University of California, Irvine
CAMERON SMITH, Adobe Research
MILOŠ HAŠAN, Adobe Research
KALYAN SUNKAVALLI, Adobe Research
SHUANG ZHAO, University of California, Irvine

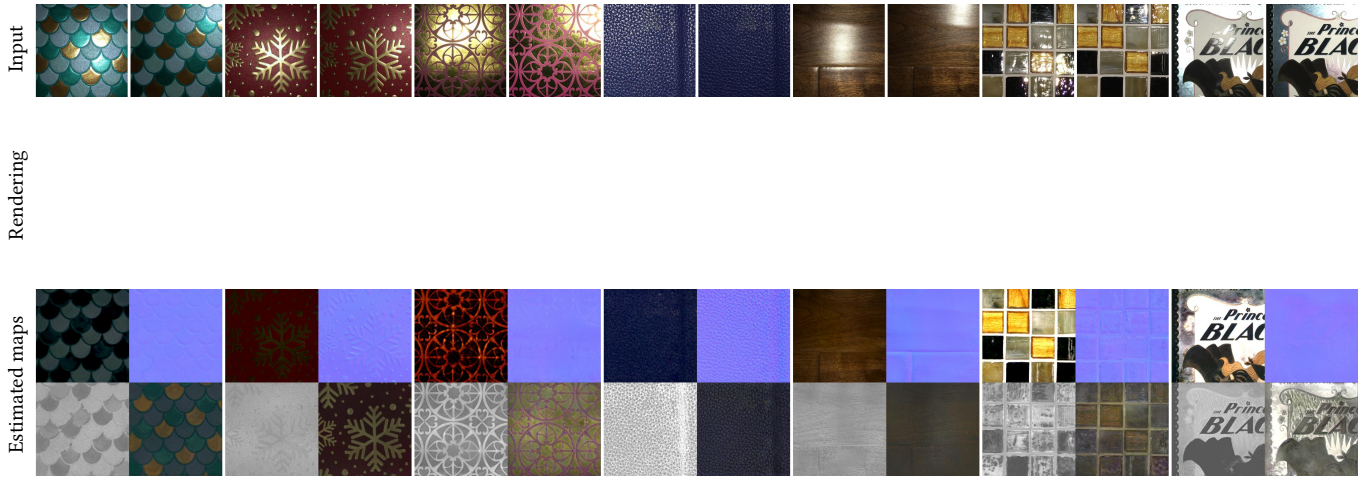


Fig. 1. We introduce a method to capture SVBRDF material maps from a small number of mobile flash photographs, achieving high quality results both on original and novel views. Our key innovation is optimization in the latent space of MaterialGAN, a generative model trained to produce plausible material maps; MaterialGAN thus serves as a powerful implicit prior for result realism. Here we show re-rendered views for several different materials under environment illumination. We use 7 inputs for these results (with 2 of them shown). (Please use Adobe Acrobat and click the renderings to see them animated.)

We address the problem of reconstructing spatially-varying BRDFs from a small set of image measurements. This is a fundamentally under-constrained problem, and previous work has relied on using various regularization priors or on capturing many images to produce plausible results. In this work, we present *MaterialGAN*, a deep generative convolutional network based on StyleGAN2, trained to synthesize realistic SVBRDF parameter maps. We show that MaterialGAN can be used as a powerful material prior in an inverse rendering framework: we optimize in its latent representation to generate material maps that match the appearance of the captured images when rendered. We demonstrate this framework on the task of reconstructing SVBRDFs from images captured under flash illumination using a hand-held mobile phone. Our method succeeds in producing plausible material maps that accurately reproduce the target images, and outperforms previous state-of-the-art material capture methods in evaluations on both synthetic and real data. Furthermore, our GAN-based latent space allows for high-level semantic material editing operations such as generating material variations and material morphing.

CCS Concepts: • **Computing methodologies** → **Rendering**.

Additional Key Words and Phrases: SVBRDF capture, generative adversarial network.

ACM Reference Format:
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, Article 254 (December 2020), 13 pages. https://doi.org/10.1145/3414685.3417779

1 INTRODUCTION

Despite a few decades of effort in computer graphics and vision, capturing spatially-varying reflectance of real-world materials remains a challenging and actively researched task. Measurement methods have traditionally used custom hardware systems to densely sample illumination and viewing directions [Marschner et al. 1999; Matusik et al. 2003], followed by post-processing such as fitting parametric BRDF models [Ngan et al. 2005]. However, such approaches are restricted to laboratory conditions.

Recent work has explored methods for casual capture of spatially-varying BRDFs (SVBRDFs) using commodity hardware and in less constrained environments [Aittala et al. 2013, 2015; Francken et al.

Authors' addresses: Yu Guo, University of California, Irvine; Cameron Smith, Adobe Research; Miloš Hašan, Adobe Research; Kalyan Sunkavalli, Adobe Research; Shuang Zhao, University of California, Irvine.

0730-0301/2020/12-ART254
https://doi.org/10.1145/3414685.3417779

2009; Hui et al. 2017; Ren et al. 2011]. These methods usually follow an *inverse-rendering* approach: they define a forward rendering model and optimize reflectance parameters so that the simulated appearance matches physical measurements under certain image metrics. With a small number of measured images, this approach is fundamentally under-constrained: there usually exist many material estimates capable of producing renderings that match the measurements, but many of these estimates can be unrealistic and may not generalize to novel illumination and viewing conditions. The solution to this problem has been to *regularize* the optimization using pre-determined *material priors* such as linear low-dimensional BRDF models [Hui et al. 2017; Ren et al. 2011] or stationary stochastic textures [Aittala et al. 2016, 2015]. However, such hand-crafted priors do not generalize to a wide range of real-world materials.

More recently, learning-based approaches have demonstrated remarkable results for reconstructing SVBRDFs from one [Deschaintre et al. 2018; Li et al. 2018] or more images [Deschaintre et al. 2019]. While these methods use rendering-based losses (similar to the inverse rendering approaches) during training, at test time they predict SVBRDFs from images using a single feed-forward pass through a deep network. As a result, the reconstructed material parameters may not accurately reproduce the measured appearance. In contrast, Gao et al. [2019] propose using an optimization-based approach in conjunction with a learned material prior. Specifically, they train a fully-convolutional auto-encoder on a large material dataset and optimize in the latent space of this auto-encoder. This ensures that the reconstructed SVBRDF parameters both reproduce the measurements and are plausible real-world materials. However, while this learned material prior is a significant improvement over hand-crafted priors, it still produces a relatively localized and highly flexible latent space that requires a good initialization (for example, from single image methods [Deschaintre et al. 2018; Li et al. 2018]) and even then can fail to produce good results.

In this paper, we propose a different material prior that builds on the remarkable progress in image synthesis using deep Generative Adversarial Networks (GANs) [Goodfellow et al. 2014a; Karras et al. 2018a,b]. We train *MaterialGAN*—a StyleGAN2-based deep convolutional neural network [Karras et al. 2019]—to generate plausible materials from a large-scale, spatially-varying material dataset [Deschaintre et al. 2018]. MaterialGAN learns *global* correlations in material parameters, both spatially (thus encoding texture patterns) as well as across parameters (for example, relationships between diffuse and specular parameters). As illustrated in Figure 2, sampling from the MaterialGAN latent space produces plausible, realistic materials with complex variations and diverse appearance.

While GANs have traditionally been used to synthesize images, we demonstrate a very different application, using MaterialGAN as a powerful prior in an inverse rendering-based material capture framework. We append a rendering layer to MaterialGAN, setting up a differentiable pipeline from the learned latent space, through generating material maps, to rendering images under specified views and lighting. This allows us to optimize the MaterialGAN latent vector(s) to minimize the error between the rendered and measured images and reconstruct the corresponding material maps. Doing so ensures that the reconstructed SVBRDFs lie on the "manifold of realistic materials", while at the same time accurately reproducing the captured images.

We demonstrate that our GAN-based optimization framework produces high-quality SVBRDF reconstructions from a small number (3-7) images captured under flash illumination using hand-held mobile phones, and improves upon previous state-of-the-art methods [Deschaintre et al. 2019; Gao et al. 2019]. In particular, it produces cleaner, more realistic material maps that better reproduce the appearance of the captured material under both input *and novel* lighting. Moreover, as illustrated in Figure 10, MaterialGAN adapts to a wide range of SVBRDF samples ranging from diffuse to specular materials and near-stochastic textures to structured patterns with multiple distinct, complex materials.

Furthermore, our GAN-based latent space offers the ability to edit the latent vector in semantically meaningful ways (via operations like interpolation in the latent space) and generate realistic materials that go beyond the captured images. This is not possible with current material capture methods that do not afford any control over their per-pixel BRDF estimates.

2 RELATED WORK

Reflectance capture. Acquiring material data from physical measurements is the goal of a broad range of methods. Please refer to surveys [Dong 2019; Guarnera et al. 2016; Weyrich et al. 2009] for more comprehensive introduction to the related works.

Most reflectance capture approaches observe a material sample under varying viewing and lighting configurations. They differ in the number of light patterns required and their types such as moving linear light [Gardner et al. 2003; Ren et al. 2011], Gray code patterns [Francken et al. 2009], spherical harmonic illumination [Ghosh et al. 2009], and Fourier patterns [Aittala et al. 2013].

Methods have also been proposed for material capture "in the wild", i.e., under uncontrolled environment conditions with commodity hardware, typically captured with a hand-held mobile phone with flash illumination. Some of these methods impose strong priors on the materials, such as linear combinations of basis BRDFs [Hui et al. 2017; Xu et al. 2016] (where the basis BRDFs can come from the measured data [Matusik et al. 2003]). Later work by Aittala et al. [2016; 2015] estimated per-pixel parameters of stationary spatially-varying SVBRDFs from two-shot and one-shot photographs. In the latter case, the approach used a neural Gram-matrix texture descriptor based on the texture synthesis and feature transfer work of Gatys [2015; 2016] to compare renderings with similar texture patterns but without pixel alignment.

More recently, deep learning-based approaches have demonstrated remarkable progress in the quality of SVBRDF estimates from single images (usually captured under flash illumination) [Deschaintre et al. 2018; Li et al. 2017, 2018]. These methods train deep convolutional neural networks with large datasets of artistically created SVBRDFs, and with a combination of losses that evaluate the difference in material maps and renderings from the dataset ground truth.

Deschaintre [2019] extended the single-shot approach to multiple images. The key idea is to extract features from the input images with a shared encoder, max-pooling the features and decoding the

final maps from the pooled features. This architecture has the benefit of being independent of the number of inputs, while also not requiring explicit light position information. In our experience, this approach produces smooth, plausible maps with low artifacts; however, re-rendering the maps tends to be not as close to the target measurements because the network cannot "check" its results at runtime. Moreover, we find that especially on real data, this method also has strong biases such as dark diffuse albedo maps and exaggerating surface normals (especially along strong image gradients that might be caused by albedo variations). We believe this is not due to any technical flaw; the method may be reaching the limit of what is possible using current feed-forward convolutional architectures and currently available datasets.

Gao et al. [2019] introduced an inverse rendering-based material capture approach that optimizes for material maps to minimize error with respect to the captured images. Since this is an under-constrained problem, they propose optimizing over the latent space of a learned material auto-encoder network to minimize rendering error. This approach has the benefit of explicitly matching the appearance of the captured image measurements, while also using the auto-encoder as a material "prior". Moreover, the encoder and decoder are fully convolutional, which has the advantage of resolution independence. However, we find that the convolutional nature of this model also has the disadvantage of only providing local regularization and not capturing global patterns in the material, such as the long-range spatial patterns and correlations between the different material parameter maps. As a result, this method relies on previous methods (for example, Deschaintre et al. [2018]) to provide a good initialization, without which it can converge to poor results. In contrast, our MaterialGAN is a more globally robust latent space and produces higher quality reconstructions without requiring accurate initializations, though it is no longer resolution-independent.

Generative adversarial networks. GANs [Goodfellow et al. 2014b] have become extremely successful in the past few years in various domains, including images [Radford et al. 2015], video [Tulyakov et al. 2018], audio [Donahue et al. 2018], and 3D shapes [Li et al. 2019]. A GAN typically consists of two competing networks; a generator, whose goal is to produce results that are indistinguishable from the real data distribution, and a discriminator, whose job is to learn to identify generated results from real ones. For generating realistic images (especially of human faces), there has been a sequence of improved models and training strategies, including ProgressiveGAN [Karras et al. 2018a], StyleGAN [2018b] and StyleGAN2 [2019]. StyleGAN2 in particular is the state-of-the-art GAN model and our work is based on its architecture, modified to output more channels.

Recently, GANs have also been used to solve inverse problems [Asim et al. 2019; Bora et al. 2017; O'Malley et al. 2019]. In computer graphics and vision, this work has focused on embedding images into the latent space, with the goal of editing the images in semantically meaningful ways via latent vector manipulations [Zhu et al. 2016]. This embedding requires solving an optimization problem to find the latent vector. More recent work such as Image2StyleGAN [Abdal et al. 2019b] and Image2StyleGAN++ [Abdal et al. 2019a] has looked at problem of embedding images specifically into the the StyleGAN latent space. While these methods focus on projecting portrait images into face-specific StyleGAN models, we find their analysis can be adapted to our problem. We build on this to propose a GAN embedding-based inverse rendering approach.

3 MATERIALGAN: A GENERATIVE SVBRDF MODEL

Generative Adversarial Networks [Goodfellow et al. 2014a] are trained to map an input from a latent space (often randomly sampled from a multi-variate normal distribution) to a plausible instance of the target distribution. In recent years, GANs have made remarkable progress in synthesizing high-resolution, photo-realistic images. Inspired by this progress, we propose MaterialGAN, a GAN that is trained to generate plausible materials, thus implicitly learning an SVBRDF manifold. MaterialGAN is based on the architecture of StyleGAN2 [Karras et al. 2019].

3.1 Overview of StyleGAN and its latent spaces

StyleGAN2 [Karras et al. 2019] is an improvement of StyleGAN [Karras et al. 2018b] and is the state-of-the-art generative adversarial network (GAN) for image synthesis, especially for human faces. The architecture has several advantages over previous models like ProgressiveGAN [Karras et al. 2018a] and DCGAN [Radford et al. 2015]. For our purposes, the main advantage is that the model is not simply a black-box stack of convolution and upsampling layers, but has additional, more specific structure, allowing for much easier inversion (latent space optimization). The StyleGAN2 architecture starts with a learned constant $4 \times 4 \times 512$ tensor and progressively upsamples it to the final output target resolution via a sequence of convolutional and upsampling layers (7 in total to end with a final image resolution of 256×256). Given an input latent code vector $\boldsymbol{z} \in \mathcal{Z} \subset \mathbb{R}^{512}$, StyleGAN2 transforms it through a non-linear mapping network of fully-connected layers into an intermediate latent vector $\boldsymbol{w} \in \mathcal{W} \subset \mathbb{R}^{512}$. The rationale for the introduction of the space of $\mathcal{W}$ is that while $\mathcal{Z}$ requires (almost) every latent $\boldsymbol{z} \in \mathcal{Z}$ to correspond to a realistic output, vectors $\boldsymbol{w} \in \mathcal{W}$ are free from this overly stringent constraint, which leads to a less "entangled" mapping, with more meaningful dimensions (see [Karras et al. 2018b, 2019] for more discussion). In the original StyleGAN, the vector $\boldsymbol{w} \in \mathcal{W}$ is mapped via a learned affine transformation to mean and variance "style" vectors that control adaptive instance normalizations (AdaIN) [Huang and Belongie 2017] that are applied before and after every convolution in the generation process (thus $7 \times 2 = 14$ times for a model of resolution 256×256). The statistics of the AdaIN normalizations caused the feature maps and output images of StyleGAN to suffer from droplet artifacts. StyleGAN2 removes the droplet artifacts entirely by replacing the AdaIn normalization layers with a demodulation operation which bakes the entire style block into a single layer while maintaining the same scale-specific control as StyleGAN. We construct a matrix $\boldsymbol{w}^+ \in \mathcal{W}^+ \subset \mathbb{R}^{512\times14}$ by replicating $\boldsymbol{w}$ 14 times. During training and standard synthesis, the columns of $\boldsymbol{w}^+$ are identical, and correspond to $\boldsymbol{w}$. However, as we will discuss later (and similar to Abdal et al. [2019b]), we relax this constraint when optimizing for an embedding; $\mathcal{W}^+$ thus becomes an extended latent space, more powerful than $\mathcal{W}$ or $\mathcal{Z}$. Additionally, StyleGAN2 injects Gaussian noise, $\boldsymbol{\xi}$, into each of the 14 layers of the generator. This noise gives StyleGAN2 the ability

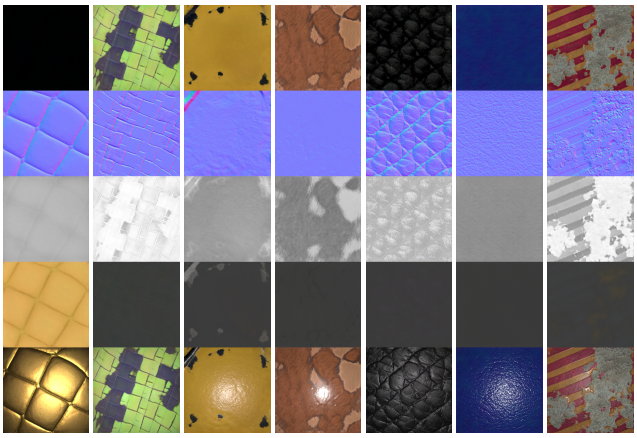


Fig. 2. Seven materials generated by randomly sampling MaterialGAN. Top to bottom: diffuse albedo, normal, roughness, specular albedo and renderings under flash illumination. As can be seen, the material maps are high-quality with meaningful correlations both spatially and across materials parameters, and visually look like plausible real-world materials.

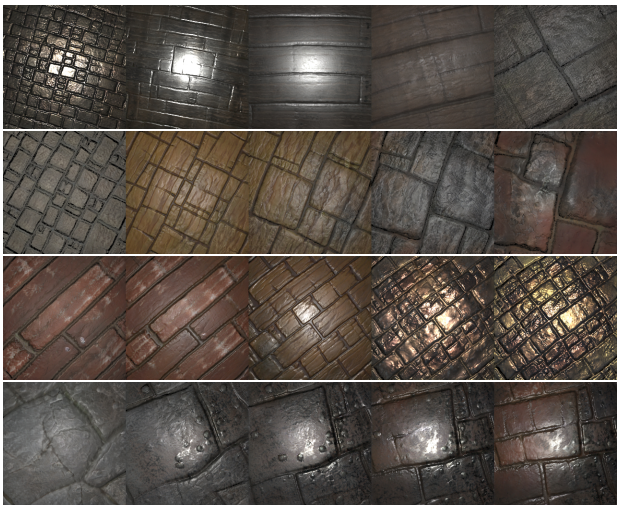


Fig. 3. **Interpolation in MaterialGAN latent space.** Each row shows an example of interpolation between two randomly generated materials, demonstrating non-linear morphing behavior.

to synthesize stochastic details at multiple resolutions. Abdal et al. [2019a] make the observation that one can also treat these noise inputs ξ as a latent space $\mathcal{N}$. Thus, combining these two spaces defines yet another latent space $\mathcal{W}^+\mathcal{N}$.

3.2 MaterialGAN training

MaterialGAN was trained with the dataset provided by Deschaintre et al. [2018] (and also used in Gao et al. [2019]). They generated this dataset by sampling the parameters of procedural material graphs from Allegorithmic Substance Share to create an initial set of 155 high-quality SVBRDFs at resolution 4096×4096. The dataset was augmented by blending multiple SVBRDFs and generating 256×256 resolution crops at random positions, scales and rotations. The final dataset consists of around 200,000 SVBRDFs. For detailed information about the curation of dataset we refer the reader to [Deschaintre et al. 2018]. Since pairs of SVBRDFs in the dataset were the same with only a slight variation, we selected 100,000 SVBRDFs. The maps for each SVBRDF are stacked in 9 channels (3 for albedo, 2 for normals, 1 for roughness, and 3 for specular albedo). We account for this by adapting the MaterialGAN architecture to output 9-channel outputs. MaterialGAN is trained in TensorFlow (version 1.15) with the same loss functions and similar hyper-parameters from StyleGAN2 [Karras et al. 2019]. StyleGAN2 configuration F was used for all experiments. The generator and discriminator were trained using Adam optimizers. The learning rate was increased per resolution from 0.001 to 0.0025 for both the generator and the discriminator. The discriminator was shown 25 million images. Training on 8× Nvidia Tesla V100 takes about 5 days. Figure 2 shows materials generated by randomly sampling the MaterialGAN latent space and images rendered from them. As can be seen here, MaterialGAN generates a wide variety of nearly photorealistic materials ranging from structured to stochastic, diffuse to specular, and with large-scale variations to fine detail. Furthermore, Figure 3 and the accompanying video show example interpolations between pairs of generated materials in the latent space, producing plausible non-linear morphing results.

4 SVBRDF CAPTURE USING MATERIALGAN

We utilize MaterialGAN, the powerful generative model described in the previous section, in a fundamentally new fashion: to *capture* SVBRDF maps. Specifically, we use MaterialGAN as a *material prior* for SVBRDF acquisition via an inverse rendering framework. Our goal is to estimate the SVBRDF parameter maps from one or a small number of photographs of a near-planar material sample. We utilize a common BRDF model that involves a diffuse and a specular component using the microfacet BRDF with the GGX normal distributions [Walter et al. 2007]. Our unknown parameter vectors $\boldsymbol{\theta} := (\boldsymbol{a}, \boldsymbol{n}, r, \boldsymbol{s})$ encode the four per-pixel parameter maps: diffuse albedo $\boldsymbol{a}$, surface normal $\boldsymbol{n}$, roughness r, and specular albedo $\boldsymbol{s}$. To recover the unknown parameter maps, we capture k images $\boldsymbol{I}_1, \cdots, \boldsymbol{I}_k$. We assume known viewing and lighting configurations for each image, which we denote as (L_i, C_i). Further, we assume that the material is lit by a single point source, collocated with the camera.[1] The images can be reprojected into a common frontal view (which is straightforward with a known viewing configuration). We introduce a differentiable rendering operator $\mathcal{R}$ that takes as input the parameter maps as well as the viewing and lighting configurations, and synthesizes corresponding images of the material. Under this setup, our goal is to find values of the unknown parameters $\boldsymbol{\theta}$ so that renderings with these parameters match the measurements $\boldsymbol{I}_i$. In other words, we focus on solving the following optimization problem:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{a},\boldsymbol{n},r} \sum_{i=1}^{k} \mathcal{L}(\mathcal{R}(\boldsymbol{\theta}; L_i, C_i), \boldsymbol{I}_i), \tag{1}$$

[1] In theory, non-collocated lights, area lights or projection patterns (e.g. on an LCD or similar screen) can be used as well, and would require a straightforward modification to our forward rendering process.

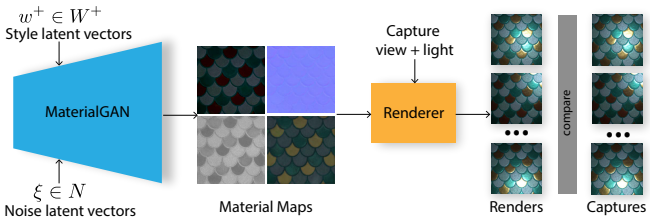


Fig. 4. Our inverse rendering pipeline. We optimize for latent vectors $\boldsymbol{w}^+$ and $\boldsymbol{\xi}$, that feed into the layers of the StyleGAN2-based MaterialGAN model. The MaterialGAN generator produces material maps (diffuse albedo, normal, roughness and specular albedo), that are rendered under the captured view/light settings. Finally, the renderings and measurements are compared using a combination of L2 and perceptual losses.

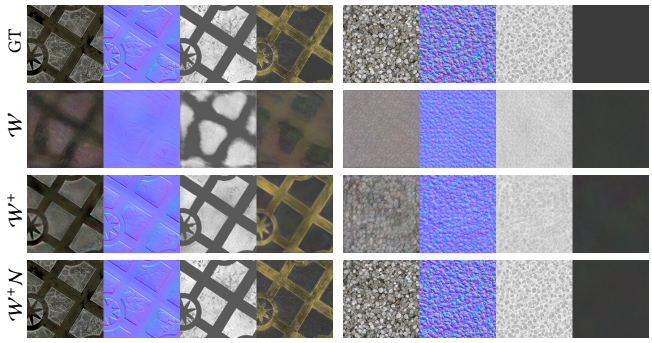


Fig. 5. **Embedding SVBRDFs into different latent spaces.** We take two synthetic SVBRDF material maps (top) and embed them into different latent spaces with and without the noise space (second–fourth rows). For illustration, we also embed the maps into a pure-noise space *only*; this is unable to recover the color at all.

where $\mathcal{L}$ is a loss function that measures the difference between the captured images, $\boldsymbol{I}_i$ and the renderings generated from the estimated SVBRDF parameters, $\mathcal{R}(\boldsymbol{\theta}; L_i, C_i)$.

4.1 Incorporating the MaterialGAN prior

Eq. (1) is, in general, a challenging optimization to solve due to its under-constrained nature. Given a small number of input measurements, the optimization can overfit to the input, producing implausible maps that do not generalize to novel views and lighting. To overcome this challenge, we leverage the MaterialGAN prior: instead of directly optimizing for the parameter maps $\boldsymbol{\theta}$, we can optimize for a vector $\boldsymbol{u}$ in the MaterialGAN *latent space* and map (decode) this latent vector back into material maps $\boldsymbol{\theta}$. The optimization problem then becomes:

$$\boldsymbol{u}^* = \arg\min_{\boldsymbol{u}} \sum_{i=1}^{k} \mathcal{L}(\mathcal{R}(\mathcal{G}(\boldsymbol{u}); L_i, C_i), \boldsymbol{I}_i), \tag{2}$$

where $\mathcal{G}$ is the learned MaterialGAN generator. Given that both $\mathcal{G}$ and $\mathcal{R}$ are differentiable operations, Eq. (2) can be optimized via gradient-based methods to estimate $\boldsymbol{u}^*$ and the corresponding SVBRDF maps $\mathcal{G}(\boldsymbol{u}^*)$. The above operation is similar to recent work on embedding images in the StyleGAN latent space [Abdal et al. 2019a,b]. The key difference is that we do not match material parameters directly, but evaluate their error through the rendering operator $\mathcal{R}(\cdot)$. To our knowledge, ours is the first approach to use a GAN latent space in combination with a rendering operator.

Loss function. We optimize Eq. 2 using a combination of a standard per-pixel L2 loss and a "perceptual loss" [Johnson et al. 2016] that has been shown to produce sharper results in image synthesis tasks:

$$\mathcal{L}(\boldsymbol{I}, \boldsymbol{I}') = \lambda_1 \mathcal{L}_{\text{pixel}} + \lambda_2 \mathcal{L}_{\text{percept}}, \tag{3}$$

The perceptual loss is defined as:

$$\mathcal{L}_{\text{percept}}(\boldsymbol{I}, \boldsymbol{I}') = \sum_{j=1}^{4} w_j^{\text{percept}} \left\| F_j(\boldsymbol{I}) - F_j(\boldsymbol{I}') \right\|_2^2, \tag{4}$$

where $F_1, \cdots, F_4$ are the flattened feature maps corresponding to the outputs of VGG-19 layers conv1_1, conv1_2, conv3_2, and conv4_2 from a pre-trained VGG network [Simonyan and Zisserman 2015]. See section 4.3 for more details.

Optimization details. We convert the TensorFlow-trained MaterialGAN model to PyTorch, in which our optimization framework is implemented. We optimize Eq. 2 using the Adam optimizer in PyTorch, with a learning rate of 0.01. We set all other hyper-parameters to default values. Now that our basic optimization framework is set up, there remain two key ingredients to implement our GAN-based optimization framework (Eq. (2)): (i) the choice of *latent space* that we optimize $\boldsymbol{u}$ over, and (ii) our optimization strategy to minimize the objective function. In the following sections, we describe our approach, along with an empirical analysis of these design choices.

4.2 Latent space

As discussed in Sec. 3.1, StyleGAN2 (and consequently, MaterialGAN) has a number of potential latent spaces. In particular, MaterialGAN uses three different *style* latent spaces: the input latent code $\boldsymbol{z} \in \mathcal{Z}$, the intermediate latent code $\boldsymbol{w} \in \mathcal{W}$ and per-layer styles $\boldsymbol{w}^+ \in \mathcal{W}^+$. StyleGAN2 also injects noise $\boldsymbol{\xi} \in \mathcal{N}$ into every layer of the network to generate stochastic variations. The typical forward generation process of the GAN only uses $\boldsymbol{z}$, with $\boldsymbol{w}$ being generated from $\boldsymbol{z}$ via a mapping network, and $\boldsymbol{w}^+$ being generated from $\boldsymbol{w}$ via affine transformations. However, Abdal et al. [2019b] note that the space of $\mathcal{Z}$ is too restrictive for accurate embedding of faces or other content into the GAN space. In other words, given the image of a human face, it is generally impossible to find a single $\boldsymbol{z} \in \mathcal{Z}$ such that the generated image closely matches the target. This remains the case even when extending the space to $\mathcal{W}$, i.e., when searching for a $\boldsymbol{w}$ instead of a $\boldsymbol{z}$. The space $\mathcal{W}^+$, on the other hand, offers much stronger representative power. Our experiments on embedding material maps into MaterialGAN demonstrate that optimizing for $\mathcal{W}^+$ is also needed for MaterialGAN to accurately reproduce input maps. We demonstrate this in Figure 5, via an experiment where we embed a given material (with known material maps) into MaterialGAN. As shown in rows (2) and (3), maps generated by optimizing $\boldsymbol{w}^+ \in \mathcal{W}^+$ contain more detail compared to those using $\boldsymbol{w} \in \mathcal{W}$.

On the other hand, some small-scale details are still missing. In fact, according to our experiments, only colors and large-scale features can be captured by the $\mathcal{W}^+$ space. For depicting high-frequency patterns, as demonstrated in rows (4) and (5) of Figure

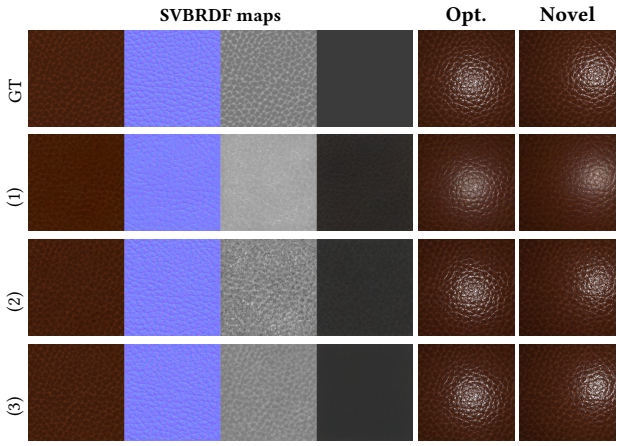


Fig. 6. **Optimization strategy.** We evaluated three optimization strategies: (1) optimize $\boldsymbol{w}^+$ first, then $\boldsymbol{\xi}$; (2) jointly optimize $\boldsymbol{w}^+$ and $\boldsymbol{\xi}$; (3) alternate between $\boldsymbol{w}^+$ and $\boldsymbol{\xi}$ every 10 iterations. Strategy (1) causes artifacts during the optimization, and (2) brings more noise into the maps. Particularly, for textures with small features, (1) and (2) may drive the optimization to bad local minima while the per-pixel loss could still be very low. Strategy (3) appears to be a good compromise, giving us better results in most cases. Note: "Opt." means an optimized input view or its re-rendering, i.e. not a novel view.

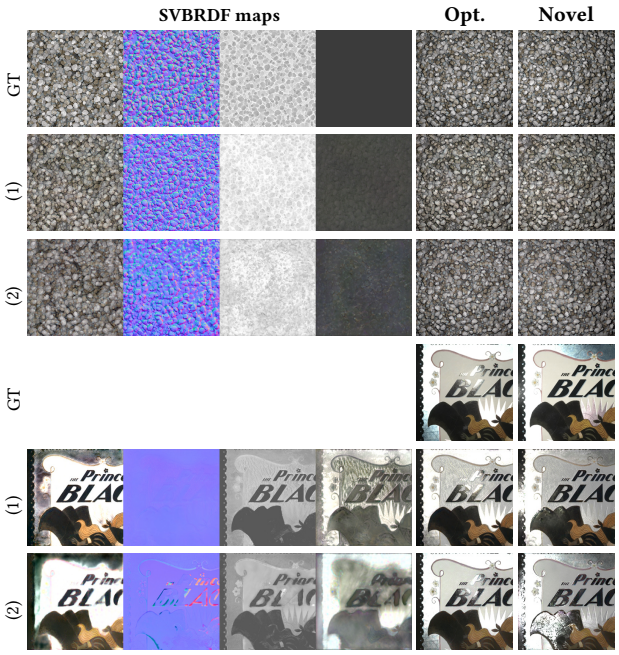


Fig. 7. **Noise optimization vs. post-refinement.** (1) Optimize $\boldsymbol{w}^+$ and $\boldsymbol{\xi}$ but no post-refinement; (2) Optimize $\boldsymbol{w}^+$ only but with post-refinement. This shows that $\boldsymbol{\xi}$ takes an important role; optimizing only $\boldsymbol{w}^+$ has too little expressive power and converges to suboptimal solutions, which post-refinement cannot fix (see especially normal maps in (2)).

5, we need to go even further and optimize the noise vector $\boldsymbol{\xi}$ (instead of drawing it from multi-variate normal distributions). We note that optimizing for the noise component is even more important in MaterialGAN, compared to embedding faces in StyleGAN or StyleGAN2. We suspect that this is because with human faces, the distinction between large-scale features (e.g., eyes, noise, and mouth) and small-scale features (e.g., winkles) is very prominent, allowing the $\mathcal{W}^+$ space to focus mostly on the large-scale features while leaving the small-scale ones to the noise vector $\boldsymbol{\xi} \in \mathcal{N}$. In our case, the boundary between large-scale and small-scale material features is much less distinct. The physical scales of real-world materials varies in a continuous fashion, making it virtually impossible to assign them to only one of the $\mathcal{W}^+$ and $\mathcal{N}$ spaces. We hypothesize that for this reason, we need to focus on both $\mathcal{W}^+$ and $\mathcal{N}$ to achieve high-quality reconstruction of SVBRDF maps. Based on these empirical observations, estimating SVBRDF parameter maps from photographs using our pre-trained MaterialGAN boils down to solving the following optimization:

$$\boldsymbol{u}^* = \underset{\boldsymbol{w}^+ \in \mathcal{W}^+,\, \boldsymbol{\xi} \in \mathcal{N}}{\arg\min} \sum_{i=1}^{k} \mathcal{L}(\mathcal{R}(\mathcal{G}(\boldsymbol{w}^+, \boldsymbol{\xi}); L_i, C_i), \boldsymbol{I}_i). \tag{5}$$

Since there are two variables $\boldsymbol{w}^+$ and $\boldsymbol{\xi}$ that behave in a correlated fashion, a proper optimization strategy is crucial to achieve high-quality results. We now discuss our alternating two-step optimization method.

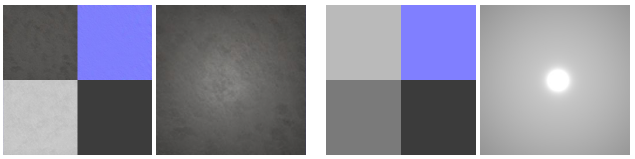
4.3 Optimization strategy

Abdal et al. [2019a; 2019b] recommended using a two-stage setting by first optimizing $\boldsymbol{w}^+$ (with $\boldsymbol{\xi}$ fixed) and then $\boldsymbol{\xi}$ (with $\boldsymbol{w}^+$ fixed). In our case, this approach does work in some cases but is not always the top-performing option. In addition to this strategy, we propose two alternatives, leading to three different optimization schemes:

(1) **Strategy 1**: Optimize $\boldsymbol{w}^+$ first, then optimize $\boldsymbol{\xi}$;
(2) **Strategy 2**: Jointly optimize both $\boldsymbol{w}^+$ and $\boldsymbol{\xi}$;
(3) **Strategy 3**: Alternatively optimize $\boldsymbol{w}^+$ and $\boldsymbol{\xi}$ for a small number (for example, 10) of iterations each.

Figure 6 shows a comparison of these strategies. All of them give reasonable results, but Strategy 1 is better suited for materials with strong large-scale features. Strategy 2 provides the fastest convergence because it allows the noise vector $\boldsymbol{\xi}$ to be modified from the very beginning. This, however, generally causes the optimization to use $\boldsymbol{\xi}$ for encoding higher-level features and is prone to overfitting. Finally, Strategy 3—a hybrid of Strategies 1 and 2—behaves in a more robust fashion than either of the previous strategies in most cases. We use Strategy 3 for all the results in our paper. Additionally, our experiments indicate that it is desirable to use different VGG layer weights for the optimization of $\boldsymbol{w}^+$ and $\boldsymbol{\xi}$. The weights we are using are, for $\boldsymbol{w}^+$: [1/512,1/512,1/128,1/64]; for $\boldsymbol{\xi}$: [1/64,1/64,1/256,1/512].

Noise optimization vs. post-refinement. Instead of optimizing latent space $\boldsymbol{w}^+$ with noise $\boldsymbol{\xi}$, another option is to apply post-refinement (that is, pixel-space optimization without any latent space) after optimizing $\boldsymbol{w}^+$ only. However, the space $\boldsymbol{w}^+$ is too small to realistically match per-pixel detail: if optimizing $\boldsymbol{w}^+$ only, the resulting maps have significant artifacts. Adding post-refinement to such a result



(a) Mean $\boldsymbol{w}$ initialization (b) Low roughness initialization

Fig. 8. **Visualization of our constant initializations.** We initialize our optimization with the two materials shown here and pick the result with the lowest final loss. (This applies in cases where we do not use the result from Deschaintre et al. as initialization, as detailed in the results section and supplementary materials.) Left: Material maps generated from the mean latent vector $\boldsymbol{w}$. Right: An additional low roughness, specular initialization.

essentially becomes per-pixel optimization (with little regularization), which tends to to work poorly with a small number of inputs. Optimizing $\boldsymbol{\xi}$ offers more powerful regularization, as the noise is inserted into all layers of the generator, rather than just appended at the end (like post-refinement). We show two failure examples in Figure 7, where optimizing $\boldsymbol{w}^+$ leads to unsatisfactory texture maps.

4.4 Initialization

We find that our method is robust to the initialization of the latent vectors. We experimented with using the same initial configuration—represented by the material produced by the mean $\boldsymbol{w}$ of our GAN training data (see Figure 8(a))—and found that is works well for most of the materials we tried (both synthetic and real). However, this initialization represents a material with a high roughness (reflecting a bias in our training data) and sometimes leads to errors when fitting highly specular / low roughness materials. Therefore, we add an additional low roughness initialization (see Figure 8(b)). In practice, given the captured images, we run our MaterialGAN optimization starting from both initializations and retain the result with the lowest optimization error of Eq. (3). All of our results in this paper followed this scheme.

5 RESULTS

Only a small subset of our results fits into the paper. Please see our supplemental material and video for more results.

Test data. For synthetic tests, we use several examples from the test set of Deschaintre [2018], as well as some from the Adobe Stock dataset [Li et al. 2018]. This gives a total of 39 synthetic results. For our real results, we use a hand-held mobile phone to capture images with flash, resulting in a collocated camera and point light illumination. Similar to previous work [Deschaintre et al. 2019; Hui et al. 2017], we use a paper frame to register the multiple images. We add markers to the frame to improve camera pose estimation. Using this process, we capture 28 physical samples with nine images per material, roughly covering the sample with 3×3 specular highlights. Unless otherwise specified, all our results use seven images for inverse-rendering optimizations and the remaining two (under novel lighting) for evaluating the results.

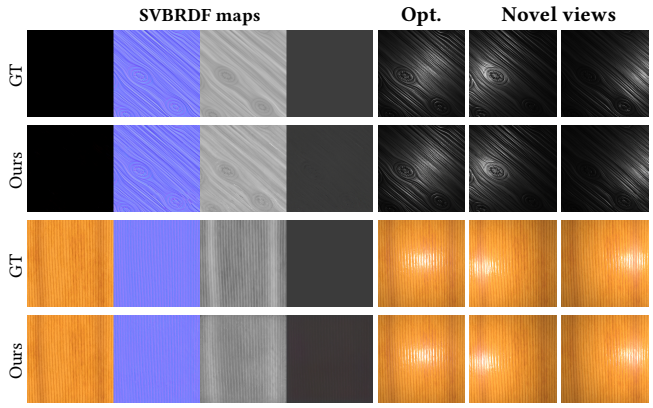


Fig. 9. **SVBRDF reconstruction on synthetic data.** We demonstrate results on synthetic SVBRDFs, one from [Deschaintre et al. 2019] (top) and one from the Adobe Stock Material dataset (bottom). We are able to accurately reconstruct these materials from 7 input images (one input shown). Many more synthetic results are available in supplementary materials.

Inverse-rendering performance. Our optimization takes about 2 minutes to complete 2000 iterations on a Titan RTX GPU. In many cases, the results converge after 500 iterations, but we use 2000 everywhere for simplicity.

Testing on synthetic data. Figure 9 contains two synthetic results using our method, showing a close match both in maps and in novel view renderings. For more results, please refer to supplemental materials. We note that all methods perform better on synthetic data than on real data, possibly because of the exact BRDF model match and perfect calibration, and also because the synthetic test set, while distinct from the training set, is relatively similar in style.

5.1 Comparison with prior work on real data

Here we compare our method and Gao et al. [2019]. For more results and comparisons, including with Deschaintre et al. [2019], and including with and without initialization for ours and Gao's method, please refer to supplemental materials. We show 10 real examples from our cell phone capture pipeline in Figure 10. Note that Gao's method is significantly dependent on initialization, while the same is not true for our method. Therefore, in this figure, we show Gao's result *with initialization* by Deschaintre et al. [2019], while our result is shown *without initialization*. Furthermore, note that we are initializing Gao's method with the 2019 multi-input method by Deschaintre, which is a better initialization than the 2018 single-input method. Thus the baseline we are comparing against is, strictly speaking, even higher than what is published in Gao et al., and combines the two best methods published at this time. Generally, we find that our method produces cleaner maps and is less prone to overfitting (burn-in) than Gao's, while producing more accurate re-renderings under original and novel lighting. Table 1 shows a quantitative evaluation of the re-rendering quality on novel lighting. As these novel views would be hard to match pixel-wise using any method, as they have never been observed, we use a perceptual

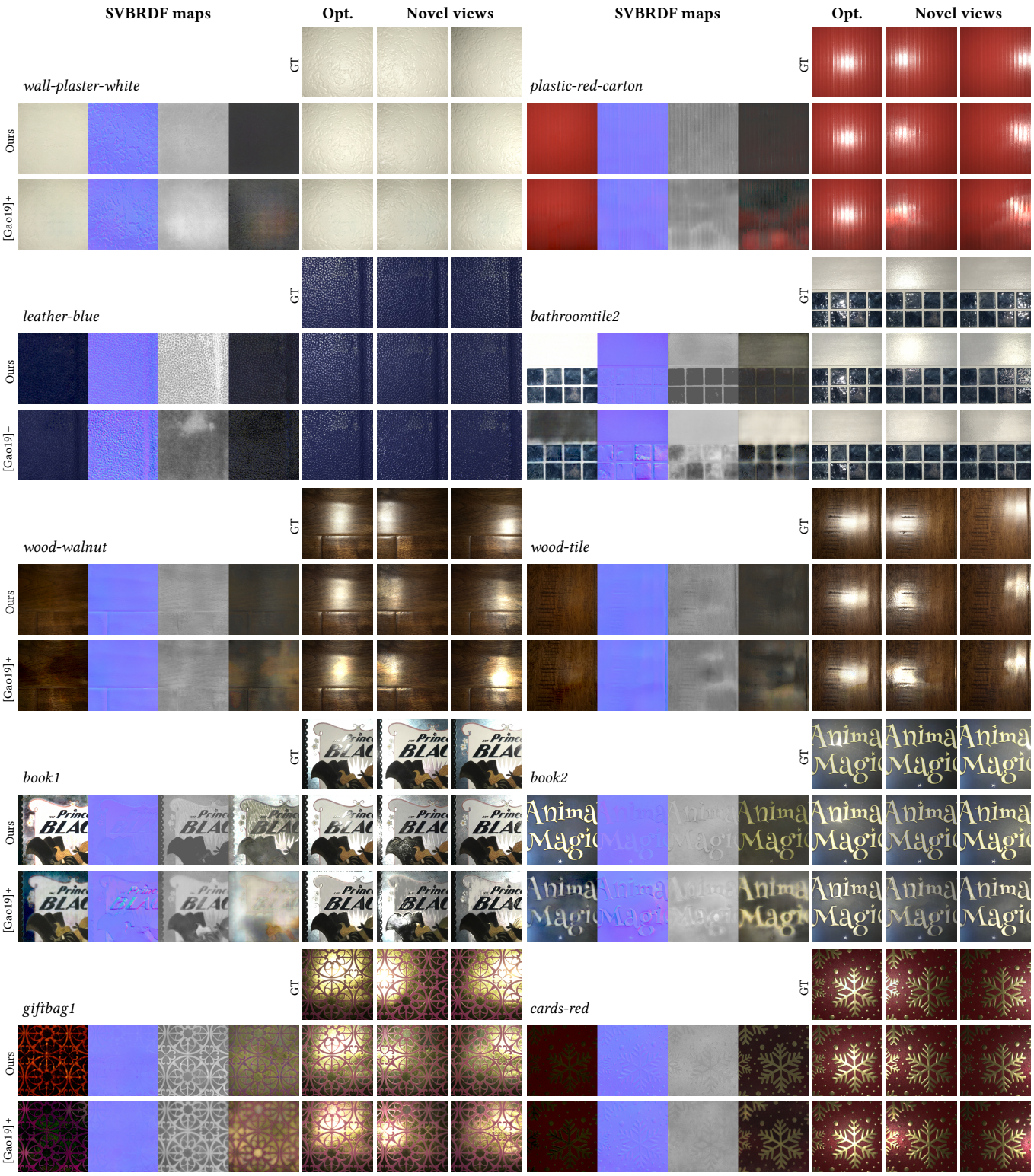


Fig. 10. **SVBRDF reconstruction on real data.** We reconstruct SVBRDF maps from 7 inputs, and compare the resulting maps and images rendered under 2 novel views. Gao's method [2019] initialized with Deschaintre's [2019] direct predictions (denoted as "[Gao19]+") tends to have complex reflectance burnt into the specular albedo map, leading to inaccurate predictions under novel views. Our method with simple initializations, in contrast, is less prone to such burn-ins and generally produces more accurate renderings under novel views. Please refer to Table 1 for more information on the quality of these renderings.

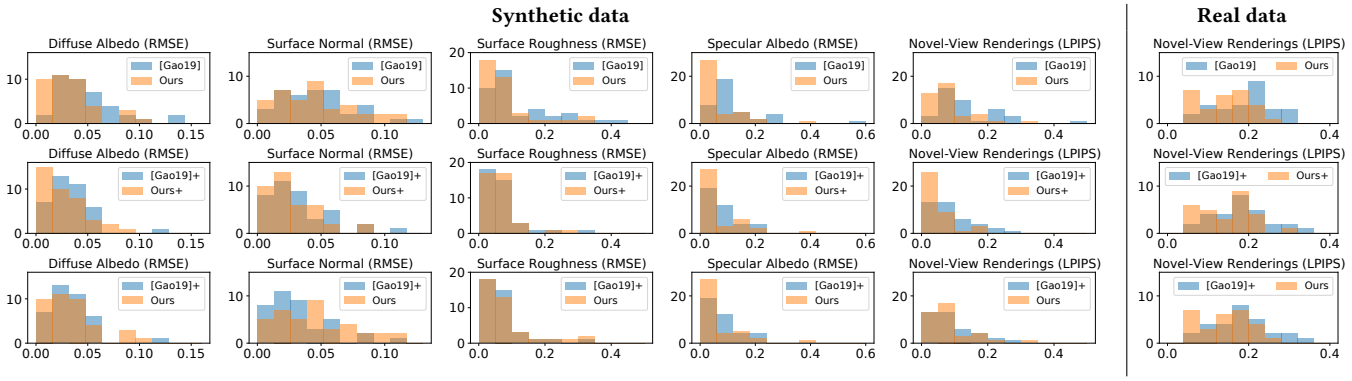


Fig. 11. **Performance statistics** of Gao [2019] and our method. For each technique, we compute (i) the Learned Perceptual Image Patch Similarity (LPIPS) metric between renderings of the output SVBRDF maps and the reference images for 28 *real* and 39 *synthetic* examples; and (ii) the root-measure-square error (RMSE) of the inferred maps for the *synthetic* examples. For both metrics, a lower score indicates a better accuracy. Using identical initializations, our technique ("Ours" and "Ours+") outperforms Gao's ("[Gao19]" and "[Gao19]+") consistently for both real and synthetic examples, as demonstrated in the top and the middle row. Furthermore, our technique with constant initializations ("Ours") has a similar performance with Gao's method initialized using Deschaintre's [2019] direct predictions ("[Gao19]+") on the synthetic examples and outperforms the latter on the real examples, as shown on the bottom.

Table 1. Accuracy of the novel-view renderings shown in Figure 10 measured using the Learned Perceptual Image Patch Similarity (LPIPS) metric where our method produces better predictions than Gao's [2019] in most cases.

Material	Ours	[Gao19]+	Material	Ours	[Gao19]+
wall-plaster-white	**0.071**	0.132	*plastic-red-carton*	**0.095**	0.166
leather-blue	**0.146**	0.356	*bathroomtile2*	**0.225**	0.231
wood-walnut	**0.226**	0.252	*wood-tile*	0.202	**0.192**
book1	**0.147**	0.318	*book2*	**0.042**	0.122
giftbag1	**0.183**	0.218	*cards-red*	**0.059**	0.092

method, specifically the Learned Perceptual Image Patch Similarity (LPIPS) metric [Zhang et al. 2018] (lower is better). Note that our method (without initialization by Deschaintre's method) produces better scores for novel views than Gao's method (with initialization) for most images; even in the case where our LPIPS score is worse, our maps still look more plausible overall. We also report quantitative evaluations (histograms) for our entire set of results (see Figure 11). For synthetic data, we compare the RMSE of all predicted maps (diffuse albedo, normal, roughness, specular albedo), as we do know the ground truth for them. For both synthetic and real data, we compare the LPIPS scores on novel lighting. We use a + sign to indicate initialization by Deschaintre et al. In the top row, we compare both methods without initialization by Deschaintre's method, while in the middle row, both methods are initialized, and in the bottom row, we compare our method without initialization to Gao's with initialization. Generally, we find that if both methods are initialized the same way, our method outperforms Gao's. Even in the last row, our performance is comparable on synthetic data (worse on normal map and better on diffuse/specular maps) and still better on real data overall.

Note about Deschaintre et al. We find that the results from [Deschaintre et al. 2019] have much less accurate re-rendering than either ours or Gao's method, as they are not doing any optimization to precisely fit the target images. The mismatches we observe are definitely not due to simple scaling or gamma correction issues, as that would be consistent across examples; rather, we find that the method performs much better on synthetic examples that match the visual style of its training set. On the other hand, their method is fast and results tend to be clean and artifact-free, so they are very suitable for initialization of optimization methods.

5.2 Additional comparisons

Optimization with different initializations. In Figure 12, we compare our method to Gao's with and without initialization by Deschaintre's method in all 4 combinations, on a synthetic and a real example. This shows that Gao's method more significantly dependent on good initialization that ours (even though our method can still occasionally benefit).

Post-refinement. In general, the quality of our maps is sufficient after using our MaterialGAN-based optimization. However, Gao's method introduced a post-refinement step, where the maps are further optimized without any latent space, and with at most minor regularization. Therefore, we also implement a similar post-refinement step. However, like good initialization, this post-refinement makes less of a difference in our method, and Gao's method is more dependent on it, as it produces significantly blurry maps without it. This is shown in Figure 13; note the difference in sharpness of the maps.

Optimization with different numbers of input images. While most of our results are shown with 7 inputs, using two additional inputs for novel lighting evaluation, our method does work with various numbers of input images. We show 3 synthetic examples in Figure 14, with different numbers of inputs from 1 to 25. All the three examples are the same as used in Gao's work. The errors of both reconstructed SVBRDF maps and novel-view renderings generally decrease with more input images, as is expected for an inverse-rendering method. In Figure 15, we compare real capture results

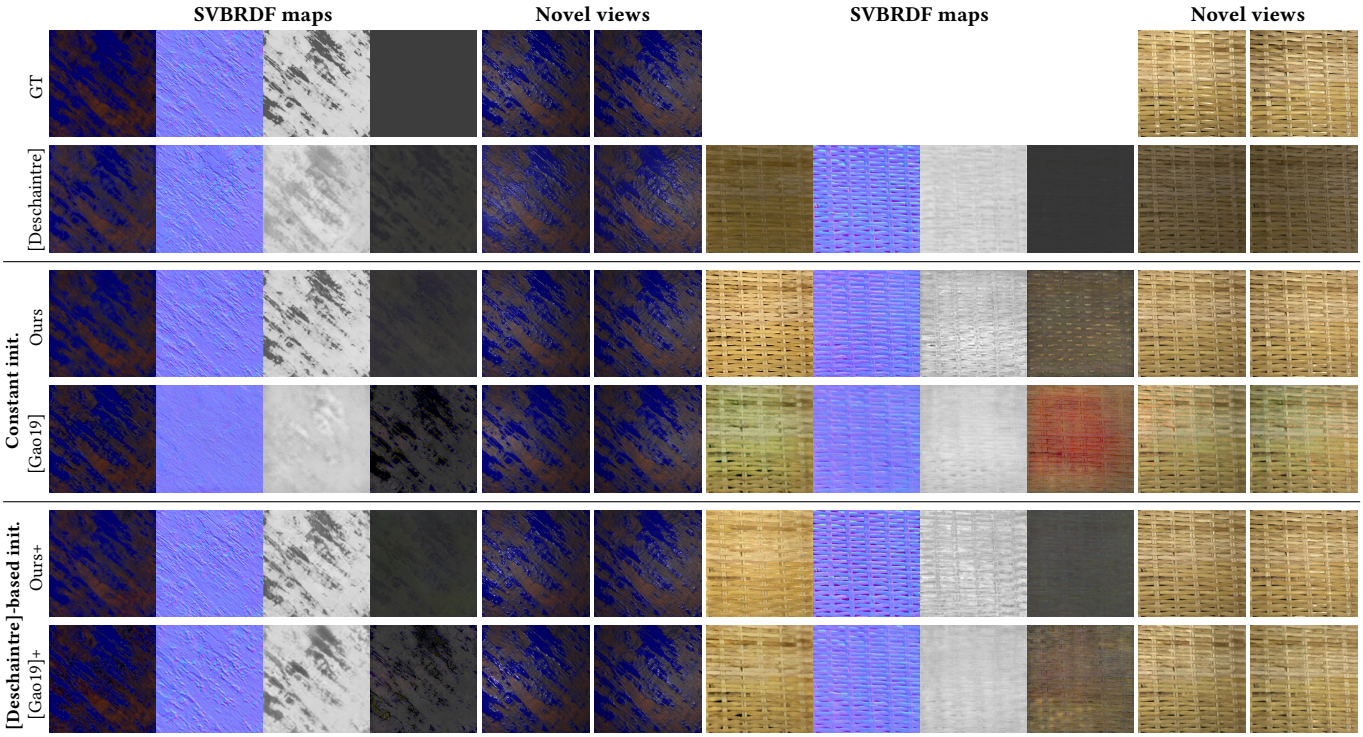


Fig. 12. **SVBRDF results with different initialization** Unlike Gao's method, ours is less strongly dependent on a good initialization from Deschaintre's method [2019]. In most of cases, starting from simple texture maps (given by our constant initializations) is already good enough to converge to a clean solution. We show all combinations (with and without good initializations) for both methods, for one synthetic and one real example, where techniques initialized with [Deschaintre] are denoted with the suffix "+" (i.e., "Ours+" and "[Gao19]+"). Note the failure of Gao's method without good initializations (i.e., "Gao19").

with 1, 3, and 7 inputs, with and without initialization by Deschaintre's method, and also include Gao's results for 3 and 7 inputs (with initialization). Our result remains plausible with 3 inputs, though artifacts do get reduced with more inputs. For all numbers of inputs, our result (with or without initialization) tends to be cleaner than Gao's.

Editing operations. An additional advantage of the StyleGAN-based latent space is the ability to achieve semantically meaningful operations such as morphing, by interpolating two or more parent latent codes to create a hybrid offspring material. Morphing in latent space often preserves semantic features qualitatively better than naive interpolation in pixel space. Figure 16 and the supplemental video show morphing of a few real materials using linear interpolation in latent space, compared to the corresponding naive interpolation (linear in pixel space).

6 CONCLUSION AND FUTURE WORK

Discussion and limitations. While we believe our framework improves upon the state of the art, there are also some limitations. Our current BRDF model is shared by previous work, but certain common effects (layering on book covers, subsurface fiber scattering in woods, anisotropy in fabrics) are not correctly captured by it. An extension of our generative model and rendering operator would be possible, though the key challenge is finding high-quality training data for these effects.

Our assumption of almost flat samples will fail for materials with strong relief patterns, and will produce blurring or ghosting if there are obvious parallax effects in the aligned captured images. Strong self-shadowing or inter-reflections are also not currently handled. Solving for height instead of normal, with a more advanced rendering operator, may be able to resolve parallax effects and to correctly predict (and undo) shadowing effects from strong height variations.

Furthermore, more precise calibration may improve our accuracycd. This would likely require knowledge of the cell phone hardware, and/or pre-calibration of its properties (e.g. flash light falloff, lens vignetting, and color processing properties).

Conclusion. We propose a novel method for acquiring SVBRDFs from a small number of input images, typically 3 to 7, captured using a hand-held mobile phone. We use an optimization framework that leverages a powerful material prior, based on a generative network, MaterialGAN, trained to synthesize plausible SVBRDFs. MaterialGAN learns correlations in SVBRDF parameters and provides local and global regularization to our optimization. This produces high-quality SVBRDFs that accurately reconstruct the input images, and

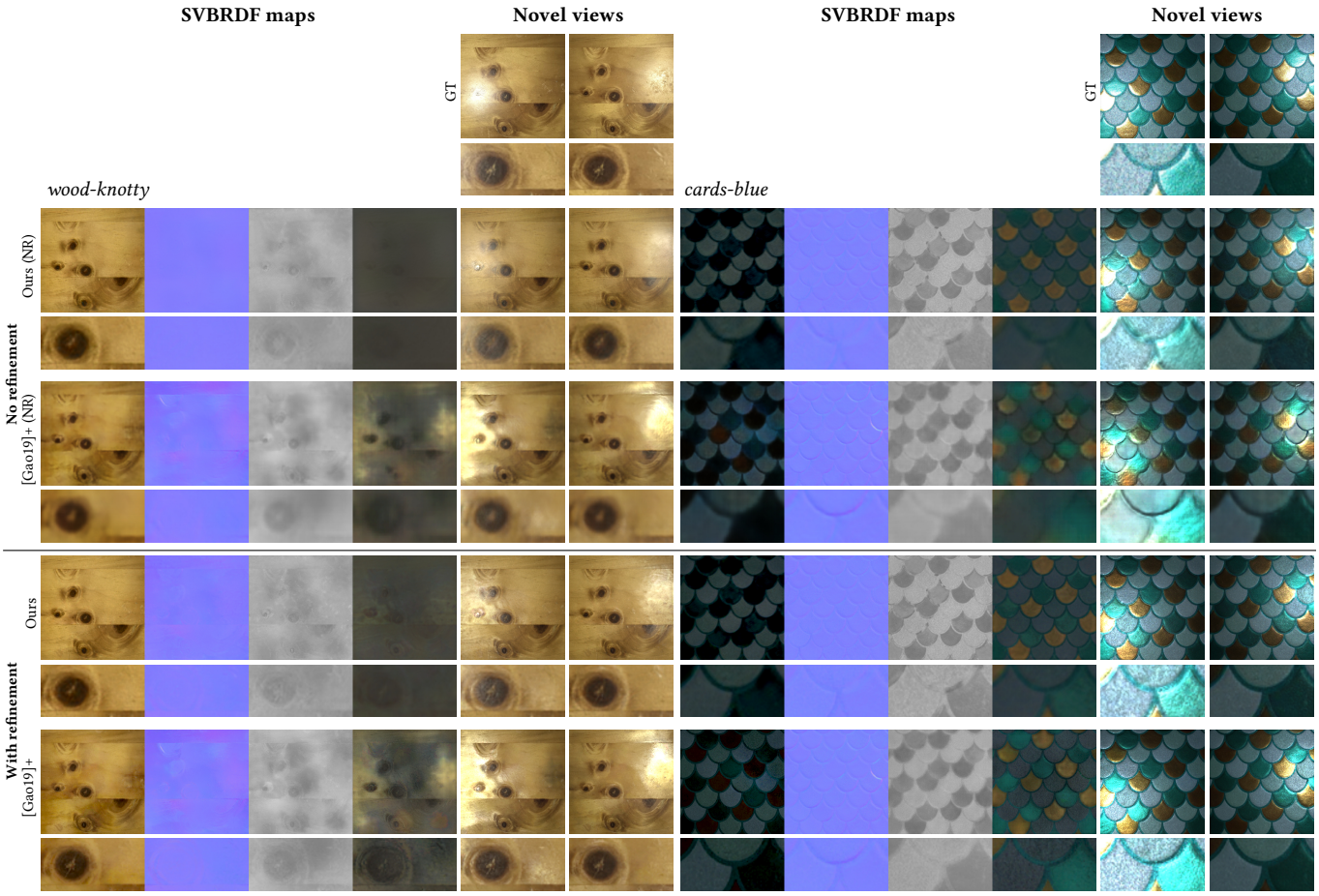


Fig. 13. **Per-pixel post-refinement.** Unlike Gao's method, post-refinement via per-pixel optimization makes less of a difference in our method. Without post-refinement, [Gao19]+ (i.e., Gao's method initialized with Deschaintre's [2019] direct predictions) usually produces blurry results, as shown in the row marked as "[Gao19]+ (NR)". Our method, on the contrary, does not rely nearly as heavily on post-refinement: Without it, our results are already quite sharp (see "Ours (NR)"), thanks to the generative power of our MaterialGAN. A zoomed-in version is attached below each SVBRDF map and novel-view image.

because of our MaterialGAN prior, lie on a plausible material manifold. As a result, our reconstructions generalize better to novel views and lighting than previous state-of-the-art methods.

We believe that our work is only a first step toward GAN-based material analysis and synthesis and our experiments suggest many avenues for further exploration including improving material latent spaces and optimization techniques using novel architectures and losses, learning disentangled and editable latent spaces, and expanding beyond our current isotropic BRDF model.

ACKNOWLEDGMENTS

This research was started during Yu Guo's internship at Adobe Research. We thank TJ Rhodes for help with material capture hardware setup. This work was supported in part by NSF IIS-1813553.

REFERENCES

Rameen Abdal, Yipeng Qin, and Peter Wonka. 2019a. Image2StyleGAN++: How to Edit the Embedded Images? arXiv:1911.11544

Rameen Abdal, Yipeng Qin, and Peter Wonka. 2019b. Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space? arXiv:1904.03189

Miika Aittala, Timo Aila, and Jaakko Lehtinen. 2016. Reflectance Modeling by Neural Texture Synthesis. *ACM Trans. Graph.* 35, 4 (2016), 65:1–65:13.

Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. 2013. Practical SVBRDF Capture in the Frequency Domain. *ACM Trans. Graph.* 32, 4 (2013), 110:1–110:12.

Miika Aittala, Tim Weyrich, and Jaakko Lehtinen. 2015. Two-shot SVBRDF Capture for Stationary Materials. *ACM Trans. Graph.* 34, 4 (2015), 110:1–110:13.

Muhammad Asim, Ali Ahmed, and Paul Hand. 2019. Invertible generative models for inverse problems: mitigating representation error and dataset bias. *arXiv preprint arXiv:1905.11672* (2019).

Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G. Dimakis. 2017. Compressed Sensing using Generative Models *(Proceedings of Machine Learning Research)*, Vol. 70. 537–546.

Valentin Deschaintre, Miika Aittala, Fredo 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, Frédo Durand, George Drettakis, and Adrien Bousseau. 2019. Flexible SVBRDF Capture with a Multi-Image Deep Network. *Computer Graphics Forum* 38, 4 (2019).

Chris Donahue, Julian McAuley, and Miller Puckette. 2018. Synthesizing Audio with Generative Adversarial Networks. *CoRR* abs/1802.04208 (2018). arXiv:1802.04208

Yue Dong. 2019. Deep appearance modeling: A survey. *Visual Informatics* 3, 2 (2019), 59–68.

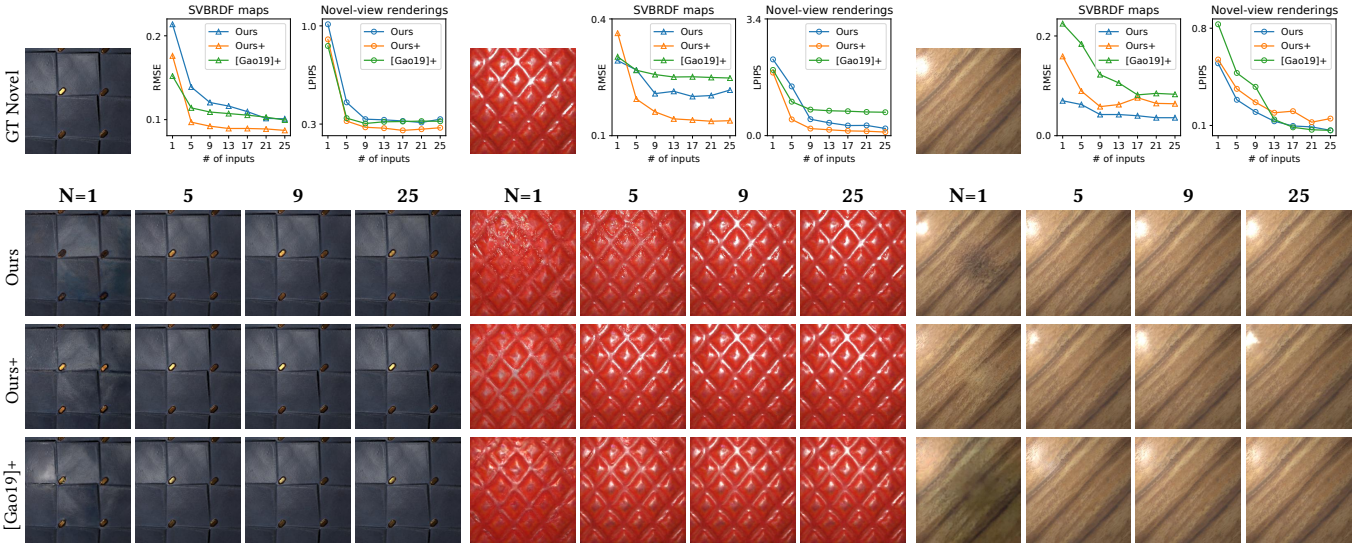


Fig. 14. **Performance using different numbers of input images (synthetic data).** The quality of recovered SVBRDF maps, as demonstrated by the plots, generally improves with more input images for both our and Gao's [2019] methods. Our method with constant (Ours) and neural (Ours+) initializations are comparable or better than Gao's ([Gao19]+) with neural initialization [Deschaintre et al. 2019] . For a highly specular material shown on the right, although the LPIPS metric computed using renderings under 5 novel views of our results is similar to that of Gao's, ours better preserve the specular highlight. For each material, all the renderings including the references (GT Novel) are generated using one of the 5 novel views.

Yannick Francken, Tom Cuypers, Tom Mertens, and Philippe Bekaert. 2009. Gloss and Normal Map Acquisition of Mesostructures Using Gray Codes. In *Advances in Visual Computing*, Vol. 5876. Springer, 788–798.

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).

Andrew Gardner, Chris Tchou, Tim Hawkins, and Paul Debevec. 2003. Linear Light Source Reflectometry. *ACM Trans. Graph.* 22, 3 (2003), 749–758.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2015. A Neural Algorithm of Artistic Style. arXiv:1508.06576

L. A. Gatys, A. S. Ecker, and M. Bethge. 2016. Image Style Transfer Using Convolutional Neural Networks. In *CVPR 2016*. 2414–2423.

Abhijeet Ghosh, Tongbo Chen, Pieter Peers, Cyrus A. Wilson, and Paul Debevec. 2009. Estimating Specular Roughness and Anisotropy from Second Order Spherical Gradient Illumination. In *EGSR 2009*. 1161–1170.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014a. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27*. 2672–2680.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014b. 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).

Xun Huang and Serge Belongie. 2017. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. In *ICCV 2017*.

Zhuo Hui, Kalyan Sunkavalli, Joon-Young Lee, Sunil Hadap, Jian Wang, and Aswin C. Sankaranarayanan. 2017. Reflectance Capture Using Univariate Sampling of BRDFs. In *ICCV 2017*.

Justin Johnson, Alexandre Alahi, and Fei-Fei Li. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *ECCV 2016*.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018a. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *ICLR 2018*.

Tero Karras, Samuli Laine, and Timo Aila. 2018b. A Style-Based Generator Architecture for Generative Adversarial Networks. arXiv:1812.04948

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

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.

Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. 2019. Synthesizing 3d shapes from silhouette image collections using multi-projection generative adversarial networks. In *CVPR 2019*. 5535–5544.

Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. 2018. Materials for Masses: SVBRDF Acquisition with a Single Mobile Phone Image. In *ECCV 2018*, Vol. 11207. 74–90.

Stephen R. Marschner, Stephen H. Westin, Eric P. F. Lafortune, Kenneth E. Torrance, and Donald P. Greenberg. 1999. Image-Based BRDF Measurement Including Human Skin. In *Eurographics Workshop on Rendering*.

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

Addy Ngan, Frédo Durand, and Wojciech Matusik. 2005. Experimental Analysis of BRDF Models. In *EGSR 2005*. 117–226.

Daniel O'Malley, John K Golden, and Velimir V Vesselinov. 2019. Learning to regularize with a variational autoencoder for hydrologic inverse analysis. *arXiv preprint arXiv:1906.02401* (2019).

Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *CoRR* abs/1511.06434 (2015). arXiv:1511.06434

Peiran Ren, Jiaping Wang, John Snyder, Xin Tong, and Baining Guo. 2011. Pocket Reflectometry. *ACM Trans. Graph.* 30, 4 (2011), 45:1–45:10.

Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR 2015*.

Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. 2018. MoCoGAN: Decomposing Motion and Content for Video Generation. In *CVPR 2018*.

Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. 2007. Microfacet Models for Refraction Through Rough Surfaces. *EGSR 2007* (2007), 195–206.

Tim Weyrich, Jason Lawrence, Hendrik PA Lensch, Szymon Rusinkiewicz, and Todd Zickler. 2009. *Principles of appearance acquisition and representation*. Now Publishers Inc.

Zexiang Xu, Jannik Boll Nielsen, Jiyang Yu, Henrik Wann Jensen, and Ravi Ramamoorthi. 2016. Minimal BRDF Sampling for Two-Shot near-Field Reflectance Acquisition. *ACM Trans. Graph.* 35, 6 (2016), 188:1–188:12.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *CoRR* abs/1801.03924 (2018).

Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. 2016. Generative Visual Manipulation on the Natural Image Manifold. arXiv:1609.03552

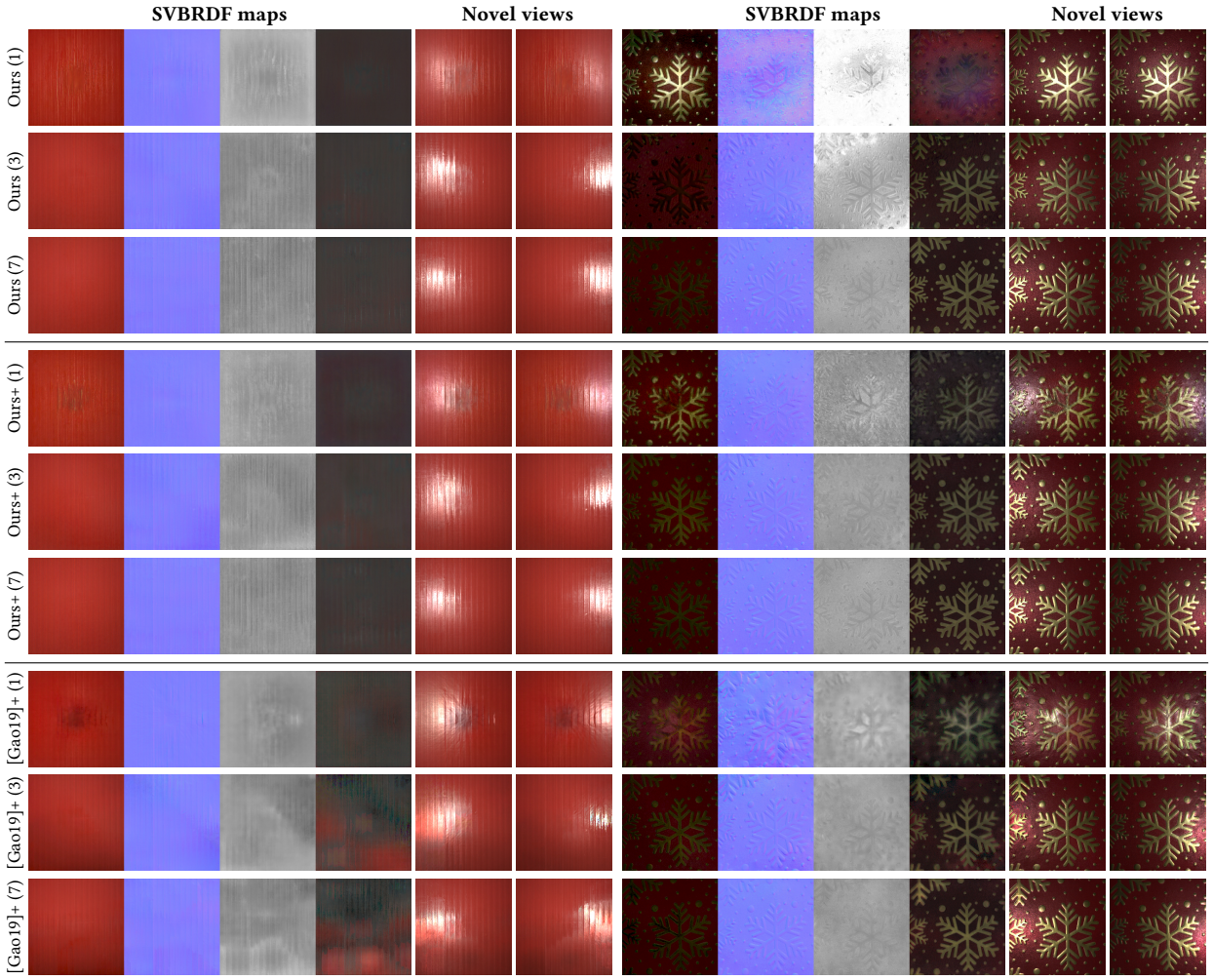


Fig. 15. **Performance using different numbers of input images (real data).** The quality of SVBRDF maps recovered by our method generally improves with more input images under both constant initialization (see "Ours") and Deschaintre [2019] initialization (see "Ours+").

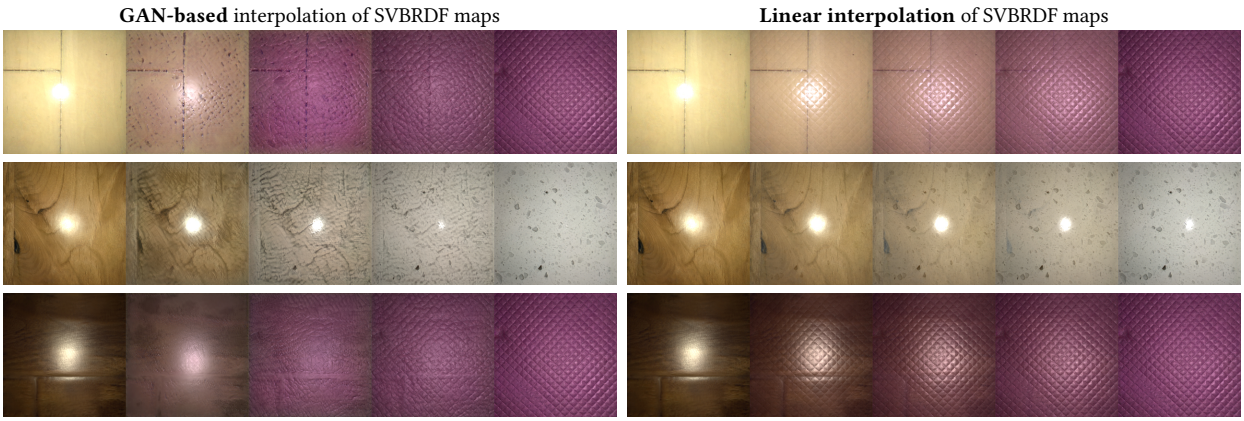


Fig. 16. **Material interpolation.** Renderings of interpolations between two SVBRDFs recovered from real images using our method. Results on the left and right columns are obtained, respectively, using our GAN latent space and naïve linear interpolation.