

GRM: Large Gaussian Reconstruction Model for Efficient 3D Reconstruction and Generation

Yinghao Xu^{1*}, Zifan Shi^{1,2*}, Wang Yifan¹, Hansheng Chen¹
Ceyuan Yang³, Sida Peng⁴, Yujun Shen⁵, and Gordon Wetzstein¹

¹ Stanford University

² The Hong Kong University of Science and Technology

³ Shanghai AI Laboratory

⁴ Zhejiang University

⁵ Ant Group

Abstract. We introduce GRM, a large-scale reconstructor capable of recovering a 3D asset from sparse-view images in around 0.1s. GRM is a feed-forward transformer-based model that efficiently incorporates multi-view information to translate the input pixels into pixel-aligned Gaussians, which are unprojected to create a set of densely distributed 3D Gaussians representing a scene. Together, our transformer architecture and the use of 3D Gaussians unlock a *scalable* and *efficient* reconstruction framework. Extensive experimental results demonstrate the superiority of our method over alternatives regarding both reconstruction quality and efficiency. We also showcase the potential of GRM in generative tasks, i.e., text-to-3D and image-to-3D, by integrating it with existing multi-view diffusion models. Our project website is at: <https://justimyhxu.github.io/projects/grm/>.

Keywords: Gaussian splatting · 3D reconstruction · 3D generation

1 Introduction

The availability of high-quality and diverse 3D assets is critical in many domains, including robotics, gaming, architecture, among others. Yet, creating these assets has been a tedious manual process, requiring expertise in difficult-to-use computer graphics tools.

Emerging 3D generative models offer the ability to easily create diverse 3D assets from simple text prompts or single images [70]. Optimization-based 3D generative methods can produce high-quality assets, but they often require a long time—often hours—to produce a single 3D asset [50, 71, 93, 98, 101]. Recent feed-forward 3D generative methods have demonstrated excellent quality and diversity while offering significant speedups over optimization-based 3D generation approaches [2, 12, 30, 38, 46, 54, 78, 91, 106]. These state-of-the-art

* Equal Contribution

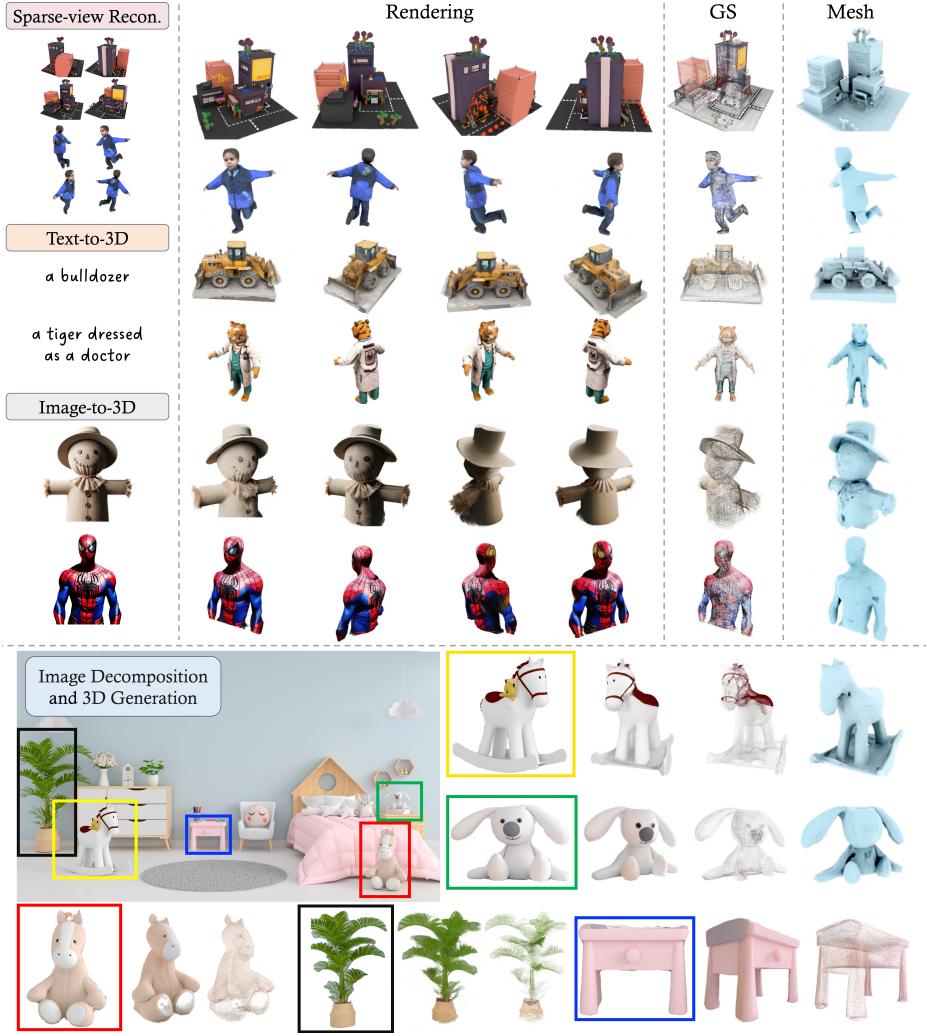


Fig. 1: High-fidelity 3D assets produced by GRM—a transformer-based reconstruction model built on 3D Gaussians. Trained for fast sparse-view reconstruction (top, ~0.1s), GRM works in synergy with other tools (e.g., text-to-multiview generation [46], image-to-multiview model [79], and 2D segmentation [45]) to enable text-to-3D (center top) and image-to-3D (center bottom) generation as well as real-world object reconstruction (bottom).

(SOTA) models, however, typically build on the triplane representation [5], which requires inefficient volume rendering. This inefficient rendering step not only hinders fast inference but it often also requires the models to operate at a reduced 3D resolution, limiting representational capacity.

We introduce the Gaussian Reconstruction Model (GRM) as a new feed-forward 3D generative model. At its core, GRM provides a novel sparse-view reconstructor that takes four different views of a 3D object as input and outputs the corresponding 3D scene. GRM implements two key insights: first, we replace the triplane scene representation of recent feed-forward generative frameworks [30, 46, 106] by 3D Gaussians; second, we design a pure transformer architecture to translate the set of input pixels to the set of pixel-aligned 3D Gaussians defining the output 3D scene. While parts of this architecture uses standard vision transformers (ViT) [19], we introduce a new upsampler that utilizes a variation of windowed self-attention layers [3]. This upsampler is unique in being able to efficiently pass non-local cues. As demonstrated in our experiment, it is critical for the reconstruction of high-frequency appearance details. Instead of attempting to synthesize missing regions from incomplete views—a highly ill-defined problem—we opt to train our model with sparse yet well-distributed views to cover enough information of a scene. This allows us to allocate the model’s capacity for fine-grained detail reconstruction, leading to significantly higher fidelity than relevant baselines for object-level 3D reconstruction. When combined with multi-view diffusion models, GRM achieves SOTA quality and speed for text-to-3D and single image-to-3D object generation.

Specifically, our contributions include

- a novel and efficient feed-forward 3D generative model that builds on 3D Gaussian splatting;
- the design of a sparse-view reconstructor using a pure transformer architecture, including encoder and upsampler, for pixel-to-3D Gaussian translation;
- the demonstration of SOTA quality and speed for object-level sparse-view 3D reconstruction and, when combined with existing multi-view diffusion models, also text-to-3D and image-to-3D generation.

2 Related Work

Sparse-view Reconstruction. Neural representations, as highlighted in prior works [9, 62–64, 69, 84, 86], present a promising foundation for scene representation and neural rendering [95]. When applied to novel-view synthesis, these methods have demonstrated success in scenarios with multi-view training images, showcasing proficiency in single-scene overfitting. Notably, recent advancements [10, 33, 51, 59, 100, 109] have extended these techniques to operate with a sparse set of views, displaying improved generalization to unseen scenes. These methods face challenges in capturing multiple modes within large-scale datasets, resulting in a limitation to generate realistic results. Recent works [30, 99, 114] further scale up the model and datasets for better generalization. But relying on neural volume-based scene representation proves inadequate for efficiently synthesizing high-resolution and high-fidelity images. Our proposed solution involves the use of pixel-aligned 3D Gaussians [8, 90] combined with our effective transformer architecture. This approach is designed to elevate both the efficiency and quality of the sparse-view reconstructor when provided with only four input images.

3D Generation. The advances of 3D GANs have set the foundation of 3D scene generation. Leveraging various successful 2D GAN architectures [4, 23, 37, 39–42, 112], 3D GANs [5, 6, 22, 24, 65, 67, 77, 82, 87, 88, 104, 105] combine 3D scene representations and neural rendering to generate 3D-aware content in a feed-forward fashion. Recently, Diffusion Models (DM) have emerged as a more powerful generative model, surpassing GANs in 2D generation [18, 29, 75, 89]. With its extension in 3D being actively explored, we review the most relevant work and refer readers to [70] for a comprehensive review. One research line seek to directly train 3D DMs using 3D [26, 36, 66, 68, 83] or 2D supervision [2, 12, 25, 38, 55, 78]. Though impressive, these work either cannot leverage the strong priors from pretrained 2D DMs or they suffer from 3D inconsistency. Other researchers propose to exploit 2D diffusion priors using an optimization procedure known as Score Distillation Sampling (SDS) and its variant [13, 15, 27, 49, 50, 55, 71, 81, 93, 98, 101]. These methods yield high-quality 3D generations, but require hours for the optimization process to converge. Therefore, there is a need to combine the feed-forward generation framework with expressive generative powers from DMs. To this end, many recent works first use 2D multi-view diffusion and then lift the multi-view inputs to 3D [7, 54, 56, 58, 79, 94, 96]. Recently, the Large Reconstruction Model (LRM) [30] scales up both the model and the dataset to predict a neural radiance field (NeRF) from single-view images. Although LRM is a reconstruction model, it can be combined with DMs to achieve text-to-3D and image-to-3D generation, as demonstrated by extensions such as Instant3D [46] and DMV3D [106]. Our method also builds on a strong reconstruction model and uses pretrained 2D DMs to provide input images for 3D generation in a feed-forward fashion. However, we adopt highly efficient 3D Gaussians [43] for representing and rendering a scene. This design lifts the computation and memory bottleneck posed by NeRF and volumetric rendering, allowing us to generate high-quality 3D assets within seconds.

Some concurrent works, such as LGM [92], AGG [103], and Splatter Image [90], also use 3D Gaussians in a feed-forward model. Our model differs from them in the choice of architecture—instead of using conventional convolution-based U-Net, we opt for a purely transformer-based encoder and a highly efficient transformer-based upsampler to generate a large number of pixel-aligned 3D Gaussians, which offering superior reconstruction quality.

Generalizable Gaussians. 3D Gaussians [43, 44] and differentiable splatting [43] have gained broad popularity thanks to their ability to efficiently reconstruct high-fidelity 3D scenes from posed images using only a moderate number of 3D Gaussians. This representation has been quickly adopted for various application, including image- or text-conditioned 3D [14, 47, 49] and 4D generation [52, 74], avatar reconstruction [31, 48, 72, 76, 113], dynamic scene reconstruction [61, 102, 107, 108], among others [11, 21, 97]. All of these aforementioned work focus on single-scene optimization, although very recent work also adopts 3D Gaussians into a GAN framework for 3D human generation [1].

3 Method

GRM is a feed-forward sparse-view 3D reconstructor, utilizing four input images to efficiently infer underlying 3D Gaussians [43]. Supplied with a multi-view image generator head [46, 79], GRM can be utilized to generate 3D from text or a single image. Different from LRM [30, 46, 99, 106], we leverage *pixel-aligned Gaussians* (Sec. 3.1) to enhance efficient and reconstruction quality and we adopt a transformer-based network to predict the properties of the Gaussians by associating information from all input views in a memory-efficient way (Sec. 3.2). Finally, we detail the training objectives in Sec. 3.3 and demonstrate high-quality text-to-3D and image-to-3D generation in a few seconds (Sec. 3.4).

3.1 Pixel-aligned Gaussians

3D Gaussians use a sparse set of primitives $\mathcal{G} = \{\mathbf{g}_i\}_{i=1}^N$ to represent the geometry and appearance of a 3D scene, where each Gaussian is parameterized with location $\boldsymbol{\mu} \in \mathbb{R}^3$, rotation quaternion $\mathbf{r} \in \mathbb{R}^4$, scale $\mathbf{s} \in \mathbb{R}^3$, opacity $o \in \mathbb{R}$, and the spherical harmonics (SH) coefficients $\mathbf{c} \in \mathbb{R}^D$, with D denoting the number of SH bases. These Gaussians can be rendered in real time using the differentiable rasterizer [43]. 3D Gaussians have gained tremendous popularity for single-scene optimization (SSO), but utilizing them in a generalizable framework remains challenging. A primary reason is that the properties of Gaussians are highly inter-dependent—multiple configurations can lead to the same visual representation, causing optimization difficulty. On the other hand, 3D Gaussians are an unstructured 3D representation, making it challenging to integrate effectively with neural networks for predictions which potentially increases the complexity of the prediction process. We introduce pixel-aligned Gaussians [8, 90] to address the above challenges. Instead of directly predicting a set of Gaussians and expect them to accurately cover the entire shape, we constrain the **Gaussians' locations** along the input viewing rays, *i.e.*,

$$\boldsymbol{\mu} = \mathbf{c}_o + \tau \mathbf{r}, \quad (1)$$

where **\mathbf{c}_o and \mathbf{r} denote the camera center and the ray direction**. Specifically, for every input view we predict a Gaussian attribute map $\mathbf{T} \in \mathbb{R}^{H \times W \times C}$ of $C = 12$ channels, **corresponding to depth τ , rotation, scaling, opacity, and the DC term of the SH coefficients**. We then unproject the pixel-aligned Gaussians into 3D, producing a total of $V \times H \times W$ densely distributed 3D Gaussians. Pixel-aligned Gaussians establish connections between input pixels and the 3D space in a more direct way, alleviating the learning difficulty, resulting to better reconstruction quality as we empirically show in Sec. 4.5.

3.2 Large Gaussian Reconstruction Model

In the following, we introduce our network, which transforms a set of input images $\mathcal{I} = \{\mathbf{I}_v\}_{v=1}^V$ and their camera poses $\mathcal{C} = \{\mathbf{c}_v\}_{v=1}^V$ to the Gaussian maps $\mathcal{T} = \{\mathbf{T}_v\}_{v=1}^V$.

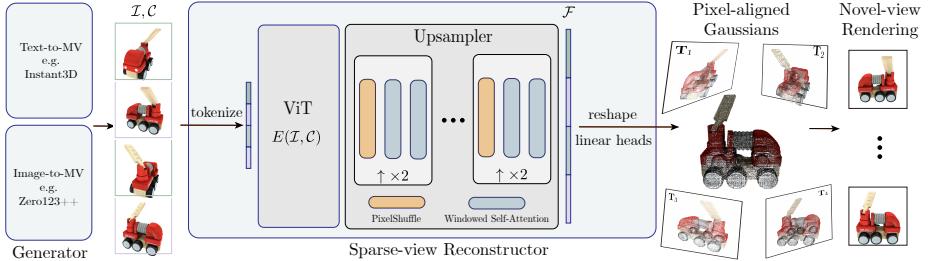


Fig. 2: GRM pipeline. Given 4 input views, which can be generated from text [46] or a single image [79], our sparse-view reconstructor estimates the underlying 3D scene in a single feed-forward pass using pixel-aligned Gaussians. The transformer-based sparse-view reconstructor, equipped with a novel transformer-based upsampler, is capable of leveraging long-range visual cues to efficiently generate a large number of 3D Gaussians for high-fidelity 3D reconstruction.

Transformer-based Encoder. For a given input image $\mathbf{I}_v \in \mathbb{R}^{H \times W \times 3}$, we first inject the camera information to every pixel following [85, 106] with Plücker embedding [34, 85]. Then we use a convolutional image tokenizer with kernel and stride 16 to extract local image features, resulting in a $H/16 \times W/16$ feature map. The features from every view are concatenated together to a single feature vector of length $(V \times H/16 \times W/16)$. Following common practice in vision transformers, we append learnable image position encodings for each token to encode the spatial information in the image space. The resulting feature vector is subsequently fed to a series of self-attention layers. The self-attention layers attend to all tokens across all the input views, ensuring mutual information exchange among all input views, resembling traditional feature matching and encouraging consistent predictions for pixels belonging to different images. The output of the transformer-based encoder is a $V \times H/16 \times W/16$ -long feature vector, denoted as \mathbf{F} . Formally, the transformer function can be written as

$$\mathbf{F} = E_{\theta, \phi}(\mathcal{I}, \mathcal{C}), \quad (2)$$

where θ and ϕ denote the network parameters and the learnable image position encodings.

In the transformer encoder, we utilize patch convolution to tokenize the input images, resulting in the output feature \mathbf{F} with a smaller spatial dimension. While this is advantageous for capturing broader image context, it is limited in modeling high-frequency details. To this end, we introduce a transformer-based upsampler to improve the detail reconstruction.

Transformer-based Upsampler. Inspired by previous work [3, 57], we use windowed attention to balance the need for non-local multi-view information aggregation and feasible computation cost. Specifically, we construct multiple upsampler blocks to progressively upsample the features by factors of 2 until we reach the original input image resolution. In each block, we first quadruple the

feature dimensions with a linear layer and then double the spatial dimension with a PixelShuffle layer [80]. The upsampled feature maps are grouped and passed to a self-attention layer in a sliding window of size W and shift $W/2$. While the self-attention operation is performed within each distinct window, to maintain manageable memory and computation, the overlapping between shifted windows improves non-local information flow. Formally, an upsampler block contains the following operations:

$$\mathbf{F} = \text{PIXELSHUFFLE}(\text{LINEAR}(\mathbf{F}), 2), \quad (3)$$

$$\mathbf{F} = \text{SELFATTENTION}(\mathbf{F}, W), \quad (4)$$

$$\mathbf{F} = \text{SHIFT}(\text{SELFATTENTION}(\text{SHIFT}(\mathbf{F}, W/2), W), -W/2). \quad (5)$$

After several blocks, the context length expands to the same spatial dimension as the input. We reshape the features back to 2D tensors, resulting in V feature maps with a resolution of $H \times W$, denoted as $\mathcal{F} = \{\mathbf{F}_v\}_{v=1}^V$.

Rendering with Gaussian Splatting. From the upsampled features \mathcal{F} , we predict the Gaussian attribute maps $\{\mathbf{T}_v\}_{v=1}^V$ for pixel-aligned Gaussians using separate linear heads. As mentioned in Sec. 3.1, these are then unprojected along the viewing ray according to the predicted depth, from which a final image $\mathbf{I}_{v'}$ and alpha mask $\mathbf{M}_{v'}$ (used for supervision) can be rendered at an arbitrary camera view $\mathbf{c}_{v'}$ through Gaussian splatting.

3.3 Training

During the training phase, we sample $V = 4$ input views that sufficiently cover the whole scene, and supervise with additional views to guide the reconstruction. To remove floaters, we also supervise the alpha map from Gaussian splatting with the ground truth object mask available from the training data.

Given V' supervision views, the training objective is

$$\mathcal{L} = \frac{1}{V'} \sum_{1 \leq v' \leq V'} \mathcal{L}_{\text{img}} + \mathcal{L}_{\text{mask}}, \quad (6)$$

$$\mathcal{L}_{\text{img}} = L_2 \left(\mathbf{I}_{v'}, \hat{\mathbf{I}}_{v'} \right) + 0.5 L_p \left(\mathbf{I}_{v'}, \hat{\mathbf{I}}_{v'} \right), \quad (7)$$

$$\mathcal{L}_{\text{mask}} = L_2 \left(\mathbf{M}_{v'}, \hat{\mathbf{M}}_{v'} \right), \quad (8)$$

where $\hat{\mathbf{I}}_{v'}$ and $\hat{\mathbf{M}}_{v'}$ denote ground truth image and alpha mask, respectively. L_2 and L_p are L2 loss and perceptual loss [35].

To further constrain the Gaussian scaling, we employ the following activation function corresponding to the output \mathbf{s}_o of the linear head for scale. Subsequently, we conduct linear interpolation within predefined scale values s_{\min} and s_{\max} :

$$\mathbf{s} = s_{\min} \text{SIGMOID}(\mathbf{s}_o) + s_{\max} (1 - \text{SIGMOID}(\mathbf{s}_o)). \quad (9)$$

3.4 Reconstructor for 3D Generation

Our reconstructor alone is able to efficiently estimate a 3D scene from 4 input images. We can seamlessly integrate this reconstructor with any diffusion model that generates multi-view images to enable fast text-to-3D and image-to-3D generation, similar to Instant3D [46]. Specifically, we use the first stages of Instant3D [46] and Zero123++ [79] to produce 4 multi-view images from a text prompt or a single image, respectively. Note that Zero123++ generates 6 images, from which we select the 1st, 3rd, 5th, and 6th as input to our reconstructor.

4 Experiments

4.1 Experimental Settings

Training Settings. The encoder E consists of 1 strided convolution layer to tokenize the image and 24 self-attention layers with channel width 768. The upsampler consists of 4 upsampler blocks and each block contains 2 attention layers. For training, we use AdamW [60] with a learning rate initialized at 0.0003 decayed with cosine annealing after $3k$ steps. Deferred back-propagation [110] is adopted to optimize GPU memory. We train our model on 32 NVIDIA A100 GPUs for 40M images on the resolution of 512×512 , using a batch size of 8 per GPU and taking about 4 days to complete. The window size in the transformer-upsampler is 4096. The values for s_{min} and s_{max} are set to 0.005 and 0.02.

Training Data. We obtain multi-view images from Objaverse [17] as training inputs. Objaverse contains more than 800k 3D objects with varied quality. Following [46], we filter 100k high-quality objects, and render 32 images at random viewpoints with a fixed 50° field of view under ambient lighting.

Test Data. We use Google Scanned Objects (GSO) [20], and render a total of 64 test views with equidistant azimuth at {10, 20, 30, 40} degree elevations. In sparse-view reconstruction, the evaluation uses full renderings from 100 objects to assess all models. For single-view reconstruction, we restrict the analysis to renderings generated at an elevation angle of 20 from 250 objects. More details about training settings and data are presented in *Supplementary Material*.

4.2 Sparse-view Reconstruction

Baselines and Metrics. We compare our method with Gaussian Splatting [43], SparseNeuS [54, 59], IBRNet [100], MV-LRM [46], and the concurrent LGM [92]. Since MV-LRM neither released their code nor model, we reproduce it following their paper. The original SparseNeuS does not support 360° reconstruction, so we use the improved version trained in One-2-3-45 [54]. The remaining baselines are evaluated using their original implementations. For all baselines, except for SparseNeuS and IBRNet, we use the same set of four input views that roughly cover the entire object. SparseNeuS and IBRNet are originally intended for



Fig. 3: Sparse-view reconstruction. Given the same sparse-view inputs (top), we compare the 3D reconstruction quality with strong baselines, among which GS [43] is optimization based. SparseNeus [59] (trained in One-2-3-45 [54]) and IBRNet [100] require 16 views (only 4 of those are visualized in the top row). GRM more accurately reconstructs the geometric structure as well as the finer appearance details.

denser input views, hence we additionally sample another 12 views, resulting in 16 roughly uniformly distributed views. Following MV-LRM, we use PSNR, SSIM, and LPIPS to measure the reconstruction quality. Additional evaluations

for geometry reconstruction accuracy is included in the *Supplementary Material*. We conduct all the comparisons using a resolution of 512×512 .

Table 1: Sparse-view reconstruction. We compare the reconstruction quality for 64 novel views with 100 objects from GSO [20]. GRM yields superior quality while maintaining fast speed. INF. Time is time from image inputs to the corresponding 3D representation (e.g., 3D Gaussians or Triplane NeRF); REND. Time is the time used to render a 512^2 image from the 3D representation.

| Method | #views | PSNR \uparrow | SSIM \uparrow | LPIPS \downarrow | INF. Time \downarrow | REND. Time \downarrow |
|---------------------|--------|-----------------|-----------------|--------------------|------------------------|-------------------------|
| GS [43] | 4 | 21.22 | 0.854 | 0.140 | 9 min | Real time |
| IBRNet [100] | 16 | 21.50 | 0.877 | 0.155 | 21 sec | 1.2 sec |
| SparseNeuS [54, 59] | 16 | 22.60 | 0.873 | 0.132 | 6 sec | Real time |
| LGM [92] | 4 | 23.79 | 0.882 | 0.097 | 0.07 sec | Real time |
| MV-LRM [46] | 4 | 25.38 | 0.897 | 0.068 | 0.25 sec | 1.7 sec |
| GRM (Ours) | 4 | 30.05 | 0.906 | 0.052 | 0.11 sec | Real time |

Results. As Tab. 1 shows, our method significantly outperforms all baselines across all metrics by a large margin, even though SparseNeuS and IBRNet require 4 times more input views. At the same time, our inference speed is among the fastest two, only second to the concurrent LGM. However, ours predicts 16 times more Gaussians than they do, which leads to a much higher reconstruction fidelity. Fig. 3 shows the novel-view rendering results. Compared to other methods, our reconstructions accurately reflect the geometric structures, containing no visible floaters, and they capture better appearance details than the baselines.

4.3 Single Image-to-3D Generation

As shown in Sec. 3.4, GRM can be used for single image-to-3D generation by combining it with image-to-multiview diffusion models, such as Zero123++ [79].

Baselines and Metrics. The baselines include SOTA single-image 3D generation methods: ShapE [36], One-2-3-45 [54], One-2-3-45++ [53], DreamGaussian [93], Wonder3D [58], TriplaneGaussian [114], and LGM [92]. For all methods, we use the same input image that is selected randomly from the 4 input views in the sparse-view reconstruction task. All the comparisons are done using a resolution of 256×256 .

Similar to sparse-view reconstruction, we compute PSNR, SSIM and LPIPS. We also include CLIP scores [73] and FID [28], which are two common metrics to evaluate image similarity in generation tasks [54, 55, 106]. Geometry evaluations are included in the *Supplementary Material*.

Results. The quantitative results are presented in Tab. 2. Notably, GRM outperforms all baselines across all metrics. Our model only takes 5 seconds in total to generate 3D Gaussians from the input image, which includes the runtime of

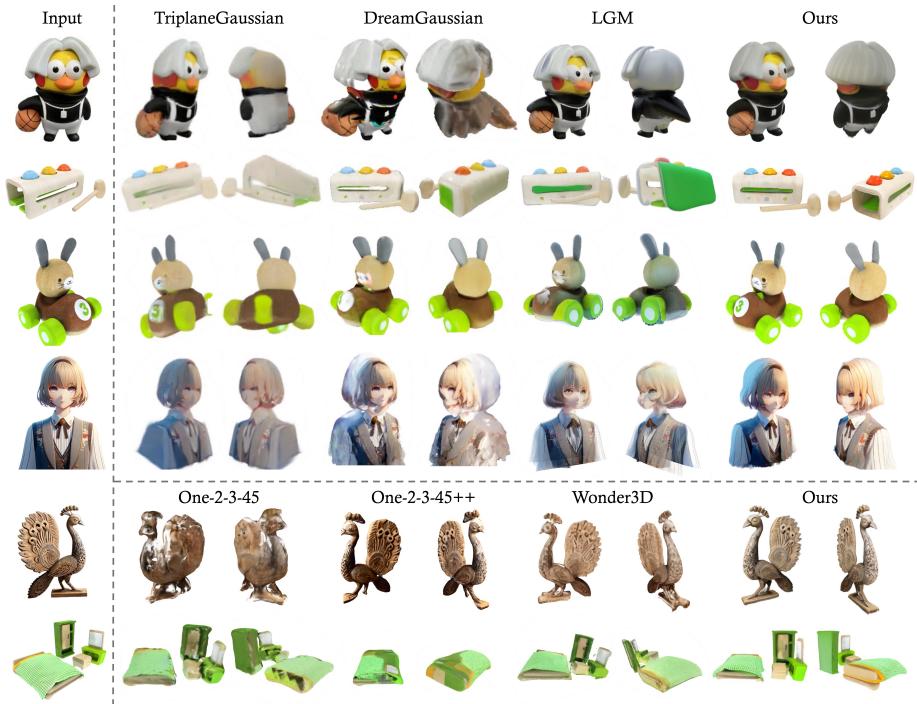


Fig. 4: Single image-to-3D generation. We compare with methods using Gaussians (top) and non-Gaussians (bottom) as 3D representations. Reconstruction methods, e.g., TriplaneGaussian [114] struggle to realistically complete the unseen region (rows 1–2). SDS-based methods, e.g., DreamGaussian [93] suffer from considerable inconsistencies with the input image. LGM [92], One-2-3-45 [54], One-2-3-45++ [53], and Wonder3D [58] also combine multiview diffusion and reconstruction for single image-to-3D generation, but they produce blurrier texture and noticeable geometry artifacts. Our results contain more appearance details and shows significantly better consistency with the input.

Table 2: Single image-to-3D generation. Combined with an image-to-multiview diffusion model [79], GRM can be used for single image-to-3D generation. Our method outperforms relevant baselines in terms of the quality of the synthesized novel views with fast inference speed.

| Method | PSNR↑ | SSIM↑ | LPIPS↓ | CLIP↑ | FID↓ | INF. | Time↓ |
|------------------------|--------------|--------------|--------------|--------------|-------------|---------|-------|
| One-2-3-45 [54] | 17.84 | 0.800 | 0.199 | 0.832 | 89.4 | 45 sec | |
| Shap-E [36] | 15.45 | 0.772 | 0.297 | 0.854 | 56.5 | 9 sec | |
| DreamGaussian [93] | 19.19 | 0.811 | 0.171 | 0.862 | 57.6 | 2 min | |
| Wonder3D [58] | 17.29 | 0.815 | 0.240 | 0.871 | 55.7 | 3 min | |
| One-2-3-45++ [53] | 17.79 | 0.819 | 0.219 | 0.886 | 42.1 | 1 min | |
| TriplaneGaussian [114] | 16.81 | 0.797 | 0.257 | 0.840 | 52.6 | 0.2 sec | |
| LGM [92] | 16.90 | 0.819 | 0.235 | 0.855 | 42.1 | 5 sec | |
| GRM (Ours) | 20.10 | 0.826 | 0.136 | 0.932 | 27.4 | 5 sec | |

the generation head. While this is slower than TriplaneGaussian, we achieve significantly higher reconstruction quality. Our advantage is further demonstrated in the qualitative results shown in Fig. 4. On the top, we compare with other 3D Gaussian-based methods. The pure reconstruction method TriplaneGaussian struggles to fill in the missing content realistically (see rows 1–2). DreamGaussian, using SDS optimization, shows various geometry artifacts (row 1) and overall noticeable inconsistencies with the input image. LGM, also using an image-to-MV generation head, produces blurrier and inconsistent texture and geometry.

The bottom of Fig. 4 shows non-Gaussians based approaches. These methods all display various geometry and appearance artifacts, inconsistent with the input. In contrast, our scalable GRM learns robust data priors from extensive training data, demonstrating strong generalization ability on generated multi-view input images with accurate geometry and sharper details. This leads to fast 3D generation and state-of-the-art single-image 3D reconstruction.

4.4 Text-to-3D Generation

By using a text-to-MV diffusion model, such as the first stage of Instant3D [46], GRM can generate 3D assets from text prompts.

Baselines and metrics. We choose Shap-E [36], Instant3D [46], LGM [92], and MVDream [81] as baselines. MVDream represents the SOTA of optimization-based methods, while others are feed-forward methods. We use the 200 text prompts from DreamFusion [71]. The metrics we use are CLIP Precisions [32, 93], Averaged Precision [106], CLIP Score [46, 106], which measure the alignment between the text and images. All the comparisons are done using a resolution of 512×512 . Additionally, we include a preference study on Amazon Mechanical Turk, where we recruited 90 unique users to compare the generations for 50 text prompts.

Results. As shown in Tab. 3, our method consistently ranks the highest among feed-forward methods (rows 1–3) and compares onpar with optimization-based MVDream. Visually, as shown in Fig. 5, our method excels at generating plausible geometry and highly detailed texture. MVDream, using SDS-based optimization, requires 1 hours to generate a single scene. It delivers impressive visual quality, but exhibits sub-optimal text-image alignment, as indicated by the CLIP score and the ‘a cat wearing eyeglasses’ example in Fig. 4.

4.5 Ablation Study

We analyze our model components and architectural design choices on the training resolution of 256. The results are shown in Tab. 4. Note that all ablations are trained with 16 GPUs for 14M images.



Fig. 5: Text-to-3D Generation. Our method creates high-quality 3D assets from text prompts with accurate text alignment. GRM only requires 8 seconds to generate comparable results to the SOTA optimization-based MVDream, which takes 1 hour.

Table 3: Text-to-3D Generation. Combined with a text-to-multiview diffusion model [46], GRM can be used for text-to-3D generation and it achieves a competitive CLIP score. Our method is most often preferred in our user study while maintains very fast inference speed.

| Method | R-Prec↑ | AP↑ | CLIP↑ | User Pref↑ | INF. | Time |
|------------------|-------------|-------------|-------------|--------------|--------|------|
| Shap-E [36] | 12.7 | 17.7 | 17.3 | 15.7% | 9 sec | |
| LGM [92] | 35.8 | 41.4 | 17.2 | 13.3% | 5 sec | |
| Instant3D [46] | 59.3 | 64.3 | 17.9 | 15.7% | 20 sec | |
| MVDream-SDS [81] | 70.1 | 74.4 | 17.6 | 25.9% | 1 hour | |
| GRM (ours) | 67.5 | 72.0 | 18.5 | 29.5% | 8 sec | |

Scale Activation. We conduct an experiment to study the function of the activation function for the Gaussians’ scales. Conventionally, gaussian splatting [43] takes the scales with an exponential activation. However, the exponential activation can easily result in very large Gaussians, resulting in unstable training and blurry images. With the linear interpolation between the predefined scale ranges, the model achieves better appearance regarding all metrics in the first two rows of Tab. 4.

Number of Upsampler Blocks. We analyze the effects of different number of upsampler blocks. We can see that the model performance increases as the number of upsampler blocks grows from 0 to 3 (Tab. 4, left, rows 2–4), benefiting from the detailed spatial features modulated by our transformer-based upsampler.

Table 4: Ablation. Left: Using the SIGMOID activation improves the visual quality across all metrics; increasing the number of sampling blocks also increases the Gaussians’ density and their modeling capability, as demonstrated by the growing trend of PSNR; finally supervising the alpha channel further boost the reconstruction quality by removing outlier Gaussians. **Right:** We ablate the proposed transformer-based upsampler and pixel-aligned Gaussians using alternative approaches, and demonstrate that each component is critical for the final reconstruction quality.

| Scale | Act | #Up | α -reg | PSNR | SSIM | LPIPS |
|--------------|-----|--------------|---------------|-------|-------|-------|
| \times | 0 | \times | | 24.43 | 0.638 | 0.133 |
| \checkmark | 0 | \times | | 27.51 | 0.900 | 0.044 |
| \checkmark | 1 | \times | | 29.11 | 0.922 | 0.037 |
| \checkmark | 3 | \times | | 29.38 | 0.917 | 0.036 |
| \checkmark | 3 | \checkmark | | 29.48 | 0.920 | 0.031 |

| Method | PSNR | SSIM | LPIPS |
|----------------|-------|-------|-------|
| Conv-Upsampler | 27.23 | 0.894 | 0.063 |
| XYZ prediction | 28.61 | 0.910 | 0.037 |
| Full model | 29.48 | 0.920 | 0.031 |

Alpha Regularization. We ablate the alpha regularization used during training. Without alpha regularization, floaters are observable around the objects. The model can successfully remove those floaters with the help of alpha regularization as shown in Fig. 6, leading to a improvement over all metrics (Tab. 4, left, rows 4–5).

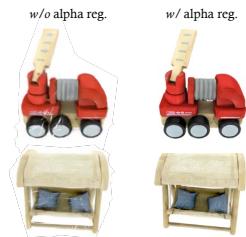


Fig. 6: Comparison on alpha regularization.

Upsampler Architecture. There is an alternative design of the upsampler, which mimics the conventional 2D upsamplers by replacing the transformers with CNNs. We find that CNN-based upsampler leads to worse results (Tab. 4, right). We conjecture that the transformer can capture multi-view correspondences and further enhance the spatial details.

Depth vs. XYZ. In Tab. 4 (right), we conduct an ablation where we predict the 3D coordinates of each Gaussian instead of the depth value. We observe a performance drop across all metrics. Without the constraint of camera rays, the positions of the Gaussians in 3D space become unstructured, making it prone to getting stuck at local minima, resulting in a degraded performance.

5 Discussion

In this paper, we introduce the Gaussian Reconstruction Model (GRM)—a new feed-forward 3D generative model that achieves state-of-the-art quality and speed. At the core of GRM is a sparse-view 3D reconstructor, which leverages a novel transformer-based architecture to reconstruct 3D objects represented by pixel-aligned Gaussians. We plan to release the code and trained models to make this advancement in 3D content creation available to the community.

Limitations and Future Work. The output quality of our sparse-view reconstructor suffers when the input views are inconsistent. The reconstructor

is deterministic in nature and future work could embed it in a probabilistic framework, akin to DMV3D [106]. Our current framework is limited to object-centric scenes due to the lack of large-scale 3D scene datasets. Future work could explore the generation of larger and more complicated scenes.

Ethics. Generative models pose a societal threat—we do not condone using our work to generate deep fakes intending to spread misinformation.

Conclusion. Our work represents a significant advancement towards efficient and high-quality 3D reconstruction and generation.

Acknowledgement. We would like to thank Shangzhan Zhang for his help with the demo video, and Minghua Liu for assisting with the evaluation of One-2-3-45++. This project was supported by Google, Samsung, and a Swiss Postdoc Mobility fellowship.

GRM: Large Gaussian Reconstruction Model for Efficient 3D Reconstruction and Generation

Supplementary Material

Yinghao Xu^{1*}, Zifan Shi^{1,2*}, Yifan Wang¹, Hansheng Chen¹
Ceyuan Yang³, Sida Peng⁴, Yujun Shen⁵, and Gordon Wetzstein¹

¹ Stanford University

² The Hong Kong University of Science and Technology

³ Shanghai AI Laboratory

⁴ Zhejiang University

⁵ Ant Group

This supplementary material is organized as follows. We first introduce implementation details of our GRM (Appendix A). Then, we evaluate the geometry quality of our GRM against the baselines (Appendix B). We also present the details of mesh extraction from 3D Gaussians in Appendix C. Finally, we show additional results on 3D reconstruction and generation to evaluate the flexibility and effectiveness of our approach (Appendix D).

A Implementation Details

Network Architecture and Training Details. We illustrate the details of network architecture and training in Tab. 1.

Deferred Backpropagation. Our model generates $4 \times 512 \times 512$ Gaussians, consuming a significant amount of GPU memory. We would only be able to train our model with a batch size of 2 on 80GB A100 GPUs. Deferred backpropagation [110] is an important technique for saving GPU memory in large batch-size training. With it, we are able to scale up the batch size to 8, consuming only 38GB per GPU. We provide a pseudo code (Algorithm 1) to demonstrate how we implement it in our model training.

Perceptual loss. We have experimented with an alternative loss to the conventional perceptual loss mentioned in the paper, known as the Learned Perceptual Image Patch Similarity (LPIPS) loss [111]. However, we observe severe unstable training and the model cannot converge well.

B Geometry Evaluation

Here, we demonstrate the geometry evaluation results on sparse-view reconstruction and single-image-to-3D generation. We report Chamfer Distance (CD) and

* Equal Contribution

Table 1: Implementation details.

| | | |
|--------------------|---------------------------------|---|
| Encoder | Convolution layer Att layers | 1, kernel size 16, stride 16 24, channel width 768, # heads 12 |
| Upsampler block | Pixelshuffle per block | 1, scale factor 2 |
| | Att layers per block | 2, # heads 12 |
| | Channel width | starting from 768, decay ratio of 2 per block |
| | # Blocks | 4 |
| Gaussian splatting | Color activation | sigmoid |
| | Rotation activation | normalize |
| | Opacity activation | sigmoid |
| | Scale activation | sigmoid |
| | Position activation | None |
| Training details | Learning rate | 3e-4 |
| | Learning rate scheduler | Cosine |
| | Optimizer | AdamW |
| | (Beta1, Beta2) | (0.9, 0.95) |
| | Weight decay | 0.05 |
| | Warm-up | 3000 |
| | Batch size | 8 per GPU |
| | # GPU | 32 |

F-score as the evaluation metrics. Specifically, we use different thresholds for F-score to reduce the evaluation uncertainty. We use ICP alignment to register all the 3D shapes into the same canonical space. All metrics are evaluated on the original scale in the GSO dataset.

Sparse-view Reconstruction. We compare with SparseNeuS [59] which trained in One-2-3-45 [54], and LGM [92] in Tab. 2. The SparseNeuS exhibits a very high CD score with 16 views for reconstruction (32 views in the original paper) because of the far-away floaters. GRM achieves better geometry scores across all metrics, particularly on the F-score with small thresholds.

Table 2: Geometry evaluation on Sparse-view Reconstruction. SparseNeuS [54, 59] exhibits an exceptionally high CD due to the far-away floaters.

| Method | #Views | CD↓ | F-Score(0.01)↑ | F-Score(0.005)↑ |
|---------------------|--------|----------------|----------------|-----------------|
| SparseNeuS [54, 59] | 16 | 0.02300 | 0.3674 | 0.5822 |
| LGM [92] | 4 | 0.00393 | 0.9402 | 0.7694 |
| GRM (Ours) | 4 | 0.00358 | 0.9560 | 0.8239 |

Single-Image-to-3D Generation. We compare GRM against baselines on geometry quality in Tab. 3. The original implementation of One-2-3-45++ [53] suffers from a limitation where it can only generate a single component in multi-object scenes, resulting in geometry metrics that are not as good as other baseline methods. GRM outperforms all baseline methods across all metrics. Moreover,

Algorithm 1 Pseudocode of Deferred Backpropagation on Gaussian Rendering in PyTorch-like style.

Render: Rendering process;

```

class DBGaussianRender(torch.autograd.Function):
    def forward(ctx, gaussians, cameras):
        # save for backpropagation
        ctx.save_for_backward(gaussians, cameras)

        with torch.no_grad():
            images = Render(gaussians, cameras)

        return images

    def backward(ctx, grad_images):
        # restore input tensor
        gaussians, cameras = ctx.saved_tensors

        with torch.enable_grad():
            images = Render(gaussians, cameras)
            images.backward(grad_images)

        return gaussians.grad

```

when compared to optimization-based methods, such as DreamGaussian [93], our approach demonstrates notable runtime advantages along with superior geometry quality.

Table 3: Geometry evaluation on Single-image-to-3D generation. Note that the original implementation of One-2-3-45++ [53] suffers from a limitation where it can only generate a single component in multi-object scenes.

| Method | CD \downarrow | F-Score(0.01) \uparrow | F-Score(0.025) \uparrow |
|--------------------|-----------------|--------------------------|---------------------------|
| One-2-3-45++ [53] | 0.0145 | 0.6419 | 0.8362 |
| Wonder3D [58] | 0.0131 | 0.6384 | 0.8576 |
| One-2-3-45 [54] | 0.0134 | 0.6689 | 0.8682 |
| Shap-E [36] | 0.0118 | 0.6990 | 0.8820 |
| LGM [92] | 0.0123 | 0.6853 | 0.8591 |
| DreamGaussian [93] | 0.0077 | 0.7616 | 0.9506 |
| GRM (Ours) | 0.0058 | 0.8758 | 0.9775 |

C Mesh Extraction from 3D Gaussians

We utilize the Fibonacci sampling method to sample 200 uniformly distributed cameras on sphere for rendering images and depth maps based on the 3D Gaussians of the scene. Subsequently, we fuse the RGB-D data using the TSDFVolume [16] method to generate a mesh. We must take into account that due to the Gaussian distribution, some points may scatter outside the surface of the object. Therefore, we employ clustering to remove very small floaters outside the object’s surface in order to smooth the generated mesh.



Fig. 1: Blender scene constructed with our textured mesh.

D Additional Visual Results

We assemble the extracted texture mesh in Blender to construct a 3D scene. We attach the scene image in Fig. 1. We include more qualitative results on sparse-view reconstruction, text-to-3D generation and image-to-3D generation in Fig. 2, Fig. 3 and Fig. 4, respectively.

E Limitations

Despite the high-quality reconstruction, image-to-3D and text-to-3D generation results we achieved, our model relies on the input information for reconstruction and lacks the capability for hallucination. For example, if a region is not observed in any of the input images, the model may produce blurry textures for it.

References

1. Abdal, R., Yifan, W., Shi, Z., Xu, Y., Po, R., Kuang, Z., Chen, Q., Yeung, D.Y., Wetzstein, G.: Gaussian shell maps for efficient 3d human generation. arXiv preprint arXiv:2311.17857 (2023) [4](#)

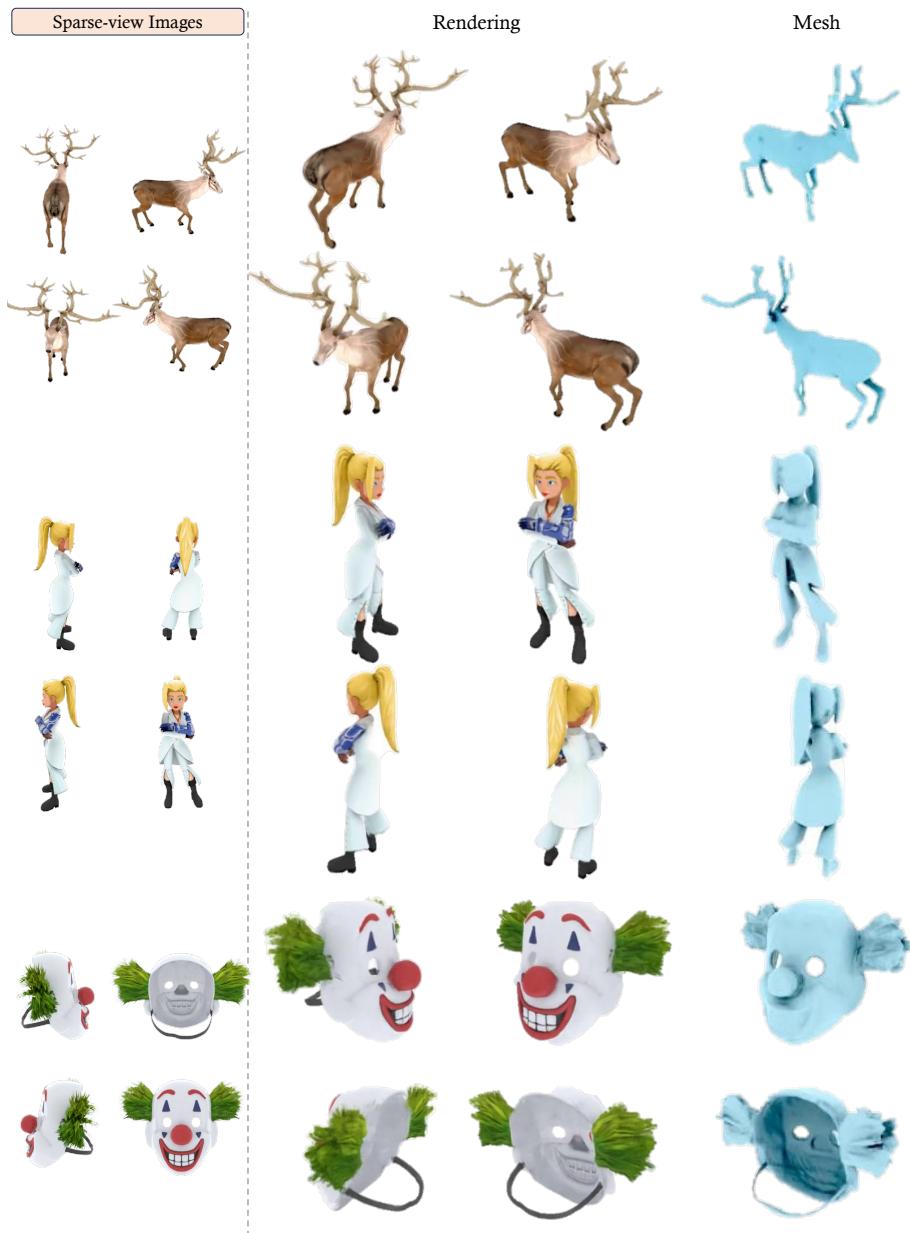


Fig. 2: Sparse-view Reconstruction.

2. Anciukevičius, T., Xu, Z., Fisher, M., Henderson, P., Bilen, H., Mitra, N.J., Guerrero, P.: Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. In: IEEE Conf. Comput. Vis. Pattern Recog. (2023) 1, 4

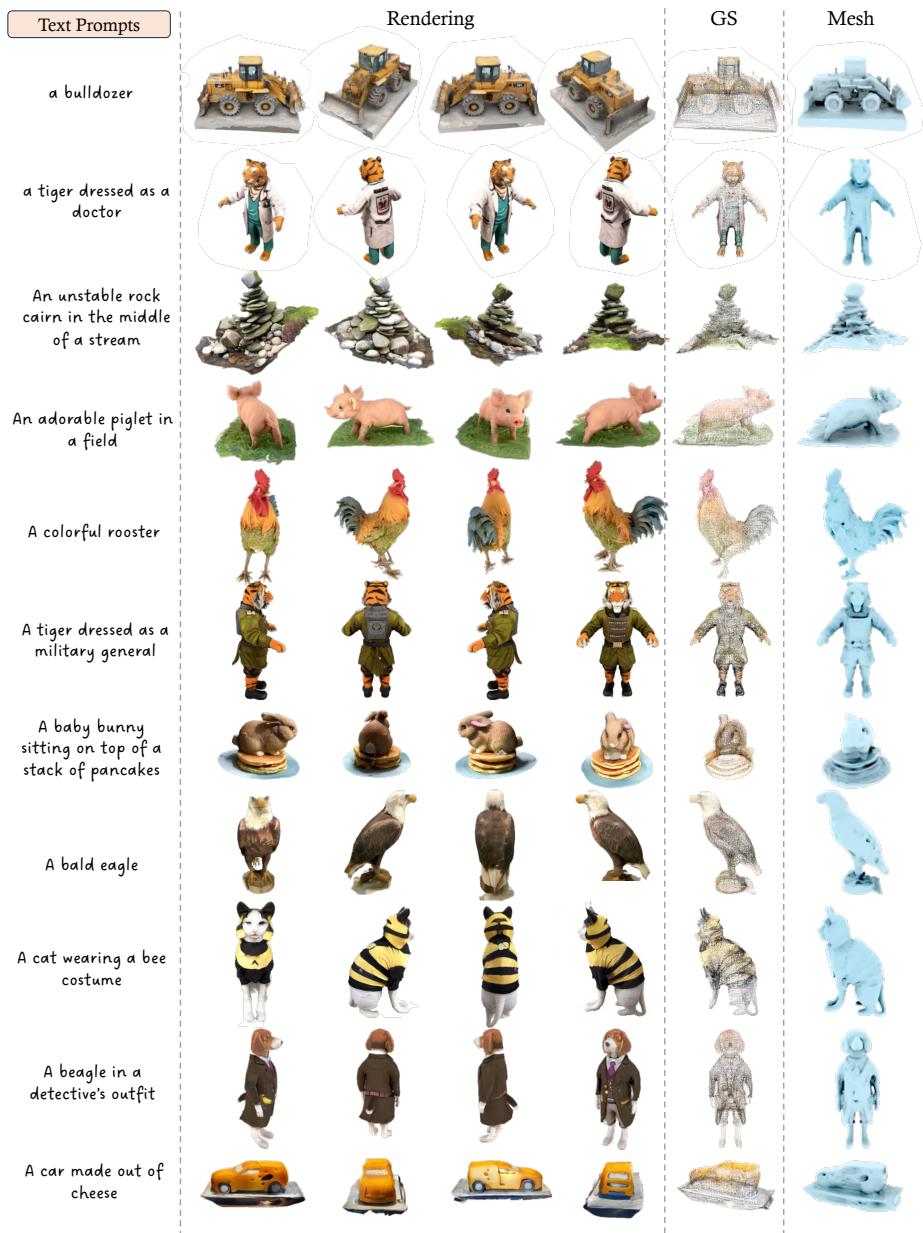


Fig. 3: Text-to-3D Generation.

**Fig. 4: Single-image-to-3D Generation.**

4. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096 (2018) 4

5. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L.J., Tremblay, J., Khamis, S., et al.: Efficient geometry-aware 3d generative adversarial networks. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022) **2**, 4
6. Chan, E.R., Monteiro, M., Kellnhofer, P., Wu, J., Wetzstein, G.: pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021) **4**
7. Chan, E.R., Nagano, K., Chan, M.A., Bergman, A.W., Park, J.J., Levy, A., Aittala, M., De Mello, S., Karras, T., Wetzstein, G.: Generative novel view synthesis with 3d-aware diffusion models. Int. Conf. Comput. Vis. (2023) **4**
8. 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) **3**, 5
9. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: European Conference on Computer Vision (ECCV) (2022) **3**
10. Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., Su, H.: Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In: Int. Conf. Comput. Vis. (2021) **3**
11. Chen, G., Wang, W.: A survey on 3d gaussian splatting. arXiv preprint arXiv:2401.03890 (2024) **4**
12. 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) **1**, 4
13. 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) **4**
14. Chen, Z., Wang, F., Liu, H.: Text-to-3d using gaussian splatting. arXiv preprint arXiv:2309.16585 (2023) **4**
15. Chung, J., Lee, S., Nam, H., Lee, J., Lee, K.M.: Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. arXiv preprint arXiv:2311.13384 (2023) **4**
16. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (1996) **18**
17. 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: IEEE Conf. Comput. Vis. Pattern Recog. pp. 13142–13153 (2023) **8**
18. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021) **4**
19. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) **3**
20. Downs, L., Francis, A., Koenig, N., Kinman, B., Hickman, R., Reymann, K., McHugh, T.B., Vanhoucke, V.: Google scanned objects: A high-quality dataset of 3d scanned household items. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 2553–2560. IEEE (2022) **8**, 10
21. Fei, B., Xu, J., Zhang, R., Zhou, Q., Yang, W., He, Y.: 3d gaussian as a new vision era: A survey. arXiv preprint arXiv:2402.07181 (2024) **4**

22. Gao, J., Shen, T., Wang, Z., Chen, W., Yin, K., Li, D., Litany, O., Gojcic, Z., Fidler, S.: Get3d: A generative model of high quality 3d textured shapes learned from images. *Adv. Neural Inform. Process. Syst.* (2022) **4**
23. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: *Adv. Neural Inform. Process. Syst.* (2014) **4**
24. Gu, J., Liu, L., Wang, P., Theobalt, C.: Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis. arXiv preprint arXiv:2110.08985 (2021) **4**
25. Gu, J., Trevithick, A., Lin, K.E., Susskind, J.M., Theobalt, C., Liu, L., Ramamoorthi, R.: Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. In: *Int. Conf. Mach. Learn.* (2023) **4**
26. Gupta, A., Xiong, W., Nie, Y., Jones, I., Oğuz, B.: 3dgen: Triplane latent diffusion for textured mesh generation. arXiv preprint arXiv:2303.05371 (2023) **4**
27. Hertz, A., Aberman, K., Cohen-Or, D.: Delta denoising score. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 2328–2337 (2023) **4**
28. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* **30** (2017) **10**
29. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *Adv. Neural Inform. Process. Syst.* (2020) **4**
30. 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) **1, 3, 4, 5**
31. Hu, L., Zhang, H., Zhang, Y., Zhou, B., Liu, B., Zhang, S., Nie, L.: Gaussiana-vatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians. arXiv preprint arXiv:2312.02134 (2023) **4**
32. Jain, A., Mildenhall, B., Barron, J.T., Abbeel, P., Poole, B.: Zero-shot text-guided object generation with dream fields. In: *IEEE Conf. Comput. Vis. Pattern Recog.* (2022) **12**
33. Jain, A., Tancik, M., Abbeel, P.: Putting nerf on a diet: Semantically consistent few-shot view synthesis. In: *Int. Conf. Comput. Vis.* (2021) **3**
34. Jia, Y.B.: Plücker coordinates for lines in the space. Problem Solver Techniques for Applied Computer Science, Com-S-477/577 Course Handout (2020) **6**
35. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: *ECCV*. Springer (2016) **7**
36. Jun, H., Nichol, A.: Shap-e: Generating conditional 3d implicit functions. arXiv preprint arXiv:2305.02463 (2023) **4, 10, 11, 12, 13, 18**
37. Kang, M., Zhu, J.Y., Zhang, R., Park, J., Shechtman, E., Paris, S., Park, T.: Scaling up gans for text-to-image synthesis. In: *IEEE Conf. Comput. Vis. Pattern Recog.* (2023) **4**
38. Karnewar, A., Vedaldi, A., Novotny, D., Mitra, N.J.: Holodiffusion: Training a 3d diffusion model using 2d images. In: *IEEE Conf. Comput. Vis. Pattern Recog.* (2023) **1, 4**
39. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: *Int. Conf. Learn. Represent.* (2018) **4**
40. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: *Adv. Neural Inform. Process. Syst.* (2021) **4**

41. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: IEEE Conf. Comput. Vis. Pattern Recog. (2019) **4**
42. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: IEEE Conf. Comput. Vis. Pattern Recog. (2020) **4**
43. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (2023) **4, 5, 8, 9, 10, 13**
44. Keselman, L., Hebert, M.: Approximate differentiable rendering with algebraic surfaces. In: European Conference on Computer Vision. pp. 596–614. Springer (2022) **4**
45. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023) **2**
46. 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. <https://arxiv.org/abs/2311.06214> (2023) **1, 2, 3, 4, 5, 6, 8, 10, 12, 13**
47. Li, X., Wang, H., Tseng, K.K.: Gaussiandiffusion: 3d gaussian splatting for denoising diffusion probabilistic models with structured noise. arXiv preprint arXiv:2311.11221 (2023) **4**
48. Li, Z., Zheng, Z., Wang, L., Liu, Y.: Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. arXiv preprint arXiv:2311.16096 (2023) **4**
49. Liang, Y., Yang, X., Lin, J., Li, H., Xu, X., Chen, Y.: Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. arXiv preprint arXiv:2311.11284 (2023) **4**
50. 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: IEEE Conf. Comput. Vis. Pattern Recog. pp. 300–309 (2023) **1, 4**
51. Lin, K.E., Yen-Chen, L., Lai, W.S., Lin, T.Y., Shih, Y.C., Ramamoorthi, R.: Vision transformer for nerf-based view synthesis from a single input image. In: IEEE Winter Conf. Appl. Comput. Vis. (2023) **3**
52. 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**
53. Liu, M., Shi, R., Chen, L., Zhang, Z., Xu, C., Wei, X., Chen, H., Zeng, C., Gu, J., Su, H.: One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. arXiv preprint arXiv:2311.07885 (2023) **10, 11, 17, 18**
54. Liu, M., Xu, C., Jin, H., Chen, L., T, M.V., Xu, Z., Su, H.: One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization (2023) **1, 4, 8, 9, 10, 11, 17, 18**
55. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9298–9309 (2023) **4, 10**
56. Liu, Y., Lin, C., Zeng, Z., Long, X., Liu, L., Komura, T., Wang, W.: Syncdreamer: Generating multiview-consistent images from a single-view image. In: The Twelfth International Conference on Learning Representations (2023) **4**

57. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021) 6
58. 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) 4, 10, 11, 18
59. Long, X., Lin, C., Wang, P., Komura, T., Wang, W.: Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In: Eur. Conf. Comput. Vis. (2022) 3, 8, 9, 10, 17
60. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 8
61. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. arXiv preprint arXiv:2308.09713 (2023) 4
62. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: IEEE Conf. Comput. Vis. Pattern Recog. (2019) 3
63. 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: Eur. Conf. Comput. Vis. (2020) 3
64. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4), 102:1–102:15 (Jul 2022). <https://doi.org/10.1145/3528223.3530127>, <https://doi.org/10.1145/3528223.3530127> 3
65. Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., Yang, Y.L.: Hologan: Unsupervised learning of 3d representations from natural images. In: Int. Conf. Comput. Vis. (2019) 4
66. 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
67. Niemeyer, M., Geiger, A.: Giraffe: Representing scenes as compositional generative neural feature fields. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021) 4
68. Ntavelis, E., Siarohin, A., Olszewski, K., Wang, C., Van Gool, L., Tulyakov, S.: Autodecoding latent 3d diffusion models. arXiv preprint arXiv:2307.05445 (2023) 4
69. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deep sdf: Learning continuous signed distance functions for shape representation. In: IEEE Conf. Comput. Vis. Pattern Recog. (2019) 3
70. Po, R., Yifan, W., Golyanik, V., Aberman, K., Barron, J.T., Bermano, A.H., Chan, E.R., Dekel, T., Holynski, A., Kanazawa, A., et al.: State of the art on diffusion models for visual computing. arXiv preprint arXiv:2310.07204 (2023) 1, 4
71. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. In: The Eleventh International Conference on Learning Representations (2022) 1, 4, 12
72. Qian, S., Kirschstein, T., Schoneveld, L., Davoli, D., Giebenhain, S., Nießner, M.: Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians. arXiv preprint arXiv:2312.02069 (2023) 4

73. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021) [10](#)
74. 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](#)
75. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022) [4](#)
76. Saito, S., Schwartz, G., Simon, T., Li, J., Nam, G.: Relightable gaussian codec avatars. arXiv preprint arXiv:2312.03704 (2023) [4](#)
77. Schwarz, K., Liao, Y., Niemeyer, M., Geiger, A.: Graf: Generative radiance fields for 3d-aware image synthesis. In: Adv. Neural Inform. Process. Syst. (2020) [4](#)
78. Shen, B., Yan, X., Qi, C.R., Najibi, M., Deng, B., Guibas, L., Zhou, Y., Anguelov, D.: Gina-3d: Learning to generate implicit neural assets in the wild. In: IEEE Conf. Comput. Vis. Pattern Recog. pp. 4913–4926 (2023) [1, 4](#)
79. 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. arXiv preprint arXiv:2310.15110 (2023) [2, 4, 5, 6, 8, 10, 11](#)
80. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1874–1883 (2016) [7](#)
81. Shi, Y., Wang, P., Ye, J., Mai, L., Li, K., Yang, X.: Mvdream: Multi-view diffusion for 3d generation. In: The Twelfth International Conference on Learning Representations (2023) [4, 12, 13](#)
82. Shi, Z., Peng, S., Xu, Y., Andreas, G., Liao, Y., Shen, Y.: Deep generative models on 3d representations: A survey. arXiv preprint arXiv:2210.15663 (2022) [4](#)
83. Shue, J.R., Chan, E.R., Po, R., Ankner, Z., Wu, J., Wetzstein, G.: 3d neural field generation using triplane diffusion. In: IEEE Conf. Comput. Vis. Pattern Recog. (2023) [4](#)
84. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. Advances in neural information processing systems **33**, 7462–7473 (2020) [3](#)
85. Sitzmann, V., Rezhikov, S., Freeman, B., Tenenbaum, J., Durand, F.: Light field networks: Neural scene representations with single-evaluation rendering. NeurIPS (2021) [6](#)
86. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. Advances in Neural Information Processing Systems **32** (2019) [3](#)
87. Skorokhodov, I., Siarohin, A., Xu, Y., Ren, J., Lee, H.Y., Wonka, P., Tulyakov, S.: 3d generation on imagenet. In: International Conference on Learning Representations (2023), <https://openreview.net/forum?id=U2WjB9xxZ9q> [4](#)
88. Skorokhodov, I., Tulyakov, S., Wang, Y., Wonka, P.: Epigraf: Rethinking training of 3d gans. In: Adv. Neural Inform. Process. Syst. (2022) [4](#)
89. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020) [4](#)
90. Szymanowicz, S., Rupprecht, C., Vedaldi, A.: Splatter image: Ultra-fast single-view 3d reconstruction. arXiv preprint arXiv:2312.13150 (2023) [3, 4, 5](#)

91. Szymanowicz, S., Rupprecht, C., Vedaldi, A.: Viewset diffusion:(0-) image-conditioned 3d generative models from 2d data. arXiv preprint arXiv:2306.07881 (2023) 1
92. Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., Liu, Z.: Lgm: Large multi-view gaussian model for high-resolution 3d content creation. arXiv preprint arXiv:2402.05054 (2024) 4, 8, 10, 11, 12, 13, 17, 18
93. 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) 1, 4, 10, 11, 12, 18
94. Tang, J., Wang, T., Zhang, B., Zhang, T., Yi, R., Ma, L., Chen, D.: Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. arXiv preprint arXiv:2303.14184 (2023) 4
95. Tewari, A., Thies, J., Mildenhall, B., Srinivasan, P., Tretschk, E., Yifan, W., Lassner, C., Sitzmann, V., Martin-Brualla, R., Lombardi, S., et al.: Advances in neural rendering. In: Computer Graphics Forum. pp. 703–735 (2022) 3
96. Tewari, A., Yin, T., Cazenavette, G., Rezhnikov, S., Tenenbaum, J., Durand, F., Freeman, B., Sitzmann, V.: Diffusion with forward models: Solving stochastic inverse problems without direct supervision. Advances in Neural Information Processing Systems **36** (2024) 4
97. Tosi, F., Zhang, Y., Gong, Z., Sandström, E., Mattoccia, S., Oswald, M.R., Poggi, M.: How nerfs and 3d gaussian splatting are reshaping slam: a survey. arXiv preprint arXiv:2402.13255 (2024) 4
98. Wang, H., Du, X., Li, J., Yeh, R.A., Shakhnarovich, G.: Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12619–12629 (2023) 1, 4
99. 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) 3, 5
100. Wang, Q., Wang, Z., Genova, K., Srinivasan, P.P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: Ibrnet: Learning multi-view image-based rendering. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021) 3, 8, 9, 10
101. 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) 1, 4
102. Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., Wang, X.: 4d gaussian splatting for real-time dynamic scene rendering. arXiv preprint arXiv:2310.08528 (2023) 4
103. 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
104. Xu, Y., Chai, M., Shi, Z., Peng, S., Skorokhodov, I., Siarohin, A., Yang, C., Shen, Y., Lee, H.Y., Zhou, B., et al.: Discoscene: Spatially disentangled generative radiance fields for controllable 3d-aware scene synthesis. In: IEEE Conf. Comput. Vis. Pattern Recog. (2023) 4
105. Xu, Y., Peng, S., Yang, C., Shen, Y., Zhou, B.: 3d-aware image synthesis via learning structural and textural representations. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022) 4
106. 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) 1, 3, 4, 5, 6, 10, 12, 15
107. Yang, Z., Yang, H., Pan, Z., Zhu, X., Zhang, L.: Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. arXiv preprint arXiv:2310.10642 (2023) 4
108. Yang, Z., Gao, X., Zhou, W., Jiao, S., Zhang, Y., Jin, X.: Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. arXiv preprint arXiv:2309.13101 (2023) 4
109. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021) 3
110. Zhang, K., Kolkin, N., Bi, S., Luan, F., Xu, Z., Shechtman, E., Snavely, N.: Arf: Artistic radiance fields (2022) 8, 16
111. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: IEEE Conf. Comput. Vis. Pattern Recog. (2018) 16
112. Zhu, J., Yang, C., Zheng, K., Xu, Y., Shi, Z., Shen, Y.: Exploring sparse moe in gans for text-conditioned image synthesis. arXiv preprint arXiv:2309.03904 (2023) 4
113. Zielonka, W., Bagautdinov, T., Saito, S., Zollhöfer, M., Thies, J., Romero, J.: Drivable 3d gaussian avatars. arXiv preprint arXiv:2311.08581 (2023) 4
114. 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) 3, 10, 11