LGM: Large Multi-View Gaussian Model for High-Resolution 3D Content Creation

Jiaxiang Tang[1⋆], Zhaoxi Chen[2], Xiaokang Chen[1], Tengfei Wang[3], Gang Zeng[1], and Ziwei Liu[2]

[1] National Key Lab of General AI, Peking University
[2] S-Lab, Nanyang Technological University
[3] Shanghai AI Lab

https://me.kiui.moe/lgm

Fig. 1: Our method generates **high-resolution** 3D Gaussians in **5 seconds** from single-view images or texts.

Abstract. 3D content creation has achieved significant progress in terms of both quality and speed. Although current feed-forward models can produce 3D objects in seconds, their resolution is constrained by the intensive computation required during training. In this paper, we introduce **Large Multi-View Gaussian Model (LGM)**, a novel framework designed to generate high-resolution 3D models from text prompts or single-view images. Our key insights are two-fold: **1) 3D Representation:** We propose multi-view Gaussian features as an efficient yet powerful representation, which can then be fused together for differentiable rendering. **2) 3D Backbone:** We present an asymmetric U-Net as a high-throughput backbone operating on multi-view images, which can be

⋆ Work done while visiting S-Lab, Nanyang Technological University.

produced from text or single-view image input by leveraging multi-view diffusion models. Extensive experiments demonstrate the high fidelity and efficiency of our approach. Notably, we maintain the fast speed to generate 3D objects within 5 seconds while boosting the training resolution to 512, thereby achieving high-resolution 3D content generation.

Keywords: 3D Generation · Gaussian Splatting · High Resolution

1 Introduction

Automatic 3D content creation has great potential in numerous fields such as digital games, virtual reality, and films. The fundamental techniques, like image-to-3D and text-to-3D, provide significant benefits by remarkably decreasing the requirement for manual labor among professional 3D artists, enabling those without expertise to participate in 3D asset creation.

Previous research on 3D generation has predominantly focused on score distillation sampling (SDS) [22,24,36,47] to lift 2D diffusion priors into 3D. These optimization-based methods can create highly detailed 3D objects from text or single-view image inputs, but they often face issues such as slow generation speed and limited diversity. Recent advancements have significantly decreased the time required to generate 3D objects using large reconstruction models from single-view or few-shot images [15,19,52,55,57]. These methods utilize transformers to directly regress triplane-based [2] neural radiance fields (NeRF) [32]. However, these methods cannot produce detailed textures and complicated geometry due to the low-resolution training. We argue that their bottlenecks are **1)** inefficient 3D representation, and **2)** heavily parameterized 3D backbone. For instance, given a fixed compute budget, the triplane representation of LRM [15] is limited to the resolution of 32, while the resolution of the rendered image is capped at 128 due to the online volume rendering. Despite this, these methods suffer from the computationally intensive transformer-based backbone, which also leads to a limited training resolution.

To address these challenges, we present a novel method to train a few-shot 3D reconstruction model without relying on triplane-based volume rendering or transformers [15]. Instead, our approach employs 3D Gaussian splatting [17] of which features are predicted by an asymmetric U-Net as a high-throughput backbone [40,46]. The motivation of this design is to achieve high-resolution 3D generation, which necessitates an expressive 3D representation and the ability to train at high resolutions. Gaussian splatting stands out for 1) the expressiveness of compactly representing a scene compared with a single triplane, and 2) rendering efficiency compared with heavy volume rendering, which facilitates high-resolution training. However, it requires a sufficient number of 3D Gaussians to accurately represent detailed 3D information. Inspired by splatter image [46], we found that U-Net is effective in generating a sufficient number of Gaussians from multiview pixels, which maintains the capacity for high-resolution training at the same time. Note that, compared to previous methods [15,62], our default model is capable of generating 3D models with up to $65,536$ Gaussians and can be

trained at a resolution of 512, while still maintaining the rapid generation speed of feed-forward regression models. As shown in Figure 1, our model supports both image-to-3D and text-to-3D tasks, capable of producing high-resolution, richly detailed 3D Gaussians in approximately 5 seconds.

Our method adopts a multi-view reconstruction setting similar to Instant3D [19]. In this process, the image and camera embedding from each input view are transformed into a feature map, which can be decoded and fused as a set of Gaussians. Differentiable rendering is applied to render novel views from the fused 3D Gaussians, allowing end-to-end image-level supervision in high resolution. To enhance information sharing across all input views, attention blocks are integrated into the deeper layers of the U-Net. This enables us to train our network on multi-view image datasets [12] using only regressing objectives. During inference, our method leverages existing image or text to multi-view diffusion models [27, 43, 44, 51] to produce multi-view images as inputs for our Gaussian fusion network. To overcome the domain gap between multi-view images rendered from actual 3D objects and synthesized using diffusion models, we further propose two proper data augmentations for robust training. Finally, considering the preference for polygonal meshes in downstream tasks, we design a general algorithm to convert generated 3D Gaussians to smooth and textured meshes.

In summary, our contributions are:

1. We propose a novel framework to generate high-resolution 3D Gaussians by fusing information from multi-view images, which can be generated from text prompts or single-view images.
2. We design an asymmetric U-Net based architecture for efficient end-to-end training with significantly higher resolution, investigate data augmentation techniques for robust training, and propose a general mesh extraction approach from 3D Gaussians.
3. Extensive experiments demonstrate the superior quality, resolution, and efficiency of our method in both text-to-3D and image-to-3D tasks.

2 Related Work

High-Resolution 3D Generation. Current approaches for generating high-fidelity 3D models mostly rely on SDS-based optimization techniques. It requires both an expressive 3D representation and high-resolution supervision to effectively distill detailed information from 2D diffusion models into 3D. Due to the significant memory consumption associated with high-resolution rendering of NeRF, Magic3D [22] first converts NeRF to DMTet [42] and subsequently trains a second stage for finer resolution refinement. The hybrid representation of DMTet geometry and hash grid [34] textures enables the capture of high-quality 3D information, which can be efficiently rendered using differentiable rasterization [18]. Fantasia3D [6] explores to directly train DMTet with disentangled geometry and appearance generation. Subsequent studies [8, 20, 21, 47, 49, 54] also employ a similar mesh-based stage, enabling high-resolution supervision for enhanced detail. Another promising 3D representation is Gaussian splatting [17]

for its expressiveness and efficient rendering capabilities. Nonetheless, achieving rich details with this method necessitates appropriate initialization and careful densification during optimization [10, 59]. In contrast, our work investigates a feed-forward approach to directly generate a sufficient number of 3D Gaussians.

Efficient 3D Generation. In contrast to SDS-based optimization methods, feed-forward 3D native methods are able to generate 3D assets within seconds after training on large-scale 3D datasets [11, 12]. Some works attempt to train text-conditioned diffusion models on 3D representations such as point clouds and volumes [1, 5, 9, 16, 26, 33, 35, 53, 58, 61]. However, these methods either cannot generalize well to large datasets or only produce low-quality 3D assets with simple textures. Recently, LRM [15] first shows that a regression model can be trained to robustly predict NeRF from a single-view image in just 5 seconds, which can be further exported to meshes. Instant3D [19] trains a text to multi-view images diffusion model and a multi-view LRM to perform fast and diverse text-to-3D generation. The following works extend LRM to predict poses given multi-view images [52], combine with diffusion [57], and specialize on human data [55]. These feed-forward models can be trained with simple regression objectives and significantly accelerate the speed of 3D object generation. However, their triplane NeRF-based representation is restricted to a relatively low resolution and limits the final generation fidelity. Our model instead seeks to train a high-fidelity feed-forward model using Gaussian splatting and U-Net.

Gaussian Splatting for Generation. We specifically discuss recent methods in generation tasks using Gaussian splatting [4, 7, 23, 38, 56]. DreamGaussian [47] first combines 3D Gaussians with SDS-based optimization approaches to decrease generation time. GSGen [10] and GaussianDreamer [59] explore various densification and initialization strategies for text to 3D Gaussians generation. Despite the acceleration achieved, generating high-fidelity 3D Gaussians using these optimization-based methods still requires several minutes. TriplaneGaussian [62] introduces Gaussian splatting into the framework of LRM. This method starts by predicting Gaussian centers as point clouds and then projects them onto a triplane for other features. Nonetheless, the number of Gaussians and the resolution of the triplane are still limited, affecting the quality of the generated Gaussians. Splatter image [46] proposes to predict 3D Gaussians as pixels on the output feature map using U-Net from single-view images. This approach mainly focuses on single-view or two-view scenarios, limiting its generalization to large-scale datasets. Similarly, PixelSplat [3] predicts Gaussian parameters for each pixel of two posed images from scene datasets. We design a 4-view reconstruction model combined with existing multi-view diffusion models for general text or image to high-fidelity 3D object generation.

3 Large Multi-View Gaussian Model

We first provide the background information on Gaussian splatting and multi-view diffusion models (Section 3.1). Then we introduce our high-resolution 3D content generation framework (Section 3.2), where the core part is an asymmet-

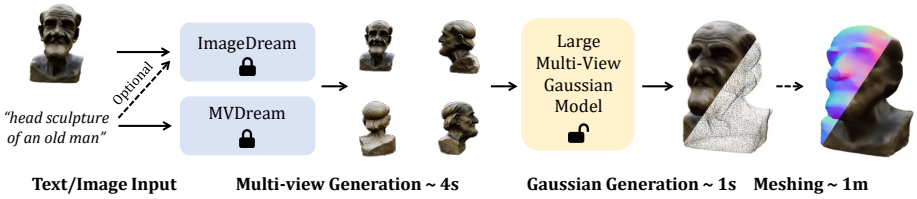

Fig. 2: Pipeline. Our model is trained to reconstruct 3D Gaussians from multi-view images, which can be synthesized by off-the-shelf models [44,51] at inference time from only text, or only image, or both input. Polygonal meshes can be extracted optionally.

ric U-Net backbone to predict and fuse 3D Gaussians from multi-view images (Section 3.3). We design careful data augmentation and training pipeline to enhance robustness and stability (Section 3.4). Finally, we describe an effective method for smooth textured mesh extraction from the generated 3D Gaussians (Section 3.5).

3.1 Preliminaries

Gaussian Splatting. As introduced in [17], Gaussian splatting employs a collection of 3D Gaussians to represent 3D data. Specifically, each Gaussian is defined by a center $\mathbf{x} \in \mathbb{R}^3$, a scaling factor $\mathbf{s} \in \mathbb{R}^3$, and a rotation quaternion $\mathbf{q} \in \mathbb{R}^4$. Additionally, an opacity value $\alpha \in \mathbb{R}$ and a color feature $\mathbf{c} \in \mathbb{R}^C$ are maintained for rendering, where spherical harmonics can be used to model view-dependent effects. These parameters can be collectively denoted by Θ, with $\Theta_i = \{\mathbf{x}_i, \mathbf{s}_i, \mathbf{q}_i, \alpha_i, \mathbf{c}_i\}$ representing the parameters for the i-th Gaussian. Rendering of the 3D Gaussians involves projecting them onto the image plane as 2D Gaussians and performing alpha composition for each pixel in front-to-back depth order, thereby determining the final color and alpha.

Multi-View Diffusion Models. Original 2D diffusion models [39,41] primarily focus on generating single-view images and do not support 3D viewpoint manipulation. Recently, several methods [20,27,43,44,51] propose to fine-tune multi-view diffusion models on 3D datasets to incorporate camera poses as an additional input. These approaches enable the creation of multi-view images of the same object, either from a text prompt or a single-view image. However, due to the absence of an actual 3D model, inconsistencies may still occur across the generated views.

3.2 Overall Framework

As illustrated in Figure 2, we adopt a two-step 3D generation pipeline at inference. Firstly, we take advantage of off-the-shelf text or image to multi-view diffusion models to generate multi-view images. Specifically, we adopt MVDream [44]

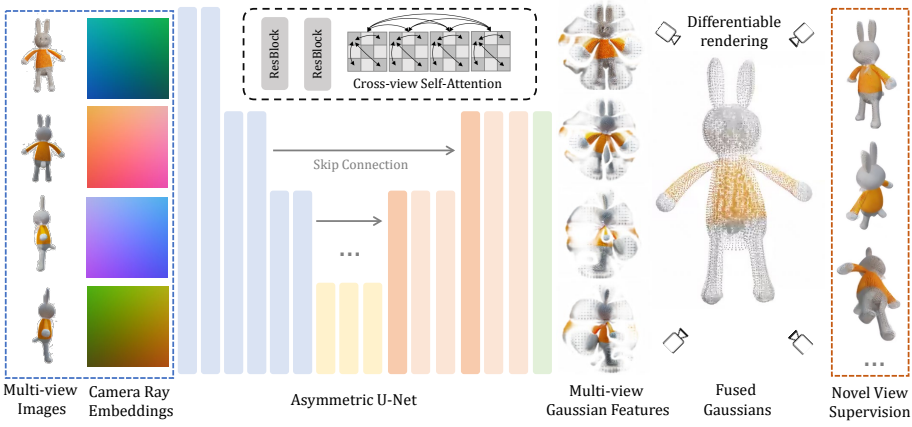

Fig. 3: Architecture of LGM. Our network adopts an asymmetric U-Net based architecture with cross-view self-attentions. We take four images with camera ray embeddings as the input, and output four feature maps which are interpreted and fused into 3D Gaussians. The Gaussians are then rendered at novel views and supervised with ground truth images.

for text input and ImageDream [51] for image (and optionally text) input. Both models are designed to generate multi-view images at four orthogonal azimuths and a fixed elevation. In the second step, we use a U-Net based model to predict 3D Gaussians from these sparse view images. Specifically, our model is trained to take four images with camera pose embeddings as input and predict four sets of Gaussians, which are fused to form the final 3D Gaussians. The generated Gaussians can be optionally converted to polygonal meshes using an extra conversion step, which is more suitable for downstream tasks.

3.3 Asymmetric U-Net for 3D Gaussians

At the core of our framework is an asymmetric U-Net to predict and fuse Gaussians from multi-view images. The network architecture is shown in Figure 3. We take four images and corresponding camera poses as the input. Following previous works [57], we use the Plücker ray embedding to densely encode the camera poses. The RGB value and ray embedding are concatenated into a 9-channel feature map as the input to the first layer:

$$\mathbf{f}_i = \{\mathbf{c}_i, \mathbf{o}_i \times \mathbf{d}_i, \mathbf{d}_i\} \tag{1}$$

where $\mathbf{f}_i$ is the input feature for pixel i, $\mathbf{c}_i$ is the RGB value, $\mathbf{d}_i$ is the ray direction, and $\mathbf{o}_i$ is the ray origin.

The U-Net is built with residual layers [13] and self-attention layers [50] similar to previous works [14, 31, 46]. We only add self-attention at deeper layers where the feature map resolution is down-sampled to save memory. To propagate information across multiple views, we flatten the four image features and

concatenate them before applying self-attention, similar to previous multi-view diffusion models [44, 51].

Each pixel of the output feature map is treated as a 3D Gaussian inspired by splatter image [46]. Differently, our U-Net is designed to be asymmetric with a smaller output resolution compared to input, which allows us to use higher resolution input images and limit the number of output Gaussians. We discard the depth prediction required by explicit ray-wise camera projection in [46]. The output feature map contains 14 channels corresponding to the original attributes of each Gaussian Θ_i. To stabilize the training, we choose some different activation functions compared to the original Gaussian Splatting [17]. We clamp the predicted positions $\mathbf{x}_i$ into $[-1, 1]^3$, and multiply the softplus-activated scales $\mathbf{s}_i$ with 0.1, such that the generated Gaussians at the beginning of training is close to the scene center. For each input view, the output feature map is transformed into a set of Gaussians. We simply concatenate these Gaussians from all four views as the final 3D Gaussians, which are used to render images at novel views for supervision.

3.4 Robust Training

Data Augmentation. We use multi-view images rendered from the Objaverse [12] dataset for training. However, at inference, we use synthesized multi-view images by diffusion models [44, 51]. To mitigate the domain gap between these different multi-view images, we design two types of data augmentation for more robust training.

Grid Distortion. Synthesizing 3D consistent multi-view images using 2D diffusion models has been explored by many works [25,43,44,51]. However, since there is no underlying 3D representation, the generated multi-view images often suffer from subtle inconsistency across different views. We try to simulate such inconsistency using grid distortion. Except for the first input view, which is usually the front reference view, the other three input views are randomly distorted with a random grid during training. This makes the model more robust to inconsistent multi-view input images.

Orbital Camera Jitter. Another problem is that the synthesized multi-view images may not accurately follow the given camera poses. Following [15], we always normalized the camera poses at each training step such that the first view's camera pose is fixed. We therefore apply camera jitter to the last three input views during training. Specifically, we randomly rotate the camera pose orbiting the scene center so the model is more tolerant to inaccurate camera poses and ray embeddings.

Loss Function. To supervise the concatenated Gaussians, we use the differentiable renderer implementation from [17] to render them. At each training step, we render the RGB image and alpha image of eight views, including four input views and four novel views. Following [15], we apply mean square error loss and

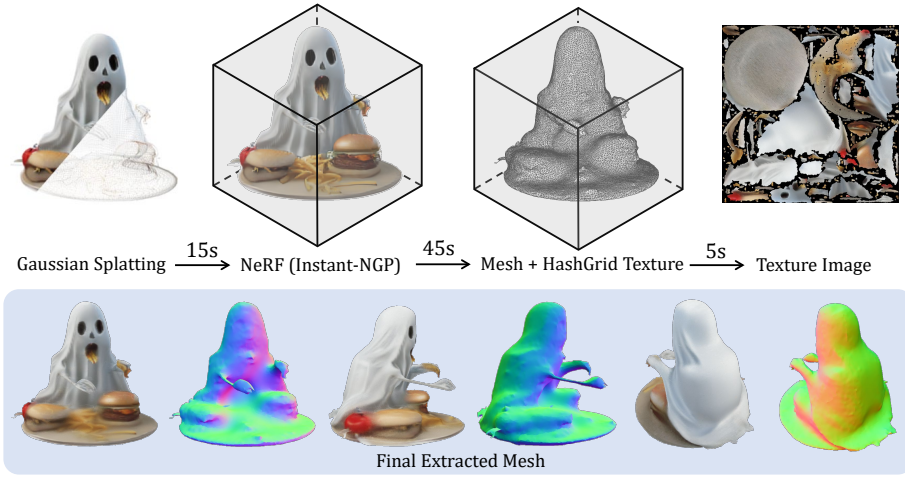

Fig. 4: Mesh Extraction Pipeline. We implement an efficient pipeline to convert the 3D Gaussians into smooth and textured meshes.

VGG-based LPIPS loss [60] to the RGB image:

$$\mathcal{L}_{\text{rgb}} = \mathcal{L}_{\text{MSE}}(I_{\text{rgb}}, I^{\text{GT}}_{\text{rgb}}) + \lambda \mathcal{L}_{\text{LPIPS}}(I_{\text{rgb}}, I^{\text{GT}}_{\text{rgb}}) \tag{2}$$

We further apply mean square error loss on the alpha image for faster convergence of the shape:

$$\mathcal{L}_{\alpha} = \mathcal{L}_{\text{MSE}}(I_{\alpha}, I^{\text{GT}}_{\alpha}) \tag{3}$$

3.5 Mesh Extraction

Since polygonal meshes are still the most widely used 3D representation in downstream tasks, we hope to further extract meshes from our generated Gaussians. Previous works [47] have tried to directly convert the opacity value of 3D Gaussians into an occupancy field for mesh extraction. However, we find this method dependent on aggressive densification during the optimization of 3D Gaussians to produce smooth occupancy field. On the contrary, the generated Gaussians in our method are usually sparse and cannot produce a suitable occupancy field, leading to an unsatisfactory surface with visible holes.

Instead, we propose a more general mesh extraction pipeline from 3D Gaussians as illustrated in Figure 4. We first train an efficient NeRF [34] using the rendered images from 3D Gaussians on-the-fly, and then convert the NeRF to polygonal meshes [48]. Specifically, we train two hash grids to reconstruct the geometry and appearance from Gaussian renderings. Marching Cubes [28] is applied to extract a coarse mesh, which is then iteratively refined together with the appearance hash grid using differentiable rendering. Finally, we bake the

appearance field onto the refined mesh to extract texture images. For more details, please refer to the supplementary materials and NeRF2Mesh [48]. With adequately optimized implementation, it takes only about 1 minute to perform this Gaussians to NeRF to mesh conversion.

4 Experiments

4.1 Implementation Details

Datasets. We use a filtered subset of the Objaverse [12] dataset to train our model. Since there are many low-quality 3D models (*e.g.*, partial scans, missing textures) in the original Objaverse dataset, we filter the dataset by two empirical rules: (1) We manually examine the captions and rendered images from Cap3D [30], and curate a list of words that usually appears in bad models (*e.g.*, 'resembling', 'debris', 'frame'), which is then used to filter all models whose caption includes any of these words. (2) We discard models with mostly white color after rendering, which usually indicates missing texture. These lead to a final set of around 80K 3D objects. We render the RGBA image from 100 camera views at the resolution of 512×512 for training and validation.

Network Architecture. Our asymmetric U-Net model consists of 6 down blocks, 1 middle block, and 5 up blocks, with the input image at 256×256 and output Gaussian feature map at 128×128. We use 4 input views, so the number of output Gaussians is $128 \times 128 \times 4 = 65,536$. The feature channels for all blocks are $[64, 128, 256, 512, 1024, 1024]$, $[1024]$ and $[1024, 1024, 512, 256, 128]$ respectively. Each block contains a series of residual layers and an optional downsample or up-sample layer. For the last 3 of down blocks, the middle block, and the first 3 up blocks, we also insert cross-view self-attention layers after the residual layers. The final feature maps are processed by a 1×1 convolution layer to 14-channel pixel-wise Gaussian features. Following previous works [39, 46], we adopt `Silu` activation and group normalization for the U-Net.

Training. We train our model on 32 NVIDIA A100 (80G) GPUs for about 4 days. A batch size of 8 for each GPU is used under `bfloat16` precision, leading to an effective batch size of 256. For each batch, we randomly sample 8 camera views, with the first 4 views as the input, and all 8 views as the output for supervision. Similar to LRM [15], we transform the cameras of each batch such that the first input view is always the front view with an identity rotation matrix and fixed translation. The input images are assumed to have a white background. The output 3D Gaussians are rendered at 512×512 resolution for mean square error loss. We resize the images to 256×256 for LPIPS loss to save memory. The AdamW [29] optimizer is adopted with the learning rate of 4×10^{-4}, weight decay of 0.05, and betas of $(0.9, 0.95)$. The learning rate is cosine annealed to 0 during the training. We clip the gradient with a maximum norm of 1.0. The probability for grid distortion and camera jitter is set to 50%.

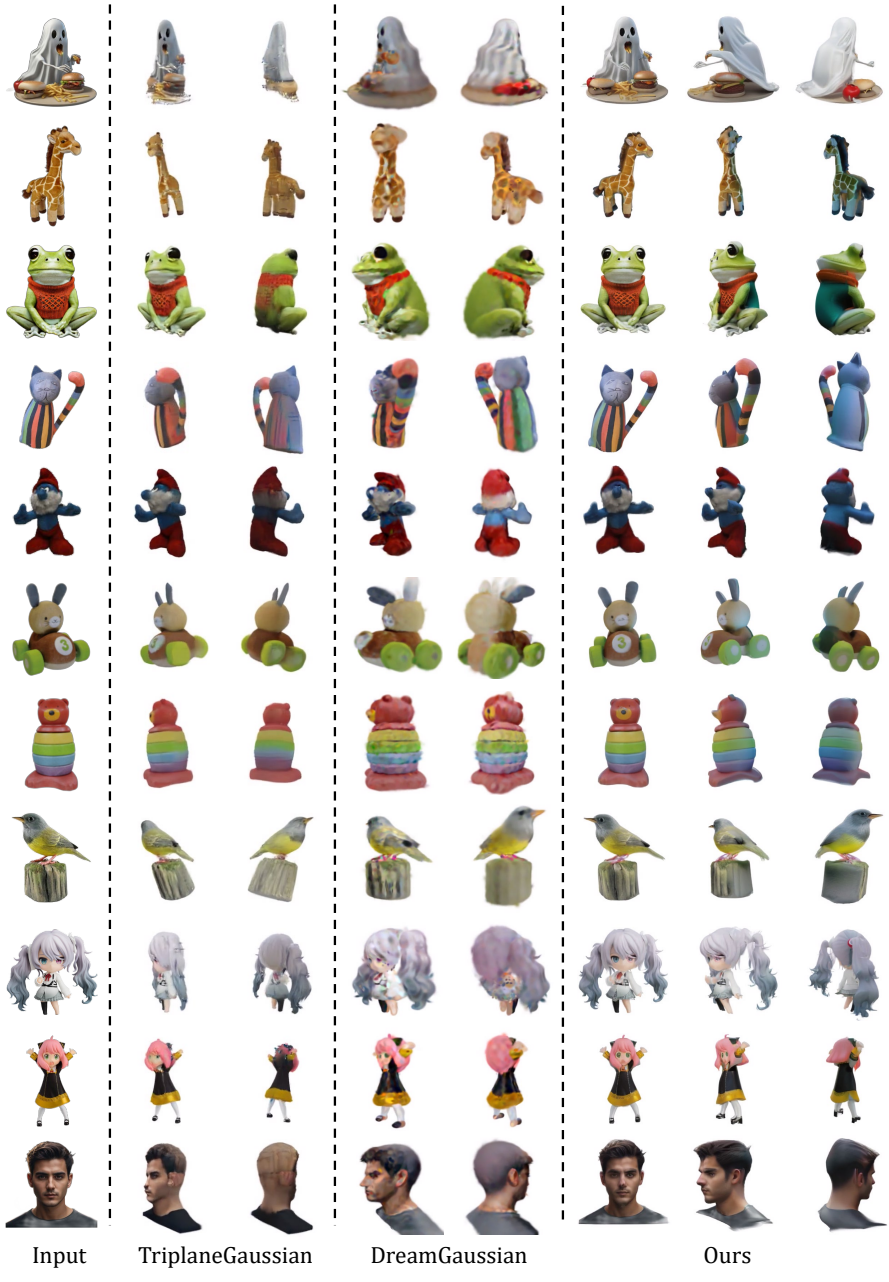

Fig. 5: Comparisons of generated 3D Gaussians for image-to-3D. Our method generates Gaussian splatting with better visual quality on various challenging images.

Fig. 6: Comparisons with LRM for image-to-3D. We compare our method with available results from LRM [15].

Inference. Our whole pipeline, including two multi-view diffusion models, takes only about 10 GB of GPU memory for inference, which is friendly for deployment. For the multi-view diffusion models, we use a guidance scale of 5 for ImageDream [51] and 7.5 for MVDream [44] following the original paper. The number of diffusion steps is set to 30 using the DDIM [45] scheduler. The camera elevation is fixed to 0, and azimuths to $[0, 90, 180, 270]$ degree for the four generated views. For ImageDream [51], the text prompt is always left empty so the only input is a single-view image. Since the images generated by MVDream may contain various backgrounds, we apply background removal [37] and use white background.

4.2 Qualitative Comparisons

Image-to-3D. We first compare against recent methods [47,62] that are capable of generating 3D Gaussians. Figure 5 shows images rendered from the generated 3D Gaussians for comparison. The 3D Gaussians produced by our method have better visual quality and effectively preserve the content from the input view. Our high-resolution 3D Gaussians can be transformed into smooth textured meshes with minimal loss of quality in most cases. We also compare our results against LRM [15] using the available videos from their website in Figure 6. Specifically, our multi-view setting successfully mitigates the issue of blurry back views and flat geometry, resulting in enhanced detail even in unseen views.

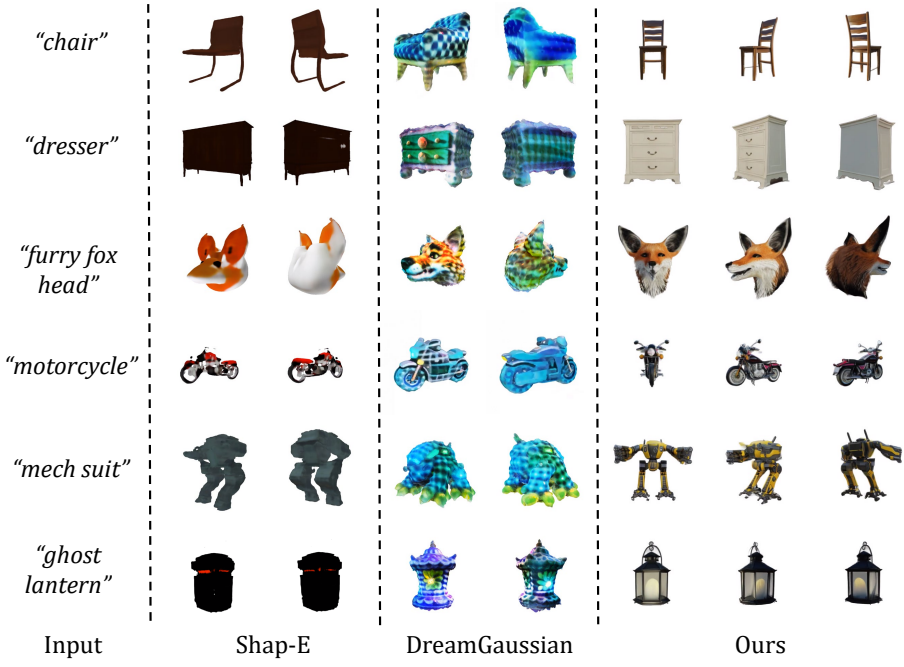

Fig. 7: Comparisons of generated 3D models for text-to-3D. Our method achieves better text alignment and visual quality.

Fig. 8: Diversity of our 3D generation. We can generate diverse 3D models given an ambiguous text description or single-view image.

Text-to-3D. We then compare with recent methods [16,47] on text-to-3D tasks. We observe an enhanced quality and efficiency in our method, generating more

	Image Consistency ↑	Overall Quality ↑
DreamGaussian [47]	2.30	1.98
TriplaneGaussian [62]	3.02	2.67
LGM (Ours)	**4.18**	**3.95**

Table 1: User Study on the quality of generated 3D Gaussians for image-to-3D tasks. The rating is of scale 1-5, the higher the better.

realistic 3D objects, as illustrated in Figure 7. Due to the multi-view diffusion models, our model is also free from multi-face problems.

Diversity. Notably, our pipeline exhibits high diversity in 3D generation, owing to the capability of multi-view diffusion model [44, 51]. As shown in Figure 8, with different random seeds, we can generate a variety of feasible objects from the same ambiguous text prompt or single-view image.

4.3 Quantitative Comparisons

We majorly conduct a user study to quantitatively evaluate our image-to-3D Gaussians generation performance. For a collection of 30 images, we render 360-degree rotating videos of the 3D Gaussians generated from DreamGaussain [47] (only the first stage), TriplaneGaussian [62], and ours. There are in total 90 videos for evaluation in our user study. Each volunteer is shown 30 samples from mixed random methods, and asked to rate in two aspects: image consistency and overall model quality. We collect results from 20 volunteers and get 600 valid scores in total. As shown in Table 1, our method is preferred as it aligns with the original image content and shows better overall quality.

4.4 Ablation Study

Number of Views. We train an image-to-3D model with only one input views similar to splatter image [46], *i.e.*, without the multi-view generation step. The U-Net takes the single input view as input with self-attention, and outputs Gaussian features as in our multi-view model. To compensate the number of Gaussians, we predict two Gaussians for each pixel of the output feature maps, leading to $128 \times 128 \times 2 = 32,768$ Gaussians. As illustrated in the top-left part of Figure 9, the single-view model can reconstruct faithful front-view, but fails to distinguish the back view and results in blurriness. This is as expected since the regressive U-Net is more suitable for reconstruction tasks, and it's hard to generalize to large datasets in our experiments.

Data Augmentation. We train a smaller model with or without applying data augmentation to validate its effectiveness. Although we observe a lower training

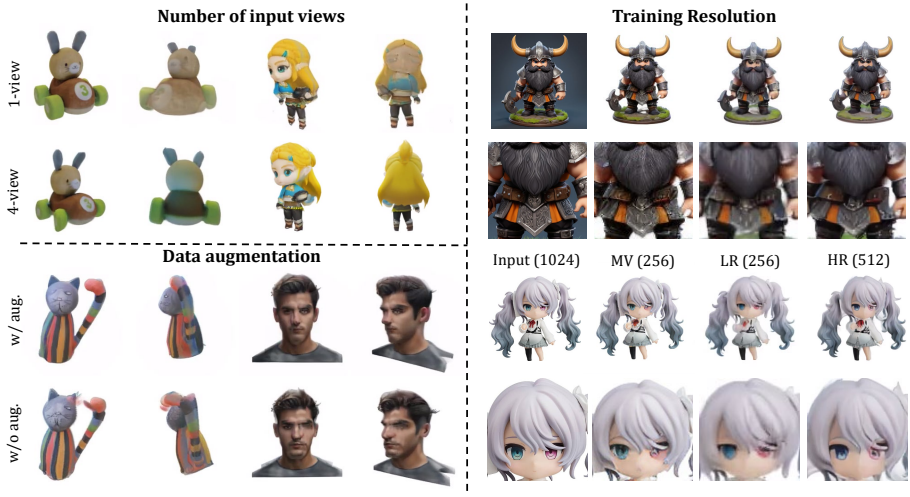

Fig. 9: Ablation Study. We carry out ablation study on designs of our method.

loss for the model without data augmentation, the domain gap during inference leads to more floaters and worse geometry as shown in the bottom-left part of Figure 9. The model with data augmentation strategy can better correct the 3D inconsistency and inaccurate camera poses in the generated multi-view images.

Training Resolution. Lastly, we train a model with a fewer number of Gaussians and smaller rendering resolution as in the right part of Figure 9. We remove the last up block of the U-Net so the number of output Gaussians is $64 \times 64 \times 4 = 16,384$, and we render it at 256×256 for supervision. The model can still converge and successfully reconstruct 3D Gaussians, but the details are worse compared to the 256×256 input multi-view images. In contrast, our large resolution model at 512×512 can capture better details and generate Gaussians with higher resolution.

4.5 Limitations

Despite the promising results, our method still has some limitations. Since our model is essentially a multi-view reconstruction model, the 3D generation quality highly depends on the quality of four input views. However, current multi-view diffusion models [44, 51] are far from perfect: (1) There can be 3D inconsistency which misleads the reconstruction model to generate floaters in the 3D Gaussians. (2) The resolution of synthesized multi-view images is restricted to 256×256, constraining our model to further improve resolution. (3) ImageDream [51] also fails to handle input image with a large elevation angle. We expect these limitations can be mitigated with better multi-view diffusion models in future works.

5 Conclusion

In this work, we present a large multi-view Gaussian model for high-resolution 3D content generation. Our model, distinct from previous methods reliant on NeRF and transformers, employs Gaussian splatting and U-Net to address the challenges of high memory requirements and low-resolution training. Additionally, we explore data augmentation for better robustness, and introduce a mesh extraction algorithm for the generated 3D Gaussians. Our approach achieves both high-resolution and high-efficiency for 3D objects generation, proving its versatility and applicability in various contexts.

Acknowledgements. This work is supported by the Sichuan Science and Technology Program (2023YFSY0008), China Tower-Peking University Joint Laboratory of Intelligent Society and Space Governance, National Natural Science Foundation of China (61632003, 61375022, 61403005), Grant SCITLAB-20017 of Intelligent Terminal Key Laboratory of SiChuan Province, Beijing Advanced Innovation Center for Intelligent Robots and Systems (2018IRS11), and PEK-SenseTime Joint Laboratory of Machine Vision. This project is also funded by the Ministry of Education, Singapore, under its MOE AcRF Tier 2 (MOE-T2EP20221-0012), NTU NAP, and under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner(s).

References

1. Cao, Z., Hong, F., Wu, T., Pan, L., Liu, Z.: Large-vocabulary 3d diffusion model with transformer. arXiv preprint arXiv:2309.07920 (2023) 4
2. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., Mello, S.D., Gallo, O., Guibas, L., Tremblay, J., Khamis, S., Karras, T., Wetzstein, G.: Efficient geometry-aware 3D generative adversarial networks. In: CVPR (2022) 2
3. Charatan, D., Li, S., Tagliasacchi, A., Sitzmann, V.: pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. arXiv preprint arXiv:2312.12337 (2023) 4
4. Chen, G., Wang, W.: A survey on 3d gaussian splatting. arXiv preprint arXiv:2401.03890 (2024) 4
5. Chen, H., Gu, J., Chen, A., Tian, W., Tu, Z., Liu, L., Su, H.: Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. arXiv preprint arXiv:2304.06714 (2023) 4
6. Chen, R., Chen, Y., Jiao, N., Jia, K.: Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. arXiv preprint arXiv:2303.13873 (2023) 3
7. Chen, Y., Chen, Z., Zhang, C., Wang, F., Yang, X., Wang, Y., Cai, Z., Yang, L., Liu, H., Lin, G.: Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. arXiv preprint arXiv:2311.14521 (2023) 4
8. Chen, Y., Zhang, C., Yang, X., Cai, Z., Yu, G., Yang, L., Lin, G.: It3d: Improved text-to-3d generation with explicit view synthesis. arXiv preprint arXiv:2308.11473 (2023) 3

9. Chen, Z., Hong, F., Mei, H., Wang, G., Yang, L., Liu, Z.: Primdiffusion: Volumetric primitives diffusion for 3d human generation. arXiv preprint arXiv:2312.04559 (2023) 4
10. Chen, Z., Wang, F., Liu, H.: Text-to-3d using gaussian splatting. arXiv preprint arXiv:2309.16585 (2023) 4
11. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., et al.: Objaverse-xl: A universe of 10m+ 3d objects. arXiv preprint arXiv:2307.05663 (2023) 4
12. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: CVPR. pp. 13142–13153 (2023) 3, 4, 7, 9
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 6
14. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. NeurIPS **33**, 6840–6851 (2020) 6
15. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: Lrm: Large reconstruction model for single image to 3d. arXiv preprint arXiv:2311.04400 (2023) 2, 4, 7, 9, 11
16. Jun, H., Nichol, A.: Shap-e: Generating conditional 3d implicit functions. arXiv preprint arXiv:2305.02463 (2023) 4, 12
17. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ToG **42**(4), 1–14 (2023) 2, 3, 5, 7
18. Laine, S., Hellsten, J., Karras, T., Seol, Y., Lehtinen, J., Aila, T.: Modular primitives for high-performance differentiable rendering. ToG **39**(6) (2020) 3
19. Li, J., Tan, H., Zhang, K., Xu, Z., Luan, F., Xu, Y., Hong, Y., Sunkavalli, K., Shakhnarovich, G., Bi, S.: Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. arXiv preprint arXiv:2311.06214 (2023) 2, 3, 4
20. Li, W., Chen, R., Chen, X., Tan, P.: Sweetdreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d. arXiv preprint arXiv:2310.02596 (2023) 3, 5
21. Li, Y., Dou, Y., Shi, Y., Lei, Y., Chen, X., Zhang, Y., Zhou, P., Ni, B.: Focaldreamer: Text-driven 3d editing via focal-fusion assembly. arXiv preprint arXiv:2308.10608 (2023) 3
22. 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. pp. 300–309 (2023) 2, 3
23. Ling, H., Kim, S.W., Torralba, A., Fidler, S., Kreis, K.: Align your gaussians: Text-to-4d with dynamic 3d gaussians and composed diffusion models. arXiv preprint arXiv:2312.13763 (2023) 4
24. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. arXiv preprint arXiv:2303.11328 (2023) 2
25. Liu, Y., Lin, C., Zeng, Z., Long, X., Liu, L., Komura, T., Wang, W.: Syncdreamer: Generating multiview-consistent images from a single-view image. arXiv preprint arXiv:2309.03453 (2023) 7
26. Liu, Z., Feng, Y., Black, M.J., Nowrouzezahrai, D., Paull, L., Liu, W.: Meshdiffusion: Score-based generative 3d mesh modeling. arXiv preprint arXiv:2303.08133 (2023) 4
27. Long, X., Guo, Y.C., Lin, C., Liu, Y., Dou, Z., Liu, L., Ma, Y., Zhang, S.H., Habermann, M., Theobalt, C., et al.: Wonder3d: Single image to 3d using cross-domain diffusion. arXiv preprint arXiv:2310.15008 (2023) 3, 5

28. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: Seminal graphics: pioneering efforts that shaped the field, pp. 347–353 (1998) 8
29. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 9
30. Luo, T., Rockwell, C., Lee, H., Johnson, J.: Scalable 3d captioning with pretrained models. arXiv preprint arXiv:2306.07279 (2023) 9
31. Metzer, G., Richardson, E., Patashnik, O., Giryes, R., Cohen-Or, D.: Latent-nerf for shape-guided generation of 3d shapes and textures. arXiv preprint arXiv:2211.07600 (2022) 6
32. 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) 2
33. Müller, N., Siddiqui, Y., Porzi, L., Bulo, S.R., Kontschieder, P., Nießner, M.: Diffrf: Rendering-guided 3d radiance field diffusion. In: CVPR. pp. 4328–4338 (2023) 4
34. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM TOG (2022) 3, 8
35. 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) 4
36. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv preprint arXiv:2209.14988 (2022) 2
37. Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O.R., Jagersand, M.: U2-net: Going deeper with nested u-structure for salient object detection. Pattern recognition **106**, 107404 (2020) 11
38. Ren, J., Pan, L., Tang, J., Zhang, C., Cao, A., Zeng, G., Liu, Z.: Dreamgaussian4d: Generative 4d gaussian splatting. arXiv preprint arXiv:2312.17142 (2023) 4
39. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022) 5, 9
40. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. pp. 234–241. Springer (2015) 2
41. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. NeurIPS **35**, 36479–36494 (2022) 5
42. 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) 3
43. Shi, R., Chen, H., Zhang, Z., Liu, M., Xu, C., Wei, X., Chen, L., Zeng, C., Su, H.: Zero123++: a single image to consistent multi-view diffusion base model (2023) 3, 5, 7
44. Shi, Y., Wang, P., Ye, J., Long, M., Li, K., Yang, X.: Mvdream: Multi-view diffusion for 3d generation. arXiv preprint arXiv:2308.16512 (2023) 3, 5, 7, 11, 13, 14
45. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020) 11
46. Szymanowicz, S., Rupprecht, C., Vedaldi, A.: Splatter image: Ultra-fast single-view 3d reconstruction. In: arXiv (2023) 2, 4, 6, 7, 9, 13

47. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. arXiv preprint arXiv:2309.16653 (2023) 2, 3, 4, 8, 11, 12, 13, 19
48. Tang, J., Zhou, H., Chen, X., Hu, T., Ding, E., Wang, J., Zeng, G.: Delicate textured mesh recovery from nerf via adaptive surface refinement. arXiv preprint arXiv:2303.02091 (2022) 8, 9
49. Tsalicoglou, C., Manhardt, F., Tonioni, A., Niemeyer, M., Tombari, F.: Textmesh: Generation of realistic 3d meshes from text prompts. arXiv preprint arXiv:2304.12439 (2023) 3
50. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS **30** (2017) 6
51. Wang, P., Shi, Y.: Imagedream: Image-prompt multi-view diffusion for 3d generation. arXiv preprint arXiv:2312.02201 (2023) 3, 5, 6, 7, 11, 13, 14, 20
52. Wang, P., Tan, H., Bi, S., Xu, Y., Luan, F., Sunkavalli, K., Wang, W., Xu, Z., Zhang, K.: Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. arXiv preprint arXiv:2311.12024 (2023) 2, 4
53. Wang, T., Zhang, B., Zhang, T., Gu, S., Bao, J., Baltrusaitis, T., Shen, J., Chen, D., Wen, F., Chen, Q., et al.: Rodin: A generative model for sculpting 3d digital avatars using diffusion. In: CVPR. pp. 4563–4573 (2023) 4
54. 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. arXiv preprint arXiv:2305.16213 (2023) 3
55. Weng, Z., Liu, J., Tan, H., Xu, Z., Zhou, Y., Yeung-Levy, S., Yang, J.: Single-view 3d human digitalization with large reconstruction models. arXiv preprint arXiv:2401.12175 (2024) 2, 4
56. Xu, D., Yuan, Y., Mardani, M., Liu, S., Song, J., Wang, Z., Vahdat, A.: Agg: Amortized generative 3d gaussians for single image to 3d. arXiv preprint arXiv:2401.04099 (2024) 4
57. Xu, Y., Tan, H., Luan, F., Bi, S., Wang, P., Li, J., Shi, Z., Sunkavalli, K., Wetzstein, G., Xu, Z., et al.: Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. arXiv preprint arXiv:2311.09217 (2023) 2, 4, 6
58. Yariv, L., Puny, O., Neverova, N., Gafni, O., Lipman, Y.: Mosaic-sdf for 3d generative models. arXiv preprint arXiv:2312.09222 (2023) 4
59. Yi, T., Fang, J., Wu, G., Xie, L., Zhang, X., Liu, W., Tian, Q., Wang, X.: Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. arXiv preprint arXiv:2310.08529 (2023) 4
60. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) 8
61. Zhao, Z., Liu, W., Chen, X., Zeng, X., Wang, R., Cheng, P., Fu, B., Chen, T., Yu, G., Gao, S.: Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. arXiv preprint arXiv:2306.17115 (2023) 4
62. Zou, Z.X., Yu, Z., Guo, Y.C., Li, Y., Liang, D., Cao, Y.P., Zhang, S.H.: Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. arXiv preprint arXiv:2312.09147 (2023) 2, 4, 11, 13

A More Implementation Details

Datasets. The full list of words for filtering the Objaverse dataset is: 'flying, mountain, trash, featuring, a set of, a small, numerous, square, collection, broken, group, ceiling, wall, various, elements, splatter, resembling, landscape, stair, silhouette, garbage, debris, room, preview, floor, grass, house, beam, white, background, building, cube, box, frame, roof, structure'. The 100 camera views we use form a spiral path on the sphere surface. The camera radius is fixed to 1.5, and the field-of-view along the Y-axis is fixed to 49.1 degree.

B More Results

Fig. 10: Comparisons between different meshing method from Gaussians. We compare our meshing method with DreamGaussian [47].

Different Meshing Method. Figure 10 presents a comparison between our meshing algorithm and the technique introduced in DreamGaussian [47]. Our algorithm generates a smoother surface, which is advantageous for subsequent tasks such as relighting. Moreover, our method operates independently of the underlying 3D Gaussians, as it relies solely on the rendered images.

Limitations. We visualize failure cases of our method in Figure 11 to gain a deeper understanding of its weaknesses. As previously mentioned in the main paper, the primary causes of these failures stem from the flawed multi-view images produced in the initial step. The resolution of these multi-view images is limited

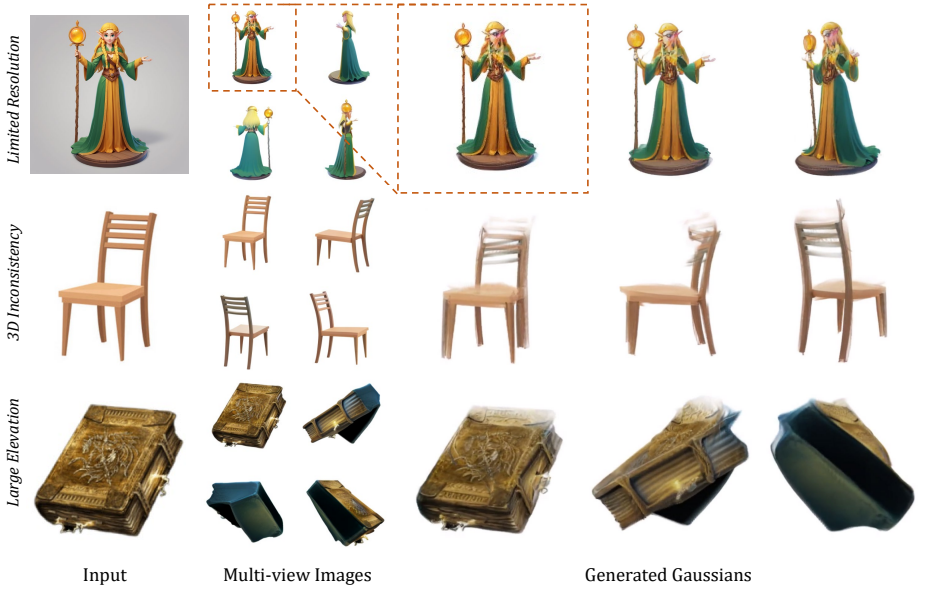

Fig. 11: Visualization of our limitations. We show three major reasons for failure cases of our method.

to 256×256, which can diminish the quality of the input image. Despite implementing data augmentation during training to emulate 3D inconsistencies and attempting to bridge the domain gap, this approach still results in inaccuracies for slender structures, such as chairs. Additionally, ImageDream [51] struggles with images that have significant elevation angle, occasionally producing images with a dark appearance.