DreamMesh: Jointly Manipulating and Texturing Triangle Meshes for Text-to-3D Generation

Haibo Yang[1,2*], Yang Chen[3], Yingwei Pan[3], Ting Yao[3], Zhineng Chen[1,2†], Zuxuan Wu[1,2], Yu-Gang Jiang[1,2], and Tao Mei[3]

[1] School of Computer Science, Fudan University
[2] Shanghai Collaborative Innovation Center of Intelligent Visual Computing
[3] HiDream.ai Inc.
yanghaibo.fdu@gmail.com, {c1enyang, pandy, tiyao}@hidream.ai, {zhinchen, zxwu, ygj}@fudan.edu.cn, tmei@hidream.ai

Abstract. Learning radiance fields (NeRF) with powerful 2D diffusion models has garnered popularity for text-to-3D generation. Nevertheless, the implicit 3D representations of NeRF lack explicit modeling of meshes and textures over surfaces, and such surface-undefined way may suffer from the issues, e.g., noisy surfaces with ambiguous texture details or cross-view inconsistency. To alleviate this, we present DreamMesh, a novel text-to-3D architecture that pivots on well-defined surfaces (triangle meshes) to generate high-fidelity explicit 3D model. Technically, DreamMesh capitalizes on a distinctive coarse-to-fine scheme. In the coarse stage, the mesh is first deformed by text-guided Jacobians and then DreamMesh textures the mesh with an interlaced use of 2D diffusion models in a tuning free manner from multiple viewpoints. In the fine stage, DreamMesh jointly manipulates the mesh and refines the texture map, leading to high-quality triangle meshes with high-fidelity textured materials. Extensive experiments demonstrate that DreamMesh significantly outperforms state-of-the-art text-to-3D methods in faithfully generating 3D content with richer textual details and enhanced geometry. Our project page is available at https://dreammesh.github.io.

Keywords: Text-to-3D Generation · Diffusion Models · Triangle Meshes

1 Introduction

Diffusion models [16, 17, 55] have emerged as the basis of the powerful modern generative networks for producing realistic and diverse visual content (e.g., images and videos [8, 9, 31]). In between, a massive leap forward has been attained in text-driven visual content generation tasks, e.g., text-to-image generation [29, 34, 36, 38, 39] and text-to-video generation [18, 50, 53]. The success is attributed to several factors like billion-level multi-modal data and scalable

* This work was performed when Haibo Yang was visiting HiDream.ai as a research intern.
† Corresponding author.

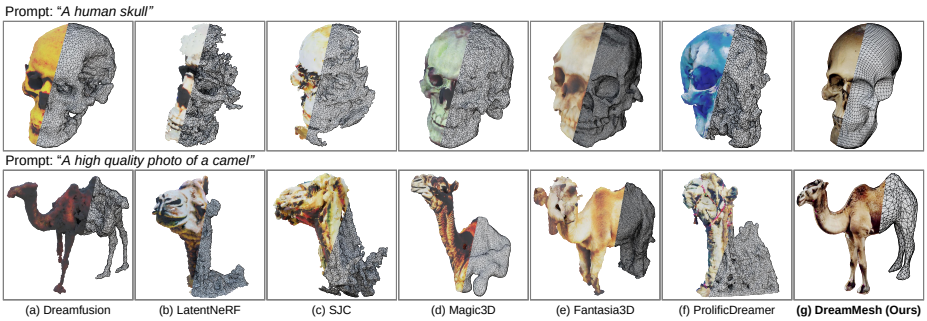

Fig. 1: Existing methods [4, 25, 26, 33, 47, 48] mostly hinge on implicit or hybrid 3D representation and produce noisy surfaces. Instead, our DreamMesh pivots on completely explicit 3D representation, yielding high-quality 3D meshes that exhibit clean, organized topology, devoid of any redundant vertices & faces.

denoising diffusion-based generative modeling. Nevertheless, it is not trivial to directly train a robust 3D-specific diffusion model for text-to-3D generation, since the paired text-3D data is relatively scarce, and 3D scenes have more complex geometric structures and multi-view visual appearances than 2D images.

The recent advance of Dreamfusion [33] nicely sidesteps the requirement of massive paired text-3D data for text-to-3D generation task, and learns implicit 3D scene representation (NeRF [28]) with only 2D diffusion models pre-trained over images. The core learning objective is to optimize implicit 3D scene with 2D observations of each sampled views derived from 2D diffusion models via Score Distillation Sampling (SDS). Despite having impressive quantitative results through SDS, qualitative analysis shows that such text-to-3D generation often results in cross-view inconsistency or ambiguous texture details due to the intrinsic bias of 2D diffusion priors. Later on, a series of efforts [4, 25, 26, 47, 48] have been dedicated to upgrading the 2D diffusion priors in SDS with 3D-aware knowledge, aiming to strengthen the capabilities to produce cross-view consistent 3D scene. Note that these text-to-3D works predominantly revolve around implicit 3D scene representation of density-based geometry with undefined surface boundaries. As shown in Figure 1, this surface-undefined framing easily leads to noisy extracted surfaces and over-saturated/over-smoothed textures. Moreover, the learnt 3D assets with implicit 3D scene fail to be directly integrated into graphics pipeline, and necessitate additional conversion from implicit to explicit 3D scene. The conversion might inject more noise over surfaces, thereby hindering the usage particularly in various high-quality 3D applications.

To address these challenges, our work shapes a new way to frame text-to-3D generation on the basis of completely explicit 3D scene representation of the ubiquitous and well-defined surface (triangle meshes). We propose a novel text-to-3D framework, namely DreamMesh, that executes the learning of textured triangle meshes into two stages. Specifically, in the first coarse stage, DreamMesh deforms the triangle meshes by text-guided Jacobians, obtaining globally smooth

coarse mesh. Next, the corresponding coarse texture is attained through a tuning-free process with an interlaced use of pre-trained 2D diffusion models. In the second fine stage, DreamMesh jointly manipulates the coarse mesh and refines the coarse texture map. This scheme learns the surface and material/texture of explicit 3D representation in a coarse-to-fine fashion. Eventually the explicit 3D model by DreamMesh faithfully reflects the high-quality geometry (clean and organized topology) with rich texture details (see Figure 1).

In summary, we have made the following contributions: (1) We novelly frame text-to-3D generation based on a completely explicit 3D scene representation of triangle meshes, which is shown capable of mitigating the issue associated with implicit 3D scenes and learning more smooth surfaces. (2) The exquisitely designed coarse-to-fine strategy pivoting on explicit 3D scene representation is shown able to facilitate the manipulation and texturing of triangle meshes. (3) The proposed DreamMesh has been analyzed and verified through extensive experiments over a comprehensive text-to-3D benchmark (T^3Bench [15]), demonstrating superior results when compared to state-of-the-art approaches.

2 Related Works

Text-to-3D Generation. Recently, the text-to-3D generation has drawn increasing research attention. Pioneering works [33, 47] utilize pre-trained 2D diffusion models to accomplish text-to-3D generation in a zero-shot fashion, mitigating the reliance on massive training data and becoming the mainstream. The key technique underpinning these methods is score distillation sampling (SDS), which enables distilling knowledge from the 2D diffusion model to optimize an underlying 3D representation (e.g., NeRF [28]) and showcases remarkable 3D generation capability. Subsequently, there has been a series of related works [4–7, 21, 25, 26, 41, 46, 48, 49, 52, 54] that continue to refine and strengthen this methodology in different ways. For instance, Latent-NeRF [26] and Control3D [6] incorporate additional user-provided sketch mesh or image to guide the text-to-3D generation process. Magic3D [25] and Fantasia3D [4] upgrade the implicit NeRF representation used in DreamFusion to an implicit-explicit hybrid 3D representation (i.e., DMTet [40]) for SDS optimization on higher-resolution renderings. VP3D [7] leverages 2D visual prompt to boost text-to-3D generation.

Although the aforementioned works can generate high-quality renderings, they all predominantly adopt implicit (NeRF) or implicit-explicit hybrid (DMTet) 3D representation. Integrating them into the mainstream graphics pipeline requires an additional conversion from implicit/hybrid 3D representation to widely used textured mesh. Unfortunately, this conversion may result in sub-optimal results due to the absence of explicit modeling meshes and textures during optimization, which prevents the usage of these methods in real-world deployments. In contrast, we formulate the text-to-3D generation process from a new perspective based on the completely explicit 3D representation of triangle meshes, leading to more clean & well-organized surfaces and photo-realistic textures that can be seamlessly compatible with existing 3D engines (e.g., Blender).

Text-Driven Shape Manipulation/Texturing. Manipulating and texturing 3D shapes are key components in animation creation and computer-aided design pipelines, gaining a surge of interest in the literature. Classical approaches [19,42,43,45] perform shape manipulation by predicting mesh deformations from user-provided handles and cast this problem as an optimization task, where the source mesh is iteratively deformed to minimize the fitting error from the source to target shape. Instead of controlling the deformation through handle movements, a recent work [13] guides the deformation process solely from a text prompt by utilizing a pre-trained CLIP model [35] and differentiable rendering. Another direction of research focuses on text-driven 3D shape texturing that automatically generates textures for 3D bare meshes from the given text prompt. State-of-the-art methods [3,37] utilize pre-trained diffusion models (e.g., depth-to-image diffusion model and inpainting diffusion model) to "paint" the input bare mesh with generated textures. Unlike the aforementioned approaches, our work frames text-to-3D generation by jointly manipulating and texturing triangle meshes. Instead of formulating either a mesh deformation task or a mesh texturing problem in [3,12,13,37,49], we uniquely look into the text-to-3D problem via generating explicit high-quality triangle meshes on the input text prompt.

3 Approach

3.1 Preliminaries

We first briefly review the typical score distillation sampling (SDS) method, and then discuss the relations and differences between our DreamMesh and related methods based on implicit-explicit hybrid 3D representation.

Score Distillation Sampling (SDS). SDS is first introduced by Dreamfusion [33] that leverages pre-trained text-to-image diffusion models to enable zero-shot text-to-3d generation. Specifically, DreamFusion employs Neural Radiance Fields (NeRF) [28] to parameterize the implicit 3D scene as θ. Next, a differentiable renderer is utilized to render an image x from the 3D scene. In an effort to distill the knowledge of 2D diffusion model (e.g., Imagen [39]) into 3D scene, random noise ϵ is initially added to the image x:

$$x_t = \sqrt{\bar{\alpha}_t}x + \sqrt{1-\bar{\alpha}_t}\epsilon, \tag{1}$$

where $\epsilon \sim \mathcal{N}(0, I)$, and $\bar{\alpha}_t$ is a time-variant constant. After that, DreamFusion employs the denoiser of diffusion model (parameterized as ϵ_ϕ) to estimate the added noise ϵ from the noisy image x_t. The 3D scene parameters θ are thus updated according to the per-pixel gradient of difference between the actual and predicted noise:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, x) = \mathbb{E}_{t,\epsilon}\left[w(t)(\epsilon_\phi(x_t; y, t) - \epsilon)\frac{\partial x}{\partial \theta}\right], \tag{2}$$

where $w(t)$ is a weighting function and y is the input text prompt. In this way, the pixel-level gradient is back-propagated to optimize the 3D scene, thereby driving the learnt 3D scene to resemble the input text prompt.

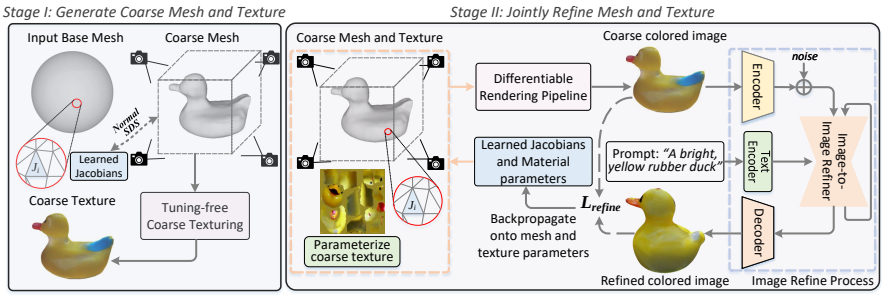

Fig. 2: An overview of our DreamMesh that fully capitalizes on explicit 3D scene representation (triangle meshes) for text-to-3D generation in a coarse-to-fine scheme. In the first coarse stage, DreamMesh learns text-guided Jacobians matrices to deform a base mesh into the coarse mesh, and then textures it through a tuning-free process. In the second fine stage, both coarse mesh and texture are jointly optimized, yielding high-quality mesh with high-fidelity texture.

Implicit-Explicit Hybrid 3D Representations. Recent advances in text-to-3D generation predominantly employ implicit representations [28] for modeling 3D scenes. A notable limitation of these techniques is the low-quality explicit mesh with noisy surfaces extracted from implicit fields. To alleviate this issue, several methods [4, 25, 48] employ implicit-explicit hybrid 3D representation (DMTet [40]). Nevertheless, the meshes extracted from these DMTet-based methods still suffer from the problems such as excessive faces and poor topological structures. Such downside severely hinders the seamless integration of these meshes into traditional graphics rendering pipelines, thereby limiting their deployments in standard visualization and animation processes. As an alternative, our DreamMesh pivots on completely explicit 3D representation (triangle meshes) for modeling 3D objects, thereby getting rid of the gap between implicit and explicit representations. Formally, let $\mathcal{M}_0$ be a given base triangular mesh, which comprises a set of vertices $\mathcal{V} \in \mathbb{R}^{n\times 3}$ and triangular faces $T \in \mathbb{R}^{m\times 3}$. The base mesh can be a basic sphere, user-provided, or a low-quality mesh generated by 3D generative methods [10, 20, 24, 30]. Through joint manipulation and texturing of triangle meshes, our DreamMesh learns and refines the 3D mesh and the corresponding mesh textures $\mathcal{T}$ that adhere to the given text prompt y.

3.2 DreamMesh Optimization

In this section, we elaborate our DreamMesh, which frames text-to-3D generation based on completely explicit 3D representation in a coarse-to-fine fashion. Figure 2 depicts the detailed framework, consisting of two stages: the coarse stage to produce coarse mesh and texture, and the fine stage to jointly refine mesh and texture with a diffusion-based 2D image-to-image refiner.

Stage I: Generate Coarse Mesh and Texture. In this stage, we aim to generate a coarse triangle mesh and textured map that respects the input text prompt. To achieve this, we first deform a base mesh into the coarse mesh and then texture it through a tuning-free process. Practically, we use Shap-E [20] outputs as the input base mesh. Note that Shap-E is trained on millions of paired text-3D data and can easily produce 3D objects with reasonable geometry. That makes Shpe-E an ideal choice to serve as the initial base mesh.

Coarse Mesh Deformation. Given a base triangular mesh $\mathcal{M}_0$, DreamMesh first learns to deform it into a target triangular mesh that faithfully matches the input text prompt. Technically, we formulate this learning process as the optimization of a displacement map $\mathbf{D} : \mathbb{R}^{3\times3} \rightarrow \mathbb{R}^{3\times3}$ over the vertices. Such piecewise linear mapping $\mathbf{D}$ of a mesh can be defined by assigning a new position to each one of the vertices: $\mathcal{V}_i \rightarrow D_i$. Nevertheless, direct optimization on the vertex positions of a triangular mesh can easily suffer from degeneracy and local minima, and thus overly distorts the original shape. To alleviate this issue, we take the inspiration from [1, 13] and parameterize the mesh deformation by using a set of per-triangle Jacobians $J_i = \mathbf{D}\nabla_i^T (J_i \in \mathbb{R}^{3\times3})$, where ∇_i^T is the gradient operator of triangle $t_i \in T$. Given an arbitrary assignment of input matrix $M_i \in \mathbb{R}^{3\times3}$ for every triangle, we can achieve new vertex positions $\mathbf{D}^*$ whose Jacobians $J_i = \mathbf{D}^*\nabla_i^T$ are least-squares closest to M_i. And we can easily obtain the deformed vertex positions $\mathbf{D}^*$ by solving linear system:

$$\mathbf{D}^* = L^{-1}\mathcal{A}\nabla^T M, \tag{3}$$

where $\mathcal{A}$ is the mesh's mass matrix, L is the mesh's Laplacian, and M is the stacking of the input matrices M_i. Accordingly, the optimization of the deformation mapping ($\mathbf{D}$) can be interpreted as the optimization of the learnable Jacobians matrices M_i. In practice, we initialize these Jacobians matrices as identity matrices, where $\mathbf{D}^*$ is inherently established as the identity mapping. Please refer to [1] for the full technical details.

Coarse Diffusion Guidance. To achieve text-driven deformation $\mathbf{D}^*$ that aligns with input text prompt, we exploit the powerful text-to-image diffusion model (Stable Diffusion [38]) as coarse diffusion guidance to facilitate Jacobians deformation. Specifically, given the base mesh $\mathcal{M}_0$ and deformation mapping $\mathbf{D}^*$, we utilize a differentiable renderer g_n [23] to render a normal map n:

$$n = g_n(\mathbf{D}^*(\mathcal{M}_0), c), \tag{4}$$

where c represents a camera pose that is arbitrarily sampled within the spherical coordinate system. Such random sampling strategy ensures a uniform distribution of camera poses across the sphere, providing comprehensive coverage and variability. Next, during t-th timestep of diffusion process, we encode the rendered normal map n into the latent space to obtain the latent code z^n, and add Gaussian noise ϵ to get z_t^n. The typical latent space SDS loss is thus utilized to optimize the deformation $\mathbf{D}^*$ by measuring the gradient w.r.t. $\mathbf{D}^*$ as:

$$\nabla_{\mathbf{D}^*}\mathcal{L}_{\text{SDS}}(\phi, n) = \mathbb{E}\left[w(t)(\hat{\epsilon}_\phi(z_t^n; y, t) - \epsilon)\frac{\partial n}{\partial \mathbf{D}^*}\frac{\partial z^n}{\partial n}\right], \tag{5}$$

where $\hat{\epsilon}_\phi$ denotes denoiser in Stable Diffusion. It is worthy to note that instead of using Stable Diffusion's image encoder, here we exploit a downsampled version of the normal map n as the latent code [4,26], which leads to a faster convergence of $\mathbf{D}^*$. By randomly sampling views and backpropagating the gradient in Eq. (5) to the learnable parameters in $\mathbf{D}^*$, this way eventually achieves a target coarse mesh $\mathcal{M}_1 = \mathbf{D}^*(\mathcal{M}_0)$ that resembles the input text prompt.

Coarse Texture Generation. Next, we target for producing realistic coarse textures for the learnt coarse mesh $\mathcal{M}_1$. We apply a tuning-free approach to progressively generate coarse textures on the 3D triangle mesh with an interlaced use of pre-trained 2D diffusion models [37]. In particular, the texture is represented as an atlas ($\mathcal{T}_0$) learnt through UV mapping process [51]. At the initialization step, we use a differentiable renderer $\mathcal{R}$ [11] to render a depth map $\mathcal{D}_0$ from an arbitrary initial viewpoint v_0, and use a pretrained depth-to-image diffusion model $\mathcal{M}_{depth}$ [38] conditioned on the rendered depth map to generate an initial colored image I_0. The generated image I_0 is then projected back to the texture atlas $\mathcal{T}_0$ to color the shape's visible parts from v_0. Following this initialization step, we iteratively change the viewpoint and alternatively use a pretrained inpainting diffusion model $\mathcal{M}_{paint}$ [38] or $\mathcal{M}_{depth}$ to generation new colored image. These colored images are projected back onto the initial texture $\mathcal{T}_0$. We repeat this process until a complete coarse texture map $\mathcal{T}_1$ is formed.

Stage II: Jointly Refine Mesh and Texture. Recall that at the first coarse stage, the optimization process of coarse mesh deformation solely focuses on the primary mesh irrespective of any texture. Such process might inevitably simulate textured results and lead to excessive modifications of meshes. Meanwhile, the coarse texture generation in first stage also encounters the inconsistency issue across all viewpoints. We speculate that the sub-optimal texturing results might be caused by the tuning-free texturing strategy that performs progressive texture mapping starting from a singular viewpoint, and it is non-trivial to maintain both local and global consistency. To alleviate these, we novelly devise a fine stage to jointly optimize both the mesh and texture with fine diffusion guidance derived from a pretrained diffusion-based image refiner [32]. This innovative stage establishes a symbiotic relationship between the learnt meshes and textures, harmonizing the enhancement that contributes to the overall realism and consistency of synthetic 3D content.

Technically, in the fine stage, we adopt the same mesh deformation methodology as in the coarse stage, i.e., the optimization of Jacobian Matrices, to refine the coarse mesh $\mathcal{M}_1$. However, different from the tuning-free texturing strategy in coarse stage, here we concurrently parameterize the coarse texture map $\mathcal{T}_1$, and trigger a joint optimization of $\mathcal{M}_1$ and $\mathcal{T}_1$ in the fine stage. By doing so, we employ a differentiable rendering pipeline g, which includes a sequence of mesh operations, a rasterizer, and a deferred shading stage [14] to render a coarse colored image x_{coarse} derived from the deformering mesh $\mathcal{M}_1$ and parameterized texture map $\mathcal{T}_1$, conditioned on a random camera pose c:

$$x_{coarse} = g(\mathbf{D}^*_{fine}(\mathcal{M}_1), \mathcal{T}_1, c). \tag{6}$$

Fine Diffusion Guidance. Our fine stage necessitates intricate adjustments to the coarse mesh structure and dedicated efforts to enhance texture consistency. One natural way to optimize such fine process is to use the same coarse diffusion guidance (SDS loss) as in coarse stage for supervision. Nevertheless, such way will result in various artifacts like oversaturated color blocks. Instead, we excavate the fine diffusion guidance by additionally refining rendered coarse colored image x_{coarse} with diffusion-based image refiner $\mathcal{E}$ [32]. This refined colored image $x_{refine} = \mathcal{E}(x_{coarse}, y)$ is further utilized to guide the joint optimization of mesh and texture through Mean Squared Error (MSE) loss:

$$\begin{aligned} &\mathcal{L}_{refine}(\mathbf{D}^*_{fine}, \mathcal{T}_1) = ||x_{refine} - x_{coarse}||_2^2 \\ &= ||\mathcal{E}(g(\mathbf{D}^*_{fine}(\mathcal{M}_1), \mathcal{T}_1, c), y) - g(\mathbf{D}^*_{fine}(\mathcal{M}_1), \mathcal{T}_1, c)||_2^2. \end{aligned} \tag{7}$$

By minimizing this objective, our DreamMesh enforces the rendered image x_{coarse} visually similar as the refined image x_{refine} that faithfully matches with text prompt, thereby yielding high-quality mesh with high-fidelity texture map.

Discussion. Some related works [3, 13, 37] also explore text-driven mesh deformation or texturing, while our DreamMesh targets for a different task of text-to-3D generation. Instead of a simple combination of existing mesh deformation & texturing techniques (see degenerated results in Fig. 5), our DreamMesh novelly upgrades mesh deformation with geometry-aware supervision and further bridges both worlds by jointly optimizing mesh and texture with a fine-grained image-to-image refiner in fine stage.

4 Experiments

4.1 Experimental Settings

Implemention Details. At the coarse stage, we utilize an Adam optimizer with a learning rate of 2×10^{-3} and render 12 normal maps per iteration. The coarse textures are generated from 10 different viewpoints. In the fine stage, we set the learning rate as 2×10^{-3} for mesh optimization and 1×10^{-2} for texture material refinement. The diffusion-based image refiner performs denoising operations over 15 steps to produce refined images. The whole experiment is conducted on a single NVIDIA RTX 3090 GPU, and the learning process of each sample takes approximately 30 minutes.

Dataset. Most existing text-to-3D generation methods solely perform case studies and user surveys for evaluation, but lack quantitative assessment due to the absence of standard benchmark. Thanks to the newly released T^{3}Bench [15], we perform quantitative comparisons over this first comprehensive benchmark for text-to-3D generation. Specifically, T^{3}Bench contains 100 test prompts for each of three categories (single object, single object with surroundings, and multiple objects). Two automatic metrics are designed to evaluate the subjective quality and textual alignment, based on the rendered multi-view images generated from 3D models. The quality metric combines multi-view text-image scoring with regional convolution to assess both quality and view consistency. The alignment

Table 1: Quantitative comparison between our DreamMesh and various text-to-3D generation approaches on T^{3}Bench [15] benchmark.

	3D Representation	*Single Object*			*Single Object with Surroundings*			*Multiple Objects*		
		Quality	Alignment	Average	Quality	Alignment	Average	Quality	Alignment	Average
Dreamfusion [33]	Implict Representation (NeRF [28])	24.9	24.0	24.4	19.3	29.8	24.6	17.3	14.8	16.1
LatentNeRF [26]		34.2	32.0	33.1	23.7	37.5	30.6	21.7	19.5	20.6
SJC [47]		26.3	23.0	24.7	17.3	22.3	19.8	17.7	5.8	11.7
Magic3D [25]	Implicit-Explicit Hybrid Representation (DMTet [40])	38.7	35.3	37.0	29.8	41.0	35.4	26.6	24.8	25.7
Fantasia3D [4]		29.2	23.5	26.4	21.9	32.0	27.0	22.7	14.3	18.5
ProlificDreamer [48]		51.1	47.8	49.4	42.5	47.0	44.8	45.7	25.8	35.8
DreamMesh (Ours)	Explicit Representation (Triangle Mesh)	**55.6**	**53.8**	**54.7**	**43.1**	**54.3**	**48.7**	**47.6**	**30.8**	**39.2**

metric first employs 3D-to-text caption model to achieve multi-view captions and then leverages Large Language Model (GPT-4) to merge captions into 3D caption for text-3D alignment assessment.

Compared Methods. To empirically verify the merit of our DreamMesh, we include six state-of-the-art approaches for comparison. Specifically, **DreamFusion** [33], **LatentNeRF** [26], and **SJC** [47] fully hinge on implicit 3D representation (NeRF [28]) for text-to-3D generation. **Magic3D** [25] upgrades DreamFusion with additional stage that capitalizes on implicit-explicit hybrid 3D representation (DMTet [40]) to enhance texture details. **Fantasia3D** [4] disentangles geometry and appearance modeling in two stages, i.e., first generating meshes based on DMTet and then leveraging Bidirectional Reflectance Distribution Function (BRDF) to produce textures. **ProlificDreamer** [48] extends Magic3D by generalizing SDS in the variational formulation, aiming to alleviate the restricted diversity issue rooted in typical SDS.

4.2 Quantitative Results

Table 1 summarizes the quantitative performance comparisons over three categories of T^{3}Bench benchmark between our DreamMesh and six state-of-the-art approaches. Overall, for each category of text prompts, our DreamMesh consistently achieves better performances against the existing methods across all metrics, including both implicit 3D representation-based methods (Dreamfusion, LatentNeRF, SJC) and implicit-explicit hybrid 3D representation-based methods (Magic3D, Fantasia3D, ProlificDreamer). In particular, the average score of quality and alignment of DreamMesh can reach 54.7%, 48.7%, and 39.2% for each category, which leads to the absolute improvement of 5.3%, 3.9%, and 3.4% against the best competitor ProlificDreamer. The results clearly demonstrate the key advantage of joint manipulating and texturing based on completely explicit 3D representation to facilitate text-to-3D generation.

More specifically, Dreamfusion enables a zero-shot solution of optimizing implicit 3D representation with 2D diffusion priors, yielding promising results even under challenging prompts in the categories of "single object with surroundings" and "multiple objects"). SJC remoulds Dreamfusion by performing score jacobian chaining within the voxel version of NeRF [2, 44], which easily results in

degraded 3D models with a significant amount of floating density. In contrast, LatentNeRF upgrades Dreamfusion with a coarse-to-fine paradigm by using the implicit 3D representations of latent NeRF that is more tailored to the 2D latent diffusion model (Stable Diffusion [38]), thereby leading to clear performance boosts. Furthermore, compared to aforementioned three methods that solely exploit implicit 3D representations, Magic3D exhibits better performances by capitalizing on implicit-explicit hybrid 3D representation (DMTet) to learn textured 3D mesh. Fantasia3D also explores DMTet in a decoupled mesh generation stage and leverages BRDF for texture generation that emphasizes rich object textures, while it fails to create complex and high-fidelity meshes (e.g., "multiple objects" category). ProlificDreamer further boosts up the performances by upgrading SDS of Magic3D in variational formulation to address the restricted diversity issue. Nevertheless, ProlificDreamer still relies on implicit-explicit hybrid 3D representation (DMTet) and commonly suffers from noisy surfaces with over-smoothed textures. In contrast, our DreamMesh completely eliminates the use of implicit 3D representation and achieves the best performances through coarse-to-fine strategy pivoting on explicit 3D representation of triangle meshes.

4.3 Qualitative Results

As indicated by these exemplar results in Figure 3, all the methods can generate somewhat reasonable meshes and textures, while our DreamMesh can synthesize higher quality meshes with richer textures that faithfully adhere to text prompts by pivoting on completely explicit 3D representation. For instance, given the first text prompt *"A bright red fire hydrant"*, the implicit 3D representation-based methods (Dreamfusion, LatentNeRF, SJC) produce noisy surfaces and simple textures with obvious deformation. By exploring implicit-explicit hybrid 3D representation for text-to-3D generation, Magic3D, Fantasia3D, and ProlificDreamer further yield more complete and accurate meshes. Nevertheless, these meshes generated via DMTet are inherently complex, necessitating excessive faces and vertices, resulting in unsatisfactory triangle topologies. In contrast, our DreamMesh novelly manipulates and textures well-defined surface (triangle meshes), leading to high-quality textured meshes that reflect clean and well-organized topology with neatly arranged vertices, edges, and faces.

4.4 Experimental Analysis

Ablation Studies. In an effort to study the effectiveness of each design in our DreamMesh, we depict the qualitative results of several ablated runs in Figure 4. DreamMesh$^{-}_{coarse}$ is an alternative version of our coarse stage by simultaneously optimizing mesh and texture from scratch via typical SDS, which easily leads to sub-optimal results. By decoupling mesh deformation and texturing into two separate processes, DreamMesh$_{coarse}$ further enhances the quality of textured meshes. The results validate the effectiveness of decoupled mesh and texture modeling in coarse stage. Nevertheless, DreamMesh$_{coarse}$ still suffers from some

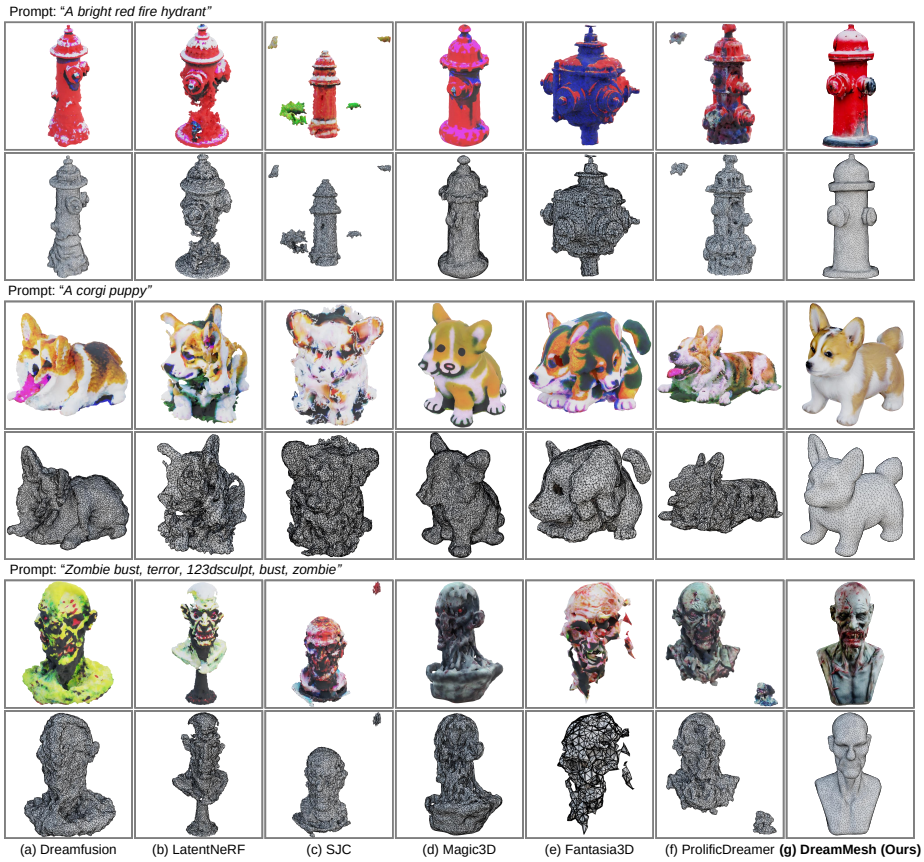

Fig. 3: Qualitative comparison of texture and wireframe results (rendering in Blender) between our DreamMesh and other baseline methods.

overly distorted meshes. DreamMesh upgrades DreamMesh$_{coarse}$ with an additional fine stage to jointly manipulate coarse meshes and refine coarse textures, yielding higher-quality fine meshes and textures.

To further verify the leverage of explicit 3D representation for text-to-3D generation, we also compare our DreamMesh with the run of integrating the state-of-the-art text driven mesh deformation technique [13] and the advanced texturing methods of TEXTure [37] or Text2Tex [3]. Figure 5 shows the comparisons. It is not surprising that TextDeformer and TextDeformer+TEXTure/Text2Tex produce unsatisfactory surfaces and textures since these methods are not particularly tailored for text-to-3D generation. Our DreamMesh, in comparison, introduces more powerful diffusion model for mesh deformation and benefits from the coarse-to-fine optimization scheme, making the mesh more clean and the texture more realistic. Moreover, we conduct quantitative evaluations for the aforementioned runs on a random subset of T^{3}bench (50 prompts) and Ta-

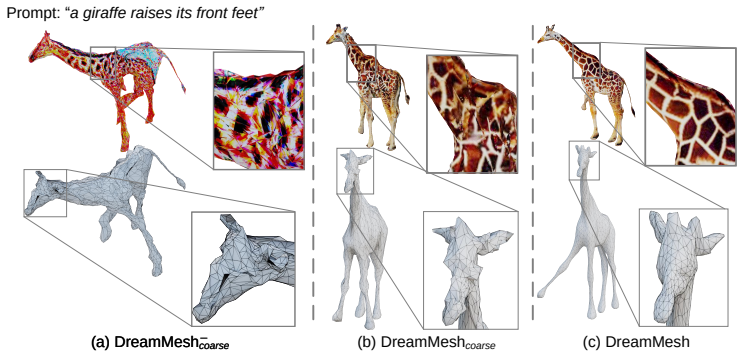

Fig. 4: Ablation study of our DreamMesh given the same text prompt. DreamMesh$^{-}_{coarse}$ is a degraded version of coarse stage that jointly optimizes mesh deformation and textures via SDS. DreamMesh$_{coarse}$ is the complete coarse stage that decouples the learning of coarse meshes and textures. DreamMesh is our full run with both coarse and fine stages.

Table 2: Quantitative comparisons on the T^{3}Bench subset.

	TextDeformer + TEXTure	TextDeformer + Text2Tex	DreamMesh$^{-}_{coarse}$	DreamMesh$_{coarse}$	DreamMesh
Quality	20.2	19.3	24.4	40.5	**47.1**
Alignment	19.0	16.5	22.0	38.5	**45.5**
Average	19.6	17.9	23.2	39.5	**46.3**

ble 2 lists the results. The runs of TextDeformer+TEXTure/Text2Tex indicate poorest quality and alignment, again revealing the weakness of simple combination of mesh deformation and texture techniques for text-to-3D generation. DreamMesh$^{-}_{coarse}$ is inferior to DreamMesh$_{coarse}$, showing that entangled optimizing mesh and texture from scratch is more challenging. Finally, DreamMesh achieves the best performance, validating the effectiveness of our exquisitely designed coarse-to-fine strategy.

Comparison against Typical Mesh-based Methods without Diffusion Model. It is worthy to note that some early attempts (e.g., Text2Mesh [27] and CLIP-Mesh [22]) also explore explicit 3D representation for text-to-3D generation, while no powerful diffusion model is adopted. Figure 6 shows the qualitative comparison between our DreamMesh against Text2Mesh and CLIP-Mesh. In general, our DreamMesh significantly outperforms the conventional mesh-based methods with regard to both mesh/texture quality and text-3D alignment. This confirms the merit of exploiting explicit 3D representation for text-to-3D generation conditioned on powerful 2D diffusion priors.

Comparison against ProlificDreamer with Manual Post-processing. Recall that both implicit and implicit-explicit hybrid representations of NeRF and DMTet can be transformed into explicit meshes by using the Marching Cubes and Marching Tetrahedral layer respectively. However, such automatic

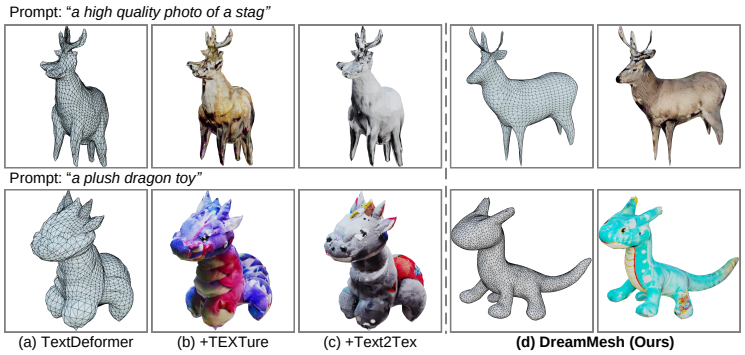

Fig. 5: Comparisons between our DreamMesh and the integration of the state-of-the-art text driven mesh deformation technique [13] and the advanced texturing methods of TEXTure [37] or Text2Tex [3] for text-to-3D generation.

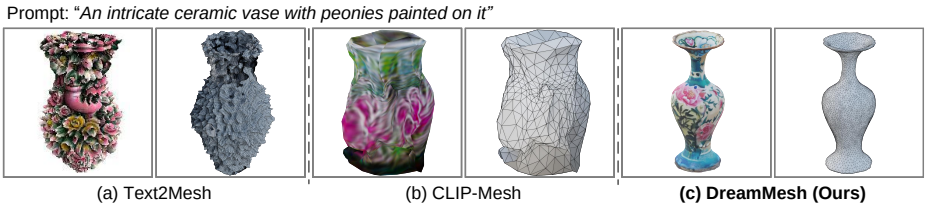

Fig. 6: Qualitative comparison of texture and wireframe results (rendering in Blender) against Text2Mesh and CLIP-Mesh.

conversion commonly injects more noise over surfaces and leads to extremely complex meshes containing a large number of vertices, edges, and faces (e.g., 36,389 faces in Figure 7 (a)). To alleviate this issue, manual post-processing (e.g., cleaning, smoothing and simplification) can be employed to improve the mesh quality, while losing many geometric details (e.g., the missing of nose in Figure 7 (b)). In contrast, as shown in Figure 7 (c), our DreamMesh manages to achieve high-quality textured meshes that exhibit clean and organized topology.

Comparison with Different Base Meshes. Our DreamMesh is able to perform text-to-3D generation based on different kinds of input base meshes. For example, given the input text prompt "A recliner chair", we can generate it directly from a sphere. Alternatively, users can quickly create a rough 3D shape that approximately aligns with text prompt in 3D engines (e.g., Blender). Such user-provided rough mash can be fed into DreamMesh. Additionally, we can take the low-quality mesh automatically generated by 3D generative models (e.g., Shap-E [20]) as the inputs. As shown in Figure 8, all text-to-3D generation results with different base meshes (i.e., basic sphere, user-provided shape, or Shap-E outputs) can produce higher-quality 3D assets, which generally demonstrate the generalization ability of our DreamMesh.

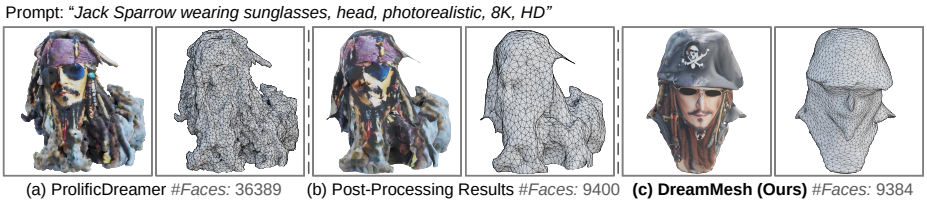

Fig. 7: Qualitative comparison of texture and wireframe results (rendering in Blender) against ProlificDreamer with additional manual post-processing.

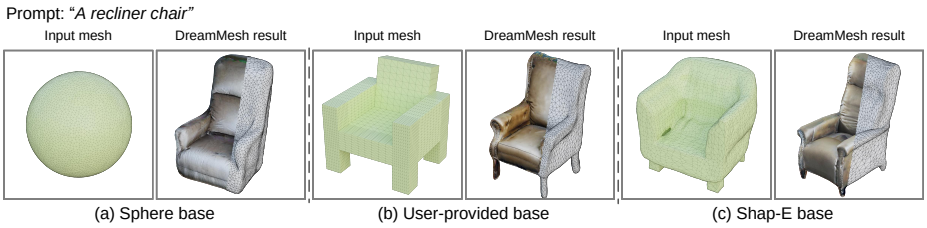

Fig. 8: Text-to-3D generation results (rendering in Blender) of DreamMesh with different input base meshes.

5 Conclusions

In this paper, we propose DreamMesh, a novel framework for text-to-3D generation that fully relies on explicit 3D representations in a coarse-to-fine manner. Specifically, in the coarse stage, we leverage neural jacobian fields to deform a triangle mesh and then texture the generated coarse mesh through a tuning-free process with an interlaced use of pre-trained 2D diffusion models. In the fine stage, we jointly refine the coarse mesh and texture to produce high-quality 3D model with rich texture details and enhanced 3D geometry. We evaluate our proposal on T^3Bench benchmark and demonstrate its superiority over state-of-the-art techniques through both qualitative and quantitative comparisons.

Limitations and Broader Impact. DreamMesh may suffer from multi-face Janus problems in some cases due to the limited 3D awareness of the prior 2D diffusion model. Finetuning the diffusion model on 3D data might alleviate this problem. Since the generated meshes can be seamlessly compatible with existing 3D engines, DreamMesh has great potential to displace creative workers via automation, which may enable growth for the creative industry. Nevertheless, it could also be potentially applied to unexpected scenarios such as generating fake and malicious content, which needs more caution.

Acknowledgement: This work is supported by the National Natural Science Foundation of China (No. 32341012 and No. 62172103).

References

1. Aigerman, N., Gupta, K., Kim, V.G., Chaudhuri, S., Saito, J., Groueix, T.: Neural jacobian fields: Learning intrinsic mappings of arbitrary meshes. In: SIGGRAPH (2022)
2. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: ECCV (2022)
3. Chen, D.Z., Siddiqui, Y., Lee, H.Y., Tulyakov, S., Nießner, M.: Text2tex: Text-driven texture synthesis via diffusion models. In: ICCV (2023)
4. Chen, R., Chen, Y., Jiao, N., Jia, K.: Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In: ICCV (2023)
5. Chen, Y., Chen, J., Pan, Y., Tian, X., Mei, T.: 3d creation at your fingertips: From text or image to 3d assets. In: ACM MM (2023)
6. Chen, Y., Pan, Y., Li, Y., Yao, T., Mei, T.: Control3d: Towards controllable text-to-3d generation. In: ACM MM (2023)
7. Chen, Y., Pan, Y., Yang, H., Yao, T., Mei, T.: Vp3d: Unleashing 2d visual prompt for text-to-3d generation. In: CVPR (2024)
8. Chen, Y., Pan, Y., Yao, T., Tian, X., Mei, T.: Animating your life: Real-time video-to-animation translation. In: ACM MM (2019)
9. Chen, Y., Pan, Y., Yao, T., Tian, X., Mei, T.: Mocycle-gan: Unpaired video-to-video translation. In: ACM MM (2019)
10. Cheng, Y.C., Lee, H.Y., Tuyakov, S., Schwing, A., Gui, L.: SDFusion: Multimodal 3d shape completion, reconstruction, and generation. In: CVPR (2023)
11. Fuji Tsang, C., Shugrina, M., Lafleche, J.F., Takikawa, T., Wang, J., Loop, C., Chen, W., Jatavallabhula, K.M., Smith, E., Rozantsev, A., Perel, O., Shen, T., Gao, J., Fidler, S., State, G., Gorski, J., Xiang, T., Li, J., Li, M., Lebaredian, R.: Kaolin: A pytorch library for accelerating 3d deep learning research. `https://github.com/NVIDIAGameWorks/kaolin` (2022)
12. Gao, C., Jiang, B., Li, X., Zhang, Y., Yu, Q.: Genesistex: Adapting image denoising diffusion to texture space. In: CVPR (2024)
13. Gao, W., Aigerman, N., Thibault, G., Kim, V., Hanocka, R.: Textdeformer: Geometry manipulation using text guidance. In: SIGGRAPH (2023)
14. Hasselgren, J., Munkberg, J., Lehtinen, J., Aittala, M., Laine, S.: Appearance-driven automatic 3d model simplification. In: EGSR (2021)
15. He, Y., Bai, Y., Lin, M., Zhao, W., Hu, Y., Sheng, J., Yi, R., Li, J., Liu, Y.J.: T^3bench: Benchmarking current progress in text-to-3d generation. arXiv preprint arXiv:2310.02977 (2023)
16. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020)
17. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: NeurIPS Workshop (2022)
18. Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., Fleet, D.J.: Video diffusion models. arXiv preprint arXiv:2204.03458 (2022)
19. Huang, Q., Huang, X., Sun, B., Zhang, Z., Jiang, J., Bajaj, C.: Arapreg: An as-rigid-as possible regularization loss for learning deformable shape generators. In: ICCV (2021)
20. Jun, H., Nichol, A.: Shap-e: Generating conditional 3d implicit functions. arXiv preprint arXiv:2305.02463 (2023)
21. Katzir, O., Patashnik, O., Cohen-Or, D., Lischinski, D.: Noise-free score distillation. In: ICLR (2024)

22. Khalid, N.M., Xie, T., Belilovsky, E., Tiberiu, P.: Clip-mesh: Generating textured meshes from text using pretrained image-text models. In: SIGGRAPH (2022)
23. Laine, S., Hellsten, J., Karras, T., Seol, Y., Lehtinen, J., Aila, T.: Modular primitives for high-performance differentiable rendering. ACM Trans. Graph. (2020)
24. Li, M., Duan, Y., Zhou, J., Lu, J.: Diffusion-sdf: Text-to-shape via voxelized diffusion. In: CVPR (2023)
25. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3d: High-resolution text-to-3d content creation. In: CVPR (2023)
26. Metzer, G., Richardson, E., Patashnik, O., Giryes, R., Cohen-Or, D.: Latent-nerf for shape-guided generation of 3d shapes and textures. In: CVPR (2023)
27. Michel, O., Bar-On, R., Liu, R., Benaim, S., Hanocka, R.: Text2mesh: Text-driven neural stylization for meshes. In: CVPR (2022)
28. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
29. Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., Chen, M.: Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741 (2021)
30. Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-e: A system for generating 3d point clouds from complex prompts. arXiv preprint arXiv:2212.08751 (2022)
31. Pan, Y., Qiu, Z., Yao, T., Li, H., Mei, T.: To create what you tell: Generating videos from captions. In: ACM Multimedia (2017)
32. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. In: ICLR (2024)
33. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. In: ICLR (2023)
34. Qian, Y., Cai, Q., Pan, Y., Li, Y., Yao, T., Sun, Q., Mei, T.: Boosting diffusion models with moving average sampling in frequency domain. In: CVPR (2024)
35. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
36. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125 (2022)
37. Richardson, E., Metzer, G., Alaluf, Y., Giryes, R., Cohen-Or, D.: Texture: Text-guided texturing of 3d shapes. In: SIGGRAPH (2023)
38. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
39. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S.K.S., Ayan, B.K., Mahdavi, S.S., Lopes, R.G., Salimans, T., Ho, J., Fleet, D.J., Norouzi, M.: Photorealistic text-to-image diffusion models with deep language understanding. In: NeurIPS (2022)
40. Shen, T., Gao, J., Yin, K., Liu, M.Y., Fidler, S.: Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In: NeurIPS (2021)
41. Shi, Y., Wang, P., Ye, J., Long, M., Li, K., Yang, X.: Mvdream: Multi-view diffusion for 3d generation. In: ICLR (2024)
42. Sorkine, O., Alexa, M.: As-rigid-as-possible surface modeling. In: SGP. Citeseer (2007)

43. Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., Seidel, H.P.: Laplacian surface editing. In: SGP (2004)
44. Sun, C., Sun, M., Chen, H.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: CVPR (2022)
45. Tang, J., Markhasin, L., Wang, B., Thies, J., Nießner, M.: Neural shape deformation priors. In: NeurIPS (2022)
46. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In: ICLR (2024)
47. Wang, H., Du, X., Li, J., Yeh, R.A., Shakhnarovich, G.: Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In: CVPR (2023)
48. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In: NeurIPS (2023)
49. Yang, H., Chen, Y., Pan, Y., Yao, T., Chen, Z., Mei, T.: 3dstyle-diffusion: Pursuing fine-grained text-driven 3d stylization with 2d diffusion models. In: ACM MM (2023)
50. Yang, R., Srivastava, P., Mandt, S.: Diffusion probabilistic modeling for video generation. arXiv preprint arXiv:2203.09481 (2022)
51. Young., J.: Xatlas: Mesh parameterization / uv unwrapping library. `https://github.com/jpcy/xatlas` (2022)
52. Yu, X., Guo, Y.C., Li, Y., Liang, D., Zhang, S.H., Qi, X.: Text-to-3d with classifier score distillation. In: ICLR (2024)
53. Zhang, Z., Long, F., Pan, Y., Qiu, Z., Yao, T., Cao, Y., Mei, T.: Trip: Temporal residual learning with image noise prior for image-to-video diffusion models. In: CVPR (2024)
54. Zhu, J., Zhuang, P., Koyejo, S.: Hifa: High-fidelity text-to-3d generation with advanced diffusion guidance. In: ICLR (2024)
55. Zhu, R., Pan, Y., Li, Y., Yao, T., Sun, Z., Mei, T., Chen, C.W.: Sd-dit: Unleashing the power of self-supervised discrimination in diffusion transformer. In: CVPR (2024)