MatFuse: Controllable Material Generation with Diffusion Models

Giuseppe Vecchio* Renato Sortino* Simone Palazzo Concetto Spampinato
giuseppe.vecchio@phd.unict.it renato.sortino@phd.unict.it
University of Catania

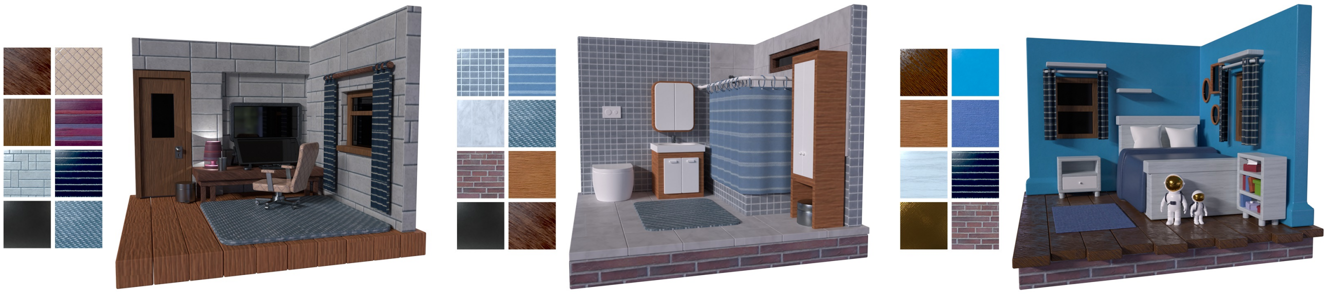

Figure 1. **Sample scenes textured using materials generated with MatFuse.** For each of the three scenes we show the materials used and the final rendering.

Abstract

*Creating high-quality materials in computer graphics is a challenging and time-consuming task, which requires great expertise. To simplify this process, we introduce **MatFuse**, a unified approach that harnesses the generative power of diffusion models for creation and editing of 3D materials. Our method integrates multiple sources of conditioning, including color palettes, sketches, text, and pictures, enhancing creative possibilities and granting fine-grained control over material synthesis. Additionally, MatFuse enables map-level material editing capabilities through latent manipulation by means of a multi-encoder compression model which learns a disentangled latent representation for each map. We demonstrate the effectiveness of MatFuse under multiple conditioning settings and explore the potential of material editing. Finally, we assess the quality of the generated materials both quantitatively in terms of CLIP-IQA and FID scores and qualitatively by conducting a user study.*
Source code for training MatFuse and supplemental materials are publicly available at https://gvecchio.com/matfuse.

1. Introduction

Materials are central in computer graphics, playing a pivotal role in achieving high-quality, realistic digital imagery. As the computational power of professional and consumer hardware has increased, high-quality CGI has experienced a growing demand, fueled by the expanding field of application of 3D models, from game engines to architectural and industrial prototyping and simulation [38, 44, 49]. However, the creation of high-quality materials remains a challenging and time-consuming process, which requires complex tools and high expertise. Following the promising results achieved by Generative Adversarial Networks (GANs) [13] for the generation of natural images, several works have successfully employed adversarial training to generate high-quality materials [18, 19, 23, 54]. These approaches, however, provide a limited degree of control over material synthesis. Additionally, GANs are generally hard to train, due to the inherent instability of their adversarial training, leading to mode collapse and limited variability.

Recently, diffusion models (DMs) have set a new state-of-the-art in image generation [9, 22, 39], overcoming the training limitations of GANs. Furthermore, diffusion models can be easily conditioned during the "denoising" process, with *global* or *local* conditions, respectively controlling the overall appearance of the image (e.g., text prompt), or specific regions of the output image (e.g., sketches). Recent approaches like Composer [24], propose to combine multiple sources of conditioning, both global and local, by

[*] Both authors contributed equally to this research.

considering an image as the sum of its independent components [30]. This approach expands the control space, giving designers the degree of control required to finely guide the generation.

We design **MatFuse** to improve material synthesis using the generative capabilities of diffusion models and exploiting the image compositionality approach to combine multiple conditioning sources in a single model. Following Rombach et al. [39], the proposed model consists of a VQ-GAN [11], trained to learn a bidirectional mapping between the pixel space and the latent space, and a diffusion model, trained to generate a latent representation of a material starting from noise and one or more optional conditions.

We evaluate the effectiveness of our approach when being conditioned with both a single or multiple conditions. We also test our model for material editing through what we define as *volumetric inpainting*, by partially or totally masking, single SVBRDF maps for a given material, and letting the model reconstruct the missing parts. The results show the potential of MatFuse in generating a wide range of diverse and realistic materials, as well as in adapting to several combinations of conditioning inputs.

In summary, the contributions of this work are:

- We present MatFuse, a unified, multi-conditional method leveraging the generation capabilities of diffusion models to tackle the task of high-quality material synthesis as a set of SVBRDF maps.
- We propose a multi-encoder extension to the autoencoder by Rombach et al. [39], using 4 different encoders, to learn map-specific latent spaces and add a rendering loss to its training.
- We demonstrate the generation capabilities and flexibility of MatFuse, through different conditioning mechanisms, which allow for an unprecedented level of control for material generation.
- We show the ability to use MatFuse for material editing purposes through "volumetric inpainting", to generate single portions of input materials or entire maps.

2. Related Work

Controllable material generation. Materials synthesis is a challenging task in computer graphics [14], with many recent data-driven approaches focusing on the task of estimating SVBRDF maps from an input image [1, 3, 7, 8, 12, 17, 31, 32, 34, 48, 53].

Controllable generation of materials, in contrast, remains a relatively underexplored task. Guehl et al. [15] propose an approach consisting of a procedural structure synthesis step, followed by data-driven color synthesis to propagate existing material properties to the generated structure. MaterialGAN [18] proposes a generative network based on StyleGAN2 [28], trained to synthesize realistic SVBRDF parameter maps. This approach exploits the properties of the latent space learned by StlyleGAN2 to generate material maps that match the appearance of the captured images when rendered. Hu et al. [23] extend the capabilities of MaterialGAN with the generation of novel materials, by transferring the micro- and meso-structure of a texture to a set of input material maps. However, both MaterialGAN [18] and Hu et al. [23] rely on alterations of pre-existing material inputs and lack any generation capabilities. Recently, Zhou et al. [54] proposed TileGen, a generative model for SVBRDFs capable of producing tileable materials, optionally conditioned through an input structure pattern. However, its generation capabilities are strongly limited to class-specific training. He et al. [19] proposed Text2Mat, an architecture based on diffusion models for text-to-material generation. However, controlling the generation of materials remains a challenging task. To fill this gap, MatFuse leverages diffusion models to provide full control and flexibility over the generation process by ingesting multiple conditions to guide the diffusion process.

Generative models. Image generation is a long-standing challenge in computer vision due to the high dimensionality of images and the difficulty in modeling complex data distributions. Generative Adversarial Networks (GAN) [13] enabled the generation of high-quality images [4, 26, 27] but are characterized by unstable convergence at training time [2, 16, 35], due to the adversarial training, and are unable to fully model complex data distributions [36], often exhibiting mode collapse behavior.

Recently, Diffusion Models (DMs) [22, 42] have emerged as an alternative to GANs, achieving state-of-the-art results in image generation tasks [9], besides showing a more stable training behavior. However, optimizing these models tends to be expensive in terms of training times and computational costs. To address these limitations, Rombach et al. [39] propose to apply the diffusion process to a smaller, and less computationally demanding, latent space, perceptually equivalent to the pixel space. This shift to the latent space reduces computational requirements, without altering generation quality, and enables a whole new classifier-free conditioning mechanism [21] through cross-attention between latent image representations and conditioning data. More recently, Composer [24] showed how it is possible to combine multiple semantically different conditions to control diffusion models.

Building on these advancements, MatFuse a) introduces a multi-encoder VQ-GAN to account for individual SVBRDF map peculiarities and integrates a rendering loss [7] to enforce coherence and consistency in the output results; b) extends the LDM conditioning mechanism to include multiple modalities in a compositional way; c) enables inpainting both at spatial and map level, by exploiting the multi-encoder approach, providing users fine control over the generation process.

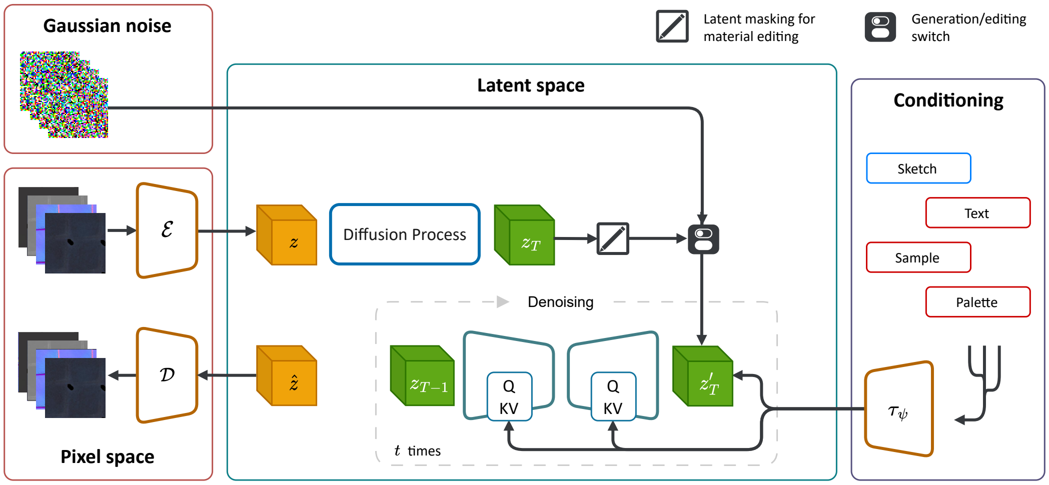

Figure 2. **Overview of the MatFuse framework**: At training time, VQ-GAN encoder $\mathcal{E}$ projects data from the pixel space to a more compact latent embedding z; the diffusion process runs on this latent space; conditioning is carried out through cross-attention for global conditions (red in figure), and through concatenation with the noise for local conditions (blue in figure); the output maps are finally obtained by projecting the conditioned reconstructed latent space $\hat{z}$ back into the pixel space through VQ-GAN decoder $\mathcal{D}$.

3. MatFuse Architecture

Motivated by the lack of a unified model capable of accepting different sources of control for material generation, we propose MatFuse, a conditional generative model that produces high-quality pixel-level reflectance properties for arbitrary materials, while simultaneously combining multiple conditions. To this end, we leverage the compositionality of images, by deconstructing them into primitives, such as color palettes, sketches, etc. which can then be combined to guide the generation [24].

Inspired by the LDM [39] architecture, MatFuse consists of two main components: 1) a compression network that projects data from the pixel space $\mathcal{X}$ to the latent space $\mathcal{Z}$ and vice-versa, and 2) a diffusion model, which learns the distribution of the latent feature vectors to enable the generation of new samples.

The general architecture of MatFuse is shown in Fig. 2. In the following, we introduce and describe each module of the proposed framework.

3.1. Latent Diffusion Model

Map Compression. We use a multi-encoder VQ-GAN [11] to learn a map-specific latent representation. This allows the model to extract disentangled features from each map, which will be concatenated in the latent space and combined by the diffusion model via self-attention. The architecture is illustrated in Fig. 3.

Given a set of N maps $x = \{\mathbf{M}^1, \mathbf{M}^2, \dots, \mathbf{M}^N\}$ and encoders $\mathcal{E} = \{\mathcal{E}^1, \mathcal{E}^2, \dots, \mathcal{E}^N\}$, each map $\mathbf{M}^i \in \mathbb{R}^{H \times W \times 3}$ is encoded into a latent representation $z^i = \mathcal{E}^i(\mathbf{M}^i)$, where $z^i \in \mathbb{R}^{h \times w \times c^i}$, $i \in \{1, \dots, N\}$, and c^i is the number of channels of each encoded map. The N latent representations (one per input map) are then concatenated along the channel dimension, obtaining $z = concat\left(z^1, z^2, \dots, z^N\right)$, and then fed into the decoder $\mathcal{D}$ that reconstructs the set of input maps $\hat{x} = \left\{\hat{\mathbf{M}}^1, \hat{\mathbf{M}}^2, \dots, \hat{\mathbf{M}}^N\right\}$. Here, $\hat{x} = \mathcal{D}(z)$, $z \in \mathbb{R}^{h \times w \times c}$, where c is the number of channels of the concatenated maps, i.e., $c = \sum_i c^i$. Before decoding the feature vectors, we regularize the latent space by learning a representative *codebook* for each map, which is then used to quantize the latent vectors z_i before concatenation.

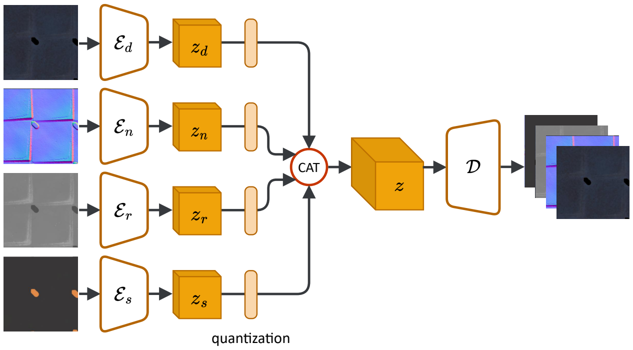

Figure 3. **Overview of the compression model architecture**. Reflectance maps (diffuse, normal, roughness, and specular) are fed to the encoders. Features extracted for each map are quantized and concatenated before being passed to the decoder, which reconstructs the original maps.

Following the work of Rombach et al. [39], we train the encoder $\mathcal{E}$ using a combination of pixel-space L_2 loss $\mathcal{L}_{\text{pixel}}$, a perceptual LPIPS loss $\mathcal{L}_{\text{perc}}$ [51], a patch-based adversarial objective $\mathcal{L}_{\text{adv}}$ [10, 11, 25], and a codebook commitment loss $\mathcal{L}_{\text{comm}}$ [46]. To improve the reconstruction of material map details in the latent space, we add a rendering loss $\mathcal{L}_{\text{render}}$ [7], computed as the MSE between ground-truth renders and prediction renders, to the VQ-GAN training, enforcing coherence and consistency between the individual maps.

Diffusion Model. After learning a latent space that efficiently encodes information from multiple maps, we train a diffusion model [22] to estimate the prior distribution of the latent vectors to synthesize real samples. We follow the architecture proposed in Rombach et al. [39], which consists of a U-Net [40] with self-attention between residual blocks operating on the latent representation z of the input maps, rather than on the pixel space.

In particular, the diffusion network ϵ_θ is trained to estimate at each step the noise added in the forward diffusion process and subtracts it from the noisy latent to obtain the denoised data. We optimize the diffusion model with an $\mathcal{L}_{\text{diff}}$ objective between the estimated noise and the noise added in the forward diffusion process, as in Ho et al. [22].

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{t,z_0,\epsilon}\left[\|\epsilon_t - \epsilon_\theta\left(z_t, t\right)\|^2\right] \tag{1}$$

Here, ϵ_t is the noise added at the timestep t in the forward diffusion process, while $\epsilon_\theta\left(z_t, t\right)$ is the noise estimated by the U-Net model at time t.

The trained model allows for generating new samples by denoising the noise vector sampled from a normal distribution into a valid latent space point.

3.2. Conditioning Mechanisms

MatFuse allows for controlling the generation process via two types of conditioning information: 1) global conditioning for a high-level control via text or visual prompts, as well as color palettes, and; 2) local conditioning, for a fine-grained localized control, via sketches.

Global conditioning. It enables control over the generation via high-level prompts, descriptive of the global material appearance. Conditions are embedded in a one-dimensional feature vector, which is provided to the diffusion model through multi-head cross-attention [47] at each denoising step between the flattened noise tensor z and the conditioning vector (i.e., QKV in Fig. 2).

MatFuse can be globally conditioned via text and image prompts, as well as color palettes. Image and text embeddings are computed using a pre-trained CLIP [37] model as a feature extractor. The color palette embedding is computed by counting color occurrences in an input image, clustering those within a certain CIE76 distance threshold, and selecting the top 5 most prevalent colors. Finally, these values are projected into a 1D vector through an MLP, which is optimized in conjunction with the diffusion model.

Local conditioning. Local conditions are used to achieve control over the generation structure. These conditions are first projected into a low-resolution representation to match the dimensionality of the latent vector z using a small convolutional network which is trained jointly with the U-Net. The resulting embedding is concatenated to the noisy latent z. We identify sketches as a relevant local condition for materials, giving control over the represented pattern. We extract sketches from the material render, under a diffuse light, using a Canny edge detector [5].

Multimodal fusion. We enable multimodal composable generation in MatFuse by extending the classifier-free guidance training strategy [21]. This strategy allows not only to combine different types of conditioning but also to generate quality output regardless of the number of conditions provided, thus allowing compositionality. In particular, during training, we randomly drop each condition with a probability of 50% and drop all conditions with a 10% probability.

With conditioning, the training objective becomes

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{t,z_0,\epsilon}\left[\|\epsilon_t - \epsilon_\theta\left(z_t, t, \tau(y)\right)\|^2\right] \tag{2}$$

3.3. Material Editing via Volumetric Inpainting

The use of a multi-encoder architecture, as described in Sec. 3.1, allows the model to learn a disentangled latent representation of each material map, hardly achievable using a single encoder, by encoding each map separately. This latent representation allows us to manipulate specific parts of the latent space, knowing which material property they encode, thus enabling an unprecedented level of material editing capabilities. We propose a novel "volumetric inpainting" [1] approach by jointly masking portions of the noise tensor in both spatial and channel dimensions. Formally, given a latent representation z of the material maps, encoded through $\mathcal{E}$, we compute the latent tensor at the denoising step t as $z \odot m + z_{t+1} \odot (1 - m)$, with m being the volumetric binary mask.

By masking portions or channels corresponding to a specific map, we can force the model to generate only that particular map. This is particularly useful for incomplete materials, missing some maps, or where the properties of one map are not satisfying. In combination with traditional inpainting, it is possible to generate only specific areas of the material for all maps or for a reduced set only.

The application of "volumetric inpainting" for material editing is demonstrated in Sec. 4.6.

4. Experimental results

In this section, we first introduce the datasets employed in our work: the synthetic dataset by Deschaintre et al. [7],

[1] The name was chosen to highlight that masking occurs in both the spatial and channel (encoding material property information) dimensions independently.

and a new procedurally-generated synthetic dataset created for the task at hand. Then, we evaluate the accuracy of our approach on two different training setups: 1) with a single condition, providing examples for each condition individually, and 2) with multiple conditions combined together. We compare our approach against TileGen [54] and evaluate its performance in terms of CLIP-IQA [50] and Fréchet Inception Distance (FID) [20] to provide a quantitative measure. We also conduct a user study to evaluate the user preference when comparing the two methods. In addition, we demonstrate the material editing capabilities of MatFuse, thus confirming the advantage introduced by a multi-encoder architecture. Finally, we ablate the proposed architectural components, in particular the multi-encoder compression model, and losses to assess their contribution.

4.1. Datasets

We employ the SVBRDF dataset introduced by [7], which is based on the Allegorithmic Substance Share collection[2]. The entire dataset includes about 20,000 blended materials represented with the *diffuse*, *normal*, *specular*, and *roughness* maps. We use the training/test splits introduced by Deschaintre et al. [7].

We extended the dataset with 320 materials collected from the PolyHaven[3] library. As a form of data augmentation for this dataset, we extract crops at different scales from each material at the 4K resolution. For each material, we collect 1 full-scale (4K resolution) crop, 4 crops at half-scale (2K resolution), 16 crops at a quarter of scale (1K resolution), 64 crops at one eight of resolution (512), and 256 crops at one-sixteenth of resolution (256). The full dataset consists of 431 crops for each material and a total number of 140,508 crops. Each crop is then rescaled to a resolution of 256×256 pixels.

The combination of the two datasets sums up to about 160K materials. We render each material under different lighting conditions using five different environment maps, each rotated four times (0°, 90°, 180°, 270°) to light the material from different angles. For each material crop, we produce 20 renders, for a total of 3.2 million renders.

4.2. Training Procedure

The compression model is optimized via a combination of supervised and adversarial training. It is trained with mini-batch gradient descent, using the Adam [29] optimizer and a batch size of 4. The learning rate is set to 10^{-4} and the training is carried out for 4,000,000 iterations. $\mathcal{L}_{\text{adv}}$ is enabled after 300,000 iterations, when the compression model starts to learn low-frequency components of maps, thus allowing the compression model to recover the high-frequency ones. For our encoders we choose a downsampling factor of $f = 8$, giving the best compromise between efficiency and image quality, as shown in [39].

The diffusion model is trained for $500,000$ iterations with a batch size of 20 using an AdamW [33] optimizer, with a learning rate value of 10^{-4}, with a linear learning rate warm-up starting from 10^{-6}. We used a linear schedule for β and denoise using the DDIM (Denoising Diffusion Implicit Models) [43] sampling schedule at inference time with $T = 50$ steps.

4.3. Generation Results

We evaluate our model under different conditioning settings to understand its capability of combining multiple sources of information while producing high-quality material maps. In particular, we explore single-conditional and multi-conditional generation. Unconditional samples and additional conditional generation examples are included in the supplemental material.

4.3.1 Single-conditional generation

We first assess the generation capabilities of MatFuse when controlled with a single input condition.

Global conditioning. MatFuse can be globally conditioned via text prompts, image prompts, and a color palette. Fig. 4 shows that our approach successfully captures the condition features in the generated materials. It is worth noting that the use of adjectives like "shiny" in the text prompt alters the visual appearance accordingly, for example by making the material less rough (Fig. 4, second row). Similarly, visual features from image prompts, like the highlight in the stone tiles, are correctly captured, as it is clearly visible in the resulting rendering. Finally, colors from the color palette are accurately reproduced in the diffuse component of the generated material.

Local conditioning. MatFuse can be locally conditioned through pattern sketches as demonstrated in Fig. 5. The model can correctly process hand-drawn, clean, sketches as well as noisy sketches generated using a Canny edge detector. The model correctly transfers the structure defined in the sketch to the produced material by acting on its normal map, providing it with the desired geometry.

4.3.2 Multi-conditional generation

The main strength of MatFuse resides in the possibility of combining multiple conditions for a finer generation control. In particular, combining a local and a global condition gives control over both the geometry and the visual features of the material. Fig. 6 shows the materials generated when combining the sketch with each of the global conditions. We can see that the model is able to accurately follow the spatial structure given by the sketch while showing the semantics of the global conditions.

[2] https://share.substance3d.com/
[3] https://polyhaven.com/textures/

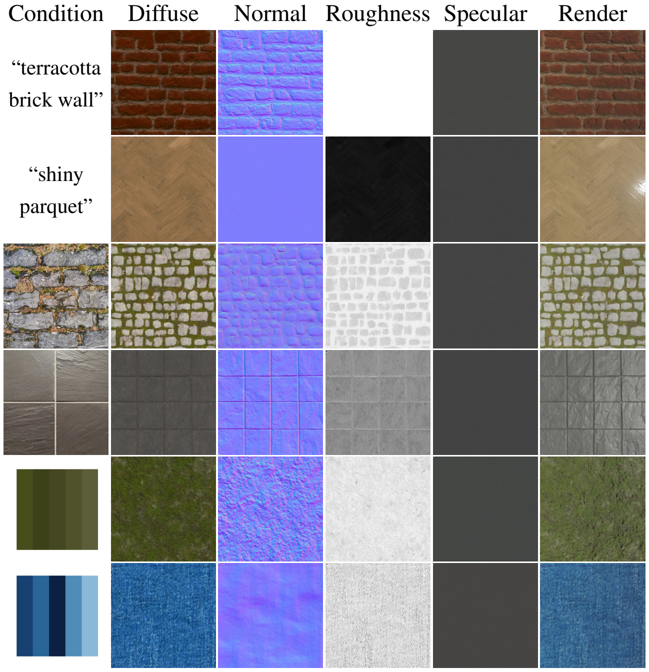

Figure 4. **Globally conditioned material generation**. We evaluate MatFuse when guided with single conditions. First two rows: text-conditioned map generation; mid two rows: image-prompted generation, yielding maps with features of the input image; last two rows: palette-conditioned generation.

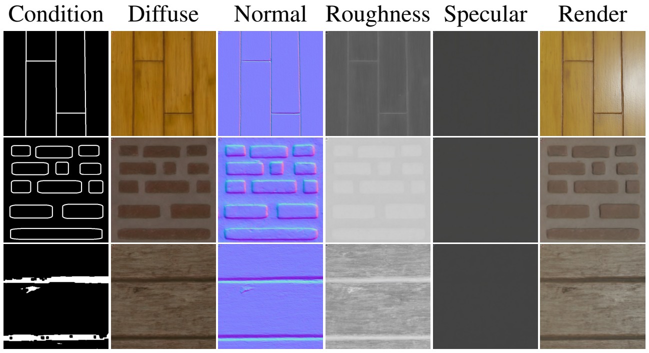

Figure 5. **Locally conditioned material generation**. We provide sketches to condition MatFuse and produce maps with well-defined edges. The first two rows present hand-drawn sketches, while the latter is obtained from a material picture. This shows the robustness of MatFuse in handling both clean and noisy sketches.

4.4. Quantitative Evaluation

Assessing material generation quality is a challenging task, due to their inherently different data distribution compared to natural images. Established metrics such as the Fréchet Inception Distance (FID) [20] or the Inception Score [41] rely on the InceptionV3 [45] architecture pre-trained on ImageNet [6], which includes natural images.

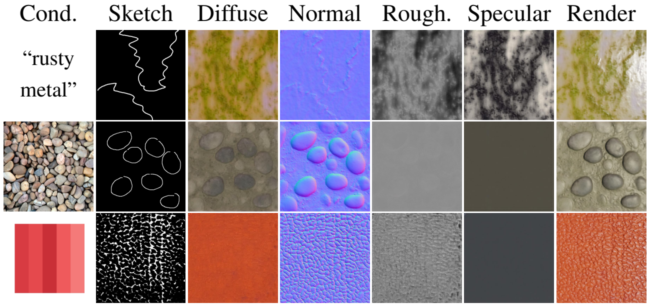

Figure 6. **Multimodal conditioned material generation**. First row: text prompt + sketch. Second row: image prompt + sketch. Third row: color palette + sketch.

To evaluate our results we employ the CLIP-IQA, recently introduced by Wang et al. [50]. This approach uses contrastive prompts to determine the appearance of an image. In particular, we evaluated the quality of the generated materials using the “high-quality/low-quality” contrastive pair. We believe that this metric is more suitable for the task at hand due to the much wider range of data used to train the CLIP model [37].

To this aim, we render 2,000 unconditionally generated materials at 512×512 resolution, and measure both the CLIP-IQA and FID on these renders. For reference, we also compute the CLIP-IQA on the ground truth renders from our dataset, as well on samples generated by vanilla LDM [39] and TileGen [54][4]. Tab. 1 reports the results of this analysis. On the CLIP-IQA metric, MatFuse improves significantly over the baseline model and produces scores close to the target upper bound, i.e., ground truth samples from the employed dataset. Notably, MatFuse is almost on par with TileGen [54], although the latter is trained on a more restricted set of classes. FID scores confirm a higher similarity between MatFuse's samples and ground-truth images, compared to the baseline. It is interesting to note that, in this metric, MatFuse significantly outperforms TileGen too, which is likely due to TileGen's limited sample diversity, captured by the FID score.

4.5. Qualitative Comparison and User Study

We further evaluated MatFuse performances by conducting a pairwise comparison study between MatFuse and TileGen [54]. The study involved 100 MS/PhD students in computer science who were asked to choose their preferred materials based on both realism and rendering quality. We presented them with 25 randomly selected material pairs, drawn from a larger pool of 100 samples from the “leather”, “wood”, “marble”, “stone” and “ceramic” classes. These classes were already available in TileGen and were used for

[4] As the model has not been publicly released, a batch of samples generated by TileGen was kindly provided by the authors.

Model	CLIP-IQA (↑)	FID (↓)
Ground Truth	0.471 ± 0.191	-
TileGen [54]	0.433 ± 0.161	184.81
LDM (baseline)	0.269 ± 0.118	231.64
MatFuse	$\mathbf{0.431} \pm 0.151$	158.53

Table 1. **Performance of MatFuse in terms of CLIP-IQA**. The CLIP-IQA values for the datasets used during the training serve as our upper bound. We compare the generation quality to TileGen [54] and a baseline Stable Diffusion model trained to generate materials unconditionally. The CLIP-IQA metric is computed using the "high-quality/low-quality" contrastive pair.

conditional generation in MatFuse. Visual samples from the user study are presented in Figure 7. Our method employs global conditioning by embedding class names as a condition for generation.

In our test, MatFuse received a higher number of votes (MatFuse=1078, TileGen=949, No Pref.=473), and a chi-square test establishes statistical significance in the user's preference for MatFuse over TileGen (χ^2 =16.41, $p < 0.05$). It is important to note that both our method and TileGen generate results based solely on the specified class, leading to different appearances. Although MatFuse is not specifically trained for a semantic class as opposed to TileGen, it is capable of producing high-quality materials with fine detail and a realistic appearance.

4.6. Material Editing Results

We demonstrate here the editing capabilities of MatFuse which are made possible by the use of a multi-encoder architecture. In particular, the known structure in the latent space enables a deeper level of control over the generation by editing only specific maps or portions thereof through volumetric inpainting. Fig. 8 shows the application of this technique to different use cases and its combination with multimodal conditioning. Volumetric inpainting finds its most relevant application in generating missing maps for incomplete materials, as shown in the first and last rows of Fig. 8. Results show the method's ability to be coherent with the provided map structure (e.g., normal map in the last example) while capturing the semantics of the condition into the final material.

4.7. Ablation Study

We perform an ablation study to substantiate our architectural design and training strategy choices evaluating the contribution of the multi-encoder architecture and of the rendering loss. Qualitative results for the ablation study are included in the supplemental materials.

Results in Tab. 2 show the performance gain of the multi-encoder architecture compared to the baseline. This approach allows for better capturing map-specific features and efficiently compressing their information, resulting in a lower reconstruction distance between input and output. To further substantiate our claim we explore different codebook size configurations for the baseline, ranging from 4096 codes to 16384 codes. Besides the lower reconstruction error, separate map representations give better control over the latent space and enable advanced material editing techniques.

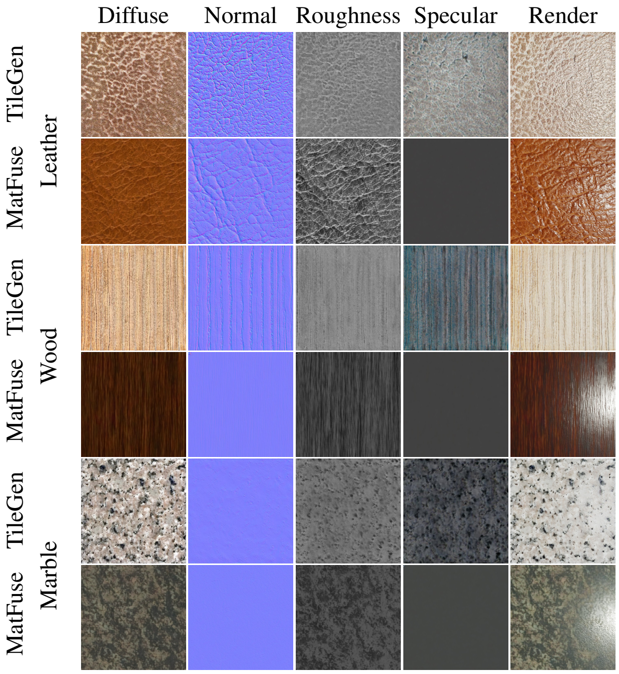

Figure 7. **Comparison to TileGen [54].** We compare MatFuse to TileGen models trained on three categories (Leather, Wood, Marble), conditioning our model with the category name. Additional samples are provided in the supplemental materials.

We evaluate the contribution of the rendering loss $\mathcal{L}_{\text{render}}$ [7], when training the VQ-GAN. Our baseline network is trained using the loss proposed in [39]. Results in Tab. 3 demonstrate that the introduction of the rendering loss improves the reconstruction quality by enforcing consistency in the rendered material.

5. Limitations and Future Work

The generative capabilities of diffusion models come at the cost of computational resources. Although MatFuse is not explicitly limited to a specific resolution, 512×512 generation takes ~18 GB of GPU memory, with 768×768 requiring slightly less than ~24 GB. Such memory consumption indeed limits the scalability of MatFuse to higher resolutions and, consequently, the representation of fine details,

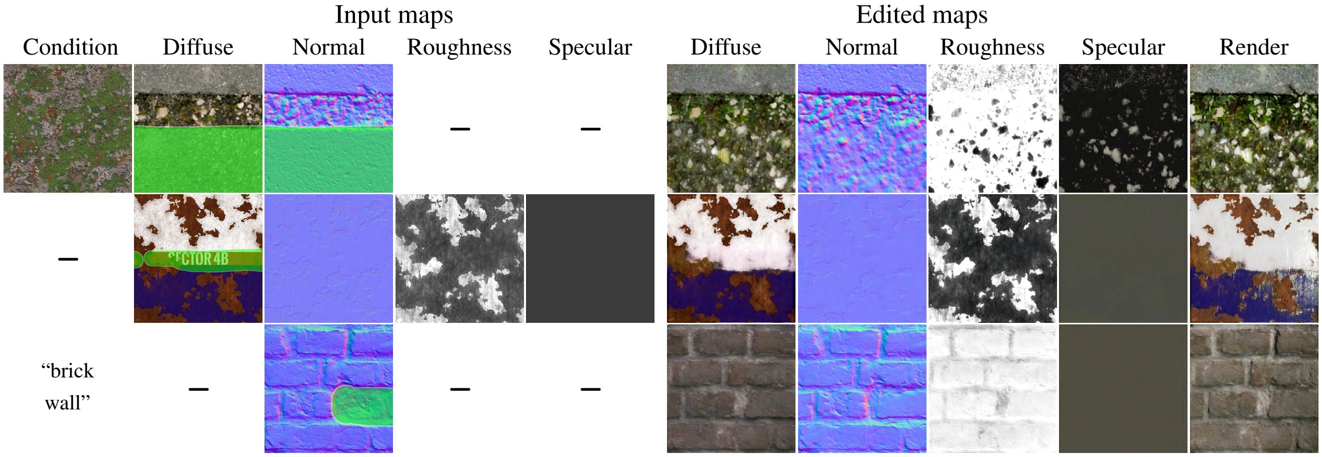

Figure 8. **Material editing with inpainting**. The results show the flexibility of MatFuse by being able to edit materials by inpainting, while using a condition (when provided) to determine the content of the generated part. The masked areas are highlighted in green, while fully masked maps are replaced with the ‘–’ symbol. Additional samples are provided in the supplemental materials.

Architecture	**Diff.**	**Nrm.**	**Rgh.**	**Spec.**	**Rend.**
Base (4096)	0.057	0.061	0.114	0.166	0.267
Base (8192)	0.049	0.052	0.098	0.144	0.233
Base (16384)	0.047	0.051	0.102	0.152	0.227
Multi Enc.	**0.016**	**0.024**	**0.022**	**0.020**	**0.041**

Table 2. **Ablation study of architectural components**. Performance is measured in terms of RMSE between predicted and ground-truth maps. We report the codebook size between brackets for the *“Base”* single encoder architecture.

Losses	**Diff.**	**Nrm.**	**Rgh.**	**Spec.**	**Rend.**
$\sum \mathcal{L}$ from [11]	0.038	0.030	0.047	0.033	0.064
$+ \mathcal{L}_{\text{render}}$	**0.016**	**0.024**	**0.022**	**0.020**	**0.041**

Table 3. **Ablation study of the contribution of the rendering loss when training the VQ-GAN**. Performance is measured in terms of RMSE between predicted and ground-truth maps.

particularly for textures with high-frequency patterns. A potential solution could leverage a patch-based approach to alleviate computational burdens and enhance the model’s applicability to higher resolutions. Moreover, a noted limitation in the current implementation of MatFuse is the lack of tileability in the generated materials. This restricts the seamless use of synthesized textures for large surfaces. Additionally, it would be possible to use the generative capabilities of MatFuse to perform SVBRDF estimation from a single image by providing a picture as an additional local condition. This could be done in combination with a more advanced form of local conditioning such as ControlNet [52].

6. Conclusion

In this paper, we present MatFuse, a learning approach for the generation of materials in the form of reflectance maps with diffusion models. The proposed approach specifically leverages the generative capabilities of recent diffusion methods to produce high-quality SVBRDF maps, supporting conditional and unconditional synthesis.

Inspired by the compositionality paradigm, MatFuse supports extensive multimodal conditioning, thus providing control over the generation process. In particular, MatFuse generates novel materials starting from a simple sketch, material samples, or textual descriptions, and supports conditioning combinations, e.g., sketch + color palette. Additionally, MatFuse introduces a novel “volumetric inpainting” strategy to perform map-level material editing. To do so, we propose a multi-encoder VQ-VAE, which learns a disentangled latent representation for each map.

MatFuse can be a promising solution to build upon for material generation tasks, by extending the conditioning mechanism to include additional input modalities (e.g., semantic segmentation) and output controls (e.g., enforcing tileability), as well as exploring methodologies and architectures to support higher resolution with limited resources.

7. Acknowledgments

Research at the University of Catania is supported by the project Future Artificial Intelligence Research (FAIR) – PNRR MUR Cod. PE0000013 - CUP: E63C22001940006. We thank Valentin Deschaintre for providing generation samples for the comparison with TileGen [54].

References

[1] Miika Aittala, Timo Aila, and Jaakko Lehtinen. Reflectance modeling by neural texture synthesis. *ACM Transactions on Graphics (ToG)*, 35(4):1–13, 2016. 2

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. 2

[3] Sai Bi, Zexiang Xu, Kalyan Sunkavalli, David Kriegman, and Ravi Ramamoorthi. Deep 3d capture: Geometry and reflectance from sparse multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5960–5969, 2020. 2

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 2

[5] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986. 4

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 6

[7] Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, and Adrien Bousseau. Single-image svbrdf capture with a rendering-aware deep network. *ACM Transactions on Graphics (ToG)*, 37(4):1–15, 2018. 2, 4, 5, 7

[8] Valentin Deschaintre, Miika Aittala, Frédo Durand, George Drettakis, and Adrien Bousseau. Flexible SVBRDF capture with a multi-image deep network. In *Computer Graphics Forum*, pages 1–13. Wiley Online Library, 2019. 2

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 1, 2

[10] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *Advances in neural information processing systems*, 29, 2016. 4

[11] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 2, 3, 4, 8

[12] Duan Gao, Xiao Li, Yue Dong, Pieter Peers, Kun Xu, and Xin Tong. Deep inverse rendering for high-resolution svbrdf estimation from an arbitrary number of images. *ACM Trans. Graph.*, 38(4):134–1, 2019. 2

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. 1, 2

[14] Darya Guarnera, Giuseppe Claudio Guarnera, Abhijeet Ghosh, Cornelia Denk, and Mashhuda Glencross. Brdf representation and acquisition. In *Computer Graphics Forum*, pages 625–650. Wiley Online Library, 2016. 2

[15] Pascal Guehl, Rémi Allegre, J-M Dischler, Bedrich Benes, and Eric Galin. Semi-procedural textures using point process texture basis functions. In *Computer Graphics Forum*, pages 159–171. Wiley Online Library, 2020. 2

[16] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017. 2

[17] Jie Guo, Shuichang Lai, Chengzhi Tao, Yuelong Cai, Lei Wang, Yanwen Guo, and Ling-Qi Yan. Highlight-aware two-stream network for single-image svbrdf acquisition. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021. 2

[18] Yu Guo, Cameron Smith, Miloš Hašan, Kalyan Sunkavalli, and Shuang Zhao. MaterialGAN: reflectance capture using a generative svbrdf model. *arXiv preprint arXiv:2010.00114*, 2020. 1, 2

[19] Zhen He, Jie Guo, Yan Zhang, Qinghao Tu, Mufan Chen, Yanwen Guo, Pengyu Wang, and Wei Dai. Text2Mat: Generating Materials from Text. In *Pacific Graphics Short Papers and Posters*. The Eurographics Association, 2023. 1, 2

[20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017. 5, 6

[21] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2, 4

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1, 2, 4

[23] Yiwei Hu, Miloš Hašan, Paul Guerrero, Holly Rushmeier, and Valentin Deschaintre. Controlling material appearance by examples. In *Computer Graphics Forum*, pages 117–128. Wiley Online Library, 2022. 1, 2

[24] Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. Composer: Creative and controllable image synthesis with composable conditions. *arXiv preprint arXiv:2302.09778*, 2023. 1, 2, 3

[25] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 4

[26] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2

[27] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 2

[28] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 2

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[30] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253, 2017. 2

[31] Xiao Li, Yue Dong, Pieter Peers, and Xin Tong. Modeling surface appearance from a single photograph using self-augmented convolutional neural networks. *ACM Transactions on Graphics (ToG)*, 36(4):1–11, 2017. 2

[32] Zhengqin Li, Kalyan Sunkavalli, and Manmohan Chandraker. Materials for masses: Svbrdf acquisition with a single mobile phone image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 72–87, 2018. 2

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[34] Rosalie Martin, Arthur Roullier, Romain Rouffet, Adrien Kaiser, and Tamy Boubekeur. Materia: Single image high-resolution material capture in the wild. In *Computer Graphics Forum*, pages 163–177. Wiley Online Library, 2022. 2

[35] Lars Mescheder. On the convergence properties of gan training. *arXiv preprint arXiv:1801.04406*, 1:16, 2018. 2

[36] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016. 2

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 4, 6

[38] Alexander Raistrick, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, Alejandro Newell, Hei Law, Ankit Goyal, Kaiyu Yang, and Jia Deng. Infinite photorealistic worlds using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12630–12641, 2023. 1

[39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 1, 2, 3, 4, 5, 6, 7

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[41] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 6

[42] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 2

[43] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 5

[44] Zhihang Song, Zimin He, Xingyu Li, Qiming Ma, Ruibo Ming, Zhiqi Mao, Huaxin Pei, Lihui Peng, Jianming Hu, Danya Yao, and Yi Zhang. Synthetic datasets for autonomous driving: A survey, 2023. 1

[45] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. corr abs/1409.4842 (2014), 2014. 6

[46] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 4

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[48] Giuseppe Vecchio, Simone Palazzo, and Concetto Spampinato. Surfacenet: Adversarial svbrdf estimation from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12840–12848, 2021. 2

[49] Giuseppe Vecchio, Simone Palazzo, Dario C Guastella, Ignacio Carlucho, Stefano V Albrecht, Giovanni Muscato, and Concetto Spampinato. Midgard: A simulation platform for autonomous navigation in unstructured environments. *arXiv preprint arXiv:2205.08389*, 2022. 1

[50] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2555–2563, 2023. 5, 6

[51] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4791–4800, 2021. 4

[52] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 8

[53] Xilong Zhou and Nima Khademi Kalantari. Adversarial single-image svbrdf estimation with hybrid training. In *Computer Graphics Forum*, pages 315–325. Wiley Online Library, 2021. 2

[54] Xilong Zhou, Milos Hasan, Valentin Deschaintre, Paul Guerrero, Kalyan Sunkavalli, and Nima Khademi Kalantari. TileGen: Tileable, controllable material generation and capture. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 1, 2, 5, 6, 7, 8