This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

GSV3D: Gaussian Splatting-based Geometric Distillation with Stable Video Diffusion for Single-Image 3D Object Generation

Ye Tao[1], Jiawei Zhang[2], Yahao Shi[1], Dongqing Zou[2,3], and Bin Zhou[1,3]*
[1]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University
[2]SenseTime Research [3]Peace-Bay Academy of Virtual Reality

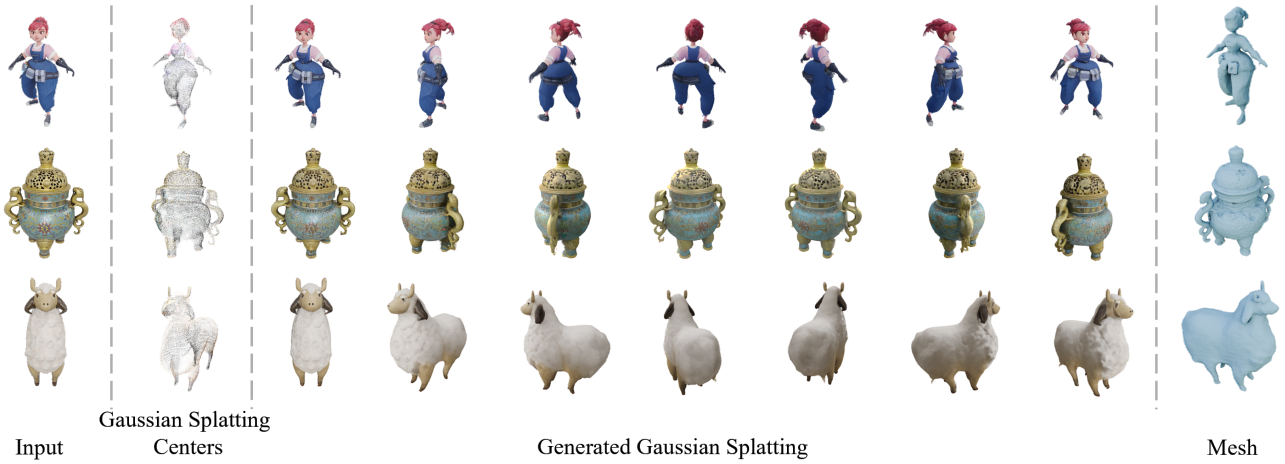


Figure 1. GSV3D utilizes Stable Video Diffusion and Gaussian Splatting decoder to generate 3D model from a single image. Meanwhile, the Gaussian Splatting-based geometric distillation is used to enhance the 3D consistency.

Abstract

Image-based 3D generation has vast applications in robotics and gaming, where high-quality, diverse outputs and consistent 3D representations are crucial. However, existing methods have limitations: 3D diffusion models are limited by dataset scarcity and the absence of strong pre-trained priors, while 2D diffusion-based approaches struggle with geometric consistency. We propose a method that leverages 2D diffusion models' implicit 3D reasoning ability while ensuring 3D consistency via Gaussian-splatting-based geometric distillation. Specifically, the proposed Gaussian Splatting Decoder enforces 3D consistency by transforming SV3D latent outputs into an explicit 3D representation. Unlike SV3D, which only relies on implicit 2D representations for video generation, Gaussian Splatting explicitly encodes spatial and appearance attributes, enabling multi-view consistency through geometric constraints. These constraints correct view inconsistencies, ensuring robust geometric consistency. As a result, our approach simultaneously generates high-quality, multi-view-consistent images and accurate 3D models, providing a scalable solution for single-image-based 3D generation and bridging the gap between 2D Diffusion diversity and 3D structural coherence. Experimental results demonstrate state-of-the-art multi-view consistency and strong generalization across diverse datasets. Our code is available at https://github.com/MOMOYATW/GSV3D.

1. Introduction

Image-based 3D generation has emerged as a crucial area of research with transformative applications in robotics, gaming, virtual reality, and other areas where accurate 3D modeling from limited information can bring substantial benefits. However, achieving this goal remains challenging due to the intrinsic limitation of 2D diffusion models in ensuring geometric consistency and the reliance on 3D diffusion models on limited datasets, leading to fidelity issues and monotonous textures. Existing purely 3D diffusion models are often limited by the scarcity of datasets, which limits their ability to produce diverse and realistic results [9, 13, 15, 16, 18, 20, 22, 37, 38]. This limitation restricts the generalization ability of these models and makes it difficult for them to capture complex details across var-

*Corresponding Author: Bin Zhou, zhoubin@buaa.edu.cn

ious object types. Conversely, 2D diffusion-based methods have shown substantial diversity in generated results due to their reliance on rich 2D datasets [26]. However, maintaining geometric consistency across multiple views remains a significant challenge. Although 2D diffusion models often struggle to maintain geometric consistency, they possess the implicit ability to generate consistent structures. Some recent approaches have attempted to address this by directly fine-tuning 2D diffusion models with additional attention mechanisms to encourage multiview consistency [28, 29, 31, 33, 36]. Yet, these methods primarily apply implicit constraints that lack the robustness needed for consistent geometric alignment, often resulting in breakdowns in multiview consistency. The absence of explicit geometric constraints frequently leads to discrepancies across views, ultimately compromising the accuracy and usability of the generated 3D models.

To address these issues, we propose a framework that combines the diversity of 2D diffusion with the geometric consistency required for 3D generation. Our approach builds on SV3D [3], an advanced diffusion model optimized for generating coherent video sequences that rotate around an object from single images. To further enhance multi-view consistency, we introduce geometric distillation, a strategy that transforms multi-view latents into a structurally aligned 3D representation to enforce cross-viewpoint coherence. Unlike standard latent diffusion processes, which operate purely in 2D space, our method applies a Gaussian Splatting-based transformation to distill multi-view information into a unified 3D-aware representation. By representing latents in 3D, this approach amplifies geometric inconsistencies between different latents, providing an opportunity to incorporate additional supervision, such as depth information, to improve their consistency. This ensures more accurate and coherent results across views. Consequently, our approach not only overcomes the geometric inconsistencies of 2D diffusion models but also directly generates 3D results without requiring additional reconstruction steps. Furthermore, compared to 3D diffusion models, our method achieves higher fidelity and more realistic, diverse textures, providing a robust and reliable solution for high-quality, geometrically consistent multiview generation.

In summary, our contributions are as follows:

- We propose a latent decoder trained to extract 3D Gaussian Splatting representations directly from multi-view latents generated by the diffusion model, enabling direct 3D model generation.
- We introduce a method that leverages Gaussian Splatting for geometric distillation within the SVD framework, enforcing structural coherence to enhance multi-view consistency and improve 3D reconstruction fidelity from a single image.
- We validate our approach through extensive experiments, showcasing its superior performance and generalization capabilities in generating consistent 3D objects from 2D inputs.

2. Related Works

2.1. Native 3D Generative Models

Recent advancements in 3D generative modeling have seen a rise in native 3D generation methods, which directly create 3D representations. These approaches are diverse, as 3D data can be represented in various forms, Signed Distance Fields (SDF), meshes, point clouds, NeRF, and Gaussian Splatting. However, it's important to highlight that certain 3D representations, such as point clouds [22, 37], SDF [16, 18, 38], and meshes [5, 20], are limited to capturing geometry and do not support texture generation. As a result, additional work is required to generate textures separately for these models. On the other hand, representations like NeRF [11, 13] and Gaussian Splatting [15, 25, 39] are more integrated, as they can model both geometry and texture simultaneously, which makes them more powerful in generating visually compelling 3D content. Despite these advancements, the scarcity and monotony of available 3D datasets [6, 7] remain significant challenges. This limitation means that even with advanced methods, generated geometries and textures often lack high fidelity and realism, reinforcing the need for further research to address these issues. Overall, while these native 3D generation methods show great promise, they still face challenges related to dataset diversity, fidelity, and the balance between geometry and texture.

2.2. 2D Generative Models for 3D Generation

While native 3D generation methods still face challenges due to the scarcity of diverse 3D datasets, 2D diffusion models, benefiting from abundant 2D data [27], have shown exceptional diversity and realism in generated results [26]. However, a major issue with 2D-generated content is the geometric inconsistency between multiple images, making it difficult to reconstruct a coherent 3D model from them. To address this, early works like DreamFusion [24] proposed using a Score-Distilled (SDS) loss to optimize a randomly initialized NeRF field, essentially distilling 3D structure from 2D diffusion models. While this approach demonstrated promising results in 3D generation, it was highly time-consuming, with some methods [34] taking hours to generate a single object. In response to the inefficiency of SDS-based methods, later research focused on equipping 2D diffusion models with some degree of 3D awareness. For example, Zero-1-to-3 [19] integrated camera pose information into the conditioning process of the 2D model, allowing it to generate images from novel viewpoints. Meth-

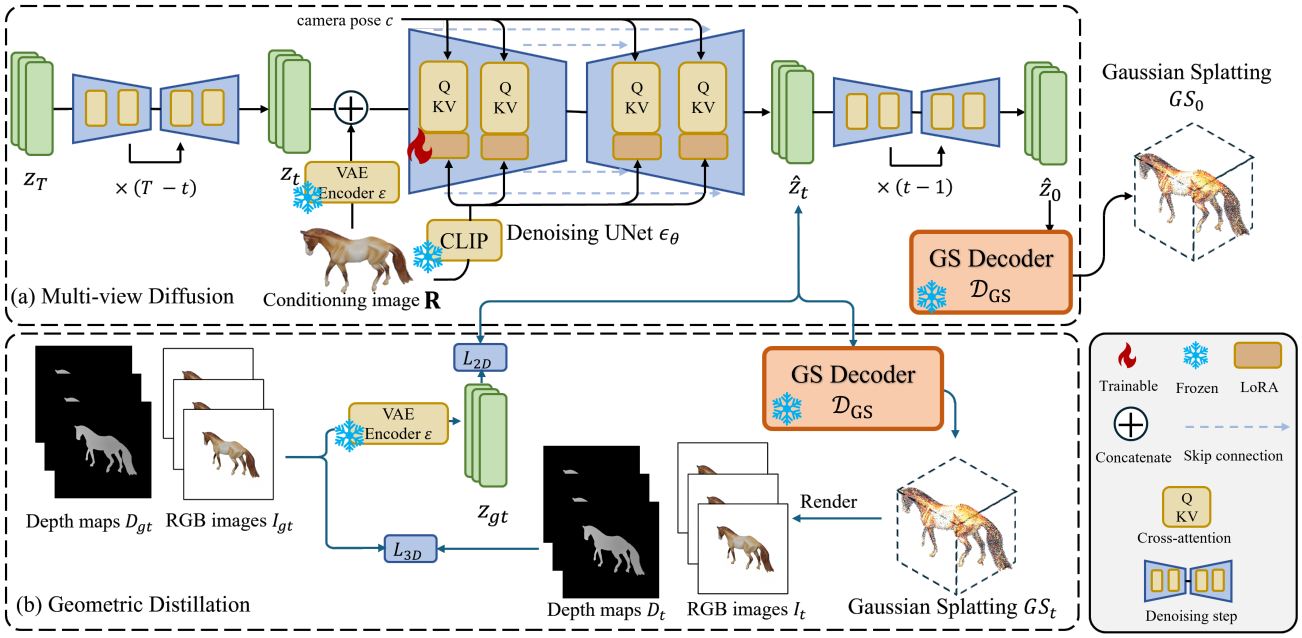


Figure 2. **Overview of GSV3D Training and Inference Pipeline.** During inference, given an initialized noise latent z_T, an input image $\mathbf{R}$ and its corresponding camera pose c, the approach follows a structured pipeline: At each step of the sampling, the input image is encoded using VAE $\mathcal{E}$, and its latent representation is concatenated with input noisy multi-view latents z_t. Simultaneously, the image $\mathbf{R}$ is encoded via CLIP and, along with the camera pose c, integrated into the diffusion process via cross-attention. For the last step, the multi-view denoised latents $\hat{z}_0$ are processed by the Gaussian Splatting Decoder to reconstruct a 3D representation GS_0, as shown in (a). During training, the geometric distillation process utilizes the Gaussian Splatting Decoder with 3D constrain to distill geometric knowledge from the multi-view latent representations $\hat{z}_t$ generated by the multi-view diffusion model at each diffusion step, which then serves as supervision $\mathcal{L}_{3D}$ to guide the learning of 3D geometry. At the same time, the multi-view latent loss $\mathcal{L}_{2D}$ is imposed in latent space, as shown in (b).

ods [28, 29, 31, 33, 36] leverage attention mechanisms to enhance multi-view consistency, either by enforcing geometric alignment across multiple views or by transferring temporal consistency from videos to 3D object generation, to ensure that the generated images adhere to the same 3D structure. Despite these advancements, the attention mechanisms that enhance multi-view consistency primarily achieve geometric alignment in appearance rather than understanding true 3D structures. These models may generate visually consistent views but still struggle with robust 3D reconstruction. When applied to tasks such as object reconstruction from multiple views, these methods often reveal geometric inconsistencies, highlighting a significant gap in the ability of current 2D diffusion models to fully comprehend and generate 3D data.

3. Method

3.1. Overview

To leverage the powerful priors of 2D diffusion models, generating a 3D representation GS_0 from a single image $\mathbf{R}$ requires synthesizing realistic consistent multi-view observations. SV3D [31] is a multi-view diffusion model that can generate 360-degree multi-view images around an object by modeling temporal dependencies (Section 3.2). While it produces visually realistic results, the absence of explicit 3D constraints results in geometrically inconsistent reconstructions. To address this, we introduce a Gaussian Splatting Decoder, which transforms the multi-view latents into an explicit 3D structure (Section 3.3). These multi-view latents are intermediate representations that capture and encode the features of multi-view images generated by the multi-view diffusion model. This decoder provides a learnable 3D representation, better capturing the underlying geometry. With the above Gaussian Splatting Decoder, we propose a framework GSV3D, which incorporates the Gaussian Splatting Decoder into its training pipeline to enhance the 3D awareness of the multi-view diffusion model. This integration effectively enables the multi-view diffusion model to generate geometrically consistent multi-view latents by incorporating 3D constraints through geometric distillation (Section 3.4). Finally, we use the Gaussian Splatting Decoder to directly convert the multi-view latents generated by GSV3D to Gaussian Splatting, resulting in a 3D representation. An overview is shown in Figure 2.

3.2. Multi-view Diffusion Model for 3D Generation

SV3D is a multi-view diffusion model proposed in prior work which is built upon the Stable Video Diffusion [3] (SVD) framework. It is capable of generating a 360-degree

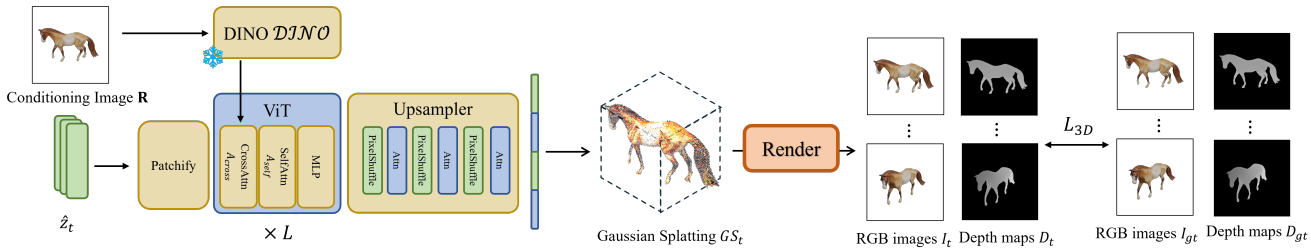


Figure 3. **Overview of the Gaussian Splatting Decoder pipeline.** During both geometric distillation and GSV3D inference, the conditioning image $\mathbf{R}$ is first processed by a pre-trained DINO encoder $\mathcal{DINO}$ to extract features, which are then integrated into the ViT via cross-attention. The ViT processes the latents $\hat{z}_t$ generated by GSV3D. The output of the ViT is passed through an upsampler before being converted into a 3D Gaussian Splatting representation. In order to train the Gaussian Splatting Decoder, the latents $\hat{z}_t$ are substituted with the latents z_{input} encoded from the input multi-view images I_{input} using a pre-trained VAE encoder $\mathcal{E}$. These input multi-view images consist of N images, each corresponding to a viewpoint uniformly distributed around the object.

rotation video around an object from a single image. This capability is achieved by fine-tuning the SVD model on a 3D dataset and incorporating a camera pose control mechanism that controls the viewpoint of the generated images. In this section, we provide a brief overview of the SV3D architecture as described in the original work, which serves as a foundation of our approach.

The conditioning image $\mathbf{R}$ is encoded into two distinct representations using pre-trained encoders: a VAE encoder and a CLIP encoder. The VAE encoder $\mathcal{E}$ captures geometric and appearance features, producing a latent representation that is concatenated with the noisy multi-view latents z_t at each denoising step. On the other hand, the CLIP encoder extracts semantic information, which is integrated into the denoising process through a cross-attention mechanism. This dual-encoding strategy ensures that both low-level geometric details and high-level semantic context are effectively utilized during generation. Additionally, to control the generated view, the camera pose c for each generated image is encoded using cosine encoding and incorporated into the cross-attention mechanism. The UNet ϵ_θ is a denoising model, which utilizes cross-frame attention to ensure spatial consistency in the denoised multi-view latents.

During inference, the process starts from pure noise z_T in the latent space, and the denoising model ϵ_θ iteratively refines it through a series of denoising steps. The final multi-view latent representations $\hat{z}_0$ are then decoded by the VAE decoder to reconstruct the output images, which consist of N images corresponding to distinct viewpoints that are approximately uniformly distributed around the object. However, although the generated images appear visually plausible, their lack of geometric consistency poses significant challenges for accurate 3D reconstruction, which heavily relies on precise geometric consistency.

3.3. Gaussian Splatting Decoder

To achieve consistent multi-view results in GSV3D, we propose the Gaussian Splatting Decoder. This decoder extracts the implicit geometric information from the multi-view latents generated by the multi-view diffusion model. Through the geometric distillation process(detailed in Section 3.4), it introduces 3D constraints into GSV3D, enhancing its ability to produce geometrically consistent multi-view latents. Additionally, the Gaussian Splatting Decoder can convert the multi-view latents generated by GSV3D into a Gaussian Splatting representation directly to obtain the final 3D result.

To achieve this, we adopt a structure inspired by GRM [35], where a VisionTransformer (ViT) [1] is used to extract features from latents, followed by an Upsampler to generate a high-resolution Gaussian Splatting output, as shown in Figure 3.

Specifically, latents $\hat{z}_t$ are processed by a ViT with $L = 16$ layers, which extracts hierarchical features through self-attention mechanisms A_{self}. These mechanisms capture both local and global dependencies by operating on patches of the latent representations. To improve the quality of generated Gaussian Splatting, we also introduce a cross-modal feature fusion strategy. This strategy integrates high-level semantic features from conditioning image $\mathbf{R}$ to enhance the latent space representation. Specifically, we extract global structural cues using a pre-trained DINO [4, 23] model $\mathcal{DINO}$, which provides a feature representation denoted as $\mathbf{F}_{\text{DINO}}$:

$$\mathbf{F}_{\text{DINO}} = \mathcal{DINO}(\mathbf{R}). \tag{1}$$

The extracted DINO features $\mathbf{F}_{\text{DINO}}$ capture important semantic context, which helps guide the model in generating more coherent and structurally consistent reconstructions. To incorporate these global cues into ViT processing, we introduce a cross-attention mechanism that allows the DINO features $\mathbf{F}_{\text{DINO}}$ to attend to the intermediate features of ViT $\mathbf{F}_{\text{ViT}}$. This mechanism is integrated into each ViT layer and operates as:

$$\mathbf{F}_{\text{ViT}} \leftarrow \mathcal{A}_{\text{cross}}(\mathbf{F}_{\text{ViT}}, \mathbf{F}_{\text{DINO}}), \tag{2}$$

where $\mathcal{A}_{\text{cross}}(\cdot, \cdot)$ denotes the cross-attention operation. This integration allows the latent space to be enriched with

global structural priors, which improves the quality of the Gaussian Splatting reconstruction. Finally, these extracted features are processed by the Upsampler to generate the final Gaussian Splatting reconstruction GS. The Upsampler consists of a series of pixel shuffle layers and transformer layers that gradually increase the spatial resolution of the features while reducing the number of channels. The output of the Upsampler is a high-resolution 3D Gaussian Splatting representation.

3.4. Training Strategy

3.4.1. Training Gaussian Splatting Decoder

In order to train the Gaussian Splatting Decoder, we render Q images for each of the P objects in the training set, with these images evenly distributed around the object. For each object, N images $\{I_{input}\}_{n=1}^{N}$ are uniformly selected from Q images as input views. All Q images serve as ground-truth images $\{I_{gt}\}_{q=1}^{Q}$ to supervise the corresponding object. Unlike in the geometric distillation process, we need to generate multi-view latents z_{input} to replace the multi-view latents $\hat{z}_t$ as the input of the Gaussian Splatting Decoder. These latents z_{input} are obtained by encoding the input multi-view images I_{input} using a pre-trained VAE $\mathcal{E}$. Meanwhile, the first input image I^0_{input} is used as the conditioning image $\mathbf{R}$.

To optimize the Gaussian Splatting Decoder, we render the RGB images I_{output} and depth maps D_{output} from the Gaussian Splatting representation GS_{output} generated by the model, as shown in Figure 3. Specifically, we use $I_{\text{gt}}^{(p,q)}$, $I_{\text{output}}^{(p,q)}$, $D_{\text{gt}}^{(p,q)}$, and $D_{\text{output}}^{(p,q)}$ to refer to the q-th ground-truth and rendered image/depth map for the p-th object. Notably, we render both the input views and the novel views to ensure a comprehensive optimization of the performance of the decoder across different perspectives. These rendered outputs are then used in the loss calculation, which involves a combination of loss functions. The RGB loss $\mathcal{L}_{\text{rgb}}$ is computed as the pixel-wise mean squared error (MSE) between the rendered RGB images I_{t} and the corresponding ground truth images I_{gt}:

$$\mathcal{L}_{\text{rgb}} = \frac{1}{P \cdot Q} \sum_{p=1}^{P} \sum_{q=1}^{Q} \left\| I_{\text{output}}^{(p,q)} - I_{\text{gt}}^{(p,q)} \right\|_2^2. \tag{3}$$

To further constrain the geometry of the generated structure, we apply depth loss $\mathcal{L}_{\text{depth}}$, which compares the rendered depth maps D_{output} with the ground-truth depth maps D_{gt}:

$$\mathcal{L}_{\text{depth}} = \frac{1}{P \cdot Q} \sum_{p=1}^{P} \sum_{q=1}^{Q} ||D_{\text{output}}^{(p,q)} - D_{\text{gt}}^{(p,q)}||_2^2. \tag{4}$$

The total loss function $\mathcal{L}_{\text{3D}}$ is defined as a weighted sum of multiple components, each contributing to different aspects of the reconstruction quality:

$$\mathcal{L}_{\text{3D}} = \mathcal{L}_{\text{rgb}} + \lambda_{\text{depth}} \mathcal{L}_{\text{depth}}, \tag{5}$$

The weight parameterλ_{depth} is introduced to balance the contributions of perceptual quality and depth supervision.

3.4.2. Geometric Distillation Process for GSV3D

In order to train GSV3D, we utilize the same method to generate the training dataset as we mentioned in Section 3.4.1. We choose the first ground-truth frame I^0_{input} as the conditioning image $\mathbf{R}$ for GSV3D. The ground-truth latent representations z_{gt} are encoded from the ground-truth multi-view images I_{gt} using the pre-trained VAE encoder $\mathcal{E}$.

While the multi-view diffusion model inherently possesses potential 3D reasoning capabilities, as it is able to generate multi-view perspectives that rotate around the object from a single 2D input image, its full 3D potential can only be unlocked with a framework that directs the model to better understand and leverage 3D structural information. To enhance the 3D awareness of the multi-view diffusion model, we introduce a geometric distillation framework that implicitly encodes 3D structural constraints into the generative process. It ensures that the model learns to reason about spatial relationships and multi-view consistency in a way that respects geometric constraints. The core of this approach is the Gaussian Splatting Decoder, which acts as the 3D-Aware Branch. At each time step t, Gaussian noise is added to z_{gt}, producing noisy latents z_t according to a predefined noise schedule. These noisy latents z_t are then processed by the multi-view diffusion model, which outputs denoised latents $\hat{z}_t$. The denoised latents, along with the conditioning image $\mathbf{R}$, are then passed through the Gaussian Splatting Decoder. We freeze the Gaussian Splatting Decoder and distill the multi-view diffusion model by attaching a LoRA [12] (Low-Rank Adaptation) module, which allows us to update the parameters of the multi-view diffusion model while preserving the information necessary for the 3D constrain in the Gaussian Splatting Decoder, as shown in Figure 2.

To distill the multi-view diffusion model, we apply the 3D loss $\mathcal{L}_{\text{3D}}$ in Eq. 5 to update the diffusion model. Additionally, we retain the original multi-view latents denoising loss, denoted as $\mathcal{L}_{\text{2D}}$. This loss is used to optimize the 2D diffusion model by comparing the generated noisy latents with the clean latent targets:

$$\mathcal{L}_{\text{2D}} = \mathbb{E}_{\mathbf{R}, z_t, z_{gt}} \left[\|\epsilon_\theta(z_t; \mathbf{R}, t) - z_{gt}\|_2^2 \right], \tag{6}$$

where $\mathbb{E}_{\mathbf{R}, z_t, z_{gt}}$ denotes the expectation over the joint distribution of training images $\mathbf{R}$, their corresponding noisy latents z_t and ground-truth latnets z_{gt}, $\epsilon_\theta(z_t; \mathbf{R}, t)$ is the output of the denoising network.

The total loss function, combining both the multi-view latents denoising loss ($\mathcal{L}_{\text{2D}}$) and the 3D consistency loss ($\mathcal{L}_{\text{3D}}$), is given by:

$$\mathcal{L}_{\text{distill}} = \mathcal{L}_{\text{2D}} + \lambda_{\text{3D}}\mathcal{L}_{\text{3D}}, \tag{7}$$

where λ_{3D} is a hyperparameter that controls the relative weighting of the 2D and 3D losses.

4. Experiments

4.1. Experimental Settings

Datasets. For training, we collect $P = 1 \times 10^5$ high-quality 3D models from Objaverse [6] dataset. For each model, we render $Q = 84$ sets of RGB images and depth maps. These images capture views of the object from different perspectives, where the azimuth angles are uniformly distributed around the object. While the elevation angle is fixed for each individual object, it is randomly sampled between -5 and 30 degrees across different objects. From these 84 images, we select $N = 16$ images at regular intervals to serve as the training input. For evaluation, we select a standardized subset of 100 randomly chosen models from the Google Scanned Objects (GSO) [8] dataset. For each model, we render 21 RGB images with a fixed elevation angle, which is randomly chosen between -5 and 30 degrees, and azimuth angles uniformly distributed across 360 degrees. The first rendered image of each object is designated as the input for GSV3d. The rest of the images are used for evaluation. Also, we sample 4,096 points from the mesh of each object to facilitate 3D evaluation.

Metrics. For evaluation, we assess the quality of appearance and geometry. For appearance evaluation, we render images from 21 viewpoints of the generated 3D representations. We employ various metrics: PSNR, SSIM, LPIPS, KID [2], FID [10], and CLIP-IQA [32] for appearance quality. For the geometry evaluation, we extract meshes from the generated 3D representations and then sample points from these meshes to compute the Chamfer Distance (CD), Intersection over Union (IoU), and F-Score with the ground truth point cloud, which measures geometric accuracy. We also conduct a user study to evaluate the perceptual realism and structural plausibility of the outputs.

Training Details. Our training process consists of two stages:

- **Stage One:** We begin by training a Gaussian Splatting Decoder, as we mentioned in Section 3.3. We train this model from scratch to obtain 3D representations from multi-view latents. This stage is trained for 80,000 steps with a batch size of 128.
- **Stage Two:** We integrate the Gaussian Splatting Decoder into the multi-view diffusion model in GSV3D to perform geometric distillation, as mentioned in Section 3.4.2. The Gaussian Splatting Decoder is initialized with the first-stage trained model, while the UNet, VAE encoder, and CLIP encoder are initialized with the pre-trained SV3D model. During distillation, the parameters of the Gaussian Splatting Decoder, the UNet, VAE encoder, and CLIP encoder are frozen, with only the LoRA parameters, which are incorporated into the UNet model, being updated. This stage is trained for 40,000 steps with a reduced batch size of 32.

In both stages, we use AdamW [21] with a learning rate of 1×10^{-5} to optimize the model. The hyperparameter λ_{3D} and λ_{depth} is set to 1.5 and 0.2, respectively. The experiments are conducted on 4 NVIDIA A100-80G GPUs.

Baselines. We compare our model against the following state-of-the-art methods: GaussianAnything [15] (GA), TriplaneGaussian [39] (TGS), LGM [30], Zero123Plus [28], Era3D [17], and SV3D [31]. Among these methods, GaussianAnything and TriplaneGaussian can be considered as 3D methods. They generate 3D representations without the intermediate step of estimating 2D images and are trained from scratch on 3D datasets. In contrast, LGM, Zero123Plus, Era3D, and SV3D are regarded as 2D methods. These methods are typically based on strong pre-trained 2D Diffusion models to help construct 3D representations.

4.2. Evalutaion on 3D Generation

Appearance Quality. From Figure 4, it can be observed that the appearance quality of 3D methods is inferior. Specifically, the 3D models generated by GaussianAnything exhibit color deviations from the original images (as seen in the first and second examples), while the results produced by TriplaneGaussian are blurry (as shown in the fourth example). These issues are likely attributed to the limited dataset diversity used in training 3D methods, which hampers their ability to generate realistic appearances.

In contrast, 2D methods [17, 28, 30, 31] demonstrate superior performance in terms of appearance generation. The results exhibit richer and more authentic details, as shown in the third example, where the 2D methods capture more intricate folds on the horse's back. However, due to poor image consistency, the reconstructed results of these images suffer from blurry and ghosting artifacts, which degrade the evaluation metrics, as shown in Table 1. In addition, Era3D performs poorly due to its design choice of generating images with an elevation of 0 to reduce computational costs. While this approach saves attention computation, it results in inaccurate reconstructions of the unseen parts of objects and produces fragmented outputs, as demonstrated in the first and second examples. Our method inherits the generative advantages of 2D methods while overcoming their inconsistency issues, thus achieving the best results on appearance metrics.

Geometry Quality. As can be seen from Figure 4, 3D methods [15, 39] tend to generate intact objects by directly modeling 3D structures, avoiding the information loss in-

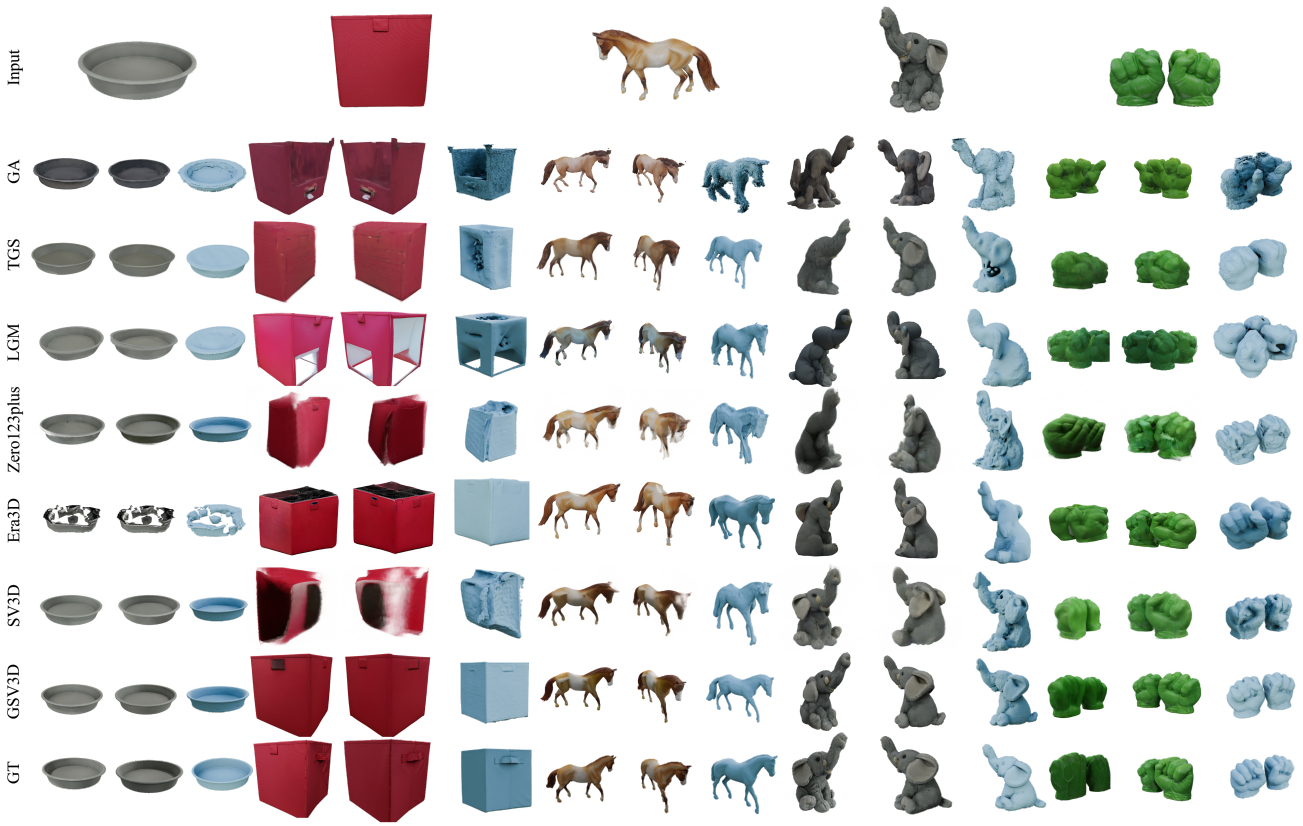


Figure 4. Performance comparison between our GSV3D and other state-of-art methods. GA and TGS are abbreviations for GaussianAnything and TriplaneGaussian, respectively. For each example, the first two columns display two different rendering views of the generated 3D representation, while the third column shows the rendered image of the extracted Mesh.

Type	Method	Appearance Quality						Geometry Quality			User Study	
		PSNR↑	SSIM↑	LPIPS↓	FID↓	KID↓	CLIP-IQA↑	CD↓	IoU ↑	F-Score↑	App. Score↑	Geo. Score↑
3D	GA [15]	15.201	0.834	0.039	95.47	1.17	0.805	0.197	0.502	0.303	3.154	5.000
	TGS [39]	18.874	0.872	0.032	85.25	1.28	0.812	0.272	0.415	0.232	4.000	3.846
2D	LGM [30]	17.013	0.845	0.033	61.66	0.45	0.819	0.238	0.431	0.308	3.538	4.000
	Zero123Plus [28]	15.787	0.827	0.037	82.80	0.82	0.825	0.165	0.525	0.409	2.088	2.154
	Era3D [17]	14.993	0.830	0.040	90.35	0.92	0.811	0.229	0.436	0.256	3.846	2.418
	SV3D [31]	17.772	0.863	0.034	74.86	0.80	0.816	0.200	0.418	0.311	4.231	2.923
2D-3D	GSV3D	20.390	0.884	0.023	50.79	0.21	0.839	0.100	0.721	0.661	6.308	6.692

Table 1. Quantitative comparisons on GSO dataset [8]. GA and TGS are abbreviations for GaussianAnything and TriplaneGaussian, respectively. GaussianAnything and TriplaneGaussians are 3D methods, while LGM, Zero123Plus, Era3D, and SV3D are 2D methods. The second and third-best performances are highlighted in red, orange, and yellow. Our GSV3D achieves the best performance on the GSO dataset.

herent in 2D-to-3D indirect representations. However, TriplaneGaussian suffers from the issue of sparse point cloud density when the object is relatively complex, which leads to holes or overly smoothness in the reconstructed mesh, as shown in the fourth and fifth examples. On the other hand, while GaussianAnything can generate denser point clouds, its fidelity is limited, as shown in the fourth and fifth examples. This limitation stems from the restricted diversity of the training data, which hinders the model's ability to generalize well to unseen or complex scenarios. 2D methods [17, 28, 30, 31] suffer from geometric inconsistencies, leading to distortions and holes in the results, as shown in the second example. Our method, constrained by explicit 3D representations, achieves better geometric quality.

User study. We conduct a user study to evaluate our method against existing approaches. For a collection of 21 images, we render 360-degree rotating videos of the 3D representations generated from these methods. There are in total 322 videos for evaluation in our user study. Each volunteer is presented with 10 samples from each of the follow-

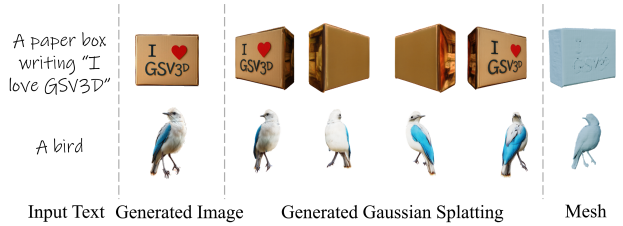


Figure 5. Examples of using GSV3D for text-to-image-to-3D generation.

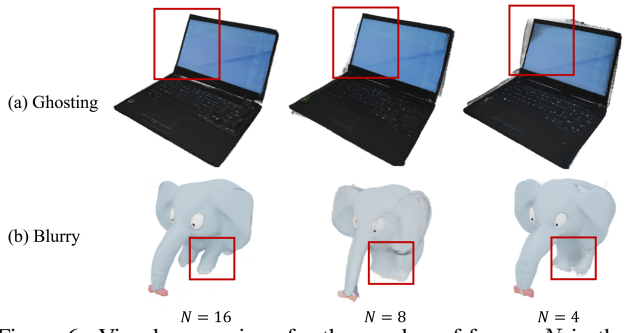


Figure 6. Visual comparison for the number of frames N in the multi-view latents generated by denoising UNet ϵ_θ. Reducing the number of frames weakens generative capability and 3D constraints, leading to ghosting and blurry regions in the outputs of GSV3D.

ing methods: GaussianAnything, TriplaneGaussian, LGM, Zero123Plus, Era3D, SV3D, and GSV3D, and asked to rank in two aspects: geometry quality and appearance quality. The top-ranked method receives 7 points and the lowest-ranked method receives 1 point. We collect results from 40 volunteers and get 2800 valid scores in total. As shown in Table 1, our method receives the highest score from participants in terms of both appearance and geometry.

Text-to-Image-to-3D. By integrating advanced text-to-image models (e.g., Stable Diffusion [26], FLUX [14]) into our GSV3D, we are capable of generating 3D models directly from textual descriptions, as shown in Figure 5.

4.3. Ablation Studies

In the ablation studies, all models are trained following the training procedures described previously.

Effect of Geometric Distillation. We evaluate the effectiveness of the RGB loss $\mathcal{L}_{\text{rgb}}$ and depth loss $\mathcal{L}_{\text{depth}}$, as mentioned in Section 3.4.2, introduced by the 3D-aware branch in the geometric distillation process. Table 2 presents the results of ablation studies conducted by distilling our GSV3D framework with different loss functions. The first row represents a version distilled without the RGB loss, while the second row represents a version distilled without the depth loss. The performance drops in both cases, underscoring the importance of supervising the 3D transition from latents through both RGB and depth constraints. Visualization results are shown in the supplementary material.

Setting	PSNR↑	SSIM↑	LPIPS↓	FID↓	KID(%)↓
Geometric Distillation					
w/o RGB loss	19.551	0.872	0.027	53.89	0.27
w/o Depth loss	19.826	0.877	0.025	53.20	0.32
Number of Frames					
N = 8	19.733	0.877	0.026	55.49	0.34
N = 4	19.548	0.874	0.027	57.85	0.42
DINO encoder					
w/o DINO encoder	19.508	0.876	0.027	55.20	0.30
Full Model (GSV3D)	**20.390**	**0.884**	**0.023**	**50.79**	**0.21**

Table 2. Ablation studies on the losses used in geometric distillation, the number of frames N in the multi-view latents generated by multi-view diffusion in GSV3D, and the use of the DINO encoder. The full model consists of RGB loss, depth loss, a configuration of $N = 16$, and a DINO encoder. All experiments are conducted on the GSO dataset.

Effect of the Number of Frames in the Latents Generated by GSV3D. To investigate the impact of the number of frames N in the multi-view latents generated by the denoising UNet ϵ_θ in GSV3D, as mentioned in Section 3.2, we conducted experiments comparing three settings to train the GSV3D, with N set to 4, 8, and 16 frames. The results in Table 2 demonstrate that the setting $N = 16$ outperforms the other settings in terms of overall performance. This improvement can be attributed to the higher number of overlapping regions between frames in the setting $N = 16$, which helps maintain consistency across views and provides a clear render result without ghosting, as shown in Figure 6 (a). In contrast, the setting $N = 4$ may leave some areas uncovered, leading to blurry results in the final generation, as shown in Figure 6 (b). Based on these results, we choose $N = 16$ as the final configuration for our model.

Effect of DINO encoder in Gaussian Splatting Decoder. We evaluate the impact of the DINO encoder within the Gaussian Splatting Decoder, as outlined in Section 3.3. By removing the DINO encoder from the Gaussian Splatting Decoder and utilizing the modified model to distill GSV3D, we observe a performance drop, as demonstrated in Table 2. This underscores the essential role of the DINO encoder in the overall framework.

5. Conclusion

This paper tackles the challenge of high-quality 3D object generation from a single image by integrating 2D diffusion models with explicit 3D constraints. While 2D diffusion models offer diversity but lack geometric consistency, and 3D diffusion models face data limitations, our hybrid approach bridges this gap. Leveraging the Stable Video Diffusion with Gaussian Splatting, we transform multi-view latents into coherent 3D representations, enhancing consistency and fidelity. Extensive experiments show the superior performance of our method and generalization capabilities in generating consistent and realistic 3D objects from 2D inputs.

Acknowledgments. We thank all the anonymous reviewers for their constructive comments and suggestions to improve the paper. This work was supported in part by National Key Research and Development Program of China (2022YFF0904303), National Natural Science Foundation of China (61932003), and Beijing Science and Technology Plan Project (Z221100006322003).

References

[1] Dosovitskiy Alexey. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv: 2010.11929*, 2020. 4

[2] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 6

[3] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 2, 3

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 4

[5] Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, et al. Meshanything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163*, 2024. 2

[6] 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*, pages 13142–13153, 2023. 2, 6

[7] 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*, 36, 2024. 2

[8] 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*, pages 2553–2560. IEEE, 2022. 6, 7

[9] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *NeurIPS*, 35:31841–31854, 2022. 1

[10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017. 6

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

[12] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 5

[13] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 1, 2

[14] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 8

[15] Yushi Lan, Shangchen Zhou, Zhaoyang Lyu, Fangzhou Hong, Shuai Yang, Bo Dai, Xingang Pan, and Chen Change Loy. Gaussiananything: Interactive point cloud latent diffusion for 3d generation. In *ICLR*, 2025. 1, 2, 6, 7

[16] Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *CVPR*, pages 12642–12651, 2023. 1, 2

[17] Peng Li, Yuan Liu, Xiaoxiao Long, Feihu Zhang, Cheng Lin, Mengfei Li, Xingqun Qi, Shanghang Zhang, Wei Xue, Wenhan Luo, et al. Era3d: High-resolution multiview diffusion using efficient row-wise attention. In *NeurIPS*, 2024. 6, 7

[18] Weiyu Li, Jiarui Liu, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman: High-fidelity mesh generation with 3d native generation and interactive geometry refiner. *arXiv preprint arXiv:2405.14979*, 2024. 1, 2

[19] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, pages 9298–9309, 2023. 2

[20] Zhen Liu, Yao Feng, Michael J. Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. Meshdiffusion: Score-based generative 3d mesh modeling. In *ICLR*, 2023. 1, 2

[21] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[22] 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. 1, 2

[23] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 4

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

[25] Barbara Roessle, Norman Müller, Lorenzo Porzi, Samuel Rota Bulò, Peter Kontschieder, Angela Dai, and Matthias Nießner. L3dg: Latent 3d gaussian diffusion. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024. 2

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image syn-

thesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 2, 8

[27] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *NeurIPS*, 35:25278–25294, 2022. 2

[28] 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. *arXiv preprint arXiv:2310.15110*, 2023. 2, 3, 6, 7

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

[30] 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. In *ECCV*, pages 1–18. Springer, 2024. 6, 7

[31] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In *ECCV*, pages 439–457. Springer, 2024. 2, 3, 6, 7

[32] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *AAAI*, 2023. 6

[33] Peng Wang and Yichun Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023. 2, 3

[34] 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. *NeurIPS*, 36, 2024. 2

[35] 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. In *ECCV*, pages 1–20. Springer, 2024. 4

[36] Haibo Yang, Yang Chen, Yingwei Pan, Ting Yao, Zhineng Chen, Chong-Wah Ngo, and Tao Mei. Hi3d: Pursuing high-resolution image-to-3d generation with video diffusion models. In *ACM MM*, pages 6870–6879, 2024. 2, 3

[37] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. In *NeurIPS*, 2022. 1, 2

[38] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *arXiv preprint arXiv:2406.13897*, 2024. 1, 2

[39] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *CVPR*, pages 10324–10335, 2024. 2, 6, 7