TripoSR: Fast 3D Object Reconstruction from a Single Image

Dmitry Tochilkin[1] David Pankratz[1] Zexiang Liu[2] Zixuan Huang[1] Adam Letts[1]
Yangguang Li[2] Ding Liang[2] Christian Laforte[1] Varun Jampani[1*] Yan-Pei Cao[2*]

[1]Stability AI, [2]Tripo AI

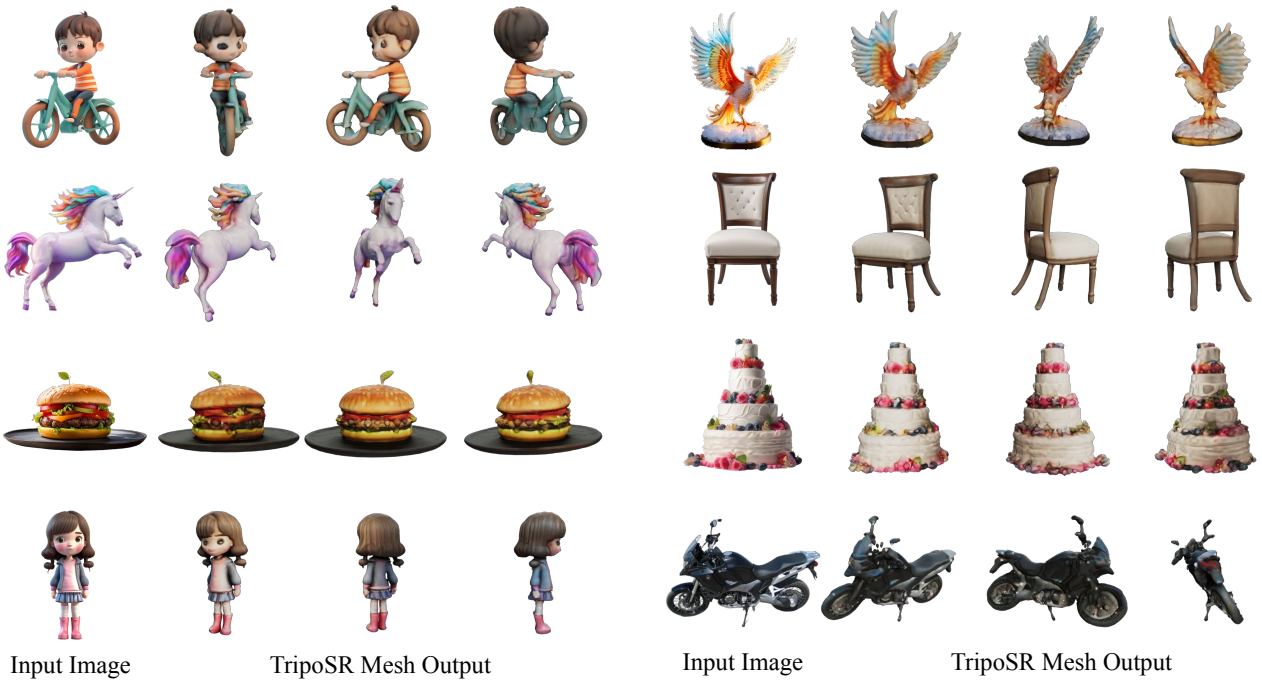

Figure 1. We present TripoSR, a 3D reconstruction model that reconstructs high-quality 3D from single images in under 0.5 seconds. Our model achieves state-of-the-art performance and generalizes to objects of various types and input images across different domains.

Abstract

This technical report introduces TripoSR, 3D reconstruction model leveraging transformer architecture for fast feed-forward 3D generation, producing 3D mesh from a single image in under 0.5 seconds. Building upon the LRM [11] network architecture, TripoSR integrates substantial improvements in data processing, model design, and training techniques. Evaluations on public datasets show that TripoSR exhibits superior performance, both quantitatively and qualitatively, compared to other open-source alternatives. Released under the MIT license, TripoSR is intended to empower researchers, developers, and creatives with the latest advancements in 3D generative AI.

[*]Equal advising.

Model: https://huggingface.co/stabilityai/TripoSR
Code: https://github.com/VAST-AI-Research/TripoSR
Demo: https://huggingface.co/spaces/stabilityai/TripoSR

1. Introduction

The landscape of 3D Generative AI has witnessed a confluence of developments in recent years, blurring the lines between 3D reconstruction from single or few views and 3D generation [3, 9, 11, 13, 17, 28, 32–34]. This convergence has been significantly accelerated by the introduction of large-scale public 3D datasets [4, 5] and advances in generative model architectures. Comprehensive reviews

of these technologies can be found in the literature such as [15] and [21].

To overcome the scarcity of 3D training data, recent efforts have explored utilizing 2D diffusion models to create 3D assets from text prompts [19, 20, 26] or input images [17, 22]. DreamFusion [19], a notable example, introduced score distillation sampling (SDS), employing a 2D diffusion model to guide the optimization of 3D models. This approach represents a pivotal strategy in leveraging 2D priors for 3D generation, achieving breakthroughs in generating detailed 3D objects. However, these methods typically face limitations with slow generation speed, due to the extensive optimization and computational demands, and the challenge of precisely controlling the output models.

On the contrary, feed-forward 3D reconstruction models achieve significantly higher computational efficiency [7, 8, 11–14, 17, 18, 23–25, 27, 30, 31, 34]. Several recent approaches [11, 13, 14, 17, 23, 25, 27, 30, 34] along this direction have shown promise in scalable training on diverse 3D datasets. These approaches facilitate rapid 3D model generation through fast feed-forward inference and are potentially more capable of providing precise control over the generated outputs, marking a notable shift in the efficiency and applicability of these models.

In this work, we introduce TripoSR model for fast feed-forward 3D generation from a single image that takes less than 0.5 seconds on an A100 GPU. Building upon the LRM [11] architecture, we introduce several improvements in terms of data curation and rendering, model design and training techniques. Experimental results demonstrate superior performance, both quantitatively and qualitatively, compared to other open-source alternatives. Figure 1 shows some sample results of the TripoSR. TripoSR is made available under the MIT license, accompanied by source code, the pretrained model, and an interactive online demo. The release aims to enable researchers, developers, and creatives to advance their work with the latest advancements in 3D generative AI, promoting progress within the wider domains of AI, computer vision, and computer graphics. Next, we introduce the technical advances in our TripoSR model, followed by the quantitative and qualitative results on two public datasets.

2. TripoSR: Data and Model Improvements

The design of TripoSR is based on the LRM [11], with a series of technical advancements in data curation, model and training strategy. We now give an overview of the model followed by our technical improvements.

2.1. Model Overview

Similar to LRM [11], TripoSR leverages the transformer architecture and is specifically designed for single-image 3D reconstruction. It takes a single RGB image as input and outputs a 3D representation of the object in the image. The core of TripoSR includes components: an image encoder, an image-to-triplane decoder, and a triplane-based neural radiance field (NeRF).

The image encoder is initialized with a pre-trained vision transformer model, DINOv1 [1], which projects an RGB image into a set of latent vectors. These vectors encode the global and local features of the image and include the necessary information to reconstruct the 3D object.

The subsequent image-to-triplane decoder transforms the latent vectors onto the triplane-NeRF representation [2]. The triplane-NeRF representation is a compact and expressive 3D representation, well-suited for representing objects with complex shapes and textures. Our decoder consists of a stack of transformer layers, each with a self-attention layer and a cross-attention layer. The self-attention layer allows the decoder to attend to different parts of the triplane representation and learn relationships between them. The cross-attention layer allows the decoder to attend to the latent vectors from the image encoder and incorporate global and local image features into the triplane representation. Finally, the NeRF model consists of a stack of multilayer perceptrons (MLPs), which are responsible for predicting the color and density of a 3D point in space.

Instead of conditioning the image-to-triplane projection on camera parameters, we have opted to allow the model to "guess" the camera parameters (both extrinsics and intrinsics) during training and inference. This is to enhance the model's robustness to in-the-wild input images at inference time. By foregoing explicit camera parameter conditioning, our approach aims to cultivate a more adaptable and resilient model capable of handling a wide range of real-world scenarios without the need for precise camera information.

The architecture's main parameters, such as the number of layers in the transformer, the dimensions of the triplanes, the specifics of the NeRF model, and the main training configurations, are detailed in Table 1. Compared to LRM [11], TripoSR introduces several technical improvements which we discuss next.

2.2. Data Improvements

Recognizing the critical importance of data, we have incorporated two improvements in our training data collection:

- **Data Curation**: By selecting a carefully curated subset of the Objaverse [4] dataset, which is available under the CC-BY license, we have enhanced the quality of training data.
- **Data Rendering**: We have adopted a diverse array of data rendering techniques that more closely emulate the distribution of real-world images, thereby enhancing the model's ability to generalize, even when trained exclusively with the Objaverse dataset.

Parameter		Value
Image Tokenizer	image resolution	512×512
	patch size	16
	# attention layers	12
	# feature channels	768
Triplane Tokenizer	# tokens	$32 \times 32 \times 3$
	# channels	16
Backbone	# channels	1024
	attention layers	16
	# attention heads	16
	attention head dim	64
	cross attention dim	768
Triplane Upsampler	factor	2
	# input channels	1024
	# output channels	40
	output shape	$64 \times 64 \times 40$
NeRF MLP	width	64
	# layers	10
	activation	SiLU
Renderer	# samples per ray	128
	radius	0.87
	density activation	exp
	density bias	-1.0
Training	learning rate	$4e{-4}$
	optimizer	AdamW
	lr scheduler	CosineAnnealingLR
	# warm-up steps	$2,000$
	λ_{LPIPS}	2.0
	λ_{mask}	0.05

Table 1. Model configuration of TripoSR.

2.3. Model and Training Improvements

Our adjustments aim to boost both the model's efficiency and its performance.

Triplane Channel Optimization. The configuration of channels within the triplane-NeRF representation plays an important role in managing the GPU memory footprint during both training and inference, due to the high computational cost of volume rendering. Moreover, the channel count significantly influences the model's capacity for detailed and high-fidelity reconstruction. In pursuit of an optimal balance between reconstruction quality and computational efficiency, experimental evaluations led us to adopt a configuration of 40 channels. This choice enables the use of larger batch sizes and higher resolutions during the training phase, while concurrently minimizing the memory requirements during inference.

Mask Loss. We incorporated a mask loss function during training that significantly reduces "floater" artifacts and improves the fidelity of reconstructions:

$$\mathcal{L}_{\text{mask}}(\hat{\boldsymbol{M}}_v, \boldsymbol{M}_v^{GT}) = \text{BCE}(\hat{\boldsymbol{M}}_v, \boldsymbol{M}_v^{GT}), \tag{1}$$

where $\hat{\boldsymbol{M}}_v$ and $\boldsymbol{M}_v^{GT}$ are rendered and ground-truth mask images of the v-th supervision view, respectively. The full training loss we minimized during training is:

$$\begin{aligned}\mathcal{L}_{\text{recon}}(\boldsymbol{I}) = \frac{1}{V}\sum_{v=1}^{V}\Big(&\mathcal{L}_{\text{MSE}}(\hat{\boldsymbol{I}}_v, \boldsymbol{I}_v^{GT}) \\ &+ \lambda_{\text{LPIPS}}\mathcal{L}_{\text{LPIPS}}(\hat{\boldsymbol{I}}_v, \boldsymbol{I}_v^{GT}) \\ &+\lambda_{\text{mask}}\mathcal{L}_{\text{mask}}(\hat{\boldsymbol{M}}_v, \boldsymbol{M}_v^{GT})\Big)\end{aligned} \tag{2}$$

Local Rendering Supervision. Our model fully relies on rendering losses for supervision, thereby imposing a need for high-resolution rendering for our model to learn detailed shape and texture reconstructions. However, rendering and supervising at high resolutions (e.g., 512×512 or higher)

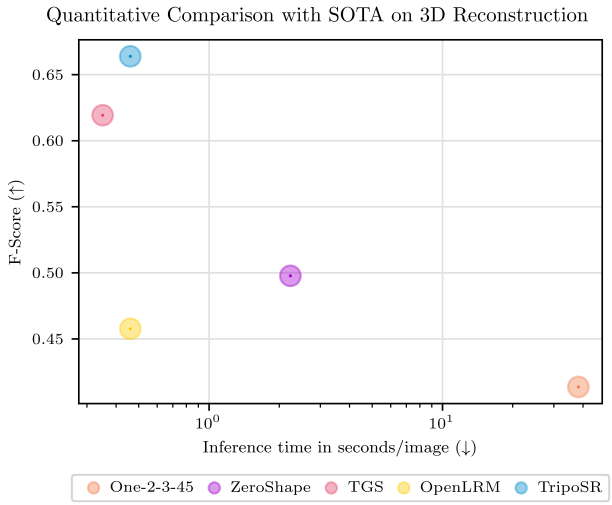

Figure 2. We outperform SOTA methods for 3D reconstruction while achieving fast inference time. In the figure, F-Score with threshold 0.1 is averaged over GSO [6] and OmniObject3D [29].

Method	CD↓	FS@0.1↑	FS@0.2↑	FS@0.5↑
One-2-3-45 [16]	0.227	0.382	0.630	0.878
ZeroShape [13]	0.160	0.489	0.757	0.952
TGS [34]	0.122	0.637	0.846	0.968
OpenLRM [10]	0.180	0.430	0.698	0.938
TripoSR (ours)	**0.111**	**0.651**	**0.871**	**0.980**

Table 2. Quantitative comparison of different techniques on GSO [6] validation set, where CD and FS refer to Chamfer Distance and F-score respectively.

Method	CD↓	FS@0.1↑	FS@0.2↑	FS@0.5↑
One-2-3-45 [16]	0.197	0.445	0.698	0.907
ZeroShape [13]	0.144	0.507	0.786	0.968
TGS [34]	0.142	0.602	0.818	0.949
OpenLRM [10]	0.155	0.486	0.759	0.959
TripoSR (ours)	**0.102**	**0.677**	**0.890**	**0.986**

Table 3. Quantitative comparison of different techniques on OmniObject3D [29] validation set, where CD and FS refers to Chamfer Distance and F-score respectively.

can overwhelm computational and GPU memory loads. To circumvent this issue, we render 128×128-sized random patches from the original 512×512 resolution images during training. Crucially, we increase the likelihood of selecting crops that cover foreground regions, thereby placing greater emphasis on the areas of interest. This importance sampling strategy ensures faithful reconstructions of object surface details, effectively balancing computational efficiency and reconstruction granularity.

3. Results

We quantitatively and qualitatively compare TripoSR to previous state-of-the-art methods using two different datasets with 3D reconstruction metrics.

Evaluation Datasets. We curate two public datasets, GSO [6] and OmniObject3D [29], for evaluations. We identify that both datasets include many simple-shaped objects (e.g., box, sphere or cylinder) and can thus cause high validation bias towards these simple shapes. Therefore we manually filter the datasets and select around 300 objects from each dataset to make sure they form a diverse and representative collection of common objects.

3D Shape Metrics. We extract the isosurface using Marching Cubes [**?**] to convert implicit 3D representations (such as NeRF) into meshes. We sample 10K points from these surfaces to calculate the Chamfer Distance (CD) and F-score (FS). Considering that some methods are not capable of predicting view-centric shapes, we use a brute-force search approach to align the predictions with the ground truth shapes. We linearly search the rotation angle by optimizing for the lowest CD and further employ the Iterative Closest Point (ICP) method to refine the alignment.

Quantitative Comparisons. We compare TripoSR with the existing state-of-the-art baselines on 3D reconstruction that use feed-forward techniques, including One-2-3-45 [16], TriplaneGaussian (TGS) [34], ZeroShape [13] and OpenLRM [10][1]. As shown in Table 2 and Table 3, our TripoSR significantly outperforms all the baselines, both in terms of CD and FS metrics, achieving the new state-of-the-art performance on this task.

Performance vs. Runtime. Another key advantage of TripoSR is its inference speed. It takes around 0.5 seconds to produce a 3D mesh from a single image on an NVIDIA A100 GPU. Figure 2 shows a 2D plot of different techniques with inference times along the x-axis and the averaged F-Score along the y-axis. The plot shows that TripoSR is among the fastest networks, while also being the best-performing feed-forward 3D reconstruction model.

Visual Results. We further show the qualitative results of different approaches in Figure 3. Because some methods do not reconstruct textured meshes, we render TripoSR reconstructions both with and without vertex color for a better comparison. As shown in the figure, ZeroShape tends to predict over-smoothed shapes. TGS reconstructs more surface details but these details sometimes do not align with the input. Moreover, both ZeroShape and TGS cannot output textured meshes directly [2]. On the other hand, One-2-3-45 and OpenLRM predict textured meshes, but their esti-

[1] We use the openlrm-large-obj-1.0 model.

[2] TGS leverages 3DGS to represent 3D objects. We follow the paper and utilize their auxiliary point cloud outputs to reconstruct the surface. However, it is non-trivial to reconstruct textures on meshes, (e.g., directly taking vertex colors from the nearest Gaussian leads to noisy textures).

Figure 3. **Qualitative results.** We compare TripoSR output meshes to other SOTA methods on GSO and OmniObject3D (first four columns are from GSO [6], last two are from OmniObject3D [29]). Our reconstructed 3D shapes and textures achieve significantly higher quality and better details than previous state-of-the-art methods.

mated shapes are often inaccurate. Compared to these baselines, TripoSR demonstrates a high reconstruction quality for both shape and texture. Our model not only captures a better overall 3D structure of the object, but also excels at modeling several intricate details.

4. Conclusion

In this report, we present an open-source feedforward 3D reconstruction model, TripoSR. The core of our model is a transformer-based architecture developed upon the LRM network [11], together with substantial technical improvements along multiple axes. Evaluated on two public benchmarks, our model demonstrates state-of-the-art reconstruction performance with high computational efficiency. We hope TripoSR empowers researchers and developers in developing more advanced 3D generative AI models.

Acknowledgements

Tripo AI. We extend our sincere gratitude to Yuan-Chen Guo and Zi-Xin Zou for their critical roles in coding, demo development, and experimentation. We also thank Peng Wang for his insightful discussions, Dehu Wang for preparing the datasets for model validation, and Sienna Huang for managing communication within our collaboration.

Stability AI. We thank Emad Mostaque, Anel Islamovic, Bryce Wilson, Ana Guillen, Adam Chen, Christian Dowell and Ella Irwin for their help in various aspects of the model development, collaboration and the release. We also thank Vikram Voleti for helpful discussions and Eric Courtemanche for his help with visual results.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2

[2] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16123–16133, 2022. 2

[3] Eric R. Chan, Koki Nagano, Matthew A. Chan, Alexander W. Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. GeNVS: Generative novel view synthesis with 3D-aware diffusion models. In *arXiv*, 2023. 1

[4] 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. 1, 2

[5] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[6] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 4, 5

[7] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9785–9795, 2019. 2

[8] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018. 2

[9] Yuan-Chen Guo, Ying-Tian Liu, Chen Wang, Zi-Xin Zou, Guan Luo, Chia-Hao Chen, Yan-Pei Cao, and Song-Hai Zhang. threestudio: A unified framework for 3d content generation, 2023. 1

[10] Zexin He and Tengfei Wang. Openlrm: Open-source large reconstruction models. https://github.com/3DTopia/OpenLRM, 2023. 4

[11] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023. 1, 2, 5

[12] Zixuan Huang, Varun Jampani, Anh Thai, Yuanzhen Li, Stefan Stojanov, and James M. Rehg. Shapeclipper: Scalable 3d shape learning from single-view images via geometric and clip-based consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12912–12922, 2023.

[13] Zixuan Huang, Stefan Stojanov, Anh Thai, Varun Jampani, and James M Rehg. Zeroshape: Regression-based zero-shot shape reconstruction. *arXiv preprint arXiv:2312.14198*, 2023. 1, 2, 4

[14] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *arXiv preprint arXiv:2311.06214*, 2023. 2

[15] Xiaoyu Li, Qi Zhang, Di Kang, Weihao Cheng, Yiming Gao, Jingbo Zhang, Zhihao Liang, Jing Liao, Yan-Pei Cao, and Ying Shan. Advances in 3d generation: A survey. *arXiv preprint arXiv:2401.17807*, 2024. 2

[16] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems*, 36, 2024. 4

[17] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023. 1, 2

[18] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2

[19] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 2

[20] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 2

[21] Zifan Shi, Sida Peng, Yinghao Xu, Andreas Geiger, Yiyi Liao, and Yujun Shen. Deep generative models on 3d representations: A survey. *arXiv preprint arXiv:2210.15663*, 2022. 2

[22] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 2

[23] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024. 2

[24] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018.

[25] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. *arXiv preprint arXiv:2311.12024*, 2023. 2

[26] 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. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[27] Chao-Yuan Wu, Justin Johnson, Jitendra Malik, Christoph Feichtenhofer, and Georgia Gkioxari. Multiview compressive coding for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9065–9075, 2023. 2

[28] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. *arXiv preprint arXiv:2312.02981*, 2023. 1

[29] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 803–814, 2023. 4, 5

[30] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein,

Zexiang Xu, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. *arXiv preprint arXiv:2311.09217*, 2023. 2

[31] Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Josh Tenenbaum, Bill Freeman, and Jiajun Wu. Learning to reconstruct shapes from unseen classes. *Advances in neural information processing systems*, 31, 2018. 2

[32] Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12588–12597, 2023. 1

[33] Zi-Xin Zou, Weihao Cheng, Yan-Pei Cao, Shi-Sheng Huang, Ying Shan, and Song-Hai Zhang. Sparse3d: Distilling multiview-consistent diffusion for object reconstruction from sparse views. *arXiv preprint arXiv:2308.14078*, 2023.

[34] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. *arXiv preprint arXiv:2312.09147*, 2023. 1, 2, 4