







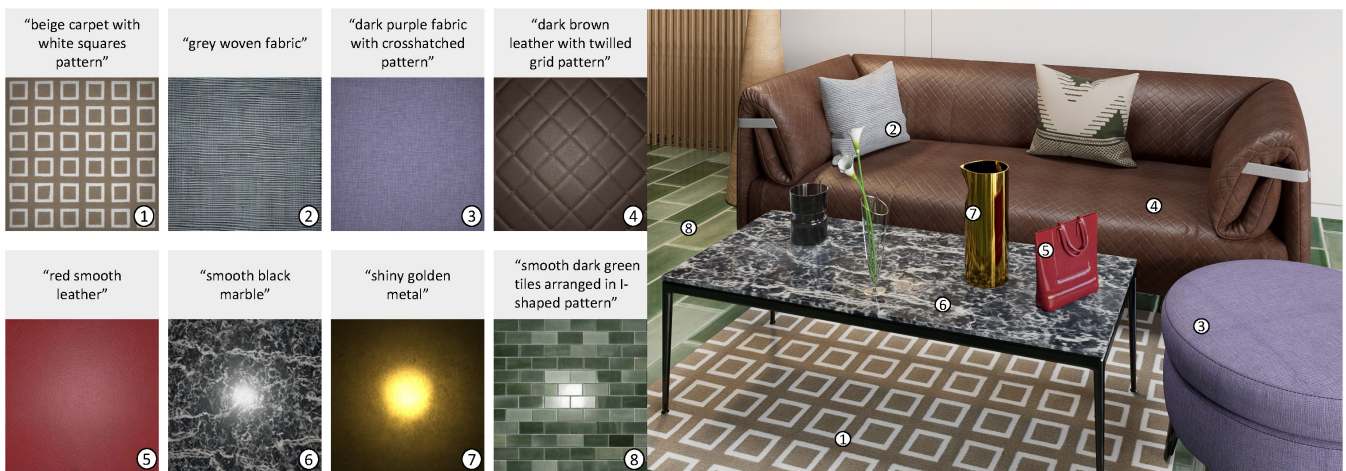


# Text2Mat: Generating Materials from Text

Zhen He<sup>1</sup>  Jie Guo<sup>1†</sup>  Yan Zhang<sup>1†</sup>  Qinghao Tu<sup>1</sup>  Mufan Chen<sup>1</sup>  Yanwen Guo<sup>1</sup>  Pengyu Wang<sup>2</sup>  and Wei Dai<sup>2</sup> 

<sup>1</sup>Nanjing University, State Key Lab for Novel Software Technology, China

<sup>2</sup>Dimension 5 Techs, China



**Figure 1:** We present Text2Mat, a text-based material generation framework which can generate complex materials with only input texts. Here, we show 8 materials generated by our proposed Text2Mat (left), and apply them to an indoor scene (right).

## Abstract

Specific materials are often associated with a certain type of objects in the real world. They simulate the way the surface of the object interacting with light and are named after that type of object. We observe that the text labels of materials contain advanced semantic information, which can be used as a guidance to assist the generation of specific materials. Based on that, we propose Text2Mat, a text-guided material generation framework. To meet the demand of material generation based on text descriptions, we construct a large set of PBR materials with specific text labels. Each material contains detailed text descriptions that match the visual appearance of the material. Furthermore, for the sake of controlling the texture and spatial layout of generated materials through text, we introduce texture attribute labels and extra attributes describing regular materials. Using this dataset, we train a specific neural network adapted from Stable Diffusion to achieve text-based material generation. Extensive experiments and rendering effects demonstrate that Text2Mat can generate materials with spatial layout and texture styles highly corresponding to text descriptions.

## CCS Concepts

• Computing methodologies → Rendering;

## 1. Introduction

Surface materials are often modeled by Spatially-Varying Bidirectional Reflectance Distribution Function (SVBRDF) [Nic65] and saved as parametric maps as digital assets. Some websites of digital asset are dedicated to providing material data designed by artists

† Corresponding authors: Jie Guo, guojie@nju.edu.cn; Yan Zhang, zhangyannju@nju.edu.cn

© 2023 The Authors.  
Proceedings published by Eurographics - The European Association for Computer Graphics.  
This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

through relevant compositing tools, but these materials are often limited in number and mostly consist of maps that cannot be edited or modified. In recent years, diffusion probabilistic models (DMs) [HJA20; SWMG15] have been applied with great success to text-to-image (text2image) generation tasks, such as Glide [NDR\*21], Stable Diffusion (SD) [RBL\*22] and other text2image generation methods. We have observed that the material labels consisting of texts are a powerful description of the material’s essential properties for both real-world materials captured by professional equipment and synthetic materials designed by artistic tools, providing a kind of high-level semantic information for guidance.

From the above observation, we aim to generate materials through a text-to-image (text2image) framework, which depends on data of images with matched text descriptions. Unfortunately, there is no such a public dataset of materials for use. To drive the task for generating materials with specified texts, we construct a dataset of PBR materials with detailed text descriptions. For better control of the texture of generated materials, we leverage the texture labels of DTD [CMK\*14] and some common texture styles of specific materials, such as tiles, paving stones, etc. Based on the dataset and SD, we propose Text2Mat, a text-conditional generation framework which can generate highly matched material maps according to specified colors, texture styles and material types by user texts.

In summary, we make the following contributions:

- A PBR material dataset of more than 2500 samples is collected, comprising detailed text descriptions of colors, texture styles, and material types, as well as corresponding SVBRDF maps ranging from 1K to 4K.
- A texture annotation method of material maps with the labels of DTD is proposed, which can present the detailed visual properties of the corresponding material.
- A text to material generation framework based on Stable Diffusion is designed, which can generate various kinds of materials with specified colors and texture styles only by texts.

## 2. Related Work

### 2.1. Material Acquisition and Generation

The traditional approach for material generation is to capture through hardware devices. Dong et al. [DWT\*10] and Kang et al. [KCW\*18] proposed to use specialized hardware to capture material properties. Although the captured material properties match those in the real world, these works often require expensive equipment, and the capture process is time-consuming and labor-intensive, resulting in high costs and poor scalability.

Advances in deep learning have enabled some works [LDPT17; DAD\*18; LSC18] to predict the corresponding SVBRDF from a single image, which is typically an image taken with a cell phone under flash. Under the condition of using only a single image taken by a cell phone, Guo et al. [GLT\*21] modified the architecture of the neural network and trained it using a GAN-based approach. Zhou and Kalantari [ZK21] performed mixed data enhancement by GAN, which allowed them to reconstruct reasonable and smooth results. Henzler et al. [HDMR21] recover the corre-

sponding SVBRDF by sampling from a prior distribution, supervising the rendered image according to the SVBRDF with the target image, center-flashed image, on a pixel-wised basis through differentiable rendering, and matching the appearance of the target image by depth optimization to obtain the corresponding SVBRDF. However, since their method only performs pixel-wised supervision on the rendered image, the parameters of their predicted SVBRDF often do not match the real ones, and it cannot recover material properties with regular textures well. Zhou et al. [ZHD\*22] used a variant StyleGAN2[KLA\*20] as a backbone for optimization as well. They can specify the pattern of generated material conditioned on a binarized image and achieve tileable results by modifying the padding way of the network architecture. But their method suffers from diversity and more control due to that each trained model can only generate a single type of materials.

The works of Hu et al. [HDR19] and Shi et al. [SLH\*20] are based on the node graph of procedural material synthesis tools. These works are based on existing material node maps in the library and can generate SVBRDF parameter maps with unlimited resolution and seamlessness. However, these works are based on existing material node maps and are not scalable enough. MaterialGAN [GSH\*20] train an unconditional material generation model, which allows the potential of the material to be used in the generator. However, this model entangles spatial layout and style and does not allow explicit control over the generated materials. Based on Adobe Substance Designer [Ado], a procedural material generation tool, MatFormer [GHS\*22], performs material node graph generation through a multi-stage transformer-based model that sequentially generates nodes, node parameters, and edges while ensuring the validity of the graph semantics.

### 2.2. Text to Image Generation

Latent Diffusion [RBL\*22] chooses to perform step-wised denoising on the low-dimensional latent space instead of the high-dimensional original data (image) space in order to reduce the computational overhead in denoising. It makes use of the Cross-Attention mechanism [HZZ\*17], which enables it to be applied to tasks guided by data of different modalities. Stable Diffusion is a large generative model based on Latent Diffusion proposed by Rombach et al. [RBL\*22], and trained specifically on the text2image task, which is an application of Latent Diffusion on the text2image task with good generalization performance by fitting a large amount of generic data during training. SD consists of a perceptual compression module and a conditional generation module. The first module is an AutoEncoder (AE), which compresses the input image into the latent space to obtain the latent feature, and recovers the reconstructed image from the latent space. The second consists of a DM and a text encoder CLIP [RKH\*21], which uses the U-Net [RFB15] structural network as the backbone to diffuse and denoise in the latent space, and uses the text encoded by CLIP as a condition to guide the denoising process.

## 3. Text-Material Dataset

For each material sample, whether it is captured by an instrument or synthesized, matches a certain type of objects in the real world,

and owns a corresponding material label, e.g., brick, jeans, leather, metal, etc. The material label mostly consists of keywords, which are already general semantic representations of the essential properties of the material, but lacks a complete representation of the rich textural features of the material surface. In contrast, a string of text usually contains multiple keywords and relationships between them, which can express more specific meaning and show more comprehensive information. In multi-modal tasks, the CLIP model [RKH\*21] learns the common semantics of images and texts to perform image classification, target detection, and other tasks more accurately with good robustness and generality. For a material, a string of text is able to show its appearance and physical properties more specifically and accurately than just keywords, and also can represent the rich texture of its surface, thus containing more effective semantic information and feeding more important feature to the model. This section details the process of creating a text-material dataset based on the above perspective.

### 3.1. Composition of Dataset

In this work, we focus on generating materials from text, which drive us to build a material dataset with text description. Our text-material dataset contains 2500+ synthetic PBR materials with complete text descriptions. Each sample contains 4 SVBRDF parameter maps and a text label, where the SVBRDFs are albedo, normal, roughness, and metallicity respectively. The text labels of the samples in this dataset match the visual effects shown by the materials after shading, i.e. the center-flashed image, and the focus is on the visual characteristics of the materials. From this point of view, in order to describe the material accurately, we introduce the color and texture labels in addition to the original material labels in the dataset. The final text labels are a specific combination of the three labels, with additional detailed descriptions added.

### 3.2. Collection and Processing of Dataset

To date, there is no publicly available dataset of textures containing text descriptions. We first collected high-quality PBR materials from 4 publicly online resource sites [Dem23; Hav23; Tex32; Sha23]. Then, we checked the parameter maps of each material, filtering out materials that were not obviously seamless and part of the PBR workflow. The data from ambientCG [Dem23] and Poly Haven [Hav23] were also checked for labels containing a main keyword (called label) and several keywords (called tags) describing the content of the material and other information, which were retained to enrich the text and assist in the creation of text labels. The above filtering process resulted in 2500 high quality material samples. Most of samples were tagged with single or combined keywords associated with real-world physical objects to represent the material, e.g. brick wall, marble, wood floor, etc. For a specific material data, it is hard to represent the details of the material related to visual perception using only labels that represent high level semantic information. At the same time, it is also difficult to distinguish between material samples of the same type using only a single keyword label due to its various textures. To address these issues, we introduced two new labels related to visual perception, color and texture, to enrich the description of material properties and to more accurately characterise the material.

Texture usually refers to the detailed parts of an image. The features of texture can be used to identify and classify different objects in order to improve the accuracy of classification tasks. As a rather important visual property, texture patterns have semantic connotations, and adding descriptions of texture patterns can provide more detailed descriptions of the content and characteristics of materials. In order to represent the rich variety of texture patterns using a common set of texture property descriptions, we resort to the DTD [CMK\*14] dataset, a publicly available multi-label dataset for describing common texture attributes of natural objects. DTD was used to research on the issue of texture description, which describes texture attributes. The generic texture attributes of the DTD dataset are also applicable to the material data we collected. To describe the texture of the material, we introduced a multi-label texture attribute classifier to assist in the annotation of the current texture attributes of the material. This classifier uses a pre-trained ResNet101 [HZRS16; DDS\*09] as the backbone network and CSRA [ZW21] as an enhancement module for multi-label classification on the DTD dataset. The module structure enhances the model's perception and classification ability for key regions by introducing an attention mechanism to weight different regions of the input image to different degrees. As the resolution of the DTD dataset varies from 256 to 600, and the resolution of the center-flashed image rendered by our material data is 512, the use of ResNet101 can be adapt well to image inputs of different resolutions.

However, when using the trained classifier for auxiliary annotation of texture attributes, we observed that the texture attribute labels in the DTD dataset were mostly descriptions of some non-regular textures and patterns, and lacked descriptions of regular spatially structured ones such as brick wall, tiles, pavement, etc. Secondly, The texture attributes of the DTD contain descriptions that are strongly related to material properties and are a subset of this material dataset. Finally, the DTD also contains some semantically similar texture attributes that are somewhat repetitive, as well as some texture attributes that are not applicable to the representation of this material dataset. Considering the lack of regular texture attributes and the potential for ambiguity and repetition, we eliminated texture attributes with highly similar semantics to the related materials, texture attributes with overlapping semantics, and texture attributes that deviate significantly from the present material data, such as gauzy, marbled, potholed, bumpy and freckled. By looking at the texture attributes of the material dataset and referring to the corresponding material generation websites, we introduced 9 new texture attributes that are strongly correlated with regular shapes and texture patterns. The new textures images are collected via the Internet and finally merged with the DTD dataset to obtain a multi-label texture classifier, which is trained to infer the corresponding texture attribute labels for the center-flashed images corresponding to the material data. At last, we made 41 labels for classification. More information about our dataset and processing are included in the supplemental document.

### 3.3. Sample Annotation

We generate complete text labels for each material sample based on the above three types of labels to accurately represent the visually perceived properties and characteristics of the material. After the

previous round of annotation, each material sample contains three types of attribute labels, namely: color label  $c$ , material label  $m$  and texture label  $t$ . The specific annotation steps for the text label  $\mathcal{T}$  are as follows. Firstly, the three labels are combined to obtain the initial form  $\mathcal{T}_1$  which is expressed as follows:

$$\mathcal{T}_1 = \mathcal{C}(c, m, t) \quad (1)$$

where  $\mathcal{C}$  denotes the keyword combination function, which specifically checks the syntax of the combined phrases and filters the semantic duplicates or contradictory keywords. The initial form  $\mathcal{T}_1$  is obtained after the first round of combination, and its content is mainly in the form of "[color] [material] with/arranged in [texture] pattern". Afterwards, all material samples are retrieved and the tags  $t_a$  retained during data collection are combined into the initial form according to the content and properties of the material sample to obtain the form  $\mathcal{T}_2$ :

$$\mathcal{T}_2 = \mathcal{C}(t_a, \mathcal{T}_1) \quad (2)$$

Finally, referring to Textures [Tex] and Poliigon [Pol], two major digital asset sites for material classification labels  $t_c$  on the form  $\mathcal{T}_2$  for content additions to obtain the text labels:

$$\mathcal{T} = \mathcal{C}(t_c, \mathcal{T}_2) \quad (3)$$

Specific examples of text label with materials are included in the supplemental document.

#### 4. Text2Mat

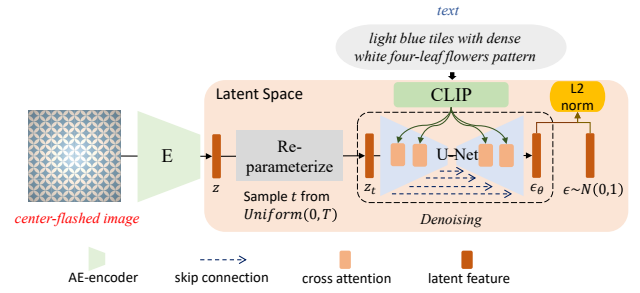
Our goal is to generate a material according to the input text. To this end we have constructed a dataset of text-material pairs, and we present Text2Mat, a material generation framework based on Stable Diffusion (SD) [RBL\*22] that can generate materials highly matching the material type, texture style and color specified in the input text.

Text2Mat mainly comprises two stages. In the first stage, we aim to generate representation of a material in the latent space conditioned on the input text. In the second stage, we perform SVBRDF reconstruction of PBR material in the latent space. Both stages share the same latent space, which allows the stage II of Text2Mat to decode the latent space features generated in the stage I to obtain the corresponding SVBRDF parameters.

##### 4.1. Stage I: Text2Image Generation

In this stage, the Diffusion Model (DM) [HJA20] is denoised by text bootstrapping to obtain the corresponding latent space features. In addition to the textual information from the material dataset with text constructed above, we also need the corresponding image data. As the pre-trained DM is trained on the large-scale text-image pair dataset Laion-5B [SBV\*22], the fitted data distribution is mainly that of natural data, while the diffusion and denoising process of the DM is performed in the hidden space. Therefore, in order to fit the natural data distribution, we rendered each material according to the simplified Disney BRDF model [BS12] with 4 SVBRDF parameters: albedo (or base color), normal, roughness and metallicity.

As illustrated in Fig. 2, based on the BRDF model, we render



**Figure 2: Overview of Stage I.** In Stage I, we fix the pre-trained encoder of AE, aiming to generate the representation of center-flashed image in the latent space of the material specified in the input text.

the shading image with a point light source at center, i.e. a center-flashed image, and then resize the image to  $512 \times 512$ , using this image and the text label as input. When training the text to the image generation module, we encode the shading image of the material sample into the latent space with the pre-trained encoder of AutoEncoder (AE), which is fixed due to the need to adapt it to the second stage. At the same time, the input text is encoded by CLIP to obtain the text features, and the DM is trained to diffuse and denoise in the latent space conditional on the text features, allowing it to fit the distribution of the material dataset corresponding to the shading images under the guidance of the text.

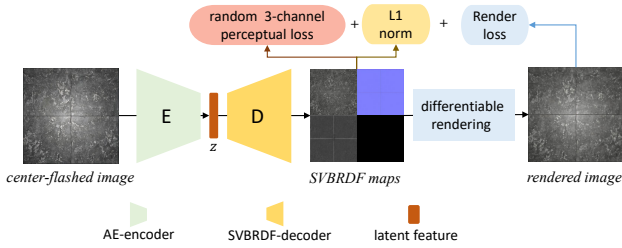
##### 4.2. Stage II: BRDF Reconstruction

Material reconstruction is an ill-posed and fundamental problem in computer graphics that aims to infer the physical properties of a real-world material based on its interaction with a light surface, usually expressed as SVBRDF parametric maps. Common approaches to material reconstruction fall into two categories: one is a modelling method, which fit large-scale data through neural networks and infer the SVBRDF parameters of the material backwards from the data distribution. The other is a combined procedural modelling method, where only a small number of parameters are predicted using the neural network and the final BRDF parameters are generated from procedural node graphs with the predicted parameters. Text2Mat is based on SD, and the SVBRDF reconstruction of the PBR is performed on the latent space features generated by DM.

As shown in Fig. 3, the input shading image is first passed through a fixed encoder of AE to generate the corresponding latent representation, after which the latent representation is reconstructed by the introduced parameter decoder to obtain 4 SVBRDF parameters. Our SVBRDF decoder adopt the same architecture as decoder in AE, but output parametric maps of 8 channels.

##### 4.3. Loss Function and Training Details

The input image during training of Text2Mat can be denoted as  $x \in R^{H \times W \times 3}$ , where  $H = W = 512$ . In Stage I,  $x$  is encoded as a



**Figure 3: Overview of Stage II.** In Stage II, we fix the encoder of AE to share the same latent space of SD, which makes the SVBRDF decoder directly decode the latent representation of shading material to output SVBRDF parameters.

representation  $z \in \mathbb{R}^{\frac{H}{f} \times \frac{W}{f} \times 4}$  in latent space through the fixed encoder of AE, where  $f = 8$  represents the compression rate. The DM then performs a diffusion and denoising training process in the latent space, and the loss function can be simplified as:

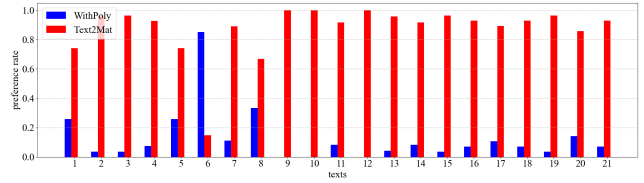
$$\mathcal{L}_{DM} = E_{z, \varepsilon \sim \mathcal{N}(0,1), t} \left[ \|\varepsilon - \varepsilon_{\theta}(z_t, t)\|_2^2 \right] \quad (4)$$

where  $\varepsilon$  is the noise sampled from the standard Gaussian distribution and  $\varepsilon_{\theta}$  represents the noise predicted by the DM. During training,  $t$  is sampled from a uniform distribution  $(0, T)$ , denoting the steps of the forward diffusion process, and  $T = 1000$ . We denote  $z_t$  as the image at step  $t$  of the forward diffusion process.

In Stage II, we fix the pre-trained encoder of AE and feed it with the input image  $x$  to obtain the latent space feature  $z$ . Then, 4 parametric maps, albedo (3), normal (3), roughness (1) and metallicity (1), with total 8 channels will be output after sending  $z$  to the SVBRDF decoder. We use the same network architecture as the SD perceptual compression module for SVBRDF decoder, and change the last convolutional layer to output 8 channels with an accompanying Sigmoid function to normalize the output to range from 0 to 1. In training, We adopt a pixel-wised L1 loss function on each SVBRDF parametric map, i.e. reconstruction loss. Meanwhile, we also apply a perceptual loss function based on VGG. To ensure consistency between different SVBRDF parameters, we randomly select 3 channels within the 8 channels' output for each step in training. Moreover, a differentiable render loss function is added to further supervise the generation of parametric maps. The final loss function is as follows:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_{3per} \mathcal{L}_{3per} + \lambda_{render} \mathcal{L}_{render} \quad (5)$$

where  $\mathcal{L}_{rec}$  and  $\mathcal{L}_{3per}$  calculate the pixel-wised loss between every predicted map and ground truth map.  $\mathcal{L}_{render}$  denotes the pixel-wised loss between the image with differentiable rendering and the input image.  $\lambda_{3per} = 0.1$  and  $\lambda_{render} = 0.05$  denote the weight for the perceptual loss and differentiable rendering loss, respectively. Both stages of Text2Mat were trained on our dataset, with 500 samples, randomly selected, for testing and 2000 samples for training. The training is conducted on one NVIDIA A6000 GPU, taking about 15000 steps on Stage I with a learning rate of  $1e-5$  and 25000



**Figure 4: Results of user study on 21 texts.** Each text is shown as a column, where different colors indicate that users preferred the corresponding method instead of the other.

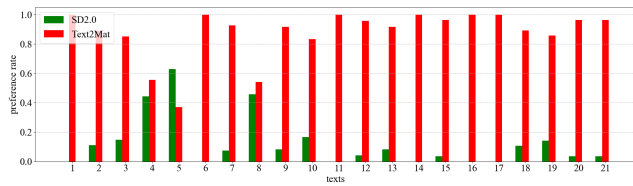
steps on Stage II with a learning rate of  $1e-4$  for a total training time about 3 days.

## 5. Experiments

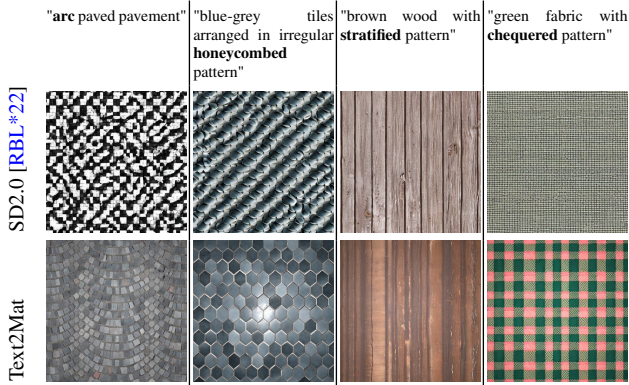
In this section, we evaluate Text2Mat and also show the comparison of the effectiveness of Text2Mat and other methods on text-based material generation tasks. To further validate the effectiveness of the text labels for the datasets constructed by us, we validate them with corresponding ablation experiments. In addition, more experiments were conducted on discussing seamless material generation. In all experiments, we used DPM-Solver [LZB\*22] sampler for inference with a sampling step count of 30 and a classifier-free guidance (cfg) [HS21] of 3.0. It should be emphasised that the focus of our work is on generating materials with simple and effective control, where we choose to make use of convenient texts. Although we use the idea of reconstruction to reconstruct SVBRDFs from latent space features in Stage II, it is still different from material reconstruction directly from single image, and comparing the reconstruction performance is not our main purpose.

### 5.1. Evaluation for Text-to-image Generation

In this subsection, we compare SD with our Text2Mat for text-to-image generation. In our experiments, we use version 2.0 of SD, and Text2Mat is also fine-tuned on SD2.0. Since SD2.0 is trained on generic text-image pairs and cannot directly generate textured images, we specify that SD2.0 generates textured images by adding "A texture map of" to the beginning of the text input of SD2.0 for a fair comparison. A total of 21 text descriptions were selected and a user study was conducted on the results generated by SD2.0 and Text2Mat, where 28 users (most are graduate students researching in CG and CV) were asked to select the result that better matched the text among the images generated by the different models based on the text. Fig. 5 shows the results of the user study, from which it can be seen that the results generated by Text2Mat outperformed SD2.0 on 20 of the samples, some of the generated results of which are shown in Fig. 6. As can be seen in Fig. 6, the original SD2.0 was unable to generate images with matching material styles directly based on the description text, whereas Text2Mat was trained with specific labels, allowing Text2Mat to generalise the original model to generate material images with colors, texture styles and materials that highly correspond to the input text, as evidenced by the results of the user study, thus validating Text2Mat's ability to move from text to material style. The results of the user study also demonstrate



**Figure 5:** Results of user study on 21 texts. Each text is shown as a column, where different colors indicate that users preferred the corresponding method instead of the other.



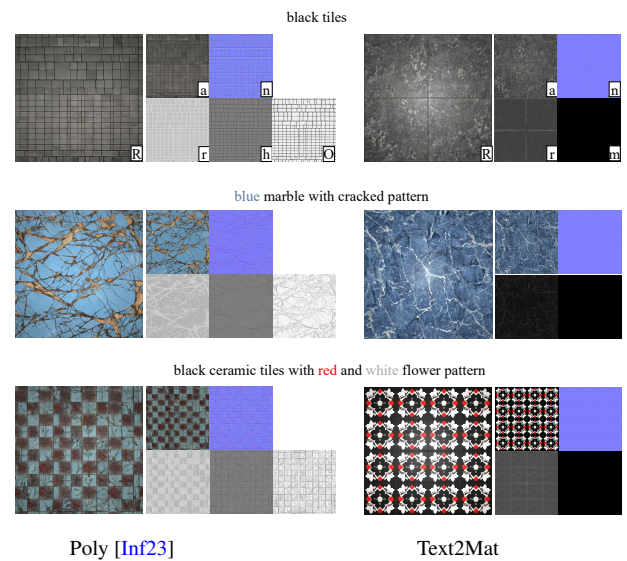
**Figure 6:** Comparison to SD2.0 [RBL\*22]. We show the results on text-to-image generation of SD2.0 and our Text2Mat.

the ability of Text2Mat to control the generation of text-to-material style images.

## 5.2. Evaluation for Material Generation

There is no publicly available method for generating materials based on text, and for text-based material generation we chose to compare Text2Mat with the text-to-material online generation site Poly [Inf23]<sup>†</sup> on this task. In our experiments, we also selected 21 texts and did a corresponding user study, asking 28 users (same people as above user study) to choose the result that better matched the input text among the images generated by the different methods. The results of the user study are shown in Fig. 4 and some samples are shown in Fig. 7, where a, n, r, m, h, O and R denote albedo, normal, roughness, metallicity, height, ambient occlusion and Render respectively. Based on the results of the user study and the comparison of the presented samples, our Text2Mat outperformed Poly on all 20 of the samples, although Poly was able to generate PBR parametric maps based on text: albedo, normal, roughness, height and ambient occlusion. Poly does not generate textures that correspond well to text with textural detail, and does not allow for guided generation of spatial structure using common basic graphic elements such as "I-shaped" and "diamond". Also the textures generated for simple generalised text do not have a regular texture. In contrast, the model Text2Mat corresponds well to both simple

<sup>†</sup> Accessed in February, 2023.



**Figure 7:** Comparison to Poly [Inf23]. We show the results on text-to-material generation of Poly and our Text2Mat. The a, n, r, m, h, O, R denote albedo, normal, roughness, height, metallicity, ambient occlusion and rendering images, respectively.

and more complex descriptions and gives realistic results. More importantly, Poly is not able to generate parametric maps that reflect specular reflections such as specular or metallicity, and cannot use text to guide the generation of metal-like materials. On the other hand, Text2Mat can generate metallic and metal-like materials and has good control over the base color of the metal, as illustrated in Fig. 8. User studies have shown that Text2Mat outperforms Poly for text-to-material generation.

## 5.3. Discussion on Text Labels

The text labels in this dataset are based on the material labels and other keywords, with the introduction of additional colour and texture labels. In order to provide a more comprehensive and detailed description of the visual properties exhibited by the textures, a comparative experiment was conducted on the effectiveness of the additional labels introduced, where -Texture denotes a model trained without the additional texture labels and original denotes a model trained with the original keywords. Fig. 9 shows that since the proposed text labels can have more intuitive and richer descriptions than the original labels, the model trained with the full labels can generate the corresponding texture styles based on the corresponding descriptions, thus generating a material mapping that better matches expectations. The original labels lacked a complete representation of the global and local style of the material, making it difficult to generate a material map that would reasonably fit the input text. As can be seen from Fig. 10, after removing the labels from the texture property descriptions, the trained -Texture model becomes less sensitive to the spatial structure in the image and has difficulty generating texture maps with reasonable texture styles.

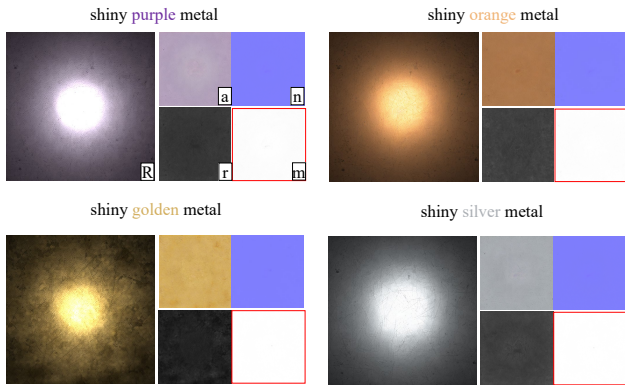


Figure 8: The results of metal-like materials generated by Text2Mat with only text inputs.

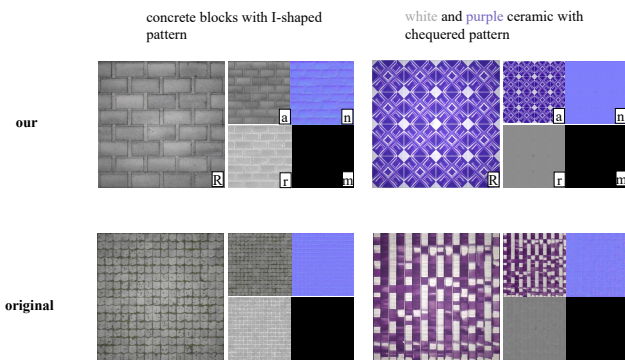


Figure 9: Comparison of the text-to-material generation results of Text2Mat after training with the original labels of constructed dataset.

### 5.4. Seamless Materials Generation

As can be seen from Fig. 11, for materials such as bricks, tiles and other textures that are regular, Text2Mat is able to generate the corresponding materials and corresponding texture patterns freely combined according to the description. At the same time, as the materials such as bricks and tiles in the constructed dataset are seamless, DM fits their corresponding distributions well, allowing Text2Mat to generate seamless brick walls, tiles and other materials.

### 5.5. Discussion on different format of text inputs

To seek the flexibility and complexity of input text that Text2Mat can support, we set different formats of text with similar meaning and one complex cases as the input. We show the rendering results of generated SVBRDFs in Fig. 12. Our Text2Mat can generate fine materials matching the different formats of text input with same meaning, but it's a little bit hard for this framework to generate a quite precise material with more complex and out-of-distribution text input, which is limited by the number and distribution of our dataset.

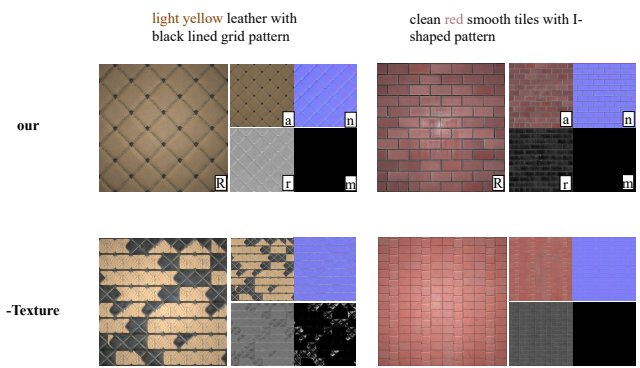


Figure 10: Comparison of the text-to-material generation results of Text2Mat after training without texture label.

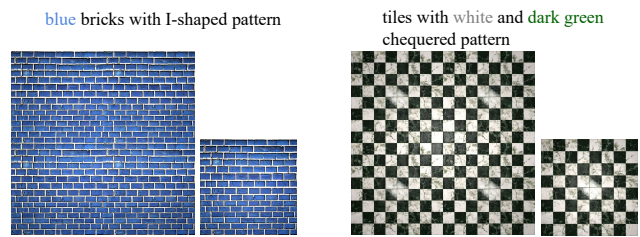


Figure 11: We show the seamless materials of tiles and bricks generated by Text2Mat.

## 6. Conclusion

In this work, we have proposed a text-guided material generation framework: Text2Mat. Based on the need of generating materials from text descriptions, we have collected and produced a text-material dataset with detailed text descriptions. After fully observing the spatial structure and texture of the materials in this dataset, a multi-label texture classifier were trained for texture recognition using the texture attributes from DTD [CMK\*14] as a benchmark. Meanwhile, we introduce different label and constructs a material dataset with text labels by the combination of each label. Based on the SD, we construct a training on proposed dataset through a two-stage training strategy and builds an end-to-end material generation framework guided by text descriptions.

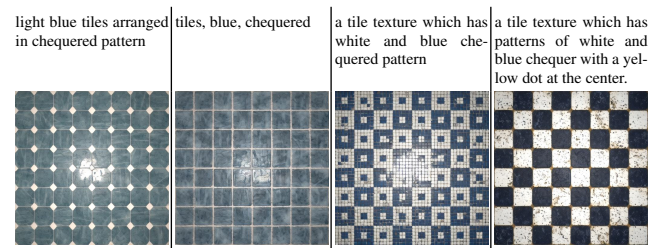


Figure 12: We show rendering results of generated materials on different text inputs with similar meaning and one more complex case.

## References

- [Ado] ADOBE. *Substance Designer*. <https://www.substance3d.com/>. 2023 2.
- [BS12] BURLEY, BRENT and STUDIOS, WALT DISNEY ANIMATION. “Physically-based shading at disney”. *Acm Siggraph*. Vol. 2012. vol. 2012. 2012, 1–7 4.
- [CMK\*14] CIMPOI, MIRCEA, MAJI, SUBHRANSU, KOKKINOS, IASONAS, et al. “Describing Textures in the Wild”. *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, 3606–3613. DOI: [10.1109/CVPR.2014.461](https://doi.org/10.1109/CVPR.2014.461) 2, 3, 7.
- [DAD\*18] DESCHAIANTRE, VALENTIN, AITTALA, MIKA, DURAND, FREDO, et al. “Single-image svbrdf capture with a rendering-aware deep network”. *ACM Transactions on Graphics (ToG)* 37.4 (2018), 1–15 2.
- [DDS\*09] DENG, JIA, DONG, WEI, SOCHER, RICHARD, et al. “Imagenet: A large-scale hierarchical image database”. *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, 248–255 3.
- [Dem23] DEMES, LENNART. *ambientCG*. <https://ambientcg.com>. 2023 3.
- [DWT\*10] DONG, YUE, WANG, JIAPING, TONG, XIN, et al. “Manifold bootstrapping for SVBRDF capture”. *ACM Transactions on Graphics (TOG)* 29.4 (2010), 1–10 2.
- [GHS\*22] GUERRERO, PAUL, HAŠAN, MILOŠ, SUNKAVALLI, KALYAN, et al. “MatFormer: A Generative Model for Procedural Materials”. *ACM Trans. Graph.* 41.4 (July 2022). ISSN: 0730-0301. DOI: [10.1145/3528223.3530173](https://doi.org/10.1145/3528223.3530173). URL: <https://doi.org/10.1145/3528223.3530173>.
- [GLT\*21] GUO, JIE, LAI, SHUICHANG, TAO, CHENGZHI, et al. “Highlight-aware two-stream network for single-image SVBRDF acquisition”. *ACM Transactions on Graphics (TOG)* 40.4 (2021), 1–14 2.
- [GSH\*20] GUO, YU, SMITH, CAMERON, HAŠAN, MILOŠ, et al. “MaterialGAN: Reflectance Capture Using a Generative SVBRDF Model”. *ACM Trans. Graph.* 39.6 (Nov. 2020). ISSN: 0730-0301. DOI: [10.1145/3414685.3417779](https://doi.org/10.1145/3414685.3417779). URL: <https://doi.org/10.1145/3414685.3417779>.
- [Hav23] HAVEN, POLY. *Poly Haven*. <https://polyhaven.com>. 2023 3.
- [HDMR21] HENZLER, PHILIPP, DESCHAIANTRE, VALENTIN, MITRA, NILOY J, and RITSCHEL, TOBIAS. “Generative Modelling of BRDF Textures from Flash Images”. *ACM Trans Graph (Proc. SIGGRAPH Asia)* 40.6 (2021) 2.
- [HDR19] HU, YIWEI, DORSEY, JULIE, and RUSHMEIER, HOLLY. “A novel framework for inverse procedural texture modeling”. *ACM Transactions on Graphics (ToG)* 38.6 (2019), 1–14 2.
- [HJA20] HO, JONATHAN, JAIN, AJAY, and ABBEEL, PIETER. “Denoising diffusion probabilistic models”. *Advances in Neural Information Processing Systems* 33 (2020), 6840–6851 2, 4.
- [HS21] HO, JONATHAN and SALIMANS, TIM. “Classifier-Free Diffusion Guidance”. *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*. 2021. URL: <https://openreview.net/forum?id=qw8AKxYbI> 5.
- [HZL\*17] HAO, YANCHAO, ZHANG, YUANZHE, LIU, KANG, et al. “An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge”. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017, 221–231 2.
- [HZRS16] HE, KAIMING, ZHANG, XIANGYU, REN, SHAOQING, and SUN, JIAN. “Deep residual learning for image recognition”. *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, 770–778 3.
- [Inf23] INFINITY, POLY. *Poly*. <https://withpoly.com/browse/textures>. 2023 6.
- [KCW\*18] KANG, KAIZHANG, CHEN, ZIMIN, WANG, JIAPING, et al. “Efficient reflectance capture using an autoencoder.” *ACM Trans. Graph.* 37.4 (2018), 127–1 2.
- [KLA\*20] KARRAS, TERO, LAINE, SAMULI, AITTALA, MIKA, et al. “Analyzing and improving the image quality of stylegan”. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, 8110–8119 2.
- [LDPT17] LI, XIAO, DONG, YUE, PEERS, PIETER, and TONG, XIN. “Modeling surface appearance from a single photograph using self-augmented convolutional neural networks”. *ACM Transactions on Graphics (ToG)* 36.4 (2017), 1–11 2.
- [LSCI18] LI, ZHENGQIN, SUNKAVALLI, KALYAN, and CHANDRAKER, MANMOHAN. “Materials for masses: SVBRDF acquisition with a single mobile phone image”. *Proceedings of the European conference on computer vision (ECCV)*. 2018, 72–87 2.
- [LZB\*22] LU, CHENG, ZHOU, YUHAO, BAO, FAN, et al. “DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps”. *arXiv preprint arXiv:2206.00927* (2022) 5.
- [NDR\*21] NICHOL, ALEX, DHARIWAL, PRAFULLA, RAMESH, ADITYA, et al. “Glide: Towards photorealistic image generation and editing with text-guided diffusion models”. *arXiv preprint arXiv:2112.10741* (2021) 2.
- [Nic65] NICODEMUS, FRED E. “Directional reflectance and emissivity of an opaque surface”. *Applied optics* 4.7 (1965), 767–775 1.
- [Pol] POLIIGON. *Poliigon*. <https://www.poliigon.com/textures>. 2023.2 4.
- [RBL\*22] ROMBACH, ROBIN, BLATTMANN, ANDREAS, LORENZ, DOMINIK, et al. “High-resolution image synthesis with latent diffusion models. 2022 IEEE”. *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, 10674–10685 2, 4, 6.
- [RFB15] RONNEBERGER, OLAF, FISCHER, PHILIPP, and BROX, THOMAS. “U-net: Convolutional networks for biomedical image segmentation”. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III* 18. Springer. 2015, 234–241 2.
- [RKH\*21] RADFORD, ALEC, KIM, JONG WOOK, HALLACY, CHRIS, et al. “Learning transferable visual models from natural language supervision”. *International conference on machine learning*. PMLR. 2021, 8748–8763 2, 3.
- [SBV\*22] SCHUHMAN, CHRISTOPH, BEAUMONT, ROMAIN, VENCU, RICHARD, et al. “Laion-5b: An open large-scale dataset for training next generation image-text models”. *arXiv preprint arXiv:2210.08402* (2022) 4.
- [Sha23] SHARETEXTURES. *ShareTextures*. <https://www.sharetextures.com>. 2023 3.
- [SLH\*20] SHI, LIANG, LI, BEICHEN, HAŠAN, MILOŠ, et al. “Match: Differentiable Material Graphs for Procedural Material Capture”. *ACM Trans. Graph.* 39.6 (Nov. 2020). ISSN: 0730-0301. DOI: [10.1145/3414685.3417781](https://doi.org/10.1145/3414685.3417781). URL: <https://doi.org/10.1145/3414685.3417781>.
- [SWMG15] SOHL-DICKSTEIN, JASCHA, WEISS, ERIC, MAHESWARANATHAN, NIRU, and GANGULI, SURYA. “Deep unsupervised learning using nonequilibrium thermodynamics”. *International Conference on Machine Learning*. PMLR. 2015, 2256–2265 2.
- [Tex] TEXTURES. *Textures*. <https://www.textures.com>. 2023.2 4.
- [Tex32] TEXTURES, 3D. *3D Textures*. <https://3dtextures.me>. 2023.2 3.
- [ZHD\*22] ZHOU, XILONG, HASAN, MILOS, DESCHAIANTRE, VALENTIN, et al. “Tilegen: Tileable, controllable material generation and capture”. *SIGGRAPH Asia 2022 Conference Papers*. 2022, 1–9 2.



[ZK21] ZHOU, XILONG and KALANTARI, NIMA KHADEMI. “Adversarial Single-Image SVBRDF Estimation with Hybrid Training”. *Computer Graphics Forum*. Vol. 40. 2. Wiley Online Library. 2021, 315–325 [2](#).

[ZW21] ZHU, KE and WU, JIANXIN. “Residual attention: A simple but effective method for multi-label recognition”. *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 184–193 [3](#).