Sampling 3D Gaussian Scenes in Seconds with Latent Diffusion Models

Paul Henderson
University of Glasgow

Melonie de Almeida
University of Glasgow

Daniela Ivanova
University of Glasgow

Titas Anciukevičius
University of Edinburgh

***Abstract*—We present a latent diffusion model that can generate 3D scenes, yet is trained using only 2D image data. To achieve this, we first design an autoencoder that maps multi-view images to 3D Gaussian splats, and simultaneously builds a compressed latent representation of these splats. Then, we train a multi-view diffusion model over the latent space to learn an efficient generative model. This pipeline does not require object masks nor depths, and is suitable for complex scenes with arbitrary camera positions. We conduct careful experiments on two large-scale datasets of complex real-world scenes – MVImgNet and RealEstate10K. We show that our approach enables generating 3D scenes in as little as 0.2 seconds, either from scratch, from a single input view, or from sparse views. It produces diverse and high-quality results yet runs an order of magnitude faster than non-latent diffusion models and NeRF-based generative models.**

I. Introduction

Learning generative models that can sample realistic 3D scenes is a compelling yet challenging problem. In games and visual effects, they enable effortless creation of 3D assets, which currently is notoriously difficult, slow and expensive. In computer vision, they enable 3D reconstruction of scenes from a single image, with the generative model synthesizing plausible 3D details even for regions not visible in the image—unlike classical 3D reconstruction methods [20], [40].

Large-scale datasets of images, text and video have enabled learning impressive generative models for those modalities [49], [46], [26]. However, there are no extant large-scale datasets of photorealistic 3D scenes. Existing 3D datasets are either large but consist primarily of isolated objects (not full scenes), often with unrealistic textures [70], [16], [8]; or they are photorealistic environments (captured with 3D scanners) but too small for learning a generative model over [15], [4]. In contrast, large-scale in-the-wild datasets of *multi-view images* are now readily available [76], [47].

It is therefore desirable to learn 3D generative models directly from datasets of multi-view images, rather than from 3D data. One naïve strategy is to apply standard 3D reconstruction techniques to every scene in such a dataset, then train a 3D generative model directly on the resulting reconstructions [41], [77]. However, this is computationally expensive. It also leads to a challenging learning task for the generative model, since by reconstructing scenes independently we do not obtain a smooth, shared space of representations (e.g. similar scenes may have very different weights when represented as a NeRF [40]). This makes it difficult to learn a prior that generalizes, rather than simply memorizing individual scenes. These limitations have inspired a line of works that learn 3D generative models directly from images [3], [58], [54], [23]. Unfortunately, such methods are typically very slow to sample, since they require expensive volumetric rendering operations after every step of a diffusion process [58], [2], [63].

In this work, we design an efficient generative model for 3D scenes that is trained using only posed multi-view images. Our key idea is to learn an autoencoder on multi-view images, that simultaneously builds a structured 3D representation while also compressing this into a lower dimensionality latent space. We can then train a denoiser on the resulting latent representations, and perform the expensive iterative denoising process in this much lower-dimensional space. Unlike [58], [2], [63], the rendering operation is *not* inside the sampling loop.

We adopt Gaussian Splats [29] as our 3D representation since they achieve a favorable trade-off between reconstruction quality and training/rendering speed. The original work of Kerbl *et al.* [29] considered only reconstruction from densely-captured images, but several more recent works aim to predict splats from one or few images [12], [9], [59], [78], [73]. However, unlike ours, these methods are not generative – they do not capture a *distribution* over 3D scenes. They therefore cannot predict the full space of scenes consistent with the input images; nor can they perform other tasks such as unconditional or class-conditional generation. Score distillation [45], [65] provides a way to leverage 2D generative models for 3D content synthesis, but such methods neither learn nor sample a true 3D prior, and so are still prone to undesirable artifacts [61], [72], [14].

Our proposed model is very fast, since it benefits from the efficient rendering and optimization of Gaussian Splats, and also from the speed-up provided by diffusion over a compressed latent space. This enables sampling a full batch of eight 3D scenes in just 1.6s—more than $20\times$ faster than the fastest NeRF-based 3D-aware diffusion model [58]. In addition, our model has the following desirable features: (i) it can represent arbitrarily large scenes, by placing 3D content anywhere in the view frustum of the cameras (in particular, it is not limited to pre-segmented or object-centric scenes); (ii) it supports several tasks—unconditional generation, single-image 3D reconstruction, sparse-view 3D reconstruction, depending on the conditioning signal provide during inference; (iii) it does not rely on any depth or segmentation information for training—it is trained from scratch using only multi-view images.

To summarize, our core contribution is **a highly efficient generative model that learns and samples a distribution over real-world scenes represented as Gaussian Splats**. Our technical contributions that enable this are: (i) we design a

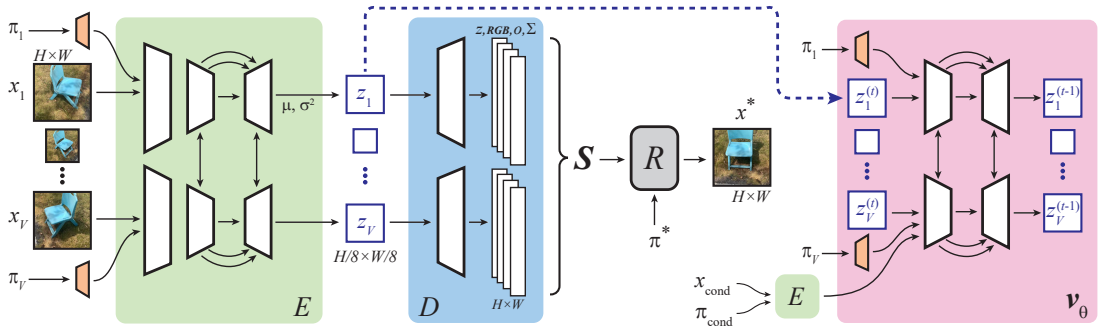


Fig. 1: Overview of our latent diffusion model for 3D scene synthesis. **Left:** We train an autoencoder to encode (green box; E) multi-view images $\{x_v\}_1^V$ to a compressed latent space $\{z_v\}_1^V$. It simultaneously learns to decode (blue box; D) the latents to parameters of Gaussian splats $\mathcal{S}$, which can be rendered back to images x^*. **Right:** We train a denoising diffusion model (pink box; $\boldsymbol{v}_\theta$) over the latents z_v. This supports unconditional generation, or generation conditioned on an input image x_{cond}. Following the efficient, low-dimensional denoising process, the resulting latents are mapped back to a 3D scene by D.

new 3D-aware autoencoder architecture that learns to represent multi-view images via a compressed latent space, that can be decoded to Gaussian Splats; (ii) we demonstrate how diverse and realistic 3D scenes can be sampled efficiently with a diffusion model on the latent representation, either unconditionally or conditioned on an input image; (iii) we show that for a given compute budget, our latent approach gives significantly better results for both unconditional generation and generative reconstruction. Our code and data will be made publicly available on acceptance.

II. Related Work

1) Reconstruction from dense views: Neural radiance fields (NeRFs) [40] represent 3D scenes implicitly by a neural network mapping position to density and color; they can be optimized by gradient descent on an image reconstruction loss. Gaussian Splatting [29] provides an alternative that allows real-time rendering and fast training, with quality approaching that of state-of-the-art NeRFs. Our work uses this efficient representation, but instead of fitting individual scenes, we build a generative model that learns to sample them conditioned on a class label or sparse set of images.

2) Reconstruction from sparse views: While the above methods can reconstruct 3D from dense (e.g. >50) sets of images, in practice, parts of a scene cannot be observed from multiple images and need to inferred. To solve this, a line of work trains models to reconstruct 3D scenes from fewer (e.g. <10) views. Most approaches [74], [10], [37], [24], [37], [68], [48] unproject 2D image features into 3D space, fuse them and apply NeRF rendering. Several recent methods [9], [12], [78], [73], [67], [59] tackle the sparse-view reconstruction task using splats as the 3D representation; others aim to predict novel views directly, without explicit 3D [32], [50]. However, these methods are not probabilistic—they do not represent uncertainty about unobserved parts of the scene (e.g. the back of an object). As a result, instead of sampling one of many plausible 3D representations, these methods output a single average solution. For example, *MVSplat* [12] and *pixelSplat* [9] train a network to map context images to 3D splats, but lack the ability to generate 3D scenes unconditionally or to sample diverse completions for occluded regions. *latentSplat* [67] does construct a posterior distribution over splats then sample this—however its mean-field posterior on splat parameters means it cannot capture complex posterior dependencies in the scene geometry, and thus cannot sample coherent shapes for unobserved areas of scenes. Most of the above methods are designed for two or more input views. Conversely, SplatterImage [59] can predict 3D splats from a single image, though it is limited to masked, object-centric scenes. While some approaches incorporate information from generative models ad-hoc to increase plausibility of uncertain regions [56], [81], [39], [44], they do not learn to sample the true posterior distribution on scenes like our method does.

3) Generative models: Diverse families of generative models [31], [19], [27] have been proposed to learn complex distributions from training data. A straightforward approach to creating generative models of 3D content is to assemble a large-scale 3D dataset [8], [16] and train a generative model directly on this [38], [64], [11], [13], [41], [66], [30]. However, large-scale, highly-realistic 3D datasets of scenes are challenging to create, and most 3D representations lack shared structure (as each datapoint is created independently, e.g. meshes have different topologies, and point-clouds different numbers of points) making learning priors difficult.

To circumvent the difficulty of learning a smooth prior over independently-reconstructed scenes, various methods learn a generative model directly from images. One approach [69], [5], [35], [62], [75], [36], [34], [28], [18], [60] is to generate multi-view images conditioned on camera poses without an explicit 3D representation, then reconstruct 3D from the generated images using classical methods. In general this is challenging since the generated images may not be multi-view-consistent.

To directly sample 3D representations, some works learn 3D-aware generative models of 2D images, which retain the mathematical formulation of generative image models, but introduce priors into the network architecture which force

the model to output images via an explicit 3D representation. Seminal approaches were based on VAEs [21], [23], [1], [22] and GANs [54], [7], [43], [42], while more recent methods use 3D-aware denoising diffusion [3], [58], [63], [71], [2], [55], [33]. Unlike score-distillation methods [45], [61] that suffer from mode-seeking behavior and do not truly sample a distribution, 3D-aware diffusion models can sample 3D scenes from the true posterior distribution. However, works using radiance fields to represent the scene are limited by slow training and generation times. In contrast, our work samples 3D scenes in less than a second (0.2s) compared to 5.4s for [58] or 51s for [2]; it can then render the 3D scene in real-time.

III. Method

Our goal is to build a model supporting conditional and unconditional generation of 3D scenes. We assume access only to a training dataset of multi-view images with camera poses (readily available from phone cameras or COLMAP SfM [53]). We do not require any additional 2D/3D supervision (e.g. annotations, pretrained models, foreground masks, depth-maps) and we do not assume that camera poses are aligned consistently across the dataset.

We achieve this by designing a latent diffusion framework that is trained in two stages. First, a 3D-aware variational autoencoder (VAE) is trained on sets of multi-view images (Sec.III-A). It encodes multi-view images to a compact latent representation, decodes this to an explicit 3D scene represented by Gaussian Splats, then renders the scene to reconstruct images. Second, we train a denoising diffusion model on the compact latent space learnt by the autoencoder (Sec.III-B). This diffusion model is trained jointly for class-conditional and image-conditional generation, and can efficiently learn a distribution on the latent space. During inference, sampled latents are decoded to splats by the autoencoder then rendered.

A. Autoencoder

Our autoencoder takes as input V views $\boldsymbol{x} = \{x_v\}_{v=1}^V$ of a scene (each of size $H \times W$ pixels), with their relative camera poses $\boldsymbol{\pi} = \{\pi_v\}_{v=1}^V$. It processes these jointly to give a set of splats $\mathcal{S}$, such that rendering $\mathcal{S}$ from each π_v should reconstruct the original image x_v. Importantly, it passes all information about the scene through a low-dimensional latent bottleneck, yielding a compressed representation from which the splats are then decoded and rendered.

1) Encoding multi-view images: The x_v are first passed independently through three downsampling residual blocks similar to [17], [49], yielding feature maps of resolution $\frac{H}{8} \times \frac{W}{8}$. These features are processed by a multi-view U-Net [51], which enables the different views to exchange information efficiently (necessary to achieve a consistent 3D reconstruction). This U-Net is based closely on [27]. To adapt it to our multi-view setting, we take inspiration from video diffusion models, notably [6], and add a small cross-view ResNet after each block that combines information from all views, for each pixel independently. We also modify all attention layers to jointly attend across features from all views. Aside from these parts, all the remainder of processing by residual blocks treats views independently. The final convolution of the U-Net outputs the mean and log-variances of a feature-map of size $\frac{H}{8} \times \frac{W}{8}$ for each view. This will be the compressed latent space $\{z_v\}_{v=1}^V$ in which we perform denoising (see Sec.III-B), and so we restrict it to have only very few channels. We assume a diagonal Gaussian posterior distribution, following [31], [49]. We denote the overall encoder mapping from $\{x_v\}_{v=1}^V$ to a latent sample $\{z_v\}_{v=1}^V$ by E (green box in Fig. 1).

2) Decoding to a 3D scene: The latent features $\{z_v\}_{v=1}^V$ are decoded to a 3D scene $\mathcal{S}$ represented as Gaussian Splats [29]. Specifically, we pass the features through three upsampling residual blocks, mirroring the initial layers of E, to give feature maps of the same size as the original images. Similarly to [59], [9], the features for each view are mapped by a convolution layer to parameters of splats supported on the view frustum. For each pixel, we predict the depth, opacity, color, rotation and scale of one splat; in total this requires 12 channels. The 3D position of each splat is then calculated by unprojecting it along the corresponding camera ray by the predicted depth. This representation (aptly termed a *splatter image* in [59]) provides a structured way to represent splats, allowing reasoning over them with standard convolution layers. The union of the $V \times H \times W$ splats across all images constitutes our scene representation $\mathcal{S}$. We denote the mapping from $\{z_v\}_{v=1}^V$ to $\mathcal{S}$ by D; it is shown by the blue box in Fig. 1. The splats $\mathcal{S}$ can be rendered to pixels x^* using arbitrary camera parameters π^*; we denote this rendering operation by $x^* = R(\mathcal{S}, \pi^*)$.

A key benefit of having splats supported on images at all denoised viewpoints is that we can represent 3D content anywhere we look. In contrast [59] can only place splats in (or very near) the view frustum of the input image, whereas ours can generate coherent content arbitrarily far away.

3) Conditioning on pose: In order to condition the autoencoder on the relative poses of the views, we design a novel strategy based on the splat renderer itself. For each view, we generate a set of splats along the edges of its view frustum; for each view, we assign a random color. We then render the resulting splat cloud from all cameras. These renderings are concatenated with the input x_v before the first residual block of the encoder. Note that without this conditioning, it is impossible for the autoencoder to learn the arbitrary scene scale (even if it successfully learns to triangulate the input images), due to the perspective depth/scale ambiguity.

4) Training: We assume access to minibatches containing V input views $\left(\boldsymbol{x}^{(\text{in})}, \boldsymbol{\pi}^{(\text{in})}\right)$, and an additional V' nearby target views $\left(\boldsymbol{x}^{(\text{tgt})}, \boldsymbol{\pi}^{(\text{tgt})}\right)$ that are not passed to E. We predict splats $\mathcal{S} = D\left(E(\boldsymbol{x}^{(\text{in})}, \boldsymbol{\pi}^{(\text{in})})\right)$, then render these at the target views giving $x^*_{v'} = R\left(\mathcal{S}, \pi^{(\text{tgt})}_{v'}\right) \forall v' = \{1 \dots V'\}$. The network is trained as a variational autoencoder [31], using the sum of L^2 and LPIPS [79] distances between input and rendered images as a reconstruction loss, and maximizing the log-probability of the sampled latents $\{z_v\}_{v=1}^V$ under a standard

Gaussian prior. This leads to the following loss:

$$\mathcal{L}_{\text{AE}} = \sum_{v'=1}^{V'} \left\{ \left|\left| x^*_{v'} - x^{(\text{tgt})}_{v'} \right|\right|_2^2 + \text{LPIPS}\left(x^*_{v'},\, x^{(\text{tgt})}_{v'}\right) + \beta\, ||z_v||_2^2 \right\} \tag{1}$$

where β adjusts the weight of the KL loss. Note that unlike methods using NeRFs, the fast rendering operation R means we can straightforwardly apply LPIPS to full-image renderings.

5) Compression: Compared with treating the splats themselves as the latent space, our approach yields a compression factor of $128\times$ when we use 6 latent channels (as in our main experiments). As shown in Sec. IV, this enables much more efficient training and inference for the denoiser.

B. Denoiser

We now define a denoising diffusion model over the low-dimensional latents $\{z_v\}_{v=1}^V = \boldsymbol{z}$ produced by the encoder E. We take a similar approach to latent diffusion for images [49], but instead of a 2D U-Net, use a very similar multi-view U-Net architecture as in the autoencoder (Sec. III-A). We condition this multi-view U-Net $\hat{\boldsymbol{v}}_\theta$ on the camera poses π_v in the same way as for the autoencoder, i.e. concatenating a representation of all view frusta, as well as the diffusion timestep. The U-Net is then responsible for learning to jointly denoise all views, passing information via cross-view attention and convolution.

1) Conditional generation: We train the denoiser jointly for image-conditional and class-conditional generation (choosing randomly which to use for each minibatch), and also dropping the conditioning entirely for 20% of minibatches to enable classifier-free guidance [25]. For class conditioning, we use a learned embedding for class indices, which is added to the timestep embedding. For image conditioning (i.e. when performing 3D reconstruction), we re-use our pretrained encoder E, to encode the conditioning images x_{cond} and their poses π_{cond}. The conditioning latents output by $E(x_{\text{cond}}, \pi_{\text{cond}})$ are concatenated with the noisy latents at the start of the denoiser.

2) Training: During training we sample minibatches of posed views $(\boldsymbol{x}, \boldsymbol{\pi})$. These are converted to latents $\boldsymbol{z}$ by passing them to E and sampling the posterior. We then sample a diffusion timestep t and sample noisy latent from the Gaussian forward process $\mathcal{N}(\boldsymbol{z}^{(t)}; \alpha_t \boldsymbol{z}, \sigma_t^2 \boldsymbol{I})$, where α_t and σ_t are specified by a linear noise schedule; we optimize the denoiser's parameters θ by gradient descent on the loss

$$\mathcal{L}_{\text{DDM}} = \mathbb{E}_{t,\, \boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}),\, \boldsymbol{z}^{(t)}} ||\hat{\boldsymbol{v}}_\theta(\boldsymbol{z}^{(t)}, t) - \boldsymbol{v}^{(t)})||_2^2 \tag{2}$$

Importantly, due to our abstract latent space, our implementation trains the denoiser $\hat{\boldsymbol{v}}_\theta(\boldsymbol{z}^{(t)}, t)$ to predict $\boldsymbol{v}^{(t)} \equiv \alpha_t \boldsymbol{\epsilon} - \sigma_t \boldsymbol{z}$ [52] which is more numerically stable than $\boldsymbol{x}^{(0)}$ prediction used by other 3D-aware diffusion models [58], [2], [3] that constrain the denoiser itself to output rendered pixels.

3) Sampling: To sample a 3D scene from our model, we begin by sampling Gaussian noise $\boldsymbol{z}^{(1000)} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ in the latent space, and choosing a set of camera poses (e.g. from a held-out validation set) as conditioning. We then use DDIM sampling [57] to find $\boldsymbol{z}^{(0)}$, with classifier-free guidance for class conditioning. From this we decode the generated 3D scene by $\mathcal{S} = D(\boldsymbol{z}^{(0)})$, which can then be rendered efficiently from arbitrary viewpoints π^* using R.

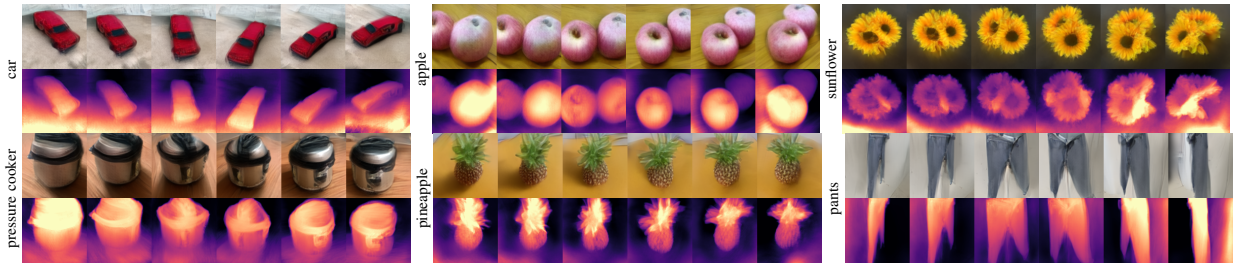
IV. Experiments

We evaluate our model and several baselines on both *a priori* generation and one-/few-view 3D reconstruction.

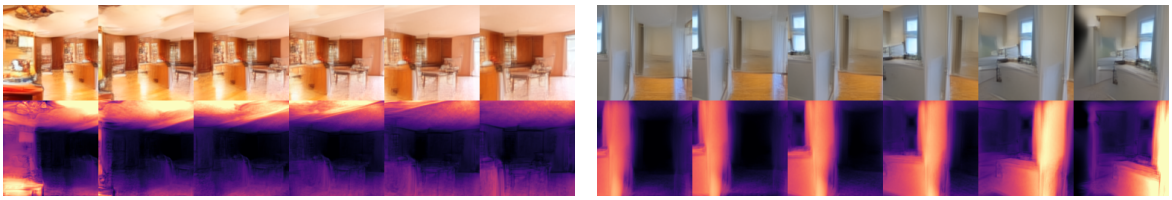
1) Datasets: We evaluate our approach on two large-scale datasets of real-world images—MVImgNet [76] and RealEstate10K [80]. MVimgNet consists of videos showing diverse objects in the wild. We use the splits from MVPNet subset, containing 87,820 videos (typically 30 frames long) covering 180 object classes, and restrict to 5000 scenes for evaluation. RealEstate10K contains 69893 videos showing indoor and outdoor views of houses (typically 50-200 frames long). We use the official splits, again limiting to 5000 scenes for evaluation. Note that the videos in both datasets depict complete scenes, not just isolated objects. For both datasets, we center crop the images with size equal to small edge, then rescale to 96×96. During training, we randomly sample sets of six frames as multi-view images. We use the camera poses provided with each dataset, but only provide *relative* poses to the model. We do not require any canonicalization of scene orientation or scale, unlike earlier methods [58], [3], [59]

2) Baselines: We compare our approach to several existing works. **GIBR** [2] and **ViewSet Diffusion** [58] are 3D-aware diffusion models over multi-view images. They encode sets of noisy views, and uses these to build a radiance field representation of the scene that is then rendered to give the denoised images. Both are relatively expensive as they perform volumetric rendering at every denoising step. **RenderDiffusion** [3] is a similar method that only requires single images during training; we use the variant adapted for the in-the-wild data by [2]. **SplatterImage** [59] is a deterministic method for 3D reconstruction from a single image, outputting splats from a single U-Net pass. The original work only considers masked views of isolated objects; we adapt their method to work for our larger, unmasked scenes. **PixelNeRF** [74] predicts a radiance field deterministically by unprojecting features from input images; we use the in-the-wild variant from [2].

3) Generation results: We first evaluate on unconditional generation of 3D scenes (for RealEstate10K), and class-conditional generation for the full (180 classes) MVImgNet dataset. Qualitative examples are given in Fig. 2. We see that our model can generate objects of diverse classes, given just the label as conditioning. The generated scenes are coherent across views, including during long camera motions in RealEstate10K. We also perform a quantitative evaluation against several existing methods. Here we closely follow the setting of [2], using the *chair*, *table* and *sofa* classes of MVImgNet. Specifically, we compare against GIBR [2], ViewSet Diffusion [58] and also the earlier RenderDiffusion [3]. The results are presented in Table I (bottom five rows). On class-conditional generation, our method significantly out-performs the baselines, achieving an FID of 89.0, vs 99.8 for the second best (GIBR). Moreover, it achieves this while being $174\times$ faster than GIBR (just 0.22s for ours, vs 45s for GIBR), since



(a) MVImgNet



(b) RealEstate10K

Fig. 2: Qualitative examples of class-conditional (MVImgNet) and unconditional 3D generations (RealEstate10K) from our method. For each example, the top row shows six rendered views of the sampled 3D scene, while the bottom row shows the corresponding depths. Note that our model samples 3D scenes containing objects with complex shape on a realistic background.

TABLE I: Results from our method and baselines on unconditional/class-conditional scene generation, 3D reconstruction from a single image, and 3D reconstruction from sparse (six) views. 'sec' is generation time in seconds.

	Generation		1-view recon.			6-view recon.	
	FID ↓	sec. ↓	PSNR ↑	LPIPS ↓	sec. ↓	PSNR ↑	LPIPS ↓
MVImgNet							
Ours	**23.1**	**0.22**	**20.6**	**0.324**	0.22	**24.7**	**0.184**
SplatterImage [59]	–	–	18.2	0.367	**0.03**	–	–
RealEstate10K							
Ours	**29.5**	**0.22**	**16.4**	0.455	0.22	**23.3**	**0.155**
SplatterImage [59]	–	–	16.2	**0.347**	**0.03**	–	–
MVImgNet furniture							
Ours	**89.0**	**0.22**	17.9	**0.407**	0.22	22.9	0.214
GIBR [2]	99.8	44.9	**18.5**	0.414	44.3	**25.4**	**0.199**
Viewset Diffn. [58]	191.4	4.98	17.6	0.540	4.25	–	–
RenderDiffusion [3]	234.1	10.2	17.4	0.622	10.2	18.4	0.601
PixelNeRF [74], [2]	–	–	16.6	0.582	**0.12**	15.7	0.647

it avoids the need to render an image at every denoising step, and also operates over a lower-dimensionality space.

4) Single-view 3D reconstruction results: We now evaluate generative 3D reconstruction from a single image. Given an image, we predict 11 other evenly-spaced viewpoints, and measure the accuracy of the predicted views using PSNR and LPIPS [79]. Since our method is generative (and can generate many plausible samples for a given input), we follow standard practice for stochastic prediction and draw multiple (20) samples for each scene then record the best. Here we compare against SplatterImage [59] on MVImgNet and RealEstate10K, and all other baselines on the furniture subset of MVImgNet. Qualitative results are given in Fig. 3. We see that our method can reconstruct plausible shapes from a single image, for diverse object classes and even entire rooms. Details in the input image are preserved, while plausible content is generated in occluded parts. Moreover, in Fig. 4, we show that given one image of an object, our model can sample diverse (yet plausible) texture and shape for the unobserved back of the object. Quantitative results are presented in Table I ('1-view recon.' columns). We see that our method performs significantly better than SplatterImage according to both PSNR and LPIPS on the full MVImgNet dataset, as well as PSNR on RealEstate10K, though SplatterImage slightly exceeds it according to LPIPS. On the MVImgNet furniture subset (following the protocol of [2]), our method is best with respect to LPIPS, while GIBR performs slightly better according to PSNR (which generally favors blurrier results). The older, deterministic PixelNeRF method performs worst according to both reconstruction metrics.

We also measure the time to reconstruct a scene using each of these methods, processing a minibatch of 8 scenes then calculating the average time per scene, on an NVIDIA RTX 3090. Of the generative methods, ours is by far the fastest (0.22s), compared with 4.3s for the next quickest (ViewSet Diffusion). GIBR is the slowest at 44s, since it must construct and render a NeRF during each denoising step. PixelNeRF is fastest overall, but trades off quality for speed and is unable to represent a posterior over 3D scenes.

5) Sparse-view 3D reconstruction results: We evaluate reconstruction from sparse (six) views, measuring PSNR and LPIPS at six other viewpoins spaced equally between the inputs. Qualitative performance is very good, with fine details preserved, and plausible depth-maps. Quantitatively (see Table I, '6-view recon.' columns), our method slightly under-performs GIBR on MVImgNet, although it is much faster. Note that for this task we use only our autoencoder, not the denoiser. Thus, these results show the 3D autoencoder preserves fine details, even while compressing them by $128\times$, justifying its use as a latent space over which to learn a generative prior.

(a) MVImgNet

(b) RealEstate10k

Fig. 3: Qualitative comparison of 3D reconstruction from a single image between our model (top row of each scene) and SplatterImage [59] (bottom row of each scene). The first column shows the input (conditioning) image, the second displays the ground truth images, while the third and fourth columns display the predicted frames and depths, respectively. Compared to the baseline, our model yields more plausible reconstructions, especially of the occluded regions and the background.

Fig. 4: Given a single image (first column), our model generates diverse plausible back-views (columns 2 through 7). When compared with the back-view generated by a deterministic model (column 8), our model's predictions are much sharper.

Fig. 5: Given one frame from RealEstate10K (first column), our model generates parts of the house that are not initially visible. Top: the camera moves into to the doorway to the left. Bottom: the camera moves along the hallway. Our model generates diverse samples for the room that is revealed.

6) Benefit of being generative and latent: We now evaluate several variants of our model, demonstrating the benefit of operating on latent space, and of treating 3D reconstruction as a generative (probabilistic) task, rather than deterministic. Results on MVImgNet are presented in Table II. Here **Deterministic** denotes a variant of our model where the denoiser is replaced by a deterministic predictor of the latent variables representing the scene, given one input image. This model has the same network architecture as the main model, but does not sample a posterior distribution over possible scenes; instead it is expected to learn the conditional expectation in latent space. We see this scores substantially lower than our model on LPIPS, and slightly lower on PSNR. This is expected since PSNR penalizes overly-smoothed solutions less heavily than LPIPS (which prefers perceptually-correct textures and edge). Moreover, qualitative results in Fig. 4 show that where our method generates many sharp, plausible samples for unobserved parts of the scene, the deterministic variant yields a blurry prediction that averages away the uncertainty. Moreover, for RealEstate10K (Fig. 5), our model can generate diverse yet plausible contents for rooms that are not visible in the input frame, e.g. when the camera

TABLE II: Non-generative and non-latent ablations of our method, on MVImgNet (all trained with equal compute budget).

	FID ↓	PSNR ↑	LPIPS ↓	Time /s ↓
Full model	**23.1**	**20.6**	**0.324**	0.22
Deterministic	–	20.0	0.466	**0.02**
Splats-as-latents	57.3	19.2	0.389	13.9
Non-latent	133.8	18.9	0.499	4.88

moves through a doorway. **Splats-as-latents** denotes a natural ablation where instead of learning a compressed latent space, we instead treat the multi-view splat parameters themselves as the latent variables, and learn a denoiser directly over these. We train this ablation for the same duration as our main model, to fairly measure the trade-off of performance vs accuracy. We see that both generation and reconstruction performance is significantly worse than ours (57.3 vs 23.1 FID, and 0.39 vs 0.32 LPIPS). Moreover, due to the $128\times$ larger dimensionality, this model takes more than $60\times$ longer than ours to sample a 3D scene. Thus, for a fixed training compute budget, our latent approach yields a clear benefit in terms of quality and accuracy. Lastly, **Non-latent** is an ablation that uses a single-stage training process, similar to [58], [2]. Here the denoiser operates directly over pixels, but incorporates our splat-based scene representation within the decoder, with clean pixels given by rendering this at each denoising step. This model performs worse still, reaching an FID of 133.8 and LPIPS of 0.499.

V. Conclusion

We introduced a latent diffusion model that generates and reconstructs large real-world scenes represented as 3D Gaussians in as little as 0.2 seconds. Our evaluations showed that this model performs $20\times$ faster than existing 3D-aware diffusion models whilst maintaining scene realism and enabling real-time rendering. Trained solely on posed multi-view images, our model enables the use of diverse, in-the-wild datasets.

References

[1] T. Anciukevicius, P. Fox-Roberts, E. Rosten, and P. Henderson. Unsupervised causal generative understanding of images. *Advances in Neural Information Processing Systems*, 35:37037–37054, 2022.

[2] T. Anciukevičius, F. Manhardt, F. Tombari, and P. Henderson. Denoising diffusion via image-based rendering. In *The Twelfth International Conference on Learning Representations*, 2024.

[3] T. Anciukevičius, Z. Xu, M. Fisher, P. Henderson, H. Bilen, N. J. Mitra, and P. Guerrero. Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12608–12618, June 2023.

[4] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv:1702.01105*, 2017.

[5] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

[6] A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S. W. Kim, S. Fidler, and K. Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[7] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. D. Mello, O. Gallo, L. Guibas, J. Tremblay, S. Khamis, T. Karras, and G. Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022.

[8] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[9] D. Charatan, S. Li, A. Tagliasacchi, and V. Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. *arXiv preprint arXiv:2312.12337*, 2023.

[10] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su. MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proc. IEEE/CVF Int. Conf. on Computer Vision*, 2021.

[11] H. Chen, J. Gu, A. Chen, W. Tian, Z. Tu, L. Liu, and H. Su. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In *ICCV*, 2023.

[12] Y. Chen, H. Xu, C. Zheng, B. Zhuang, M. Pollefeys, A. Geiger, T.-J. Cham, and J. Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. *arXiv preprint arXiv:2403.14627*, 2024.

[13] Y.-C. Cheng, H.-Y. Lee, S. Tuyakov, A. Schwing, and L. Gui. SDFusion: Multimodal 3d shape completion, reconstruction, and generation. In *CVPR*, 2023.

[14] J. Chung, S. Lee, H. Nam, J. Lee, and K. M. Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384*, 2023.

[15] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

[16] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36, 2024.

[17] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[18] R. Gao, A. Holynski, P. Henzler, A. Brussee, R. Martin-Brualla, P. Srinivasan, J. T. Barron, and B. Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv:2405.10314*, 2024.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[20] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[21] P. Henderson and V. Ferrari. Learning single-image 3D reconstruction by generative modelling of shape, pose and shading. *International Journal of Computer Vision (IJCV)*, 2019.

[22] P. Henderson and C. H. Lampert. Unsupervised object-centric video generation and decomposition in 3D. In *Advances in Neural Information Processing Systems (NeurIPS) 33*, 2020.

[23] P. Henderson, V. Tsiminaki, and C. Lampert. Leveraging 2D data to learn textured 3D mesh generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[24] P. Henzler, J. Reizenstein, P. Labatut, R. Shapovalov, T. Ritschel, A. Vedaldi, and D. 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*, pages 4700–4709, 2021.

[25] J. Ho. Classifier-free diffusion guidance. *ArXiv*, abs/2207.12598, 2022.

[26] J. Ho, W. Chan, C. Saharia, J. Whang, R. Gao, A. Gritsenko, D. P. Kingma, B. Poole, M. Norouzi, D. J. Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv:2210.02303*.

[27] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33, 2020.

[28] H. Hu, Z. Zhou, V. Jampani, and S. Tulsiani. Mvd-fusion: Single-view 3d via depth-consistent multi-view generation. In *CVPR*, 2024.

[29] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

[30] S. W. Kim, B. Brown, K. Yin, K. Kreis, K. Schwarz, D. Li, R. Rombach, A. Torralba, and S. Fidler. Neuralfield-ldm: Scene generation with hierarchical latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[31] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[32] J. Kulhánek, E. Derner, T. Sattler, and R. Babuška. Viewformer: Nerf-free neural rendering from few images using transformers. In *European Conference on Computer Vision (ECCV)*, 2022.

[33] Y. Lan, F. Hong, S. Yang, S. Zhou, X. Meng, B. Dai, X. Pan, and C. C. Loy. Ln3diff: Scalable latent neural fields diffusion for speedy 3d generation. In *ECCV*, 2024.

[34] M. Liu, C. Xu, H. Jin, L. Chen, M. Varma T, Z. Xu, and H. Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *NeurIPS*, 36, 2024.

[35] R. Liu, R. Wu, B. V. Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *ArXiv*, 2023.

[36] Y. Liu, C. Lin, Z. Zeng, X. Long, L. Liu, T. Komura, and W. Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. In *The Twelfth International Conference on Learning Representations*, 2024.

[37] Y. Liu, S. Peng, L. Liu, Q. Wang, P. Wang, C. Theobalt, X. Zhou, and W. Wang. Neural rays for occlusion-aware image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7824–7833, 2022.

[38] S. Luo and W. Hu. Diffusion probabilistic models for 3d point cloud generation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2021.

[39] L. Melas-Kyriazi, C. Rupprecht, I. Laina, and A. Vedaldi. Realfusion: 360° reconstruction of any object from a single image. In *Arxiv*, 2023.

[40] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[41] N. Müller, Y. Siddiqui, L. Porzi, S. R. Bulo, P. Kontschieder, and M. Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4328–4338, 2023.

[42] T. Nguyen-Phuoc, C. Li, L. Theis, C. Richardt, and Y.-L. Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7588–7597, 2019.

[43] T. Nguyen-Phuoc, C. Richardt, L. Mai, Y.-L. Yang, and N. Mitra. BlockGAN: Learning 3d object-aware scene representations from unlabelled images. In *Advances in Neural Information Processing Systems*, 2020.

[44] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. M. Sajjadi, A. Geiger, and N. Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2022.

[45] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *ArXiv*, 2022.

[46] A. Radford and K. Narasimhan. Improving language understanding by generative pre-training. OpenAI, 2018.

[47] J. Reizenstein, R. Shapovalov, P. Henzler, L. Sbordone, P. Labatut, and D. Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021.

[48] C. Rockwell, D. F. Fouhey, and J. Johnson. Pixelsynth: Generating a 3d-consistent experience from a single image. In *ICCV*, 2021.

[49] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[50] R. Rombach, P. Esser, and B. Ommer. Geometry-free view synthesis: Transformers and no 3d priors. *ArXiv*, 2021.

[51] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[52] T. Salimans and J. Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022.

[53] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[54] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger. GRAF: generative radiance fields for 3d-aware image synthesis. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, 2020.

[55] K. Schwarz, S. Wook Kim, J. Gao, S. Fidler, A. Geiger, and K. Kreis. Wildfusion: Learning 3d-aware latent diffusion models in view space. In *International Conference on Learning Representations (ICLR)*, 2024.

[56] J. Shriram, A. Trevithick, L. Liu, and R. Ramamoorthi. RealmDreamer: Text-driven 3d scene generation with inpainting and depth diffusion. *ArXiv*, 2024.

[57] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[58] S. Szymanowicz, C. Rupprecht, and A. Vedaldi. Viewset diffusion: (0-)image-conditioned 3D generative models from 2D data. In *ICCV*, 2023.

[59] S. Szymanowicz, C. Rupprecht, and A. Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[60] J. Tang, Z. Chen, X. Chen, T. Wang, G. Zeng, and Z. Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024.

[61] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *The Twelfth International Conference on Learning Representations*, 2024.

[62] S. Tang, F. Zhang, J. Chen, P. Wang, and Y. Furukawa. MVDiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. In *Advances in Neural Information Processing Systems*, 2023.

[63] A. Tewari, T. Yin, G. Cazenavette, S. Rezchikov, J. B. Tenenbaum, F. Durand, W. T. Freeman, and V. Sitzmann. Diffusion with forward models: Solving stochastic inverse problems without direct supervision. In *Advances in Neural Information Processing Systems*, 2023.

[64] A. Vahdat, F. Williams, Z. Gojcic, O. Litany, S. Fidler, K. Kreis, et al. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems*, 35:10021–10039, 2022.

[65] H. Wang, X. Du, J. Li, R. A. Yeh, and G. Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. *ArXiv*, 2022.

[66] T. Wang, B. Zhang, T. Zhang, S. Gu, J. Bao, T. Baltrusaitis, J. Shen, D. Chen, F. Wen, Q. Chen, and B. Guo. Rodin: A generative model for sculpting 3d digital avatars using diffusion. *ArXiv*, 2022.

[67] C. Wewer, K. Raj, E. Ilg, B. Schiele, and J. E. Lenssen. latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. In *arXiv*, 2024.

[68] C.-Y. Wu, J. Johnson, J. Malik, C. Feichtenhofer, and G. Gkioxari. Multiview compressive coding for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9065–9075, 2023.

[69] R. Wu, B. Mildenhall, P. Henzler, K. Park, R. Gao, D. Watson, P. P. Srinivasan, D. Verbin, J. T. Barron, B. Poole, and A. Holynski. Reconfusion: 3d reconstruction with diffusion priors. *ArXiv*, 2023.

[70] T. Wu, J. Zhang, X. Fu, Y. Wang, L. P. Jiawei Ren, W. Wu, L. Yang, J. Wang, C. Qian, D. Lin, and Z. Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[71] Y. Xu, H. Tan, F. Luan, S. Bi, P. Wang, J. Li, Z. Shi, K. Sunkavalli, G. Wetzstein, Z. Xu, and K. Zhang. DMV3d: Denoising multi-view diffusion using 3d large reconstruction model. In *The Twelfth International Conference on Learning Representations*, 2024.

[72] T. Yi, J. Fang, J. Wang, G. Wu, L. Xie, X. Zhang, W. Liu, Q. Tian, and X. Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *CVPR*, 2024.

[73] X. Yinghao, S. Zifan, Y. Wang, C. Hansheng, Y. Ceyuan, P. Sida, S. Yujun, and W. Gordon. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. *ArXiv*, 2024.

[74] A. Yu, V. Ye, M. Tancik, and A. Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021.

[75] J. J. Yu, F. Forghani, K. G. Derpanis, and M. A. Brubaker. Long-term photometric consistent novel view synthesis with diffusion models. In *Proc. International Conf. on Computer Vision (ICCV)*, 2023.

[76] X. Yu, M. Xu, Y. Zhang, H. Liu, C. Ye, Y. Wu, Z. Yan, T. Liang, G. Chen, S. Cui, and X. Han. Mvimgnet: A large-scale dataset of multi-view images. In *CVPR*, 2023.

[77] B. Zhang, Y. Cheng, J. Yang, C. Wang, F. Zhao, Y. Tang, D. Chen, and B. Guo. GaussianCube: Structuring gaussian splatting using optimal transport for 3d generative modeling. *arXiv:2403.19655*, 2024.

[78] K. Zhang, S. Bi, H. Tan, Y. Xiangli, N. Zhao, K. Sunkavalli, and Z. Xu. GS-LRM: Large reconstruction model for 3d gaussian splatting. *ArXiv*, 2024.

[79] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. 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.

[80] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 37, 2018.

[81] Z.-X. Zou, W. Cheng, Y.-P. Cao, S.-S. Huang, Y. Shan, and S.-H. Zhang. Sparse3d: Distilling multiview-consistent diffusion for object reconstruction from sparse views. *arXiv:2308.14078*, 2023.