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.

Generative Gaussian Splatting: Generating 3D Scenes with Video Diffusion Priors

Katja Schwarz Norman Müller Peter Kontschieder

Meta Reality Labs Zurich, Switzerland

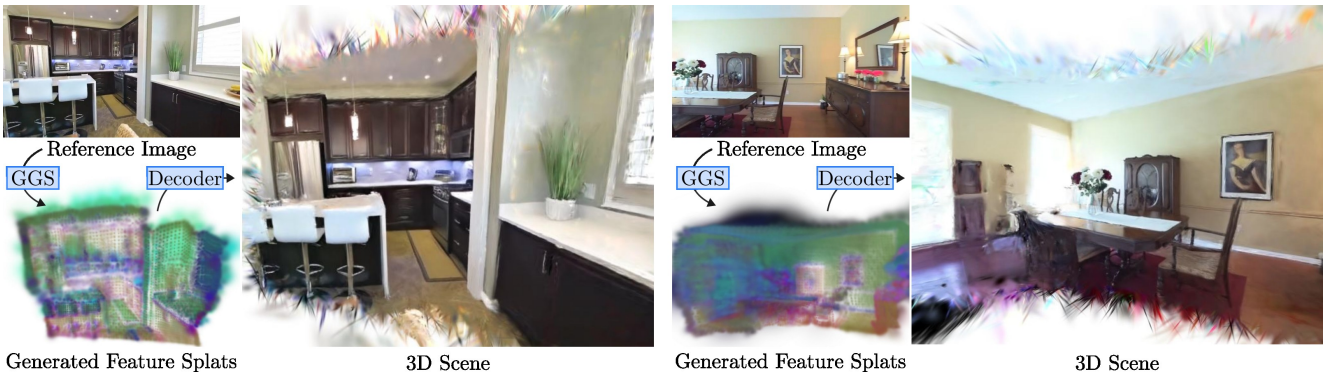


Figure 1. **Overview:** Given one or more input images, GGS leverages a video diffusion prior to directly generate a 3D radiance field parameterized via 3D Gaussian primitives. GGS first generates a feature field with a pose-conditional diffusion model and subsequently decodes the feature splats, yielding an explicit 3D representation of the generated scene. Project page: https://katjaschwarz.github.io/ggs/

Abstract

Synthesizing consistent and photorealistic 3D scenes is an open problem in computer vision. Video diffusion models generate impressive videos but cannot directly synthesize 3D representations, i.e., lack 3D consistency in the generated sequences. In addition, directly training generative 3D models is challenging due to a lack of 3D training data at scale. In this work, we present Generative Gaussian Splatting (GGS) – a novel approach that integrates a 3D representation with a pre-trained latent video diffusion model. Specifically, our model synthesizes a feature field parameterized via 3D Gaussian primitives. The feature field is then either rendered to feature maps and decoded into multi-view images, or directly upsampled into a 3D radiance field. We evaluate our approach on two common benchmark datasets for scene synthesis, RealEstate10K and ScanNet++, and find that our proposed GGS model significantly improves both the 3D consistency of the generated multi-view images, and the quality of the generated 3D scenes over all relevant baselines. Compared to a similar model without 3D representation, GGS improves FID on the generated 3D scenes by ∼20% on both RealEstate10K and ScanNet++.

1. Introduction

Diffusion models work remarkably well for generating photorealistic images and videos from noise when trained on vast amounts of data. For 3D scenes, however, 3D training data is scarce and generative models fall short in terms of quality and generalization ability compared to their 2D counterparts. As a consequence, some recent works leverage pre-trained video diffusion models as backbones to first generate multi-view images, and subsequently stitch the generated views together using 3D reconstruction algorithms [13, 70, 76]. However, the generated multi-view images often lack 3D consistency, requiring carefully tailored 3D reconstruction algorithms [13, 70] or time consuming iterative procedures [76].

Another line of works improve 3D consistency of the generated images by using a 3D representation within the diffusion model [1, 4, 60]. However, these works cannot leverage pre-trained video diffusion models, because of their custom network architectures for incorporating the 3D representation. In this work, we investigate how 3D representations can be directly integrated with powerful video diffusion priors to improve the consistency of the generated images and thereby the generated 3D scenes. One

challenge is that state-of-the-art diffusion models operate on a compressed latent space, which is spatially approximately aligned with the input images but itself is not 3D-consistent. Another challenge is that predicting noise instead of the denoised input in practice works better and is the de-facto standard in video diffusion models. However, when including a 3D representation into diffusion models, this representation should mirror the denoised input, i.e. the 3D scene, and cannot directly model the statistically independent noise added to the input images.

We find that we can solve both aforementioned issues by learning a 3D representation in feature space instead of latent space (see Figure 1). In particular, our Generative Gaussian Splatting (GGS) approach renders feature maps from the 3D representation that are subsequently decoded to latent space. This allows for enough flexibility to train with v-prediction while enabling the training procedure to generate a meaningful 3D representation. Since this refinement can again introduce inconsistencies in the generated images, we additionally propose a decoder that directly predicts a decoded 3D scene from the generated feature maps. Our experiments demonstrate that including a 3D representation indeed substantially improves the consistency across generated images. Similarly, our generated 3D scenes are inherently 3D-consistent, and their quality can be further refined by e.g. running a few iterations of a standard 3D reconstruction algorithm together with our generated 2D images. Compared to a similar model without 3D representation, GGS improves FID on the generated 3D scenes by $\sim$20%, see Table 3. Another interesting property of our approach is that using an explicit 3D representation like Gaussian splats supports training with additional depth supervision where available, resulting in additional improvements, particularly in terms of consistency. We summarize our main contributions as follows:

- We propose an approach that directly integrates an explicit 3D representation with a pre-trained latent video diffusion backbone, thereby improving 3D consistency of the generated image sequences and allowing for training with additional depth supervision where available.
- We design a custom decoder that directly predicts the decoded 3D representation of the scene from the generated feature maps.
- We train a conditional variant of our model that autoregressively generates full scenes from an arbitrary number of input views.

2. Related Works

Regression-Based Models for Novel View Synthesis. Recently, multiple works consider NVS from sparse input views, leveraging priors learned across many training scenes [5–7, 10, 12, 16, 21, 22, 26, 37, 38, 42, 44, 53, 58, 61, 64, 69, 73, 79, 84]. PixelSplat [5], Splatter Image [58], and GS-LRM [79] directly predict per-pixel Gaussian splats. All of these methods optimize a regression objective, and hence cannot perform longer-range extrapolations. Instead, we consider a generative model to enable view extrapolation. Similarly, LatentSplat [69] addresses this limitation by extending PixelSplat with a GAN-based decoder, enabling moderate-distance extrapolations. In contrast to LatentSplat, our approach leverages a strong generative backbone and thereby supports larger view extrapolation.

3D Generative Models. A few works directly apply generative models to 3D representations [32, 78, 81] but the lack of 3D training data makes it difficult to train such models at large scale. Thus, many works combine an intermediate 3D representation with differentiable rendering and train with posed images instead of 3D data [1, 2, 4, 25, 36, 41, 47–49, 60]. However, compared to images and video data, posed multi-view images are still difficult to obtain at large scale. Consequently, 3D generative models do not yet match in quality and generalization ability wrt. to their 2D counterparts. The aforementioned methods require custom architectures for incorporating the 3D representation and can hence not leverage pre-trained backbones. In contrast, we combine a 3D representation with a pre-trained video diffusion model that acts as a powerful prior. Concurrently, Prometheus [77] trains a text-to-3D diffusion model by learning to denoise depth and multi-view images jointly.

Pose-Conditional Image and Video Diffusion. Diffusion Models (DMs) [18, 54, 56] achieve state-of-the-art results in text- and image-guided synthesis [9, 11, 19, 20, 33, 35, 39, 43, 52]. Recent works propose pose-conditional variants [13, 14, 27–30, 40, 46, 50, 51, 57, 63, 66–68, 70, 71, 75, 76, 83]. PNVS [75] trains a 2D diffusion model to autoregressively generate frames along a given camera trajectory. MultiDiff [34] conditions the DM on depth-based warped images. More recently, CAT3D [13], CamCo [71], CameraCtrl [14], and Wonderland [27] condition pre-trained video models on poses parameterized by Plücker coordinates. 4DiM trains a pixel-based DM conditioned on both camera pose and time. ViewCrafter [76] proposes a point-cloud conditioned DM, leveraging DUSt3R [65] as powerful geometric prior. While pre-trained diffusion backbones enable generalization and high image fidelity, pose-conditional DMs often struggle with generating multi-view consistent images. Some approaches address these inconsistencies with carefully tuned reconstruction algorithms [13, 70] or complicated iterative approaches [76] to subsequently distill the generated images into a 3D representation. Instead, our GGS directly generates a 3D representation, which can optionally be further refined with out-of-the-box reconstruction algorithms.

3. Method

We introduce Generative Gaussian Splatting (GGS) which directly synthesizes 3D-consistent scenes from one or more posed reference images. Key to our method is the integration of an explicit 3D representation with a pre-trained video diffusion model (Sec. 3.2). Fig. 2 shows an overview. In alignment with most recent works, we consider a pre-trained U-Net diffusion model [14, 66, 76]. The video model was trained with v-prediction, and conditioned on a single input image by concatenation of the reference latent to the input sequence, as proposed in [3].

3.1. Pose-Conditional Image-To-Video Architecture

We condition the pre-trained video diffusion backbone on both camera poses and multiple reference images to enable pose-conditional view synthesis. Consider a set of posed images $\{\mathbf{I}^m, \mathbf{p}^m\}$, comprising K reference images and L target images, where $K + L = M$. These images come with known camera extrinsics and intrinsics $\{\mathbf{p}^m\}$. All images are encoded into latent space. We apply the forward diffusion process only to the target views, yielding a set of noise-free reference latents $\{\mathbf{z}^k_{ref,0}\}_{k=1}^K$ and noisy target latents $\{\mathbf{z}^l_{tgt,t}\}_{l=1}^L$. To condition the model on multiple reference images, we concatenate the reference latents channelwise to the input sequence, setting their noise level to zero, i.e., $\{[\mathbf{z}^k_{ref,0}, \mathbf{z}^k_{ref,0}]\}_k$. For the target latents, we concatenate zeros $\{[\mathbf{z}^l_{tgt,t}, \mathbf{0}]\}_l$, see Fig. 2 (left). For brevity, we denote this input sequence as $\mathbf{z}^m$. Utilizing the EDM-preconditioning framework [23], we approximate the denoising process with a neural network D_θ, parameterized as follows:

$$D_\theta(\mathbf{z}^m; \sigma_t, \{\mathbf{z}^k_{\text{ref}}\}, \{\mathbf{p}^m\}) = c_{\text{skip}}(\sigma_t)\mathbf{z}^m + c_{\text{out}}(\sigma_t)F_\theta\left(c_{\text{in}}(\sigma_t)\mathbf{z}^m; c_{\text{noise}}(\sigma_t), \{\mathbf{z}^k_{\text{ref}}\}, \{\mathbf{p}^m\}\right), \tag{1}$$

with preconditioning weights $c_{\text{skip}}(\sigma_t)$, $c_{\text{out}}(\sigma_t)$, $c_{\text{in}}(\sigma_t)$, and $c_{\text{noise}}(\sigma_t)$. See Appendix A.1 for more details. To incorporate camera poses into the backbone, we adopt the approach from CameraCtrl [14], integrating the video model with a camera encoder $\mathcal{P}$. The camera encoder processes the Plücker embeddings $\{\mathbf{P}^m\}$ of the poses $\{\mathbf{p}^m\}$ and outputs multi-scale camera embeddings, which are then used to condition the diffusion model.

3.2. Integrating 3D Constraints

By conditioning the video backbone on multiple reference images and Plücker embeddings, our model can generate new images along a specified camera trajectory. However, it cannot directly generate 3D scenes, and the resulting images lack the consistency needed for high-quality 3D reconstruction through per-scene optimization, as shown in Table 3. To address this limitation, we introduce a stronger bias in the model to learn correct spatial relationships between frames. Specifically, we integrate a 3D representation that correlates features through depth-based reprojection. We choose Gaussian splats for its fast rendering and numerical stability during training.

A key challenge is determining where to best integrate the 3D representation within the diffusion model. The model's compressed latent space may not inherently preserve 3D consistency. Therefore, imposing 3D constraints directly within this latent space may not be effective. Our experiments confirm that predicting splats in latent space does not yield good results, as seen in Table 4. Instead, we integrate the 3D representation into the final upsampling block of the U-Net. Specifically, an epipolar transformer $\mathcal{T}_{\text{epi}}$ processes the inputs to the last upsampling block, followed by two prediction heads, f_ϕ and f_g, to generate the parameters of Gaussian splats, as illustrated in Fig. 2 (middle). Similarly to PixelSplat [5], we use the epipolar transformer to correlate features along epipolar lines via attention. To maintain a manageable memory footprint, we aggregate features from the two neighboring views for each input view.

Using f_ϕ and f_g, the splats are parameterized as

$$\mu_k = f_\phi(\mathbf{f}_{\text{epi}}) \tag{2}$$

$$[\mathbf{\Sigma}_k, \alpha_k, \mathbf{c}_k, \mathbf{f}_k] = f_g(\mathbf{f}_{\text{epi}}), \tag{3}$$

where $\mathbf{f}_{\text{epi}}$ denotes the feature map predicted by $\mathcal{T}_{\text{epi}}$, and μ_k, $\mathbf{\Sigma}_k$, α_k, $\mathbf{c}_k$, $\mathbf{f}_k$ are mean, covariance, opacity, color and feature values of the k-th splat, respectively. The splats are then rendered into low resolution images $\hat{\mathbf{I}}^m_{\text{LR}}$ and feature maps $\mathbf{f}^m$ using

$$\mathbf{f}^{\mathbf{m}}(x) = \sum_{k=1}^{K} \mathbf{f}_k \alpha_k \mathbf{g}_{2D,k}(x; \mathbf{p}^m) \prod_{j=1}^{k-1} (1 - \alpha_j \mathbf{g}_{2D,j}(x; \mathbf{p}^m)), \tag{4}$$

where $\mathbf{g}_{2D,k}(x; \mathbf{p}^m)$ is the 2D projection of the k-th splat to camera pose $\mathbf{p}^m$. As shown in Fig. 2 (right), the feature maps are further refined by a block f_v with skip connections to the input. f_v outputs $\hat{\mathbf{v}}^m$, i.e., the weighted sum of predicted noise and sample used in the v-prediction objective.

Objective. We follow the training procedure and hyperparameter choices of our internal diffusion backbone, and train the model with v-prediction [23, 45]. To better regularize the 3D representation, we add reconstruction losses $\mathcal{L}_{LR}$ and $\mathcal{L}_{nv,LR}$ on the rendered low-resolution images of the input views $\{\hat{\mathbf{I}}^m_{LR}\}$ and J additional novel viewpoints $\{\hat{\mathbf{I}}^j_{nv,LR}\}_{j=1}^J$.

$$\mathcal{L}_{LR} = \frac{1}{M}\sum_{m=1}^{M} ||\hat{\mathbf{I}}^m_{\text{LR}} - \mathbf{I}^m_{\text{LR}}||_2^2 \tag{5}$$

$$\mathcal{L}_{nv,LR} = \frac{1}{J}\sum_{j=1}^{J} ||\hat{\mathbf{I}}^j_{nv,LR} - \mathbf{I}^j_{nv,LR}||_2^2. \tag{6}$$

Note that these novel views are directly rendered from the 3D representation and do not need to go through the network. Hence, this additional guidance comes at marginal

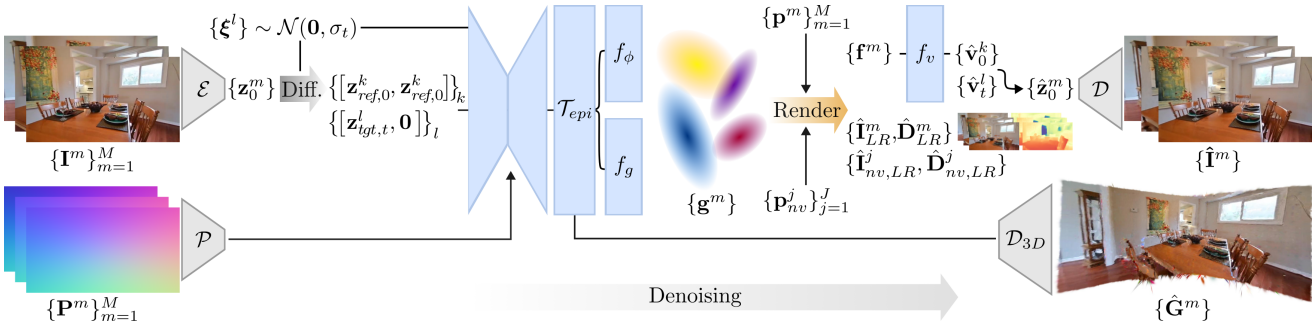


Figure 2. **Model Architecture:** Our approach, GGS, directly synthesizes a 3D representation, which is parameterized by a set of Gaussian splats $\{\mathbf{g}^m\}$, from a set of posed input images. Specifically, during training we consider a set of posed images $\{\mathbf{I}^m\}$ with associated camera poses $\{\mathbf{p}^m\}$ and corresponding Plücker embeddings $\{\mathbf{P}^m\}$. The images are first encoded into a latent representation $\{\mathbf{z}_0^m\}$, which is then partitioned into K reference images and L target images. We introduce noise only to the latents of the target images $\{\mathbf{z}_{tgt,0}^l\}_{l=1}^L$, while leaving the reference images noise-free. To ensure compatibility with the pre-trained image-to-video diffusion model, we duplicate the reference latents across the channel dimension and concatenate zeros for the target latents. The resulting latents, along with the noise level σ_t and Plücker embeddings, are fed into a U-Net architecture that produces intermediate per-latent feature maps. These feature maps are subsequently processed by an epipolar transformer $\mathcal{T}_{epi}$ to predict the parameters of the Gaussian feature splats $\{\mathbf{g}^m\}$. We render both feature maps $\{\mathbf{f}^m\}$ and low-resolution images $\{\mathbf{I}_{LR}^m\}$ for the input views, as well as low-resolution images for J novel views $\{\mathbf{I}_{nv,LR}^j\}_{j=1}^J$ to regularize the 3D representation. Finally, the rendered feature maps are decoded into a weighted combination of sample noise ξ^m and input latent to predict the noise-free latents $\{\hat{\mathbf{z}}_0^m\}$.

computational cost while improving the results considerably, as shown in Table 4.

By incorporating an explicit 3D representation, our model is inherently equipped to leverage depth supervision. Consequently, when depth is available, we train with an additional depth loss, $\mathcal{L}_d$. This loss function minimizes the Euclidean distance between the predicted mean, μ_k, of each per-pixel splat and its corresponding ground truth 3D coordinate. The ground truth is obtained by unprojecting the pixel coordinate k with its corresponding depth d_k and camera pose $\mathbf{p}^{m(k)}$

$$\mathcal{L}_d = \frac{1}{K} \sum_{k=1}^{k} ||\mu_k - \text{unproject}(d_k, \mathbf{p}^{m(k)})||_2. \quad (7)$$

In PixelSplat, f_ϕ predicts a discrete probability density over a set of depth buckets and makes sampling differentiable by setting the opacity according to the probability of the sampled depth. However, in our setting, we find that this does not work well in conjunction with additional depth supervision. Instead, we approximate depth with a Gaussian distribution, i.e., by predicting a mean and covariance and using the reparameterization trick to differentiably sample from it. We validate this design choice in Table 4, and qualitatively observe that this significantly improves the predicted depth.

3.3. Decoding Latent Gaussian Splats

So far, our model can synthesize a sequence of images from a set of reference images. However, due to the refinement block f_v and the decoder $\mathcal{D}$, the generated results can still become inconsistent. We address this issue by adding a 3D decoder $\mathcal{D}_{3D}$ that maps the intermediate feature maps of the epipolar transformer directly to Gaussian splats in image space. The decoder first increases the resolution of the input feature maps with a 2D upsampler. Next, two blocks similar to f_ϕ and f_g predict per-pixel splats $\{\hat{\mathbf{G}}^m\}$ in image space. For memory efficiency, we freeze the diffusion model and train the 3D decoder at a fixed timestep $t = 0$, i.e., we do not add noise to any of the inputs, effectively resulting in a novel view synthesis setting. The decoder is trained in a second stage, using the same losses as in Eq. (5) and Eq. (6). Our model can thus predict per-pixel splats to parameterize the full scene in image space. However, such a per-pixel representation highly overparameterizes the scene and does not allow for an adaptive resolution. Indeed, we observe that the generated splats are very small. To obtain a more compact representation, we optionally apply a per-scene optimization. As our goal is not to improve the method for 3D reconstruction itself, we rely on Splatfacto's standard reconstruction algorithm for this task. We initialize the scene with the predicted splats from our model and perform 5K iterations of standard 3DGS optimization by running Splatfacto from [59], using the decoded images from the 2D decoder. In contrast to distillation approaches, our model only requires a few steps of per-scene optimization due to the good initialization with the predicted splats and works with out-of-the-box reconstruction algorithms without careful tuning.

3.4. Splat Conditional Model

While our model can be conditioned on any number of reference images, two views are generally sufficient for achieving high-quality view interpolations. For generating scenes from more than two reference views, we can sim-

ply chain the results together. To ensure consistent content across generated batches, we condition the model on renderings of the current 3D scene, similar to the approach in [34], but without relying on pre-trained monocular priors. Specifically, we utilize Gaussian splats in feature space and replace their feature values with the corresponding predicted image latents to align with our image conditioning strategy. The resulting 3D representation is rendered from the target views, and the outputs are concatenated channel-wise with the noisy image latents. We train our conditional model with a mixture of datasets, some of which include ground truth depth. When available, we use the depth information from the reference images along with their latents to approximate the conditioning signal. We construct per-pixel splats and adjust the scale so that the re-rendered splats occupy one pixel per view. Since our model also predicts per-pixel splats of relatively small size during inference, this method of approximating the 3D representation proves effective for conditional training.

4. Experiments

Datasets. For training, we utilize three datasets of static indoor environments: RealEstate10K [82], ScanNet [8], and an internal large-scale dataset of synthetic environments. These datasets provide video sequences with registered camera parameters, and both ScanNet and our internal dataset include metric depth maps. RealEstate10K comprises sequences of approximately 30-100 frames from 10,000 real estate recordings, featuring smooth camera trajectories with minimal roll or pitch. Following [68], we also use a variant of RealEstate10K with rescaled camera poses to be approximately metric. ScanNet includes 1,513 handheld captures of indoor environments, with camera trajectories that follow a scan pattern, often exhibiting rapid changes and varied orientations. Our internal dataset contains around 95,000 synthetic indoor environments with smooth camera trajectories achieved through spline interpolation. In line with previous work, we benchmark our approach on RealEstate10K. Additionally, we evaluate on ScanNet++ [72], which consists of 460 indoor scenes. Despite the similar name, ScanNet++ features different cameras and scenes from ScanNet, allowing us to assess the generalization of our method in real-world scenarios.

Experimental Settings. In Sec. 4.1, we consider scene synthesis from a single image. Sec. 4.2 analyzes scene reconstruction and synthesis from two reference images. For the single-view setting, we subsample 8 frames for RealEstate10K with a stride of 10, similar to [68, 74], and use a stride of 4 for ScanNet, since camera trajectories in ScanNet feature stronger motion. The first image of the sequence is assigned as reference image. During training, we randomly flip the order of the camera trajectories and additionally sample 6 novel views between the target views, but otherwise apply the same sampling strategy for training and evaluation. We report results on 128 randomly selected scenes from the RealEstate10K testset and 50 scenes from ScanNet++. Since the camera trajectories in ScanNet++ can feature extremely large changes in camera motion, we select target views that have a decent overlap with the reference images, see Appendix D for details.

For the two-view conditional model, we comply with the training strategy of PixelSplat [5] and sample 8 views randomly within a maximum gap of 80 frames for RealEstate10K, and 32 frames for ScanNet. During training, we select the first and another randomly selected view as reference images and additionally sample 6 novel views between the target views. Given two reference images, we evaluate performance separately on view interpolation and view extrapolation. For view interpolation, the reference images are the first and last frame of the sequence, which is the common setting in regression-based NVS methods. For view extrapolation, the reference frames are the first two frames of the sequence and the model needs to extrapolate the remaining 6 views, making this a generative task.

Evaluation Metrics. We report metrics for image quality and 3D consistency. Peak Signal-to-Noise Ratio and LPIPS [80] quantify reconstruction quality. To measure the overall image fidelity of the generated images, we compute FID [17] and FVD [62], for more details see Appendix D. Furthermore, we evaluate the 3D consistency of our approach using the Thresholded Symmetric Epipolar Distance (TSED), which measures the alignment between pairs of views by computing the symmetric epipolar distance between corresponding points in the two views [74]. We report TSED scores with a threshold of 2.0. As some of the baselines can only produce results at resolution 256×256 pixels, we report quantitative results at this resolution unless denoted otherwise. For GGS, we report numbers on generated images using the 2D decoder.

Baselines. We compare GGS to the strongest approaches that have code available. Note that all of the baselines make use of pre-trained backbones. While this enables all methods to generalize well to ScanNet++, it can also make a direct comparison difficult, especially when diffusion priors vary in quality. For a fair comparison of a model with and without an intermediate 3D representation, we train our own purely pose-conditional model (Ours-No3D) as described in Sec. 3.1. Note that the video backbones of CameraCtrl[14] and ViewCrafter [76] require 14, and 25 frames, respectively, while we consider an 8-frame setting. We evaluate these baselines by padding the output camera trajectory and subsequently subsampling the generated results.

Implementation Details. We use a proprietary pre-trained image-2-video diffusion model with $\sim$ 2B parameters and

	RE10K					Scannet++				
	PSNR↑	LPIPS↓	FID↓	FVD↓	TSED↑	PSNR↑	LPIPS↓	FID↓	FVD↓	TSED↑
PNVS	16.7	0.385	**37.6**	443.3	0.925	13.5	0.516	102.9	498.0	0.362
MultiDiff	16.4	0.371	43.7	443.3	0.943	14.3	0.440	144.9	1130.2	0.683
CameraCtrl	17.7	0.332	46.2	366.7	0.885	15.9	0.399	82.0	516.6	0.500
ViewCrafter	16.6	0.312	44.2	313.2	0.947	18.3	**0.280**	**74.5**	437.5	0.871
Ours-No3D	19.1	**0.272**	45.9	268.6	0.984	17.1	0.351	80.4	**436.4**	0.596
GGS (Ours)	**19.2**	0.277	46.4	**267.4**	0.992	18.0	0.317	81.8	455.5	0.896
GGS (Ours)+depth	**19.2**	0.273	47.0	273.5	**0.999**	**18.4**	0.309	81.9	441.9	**0.899**

Table 1. **Baseline Comparison Given One Reference Image:** We benchmark the approaches on RealEstate10K and evaluate generalization ability on ScanNet++. The reported metrics are calculated on all generated images.

a U-Net architecture. The camera encoder and 3D decoder have ∼200M and 1.5M parameters, respectively. Our models were trained on 8 Nvidia A100 80GB GPUs with a batch size of 1 per GPU, using the AdamW optimizer [31] with a learning rate of 3×10^{-5}. Starting from a pre-trained I2V diffusion model, we first finetune its temporal attention layers together with the camera encoder for 100K iterations. Afterwards, we freeze the parameters of the camera encoder for memory efficiency, integrate the 3D representation, and finetune the full model. The ablation studies are reported after training the models for 75K iterations. Our final models were trained for 800K iterations, taking approximately 2 weeks. Inference is performed with a discrete Euler scheduler using 30 steps. The monocular depth predictor samples 3 Gaussian splats per pixel, unless we train with depth supervision, where we only sample a single splat.

4.1. Scene Synthesis From a Single Image

Table 1 compares GGS to prior work on single-image to scene synthesis. On RealEstate10K, our approach significantly improves image quality and 3D consistency over the baselines. On ScanNet++, ViewCrafter generates images with slightly higher fidelity but less consistency as indicated by a lower TSED. Note that ViewCrafter should be compared to GGS +depth, since it relies on DUSt3R [65], which was trained with ground truth depth. Interestingly, on ScanNet++, we observe large improvements in TSED for including a 3D representation in our model. We note that ScanNet++ contains trajectories that go back and forth in a scene. In these cases, Ours-No3D tends to generate inconsistent content, while with 3D representation, the model can generate consistent frames. We show qualitative examples in Fig. 3. Overall, we observe that the generated images from CameraCtrl tend to be blurry in the generated areas. ViewCrafter generates visually plausible views but can alter the colors and sometimes deforms objects, see e.g. the screen in the ScanNet++ example in Fig. 3.

4.2. Scene Synthesis From Two Images

We report results for two tasks: view interpolation and view extrapolation. For view interpolation, we also include the purely reconstruction-based PixelSplat [5] for reference in Table 2. Both, PixelSplat and LatentSplat perform well

	RE10K						Scannet++					
	Interpolation			Extrapolation			Interpolation			Extrapolation		
	PSNR↑	LPIPS↓	TSED↑	FID↓	FVD↓	TSED↑	PSNR↑	LPIPS↓	TSED↑	FID↓	FVD↓	TSED↑
PixelSplat	**23.9**	**0.170**	**1.000**	–	–	–	21.2	0.228	**0.949**	–	–	–
LatentSplat	22.1	0.195	**1.000**	59.8	428.6	0.957	**21.7**	0.274	0.919	131.4	638.4	0.840
ViewCrafter	19.9	0.230	0.898	**48.5**	320.0	0.900	20.4	**0.218**	0.838	**81.0**	**473.2**	0.844
Ours-No3D	21.6	0.214	0.916	52.7	306.2	0.892	18.0	0.317	0.532	87.5	511.0	0.595
GGS (Ours)	22.9	0.204	0.970	52.6	324.9	0.974	19.3	0.303	0.747	98.6	497.0	0.794
GGS (Ours)+depth	23.7	0.189	0.983	49.4	**293.8**	**0.990**	20.2	0.266	0.843	85.6	476.6	**0.854**

Table 2. **Baseline Comparison Given Two Reference Images:** We benchmark the approaches on RealEstate10K and evaluate generalization on ScanNet++. We report PSNR, LPIPS and TSED for view interpolation, and FID, FVD and TSED for view extrapolation. The metrics are calculated on all generated images.

on RealEstate10K and ScanNet++ for view interpolation. Our approach achieves similar results on RealEstate10K but does not reach the same reconstruction quality on ScanNet++. However, PixelSplat does not support view extrapolation, which is our primary objective. The generative decoder of LatentSplat also struggles to generate realistic views far from the reference images, reflecting in overall worse extrapolation metrics and visible artifacts, see also Fig. 4. Similar to the single image results, ViewCrafter performs particularly well on ScanNet++ but lacks 3D consistency as indicated by a lower TSED compared to GGS +depth in all settings. Qualitatively, we observe that on ScanNet++, ViewCrafter can struggle to match the viewpoint correctly, as visible in Fig. 4.

So far, we analyzed the quality and consistency of the generated output frames. But ultimately, our goal is to generate high-quality 3D scenes. We hence conduct an additional experiment in which we reconstruct a 3D scene from the generated outputs of the strongest generative baseline, ViewCrafter, and our approach. Note that ViewCrafter also proposes an iterative view synthesis strategy with a content-adaptive camera trajectory planning algorithm for long-range view extrapolation. As this is orthogonal to our approach, we evaluate the quality of the outputs from a single forward pass. However, we remark that our approach could benefit from using the same strategy for long-view extrapolation. For ViewCrafter, we run an off-the-shelf 3DGS optimization, i.e. train a Splatfacto model from [59] for 15,000 iterations on the generated multi-view images. For our method, we directly consider the generated 3D splats predicted by GGS. Additionally, we train a refined variant, for which we initialize Splatfacto with the generated splats and run it for 5,000 iterations per scene to obtain a more compact 3D scene representation. We run the per-scene optimization on all evaluation scenes and assess the quality of the resulting 3D representations by calculating FID and FVD on rendered images from viewpoints interpolated between the training views. The qualitative and quantitative results in Fig. 5 and Table 3 highlight the importance of including a 3D representation in the generative model. While the individual frames of ViewCrafter are of high quality, slight inconsistencies in the generated

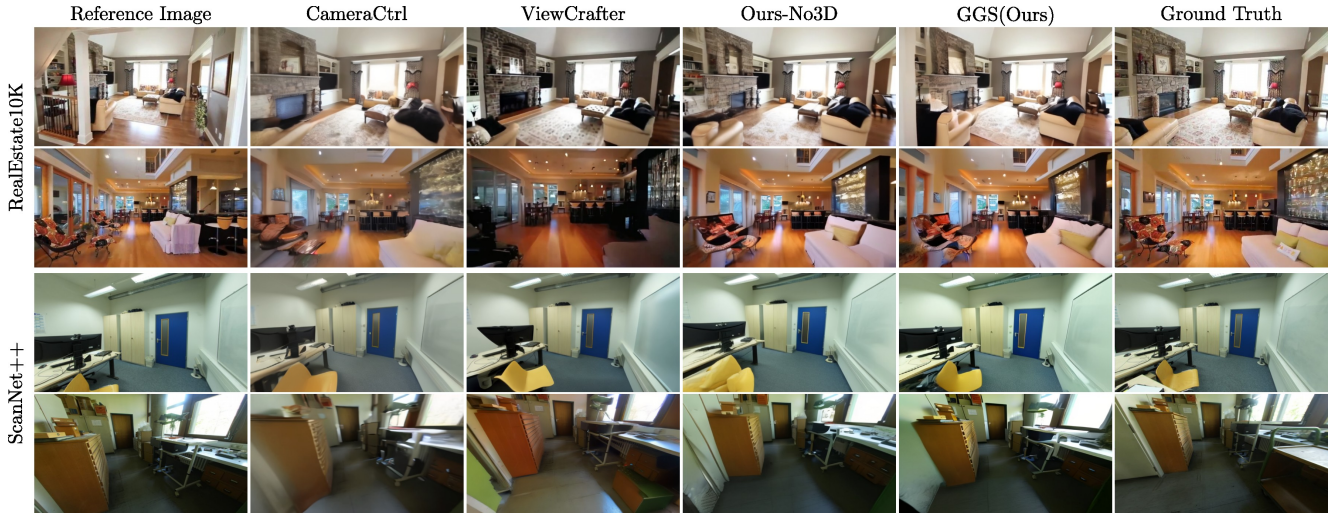


Figure 3. **Baseline Comparison Given One Reference Image:** We show results for the strongest baselines CameraCtrl [15] and ViewCrafter[76] together with our approach without (Ours-No3D) and with 3D representation (GGS). Best viewed zoomed in.



Figure 4. **Baseline Comparison For View Extrapolation Given Two Reference Images:** We show results for the strongest baselines LatentSplat [69] and ViewCrafter[76] together with our approach without (Ours-No3D) and with 3D representation (GGS). As both reference views are close together, we only include one image for reference. Best viewed zoomed in.

	RE10K		ScanNet++	
	FID↓	FVD↓	FID↓	FVD↓
ViewCrafter	96.1	1071.8	140.6	616.4
Ours-No3D	57.8	582.0	146.2	620.1
GGS (Ours)+depth	56.3	498.3	127.4	587.9
GGS (Ours)+depth+FT	**47.3**	**403.9**	**111.0**	**548.0**

Table 3. **Single Image to 3D:** FID and FVD scores for rendered views between the generated images at 576×320 pixels.

sequence lead to clearly visible artifacts in the 3D reconstruction.

4.3. Autoregressive Scene Synthesis

To extend our model from two to an arbitrary number of input views, we train a conditional variant to autoregressively generate a full scene (Sec. 3.4). At inference, we first generate multi-view images between the first two reference images, as well as an initial 3D representation of the scene. Next, we re-render the generated scene as condition for the next step, biasing the model towards 3D-consistent inpainting. We proceed by using the second and third reference images together with the rendered condition as input to our model, updating the 3D representation after each step. We continue with this strategy for all available reference images. In Fig. 6, we show a reconstructed 3D scene from GGS using only 5 reference images. For this example, we generate images between the reference images and further consider camera viewpoints with additional positive and negative pitch to obtain a more complete reconstruction. By improving 3D consistency of the generated images,

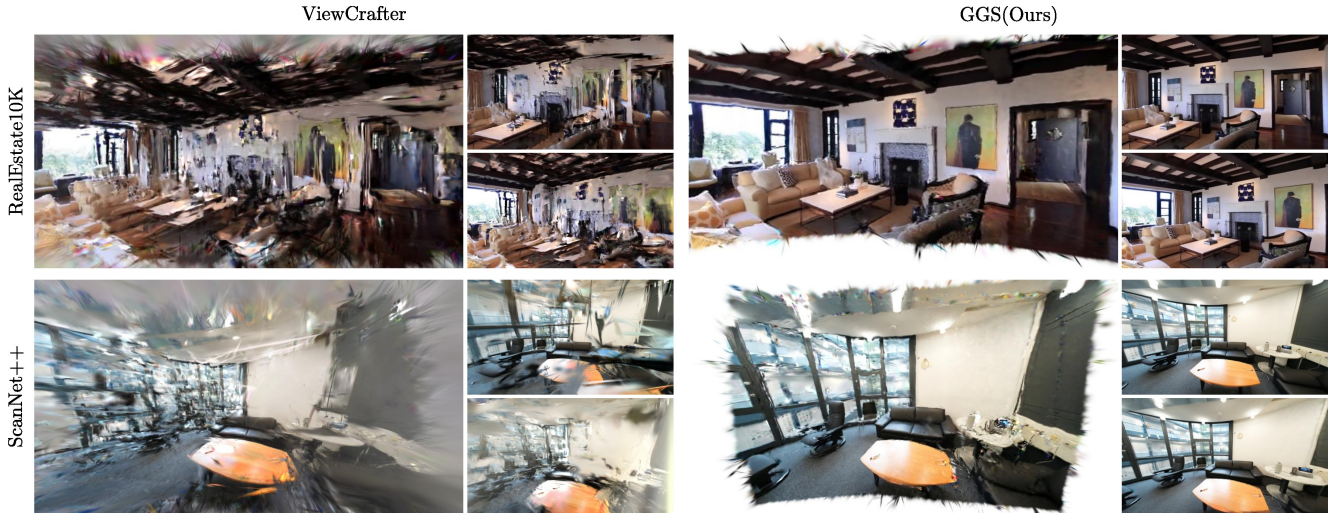


Figure 5. **3D Reconstruction Results From Generated Images:** We run an off-the-shelf 3DGS optimization on the generated multi-view images of ViewCrafter and GGS (Ours). For ViewCrafter, we use 15,000 optimization steps. For our approach, we only refine the generated splats with the generated multi-view images, using 5,000 iterations. The resulting 3D representation is shown on the left and two rendered views from novel viewpoints are included on the right.



Figure 6. **Autoregressive Scene Synthesis with GGS:** By generating consistent views between the reference images and from additional viewpoints, GGS can augment the set of 5 reference images and generate larger 3D scenes autoregressively.

our approach enables reconstructing a reasonable 3D scene from very sparse observations. With improved consistency, floating artifacts in the reconstructions are significantly reduced (see also Fig. 5), demonstrating the effectiveness of our approach. While some imperfections remain, we attribute them to the inherent sensitivity of 3D reconstruction, which tends to amplify even minor inconsistencies. For results on a longer trajectory, see Appendix E.3.

4.4. Ablation Studies

We ablate our model choices from Sec. 3 in the two-view conditional setting on RealEstate10K in Table 4. Without the loss on novel viewpoints (No $\mathcal{L}_{nv}$), image quality (PSNR) and consistency (TSED) drop significantly. Predicting the 3D representation in an intermediate feature space together with v-prediction clearly outperforms a 3D representation in latent space and training with x_0 prediction (No f_v, x_0-prediction). Compared to predicting a discrete probability distribution ϕ for the depths, our approximation with a Gaussian distribution works better when depth supervision is available and improves all metrics. Lastly, we compare two architectures for the 2D upsampler in the 3D decoder $\mathcal{D}_{3D}$: a convolutional and a transformer-based upsampler, finding that the convolutional architecture performs better. Compared to the 2D decoder (GGS), the 3D decoder improves 3D consistency but moderately degrades visual fidelity.

	Interpolation			Extrapolation		
	PSNR↑	LPIPS↓	TSED↑	FID↓	FVD↓	TSED↑
GGS	22.5	0.214	0.977	51.3	317.0	0.972
No $\mathcal{L}_{\text{nv,LR}}$	19.6	0.294	0.868	65.8	320.8	0.865
No f_v, x_0-prediction	14.9	0.611	0.753	239.4	1131.6	0.821
GGS +depth	23.0	0.204	0.973	51.0	336.3	0.963
f_ϕ: Discrete ϕ	22.0	0.211	0.960	53.7	348.9	0.956
$\mathcal{D}_{3D}$: convolutional	21.2	0.287	0.995	66.9	317.9	0.990
$\mathcal{D}_{3D}$: transformer	21.3	0.342	0.996	92.2	343.5	0.990

Table 4. **Ablation Studies:** We investigate the effectiveness of our design choices on RealEstate10K using two reference images.

5. Conclusion

We introduce GGS, a novel approach for 3D scene synthesis from few input images. It integrates 3D representations with video diffusion priors to improve the consistency of the generated images and enable direct 3D scene synthesis. Our experiments on RealEstate10k and ScanNet++ show that GGS narrows the gap to regression-based methods for view interpolation, while clearly outperforming generative baselines in 3D consistency and image quality. Nonetheless, the most recent regression-based NVS methods yield even higher image fidelity, highlighting the need to close this gap for a unified approach for scene synthesis.

References

[1] Titas Anciukevicius, Fabian Manhardt, Federico Tombari, and Paul Henderson. Denoising diffusion via image-based rendering. In *ICLR*, 2024. 1, 2

[2] Miguel Ángel Bautista, Pengsheng Guo, Samira Abnar, Walter Talbott, Alexander T Toshev, Zhuoyuan Chen, Laurent Dinh, Shuangfei Zhai, Hanlin Goh, Daniel Ulbricht, Afshin Dehghan, and Joshua M. Susskind. GAUDI: A neural architect for immersive 3D scene generation. *arXiv preprint arXiv:2207.13751*, 2022. 2

[3] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv*, 2023. 3

[4] Eric R. Chan, Koki Nagano, Matthew A. Chan, Alexander W. Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. GeNVS: Generative novel view synthesis with 3D-aware diffusion models. In *CVPR*, 2023. 1, 2

[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. 2, 3, 5, 6, 12, 13

[6] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *CVPR*, 2021.

[7] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *ECCV*, 2024. 2

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 5

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 2

[10] Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs. In *CVPR*, 2023. 2

[11] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *ICCV*, 2023. 2

[12] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, Zhangyang Wang, and Yue Wang. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. *arXiv*, 2024. 2

[13] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul P. Srinivasan, Jonathan T. Barron, and Ben Poole. CAT3D: create anything in 3d with multi-view diffusion models. *arXiv*, 2024. 1, 2

[14] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv*, 2024. 2, 3, 5

[15] Xuming He, Richard S. Zemel, and Miguel A. Carreira-Perpinan. Multiscale conditional random fields for image labeling. In *CVPR*, 2004. 7, 15

[16] Philipp Henzler, Jeremy Reizenstein, Patrick Labatut, Roman Shapovalov, Tobias Ritschel, Andrea Vedaldi, and David Novotny. Unsupervised learning of 3d object categories from videos in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4709, 2021. 2

[17] 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. In *NeurIPS*, 2017. 5

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020. 2, 12

[19] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 2

[20] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 2022. 2

[21] 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. 2

[22] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. *arXiv*, 2024. 2

[23] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022. 3, 12

[24] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 12

[25] Seung Wook Kim, Bradley Brown, Kangxue Yin, Karsten Kreis, Katja Schwarz, Daiqing Li, Robin Rombach, Antonio Torralba, and Sanja Fidler. Neuralfield-ldm: Scene generation with hierarchical latent diffusion models. In *CVPR*, 2023. 2

[26] Jonás Kulhánek, Erik Derner, Torsten Sattler, and Robert Babuska. Viewformer: Nerf-free neural rendering from few images using transformers. In *ECCV*, 2022. 2

[27] Hanwen Liang, Junli Cao, Vidit Goel, Guocheng Qian, Sergei Korolev, Demetri Terzopoulos, Konstantinos Plataniotis, Sergey Tulyakov, and Jian Ren. Wonderland: Navigating 3d scenes from a single image. *arXiv*, 2024. 2

[28] Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *CVPR*, 2024.

[29] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *CVPR*, 2024.

[30] 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. *arXiv preprint arXiv:2303.11328*, 2023. 2

[31] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *arXiv*, 2017. 6

[32] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *CVPR*, 2021. 2

[33] Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. Videofusion: Decomposed diffusion models for high-quality video generation. In *CVPR*, 2023. 2

[34] Norman Müller, Katja Schwarz, Barbara Rössle, Lorenzo Porzi, Samuel Rota Bulò, Matthias Nießner, and Peter Kontschieder. Multidiff: Consistent novel view synthesis from a single image. In *CVPR*, 2024. 2, 5

[35] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 2

[36] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 2

[37] Michael Niemeyer, Lars M. Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *CVPR*, 2020. 2

[38] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *CVPR*, 2022. 2

[39] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: improving latent diffusion models for high-resolution image synthesis. *CoRR*, arxiv preprint arxiv:2307.01952, 2023. 2

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

[41] Xuanchi Ren, Yifan Lu, Hanxue Liang, Jay Zhangjie Wu, Huan Ling, Mike Chen, Francis Fidler, Sanja annd Williams, and Jiahui Huang. Scube: Instant large-scale scene reconstruction using voxsplats. In *NeurIPS*, 2024. 2

[42] Barbara Roessle, Jonathan T. Barron, Ben Mildenhall, Pratul P. Srinivasan, and Matthias Nießner. Dense depth priors for neural radiance fields from sparse input views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

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

[44] Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lucic, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas Funkhouser, and Andrea Tagliasacchi. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *CVPR*, 2022. 2

[45] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 3

[46] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, and Jiajun Wu. ZeroNVS: Zero-shot 360-degree view synthesis from a single real image. *arXiv preprint arXiv:2310.17994*, 2023. 2

[47] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[48] Katja Schwarz, Axel Sauer, Michael Niemeyer, Yiyi Liao, and Andreas Geiger. Voxgraf: Fast 3d-aware image synthesis with sparse voxel grids. In *Advances in Neural Information Processing Systems*, 2022.

[49] Katja Schwarz, Seung Wook Kim, Jun Gao, Sanja Fidler, Andreas Geiger, and Karsten Kreis. Wildfusion: Learning 3d-aware latent diffusion models in view space. In *ICLR*, 2024. 2

[50] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *ICLR*, 2024. 2

[51] Jaidev Shriram, Alex Trevithick, Lingjie Liu, and Ravi Ramamoorthi. Realmdreamer: Text-driven 3d scene generation with inpainting and depth diffusion. *arXiv*, 2024. 2

[52] 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. In *ICLR*, 2023. 2

[53] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *arXiv preprint arXiv:1906.01618*, 2019. 2

[54] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 2, 12

[55] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, 2020. 12

[56] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2

[57] Wenqiang Sun, Shuo Chen, Fangfu Liu, Zilong Chen, Yueqi Duan, Jun Zhang, and Yikai Wang. Dimensionx: Create any 3d and 4d scenes from a single image with controllable video diffusion. *arXiv*, 2024. 2

[58] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *CVPR*, 2024. 2

[59] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. In *ACM TOG*, 2023. 4, 6, 13

[60] Ayush Tewari, Tianwei Yin, George Cazenavette, Semon Rezchikov, Joshua B. Tenenbaum, Frédo Durand, William T. Freeman, and Vincent Sitzmann. Diffusion with forward models: Solving stochastic inverse problems without direct supervision. *NeurIPS*, 2023. 1, 2

[61] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d scene representation and rendering. In *ICCV*, 2021. 2

[62] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 5

[63] 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*, 2024. 2

[64] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 2

[65] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 2, 6

[66] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM TOG*, 2024. 2, 3

[67] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. In *ICLR*, 2023.

[68] Daniel Watson, Saurabh Saxena, Lala Li, Andrea Tagliasacchi, and David J. Fleet. Controlling space and time with diffusion models. *arXiv*, 2024. 2, 5, 13

[69] Christopher Wewer, Kevin Raj, Eddy Ilg, Bernt Schiele, and Jan Eric Lenssen. latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. *arXiv*, 2024. 2, 7, 13, 15

[70] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P. Srinivasan, Dor Verbin, Jonathan T. Barron, Ben Poole, and Aleksander Holynski. Reconfusion: 3d reconstruction with diffusion priors. *arXiv*, 2023. 1, 2

[71] Dejia Xu, Weili Nie, Chao Liu, Sifei Liu, Jan Kautz, Zhangyang Wang, and Arash Vahdat. Camco: Camera-controllable 3d-consistent image-to-video generation. *arXiv*, 2024. 2

[72] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023. 5

[73] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural radiance fields from one or few images. In *CVPR*, pages 4578–4587, 2021. 2

[74] Jason J. Yu, Fereshteh Forghani, Konstantinos G. Derpanis, and Marcus A. Brubaker. Long-term photometric consistent novel view synthesis with diffusion models. In *ICCV*, 2023. 5

[75] Jason J. Yu, Fereshteh Forghani, Konstantinos G. Derpanis, and Marcus A. Brubaker. Long-term photometric consistent novel view synthesis with diffusion models. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 2

[76] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv*, 2024. 1, 2, 3, 5, 7, 14, 15

[77] Yang Yuanbo, Shao Jiahao, Li Xinyang, Shen Yujun, Geiger Andreas, and Liao Yiyi. Prometheus: 3d-aware latent diffusion models for feed-forward text-to-3d scene generation. *arXiv*, 2024. 2

[78] 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. 2

[79] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. GS-LRM: large reconstruction model for 3d gaussian splatting. In *ECCV*, 2024. 2

[80] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5

[81] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *ICCV*, 2021. 2

[82] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM TOG*, 2018. 5

[83] Zhizhuo Zhou and Shubham Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *CVPR*, 2023. 2

[84] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. FSGS: real-time few-shot view synthesis using gaussian splatting. In *ECCV*, 2024. 2