Scaling Transformer-Based Novel View Synthesis Models with Token Disentanglement and Synthetic Data

Nithin Gopalakrishnan Nair[1*] Srinivas Kaza[2*] Xuan Luo[2] Vishal M. Patel[1]
Stephen Lombardi[2] Jungyeon Park[2]
[1] Johns Hopkins University [2] Google
{ngopala2,vpatel36}@jhu.edu {srinivaskaza,xuluo,salombardi,jungyeonp}@google.com
https://scaling3dnvs.github.io

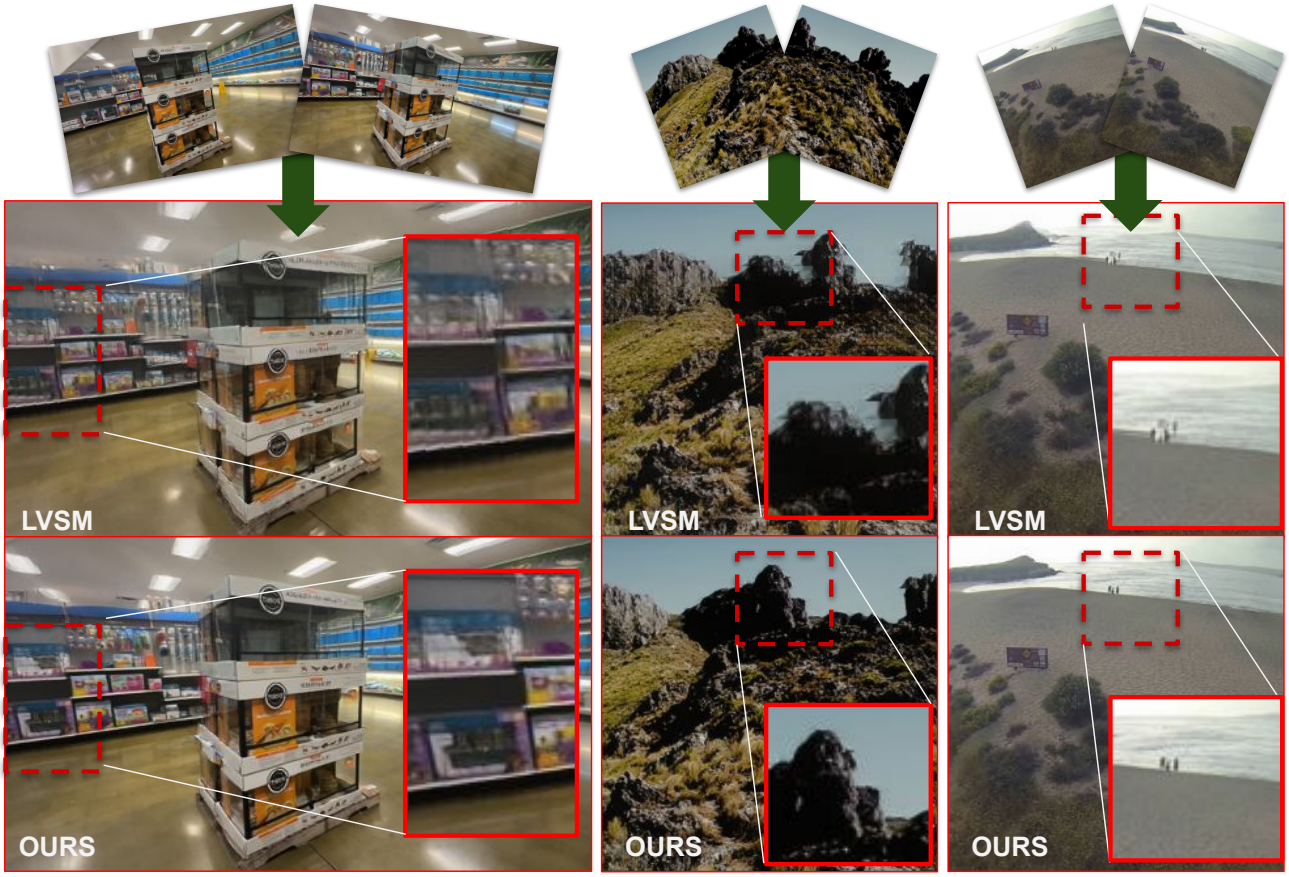


Figure 1. **Overview.** Our method performs feed-forward novel-view synthesis from a series of input images, such as the pairs shown above. We demonstrate strong results in terms of quality and generalization capacity, performing well across a variety of common novel-view synthesis datasets, including scenes that are out-of-distribution.

Abstract

Large transformer-based models have made significant progress in generalizable novel view synthesis (NVS) from sparse input views, generating novel viewpoints without the need for test-time optimization. However, these models are constrained by the limited diversity of publicly available scene datasets, making most real-world (in-the-wild) scenes out-of-distribution. To overcome this, we incorporate synthetic training data generated from diffusion models, which improves generalization across unseen domains. While synthetic data offers scalability, we identify artifacts introduced during data generation as a key bottleneck affecting reconstruction quality. To address this, we propose a token disentanglement process within the transformer architecture, enhancing feature separation and ensuring more effective learning. This refinement not only improves re-

construction quality over standard transformers but also enables scalable training with synthetic data. As a result, our method outperforms existing models on both in-dataset and cross-dataset evaluations, achieving state-of-the-art results across multiple benchmarks while significantly reducing computational costs.

1. Introduction

Novel view synthesis (NVS) [19, 26] is a well-studied and important problem in computer vision, where the task is to generate unseen perspectives of a scene from a given set of images. Many approaches utilize volumetric [2, 5, 26, 27] or differentiable rendering [19] to optimize for each scene individually, achieving high-quality NVS from arbitrary viewpoints. More recently, advancements have enabled training a single model that generalizes to novel scenes without requiring per-scene optimization. Most existing methods address NVS by incorporating hand-crafted 3D priors and architectural biases [4, 15, 38]. While these design choices provide structure, they limit scalability with data and hinder generalization.

Recently, Large View Synthesis Model (LVSM) [18] proposed a promising foundation for an NVS model scalable with large datasets. LVSM introduces an architecture that doesn't require 3D inductive biases for scene reconstruction. It employs a decoder-only transformer architecture that achieves state-of-the-art results by a significant margin, with the performance improving with increased compute. However, we observed during our experiments that the decoder-only design causes an inherent feature alignment problem which causes the target and source features to look similar at all layers. Thus, part of the transformer's computational capacity is spent modifying source token information that is ultimately discarded at the end of the transformer block, reducing efficiency. This design choice also makes LVSM susceptible to unwanted noise or compression artifacts that may be present in the source views. In addition, we noticed that LVSM presents limited cross-domain performance when tested on datasets outside the training dataset domains.

Moreover, these issues are not unique to LVSM; many NVS models face similar challenges due to data scarcity in 3D vision. All existing multi-view 3D scene datasets [23, 24, 48] combined contain fewer than 100,000 scenes, severely limiting the performance of NVS models on in-the-wild cases beyond the training distribution. One possible solution for alleviating this 3D data scarcity is using synthetic data from generative models. Recent research has explored adapting pre-trained image [32, 33] and video diffusion models [13, 14] for multi-view dataset generation [10, 25, 35, 43]. However, previous feed-forward models trained using synthetic data perform worse than those trained with real data. We hypothesize that the inability of synthetic data to improve reconstruction quality stems from two types of degradation artifacts in scenes generated by diffusion models [14, 28, 37] (1) artifacts influenced by the initial noise of the diffusion process and (2) artifacts introduced during decoding, as most diffusion-based scene synthesis models operate in latent space and rely on a diffusion VAE [32]. We address both issues, leading to improved performance when using synthetic data. We provide a detailed explanation of our data pipeline in Section 4.2.

In this work, we tackle a key challenge in developing a feed-forward NVS model that performs well on out-of-distribution data – the need for a scalable and efficient transformer-based NVS architecture. We introduce the Token Distentangled (Tok-D) transformer block, which applies layer-wise modulation of source and target tokens, explicitly distinguishing the two at each layer. These model modifications improve out-of-distribution training, which introduces the possibility of training on synthetic data. We use the CAT3D model to generate a large dataset of synthetic multi-view samples. We then employ a novel data generation strategy that significantly improves the quality of these synthetic samples. We show that the Tok-D transformer block can be trained with synthetic data augmentation, unlike the baseline LVSM method which suffers from the inclusion of synthetic data.

- We enhance the scalability of transformer architectures for NVS, enabling more efficient modeling.
- We introduce a new training scheme that is less susceptible to artifacts from synthetic data.
- We improve the training efficiency of transformer for NVS by introducing a new transformer block.
- Our approach achieves state-of-the-art results across multiple benchmarks for scene level NVS.

2. Related Works

2.1. Offline Novel View Synthesis

The advent of neural rendering in recent years has substantially improved the quality of NVS. Early neural scene representations focused on the 4D plenoptic function [11, 22] that represents the lightfield of a scene [1, 36, 38]. Other methods modeled the geometry of the scene (e.g. as a signed distance function) separately from material properties [39, 44]. Either way, a differentiable rendering process was used to render these neural representations into 2D images [26]. Most of these methods focused on fitting neural fields to sparse observations of a scene at test time—we refer to this as test-time or offline optimization. There is a substantial amount of heterogeneity in these methods, both in terms of the rendering method and the scene repre-

*Equal contribution. Nair designed the methodology, conducted pre-rebuttal experiments, and drafted the initial manuscript. Kaza helped advise the project, led the rebuttal, and conducted camera-ready experiments.

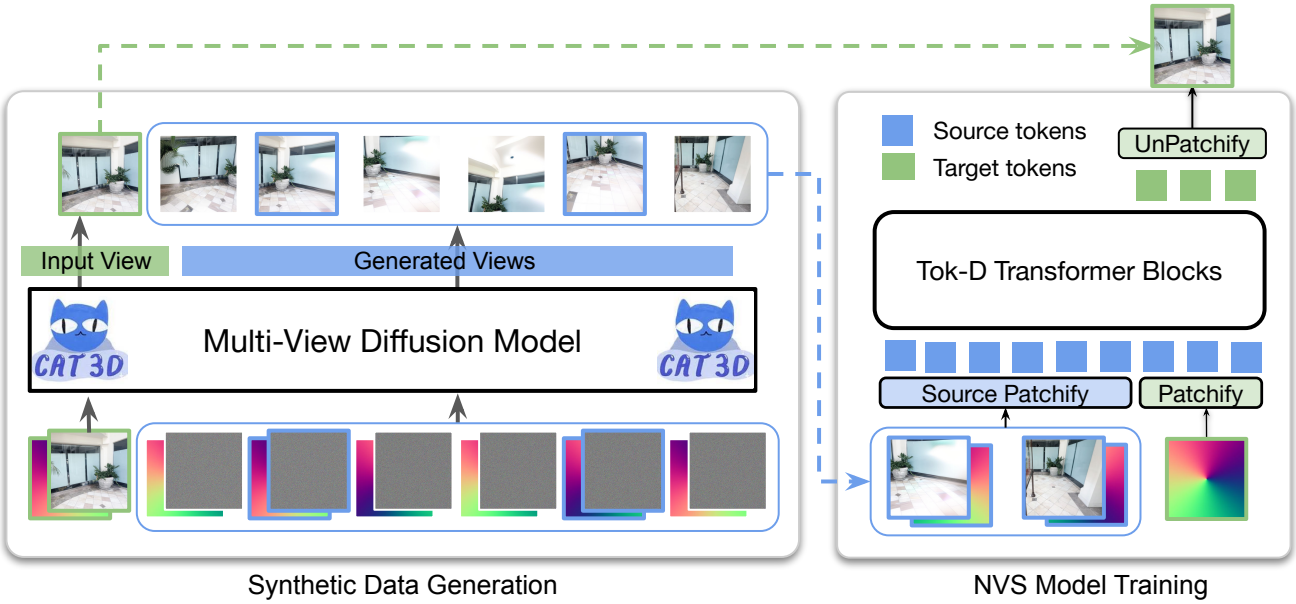


Figure 2. **An illustration of the architecture.** We use CAT3D, a multi-view diffusion model, to generate synthetic views conditioned on random spline camera trajectories and a random image. From the two random views form the generated views as the source views and the input conditioning view to be the target of our large reconstruction network. Our large reconstruction model uses a special transformer block which we name Tok-D Transformer. When real data is available, we just use the reconstruction transformer.

sentation used. Multi-layer perceptrons (MLPs) [26], voxels [9, 25], hashing-based representations [3, 27], triplanes [5], and, most recently, Gaussian splats [16, 19, 20, 30] have been used as scene representations. These methods have trade-offs between reconstruction quality, training time, inference time, memory/space requirements, capacity to model view-dependent effects, etc. Some of these offline methods can even fit dynamic scenes. These test-time optimization methods demonstrate compelling results given the sparsity of the observations provided. However, they often struggle to incorporate priors learned from larger datasets.

2.2. Online Novel View Synthesis

Sometimes referred to as "feed-forward" or "generalizable" NVS models, these methods attempt to directly produce 3D representations from input images. Early efforts include the image-based rendering-inspired IBRNet [40], which directly produces 2D images based on epipolar correspondences on the viewing ray. The Large Reconstruction Model (LRM) [15] family of methods attempt to produce a triplane that represents an object, in some cases with near-real time performance. PixelSplat [4], MVSplat [4], and GS-LRM [46] attempt to predict 3DGS [19] representations, which exploit the sparse Gaussian splat representation and fast rasterization to achieve quasi-interactive inference. These methods are trained on large datasets of real-world scenes, which helps them outperform even test-time optimization methods. Quark [8] couples an easily-rasterizable layered depth map representation with a render-and-refine strategy to achieve state-of-the-art quality at a much higher resolution. Other efforts in this space include GPNR [38] and SRT [34], which are parameterized in a similar fashion to IBRNet [40] and attempt to scale up the image and ray transformers. LRF [21] attempts to perform 3D reconstruction in the latent space of a VAE, bypassing learning 3D representation altogether [47]. Finally, the LVSM [18] removes all 3D priors by simply using one transformer to perform NVS. LVSM performs favorably compared to both geometry-free and geometry-based feed-forward models.

2.3. Synthetic Data

Recent efforts have leveraged synthetic data to train existing feed-forward NVS methods and investigate its efficacy as a training dataset. However, it is important to note that the synthetic data in many of these efforts are generated procedurally from systems like Blender, whereas ours are generated from a multi-view diffusion model. Two recent works LRM-Zero [42] and MegaSynth [17] are examples of models trained either entirely or mostly on procedurally generated synthetic data. In LRM-Zero, they demonstrate that the LRM model can be trained entirely on synthetic data. However, the synthetic-data-only model shows a substantial decrease in reconstruction quality compared to the real-world-data equivalent. Improving training data diversity using synthetic data for 4D generation has also been explored in CAT4D [41].

3. Background

LVSM is a feed-forward NVS method that has no 3D inductive bias. Since our model builds upon its architecture,

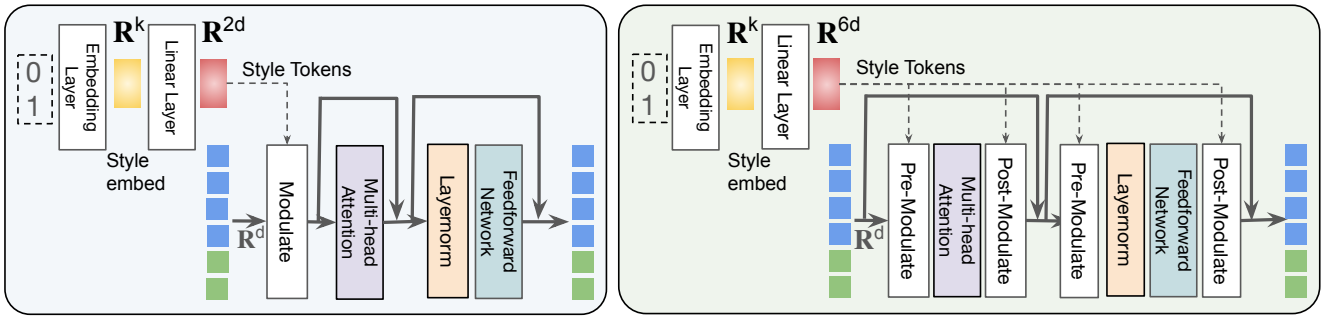


Figure 3. **An illustration of the Tok-D transformer block.** Our transformer blocks that differentiates between source and target tokens. Tok-D transformer modulates the input to all transformer blocks. Tok-D plus transformer modulates the attention and MLP layers.

we outline the details here for clarity. Where i denotes the image index and j denotes the token index, source images patches are written as $I^s_{ij} \in R^{p\times p\times 3}$, source Plücker coordinates patches $P^s_{ij} \in R^{p\times p\times 6}$, and target Plücker coordinates $P^t_j \in R^{p\times p\times 6}$. The source images and plücker embeddings are tokenized together using a linear layer embedder.

$$S_{ij} = Linear([I^s_{ij}, P^s_{ij}]) \tag{1}$$

The target Plücker coordinates are also embedded using a linear layer.

$$T_{ij} = Linear(P^t_{ij}) \tag{2}$$

Finally, the transformer network is trained to reconstruct the target output tokens O^t_j from the Plücker patch embeddings.

$$O^t_j = M(T_j|S_{ij}) \tag{3}$$

The target output tokens are detokenized using a linear layer which is converted to target image embeddings $T_j \in R^{p\times p\times 3}$

$$T_j = Linear([O^t_j]) \tag{4}$$

The target patches are unpatchified to get the target image $T \in R^{H\times W\times 3}$ (see Figure 2). The training is supervised using MSE loss and perceptual loss designed to reconstruct.

Transformer Block Consider a transformer block at layer l, which includes a *Multi-head Self Attention* layer (SelfAttn_l), a Feed-forward network (FFN_l), and a Layer Norm operation (LN_l). For an input $[\mathbf{x}^s_l, \mathbf{x}^t_l]$, where $\mathbf{x}^s_l$ and $\mathbf{x}^t_l$ represent the source and target tokens, the data flow as follows:

$$\begin{aligned}[\mathbf{x}^s_l, \mathbf{x}^t_l] &= [\mathbf{x}^s_l, \mathbf{x}^t_l] + \text{SelfAttn}_l([\mathbf{x}^s_l, \mathbf{x}^t_l]) \\ [\mathbf{x}^s_l, \mathbf{x}^t_l] &= [\mathbf{x}^s_l, \mathbf{x}^t_l] + \text{FFN}_l(LN_l([\mathbf{x}^s_l, \mathbf{x}^t_l])).\end{aligned} \tag{5}$$

Given the basic self attention based transformer blocks in LVSM. At the end of the optimization process there arises a need for all token outputs of a particular layer to be aligned since they are processed by the same set of weights. Hence, LVSM inherently has a chance to infuse noise or atifacts that maybe present in the source images to the target. Moreover this alignment also causes some part of the computational power of the model being used to model source token information although those tokens are discarded at the last layer. Hence, we call for a need to distinguish between the source and target tokens of the transformer network.

4. Method

Our proposed method consists of two major contributions. First, our *Token-Disentangled (Tok-D)* transformer block is specialized for NVS and distinguishes information from the source and target views, leading to more efficient allocation of representation capacity. Second, to address the scarcity of multi-view data, we generate synthetic data using CAT3D [10] and propose a model training scheme that is robust to artifacts in this synthetic data. In this section, we describe each component in detail.

4.1. Token-Disentangled Transformer

In LVSM, the transformer blocks process source and target tokens in the same manner, even though the source consists of images and Plücker rays, while the target includes only Plücker rays. Additionally, source and target image quality can differ when training with synthetic data. To address this, we introduce the *Token-Disentangled (Tok-D) Transformer* block (see Figure 3), which enables differentiated processing of source and target tokens through modulation. The Tok-D Transformer uses an indicator variable (δ), where $\delta = 1$ for target tokens and $\delta = 0$ for source tokens, to modulate tokens based on their origin. This mechanism extracts distinct style vectors and computes specific scale and bias parameters for each layer and token type, allowing for precise and adaptive token modulation.

$$\begin{aligned}\mathbf{style} &= \text{Linear}(\text{Embed}(\delta)) \\ \text{Mod}_l(\mathbf{x}) &= (1+\sigma_l)\mathbf{x} + \mu_l, \text{where } [\sigma_l, \mu_l] = \text{Linear}_l(\mathbf{style}) \\ [\mathbf{x}^s_l, \mathbf{x}^t_l] &= \text{Mod}^{s,t}_l([\mathbf{x}^s_l, \mathbf{x}^t_l]) = [\text{Mod}^s_l(\mathbf{x}^s_l), \text{Mod}^t_l(\mathbf{x}^t_l)]\end{aligned} \tag{6}$$

Modulating the input of each transformer block improves perforamnce. Drawing inspiration from DiT [29], we

extend this modulation to the Attention and MLP layers, achieving further improvements. This modulation is termed *pre-modulation* if applied before a layer and *post-modulation* if after. Pre-modulation includes both scaling and shifting, and post-modulation involves only scaling.

$$
\begin{aligned}
[\hat{\mathbf{x}}_l^s, \hat{\mathbf{x}}_l^t] &= \text{Mod}_{l1}^{s,t}([\mathbf{x}_l^s, \mathbf{x}_l^t]) \\
[\mathbf{x}_l^s, \mathbf{x}_l^t] &= [\mathbf{x}_l^s, \mathbf{x}_l^t] + [\sigma_{l1}^s, \sigma_{l1}^t] \odot \text{SelfAttn}([\hat{\mathbf{x}}_l^s, \hat{\mathbf{x}}_l^t]) \\
[\hat{\mathbf{x}}_l^s, \hat{\mathbf{x}}_l^t] &= \text{Mod}_{l2}^{s,t}([\mathbf{x}_l^s, \mathbf{x}_l^t]) \\
[\mathbf{x}_l^s, \mathbf{x}_l^t] &= [\mathbf{x}_l^s, \mathbf{x}_l^t] + [\sigma_{l2}^s, \sigma_{l2}^t] \odot \text{FFN}_l(\text{LN}_l([\hat{\mathbf{x}}_l^s, \hat{\mathbf{x}}_l^t]))
\end{aligned} \tag{7}
$$

where $\odot$ denotes element-wise multiplication which scales the corresponding source and target tokens.

Our Tok-D transformer block enhances the distinction between source and target tokens, as reflected in their distinct feature representations (Figure 6, Section 5.4). This specialization highlights the superior representational capacity of our model. Furthermore, when trained on synthetic data (Section 4.2), out-of-distribution artifacts can introduce quality disparities between source and target tokens. By leveraging its token-aware architecture, our model demonstrates greater robustness to these artifacts, resulting in improved performance, as shown in Section 5.3.

4.2. Synthetic Data Generation & Training Scheme

Training a naive transformer model with synthetic data can lead to degraded performance rather than improvement due to two key factors: (1) The model struggles to distinguish between tokens from source images and target images, allowing artifacts from one to propagate into the other during alignment. (2) The model is trained to generate novel views from sparse input views, and if the target is a synthetic image with artifacts, it may learn a distribution biased toward unrealistic images. While these issues might not arise with perfect synthetic data, in-practice synthetic datasets often contain noise, making the model vulnerable to errors through either mechanism. However, for image-to-multiview synthesis models like CAT3D, we propose a simple yet effective solution: assigning the conditioned image as the target view and the generated views as input views.

Formally let I_c, C_c denote the input image and camera conditioning used for the multiview diffusion model. We sample additional random spline camera trajectory poses C_{tgt} relative to this particular view, and use the state-of-the-art multi-view diffusion model CAT3D to generate the target views I_{src} conditioned on the input conditioning and target poses

$$I_{gen} \sim DM(I_{gen}|C_{gen}, C_c, I_c) \tag{8}$$

Here DM represents inferencing through the state of the art diffusion model, After obtaining the generated views, we sample 2 generated views I_{src}

$$I_{src}, C_{src} \sim I_{gen}, C_{gen} \tag{9}$$

and their camera poses as the source images I_{src}, C_{src} and utilize the conditioned image and its camera as the target I_c, C_c. Sampling the source and target images this way forces the transformer to always generate a realistic image, making our model robust to artifacts from synthetic data.

5. Experiments

5.1. Implementation Details

Training details We perform all experiments on 8 H100 GPUs. We use the AdamW optimizer with β parameters 0.9 and 0.95, and we use weight decay with a rate of 0.05 for all layers except the normalization layers. Moreover, we use a linear learning rate scheduler with with a peak learning rate of $2e^{-4}$, and a warmup of 2500 iterations. In total, all experiments have $100k$ training iterations. In addition, we use exponential moving averaging (EMA) with a rate of 0.99 for stabilizing the training process. Although previous works required gradient clipping for a stable training process, our training processes were smooth without a need for an explicit gradient clipping.

Training and Evaluation Datasets For scene-level synthesis model training, we use Re10K [48], ACID [24] and DL3DV [23] with their originally released train and test splits. We also perform an experiment where the model is trained together with a mix of all of these datasets. For scene-level synthesis, we follow LVSM and train using 2 input views and test using 6 target views fed one at a time. For DL3DV dataset evaluation, we choose the farthest camera from a randomly selected target view as the input view. The training and evaluation of DL3DV dataset for in distribution metrics is done using 2 input views and 2 target views. For cross dataset testing, we use 2 input views and 6 target views for DL3DV dataset. We use a batch size of 64 for our experiments.

Synthetic Data For generating the synthetic data we use the state-of-the-art 3D generation model CAT3D. CAT3D was trained using a single scene dataset Re10K and three object-based datasets: Objaverse [7], MVImgNet [45] and Co3D [31]. To create synthetic data, we use two variants: one with 1 conditioning view and 7 generated views, and another with 3 conditioning views and 5 generated views. We match the focal lengths of Re10K and DL3DV during generation. For the camera trajectory, we sample a random spline trajectory with a random position rotation matrix, converting it into ray maps before passing it into the network. As CAT3D is originally trained with a resolution of 512, we convert the images and camera parameters to a resolution of 256 before passing them through our network.

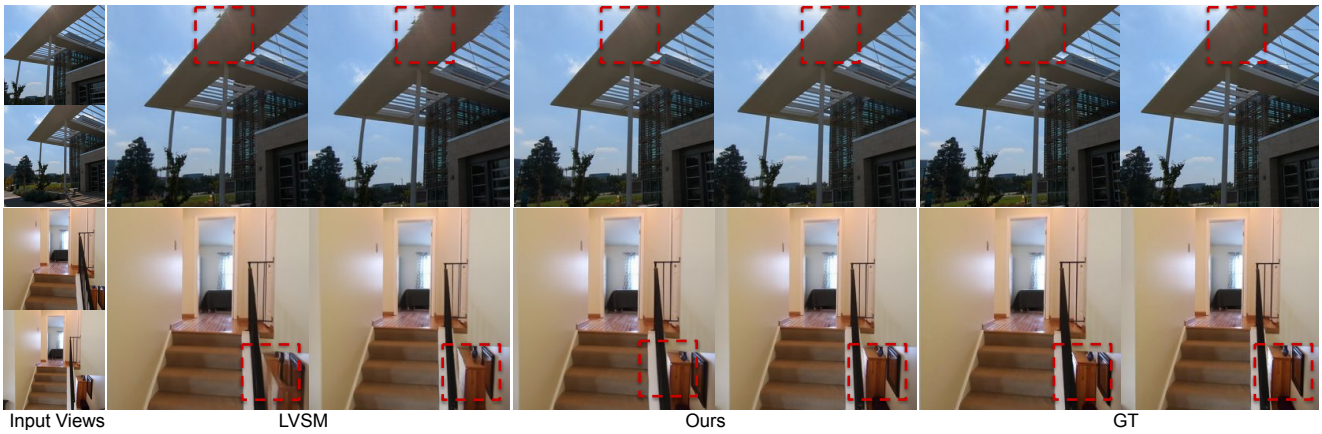


Figure 4. **Qualitative results on in-distribution datasets.** We illustrate the cases Tok-D transformer works better than LVSM. We notice that we obtain substantial improvement in cases where the novel views needs to reconstruct regions present only in one of the views as shown in the highlighted regions in the images. The results presented here are taken from our in-distribution trained model. We present two diffrent views to show that this problem is persistent across views.

Table 1. **Quantitative comparisons for in-distribution scene synthesis at 256 resolution.** LVSM and our method are trained with a batch size of 64. LVSM results are taken from the original paper rather than our re-implementation. Our method outperforms the previous SOTA method across all exisiting datasets. (, ,) denotes the first, second and third best results.

Method	Venue	RealEstate10k [48]			ACID [24]			DL3DV [23]		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
GPNR [38]	CVPR'23	24.11	0.793	0.255	25.28	0.764	0.332	-	-	-
PixelSplat [4]	CVPR'24	25.89	0.858	0.142	28.14	0.839	0.533	-	-	-
MVSplat [6]	ECCV'25	26.39	0.869	0.128	28.25	0.843	0.144	17.54	0.529	0.402
DepthSplat [43]	CVPR'25	27.44	0.887	0.119	-	-	-	19.05	0.610	0.313
LVSM [18]	ICLR'25	28.89	0.894	0.108	29.19	0.836	0.095	19.91	0.600	0.273
Ours		30.02	0.919	0.058	29.47	0.846	0.086	21.55	0.643	0.208

5.2. Scene Synthesis

We evaluate our method qualitatively and quantitatively for scene synthesis using very recent feed-forward methods GPNR, PixelSplat, MVSplat, DepthSplat and LVSM. These methods were chosen because they outperform conventional approaches in 2-view reconstruction. Quantitative results are shown in Table 1. We observe that Tok-D-Plus outperforms LVSM by 1.2 dB on the Re10K evaluation benchmark when both models are trained with 8 GPUs. Furthermore, despite using only 8 GPUs, our method still surpasses LVSM trained with 64 GPUs by a margin of 0.2 dB. Moreover we obtain an improvement of 1.6dB over LVSM in a more diverse scene-level dataset, DL3DV [23] dataset as well. We also observe that our performance improvement is 0.2 in ACID dataset. We emphasize that this happens because ACID has a relatively smaller training and testing set and the dataset is generally clean and easier to reconstruct. We also provide the corresponding qualitative comparisons on Re10K and DL3DV dataset in Figure 4 . Comparing the main results we find that our method usually outperforms LVSM when the generated content is only visible in one of the source views. When the camera is far from both views and the information is present only in one of the views, our method is still able to extract the relevant content from the corresponding input image. As can be seen from rows 1 and 2, the reconstruction form LVSM fails to reconstruct objects present in only one of the views, whereas Tok-D transformer can effectively reconstruct these regions.

5.3. Cross-Dataset Scene Synthesis

To analyze the generalization capacity of our method, we evaluate our method trained with Re10K dataset on two different datasets: ACID and DL3DV [23]. ACID is a dataset with aerial views similar to Re10K. DL3DV [23] is a diverse dataset comprising natural scenes and various indoor and outdoor settings. The scene geometry and appearance of DL3DV [23] is very different from Re10k. We test the Re10K and ACID datasets at a resolution of 256×265. For testing on DL3DV [23], we choose a resolution of 256×448 to maintain the original aspect ratio in the DL3DV [23] dataset and well as maintain consistent evaluation settings with DepthSplat. We choose 2 source views and 6 target views for all of these datasets. Looking closely at the quantitative results on Table 1 and Table 2, we find that the model trained on Re10K underperformed the in-distribution trained model by a small margin. The drop is higher in the case of DL3DV due to resolution and diversity differences in the datasets. Next we add a small portion of synthetic

Figure 5. **Out-of-distribution Evaluation:** We evaluate our the version of our method fine-tuned on synthetic data and LVSM on DL3DV and ACID (i.e. out-of-distribution datasets). We also evaluate the model with resolutions that were not used during training. We notice that LVSM's visual quality degrades when substantial camera motion reveals previously-occluded regions.

Table 2. **Quantitative comparisons for scaling up with synthetic data.** We evaluate LVSM and our method, which are both trained with a batch size of 64. A mixture of synthesized DL3DV and Re10K data is used for the synthetic tab. For MVSplat and DepthSplat we include the numbers reported in their papers

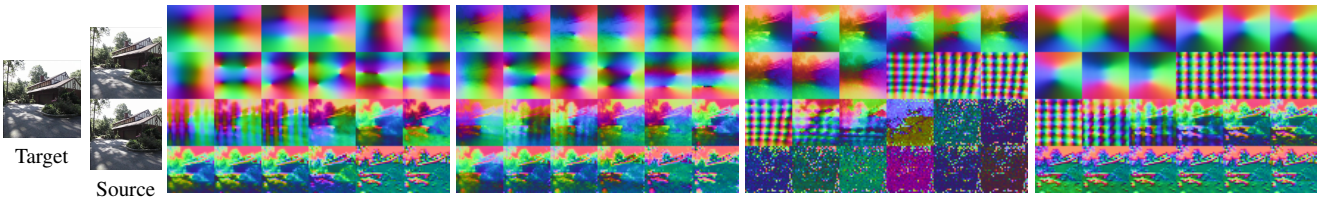
Method	Training Dataset		RealEstate10k [48]			ACID [24]			DL3DV [23]		
	Re10K[48]	Synthetic	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
MVSplat [6]	✓		26.39	0.869	0.128	28.15	0.147	0.841	17.72	0.534	0.371
Depthsplat [43]	✓		27.44	0.887	0.119	-	-	-	18.90	0.640	0.317
LVSM [18]	✓		28.89	0.894	0.108	28.29	0.809	0.104	20.52	0.621	0.223
LVSM [18]	✓	✓	28.49	0.892	0.070	28.16	0.802	0.107	20.21	0.595	0.240
Ours	✓		30.02	0.910	0.058	29.31	0.838	0.091	21.18	0.652	0.205
Ours	✓	✓	29.97	0.920	0.058	29.37	0.839	0.091	21.27	0.657	0.202

data comprising about half the size of Re10K dataset and perform training with the new framework. We also retrain LVSM for the same setting. We find that *LVSM's performance drops rather than improving when synthetic data is added*. We emphasize that this arises due to the introduction of artifacts during feature alignment. In contrast, we observe an improvement in quality on our method when a small amount of synthetic data is added.

5.4. Analysis and Discussion

Visualization of source and target features. To visually illustrate the representation alignment problem mentioned in the previous sections, we visualize the 3 channel PCA of each transformer block output after unpatchifying for all 24 layers of LVSM and our method in Figure 6. The first row shows the first 6 layer outputs, second row shows layer 6-12, and so on. We can see that for a particular scene the source and target layer tokens are aligned at all layers even though the training objective is to reconstruct the target. This causes inefficient usage of the transformer parameters to maintain the source information throughout the layers. Moreover this also makes the model prone to *noise in the source data*. However, with our Tok-D transformer there is no alignment and the source information is infused much earlier, leaving more room for the transformer blocks to reconstruct the target. Another important observation is that although both source image and Plücker coordinates are fed as input to the source, the source tokens look similar to the Plücker coordinates. Whereas in our case the image components in the source PCA components leading to much more effective information extraction from each source token.

Impact of additional real data. Incorporating synthetic data into the training process facilitates the introduction of diverse scenes and camera motion, enhancing model generalizability. While the proposed Tok-D transformer demon-



(a) LVSM Target PCA (b) LVSM Source PCA (c) Ours Source PCA (d) Ours Target PCA

Figure 6. **A visualization of the principal components of transformer layer outputs for source and target of LVSM**. The 24 images in each subfigure show the layer output of each layer of the transformer. LVSM features for source and target images looks similar even though the source is conditioned with image and Plücker coordinates and target is conditioned with Plücker coordinates alone. This leads to inefficient transformer usage requiring explicit alignment of source and target features across different layers

Table 3. **Ablation studies on scaling up with more real data.** Although including synthetic data in training is helpful for improving quality, including additional real data significantly improves reconstruction quality.

Method	Training Dataset		RealEstate10k [48]			ACID [24]			DL3DV [23]		
	Re10K [48]+ Synthetic	DL3DV [23]	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
LVSM [18]	✓		28.49	0.892	0.070	28.16	0.802	0.107	20.21	0.595	0.240
LVSM [18]	✓	✓	28.10	0.892	0.073	28.79	0.826	0.096	21.37	0.665	0.196
Ours	✓		29.97	0.920	0.058	29.37	0.839	0.091	21.27	0.657	0.202
Ours	✓	✓	29.78	0.917	0.0604	30.13	0.857	0.082	23.14	0.726	0.156

Table 4. **Ablation analysis** We analyze the performance improvement of our design choices. *Pre* and *Post* demonstrate the effects of including or not including pre/post-modulation.

Pre	Post	Whole	Attn	MLP	PSNR ↑	SSIM ↑	LPIPS ↓
					28.50	0.893	0.070
✓		✓			29.69	0.911	0.063
✓			✓	✓	28.51	0.894	0.070
✓	✓		✓	✓	30.02	0.918	0.058

strates reduced sensitivity to synthetic data artifacts and increased generative diversity, its photorealistic reconstruction performance remains comparable to the baseline model trained solely on real data. To investigate the impact of augmenting the training dataset with additional real data, we integrated the DL3DV dataset into the existing experimental setup. This modification resulted in a significant improvement in photorealistic reconstruction, as evidenced by a substantial increase in PSNR on the ACID dataset. Furthermore, the relative performance gains observed with our model, compared to LVSM, were considerably greater, suggesting a reduced susceptibility to noise.

5.5. Ablation Studies

We analyze the impact of various design choices in the network. Specifically, we examine three aspects: (1) The effect of modulation in different parts of the network, (2) The role of EMA in performance, (3) Number of input views.

Impact of modulation at different locations of Tok-D transformer. We examine the effect of modulating different parts of the network. For this, we consider four different cases. We present the corresponding results in Table 4. Having a common modulation premodulation worked better than separate premodulation for both layers.

Impact of EMA. We also observe that performing Exponential moving average (EMA) [12] during training results in a performance boost for the base model. For the sake of consistency, we show the results of our model and our re-implementation of LVSM with 1024 channels trained with and without EMA in Table 5.

Table 5. **Effect of EMA on runtime performance and quality.** Comparison performed on Re10k.

Method	Train (ms)	Render (ms)	GFLOPs	No EMA			With EMA		
				PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
LVSM-1024	706.1	171.6	2896.88	27.68	0.88	0.077	28.65	0.90	0.070
Ours	734.6	174.4	2900.78	28.75	0.90	0.064	30.02	0.92	0.058

Table 6. **Effect of adding more source views**. Our method works well as additional source views are introduced.

Method	2 views			4 views			8 views		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
Ours	30.02	0.92	0.058	31.51	0.94	0.048	33.09	0.94	0.042

Impact of number of source frames. Our model scales with the number of input views and results in better reconstruction quality when more input views are fed to the model to the model as presented in Table 6.

6. Conclusion

In this paper, we introduce a new approach to scaling up NVS by addressing two key limitations in existing models: efficiency and diversity. To enhance the efficiency of transformer-based NVS models, we propose the Token-Disentangled (Tok-D) Transformer, which reduces redundancies and improves data efficiency, enabling higher reconstruction quality with less compute. Additionally, the Tok-D Transformer mitigates training artifacts through its disentangling property, allowing for effective scaling using synthetic data. Incorporating synthetic data into training significantly improves cross-dataset performance compared to existing models. By integrating the Tok-D Transformer and synthetic data, we achieve state-of-the-art results across three large-scale NVS benchmarks, surpassing previous methods with lower computational cost and by a substantial margin.

References

[1] Benjamin Attal, Jia-Bin Huang, Michael Zollhöfer, Johannes Kopf, and Changil Kim. Learning neural light fields with ray-space embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19819–19829, 2022. 2

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19697–19705, 2023. 2

[3] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023. 3

[4] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19457–19467, 2024. 2, 3, 6

[5] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 3

[6] 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 *European Conference on Computer Vision*, pages 370–386. Springer, 2024. 6, 7

[7] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 5

[8] John Flynn, Michael Broxton, Lukas Murmann, Lucy Chai, Matthew DuVall, Clément Godard, Kathryn Heal, Srinivas Kaza, Stephen Lombardi, Xuan Luo, et al. Quark: Real-time, high-resolution, and general neural view synthesis. *ACM Transactions on Graphics (TOG)*, 43(6):1–20, 2024. 3

[9] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5501–5510, 2022. 3

[10] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin Brualla, Pratul Srinivasan, Jonathan Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *Advances in Neural Information Processing Systems*, 37:75468–75494, 2024. 2, 4

[11] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, page 43–54, New York, NY, USA, 1996. Association for Computing Machinery. 2

[12] David Haynes, Steven Corns, and Ganesh Kumar Venayagamoorthy. An exponential moving average algorithm. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–8. IEEE, 2012. 8

[13] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*, 2022. 2

[14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[15] 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 *The Twelfth International Conference on Learning Representations*, 2024. 2, 3

[16] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, page 1–11. ACM, 2024. 3

[17] Hanwen Jiang, Zexiang Xu, Desai Xie, Ziwen Chen, Haian Jin, Fujun Luan, Zhixin Shu, Kai Zhang, Sai Bi, Xin Sun, Jiuxiang Gu, Qixing Huang, Georgios Pavlakos, and Hao Tan. Megasynth: Scaling up 3d scene reconstruction with synthesized data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16441–16452, 2025. 3

[18] 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. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 3, 6, 7, 8

[19] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2, 3

[20] Shakiba Kheradmand, Delio Vicini, George Kopanas, Dmitry Lagun, Kwang Moo Yi, Mark Matthews, and Andrea Tagliasacchi. Stochasticsplats: Stochastic rasterization for sorting-free 3d gaussian splatting, 2025. 3

[21] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013. 3

[22] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, page 31–42, New York, NY, USA, 1996. Association for Computing Machinery. 2

[23] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024. 2, 5, 6, 7, 8

[24] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14458–14467, 2021. 2, 5, 6, 7, 8

[25] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances*

in Neural Information Processing Systems, 33:15651–15663, 2020. 2, 3

[26] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, 2020. 2, 3

[27] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022. 2, 3

[28] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. 2

[29] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 4

[30] Lukas Radl, Michael Steiner, Mathias Parger, Alexander Weinrauch, Bernhard Kerbl, and Markus Steinberger. StopThePop: Sorted Gaussian Splatting for View-Consistent Real-time Rendering. *ACM Transactions on Graphics*, 4 (43), 2024. 3

[31] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10901–10911, 2021. 5

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[33] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 2

[34] Mehdi SM Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, et al. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6229–6238, 2022. 3

[35] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3d generation. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[36] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *Advances in Neural Information Processing Systems*, 34: 19313–19325, 2021. 2

[37] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2

[38] Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *European Conference on Computer Vision*, pages 156–174. Springer, 2022. 2, 3, 6

[39] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *Advances in Neural Information Processing Systems*, 34:27171–27183, 2021. 2

[40] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2021. 3

[41] Rundi Wu, Ruiqi Gao, Ben Poole, Alex Trevithick, Changxi Zheng, Jonathan T Barron, and Aleksander Holynski. Cat4d: Create anything in 4d with multi-view video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26057–26068, 2025. 3

[42] Desai Xie, Sai Bi, Zhixin Shu, Kai Zhang, Zexiang Xu, Yi Zhou, Soren Pirk, Arie Kaufman, Xin Sun, and Hao Tan. LRM-zero: Training large reconstruction models with synthesized data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 3

[43] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16453–16463, 2025. 2, 6, 7

[44] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces, 2021. 2

[45] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9150–9161, 2023. 5

[46] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. *European Conference on Computer Vision*, 2024. 3

[47] Chaoyi Zhou, Xi Liu, Feng Luo, and Siyu Huang. Latent radiance fields with 3d-aware 2d representations. In *The Thirteenth International Conference on Learning Representations*, 2025. 3

[48] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Trans. Graph.*, 37(4), 2018. 2, 5, 6, 7, 8