threestudio: a modular framework for diffusion-guided 3D generation

Ying-Tian Liu[1†] Yuan-Chen Guo[1†] Vikram Voleti[2] Ruizhi Shao[1] Chia-Hao Chen[1] Guan Luo[1]
Zixin Zou[1†] Chen Wang[1] Christian Laforte[2] Yan-Pei Cao[3,4] Song-Hai Zhang[1]
[1]Tsinghua University [2]Stability AI [3] Tencent AI Lab [4]ARC Lab, Tencent PCG

Abstract

We introduce threestudio, an open-source, unified, and modular framework specifically designed for 3D content generation. This framework extends diffusion-based 2D image generation models to 3D generation guidance while incorporating conditions such as text and images. We delineate the modular architecture and design of each component within threestudio. Moreover, we re-implement state-of-the-art methods for 3D generation within threestudio, presenting comprehensive comparisons of their design choices. This versatile framework has the potential to empower researchers and developers to delve into cutting-edge techniques for 3D generation, and presents the capability to facilitate further applications beyond 3D generation. Code is available at https://github.com/threestudio-project/threestudio.

1. Introduction

Generative modeling has revolutionized the 1D text/audio and 2D image/video domains. However, 3D content generation still poses challenges due to the scarcity of 3D datasets, and the computational complexity in 3D space. To reduce the reliance on 3D models, recent methods leverage the capabilities of image diffusion models to distill 3D structures from 2D image generative spaces. The pioneering diffusion-guided 3D generation method, DreamFusion [16], introduced Score Distillation Sampling (SDS) to enable using a text-to-image model as guidance to generate a 3D Neural Radiance Field (NeRF) [14] through iterative optimization. Subsequent research has extensively investigated the diffusion-guided 3D generation framework, encompassing various scene representations, diffusion guidance, and training strategies. These explorations have aimed to enhance different aspects of the framework, such as geometric details [4], high fidelity and diversity [30], high resolution [18], mesh generation [27], faster generation [13], image-to-3D generation [12, 21, 26, 11], etc.

However, integrating all these enhancements or using specific components from these methods for a particular 3D generation task is a complex endeavor. Many of these methods are independent and built upon diverse codebases. Additionally, evaluating these methods with different parameter settings and initialization is challenging.

In this paper, we present *threestudio*, a modular framework in open-sourced codebase, designed for 3D content creation through the extension of diffusion-based 2D image generation models. By harnessing the capabilities of diffusion models and integrating them with innovative 3D representations, *threestudio* offers a versatile and modular platform that empowers researchers and developers to explore cutting-edge techniques for 3D content creation. Throughout the paper, we present an account of how various methods for text-to-3D and image-to-3D synthesis are abstracted within *threestudio*, discuss the core components of this framework, and provide best practices for achieving high-quality results.

2. Framework Design

threestudio is designed following several principles:

- **Modular**: users can easily combine different components to form a pipeline;
- **Extensible**: users are free to customize their own components and pipelines;
- **Flexible**: users can chain different pipelines together for better performance;
- **Configurable**: users can easily build custom pipelines and specify all the hyper-parameters in a single configuration file without changing the code.

Based on the observation that existing pipelines mainly follow a similar workflow, we formulate the pipeline with the combination of several independent components, namely data synthesizer, geometry, renderer, material, background, diffusion guidance, and prompt processor. By defining unified interfaces for these components, we can

[†] Work is partially done during the internship at Tencent.

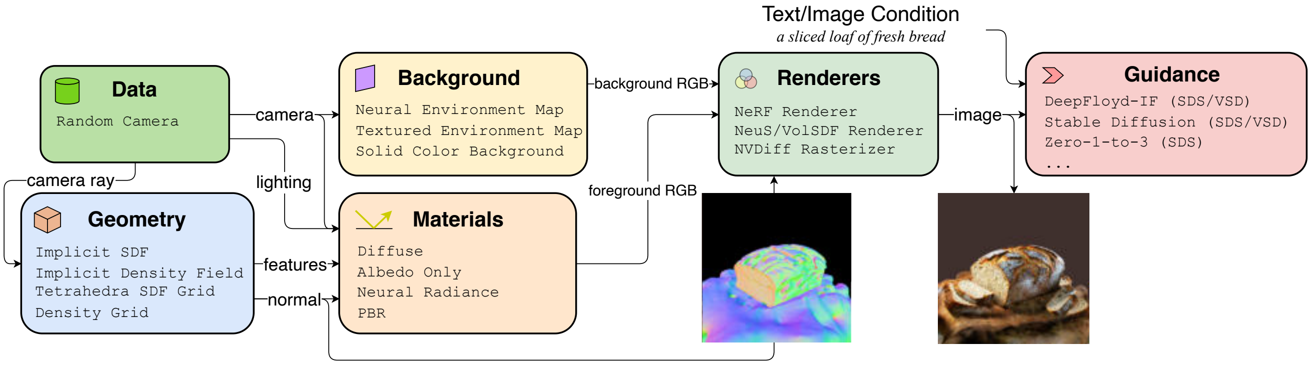


Figure 1: The abstract pipeline for text/image conditioned 3D generation in *threestudio*.

implement these pipelines in a unified framework with all the benefits mentioned above. The detailed functionality of these components are described below.

Data Synthesizer. The role of the data synthesizer is to produce camera extrinsic and intrinsic parameters used in optimization, as well as lighting conditions if necessary. Typically, the camera parameters are randomly sampled given the range of elevation angles, azimuth angles, distances to the scene center, and field of view. The lighting conditions are sampled according to a pre-defined strategy [16, 10]. The users can easily tweak these parameters to focus on a certain level of detail or region of the scene during optimization.

Geometry. The geometry component defines how the 3D object/scene is represented. There are several choices for the geometry representation, such as implicit density field, implicit signed distance field (SDF), density grid, SDF grid [24], and triangular mesh. Each representation offers distinct advantages, and no single representation is universally suitable for all scenarios. For instance, implicit density fields are easy to optimize but the resolution of the rendered images is limited, while triangular meshes can be efficiently rendered at higher resolutions but are prone to local minima in optimization. To fully utilize the advantages of different geometry representations, we implement the conversion between some of them, as shown in Fig. 2. Typically, we could start from an implicit density field or implicit SDF representation to get a coarse shape, and convert the geometry to an SDF grid to optimize the scene using high-resolution mesh renderings. This coarse-to-fine strategy is adopted in many existing pipelines [10, 27, 30].

Renderer. The differentiable renderer component serves the purpose of generating rendered images (e.g., RGB color, opacity, depth, or normal) of the scene, as well as back-propagating gradients from the diffusion guidance to scene parameters. We integrated various differentiable rendering methods suitable to the different geometry representations. For implicit density fields, we use the NeRF [14] renderer. For implicit SDF, we offer the NeuS [29] and VolSDF [31] renderer. As for the tetrahedral SDF grid representation and triangular meshes, we employ a differentiable rasterizer [9] for rendering.

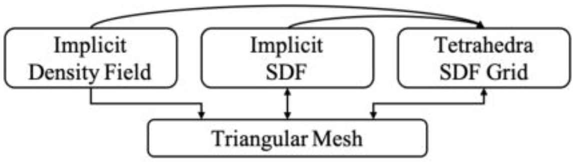
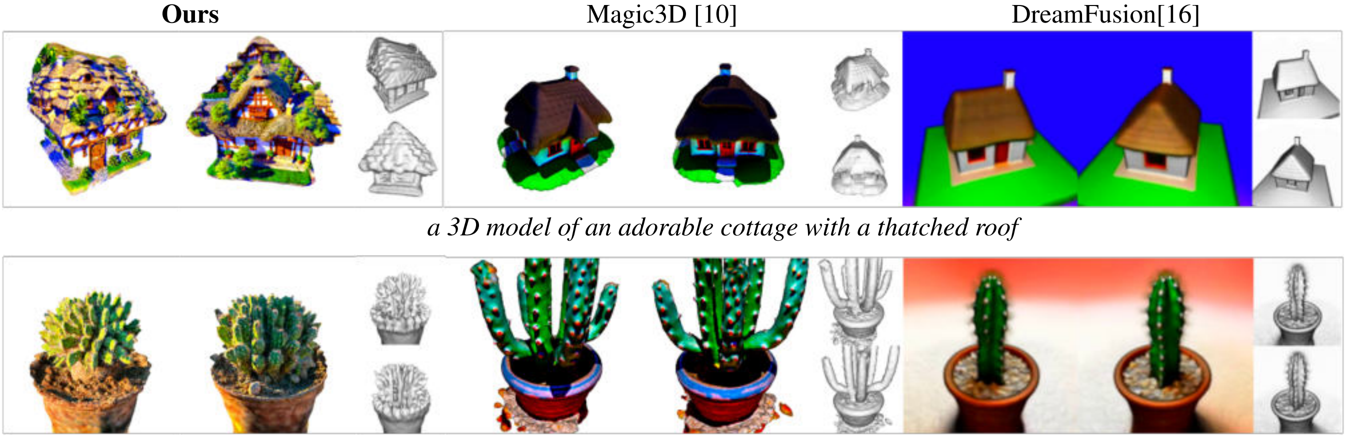


Figure 2: Conversion between geometry representations.

Material and Background. The material component plays a crucial role in determining the appearance of the 3D object under specific camera and lighting conditions. In *threestudio*, we offer a variety of commonly used materials, including albedo-only, diffuse, neural radiance [14], and physically-based rendering (PBR) materials. The background is modeled as an environment map, providing colors based on the ray directions. It can be parameterized by a small neural network, an explicit texture map [28], or simply a single color.

Diffusion Guidance. The diffusion guidance component is the core of the generation process. It leverages a pre-trained diffusion-based 2D image generation model and guides the update of scene parameters throughout the optimization process. Typical algorithms for this guidance are Score Distillation Sampling (SDS) [16], Score Jacobian Chaining (SJC) [28], and Variational Score Distillation (VSD) [30]. We have implemented all these algorithms, and support various open-source pre-trained models, such as Stable Diffusion [19], ControlNet [32], InstructPix2Pix [2], DeepFloyd-IF, and Zero-1-to-3 [11].



a 3D model of an adorable cottage with a thatched roof

a small saguaro cactus planted in a clay pot

Figure 3: Results obtained by our TextMesh-ProlificDreamer pipeline, Magic3D [10] and DreamFusion [16].

Prompt Processor. For text conditioning, we utilize a prompt processor component paired with the diffusion guidance to obtain feature embeddings for the input prompt. We implement view-dependent prompting [16], which appends a modifier to the prompt describing the direction (“front vie”, “side view”, “back view” or “overhead view”) according to the sampled camera. Additionally, we integrate techniques like PerpNeg [1] and Prompt Debiasing [8] to address the “Janus problem” [15].

3. Text to 3D

The community has recently witnessed a surge of text-to-3D generation efforts, starting with DreamFusion [16] and SJC [28], and progressing towards higher quality [10, 30], faster generation [13], and more compact geometry representation [27, 4]. Using our modular framework, all these methods can be easily implemented by combining different components.

Tab. 1 compares some of the components and design choices made by these methods. Detailed optimization strategies like regularization terms and noise level annealing are not covered in the table. Our framework allows for exploring advanced 3D generation solutions by combining different components and chaining different pipelines. For instance, leveraging the observation that implicit SDF representation can yield compact meshes and VSD [30] can add realistic texture details to the geometry, we chain the geometry stage of TextMesh [27] with the geometry and texture stages from ProlificDreamer [30], yielding promising results compared to existing methods (see Fig. 3).

4. Image to 3D

Tab. 2 compares image-to-3D methods and some of the components and design choices they make. In Fig. 4, we compare the 3D results from our Zero-1-to-3 implementation in *threestudio* with other prior methods.

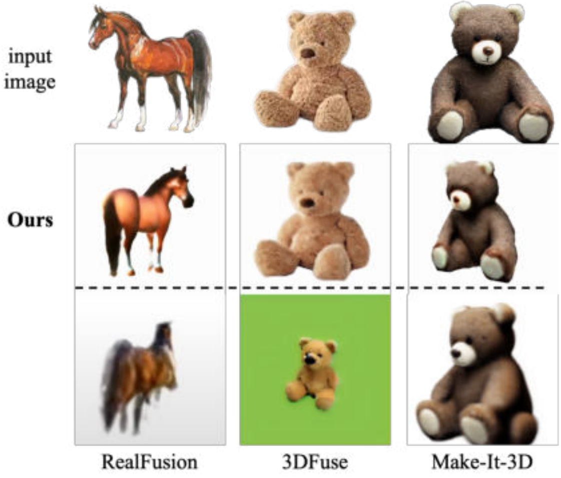


Figure 4: Image-conditioned generation results using our Zero-1-to-3 pipeline, RealFusion [12], 3DFuse [21], and Make-It-3D [26].

For the task of generating a 3D object from an image, we primarily rely on the recent Zero-1-to-3 model [11]. It was trained by fine-tuning Stable Diffusion [20] on images from Objaverse [5], a dataset of 3D objects. Zero-1-to-3 is conditioned on an image i.e., the source view of the object, and a relative pose between the source view and a target view. It then generates the target view of the same object.

This Zero-1-to-3 model can then be used as the diffusion guide to generate a 3D object from an image using SDS or SJC [11]. Through extensive experimentation, we found a new Zero1-to-3 SDS-based configuration that outperforms SJC’s visual quality and is fast: the entire optimization process takes 5 minutes on one A100 40GB. We also list some insights we found that help the generation:

- Elevation: Fig. 5 shows that specifying a suitable elevation of the conditioning image is critical to obtain 3D objects with good quality.

Table 1: State-of-the-art text-to-3D methods in *threestudio*. * Code not released, reproduced by us.

Method	Stage	Geometry	Renderer	Material	Diffusion Guide
DreamFusion* [16]	-	Implicit Density Field	NeRF Renderer	Diffuse	DeepFloyd-IF (SDS)
LatentNeRF [13]	-	Implicit Density Field	NeRF Renderer	Albedo Only	Stable Diffusion (SDS)
SJC [28]	-	Density Grid	NeRF Renderer	Albedo Only	Stable Diffusion (SJC)
Magic3D* [10]	Coarse	Implicit Density Field	NeRF Renderer	Diffuse	DeepFloyd-IF (SDS)
	Refine	Tetrahedra SDF Grid	NVDiff Rasterizer	Diffuse	Stable Diffusion (SDS)
Fantasia3D [4]	Geometry	Implicit SDF	NVDiff Rasterizer	-	Stable Diffusion (SDS)
	Texture	Tetrahedra SDF Grid	NVDiff Rasterizer	PBR	Stable Diffusion (SDS)
ProlificDreamer* [30]	Coarse	Implicit Density Field	NeRF Renderer	Albedo Only	Stable Diffusion (VSD)
	Geometry	Tetrahedra SDF Grid	NVDiff Rasterizer	-	Stable Diffusion (SDS)
	Texture	Tetrahedra SDF Grid	NVDiff Rasterizer	Albedo Only	Stable Diffusion (VSD)
TextMesh* [27]	Geometry	Implicit SDF	NeuS Renderer	Diffuse	DeepFloyd-IF (SDS)

Table 2: State-of-the-art image-to-3D methods in *threestudio*.

Method	Stage	Geometry	Renderer	Material	Diffusion Guide
RealFusion [12]	-	NeRF Renderer	Implicit Density Field	Albedo Only	Stable Diffusion (SDS)
Zero-1-to-3 [11]	-	NeRF Renderer	Implicit Density Field	Albedo Only	Zero-1-to-3 (SDS)

- Noise level: Fig. 6 shows that higher noise levels at the start of training are crucial to obtain faster results. Hence, we initially set the minimum noise level to 0.7, then anneal it to 0.2 during training, while maintaining the maximum noise level at 0.98.

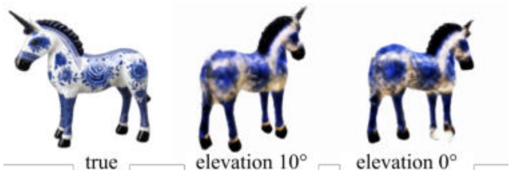


Figure 5: Left: image condition. Middle and Right: 3D generations from the left image, assuming elevations of 10° and 0°, respectively. 10° assumption is qualitatively better.

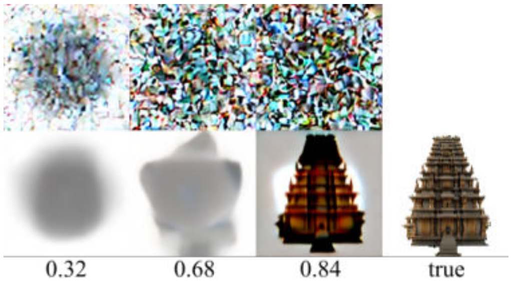


Figure 6: Effect of changing noise level (shown below). The top row is the noisy image of the initial 3D render, bottom row is 1-step predicted clean image.

5. Applications Beyond 3D Object Generation

The modular nature of *threestudio* can also facilitate potential applications beyond 3D object generation. By applying special geometry representations and diffusion guidance, *threestudio* can support advanced tasks, including:

3D Editing. *threestudio* supports several diffusion-based 3D editing methods, such as InstructNeRF2NeRF [7] and Control4D-Static [22], using InstructPix2Pix [2] and ControlNet [32] to guide the editing process respectively.

Text to 4D. *threestudio* can be extended to support text-to-4D generation [25] by implementing temporal-aware geometry representations like D-NeRF [17], HexPlane [3], K-Planes [6], and Tensor4D [23], and applying a diffusion-based video generation model as guidance.

6. Conclusion

We present *threestudio*, a modular framework for diffusion-guided 3D generation. The extensible, flexible, and configurable design enables researchers to readily explore innovative techniques in this burgeoning field. We hope that *threestudio* will inspire novel directions in algorithm design and ultimately expand the horizons of what is possible with diffusion-based 3D generation. Through continued innovation built upon *threestudio*, we envision substantial progress toward fully realizing the promise of controllable, high-fidelity 3D synthesis.

References

[1] Mohammadreza Armandpour, Ali Sadeghian, Huangjie Zheng, Amir Sadeghian, and Mingyuan Zhou. Re-imagine the negative prompt algorithm: Transform 2d diffusion into 3d, alleviate janus problem and beyond. *arXiv preprint arXiv:2304.04968*, 2023.

[2] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

[3] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023.

[4] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. *CoRR*, abs/2303.13873, 2023.

[5] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.

[6] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023.

[7] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. *arXiv preprint arXiv:2303.12789*, 2023.

[8] Susung Hong, Donghoon Ahn, and Seungryong Kim. Debiasing scores and prompts of 2d diffusion for robust text-to-3d generation. *CoRR*, abs/2303.15413, 2023.

[9] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Trans. Graph.*, 39(6):194:1–194:14, 2020.

[10] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. *CoRR*, abs/2211.10440, 2022.

[11] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *CoRR*, abs/2303.11328, 2023.

[12] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Realfusion: 360° reconstruction of any object from a single image. *CoRR*, abs/2302.10663, 2023.

[13] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. *CoRR*, abs/2211.07600, 2022.

[14] 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 (1)*, volume 12346 of *Lecture Notes in Computer Science*, pages 405–421. Springer, 2020.

[15] Ben Poole. Janus problem. `https://twitter.com/poolio/status/1578045212236034048?lang=en`, 2022. Online; accessed 24 July 2023.

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

[17] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.

[18] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, and Bernard Ghanem. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *CoRR*, abs/2306.17843, 2023.

[19] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE, 2022.

[20] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10674–10685. IEEE, 2022.

[21] Junyoung Seo, Wooseok Jang, Minseop Kwak, Jaehoon Ko, Hyeonsu Kim, Junho Kim, Jin-Hwa Kim, Jiyoung Lee, and Seungryong Kim. Let 2d diffusion model know 3d-consistency for robust text-to-3d generation. *CoRR*, abs/2303.07937, 2023.

[22] Ruizhi Shao, Jingxiang Sun, Cheng Peng, Zerong Zheng, Boyao Zhou, Hongwen Zhang, and Yebin Liu. Control4d: Dynamic portrait editing by learning 4d gan from 2d diffusion-based editor. *arXiv preprint arXiv:2305.20082*, 2023.

[23] Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16632–16642, 2023.

[24] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In *NeurIPS*, 2021.

[25] Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, et al. Text-to-4d dynamic scene generation. *arXiv preprint arXiv:2301.11280*, 2023.

[26] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d

creation from A single image with diffusion prior. *CoRR*, abs/2303.14184, 2023.

[27] Christina Tsalicoglou, Fabian Manhardt, Alessio Tonioni, Michael Niemeyer, and Federico Tombari. Textmesh: Generation of realistic 3d meshes from text prompts. *CoRR*, abs/2304.12439, 2023.

[28] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. *CoRR*, abs/2212.00774, 2022.

[29] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *NeurIPS*, pages 27171–27183, 2021.

[30] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *CoRR*, abs/2305.16213, 2023.

[31] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *NeurIPS*, pages 4805–4815, 2021.

[32] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.