# Text2Tex: Text-driven Texture Synthesis via Diffusion Models

Dave Zhenyu Chen[1] Yawar Siddiqui[1] Hsin-Ying Lee[2] Sergey Tulyakov[2] Matthias Nießner[1]
[1]Technical University of Munich [2]Snap Research
https://daveredrum.github.io/Text2Tex/

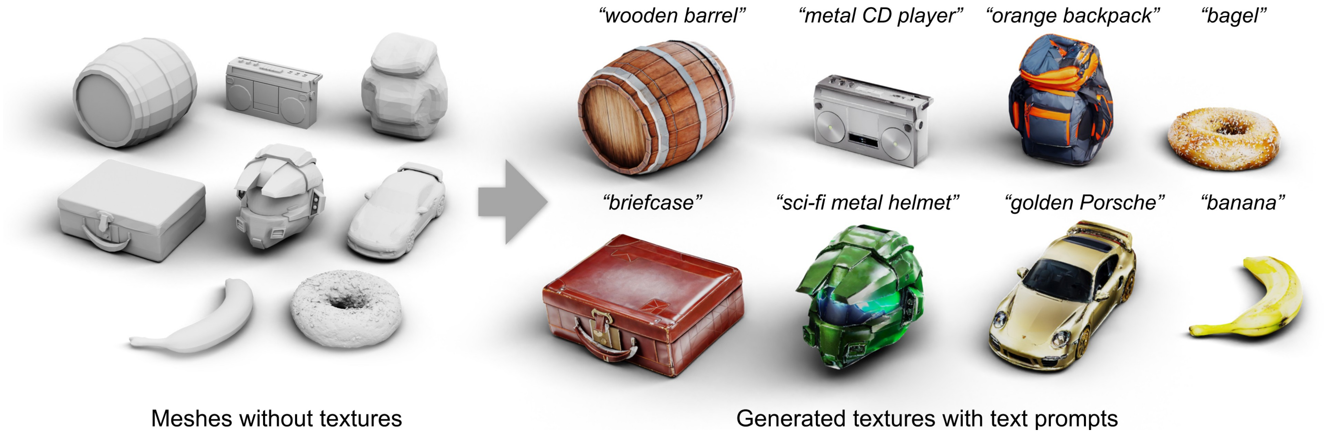

Figure 1: We introduce Text2Tex, a text-driven architecture for 3D texture synthesis. Given object geometries and text prompts as input, Text2Tex generates high quality and consistent textures via depth-aware inpainting and refinement.

## Abstract

*We present Text2Tex, a novel method for generating high-quality textures for 3D meshes from the given text prompts. Our method incorporates inpainting into a pre-trained depth-aware image diffusion model to progressively synthesize high resolution partial textures from multiple viewpoints. To avoid accumulating inconsistent and stretched artifacts across views, we dynamically segment the rendered view into a generation mask, which represents the generation status of each visible texel. This partitioned view representation guides the depth-aware inpainting model to generate and update partial textures for the corresponding regions. Furthermore, we propose an automatic view sequence generation scheme to determine the next best view for updating the partial texture. Extensive experiments demonstrate that our method significantly outperforms the existing text-driven approaches and GAN-based methods.*

## 1. Introduction

Generating high-quality 3D content is an essential component of visual applications in films, games, and upcoming AR/VR scenarios. With an increasing number of 3D content datasets, the computer vision community has witnessed significant progress in the field of 3D geometry generation [12, 39, 61, 66, 38]. Despite the remarkable success in modeling 3D geometries in recent years, fully automatic 3D content generation is still hindered by the laborious human efforts required to design textures. Therefore, automating the texture design process through alternative guidance, such as text, has become an intriguing but challenging research problem.

Recently, text-to-image generators have shown remarkable progress in the 2D domain leveraging diffusion model architectures, enabling high resolution 2D content generation based on textual descriptions [1, 48]. However, there are notable challenges for producing 3D textures via such 2D vision-language prior knowledge. Specifically, the synthesized textures are expected to be not only with high fidelity to the language cues, but also of high and consistent quality for target meshes. As such, previous attempts to paint 3D geometry from text inputs often fail to deliver well-textured 3D content.

In this paper, we introduce **Text2Tex**, a novel texture synthesis method that seamlessly texturizes 3D objects using a pre-trained depth-aware text-to-image diffusion model. The method renders a target mesh from multiple viewpoints and inpaints the missing appearance with a depth-aware text-to-image diffusion model. Text2Tex follows a *generate-then-refine* strategy. Our method progressively generates partial textures across viewpoints and back-projects them to texture space. To address stretched and inconsistent artifacts observed from rotated viewpoints, we design a *view partitioning technique* that computes similarity maps between visible texel's normal vectors and the current view direction. The generation mask created from these similarity maps guides the diffusion process by indicating regions to generate, update, keep, or ignore. This allows us to apply different diffusion strengths to respective regions, inpainting missing appearance and updating stretched artifacts. However, the autoregressive generation process via the diffusion-based image inpainting model presents a new challenge. As the inpainting and updating scheme is conditioned on previously synthesized results, a viewpoint sequence with an ill-defined order or incomplete coverage over the mesh surface may result in unsatisfactory texturization. Therefore, we propose an *automatic viewpoint selection* technique that progressively selects the next best view. The confidence of each candidate view containing the biggest relative area for generation and updating is estimated, given the partially textured mesh. This approach ensures complete coverage over the mesh surface and a high-quality texture map by consistently updating stretched regions.

We demonstrate the effectiveness of Text2Tex for synthesizing high-quality 3D textures from language cues. The proposed method performs favorably against other language-based texture synthesis methods in terms of FID [23], KID [4], and user study on a subset of the Objaverse dataset [17]. Additionally, our method also outperforms category-specific GAN-based methods on the ShapeNet car dataset [7].

To summarize, our technical contributions are threefold:

- We design a novel method for high-quality texture synthesis by progressively inpainting and updating the 3D textures via depth-aware diffusion models.
- We propose an automatic view sequence generation scheme to dynamically determine the order for generating and updating the texture space.
- We conduct extensive study on a considerable amount of 3D objects, demonstrating the proposed method is effective for large-scale 3D content generation.

## 2. Related work

**3D Generation from 3D and 2D data.** To achieve 3D generation, it is natural to train models directly on 3D data. In contrast to 2D images, there are several 3D representations available, each with its unique characteristics, leading to the development of various generative models such as those based on voxels [30, 55, 59, 62], point clouds [3, 32], meshes [68], signed distance function [12, 13, 14, 16, 36], etc. However, unlike images or videos that are ubiquitous, 3D data is inherently scarce and challenging to collect and annotate. Consequently, the synthesized samples from 3D generative models, trained on 3D data, are of limited quality and diversity, in terms of both structure and texture. Recent works have leveraged differentiable rendering to learn texture generation using only 2D images [20, 56, 65]. However, they are typically trained for specific shape categories and struggle in the quality of textures.

**Text-Guided Generation.** Recently, there has been tremendous progress in the vision-language domain [26, 57, 60, 47, 10, 8, 11, 9]. Specifically, the emergence of Contrastive Language-Image Pre-Training (CLIP) [47] has enabled the development of text-guided image generation through its semantically rich representation trained on text-image pairs. Initial efforts [15, 45] incorporated CLIP with different backbones, such as StyleGAN [27] and VQ-GAN [19]. However, diffusion models [18, 24, 25, 40, 53], which have gained attention due to their superior visual quality and training stability compared to Generative Adversarial Networks (GANs) [21], have recently been trained on large-scale text-image datasets with CLIP encodings [28, 49, 52]. Among these models, Stable Diffusion [1, 52] has garnered significant interest as an open-sourced model with numerous extensions [1, 67] that support different conditional modalities in addition to text prompts, including depth images, poses, sketches, etc. Additionally, CLIP has also been adopted in 3D to perform text-guided shape and texture generation [34, 37]. In this work, we take advantage of the depth-conditioning feature of Stable Diffusion to provide more consistent texturing.

**Text-to-3D from 2D data.** Inspired by the success of Neural Radiance Fields (NeRF) [35], NeRF-based generators [2, 5, 6, 22, 41, 43, 54, 58, 63] have been proposed to learn 3D structures from 2D images using GAN-based frameworks. A new research direction emerged by combining NeRF techniques with flourishing diffusion-based text-to-image models, enabling text-to-3D learning with only 2D supervision. To address the challenge of optimizing a NeRF field, a score-distillation loss is proposed [46] that leverages a pretrained 2D diffusion model as a critic to provide essential gradients. Subsequent efforts have focused on adopting this loss in latent space [13, 33] and in a coarse-to-fine refinement approach [29]. However, optimization-based methods are plagued by long convergence times. A recent concurrent work [51] proposes a non-optimization approach with progressive updates from multiple pre-set viewpoints. In contrast, our method iteratively updates and refines the synthesized textures from automatically selected viewpoint sequences, which minimizes human efforts with designing different viewpoint orders for various geometries.

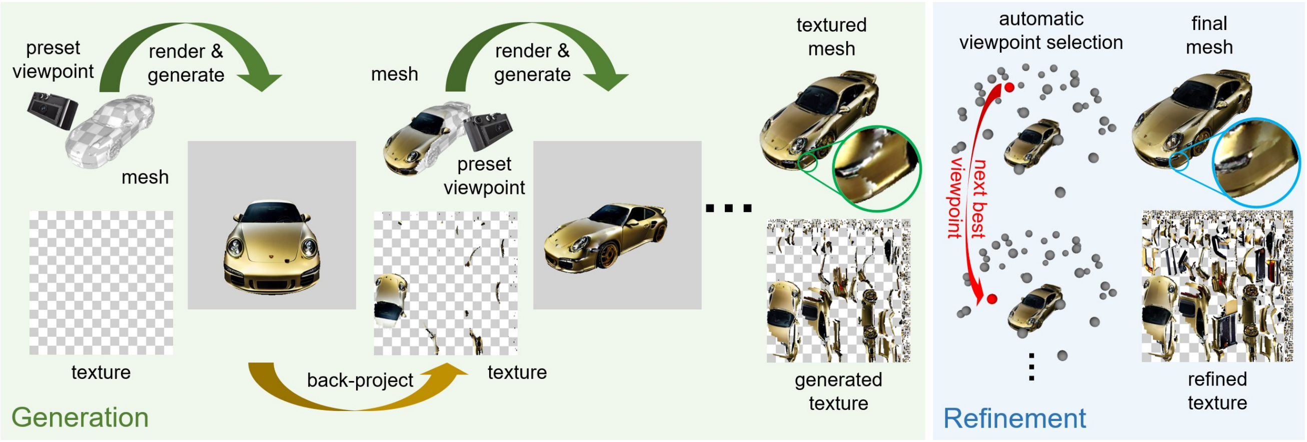

Figure 2: Overview of Text2Tex. We illustrate the pipeline using a 3D car mesh with a prompt "*golden Porsche*". We progressively generate the texture via a *generate-then-refine* scheme. In **progressive texture generation** (Sec. 3.3), we start by rendering the object from an initial preset viewpoint. We generate a new appearance according to the input prompt via a depth-to-image diffusion model, and project the generated image back to the partial texture. Then, we repeat this process until the last preset viewpoint to output the initial textured mesh. In the subsequent **texture refinement** (Sec. 3.4), we update the initial texture from a sequence of automatically selected viewpoints to refine the stretched and blurry artifacts.

# 3. Method

The objective of our work is to texture a 3D mesh using a pretrained text-to-image diffusion model. In this section, we begin by laying the foundation of the diffusion model in Sec. 3.1 and depth-aware inpainting model in Sec. 3.2. We then propose a *generate-then-refine* scheme for progressively synthesizing and updating the 3D textures in a coarse-to-fine fashion. In the progressive texture generation (Sec. 3.3), we paint the visible regions of the input geometry in an incremental fashion, following a sequence of predefined viewpoints. To ensure local and global consistency, we incorporate a view partition to guide the depth-aware inpainting objectives. Subsequently, we introduce an automatic viewpoint selection mechanism (Sec. 3.4) to perform texture refinement and address any issues of texture stretching and inconsistency.

## 3.1. Preliminary

We use a Denoising Diffusion Probabilistic Model (DDPM) [24] as the generative model. Specifically, to avoid high computational overhead, we adopt the latent diffusion model [52], where an input image $x_0$ is first encoded into latent code $z_0$ before the diffusion process. The forward pass follows a Markov Chain to gradually add noise to the input latent code $z_0$ towards the white Gaussian noise $\mathcal{N}(0,1)$. At each step in the forward pass, the noised latent code $z_t$ is obtained by adding a noise variance $\beta_t$ to the previous latent code $z_{t-1}$ scaled with $\sqrt{1-\beta_t}$:

$$z_t \sim \mathcal{N}(\sqrt{1-\beta_t} z_{t-1}, \beta_t \mathbf{I}). \tag{1}$$

The independence property enables direct transformation of the noised latent code $z_t$ at an arbitrary time step $t$ from the input latent $z_0$ via:

$$z_t \sim \mathcal{N}(\sqrt{\bar{a}_t} z_0, (1-\bar{a}_t)\mathbf{I}), \tag{2}$$

where $\bar{a}_t$ is the total noise variance, which can be calculated by $\sum_{t=i}^{T}(1-\beta_t)$ from the noise $\beta_t$ added to the input latent code $z_0$ at each time step.

During inference, the latent estimation $\hat{z}_{t-1}$ for the next time step $t-1$ is obtained by predicting $\mu_\theta(z_t,t)$ and $\sigma_\theta(z_t,t)$ of a Gaussian distribution:

$$\hat{z}_{t-1} \sim \mathcal{N}(\mu_\theta(z_t,t), \sigma_\theta(z_t,t)) \tag{3}$$

**Denoising strength.** To prevent complete randomness during the diffusion process, we introduce a scaling factor $\gamma, 0 < \gamma \leq 1$, which controls the number of diffusion steps. We assume that a white Gaussian noise $\mathcal{N}(0,1)$ can be obtained by adding noise to the input latent code $z_0$ through $T$ steps, and the final denoised latent estimation $\hat{z}_0$ is fully governed by the pure noise. By applying the scaling factor, we can start denoising the latent code at time step $\gamma T$ to guide the final latent code with the original image information. This technique is applied to refine the previously generated image contents.

## 3.2. Depth-Aware Image Inpainting

The core of the texture synthesis lies in painting the missing regions on the mesh surface. The generated texture is expected to be highly faithful to the mesh geometry and the

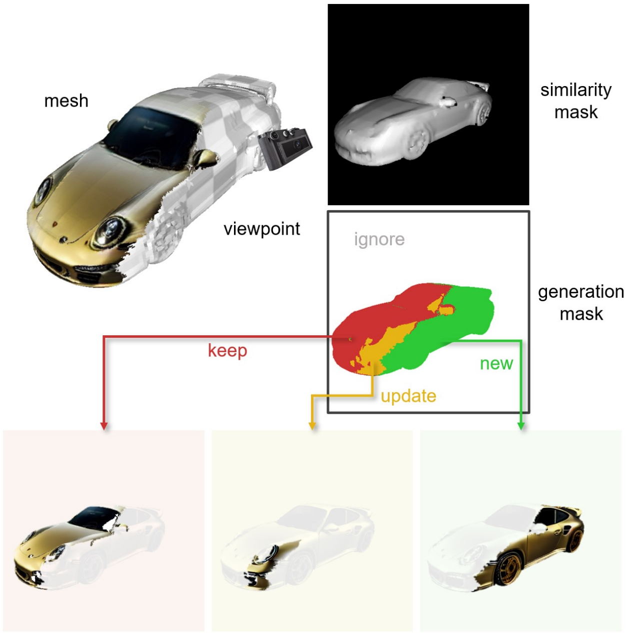

Figure 3: We dynamically partition the current view into a generation mask to guide the depth-aware inpainting model. For the "new" region, we denoise the new object appearance from white Gaussian noise. For the "update" region, we refine the previous texture by denoising the partially noised image segments. We freeze the texture in the "keep" region for this view.

input text. To achieve this, we build our method on a pre-trained depth-to-image model [1, 67] that can produce high-quality images from text while being consistent with depth cues. However, since the Depth2Image model is designed to generate entire images, we need to use an inpainting mask to guide the sampling process. This mask provides explicit hints of which regions to generate or keep fixed, similar to the denoising guidance strategy in RePaint [31].

To condition the denoising on the known regions of the input, we inject a generation mask $\mathcal{M}$ into the sampling steps. This mask explicitly blends the noised latent code $z_t$ and the denoised latent estimation $\hat{z}_t$ as follows:

$$\hat{z}_t = \hat{z}_t \odot \mathcal{M} + z_t \odot (1 - \mathcal{M}). \tag{4}$$

We then decode the final denoised latent estimation $\hat{z}_0$ to the final output image $\mathcal{O}$.

### 3.3. Progressive Texture Generation

With the customized Depth2Image model, we are able to paint the object with a high quality image from a single view. To synthesize the appearance of the input geometry, we project the generated 2D views onto the texture space of a normalized 3D object with proper UV parameterization. Assuming Y-axis as the up-axis in the world coordinate system, we define the viewpoint as $v = (\theta, \phi, r)$, where $\theta$ is the azimuth angle with respect to the Z-axis, $\phi$ is the viewpoint elevation angle with respect to the XZ-plane, and $r$ is the distance between the viewpoint and the origin.

As shown in Fig. 2, we start by generating the visible but missing texture in an initial viewpoint $v^0$. We render the object to a depth map $\mathcal{D}^0$ and a generation mask $\mathcal{M}^0$, and then use a customized Depth2Image diffusion model with $\mathcal{D}^0$ as input and $\mathcal{M}^0$ as extra guidance to generate a colored image $\mathcal{O}^0$. We then back-project the image $\mathcal{O}^0$ to the visible part of the texture $\mathcal{T}^0$. In the subsequent steps, we progressively diffuse the colored images $\mathcal{O}^k$ and back-project them to the texture $\mathcal{T}^k$ through a sequence of viewpoints.

We notice that directly inpainting the missing regions on mesh surface often results in inconsistency issues. The issue is mainly caused by the stretched artifacts that occur when the 2D views are projected back onto the curved surface of the mesh. Therefore, we design a dynamic view partitioning strategy to guide the inpainting process with respective generation objectives $\mathcal{M}$ and denoising strengths $\gamma$.

**Dynamic view partitioning** For all viewpoints $\mathcal{V} = v_i, i = 1, ..., N$, we render the similarity mask $\mathcal{S}^i$ for each viewpoint $v^k$ and map those values to the texture space. Each pixel in a similarity mask represents the reversed normalized value of the cosine similarity between the normal vectors of the visible faces and the view direction (ranging from 0 to 1). A pixel with a value of 1 indicates that the corresponding face is perpendicular to the view direction. For simplicity, we set the background to 0 . In summary, these masks indicate the extent to which a face is rotated away from the viewpoint.

Based on the similarity mask $\mathcal{S}^k$ at step $k$, we segment the rendered view into a **generation mask** $\mathcal{M}^k$, including the following 4 regions, as shown in Fig. 3: 1) **New**: This region contains pixels that have not yet been textured. We inpaint this region from pure white Gaussian noise, i.e. with denoising strength 1. 2) **Update**: This region contains pixels that have been textured, but the corresponding similarity score in $\mathcal{S}^k$ is greater than all other views. This indicates that those pixels are being observed in a better angle. Therefore, we update this region with a moderate denoising strength $\gamma_g$ to avoid stretched appearance. 3) **Keep**: Pixels in this region have been textured, but the corresponding similarity score in $\mathcal{S}^k$ is not the highest among all other views.These pixels have already been observed from a better angle, so we keep them fixed. 4) **Ignore**: This region contains pixels that belong to the background and are irrelevant to the process, so we ignore them throughout the entire process.

While the generation mask helps guide the texture inpainting process with accurate generation objectives and ap-

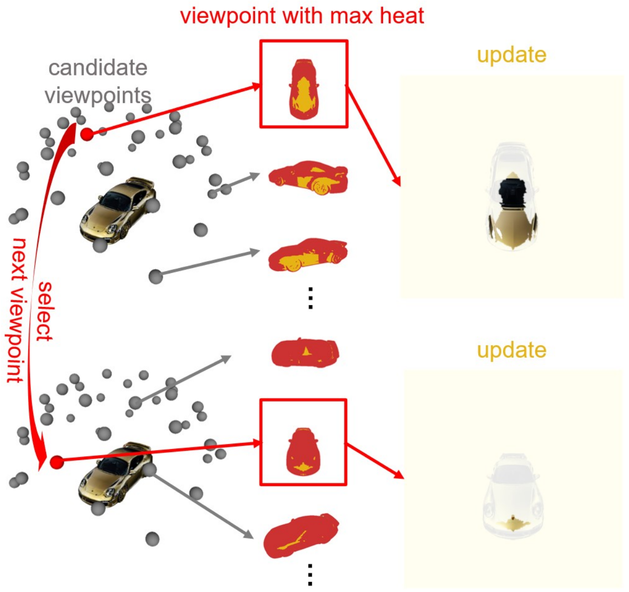

Figure 4: In the refinement stage, the sequence of viewpoints are automatically determined by selecting the viewpoint with the maximum normalized area of the "update" region at each step. We update 2D views in the "update" regions with a moderate diffusion denoising strength. The updated object appearance is then back-projected to the texture space at the end of each refinement step.

propriate denoising strengths, blurriness and stretches can still exist on the mesh surface. This is because the generation mask is limited to a predefined set of viewpoints, and the seams and stretches on the texture are still visible from a novel viewpoint. To address this issue, we propose a texture refinement technique with an automatic viewpoint selection strategy, which is described in the next section.

### 3.4. Texture Refinement with Automatic Viewpoint Selection

To remove the synthesis artifacts, a straightforward solution is to increase the number of viewpoints. However, the optimal viewpoint sequence can vary for different object geometries, making it difficult to manually pre-set the viewpoint sequence for massive synthesis targets. To address these challenges, we propose an automatic viewpoint selection strategy that effectively prevents stretches and seams, as illustrated in Fig. 4. We densely define a set of refinement viewpoints $\mathcal{V} = v_i, i = 1, ..., K$, where $K$ is larger than $N$. To distribute the refinement viewpoints evenly, we scatter them on a hemisphere, taking into account that objects are rarely observed from the bottom-up view.

Assuming that an initial texture has been applied to the object, the refinement process begins by segmenting the generation masks $\mathcal{M}$ using the similarity masks $\mathcal{S}$ from all available viewpoints. For each of the $K$ refinement viewpoints in $\mathcal{V}$, we calculate a view heat $h^i$ from the corresponding generation mask $\mathcal{M}_i$, which represents the normalized area of the "update" region with respect to the current visible area of the object. The viewpoint $v_i$ that maximizes the view heat is then selected by $\arg\max_i h_i = \frac{1}{N_p}\sum_{k=1}^{N_p} w_k$ where $N_p$ is the total number of the non-background pixels, and $w_k$ is the scaling factor for the segments in the generation mask. In order to let views with relatively more areas for updating, $w_k$ for the "update" region is set to be bigger than that for the "keep" region. We dynamically select the next view with the highest view heat for updating. To avoid conflicts with the previously generated textures, we update the "update" regions with a mild denoising strength $\gamma_r$, which preserves the original appearance cues.

## 4. Results

### 4.1. Implementation Details

We apply the Depth2Image model from Stable Diffusion v2 [1] as our generation backbone. The denoising strength $\gamma_g$ and $\gamma_r$ are set as 0.5 and 0.3 for the generation and refinement stages, respectively. We define 6 axis-aligned principles viewpoints for generation, and in total 36 viewpoints for refinement, among which we dynamically select only 20 views to reduce time cost. Each synthesis process takes around 15 minutes to complete on an NVIDIA RTX A6000. Our implementation uses the PyTorch [44] framework with PyTorch3D [50] used for rendering and texture projection.

### 4.2. Experiment Setup

**Data.** We evaluate our method on a subset of textured meshes from the Objaverse [17] dataset. We first sample 3 random meshes from each category. To ensure the quality of the input meshes, we manually filter out thin or unrecognizable meshes, such as "strainer", "sweatpants", and "legging", meshes with too simplistic textures, and meshes that do not correspond with their assigned categories. For the purpose of reducing processing time, we also remove over-triangulated and scanned objects. After this curation, there are in total 410 high quality textured meshes across 225 categories for the experiments. Note that the original textures are only used for the evaluation. To compare with GAN-based category-specific approaches, we also report results on the "car" objects from the ShapeNet dataset [7]. In particular, we use the 300 meshes from the test set used in [56].

**Baselines.** We compare our method against the following state-of-the-art text-driven texture synthesis method: 1) **Text2Mesh** [34], a neural pipeline that directly optimizes the textures and geometries via a CLIP-based optimization objective. We remove the displacement prediction so that

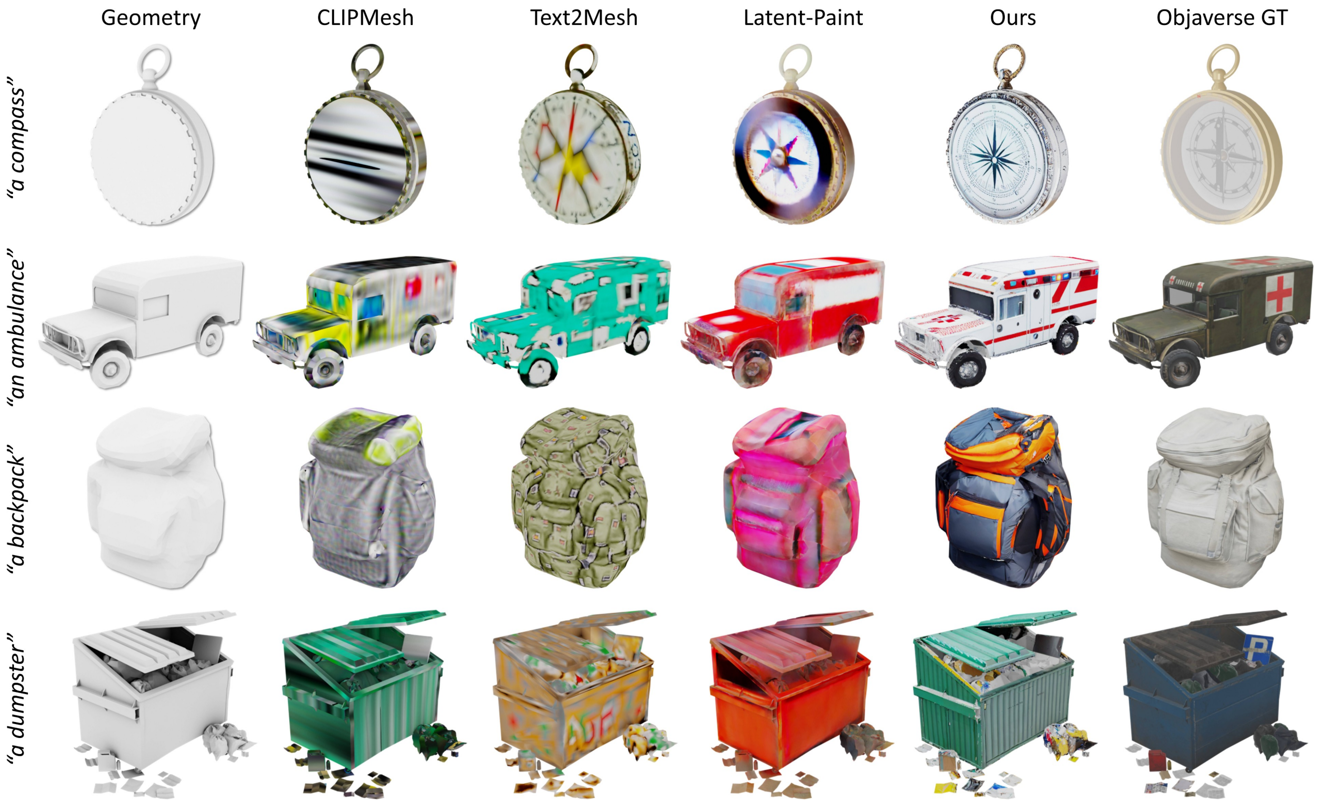

Figure 5: Qualitative comparisons on Objaverse. We compare our textured mesh against CLIPMesh [37], Text2Mesh [34], Latent-Paint [33], and the original textures from Objaverse. In comparison with the baselines, our method produces more consistent and detailed 3D textures with respect to the input geometries. Image best viewed in color.

only the surface RGBs are optimized. 2) **CLIPMesh** [37], a CLIP-based pipeline that deforms a sphere and optimizes the surface RGBs. Similar to Text2Mesh, we remove the shape deformation branch, and the texture colors are directly optimized on the surface of a given geometry. 3) **Latent-Paint** [33], a texture generation variant of the NeRF-based 3D object generation pipeline Latent-NeRF [33]. It explicitly operates on a given texture map using Stable Diffusion as a prior. In addition to text-guided methods, we also compare with category-specific GAN-based approaches, including Texture Fields [42], SPSG [16], LTG [65], and Texturify [56].

**Evaluation metrics.** We evaluate the generated textures via commonly used image quality and diversity metrics for generative models. Specifically, we report the Frechet Inception Distance (FID) [23] and Kernel Inception Distance (KID $\times 10^{-3}$ ) [4]. The generated image distribution for these metrics consists of renders of each mesh with the synthesized textures from 20 fixed viewpoints at a resolution of $512 \times 512$. For experiments on Objaverse dataset, the real distribution comprises renders of the meshes with the same settings using their artist designed textures. For experiments on ShapeNet cars, we use the 18991 background segmented images from CompCars dataset [64].

## 4.3. Quantitative results

In Tab. 1, we compare our method against the previous SOTA text-driven texture synthesis methods on Objaverse objects. As input, we uniformly feed template texts "*a* ⟨*category*⟩" to the models. Quantitatively, our method outperforms all baselines by a significant margin (19% improvement in FID and 26% improvement in KID). Such improvements demonstrate that our method is more capable of generating more realistic textures on various object geometries from numerous categories. To demonstrate the effectiveness of our method against the GAN-based approaches on category-specific data, we report experiment results on ShapeNet "car" category in Tab. 2. Notably, our method achieves superior performance over the previous GAN-based SOTA texture synthesis method Texturify, improving by 21% in FID and 12% in KID. This indicates our method is more effective with synthesizing realistic textures than GAN based approaches that were trained on specific categories.

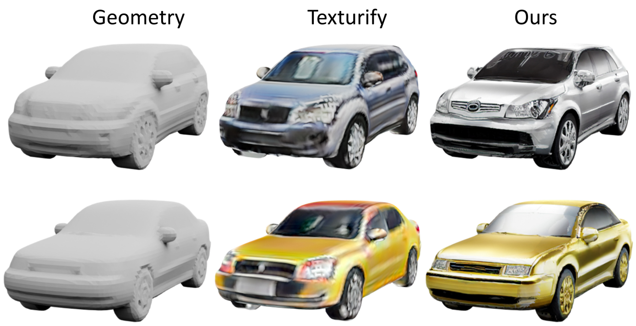

Figure 6: Qualitative comparisons on ShapeNet car. Our method generates sharper and more coherent textures with respect to the geometries compared to the state-of-the-art GAN-based method.

| Method | FID ↓ | KID (x10$^{-3}$) ↓ |
|---|---|---|
| Text2Mesh [34] | 45.38 (+9.7) | 10.40 (+2.7) |
| CLIPMesh [37] | 43.25 (+7.6) | 12.52 (+4.8) |
| Latent-Paint [33] | 43.87 (+8.1) | 11.43 (+3.7) |
| Text2Tex (Ours) | **35.68** | **7.74** |

Table 1: Quantitative comparisons on Objaverse subset. Our method performs favaorably against state-of-the-art text-driven texture synthesis methods.

| Method | FID ↓ | KID (x10$^{-3}$) ↓ |
|---|---|---|
| Texture Fields [42] | 177.15 (+130.2) | 17.14 (+12.8) |
| SPSG [16] | 110.65 (+63.7) | 9.59 (+5.2) |
| LTG [65] | 70.76 (+23.8) | 5.72 (+1.4) |
| Texturify [56] | 59.55 (+12.6) | 4.97 (+0.6) |
| Text2Tex (Ours) | **46.91** | **4.35** |

Table 2: Quantitative comparison on the ShapeNet cars. Our method outperform state-of-the-art category-specific GAN-based methods by a significant margin.

| | CLIPMesh ↑ | Text2Mesh ↑ | Latent-Paint ↑ |
|---|---|---|---|
| Ours | 83.92% | 76.47% | 64.18% |

Table 3: Percentage of users who prefer our method over the baselines in a user study with 604 responses across 41 participants. Our method is shown to be more favored by human users.

| w/ Depth2Img | w/ inpainting | w/ update | FID ↓ | KID (x10$^{-3}$) ↓ |
|---|---|---|---|---|
| ✓ | x | x | 39.88 | 9.78 |
| ✓ | ✓ | x | 38.19 | 9.11 |
| ✓ | ✓ | ✓ | **37.09** | **8.78** |

Table 4: Effect of components in the generation stage. We quantitatively ablate the efficacy of each component. Applying the inpainting and update scheme effectively improves the quality of the synthesized textures.

**User study.** We conduct user study to analyze the quality of the synthesized textures and their fidelity to the input text prompts. For each baseline method, we randomly show the users 5 pair of renders from the baseline and our method. The users are requested to choose the one that is more realistic and closer to the text prompts. More details of the questionnaire can be found in the supplemental. In the end, we receive 604 responses across 41 users. The collected preferences are reported in Tab. 3. In comparison to CLIPMesh and Text2Mesh, our method is clearly preferred by the users with preference rate 83.92% and 76.47%, respectively. Besides that, more users (64.18%) lean towards our method over the competitive baseline Latent-Paint. As can be seen, our method demonstrates the effectiveness in generating high quality textures that are favored by human users.

## 4.4. Qualitative analysis

We compare our qualitative results on Objaverse objects against text-driven baselines in Fig. 5. In comparison with CLIP-based methods CLIPMesh and Text2Mesh, our method generates more realistic and consistent textures. In particular, CLIPMesh generates sketchy textures while Text2Mesh produces repetitive patterns. Latent-Paint outputs a consistent texture capturing the semantics of the object well, but the results are often quite blurry. Clearly, our method can synthesize more consistent textures with cleaner and richer local details. We also compare our textures with GAN-based generation approach on category-specific objectives. In Fig. 6, we show the textures for ShapeNet cars of our methods and Texturify. Notably, our textures have a much cleaner appearance and provide more details with respect to the input geometries.

## 4.5. Ablation studies

**Does depth-aware inpainting and updating help?** We show in Fig. 7 that the depth-aware inpainting is essential for producing high quality textures. In particular, the plain Depth2Img model often struggles to produce consistent appearances due to the governance of random noise. When the inpainting scheme is applied, the produced textures are more consistent. However, the textures still appear to be stretched and blurry over the curved mesh surface. These artifacts are amended by the texture updating scheme from the better viewing angles. The effectiveness of the depth-aware inpainting and the updating scheme is reflected in the improved FID and KID scores in Tab. 4.

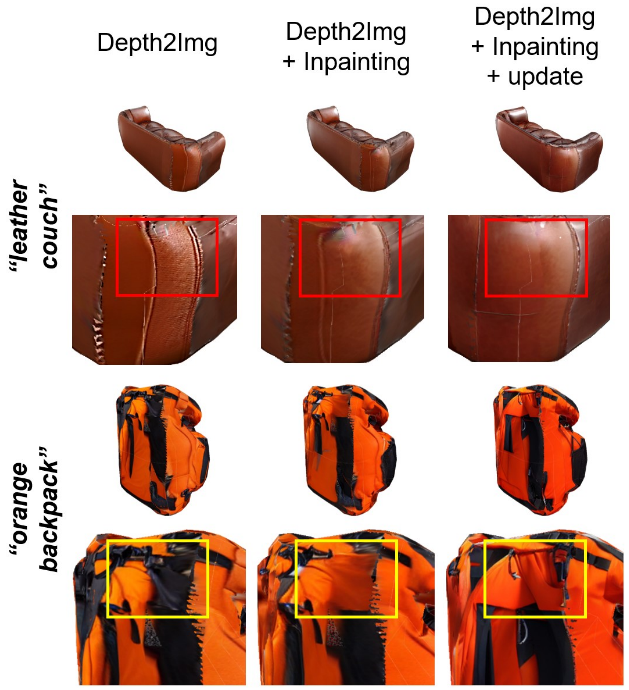

Figure 7: The proposed inpainting and update technique can effectively generate more consistent textures and eliminate stretched and blurry artifacts. Note that there is no refinement in this ablation.

| # views | 0 | 5 | 10 | 15 | 20 |
|---|---|---|---|---|---|
| ↓ FID | 37.09 | 36.67 | 36.39 | 35.98 | **35.68** |
| ↓ KID (x$10^{-3}$) | 8.78 | 8.31 | 8.12 | 7.98 | **7.74** |

Table 5: We quantitatively study the effect of selecting different number of viewpoints in the refinement stage. Refining the synthesis results with more viewpoints improves the texture quality.

**Does viewpoint selection in refinement stage help?** We compare our results with different refinement settings in Tab. 5. When the input geometries are painted with initial textures from the generation stage, the blurry artifacts and projection seams are usually not eliminated due to a limited number of viewpoints. As can be seen in Fig. 8, such flaws can be minimized by refining with more viewpoints. We also showcase the effectiveness of the automatic viewpoint selection technique, as the refinement process does not require any manual efforts with defining and fine-tuning the viewpoint sequence for different shapes.

### 4.6. Limitations.

While our method has shown the capability to produce high-quality 3D textures, we have observed that it tends to produce textures with shading effects from the diffusion backbone. Although this issue can be addressed by carefully fine-tuning the input prompts, doing so requires additional human engineering effort and may not scale well to massive generation targets. One potential solution is to fine-tune the diffusion model to remove the shading from textures. We acknowledge this challenge and leave it to future work to explore this possibility.

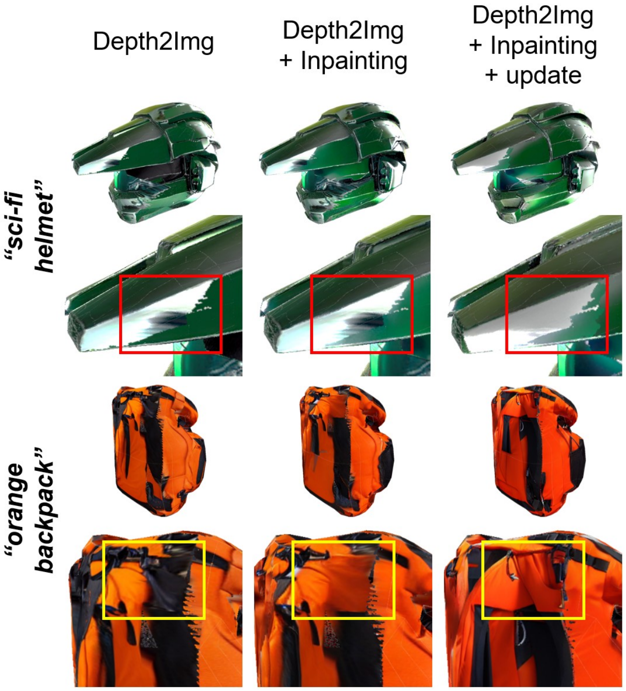

Figure 8: The proposed automatic viewpoint selection method further improves the texture quality by gradually removing the remaining artifacts from the generation stage.

## 5. Conclusion

In this paper, we present a novel method, Text2Tex, for synthesizing high quality textures for 3D meshes from the given text prompts. Our approach leverages a depth-aware image inpainting diffusion model to progressively generate high-resolution partial textures from multiple viewpoints. To avoid accumulating inconsistent and stretching artifacts across viewpoints, we dynamically segment the rendered view into a generation mask, which effectively guides the diffusion model to generate and update the corresponding partial textures. Furthermore, we propose an automatic viewpoint sequence generation scheme that utilizes the generation mask to automatically determine the next best view for refining the generated textures. Extensive experiments demonstrate that our method can effectively synthesize consistent and highly detailed 3D textures for various object geometries without extra manual effort. Overall, we hope our work can inspire more future research in the area of text-to-3D synthesis.

# Acknowledgements

This work was supported by the ERC Starting Grant Scan2CAD (804724) and the German Research Foundation (DFG) Research Unit "Learning and Simulation in Visual Computing". We further thank Guy Gafni and Manuel Dahnert for the helpful discussions, and Angela Dai for the video voiceover.

# References

[1] Stable diffusion. https://github.com/Stability-AI/stablediffusion, 2022. 1, 2, 4, 5

[2] Rameen Abdal, Hsin-Ying Lee, Peihao Zhu, Menglei Chai, Aliaksandr Siarohin, Peter Wonka, and Sergey Tulyakov. 3davatargan: Bridging domains for personalized editable avatars. In *CVPR*, 2023. 2

[3] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *ICML*, 2018. 2

[4] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *ICLR*, 2018. 2, 6

[5] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022. 2

[6] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, 2021. 2

[7] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 5

[8] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. ScanRefer: 3D object localization in RGB-D scans using natural language. In *European Conference on Computer Vision*, pages 202–221. Springer, 2020. 2

[9] Dave Zhenyu Chen, Ronghang Hu, Xinlei Chen, Matthias Nießner, and Angel X Chang. Unit3d: A unified transformer for 3d dense captioning and visual grounding. *arXiv preprint arXiv:2212.00836*, 2022. 2

[10] Dave Zhenyu Chen, Qirui Wu, Matthias Nießner, and Angel X Chang. D 3 net: A unified speaker-listener architecture for 3d dense captioning and visual grounding. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 487–505. Springer, 2022. 2

[11] Zhenyu Chen, Ali Gholami, Matthias Nießner, and Angel X Chang. Scan2cap: Context-aware dense captioning in rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3193–3203, 2021. 2

[12] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, 2019. 1, 2

[13] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander Schwing, and Liangyan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *CVPR*, 2023. 2

[14] Zezhou Cheng, Menglei Chai, Jian Ren, Hsin-Ying Lee, Kyle Olszewski, Zeng Huang, Subhransu Maji, and Sergey Tulyakov. Cross-modal 3d shape generation and manipulation. In *ECCV*, 2022. 2

[15] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. In *ECCV*, 2022. 2

[16] Angela Dai, Yawar Siddiqui, Justus Thies, Julien Valentin, and Matthias Nießner. Spsg: Self-supervised photometric scene generation from rgb-d scans. In *CVPR*, 2021. 2, 6, 7

[17] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. *arXiv preprint arXiv:2212.08051*, 2022. 2, 5

[18] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021. 2

[19] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 2

[20] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. In *Advances In Neural Information Processing Systems*, 2022. 2

[21] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NIPS*, 2014. 2

[22] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d aware generator for high-resolution image synthesis. In *ICLR*, 2022. 2

[23] 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 *NeurIPS*, 2017. 2, 6

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 2, 3

[25] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *arXiv preprint arXiv:2106.15282*, 2021. 2

[26] Ronghang Hu and Amanpreet Singh. Unit: Multimodal multitask learning with a unified transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1439–1449, 2021. 2

[27] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 2

[28] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276*, 2022. 2

[29] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fi-

dler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022. 2

[30] Chieh Hubert Lin, Hsin-Ying Lee, Willi Menapace, Menglei Chai, Aliaksandr Siarohin, Ming-Hsuan Yang, and Sergey Tulyakov. Infinicity: Infinite-scale city synthesis. *arXiv preprint arXiv:2301.09637*, 2023. 2

[31] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022. 4

[32] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *CVPR*, 2021. 2

[33] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *CVPR*, 2023. 2, 6, 7

[34] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *CVPR*, 2022. 2, 5, 6, 7

[35] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2

[36] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. AutoSDF: Shape priors for 3d completion, reconstruction and generation. In *CVPR*, 2022. 2

[37] Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Tiberiu Popa. Clip-mesh: Generating textured meshes from text using pretrained image-text models. In *SIGGRAPH Asia*, 2022. 2, 6, 7

[38] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. *arXiv preprint arXiv:2212.01206*, 2022. 1

[39] Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*, pages 7220–7229. PMLR, 2020. 1

[40] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, 2021. 2

[41] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021. 2

[42] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *ICCV*, 2019. 6, 7

[43] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. In *CVPR*, 2022. 2

[44] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[45] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, 2021. 2

[46] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2023. 2

[47] 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 *ICLR*, 2021. 2

[48] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 1

[49] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021. 2

[50] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 5

[51] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. *arXiv preprint arXiv:2302.01721*, 2023. 2

[52] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 3

[53] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. 2022. 2

[54] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *NeurIPS*, 2020. 2

[55] Aliaksandr Siarohin, Willi Menapace, Ivan Skorokhodov, Kyle Olszewski, Jian Ren, Hsin-Ying Lee, Menglei Chai, and Sergey Tulyakov. Unsupervised volumetric animation. In *CVPR*, 2023. 2

[56] Yawar Siddiqui, Justus Thies, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. Texturify: Generating textures on 3d shape surfaces. In *ECCV*, 2022. 2, 5, 6, 7

[57] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650, 2022. 2

[58] Ivan Skorokhodov, Aliaksandr Siarohin, Yinghao Xu, Jian Ren, Hsin-Ying Lee, Peter Wonka, and Sergey Tulyakov. 3d generation on imagenet. In *ICLR*, 2023. 2

[59] Edward J Smith and David Meger. Improved adversarial systems for 3d object generation and reconstruction. In *CoRL*, 2017. 2

[60] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022. 2

[61] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T Freeman, and Joshua B Tenenbaum. Learning shape priors for single-view 3d completion and reconstruction. In *Proceedings of the European Conference on*

*Computer Vision (ECCV)*, pages 646–662, 2018. 1

[62] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, and Ying Nian Wu. Learning descriptor networks for 3d shape synthesis and analysis. In *CVPR*, 2018. 2

[63] Yinghao Xu, Menglei Chai, Zifan Shi, Sida Peng, Ivan Skorokhodov, Aliaksandr Siarohin, Ceyuan Yang, Yujun Shen, Hsin-Ying Lee, Bolei Zhou, et al. Discoscene: Spatially disentangled generative radiance fields for controllable 3d-aware scene synthesis. In *CVPR*, 2023. 2

[64] Linjie Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. A large-scale car dataset for fine-grained categorization and verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3973–3981, 2015. 6

[65] Rui Yu, Yue Dong, Pieter Peers, and Xin Tong. Learning texture generators for 3d shape collections from internet photo sets. In *BMVC*, 2021. 2, 6, 7

[66] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. *arXiv preprint arXiv:2210.06978*, 2022. 1

[67] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 2, 4

[68] Song-Hai Zhang, Yuan-Chen Guo, and Qing-Wen Gu. Sketch2model: View-aware 3d modeling from single free-hand sketches. In *CVPR*, 2021. 2