NovelGS: Consistent Novel-view Denoising via Large Gaussian Reconstruction Model

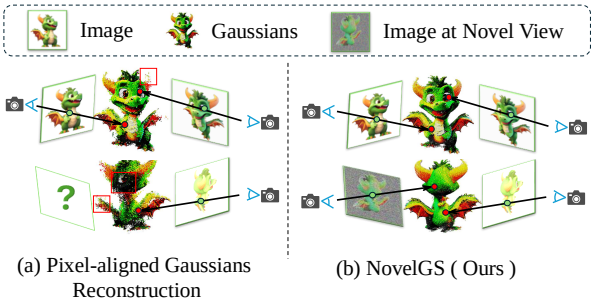
Jinpeng Liu[1], Jiale Xu[2], Weihao Cheng[2], Yiming Gao[2], Xintao Wang[2], Ying Shan[2], Yansong Tang[*,1]
[1]*Tsinghua University,* [2]*Tencent ARC Lab*

Abstract

We introduce NovelGS, a diffusion model for Gaussian Splatting (GS) given sparse-view images. Recent works leverage feed-forward networks to generate pixel-aligned Gaussians, which could be fast rendered. Unfortunately, the method was unable to produce satisfactory results for areas not covered by the input images due to the formulation of these methods. In contrast, we leverage the novel view denoising through a transformer-based network to generate 3D Gaussians. Specifically, by incorporating both conditional views and noisy target views, the network predicts pixel-aligned Gaussians for each view. During training, the rendered target and some additional views of the Gaussians are supervised. During inference, the target views are iteratively rendered and denoised from pure noise. Our approach demonstrates state-of-the-art performance in addressing the multi-view image reconstruction challenge. Due to generative modeling of unseen regions, NovelGS effectively reconstructs 3D objects with consistent and sharp textures. Experimental results on publicly available datasets indicate that NovelGS substantially surpasses existing image-to-3D frameworks, both qualitatively and quantitatively. We also demonstrate the potential of NovelGS in generative tasks, such as text-to-3D and image-to-3D, by integrating it with existing multiview diffusion models. We will make the code publicly accessible.

1. Introduction

The automation of 3D content creation holds substantial promise across various domains such as digital gaming, virtual reality, and cinematic production. Core methodologies, including image-to-3D and text-to-3D, offer considerable advantages by substantially reducing the dependency on manual labor by professional 3D artists. Some work [7, 19, 23, 30, 41, 45, 50, 54] generate 3D assets by iteratively distilling image generative models. However, methods based on Score Distillation Sampling (SDS) necessitate prolonged optimization periods per asset, often extending to several hours. Due to the limited understanding of 3D concepts in 2D diffusion models, maintaining 3D consistency is challenging. As a result, these methods are prone to producing geometric artifacts, such as the multi-faced Janus [58] and issues related to content drift.



(a) Pixel-aligned Gaussians Reconstruction (b) NovelGS (Ours)

Figure 1. **Comparison of pixel-aligned Gaussians reconstruction models and NovelGS.** (a) Most existing models [44, 57, 62] translate the input pixels into pixel-aligned Gaussians [57] based on camera rays. (b) Conversely, we propose to denoise novel view images via the large Gaussian reconstruction model where the unseen parts of the objects could be reconstructed consistently.

With the advent of large 3D datasets [8, 9] and implicit 3D representations [3, 24], some studies [14, 18, 51, 62] propose utilizing transformer-based models to map images into triplane features in a feed-forward manner. They then render novel views using volume rendering techniques [24]. While these methods are flexible, they result in dense computations during rendering, which can be time-consuming. For instance, rendering a 2-second (60 frames) video takes approximately 1.5 minutes on a single NVIDIA A100 GPU. To enhance user-friendliness, some studies [52, 55] propose combining similar frameworks with the Marching Cubes algorithm [21, 34] to generate 3D meshes directly. However, this approach is challenging and unstable during training, and the rendering quality is suboptimal.

3D Gaussians [16] features fast rendering speeds with explicit representation. As shown in Figure 1 (a), some studies [44, 57, 59, 60, 62] utilize stacks of transformer or U-Net models to map images to pixel-aligned Gaussians. However, they tend to poorly generalize to novel views that are not covered by input views. Because they correspond the pixel points of the image to spatial locations based on

Figure 2. **High-fidelity 3D assets** produced by **NovelGS**. It's designed for sparse-view reconstruction and operates in conjunction with various complementary tools, including text-to-image generation [32], and image-to-multiview modeling [35]. This collaborative framework facilitates the generation of text-to-3D (bottom) and image-to-3D (center), as well as the reconstruction of real-world objects (top).

the camera's perspective, the results tend to be poor and inconsistent for areas not illuminated by the camera.

In this paper, we propose NovelGS, a 3D Gaussian diffusion model conditioned on a few input images. NovelGS utilizes a transformer-based denoising network, which is fed with not only condition views but also a number of noisy views as shown in Figure 1 (b). These target views are preset for unseen regions, to generate parts not covered by condition views. The network then predicts pixel-aligned 3D Gaussians for all these views. During training, we expect that clean and noisy views are rendered from the predicted Gaussians and supervise them with $L2$ and $LPIPS$ loss. During inference, we initialize target views with pure noise and step-by-step denoise them by the network, and we ob-

tain the final Gaussians from the last denoising step. Specifically, we introduce the denoise of the novel view in the reconstruction process to ensure the consistent visual effect of the invisible part (see Sec. 4.3). At the same time, our model structure is flexible and can accept various combinations of different numbers and positions of noisy views and clean views befitting the application scenarios. The model is conditioned on the diffusion time step, allowing it to manage varying noise levels throughout the diffusion process.

We trained NovelGS on multi-view images of Objaverse [8] and evaluated the performance on the Google Scanned Objects [10] and OmniObject3D [53]. By integrating novel-view denoising, our model not only outperforms existing methods with the same input views but also makes it possible to handle unbalanced input images, which couldn't cover enough parts of the objects. When paired with text-to-image [33] and image-to-multi-view image models [35], NovelGS achieves outperforming quality for text and single image-to-3D object generation. Experimental results demonstrate the state-of-the-art performance of our method in sparse-view reconstruction benchmarks.

2. Related Work

2.1. Reconstruction Models

Reconstructing 3D from multi-view images is a long-standing problem in computer vision. Traditional methods rely on fitting, which usually requires a dense set of images, such as NeRF [24] and Gaussian Splatting [16]. Learning-based methods use neural networks to predict 3d representations from sparse images, e.g., MVSNeRF, PixelNeRF, NerFormer, SRT, MCC [18, 46, 48, 51, 52, 55]. Among these methods, large reconstruction models (LRMs) [14] demonstrate strong generalization ability on open-world images. By training on large-scale datasets [8, 9], LRMs effectively maps a single image to triplanes [3] via a transformer-based network. Instant3D extends LRMs to a text-to-3d method. It first uses diffusion model to generate multi-view images from text, and then uses LRM to predict triplanes from the images. As an implicit representation, Triplanes are not only effective for novel-view synthesis but also can be extracted into high-quality mesh [52, 55]. Some work [44, 57, 62] explores Gaussians [16] as the 3D representation. LGM [44] and GRM [57] utilizes an asymmetric U-Net and a transformer network to predict and fuse 3D Gaussians, respectively. GeoLRM [60] proposes to utilize occupancy grid prediction to predict geometry-aware objects. GS-LRM [62] validates the feasibility of the paradigm in a large-scale scene dataset. Compared with these methods, our NovelGS utilizes the transformer-based novel-view diffusion model to denoise noisy novel-view images utilizing conditional information from known images, explicitly exploring unseen parts of the 3D object.

2.2. 3D Generation

The field of generative models has experienced significant advancements, particularly with the development of Generative Adversarial Networks (GANs) [12] and Diffusion Models [13, 20, 40], which have demonstrated substantial efficacy in image and video generation [11, 32, 38]. In the context of 3D generation, 3D GANs are utilized to generate 3D-aware asserts [2, 25, 27, 39, 56] in early time, while they are hard to train, leading to limited performance. Although some works utilize 3D diffusion models [13, 15, 26, 28, 36] to replace 3D GANs with direct 3D supervision for 3D assert generation, the quality and diversity of their results are significantly lower compared to the performance of DMs in 2D space. This discrepancy is partly due to the computational challenges of scaling diffusion network models from 2D to 3D and the limited availability of 3D training data [4] previously. DMV3D [58] utilizes multi-view diffusion to denoise images, while it's hard to extend to the scene and NeRF [24] is time-consuming for rendering. Some studies [37] utilize an autoregressive model [31] to generate meshes directly. While mesh representation is challenging to encode and not GPU-friendly, this leads to instability during the training stage and suboptimal rendering quality. In contrast, NovelGS employs an efficient Gaussian representation and novel view denoising, resulting in improved efficiency and stability for both training and inference.

3. Method

In this section, we introduce our NovelGS model, which is designed to reconstruct high-quality 3D assets from sparse-view images. Our approach leverages a diffusion framework that effectively denoises images from noisy views through 3D Gaussian reconstruction and rendering, facilitating consistent 3D generation (see Section 3.1). Additionally, we propose a transformer-based denoiser for generating 3D Gaussians [44, 57, 62], which conditions on both the timestep and clean images. This enables precise and controllable 3D reconstruction (see Section 3.2). The final output of the denoising process is a set of 3D Gaussians, culminating in the generated 3D model. The loss functions employed in our model are detailed in Section 3.3.

3.1. Model Architecture

The pipeline of our model is shown in the Figure 3. During the training phase, our method utilizes a set of images $\{I^i\}_{i=1}^{m+n}$ along with their corresponding camera ray embeddings $\{R^i\}_{i=1}^{m+n}$ as input, where m and n represent the number of clean and noisy images, respectively. We add different levels of noise to noisy view images $\{I^i\}_{i=m+1}^{m+n}$ based on the timestep T. Moreover, a transformer-based denoiser predicts 3D Gaussians G. Finally, we render several images from the 3D Gaussians and supervise the model by

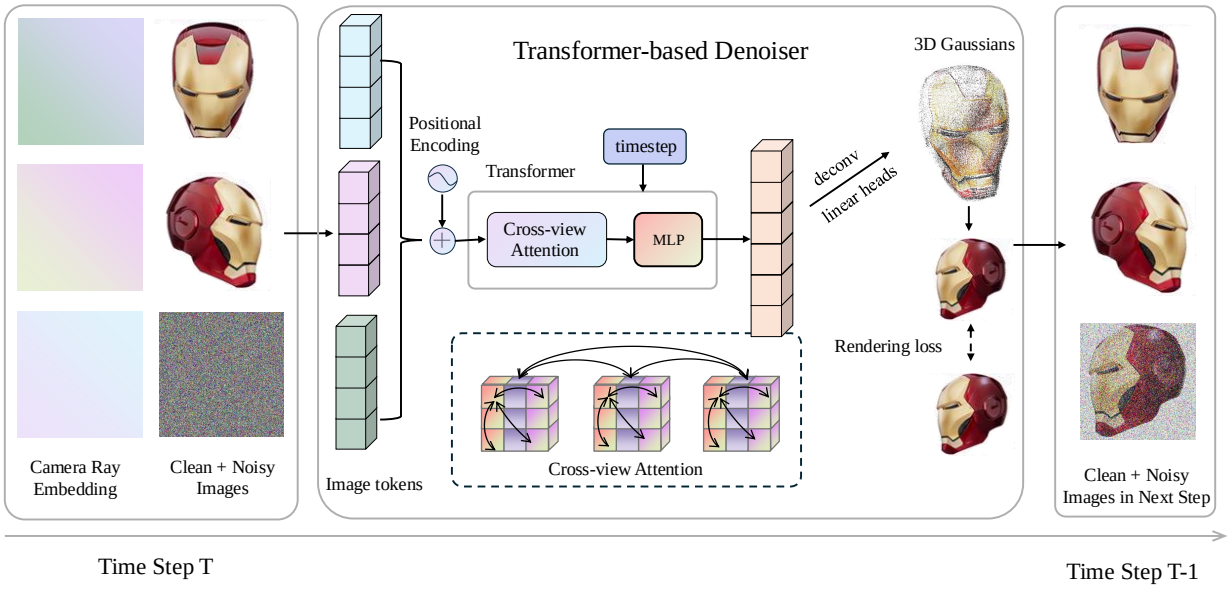


Figure 3. **Pipeline of NovelGS model.** We utilize a large transformer-based network to denoise noisy view images for 3D reconstruction. **During inference**, we initialize target views with pure noise. Then we concatenate the camera ray embedding (Plücker rays) and images (two clean views and one noisy view in the figure to reduce clutterness; four clean views and one noisy view in main experiments) as the input. Then we utilize the denoiser to predict the Gaussians and render the image from the noisy view. After that, we add noise to the noisy view images to timestep T-1. We loop this process until we get the final 3D Gaussians. **During training**, we add noise to the noisy view images based on the timestep and utilize the denoiser to predict 3D Gaussians. We train the denoiser module by rendering loss.

rendering loss. In the inference stage, we initialize the noisy view image $\{I^i\}_{i=m+1}^{m+n}$ with pure noise and concatenate it with clean view images $\{I^i\}_{i=1}^{m}$. Then we concatenate the set of images with their camera ray embeddings as the input of denoiser. Moreover, the denoiser outputs 3D Gaussians, and we render the Gaussians in noisy views. After that, we add noise to the noisy view images to timestep $T-1$ and replace the noisy view images at timestep T. Finally, they will serve as the input for the next diffusion sampling step until we get the final 3D Gaussians at timestep 0.

Gaussians and Camera Embedding. Gaussian splatting [16] represents 3D scene with a set of 3D gaussians, which are efficient for rendering. Specifically, each Gaussian is defined by a center $\mathbf{x} \in R^3$, a scaling factor $\mathbf{s} \in R^3$, and a rotation quaternion $\mathbf{q} \in R^4$. Additionally, an opacity value $\alpha \in R$ and spherical harmonics (SH) coefficients $\mathbf{c} \in R^D$, with D denoting the number of SH bases, are maintained for rendering. These parameters can be collectively denoted by Θ, with $\Theta = \{x_i, s_i, q_i, \alpha_i, c_i\}$ representing the parameters for the i-th Gaussian. Following previous methods [5, 42, 44, 57], we use the Plücker ray embedding to encode the camera poses to get camera embedding:

$$f_i = \{o_i \times d_i, d_i\} \tag{1}$$

where d_i is the ray direction, and o_i is the ray origin. Each pixel of the output feature map is treated as a 3D Gaussian inspired by splatter image [43]. Consequently, for each input view the model predicts a Gaussian attribute map $H \in R^{H \times W \times C}$ of C channels, corresponding to depth, rotation, scaling, opacity, and the DC term of the SH coefficients. Then $m+n$ views of Gaussian attribute are contacted together, generating a total of $(m+n) * H * W$ 3D Gaussians. Finally, we could render different images from any viewpoint with these 3D Gaussians G.

Input Posed Image Tokenization. NovelGS employs a streamlined tokenizer for posed images, drawing inspiration from the Vision Transformer [52] and MeshLRM [52]. Specifically, we concatenate the camera ray embedding with the RGB pixel values, resulting in a 9-channel feature map. This feature map is then divided into non-overlapping patches, which are linearly transformed to serve as input for the transformer. Although the Plücker coordinates inherently encode spatial information, we add additional positional embeddings following VIT which is different from MeshLRM. Because we want our model to be more sensitive to the position of the novel view. It is noteworthy that our image tokenizer is considerably simpler than those used in previous large reconstruction models (LRMs), which often rely on a pre-trained DINO ViT [1] for image encoding. Because DINO is primarily optimized for intra-view semantic reasoning, whereas 3D reconstruction predominantly requires inter-view low-level correspondences [52].

Transformer-based Denoiser. We concatenate multi-view image tokens with learnable triplane (positional) em-

beddings and input them into a sequence of transformer blocks [47]. Each block is composed of cross-view self-attention and multilayer perceptron (MLP) layers, with layer normalization applied before both layers and residual connections are incorporated. This deep transformer network facilitates extensive information exchange among all tokens, effectively modeling intra-view and inter-view relationships. The noisy image tokens, now contextualized by all condition views, are subsequently decoded into clean 3D tokens. Then we utilize transposed convolution to upsample the features. From the upsampled features F, we predict the Gaussian attribute maps for pixel-aligned Gaussians using separate linear heads. These attribute maps are subsequently unprojected along the viewing ray based on the predicted depth. This process allows for the rendering of a final image I^i, and an alpha mask M^i (used for supervision) at an arbitrary camera view through Gaussian splatting.

3.2. Time Step and Image Condition

Time Step Condition. Inspired by DiT [18, 29], we employ the *adaLN-Zero* module to incorporate the timestep condition. In each cross-view attention module, the timestep is injected to handle inputs with varying noise levels.

Image Condition. To enhance the adaptability of our model, we adopt an approach where the initial m views $\{I^1, I^2, ..., I^m\}$ in the denoiser input are kept free of noise to serve as conditioning images. Meanwhile, diffusion and denoising processes are applied to the remaining n views. This strategy enables the denoiser to effectively reconstruct missing pixels in the noisy, unseen views by leveraging information from the input views, analogous to the image inpainting task, which has been demonstrated to be feasible with 2D denoising models [32]. Moreover, to improve the generalizability of our image-conditioned model, we generate 3D Gaussians within a coordinate frame aligned with the conditioning views and render additional images using poses relative to these conditioning views. Specifically, we normalize all camera positions together so that the position of the first condition image view resides at $(0, y, 0)$.

3.3. Loss Function

During the training stage, we render images from random T supervision views using the predicted 3D Gaussians and minimize the image reconstruction loss and mask loss. Furthermore, we utilize perceptual image patch similarity loss [63] to make the training stage more stable. $\{I_i|i = 1, 2, ..., H\}$ represent the ground-truth views, and $\{\hat{I}_i|i = 1, 2, ..., H\}$ represent the predict views rendered by the predict Gaussian splats. $\{M_i|i = 1, 2, ..., H\}$ represent the ground-truth mask, and $\{\hat{M}_i|i = 1, 2, ..., H\}$ represent the predicted mask rendered by the predicted Gaussian splats. So our loss function is :

$$\mathcal{L} = \frac{1}{T}\sum_{i=1}^{T}(\mathcal{L}_{img}(I_i, \hat{I}_i) + \mathcal{L}_{mask}(M_i, \hat{M}_i)), \quad (2)$$

$$\mathcal{L}_{img}(I_i, \hat{I}_i) = ||I_i - \hat{I}_i||_2 + \lambda \cdot \mathcal{L}_{LPIPS}(I_i, \hat{I}_i), \quad (3)$$

$$\mathcal{L}_{mask}(M_i, \hat{M}_i) = ||M_i - \hat{M}_i||_2, \quad (4)$$

where $\mathcal{L}_{LPIPS}$ represent the perceptual image patch similarity loss, λ is the weight of it . Note that H is larger than $(m+n)$ because our model could supervise more views than input views for better performance.

4. Experiments

4.1. Implementation Details

Training Data. Our training dataset is composed of multi-view images rendered from the Objaverse [8] dataset. For each object in the dataset, we render 512 × 512 images from 32 random viewpoints. To ensure high-quality training data, we applied a thorough filtering process to curate a subset of objects that meet specific criteria (See Supplementary). By applying these filtering criteria, we curated a high-quality subset consisting of approximately 270,000 instances from the initial pool of 800,000 objects in the Objaverse dataset. This rigorous selection process ensures that our model is trained on data that is both diverse and representative of high-quality 3D objects, thereby enhancing the robustness and accuracy of the generated 3D reconstructions.

Evaluation Data. We utilize two public datasets following InstantMesh [55]: Google Scanned Objects (GSO) [10] and OmniObject3D (Omni3D) [53]. To evaluate the visual quality of the generated 3D asserts, we created the image evaluation sets for both GSO and Omni3D datasets. For the GSO dataset, which comprises approximately 1,000 objects, we randomly selected 300 objects to constitute the evaluation set. For the Omni3D dataset, we chose 28 common categories and then selected the first 5 objects from each category (totaling 130 objects, as some categories contain fewer than 5 objects) as the evaluation set. For each object, we rendered 21 images along an orbiting trajectory with uniform azimuths and varying elevations of {30°, 0°, -30°}. This systematic evaluation approach allows us to assess the visual fidelity and quality of the 3D Gaussians generated by NovelGS. By leveraging multiple views and varying angles, we ensure a comprehensive evaluation that captures the nuanced details of the reconstructed objects.

Training Settings. The training process is composed of two stages. In the first stage, we pre-train the model with a resolution of 256 × 256 and a batch size of 6 in each GPU for several epochs. We utilize the AdamW optimizer [22] with an initial learning rate of 4e-4, which is decayed via cosine annealing after 3000 steps. In the second stage, we

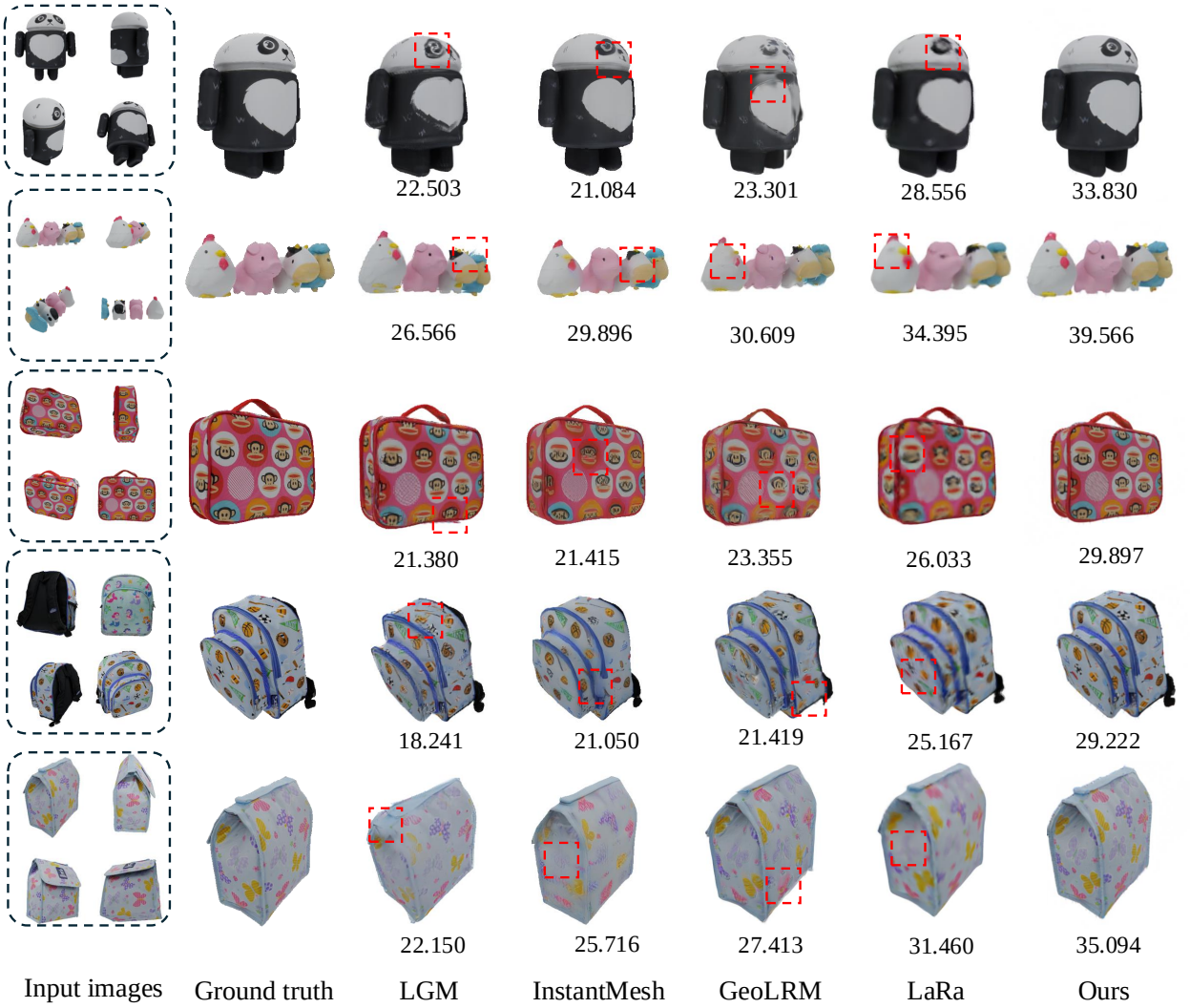


Figure 4. Visual comparisons to previous methods. The four-view input images are displayed in the leftmost column, while novel view renderings are compared on the right. Previous methods struggle to reconstruct high-frequency details and thin structures consistently. In contrast, our NovelGS demonstrates significantly improved performance in these scenarios. The PSNRs are provided beneath each image.

finetune the model with a resolution of 512 $\times$ 512 and a batch size of 2 in each GPU. We use the same optimizer [22] as the first stage with an initial learning rate of 4e-5. At each training step of both stages, we sample a set of 8 images (from 32 renderings) as a data point, from which we randomly select 4 clean views, 1 noisy view, and 3 supervision views independently. To optimize GPU memory usage, deferred back-propagation [61] and memory-efficient attention [17] are employed. The model is trained on 16 NVIDIA A100 GPUs with gradient accumulation set to 8. It requires approximately two weeks to complete the training stages.

4.2. Results and comparisons

Quantitative results. In the main experiments, we select 4 clean view images and 1 noisy view image as default. We report the quantitative results of sparse view reconstruction on different evaluation sets as shown in Table 1 and Table 2, respectively. For each metric, we highlight the top two results among all methods, and a deeper color indicates a better result. The quantitative evaluation of 2D novel view synthesis metrics indicates that NovelGS significantly outperforms the baseline models in terms of Structural Similarity Index (SSIM) [49] and Peak Signal-to-Noise Ratio (PSNR) [63]. This superior performance suggests that NovelGS generates outputs with enhanced quality. Notably, the Learned Perceptual Image Patch Similarity (LPIPS) of NovelGS is marginally lower than that of the top-performing baseline. This observation implies that the perception of novel views generated by NovelGS exhibits slight deviations from the ground truth in human views. Be-

cause it will predict a novel view based on known input images, attributed to the "dreaming" process inherent in the novel view diffusion process. Our model tries to image the unknown parts of the object that are more conscious of the true structure of the object. At the same time, it maintains consistency across multiple viewpoints rather than ignoring details to make the image look sensible in human views compared to the InstantMesh, as shown in the fourth row at Figure 4. We believe prioritizing the consistently detailed structure of objects is imperative in the reconstruction tasks.

Table 1. Evaluation results on the GSO dataset. The best and the second-best scores are marked as red and light red . ↑represents the higher the better, and ↓represents the lower the better.

	Google Scanned Objects [10]		
	PSNR ↑	SSIM ↑	LPIPS ↓
LGM [44]	24.923	0.907	0.093
InstantMesh [55]	25.124	0.924	0.059
GeoLRM [60]	25.389	0.918	0.083
LaRa [6]	28.910	0.940	0.091
Ours	31.303	0.946	0.065

Table 2. Evaluation results on Omni3D dataset.

	OmniObject3D [53]		
	PSNR ↑	SSIM ↑	LPIPS ↓
LGM [44]	24.229	0.913	0.081
InstantMesh [55]	24.292	0.929	0.053
GeoLRM [60]	24.289	0.922	0.083
LaRa [6]	28.434	0.943	0.084
Ours	31.195	0.945	0.067

Quantitative results. As illustrated in Figure 4, to compare our NovelGS with other baselines qualitatively, we select several objects from the GSO evaluation set and obtain the sparse-view recon results. For each generated, we visualize the images of the rendering from the same viewpoints. NovelGS consistently produces visually consistent appearances, whereas baseline methods often manifest distortions in the synthesized novel views. Specifically, the NeRF-based method (InstantMesh) prefers a smooth texture, which leads to blurring on some details, as shown in the third and fifth rows of the Figure 4. While other feed-forward pixel-aligned Gaussian reconstruction models would ignore some uncovered or slightly covered parts by the input view as shown in the Figure 4.

4.3. Ablation

The key design in our method is the utilization of noisy views. We analyze our approach regarding the necessity of the noisy views, the number of noisy views, and the different positions of the noisy view and clean views.

Necessity of The Noisy Views. We show qualitative comparisons of our models with and without noisy view in Table 3 and Table 4 on GSO and Omni3D elevation sets respectively. We can see that our model consistently achieves better quality when using noisy view images for denoising, benefiting from capturing more shape and appearance information through interacting with known clean views sufficiently. As shown in Figure 5 setting 5, it could not generate a reasonable appearance without noisy view denoising, which is the core limation of pixel-aligned Gaussians.

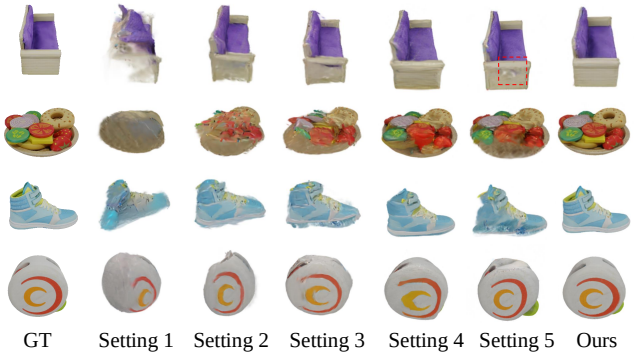


Figure 5. Qualitative results of different numbers of views. **Setting 1**: 1 clean view and 1 noisy view. **Setting 2**: 2 clean views and 1 noisy view. **Setting 3**: 3 clean views and 1 noisy view. **Setting 4**: 4 clean views and 2 noisy views. **Setting 5**: 4 clean views.

Table 3. Evaluation results on the GSO dataset [10]. ✓means it exists, × means it doesn't exist.

Noisy View	PSNR ↑	SSIM ↑	LPIPS ↓
×	29.985	0.945	0.070
✓	31.303	0.946	0.065

Table 4. Evaluation results on the Omni3D dataset [10].

Noisy View	PSNR ↑	SSIM ↑	LPIPS ↓
×	29.158	0.944	0.069
✓	31.195	0.945	0.067

Number of noisy and clean views. We present qualitative comparisons of our models with varying numbers of clean and noisy views in two different elevation sets, as detailed in Table 5 and Table 6. It reveals that the model's performance improves with an increased number of input clean images, attributable to the enhanced capture of shape and appearance information. Although novel view image denoising could promote the performance of unseen parts of the object, the computational complexity also increases significantly. So there needs to be a balance between the number of clean views and noisy views. Beyond this threshold, the presence of excessively noisy views detrimentally impacts the model's performance. As shown in the Figure 5 setting 4, more noise view images will create more noisy Gaussian points, which will blur the image.

Table 5. Evaluation results on GSO dataset. **NCV**: **N**umber of **C**lean **V**iews. **NNV**: **N**umber of **N**oisy **V**iews.

NCV	NNV	PSNR ↑	SSIM ↑	LPIPS ↓
1	1	21.668	0.895	0.167
2	1	26.913	0.922	0.100
3	1	29.574	0.938	0.075
4	2	31.256	0.941	0.069
4	1	31.303	0.946	0.065

Table 6. Evaluation results on Omni3D dataset.

NCV	NNV	PSNR ↑	SSIM ↑	LPIPS ↓
1	1	20.378	0.894	0.163
2	1	26.108	0.923	0.097
3	1	29.140	0.939	0.072
4	2	31.040	0.941	0.067
4	1	31.303	0.946	0.065

Positional relationship between noisy view and clean views. As shown in the Figure 6, we place the object in the center and surround the cameras. We fix the clean view images and camera parameters at positions 0^{th}, 3^{th}, 6^{th}, and 9^{th} as inputs, which cover the front of the object while not covering the back of the object. Moreover, we select positions 9^{th}, 10^{th}, 12^{th}, 15^{th}, and 18^{th} as the positions of the noisy view, respectively. We present quantitative comparisons of our models with varying camera poses of noisy views, as detailed in Table 7, Table 8. When choosing 15^{th} as the noisy view position, the model gets the best metric. As shown in the second and third rows of Figure 7, choosing the 15^{th} view presents the best result. Even though there are some differences between this image and the ground truth, this is a reasonable phenomenon. Because the input image does not contain the parts of the object that we expect to generate. It's reasonable for the model to imagine the unseen parts and generate a detailed image.

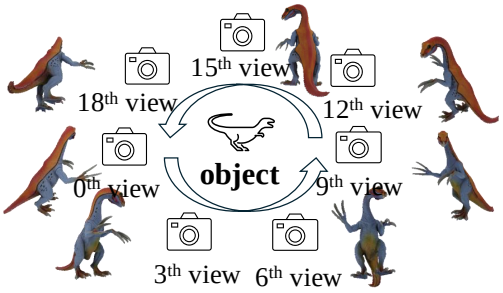


Figure 6. Camera position demonstration.

In conclusion, if we choose a noise view that is close to the known views, the model will take less account of parts that are not covered. As a result, it will lead to poor results in places that are not covered by the existing perspective. If we choose the positions of the noisy view and clean views that better cover the object, the model will take more account of the objects, producing better results.

Figure 7. Visulation of the results of different positions about the noisy view. The input images are shown on the left. i^{th} represent the position of the noisy view image as shown in Figure 6.

Table 7. Evaluation results on GSO dataset. **ICV**: **I**ndex of **C**lean **V**iews. **INV**: **I**ndex of **N**oisy **V**iews

ICV	INV	PSNR ↑	SSIM ↑	LPIPS ↓
0,3,6,9	9	31.436	0.947	0.063
0,3,6,9	10	31.592	0.948	0.062
0,3,6,9	12	31.707	0.948	0.063
0,3,6,9	18	31.643	0.949	0.062
0,3,6,9	15	32.038	0.950	0.061

Table 8. Evaluation results on Omni3D dataset.

ICV	INV	PSNR ↑	SSIM ↑	LPIPS ↓
0,3,6,9	9	32.008	0.949	0.055
0,3,6,9	10	31.979	0.948	0.055
0,3,6,9	12	31.851	0.950	0.056
0,3,6,9	18	31.807	0.952	0.056
0,3,6,9	15	32.055	0.951	0.054

5. Conclusion

In this paper, we introduce NovelGS, an innovative diffusion model designed for Gaussian Splatting (GS) using sparse-view images. Our approach employs a transformer-based network for novel view denoising, enabling the generation of 3D Gaussians. By incorporating both conditional views and noisy target views as inputs, the network predicts pixel-aligned Gaussians for each view. During the training phase, the rendered target and additional Gaussian views are supervised. In the inference phase, target views are iteratively rendered and denoised from pure noise. Our method demonstrates state-of-the-art performance in addressing the multi-view image reconstruction challenge. By generatively modeling unseen regions, NovelGS effectively reconstructs 3D objects with consistent and sharp textures. Experimental results on publicly available datasets show that NovelGS significantly outperforms existing image-to-3D frameworks, both qualitatively and quantitatively. Furthermore, we highlight the potential of NovelGS in generative tasks, such as text-to-3D and image-to-3D, by integrating it with existing multiview diffusion models.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 4

[2] Eric R Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *CVPR*, 2021. 3

[3] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *CVPR*, 2022. 1, 3

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3

[5] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, 2024. 4

[6] Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger. Lara: Efficient large-baseline radiance fields. In *ECCV*, 2024. 7

[7] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *ICCV*, 2023. 1

[8] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023. 1, 3, 5

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

[10] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *ICRA*, 2022. 3, 5, 7

[11] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 3

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NeurIPS*, 2014. 3

[13] Anchit Gupta, Wenhan Xiong, Yixin Nie, Ian Jones, and Barlas Oğuz. 3dgen: Triplane latent diffusion for textured mesh generation. *arXiv preprint arXiv:2303.05371*, 2023. 3

[14] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. In *ICLR*, 2024. 1, 3

[15] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 3

[16] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 2023. 1, 3, 4

[17] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, Daniel Haziza, Luca Wehrstedt, Jeremy Reizenstein, and Grigory Sizov. xformers: A modular and hackable transformer modelling library. https://github.com/facebookresearch/xformers, 2022. 6

[18] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In *ICLR*, 2024. 1, 3, 5

[19] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *CVPR*, 2023. 1

[20] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. *CVPR*, 2024. 3

[21] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998. 1

[22] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5, 6

[23] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *CVPR*, 2023. 1

[24] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 3

[25] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *CVPR*, 2019. 3

[26] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 3

[27] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *CVPR*, 2021. 3

[28] Evangelos Ntavelis, Aliaksandr Siarohin, Kyle Olszewski, Chaoyang Wang, Luc V Gool, and Sergey Tulyakov. Autodecoding latent 3d diffusion models. *NeurIPS*, 2023. 3

[29] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 5

[30] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1

[31] Alec Radford. Improving language understanding by generative pre-training. 2018. 3

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 2, 3, 5

[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3

[34] Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Trans. Graph.*, 2023. 1

[35] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. 2, 3

[36] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *CVPR*, 2023. 3

[37] Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In *CVPR*, 2024. 3

[38] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 3

[39] Ivan Skorokhodov, Sergey Tulyakov, Yiqun Wang, and Peter Wonka. Epigraf: Rethinking training of 3d gans. *NeurIPS*, 2022. 3

[40] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*. PMLR, 2015. 3

[41] Jingxiang Sun, Bo Zhang, Ruizhi Shao, Lizhen Wang, Wen Liu, Zhenda Xie, and Yebin Liu. Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior. In *ICLR*, 2024. 1

[42] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Viewset diffusion:(0-) image-conditioned 3d generative models from 2d data. In *ICCV*, 2023. 4

[43] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *CVPR*, 2024. 4

[44] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *ECCV*, 2024. 1, 3, 4, 7

[45] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *ICLR*, 2024. 1

[46] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, , Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024. 3

[47] A Vaswani. Attention is all you need. *NeurIPS*, 2017. 5

[48] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. In *ICLR*, 2024. 3

[49] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 2004. 6

[50] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *NeurIPS*, 2023. 1

[51] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. Crm: Single image to 3d textured mesh with convolutional reconstruction model. In *ECCV*, 2024. 1, 3

[52] Xinyue Wei, Kai Zhang, Sai Bi, Hao Tan, Fujun Luan, Valentin Deschaintre, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Meshlrm: Large reconstruction model for high-quality mesh. *arXiv preprint arXiv:2404.12385*, 2024. 1, 3, 4

[53] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *CVPR*, 2023. 3, 5, 7

[54] Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. In *CVPR*, 2023. 1

[55] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. 1, 3, 5, 7

[56] Yinghao Xu, Sida Peng, Ceyuan Yang, Yujun Shen, and Bolei Zhou. 3d-aware image synthesis via learning structural and textural representations. In *CVPR*, pages 18430–18439, 2022. 3

[57] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. *arXiv preprint arXiv:2403.14621*, 2024. 1, 3, 4

[58] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. *ICLR*, 2024. 1, 3

[59] Bowen Zhang, Yiji Cheng, Jiaolong Yang, Chunyu Wang, Feng Zhao, Yansong Tang, Dong Chen, and Baining Guo. Gaussiancube: Structuring gaussian splatting using opti-

mal transport for 3d generative modeling. *arXiv preprint arXiv:2403.19655*, 2024. 1

[60] Chubin Zhang, Hongliang Song, Yi Wei, Yu Chen, Jiwen Lu, and Yansong Tang. Geolrm: Geometry-aware large reconstruction model for high-quality 3d gaussian generation. *arXiv preprint arXiv:2406.15333*, 2024. 1, 3, 7

[61] Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *ECCV*, 2022. 6

[62] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. *arXiv preprint arXiv:2404.19702*, 2024. 1, 3

[63] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 5, 6