LVSM: A LARGE VIEW SYNTHESIS MODEL WITH MINIMAL 3D INDUCTIVE BIAS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose the Large View Synthesis Model (LVSM), a novel transformer-based approach for scalable and generalizable novel view synthesis from sparse-view inputs. We introduce two architectures: (1) an encoder-decoder LVSM, which encodes input image tokens into a fixed number of 1D latent tokens, functioning as a fully learned scene representation, and decodes novel-view images from them; and (2) a decoder-only LVSM, which directly maps input images to novel-view outputs, completely eliminating intermediate scene representations. Both models bypass the 3D inductive biases used in previous methods—from 3D representations (e.g., NeRF, 3DGS) to network designs (e.g., epipolar projections, plane sweeps)—addressing novel view synthesis with a fully data-driven approach. While the encoder-decoder model offers faster inference due to its independent latent representation, the decoder-only LVSM achieves superior quality, scalability, and zero-shot generalization, outperforming previous state-of-the-art methods by 1.5 to 3.5 dB PSNR. Comprehensive evaluations across multiple datasets demonstrate that both LVSM variants achieve state-of-the-art novel view synthesis quality. Notably, our models surpass all previous methods even with reduced computational resources (1-2 GPUs). Please see our website for more details: https://lvsm-web.github.io/.

1 INTRODUCTION

Novel view synthesis is a long-standing challenge in vision and graphics. For decades, the community has generally relied on various 3D inductive biases, incorporating 3D priors and handcrafted structures to simplify the task and improve synthesis quality. Recently, NeRF, 3D Gaussian Splatting (3DGS), and their variants (Mildenhall et al., 2020; Barron et al., 2021; Müller et al., 2022; Chen et al., 2022; Xu et al., 2022; Kerbl et al., 2023; Yu et al., 2024) have significantly advanced the field by introducing new inductive biases through carefully designed 3D representations (e.g., continuous volumetric fields and Gaussian primitives) and rendering equations (e.g., ray marching and splatting with alpha blending), reframing view synthesis as the optimization of the representations using rendering losses on a per-scene basis. Other methods have also built generalizable networks to estimate these representations or directly generate novel-view images in a feed-forward manner, often incorporating additional 3D inductive biases, such as projective epipolar lines or plane-sweep volumes, in their architecture designs (Wang et al., 2021a; Yu et al., 2021; Chen et al., 2021; Suhail et al., 2022b; Charatan et al., 2024; Chen et al., 2024).

While effective, these 3D inductive biases inherently limit model flexibility, constraining their adaptability to more diverse and complex scenarios that do not align with predefined priors or handcrafted structures. Recent large reconstruction models (LRMs) (Hong et al., 2024; Li et al., 2023; Wei et al., 2024; Zhang et al., 2024) have made notable progress in removing architecture-level biases by leveraging large transformers without relying on epipolar projections or plane-sweep volumes, achieving state-of-the-art novel view synthesis quality. However, despite these advances, LRMs still rely on representation-level biases—such as NeRFs, meshes, or 3DGS, along with their respective rendering equations—that limit their potential generalization and scalability.

In this work, we aim to *minimize 3D inductive biases* and push the boundaries of novel view synthesis with a fully data-driven approach. We propose the **Large View Synthesis Model (LVSM)**, a novel transformer-based framework that synthesizes novel-view images from posed sparse-view inputs

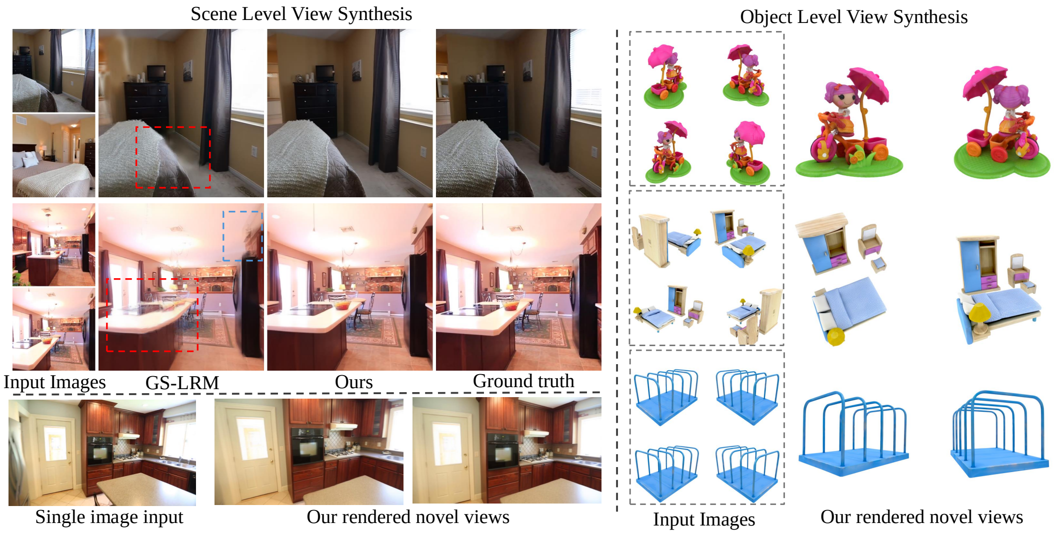

Figure 1: LVSM supports feed-forward novel view synthesis from sparse posed image inputs (even from a single view) on both objects and scenes. LVSM achieves significant quality improvements compared with the previous SOTA method, i.e., GS-LRM (Zhang et al., 2024). (Zoom in for more details.)

without predefined rendering equations or 3D structures, enabling accurate, training-efficient, and scalable novel view synthesis with photo-realistic quality (see Fig. 1 for visual examples).

To this end, we first introduce an **encoder-decoder LVSM**, removing handcrafted 3D representations and their rendering equations. We use an encoder transformer to map the input (patchified) multi-view image tokens into a fixed number of 1D latent tokens, independent of the number of input views. These latent tokens are then processed by a decoder transformer, which uses target-view Plücker rays as positional embeddings to generate the target view's image tokens, ultimately regressing the output pixel colors from a final linear layer. The encoder-decoder LVSM jointly learns a reconstructor (encoder), a scene representation (latent tokens), and a renderer (decoder) directly from data. By removing the need for predefined inductive biases in rendering and representation, LVSM offers improved generalization and achieves higher quality compared to NeRF- and GS-based approaches.

However, the encoder-decoder LVSM still retains a key bias: the need for an intermediate, albeit fully learned, scene representation. To further push the boundaries, we propose a **decoder-only LVSM**, which adopts a single-stream transformer to directly convert the input multi-view tokens into target view tokens, bypassing any intermediate representations. The decoder-only LVSM integrates the novel view synthesis process into a holistic data-driven framework, achieving scene reconstruction and rendering simultaneously in a fully implicit manner with minimal 3D inductive bias.

We present a comprehensive evaluation of variants of both LVSM architectures. Notably, our models, trained on 2-4 input views, demonstrate strong zero-shot generalization to an unseen number of views, ranging from a single input to more than 10. Thanks to minimal inductive biases, our decoder-only model consistently outperforms the encoder-decoder variant in terms of quality, scalability, and zero-shot capability with varying numbers of input views. On the other hand, the encoder-decoder model achieves much faster inference speed due to its use of a fixed-length latent scene representation. Both models, benefiting from reduced 3D inductive biases, outperform previous methods, achieving state-of-the-art novel view synthesis quality across multiple object-level and scene-level benchmark datasets. Specifically, **our decoder-only LVSM surpasses previous state-of-the-art methods, such as GS-LRM, by a substantial margin of 1.5 to 3.5 dB PSNR**. Our final models were trained on 64 A100 GPUs for 3-7 days, depending on the data type and model architecture, but we found that even with just 1–2 A100 GPUs for training, our model (with a decreased model and batch size) still outperforms all previous methods trained with equal or even more compute resources.

2 Related Work

View Synthesis. Novel view synthesis (NVS) has been studied for decades. Image-based rendering (IBR) methods perform view synthesis by weighted blending of input reference images using proxy geometry (Debevec et al., 1996; Heigl et al., 1999; Sinha et al., 2009). Light field methods build a slice of the 4D plenoptic function from dense view inputs (Gortler et al., 1996; Levoy & Hanrahan, 1996; Davis et al., 2012). Recent learning-based IBR methods incorporate convolutional networks to predict blending weights (Hedman et al., 2018; Zhou et al., 2016; 2018) or using predicted depth maps (Choi et al., 2019). However, the renderable region is usually constrained to be near the input viewpoints. Other work leverages multiview-stereo reconstructions to enable rendering under larger viewpoint changes (Jancosek & Pajdla, 2011; Chaurasia et al., 2013; Penner & Zhang, 2017). In contrast, we use more scalable network designs to learn generalizable priors from larger, real-world data. Moreover, we perform rendering at the image patch level, achieving better model efficiency, and rendering quality.

Optimizing 3D Representations. NeRF (Mildenhall et al., 2020) introduced a neural volumetric 3D representation with differentiable volume rendering, enabling neural scene reconstruction by minimizing rendering losses and setting a new standard in novel view synthesis. Later work improved NeRF with better rendering quality (Barron et al., 2021; Verbin et al., 2022; Barron et al., 2023), faster optimization or rendering speed (Reiser et al., 2021; Hedman et al., 2021; Reiser et al., 2023), and looser requirements on the input views (Niemeyer et al., 2022; Martin-Brualla et al., 2021; Wang et al., 2021b). Other work has explored hybrid representations that combine implicit NeRF content with explicit 3D information, e.g., in the form of voxels, as in DVGO (Sun et al., 2022). Spatial complexity can be further decreased by using sparse voxels (Liu et al., 2020; Fridovich-Keil et al., 2022), volume decomposition (Chan et al., 2022; Chen et al., 2022; 2023), and hashing techniques (Müller et al., 2022). Another line of works investigates explicit point-based representations (Xu et al., 2022; Zhang et al., 2022; Feng et al., 2022). Gaussian Splatting (Kerbl et al., 2023) extends these 3D points to 3D Gaussians, improving both rendering quality and speed. In contrast, we perform novel view synthesis using large transformer models (optionally with a learned latent scene representation) without the need for any inductive bias of using prior 3D representations or any per-scene optimization process.

Generalizable View Synthesis Methods. Generalizable methods enable fast NVS inference by using neural networks, trained across scenes, to predict the novel views or an underlying 3D representation in a feed-forward manner. For example, PixelNeRF (Yu et al., 2021), MVSNeRF (Chen et al., 2021) and IBRNet (Wang et al., 2021a) predict volumetric 3D representations from input views, utilizing 3D-specific priors like epipolar lines or plane sweep cost volumes. Later methods improve performance under (unposed) sparse views (Liu et al., 2022; Johari et al., 2022; Jiang et al., 2024; 2023), while other work extends to 3DGS representations Charatan et al. (2024); Chen et al. (2024); Tang et al. (2024). On the other hand, approaches that attempt to directly learn a geometry-free rendering function (Suhail et al., 2022a; Sajjadi et al., 2022; Sitzmann et al., 2021; Rombach et al., 2021) have proven not to be scalable and lack model capacity, preventing them from capturing high-frequency details. Specifically, SRT (Sajjadi et al., 2022) intends to remove the use of handcrafted 3D representations and learn a latent representation with a transformer instead, similar to our encoder-decoder model. However, it utilizes CNN to attain input tokens and adopts a cross-attention decoder transformer with non-scalable model and rendering designs. In contrast, our models are fully transformer-based with self-attention, and we introduce a more scalable decoder-only architecture that can effectively learn the novel view synthesis function with minimal 3D inductive bias, without an intermediate latent representation.

Recently, 3D large reconstruction models (LRMs) have emerged (Hong et al., 2024; Li et al., 2023; Wang et al., 2023; Xu et al., 2023; Wei et al., 2024; Zhang et al., 2024; Xie et al., 2024), utilizing scalable transformer architectures (Vaswani et al., 2017) trained on large datasets to learn generic 3D priors. While these methods avoid using epipolar projection or cost volumes in their architectures, they still rely on existing 3D representations like tri-plane NeRFs, meshes, or 3DGS, along with their corresponding rendering equations, limiting their potential. In contrast, our approach eliminates these 3D inductive biases, aiming to learn a rendering function (and optionally a scene representation) directly from data. This leads to more scalable models and significantly improved rendering quality.

In addition to the deterministic methods mentioned above, there are also some diffusion-based generative models(Watson et al., 2022; Liu et al., 2023a; Gao* et al., 2024; Zheng & Vedaldi, 2024; Kong et al., 2024; Voleti et al., 2025) supporting novel view synthesis. We discuss how our models differ from those methods in the Appendix. A.6.

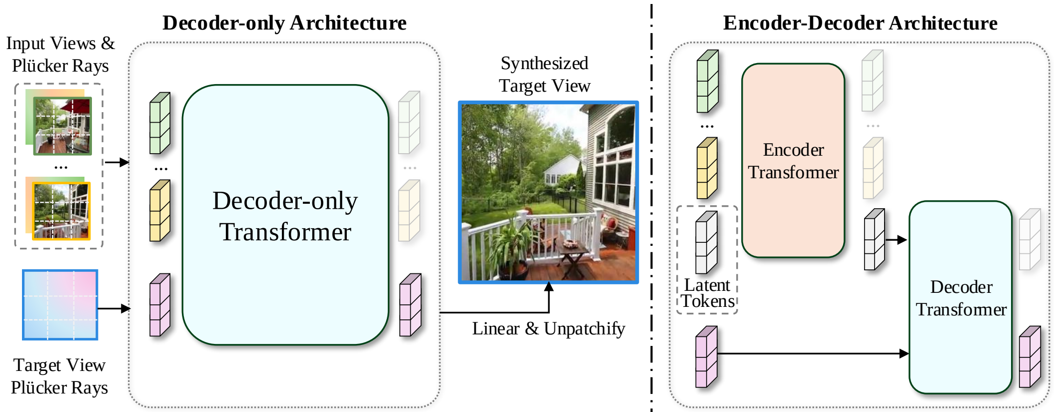

Figure 2: **LVSM model architecture.** LVSM first patchifies the posed input images into tokens. The target view to be synthesized is represented by its Plücker ray embeddings and is also tokenized. The input view and target tokens are sent to a full transformer-based model to predict the tokens that are used to regress the target view pixels. We study two LVSM transformer architectures, as a *Decoder-only* architecture (left) and a *Encoder-Decoder* architecture (right).

3 Method

We first provide an overview of our method in Sec. 3.1, then describe two different transformer-based model variants in Sec. 3.2.

3.1 Overview

Given N sparse input images with known camera poses and intrinsics, denoted as $\{(\mathbf{I}_i, \mathbf{E}_i, \mathbf{K}_i)|i = 1, \ldots, N\}$, LVSM synthesizes target image $\mathbf{I}^t$ with novel target camera extrinsics $\mathbf{E}^t$ and intrinsics $\mathbf{K}^t$. Each input image has shape $\mathbb{R}^{H\times W\times 3}$, where H and W are the image height and width (and there are 3 color channels).

Framework. As shown in Fig. 2, our LVSM method uses an end-to-end transformer model to directly render the target image. LVSM starts by tokenizing the input images. We first compute a pixel-wise Plücker ray embedding (Plucker, 1865) for each input view using the camera poses and intrinsics. We denote these Plücker ray embeddings as $\{\mathbf{P}_i \in \mathbb{R}^{H\times W\times 6}|i = 1, \ldots, N\}$. We patchify the RGB images and Plücker ray embeddings into non-overlapping patches, following the image tokenization layer of ViT (Dosovitskiy et al., 2020). We denote the image and Plücker ray patches of input image $\mathbf{I}_i$ as $\{\mathbf{I}_{ij} \in \mathbb{R}^{p\times p\times 3}|j = 1, \ldots, HW/p^2\}$ and $\{\mathbf{P}_{ij} \in \mathbb{R}^{p\times p\times 6}|j = 1, \ldots, HW/p^2\}$, respectively, where p is the patch size. For each patch, we concatenate its image patch and Plücker ray embedding patch, reshape them into a 1D vector, and use a linear layer to map it into an input patch token $\mathbf{x}_{ij}$:

$$\mathbf{x}_{ij} = \text{Linear}_{input}([\mathbf{I}_{ij}, \mathbf{P}_{ij}]) \in \mathbb{R}^d, \tag{1}$$

where d is the latent size, and $[\cdot, \cdot]$ means concatenation.

Similarly, LVSM represents the target pose to be synthesized as its Plücker ray embeddings $\mathbf{P}^t \in \mathbb{R}^{H\times W\times 6}$, computed from the given target extrinsics $\mathbf{E}^t$ and intrinsics $\mathbf{K}^t$. We use the same patchify method and another linear layer to map it to the Plücker ray tokens of the target view, denoted as:

$$\mathbf{q}_j = \text{Linear}_{target}(\mathbf{P}_j^t) \in \mathbb{R}^d, \tag{2}$$

where $\mathbf{P}_j^t$ is the Plücker ray embedding of the j^{th} patch in the target view.

We flatten the input tokens into a 1D token sequence, denoted as $x_1, \ldots, x_{l_x}$, where $l_x = NHW/p^2$ is the sequence length of the input image tokens. We also flatten the target query tokens as $q_1, \ldots, q_{l_q}$ from the ray embeddings, with $l_q = HW/p^2$ as the sequence length.

LVSM then synthesizes novel view by conditioning on the input view tokens using a full transformer model M:

$$y_1, \ldots, y_{l_q} = M(q_1, \ldots, q_{l_q}|x_1, \ldots, x_{l_x}). \tag{3}$$

Specifically, the output token y_j is an updated version of q_j, containing the information to predict the pixel values of the j^{th} patch of the target view. More details of model M are described in Sec. 3.2.

We recover the spatial structure of output tokens using the inverse operation of the flatten operation. To regress RGB values of the target patch, we employ a linear layer followed by a Sigmoid function:

$$\hat{\mathbf{I}}_j^t = \text{Sigmoid}(\text{Linear}_{out}(y_j)) \in \mathbb{R}^{3p^2}. \tag{4}$$

We reshape the predicted RGB values back to the 2D patch in $\mathbb{R}^{p \times p \times 3}$, and then we get the synthesized novel view $\hat{\mathbf{I}}^t$ by performing the same operation on all target patches independently.

Loss Function. Following prior works (Zhang et al., 2024; Hong et al., 2024), we train LVSM with photometric novel view rendering losses:

$$\mathcal{L} = \text{MSE}(\hat{\mathbf{I}}^t, \mathbf{I}^t) + \lambda \cdot \text{Perceptual}(\hat{\mathbf{I}}^t, \mathbf{I}^t), \tag{5}$$

where λ is the weight for balancing the perceptual loss (Johnson et al., 2016).

3.2 Transformer-Based Model Architecture

In this subsection, we present the two LVSM architectures—*encoder-decoder* and *decoder-only*—both designed to minimize 3D inductive biases. Following their name, '*encoder-decoder*' first converts input images to an intermediate latent representation before decoding the final image pixels, whereas '*decoder-only*' directly outputs the synthesized target view without an intermediate representation, further minimizing inductive biases in its design. Unlike most standard language model transformers (Vaswani et al., 2017; Jaegle et al., 2021; Radford et al., 2019), which typically use full attention for encoders and causal/cross attention for decoders, we adopt dense full attention across all our encoder and decoder architectures. The naming of our models is based on the output characteristics, instead of being strictly tied to the transformer architecture they utilize. Please refer to Appendix A.1 for a further detailed discussion of the naming.

Encoder-Decoder Architecture. The encoder-decoder LVSM comes with a learned latent scene representation for view synthesis, avoiding the use of NeRF, 3DGS, and other representations. The encoder first maps the input tokens to an intermediate 1D array of latent tokens (serving as a latent scene representation). Then the decoder predicts the outputs, conditioning on the latent tokens and target pose.

Similar to the triplane tokens in LRMs (Hong et al., 2024; Wei et al., 2024), we use l learnable latent tokens $\{e_k \in \mathbb{R}^d | k = 1, ..., l\}$ to aggregate information from input tokens $\{x_i\}$. The encoder, denoted as Transformer_{Enc}, uses multiple transformer layers with self-attention. We concatenate $\{x_i\}$ and $\{e_k\}$ as the input of Transformer_{Enc}, which performs information aggregation between them to update $\{e_k\}$. The output tokens that correspond to the latent tokens, denoted as $\{z_k\}$, are used as the intermediate latent scene representation. The other tokens (updated from $\{x_i\}$, denoted as $\{x_i'\}$) are unused and discarded.

The decoder uses multiple transformer layers with self-attention. In detail, the inputs are the concatenation of the latent tokens $\{z_k\}$ and the target view query tokens $\{q_j\}$. By applying self-attention transformer layers over the input tokens, we get output tokens with the same sequence length as the input. The output tokens that corresponds to the target tokens $q_1, \ldots, q_{lq}$ are treated as final outputs $y_1, \ldots, y_{lq}$, and the other tokens (updated from $\{z_i\}$, denoted as $\{z_i'\}$) are unused. This architecture can be formulated as:

$$x_1', \ldots, x_{l_x}', z_1, \ldots, z_l = \text{Transformer}_{Enc}(x_1, \ldots, x_{l_x}, e_1, \ldots, e_l) \tag{6}$$

$$z_1', \ldots, z_l', y_1, \ldots, y_{l_q} = \text{Transformer}_{Dec}(z_1, \ldots, z_l, q_1, \ldots, q_{l_q}). \tag{7}$$

Decoder-Only Architecture. Our alternate, decoder-only model further eliminates the need for an intermediate scene representation. Its architecture is similar to the decoder in encoder-decoder architecture but differs in inputs and model size. We concatenate the two sequences of input tokens $\{x_i\}$ and target tokens $\{q_j\}$. The final output $\{y_j\}$ is the decoder's corresponding output for the target tokens $\{q_j\}$. The other tokens (updated from $\{x_i\}$, denoted as $\{x_i'\}$) are unused and discarded. This architecture can be formulated as:

$$x_1', \ldots, x_{l_x}', y_1, \ldots, y_{l_q} = \text{Transformer}_{Dec\text{-}only}(x_1, \ldots, x_{l_x}, q_1, \ldots, q_{l_q}) \tag{8}$$

Here the $\text{Transformer}_{Dec\text{-}only}$ has multiple full self-attention transformer layers.

Table 1: **Quantitative comparisons on object-level (left) and scene-level (right) view synthesis.** For the object-level comparison, we matched the baseline settings with GS-LRM (Zhang et al., 2024) in both input and rendering under both resolution of 256 (Res-256) and 512 (Res-512). For the scene-level comparison, we use the same validation dataset used by pixelSplat (Charatan et al., 2024), which has 256 resolution.

	ABO (Collins et al., 2022a)			GSO (Downs et al., 2022)		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Triplane-LRM (Li et al., 2023) (Res-512)	27.50	0.896	0.093	26.54	0.893	0.064
GS-LRM (Zhang et al., 2024) (Res-512)	29.09	0.925	0.085	30.52	0.952	0.050
Ours Encoder-Decoder (Res-512)	29.81	0.913	0.065	29.32	0.933	0.052
Ours Decoder-Only (Res-512)	32.10	0.938	0.045	32.36	0.962	0.028
LGM (Tang et al., 2024) (Res-256)	20.79	0.813	0.158	21.44	0.832	0.122
GS-LRM (Zhang et al., 2024) (Res-256)	28.98	0.926	0.074	29.59	0.944	0.051
Ours Encoder-Decoder (Res-256)	30.35	0.923	0.052	29.19	0.932	0.046
Ours Decoder-Only (Res-256)	32.47	0.944	0.037	31.71	0.957	0.027

	RealEstate10k (Zhou et al., 2018)		
	PSNR ↑	SSIM ↑	LPIPS ↓
pixelNeRF (Yu et al., 2021)	20.43	0.589	0.550
GPNR (Suhail et al., 2022a)	24.11	0.793	0.255
Du et. al (Du et al., 2023)	24.78	0.820	0.213
pixelSplat (Charatan et al., 2024)	26.09	0.863	0.136
MVSplat (Chen et al., 2024)	26.39	0.869	0.128
GS-LRM (Zhang et al., 2024)	28.10	0.892	0.114
Ours Encoder-Decoder	28.58	0.893	0.114
Ours Decoder-Only	29.67	0.906	0.098

4 EXPERIMENTS

We introduce the details of used datasets and the baseline methods. Then we present results of LVSM for both object-level and scene-level novel view synthesis.

4.1 DATASETS

We train (and evaluate) LVSM on object-level and scene-level datasets separately.

Object-level Datasets. We use the Objaverse dataset (Deitke et al., 2023) to train LVSM. We follow the rendering settings in GS-LRM (Zhang et al., 2024) and render 32 random views for 730K objects. We test on two object-level datasets, i.e., Google Scanned Objects (Downs et al., 2022) (GSO) and Amazon Berkeley Objects (Collins et al., 2022b) (ABO). In detail, GSO and ABO contain 1099 and 1000 objects, respectively. Following Instant3D (Li et al., 2023) and GS-LRM (Zhang et al., 2024), we have 4 sparse views as testing inputs and another 10 views as target images.

Scene-level Datasets. We use the RealEstate10K dataset (Zhou et al., 2018), which contains 80K video clips curated from 10K Youtube videos of both indoor and outdoor scenes. We follow the train/test data split used in pixelSplat (Charatan et al., 2024).

4.2 TRAINING DETAILS

Improving Training Stability. We observe that the training of LVSM crashes with plain transformer layers (Vaswani et al., 2017) due to exploding gradients. We empirically find that using QK-Norm (Henry et al., 2020) in the transformer layers stabilizes training. This observation is consistent with Bruce et al. (2024) and Esser et al. (2024). We also skip optimization steps with gradient norm > 5.0 besides the standard 1.0 gradient clipping.

Efficient Training Techniques. We use FlashAttention-v2 (Dao, 2023) in the xFormers (Lefaudeux et al., 2022), gradient checkpointing (Chen et al., 2016), and mixed-precision training with Bfloat16 data type to accelerate training.

Other Details. For more details about the model and training, please refer to Appendix A.2, and a detailed model architecture diagram (Fig. 8).

4.3 EVALUATION AGAINST BASELINES

In this section, we describe our experimental setup and datasets (Sec. 4.1), introduce our model training details (Sec. 4.2), report evaluation results (Sec. 4.3) and perform an ablation study (Sec. 4.4).

Object-Level Results. We compare with Instant3D's Triplane-LRM (Li et al., 2023) and GS-LRM (Zhang et al., 2024) at a resolution of 512. As shown on the left side of Table 1, our LVSM achieves the best performance against all prior works. In particular, at 512 resolution, our *decoder-only* LVSM achieves a 3 dB and 2.8 dB PSNR gain against the best prior method GS-LRM on ABO and GSO, respectively; our *encoder-decoder* LVSM achieves performance comparable to GS-LRM.

We also compare with LGM (Tang et al., 2024) at the resolution of 256, as the official LGM model is trained with an input resolution of 256. We also report the performance of models trained on resolution of 256. Compared with the best prior work GS-LRM, our *decoder-only* LVSM demonstrates a 3.5 dB and 2.2 dB PSNR gain on ABO and GSO, respectively; our *encoder-decoder* LVSM shows a slightly better performance than GS-LRM.

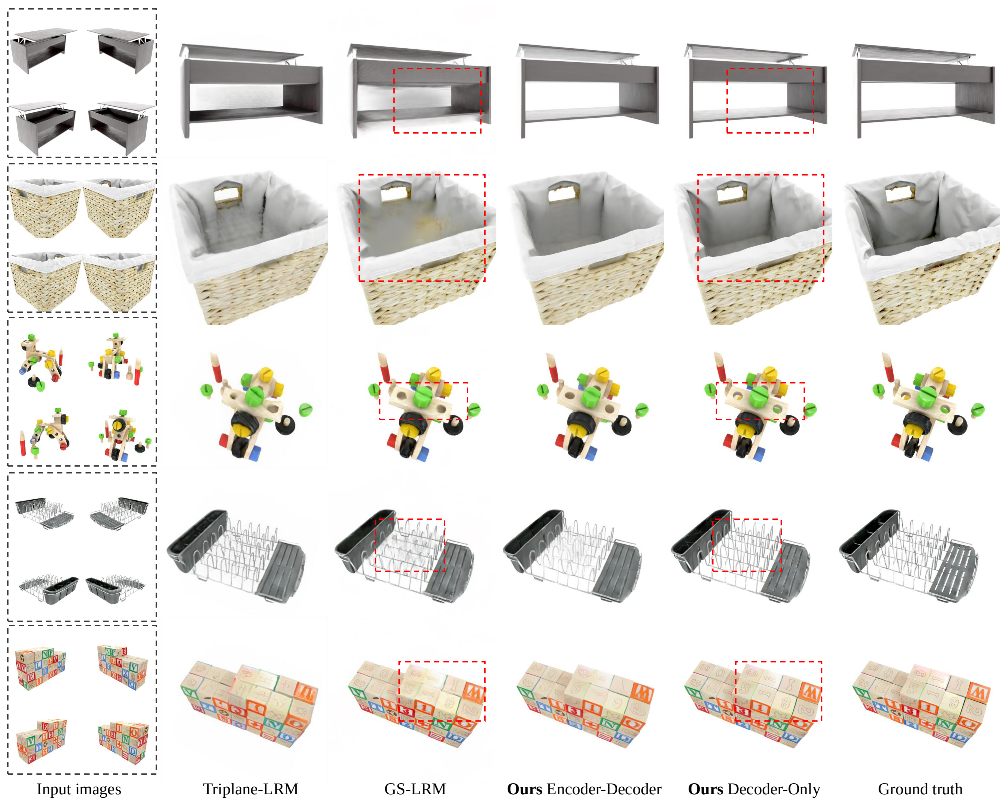

Figure 3: **Object-level visual comparison at 512 resolution.** Given 4 sparse input posed images (leftmost column), we compare our high-res object-level novel-view rendering results with two baselines: Instant3D's *Triplane-LRM* (Li et al., 2023) and *GS-LRM* (Res-512) (Zhang et al., 2024) . Both our Encoder-Decoder and Decoder-Only models exhibit fewer floaters (first example) and fewer blurry artifacts (second example), compared to the baselines. Our Decoder-Only model effectively handles complex geometry, including small holes (third example) and thin structures (fourth example). Additionally, it preserves the details of high-frequency texture (last example).

These significant performance gains validate the effectiveness of our design target of removing 3D inductive bias. More interestingly, the larger performance gain on ABO shows that our method can handle challenging materials, which are difficult for current handcrafted 3D representations. The qualitative results in Fig. 3 and Fig. 7 also validate the high degree of realism of LVSM, especially for examples with specular materials, detailed textures, and thin, complex geometry.

Scene-Level Results. We compare with prior works pixelNeRF (Yu et al., 2021), GPNR (Suhail et al., 2022a), Du et al. (2023), pixelSplat (Charatan et al., 2024), MVSplat (Chen et al., 2024) and GS-LRM (Zhang et al., 2024). As shown on the right side of Table 1, our *decoder-only* LVSM shows a 1.6 dB PSNR gain compared with the best prior work GS-LRM. Our *encoder-decoder* LVSM also demonstrates comparable results to GS-LRM. These improvements can be observed qualitatively in Fig. 4, where LVSM has fewer floaters and better performance on thin structures and specular materials, consistent with the object-level results. These outcomes again validate the efficacy of our design of using minimal 3D inductive bias.

LVSM Trained with Only 1 GPU. Limited computing is a key bottleneck for academic research. To show the potential of LVSM using academic-level resources, we train LVSM on the scene-level dataset (Zhou et al., 2018) following the setting of pixelSplat (Charatan et al., 2024) and MVSplat (Chen et al., 2024), with only a single A100 80G GPU for 7 days. In this experiment, we use a smaller decoder-only model (denoted LVSM-small) with 6 transformer layers and a smaller batch size of 64 (in contrast to the default one with 24 layers and batch size 512). Our decoder-only LVSM-small shows a performance of 27.66 dB PSNR, 0.870 SSIM, and 0.129 LPIPS. This performance surpasses the prior best 1-GPU-trained model MVSplat, with a 1.3 dB PSNR gain. We also train the decoder-only LVSM (12 transformer layers, batch size 64) with 2 GPUs for 7 days, exhibiting a performance of 28.56 dB PSNR, 0.889 SSIM, and 0.112 LPIPS. This performance is

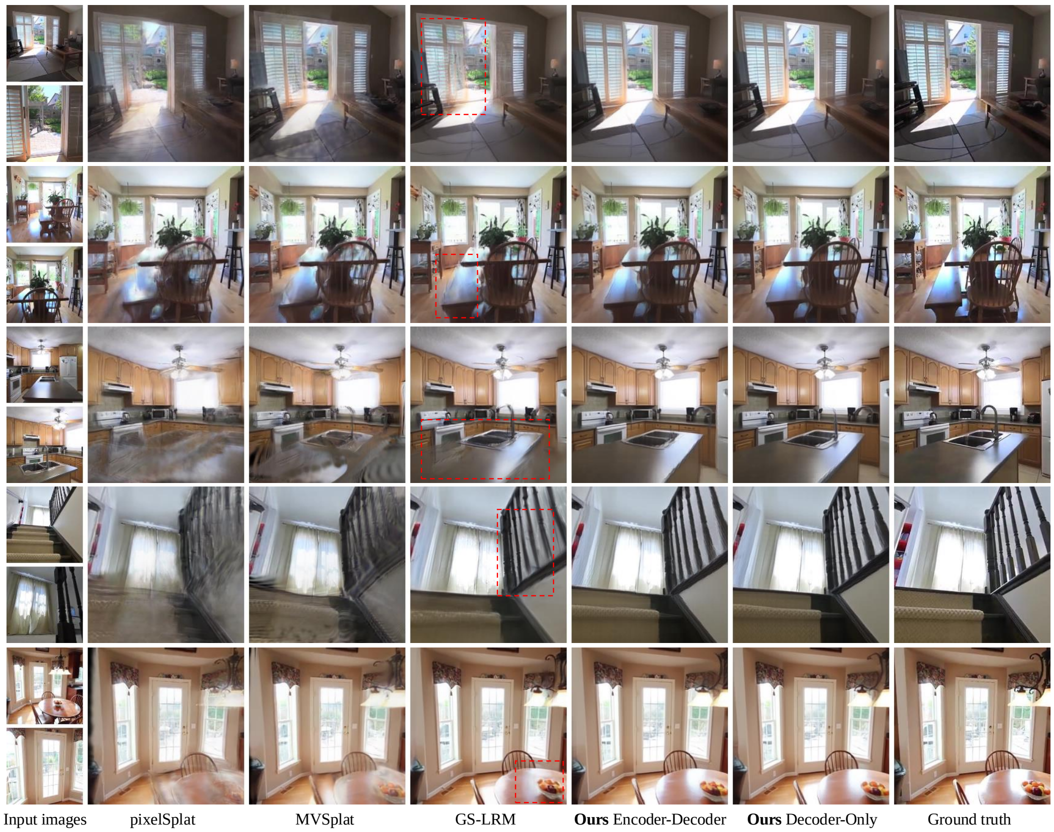

Figure 4: **Scene-level visual comparison.** We evaluate encoder-decoder and decoder-only LVSM on scene-level view synthesis, comparing them against the prior leading baseline methods, namely pixelSplat (Charatan et al., 2024), MVSplat (Chen et al., 2024), and GS-LRM (Zhang et al., 2024). Our methods exhibit less texture and geometric artifacts, generate more realistic specular reflections, and are closer to the ground truth images.

even better than GS-LRM with 24 transformer layers trained on 64 GPUs. These results show the promising potential of LVSM for academic research.

4.4 Ablation Studies

Model Size. In Tab. 2, we ablate the model size designs of both LVSM variants on both object and scene level. To save resources, the experiments are run with 8 GPUs and a total batch size of 64.

For the *encoder-decoder* LVSM, we maintain the total number of transformer layers while allocating a different number of layers to the encoder and decoder. We observe that using more decoder layers helps the performance while using more encoder layers harms the performance. We hypothesize that this is because the encoder uses the latent representation as the compression of input image information, and a deeper encoder makes this compression process harder to learn, resulting in greater compression errors. This observation suggests that using the inductive bias of the encoder and intermediate latent representation may not be optimal for the final quality, which also aligns with our observation that the decoder-only variant outperforms the encoder-decoder variant.

For the *decoder-only* LVSM, we experiment with using different numbers of transformer layers and model sizes in the decoder. The experiment verifies that *decoder-only* LVSM demonstrates an increasing performance when using more transformer layers. This phenomenon validates the scalability of the *decoder-only* LVSM.

Model Architecture. As shown in Tab. 3, we evaluate the effectiveness of our model designs. We also visualize the equivalent attention mask for each design in Fig. 9 for better illustration. To save resources, the encoder-decoder experiments here are run with 32 GPUs, a total batch size of 256, and a decreased number of training target views of 8. The decoder-only experiment is run with our original setup.

Table 2: **Ablations studies on model sizes.** The following experiments are all run with 8 GPUs.

		GSO			RealEstate10k		
	# Params	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Ours Encoder-Decoder (6 + 18)	173M	26.48	0.901	0.065	28.32	0.888	0.117
Ours Encoder-Decoder (12 + 12)	173M	25.69	0.889	0.076	27.39	0.869	0.137
Ours Encoder-Decoder (18 + 6)	173M	24.74	0.873	0.091	26.80	0.855	0.152
Ours Decoder-Only (24 layers)	171M	27.04	0.910	0.055	28.89	0.894	0.108
Ours Decoder-Only (18 layers)	128M	26.81	0.907	0.057	28.77	0.892	0.109
Ours Decoder-Only (12 layers)	86M	26.11	0.896	0.065	28.61	0.890	0.111
Ours Decoder-Only (6 layers)	43M	24.15	0.865	0.092	27.62	0.869	0.129

Table 3: **Ablations studies on model architecture.**

	GSO (Downs et al., 2022)		
	PSNR ↑	SSIM ↑	LPIPS ↓
Ours Encoder-Decoder	**28.07**	**0.920**	**0.053**
Ours w/ CNN tokenizer	27.59	0.914	0.052
Ours w/o latents' updating	26.61	0.903	0.061
Ours w/ per-patch prediction	26.27	0.897	0.072
Ours w/ pure cross-att decoder	24.60	0.876	0.085
	RealEstate10k (Zhou et al., 2018)		
	PSNR ↑	SSIM ↑	LPIPS ↓
Ours Decoder-Only	**29.67**	**0.906**	**0.098**
Ours w/ per-patch prediction	28.98	0.897	0.103

Our *encoder-decoder* LVSM leverages an encoder to transform input images into a set of 1D tokens that serve as an intermediate latent representation of the 3D scene. A decoder can then render novel view images from this latent representation. Our *encoder-decoder* LVSM is similar to SRT (Sajjadi et al., 2022) at a high level. However, *encoder-decoder* LVSM introduces a highly different architecture that significantly improves performance.

First, to generate input tokens for the encoder, our approach is inspired by ViT (Dosovitskiy et al., 2020) and recent LRMs (Wei et al., 2024; Zhang et al., 2024). Specifically, we tokenize the input images and their pose information by simply splitting the concatenated input views and Plücker ray embeddings into non-overlapping patches. In contrast, SRT relies on shallow convolutional neural networks (CNNs) to extract patch features, which are then flattened into tokens. We tried SRT's CNN-based tokenizing method and observed it makes training more unstable with a larger grad norm, which leads to worse performance. As demonstrated in Tab. 3, replacing our simple tokenizer with SRT's CNN-based tokenizer degrades performance (ours w/ CNN tokenizer).

Second, our encoder employs self-attention to progressively compress the information from posed input images into a fixed-length set of 1D latent tokens. This design ensures a consistent rendering speed, regardless of the number of input images, as shown in Fig. 6. In contrast, SRT's latent token size grows linearly with the number of input views, resulting in decreased rendering efficiency.

Third, the decoder of our encoder-decoder model utilizes pure (bidirectional) self-attention, allowing latent tokens and output target image tokens to attend to each other. This enables i) latent tokens to be updated by fusing information from themselves and other tokens, which also means the parameters of the decoder are a part of the scene representation; ii) output patch pixels can also attend to other patches for joint updates, ensuring the global awareness of the rendered target image. We ablate our full attention design choice by experimenting with different attention mechanisms, illustrated in Fig. 9. As shown in Tab.3, disabling either the latents' updating (ours w/o latents' updating) or the joint updating of direct output pixel patches (ours w/ per-patch prediction) significantly degrades performance. Prior work SRT (Sajjadi et al., 2022) does not support these functions as it misses both mechanisms by employing a decoder with pure cross-attention. We experiment with an LVSM variant by adopting SRT's decoder designs. As shown in Tab. 3 (ours w/ pure cross-att decoder), this modification leads to worse performance.

For the *decoder-only* LVSM, it further pushes the boundaries of eliminating the inductive bias and bypasses any intermediate representations. It adopts a single-stream transformer to directly convert the input multi-view tokens into target view tokens, treating the view synthesis like a sequence-to-sequence translation task, which is fundamentally different from any previous work. We then ablate the importance of joint prediction of target image patches in the *decoder-only* LVSM. We design a variant where the colors of each target pose patch are predicted independently, without applying self-attention across other target pose patches. We achieve this by letting each transformer's layer's key and value matrices only consist of the updated input image tokens, while both the updated input image tokens and target pose tokens form the query matrices. As shown on the bottom part of Tab. 3, this variant shows a worse performance with a 0.7 dB PSNR degradation. This result demonstrates the importance of using both input and target tokens as the context of each other for information propagation using the simplest full self-attention transformer, which is in line with our philosophy of reducing inductive bias.

4.5 Discussions

Zero-shot Generalization to More Input Views. We compare our LVSM with GS-LRM by taking different numbers of input views to the training. We report the results on the object level. Note that

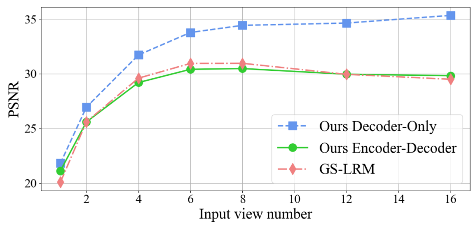

Figure 5: **Zero-shot generalization to different number of input images** on the GSO dataset (Downs et al., 2022). We note that all models are trained with just 4 input views.

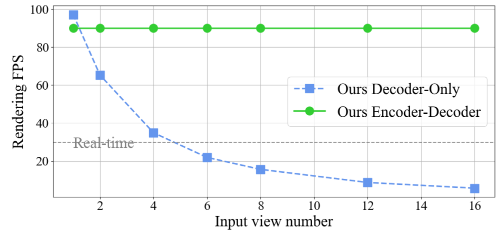

Figure 6: **Rendering FPS with different number of input images.** We test the FPS on the object level under 256×256 resolution. We refer to rendering as the decoding process, which synthesizes novel views from latent tokens or input images.

these models are trained only with 4 input views and test on other input view numbers in a zero-shot manner. As shown in Fig. 5, our *decoder-only* LVSM shows increasingly better performance when using more input views, verifying the scalability of our model design at test time. Our *encoder-decoder* LVSM shows a similar performance pattern with GS-LRM, i.e., exhibiting a performance drop when using more than 8 input views. We conjecture the reason is the inductive bias of the encoder-decoder design, i.e. using intermediate representation as a compression of input information, limits the performance. In addition, our single-input result (input view number = 1) is competitive and even beats some of our baseline which takes 4 images as input. These performance patterns validate our design target of using minimal 3D inductive bias for learning a fully data-driven rendering model and cohere with our discussion in Sec. 3.2.

Encoder-Decoder versus Decoder-Only. As we mentioned in Sec. 3, the *decoder-only* and *encoder-decoder* architectures exhibit different trade-offs in speed, quality, and potential.

The *encoder-decoder* model transforms 2D image inputs into a fixed-length set of 1D latent tokens, which serve as a compressed representation of the 3D scene. This approach simplifies the decoder, reducing its model size. Furthermore, during the rendering/decoding process, the decoder always receives a fixed number of tokens, regardless of the number of input images, ensuring a consistent rendering speed. As a result, this model offers improved rendering efficiency, as shown in Fig. 6. Additionally, the use of 1D latent tokens as the latent representation for the 3D scene opens up the possibility of integrating this model with generative approaches for 3D content generation on its 1D latent space. Nonetheless, the compression process can result in information loss, as the fixed latent tokens length is usually smaller than the original image tokens length, which imposes an upper bound on performance. This characteristic of the *encoder-decoder* LVSM mirrors prior encoder-decoder LRMs, whereas our LVSM does not have an explicit 3D structure.

In contrast, the *decoder-only* model learns a direct mapping from the input image to the target novel view, showcasing better scalability. For example, as the number of input images increases, the model can leverage all available information, resulting in improved novel view synthesis quality. However, this property also leads to a linear increase in input image tokens, causing the computational cost to grow quadratically and limiting the rendering speed.

Single Input Image. As shown in our project page, Fig. 1 and Fig. 5, we observe that our LVSM also works with a single input view for many cases, even though the model is trained with multi-view images during training. This observation shows the capability of LVSM to understand the 3D world, e.g. understanding depth, rather than performing purely pixel-level view interpolation.

5 Conclusions

In this work, we presented the Large View Synthesis Model (LVSM), a transformer-based approach designed to minimize 3D inductive biases for scalable and generalizable novel view synthesis. Our two architectures—encoder-decoder and decoder-only—bypass physical-rendering-based 3D representations like NeRF and 3D Gaussian Splatting, allowing the model to learn priors directly from data, leading to more flexible and scalable novel view synthesis. The decoder-only LVSM, with its minimal inductive biases, excels in scalability, zero-shot generalization, and rendering quality, while the encoder-decoder LVSM achieves faster inference due to its fully learned latent scene representation. Both models demonstrate superior performance across diverse benchmarks and mark an important step towards general and scalable novel view synthesis in complex, real-world scenarios.

REFERENCES

Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *ICCV*, 2021.

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

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16123–16133, 2022.

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*, pp. 19457–19467, 2024.

Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM transactions on graphics (TOG)*, 32 (3):1–12, 2013.

Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo, 2021. URL `https://arxiv.org/abs/2103.15595`.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, pp. 333–350. Springer, 2022.

Anpei Chen, Zexiang Xu, Xinyue Wei, Siyu Tang, Hao Su, and Andreas Geiger. Factor fields: A unified framework for neural fields and beyond. *arXiv preprint arXiv:2302.01226*, 2023.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

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, 2024. URL `https://arxiv.org/abs/2403.14627`.

Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7781–7790, 2019.

Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F. Yago Vicente, Thomas Dideriksen, Himanshu Arora, Matthieu Guillaumin, and Jitendra Malik. Abo: Dataset and benchmarks for real-world 3d object understanding, 2022a. URL `https://arxiv.org/abs/2110.06199`.

Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21126–21136, 2022b.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

Abe Davis, Marc Levoy, and Fredo Durand. Unstructured light fields. In *Computer Graphics Forum*, pp. 305–314, 2012.

Paul E. Debevec, Camillo Jose Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, 1996. URL https://api.semanticscholar.org/CorpusID:2609415.

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*, pp. 13142–13153, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2020. URL https://api.semanticscholar.org/CorpusID:225039882.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items, 2022. URL https://arxiv.org/abs/2204.11918.

Yilun Du, Cameron Smith, Ayush Tewari, and Vincent Sitzmann. Learning to render novel views from wide-baseline stereo pairs, 2023. URL https://arxiv.org/abs/2304.08463.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

Wanquan Feng, Jin Li, Hongrui Cai, Xiaonan Luo, and Juyong Zhang. Neural points: Point cloud representation with neural fields for arbitrary upsampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18633–18642, 2022.

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*, pp. 5501–5510, 2022.

Ruiqi Gao*, Aleksander Holynski*, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul P. Srinivasan, Jonathan T. Barron, and Ben Poole*. Cat3d: Create anything in 3d with multi-view diffusion models. *Advances in Neural Information Processing Systems*, 2024.

Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996. URL https://api.semanticscholar.org/CorpusID:2036193.

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (ToG)*, 37(6):1–15, 2018.

Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5875–5884, 2021.

Benno Heigl, Reinhard Koch, Marc Pollefeys, Joachim Denzler, and Luc Van Gool. Plenoptic modeling and rendering from image sequences taken by a hand-held camera. In *Mustererkennung 1999: 21. DAGM-Symposium Bonn, 15.–17. September 1999*, pp. 94–101. Springer, 1999.

Alex Henry, Prudhvi Raj Dachapally, Shubham Pawar, and Yuxuan Chen. Query-key normalization for transformers. *arXiv preprint arXiv:2010.04245*, 2020.

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, 2024. URL https://arxiv.org/abs/2311.04400.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021.

Michal Jancosek and Tomas Pajdla. Multi-view reconstruction preserving weakly-supported surfaces. In *CVPR 2011*, pp. 3121–3128. IEEE, 2011.

Hanwen Jiang, Zhenyu Jiang, Yue Zhao, and Qixing Huang. Leap: Liberate sparse-view 3d modeling from camera poses. *arXiv preprint arXiv:2310.01410*, 2023.

Hanwen Jiang, Zhenyu Jiang, Kristen Grauman, and Yuke Zhu. Few-view object reconstruction with unknown categories and camera poses. In *2024 International Conference on 3D Vision (3DV)*, pp. 31–41. IEEE, 2024.

Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors, 2022. URL https://arxiv.org/abs/2111.13539.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pp. 694–711. Springer, 2016.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Xin Kong, Shikun Liu, Xiaoyang Lyu, Marwan Taher, Xiaojuan Qi, and Andrew J Davison. Eschernet: A generative model for scalable view synthesis. *arXiv preprint arXiv:2402.03908*, 2024.

Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, et al. xformers: A modular and hackable transformer modelling library, 2022.

Marc Levoy and Pat Hanrahan. Light field rendering. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996. URL https://api.semanticscholar.org/CorpusID:1363510.

Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model, 2023. URL https://arxiv.org/abs/2311.06214.

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.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object, 2023a.

Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7824–7833, 2022.

Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023b.

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint arXiv:2310.15008*, 2023.

Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7210–7219, 2021.

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, 2020. URL https://arxiv.org/abs/2003.08934.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):1–15, July 2022. ISSN 1557-7368. doi: 10.1145/3528223.3530127. URL http://dx.doi.org/10.1145/3528223.3530127.

Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5480–5490, 2022.

Eric Penner and Li Zhang. Soft 3d reconstruction for view synthesis. *ACM Transactions on Graphics (TOG)*, 36(6):1–11, 2017.

Julius Plucker. Xvii. on a new geometry of space. *Philosophical Transactions of the Royal Society of London*, pp. 725–791, 1865.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 14335–14345, 2021.

Christian Reiser, Rick Szeliski, Dor Verbin, Pratul Srinivasan, Ben Mildenhall, Andreas Geiger, Jon Barron, and Peter Hedman. Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes. *ACM Transactions on Graphics (TOG)*, 42(4):1–12, 2023.

Robin Rombach, Patrick Esser, and Björn Ommer. Geometry-free view synthesis: Transformers and no 3d priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14356–14366, 2021.

Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pp. 1–10, 2022a.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4713–4726, 2022b.

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*, pp. 6229–6238, 2022.

Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model, 2023a.

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

Sudipta Sinha, Drew Steedly, and Rick Szeliski. Piecewise planar stereo for image-based rendering. In *2009 International Conference on Computer Vision*, pp. 1881–1888, 2009.

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019.

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.

Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *European Conference on Computer Vision*, pp. 156–174. Springer, 2022a.

Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Light field neural rendering, 2022b. URL https://arxiv.org/abs/2112.09687.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5459–5469, 2022.

Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. Ref-NeRF: Structured view-dependent appearance for neural radiance fields. *CVPR*, 2022.

Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In *European Conference on Computer Vision*, pp. 439–457. Springer, 2025.

Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. *arXiv preprint arXiv:2311.12024*, 2023.

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering, 2021a. URL https://arxiv.org/abs/2102.13090.

Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021b.

Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *arXiv preprint arXiv:2210.04628*, 2022.

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

Desai Xie, Sai Bi, Zhixin Shu, Kai Zhang, Zexiang Xu, Yi Zhou, Sören Pirk, Arie Kaufman, Xin Sun, and Hao Tan. Lrm-zero: Training large reconstruction models with synthesized data. *arXiv preprint arXiv:2406.09371*, 2024.

Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5438–5448, 2022.

Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, and Kai Zhang. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model, 2023.

Jiayu Yang, Ziang Cheng, Yunfei Duan, Pan Ji, and Hongdong Li. Consistnet: Enforcing 3d consistency for multi-view images diffusion. *arXiv*, 2023.

Jianglong Ye, Peng Wang, Kejie Li, Yichun Shi, and Heng Wang. Consistent-1-to-3: Consistent image to 3d view synthesis via geometry-aware diffusion models. *arXiv preprint arXiv:2310.03020*, 2023.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images, 2021. URL https://arxiv.org/abs/2012.02190.

Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19447–19456, 2024.

Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting, 2024. URL https://arxiv.org/abs/2404.19702.

Qiang Zhang, Seung-Hwan Baek, Szymon Rusinkiewicz, and Felix Heide. Differentiable point-based radiance fields for efficient view synthesis. In *SIGGRAPH Asia 2022 Conference Papers*, pp. 1–12, 2022.

Chuanxia Zheng and Andrea Vedaldi. Free3d: Consistent novel view synthesis without 3d representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9720–9731, 2024.

Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pp. 286–301. Springer, 2016.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *SIGGRAPH*, 2018.

A Appendix

We include additional results, ablations, and model details.

A.1 Naming Clarification

Importantly, we clarify that the naming of '*encoder*' and '*decoder*' are based on their output characteristics—i.e., the encoder outputs the latent while the decoder outputs the target—rather than being strictly tied to the transformer architecture they utilize. For instance, in the encoder-decoder model, the decoder consists of multiple transformer layers with self-attention (referred to as Transformer Encoder layers in the original transformer paper). However, we designate it as a decoder because its primary function is to output results. These terminologies align with conventions used in LLMs (Vaswani et al., 2017; Radford et al., 2019; Devlin et al., 2019). Notably, we apply self-attention to all tokens in every transformer block of both models, without introducing special attention masks or other architectural biases, in line with our philosophy of minimizing inductive bias.

A.2 Additional Implementation Details

We train LVSM with 64 A100 GPUs with a batch size of 8 per GPU. We use a cosine learning rate schedule with a peak learning rate of 4e-4 and a warmup of 2500 iterations. We train LVSM for 80k iterations on the object and 100k on scene data. LVSM uses a image patch size of $p = 8$ and token dimension $d = 768$. The details of transformer layers follow GS-LRM(Zhang et al., 2024) with an additional QK-Norm. Unless noted, all models have 24 transformer layers, the same as GS-LRM. The *encoder-decoder* LVSM has 12 encoder layers and 12 decoder layers, with 3072 latent tokens. Note that our model size is smaller than GS-LRM, as GS-LRM uses a token dimension of 1024.

For object-level experiments, we use 4 input views and 8 target views for each training example by default. We first train with a resolution of 256, which takes 4 days for the *encoder-decoder* model and 7 days for the *decoder-only* model. Then we finetune the model with a resolution of 512 for 10k iterations with a smaller learning rate of 4e-5 and a smaller total batch size of 128, which takes 2.5 days. For scene-level experiments We use 2 input views and 6 target views for each training example. We first train with a resolution of 256, which takes about 3 days for both *encoder-decoder* and *decoder-only* models. Then we finetune the model with a resolution of 512 for 20k iterations with a smaller learning rate of 1e-4 and a total batch size of 128 for 3 days. For both object and scene-level experiments, the view selection details and camera pose normalization methods follow GS-LRM. We use a perceptual loss weight λ as 0.5 and 1.0 on scene-level and object-level experiments, respectively.

We do not use bias terms in our model, for both Linear and LayerNorm layers. We initialize the model weights with a normal distribution of zero-mean and standard deviation of $0.02/(2 * (idx + 1)) * *0.5$, where *idx* means transform layer index.

We train our model with AdamW optimizer (Kingma, 2014). The β_1, and β_2 are set to 0.9 and 0.95 respectively, following GS-LRM. We use a weight decay of 0.05 on all parameters except the weights of LayerNorm layers.

A.3 Additional Visual Results

We show the visualization of LVSM at the object level with 256 resolution in Fig. 7. Consistent with the findings of the experiment with 512 resolution (Fig. 3), LVSM performs better than the baselines on texture details, specular material, and concave geometry.

A.4 Detailed model architecture

We have provided a detailed model architecture figure, as shown in Fig. 8.

A.5 Attention mask illustration for different design choices

In Fig. 9, we visualize the corresponding attention masks for the various design choices discussed in Sec. 4.4.

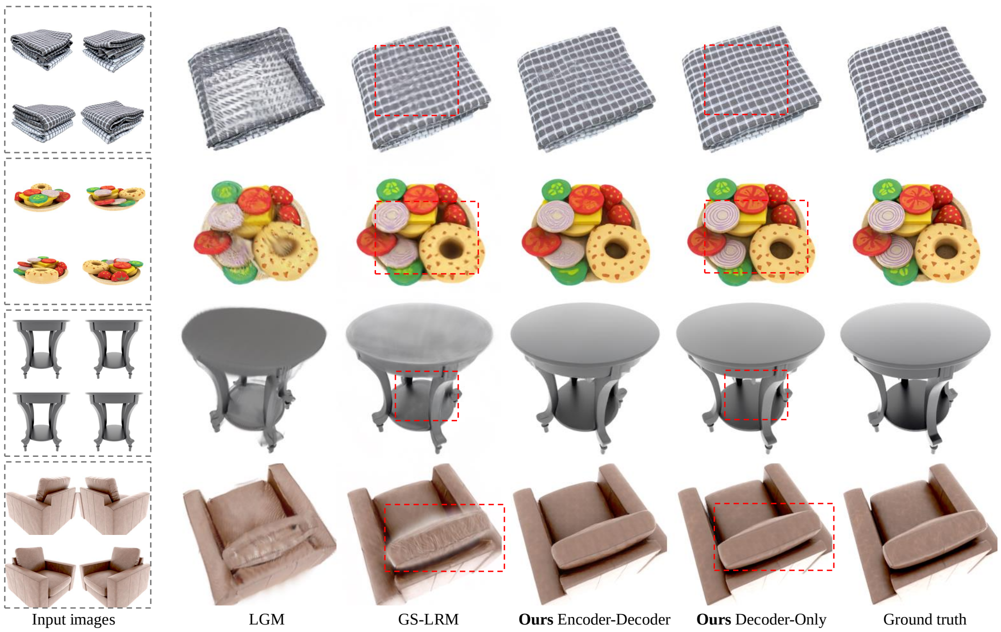

Figure 7: **Object-level visual comparison at 256 resolution.** Comparing with the two baselines: *LGM*(Tang et al., 2024) and *GS-LRM* (Res-256) (Zhang et al., 2024), both our Encoder-Decoder and Decoder-Only models have fewer floater artifacts (last example), and can generate more accurate view-dependent effects (third example). Our Decoder-Only model can better preserve the texture details (first two examples).

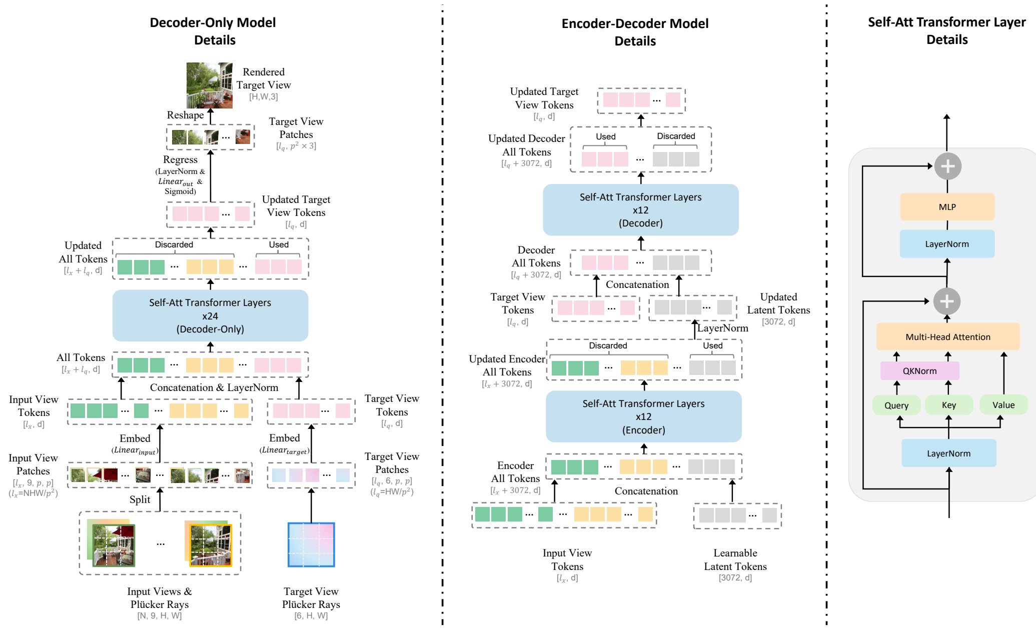

Figure 8: **Model Details.** We introduce two architectures: (1) an encoder-decoder LVSM, which encodes input image tokens into a fixed number of 1D latent tokens, functioning as a fully-learned latent scene representation, and decodes novel-view images from them; and (2) a decoder-only LVSM, which directly maps input images to novel view outputs, completely eliminating intermediate scene representations. Both models consist of pure self-attention blocks.

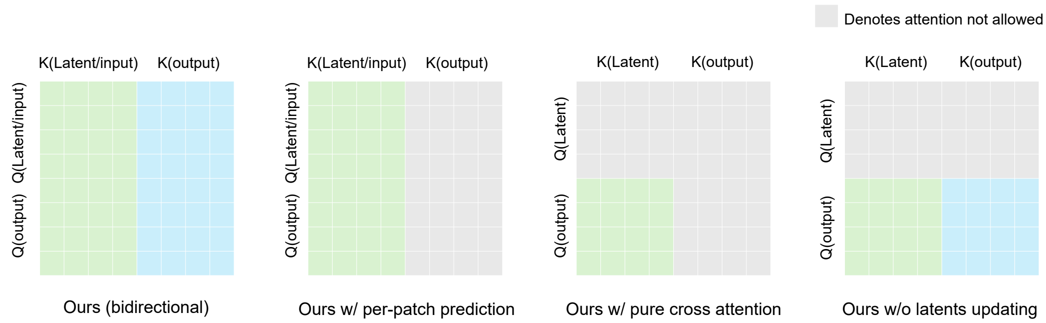

Figure 9: **Attention Mask Visualization.** Both our encoder-decoder and decoder-only models employ bidirectional self-attention modules. This figure visualizes the corresponding attention masks for the various design choices discussed in Sec. 4.4. (We use green color for the columns of latent/input tokens, and blue for the columns of output pixel patch tokens.) In our encoder-decoder architecture, the decoder utilizes pure self-attention, enabling latent tokens and different output target image tokens to jointly attend to each other. Consequently, latent tokens can be updated across transformer layers, while different output patch pixels can also attend to each other for joint updates. As shown in Table 3, disabling either the joint updating of output patch pixels (ours w/ per-patch prediction) or the latents' updating (ours w/o latents' updating) significantly degrades performance. Prior work, SRT (Sajjadi et al., 2022), eliminates both mechanisms by employing a decoder with pure cross-attention (ours w/ pure cross-att decoder), leading to even worse performance. Similarly, in our decoder-only model, disabling the joint updating of output patch pixels (ours w/ per-patch prediction) also results in a notable performance drop.

A.6 Discussion of differences with prior generative NVS models

Motivated by the success of the previous NVS geometry-free approaches (Sitzmann et al., 2021; Sajjadi et al., 2022), and the effectiveness of diffusion models in image-to-image tasks (Saharia et al., 2022a; Ramesh et al., 2022; Saharia et al., 2022b), 3DiM (Watson et al., 2022) explores training image-to-image diffusion models for object-level multi-view rendering to perform novel view synthesis without an explicit 3D representation. However, 3DiMs is trained from scratch using limited 3D data (Sitzmann et al., 2019), limiting it to category-specific settings and without zero-shot generalization.

The following work Zero-1-to-3 (Liu et al., 2023a) adopts a similar pipeline without a 3D representation but fine-tunes its model from a pretrained 2D diffusion model using a larger 3D object dataset (Deitke et al., 2023), achieving better generalization and higher quality. However, Zero-1-to-3's view synthesis results suffer from inherent multi-view inconsistency because it is a probabilistic model and it generates one target image at a time independently.

To address this inconsistency problem, several works (Liu et al., 2023b; Yang et al., 2023; Ye et al., 2023) integrate additional forms of 3D inductive bias, such as a 3D representation, epipolar attention, etc., into the diffusion denoising process, leading to increased computational cost. Other approaches (Li et al., 2023; Shi et al., 2023a;b; Long et al., 2023) predict a single image grid representing (specific) multi-view images with fixed camera pose, sacrificing the ability to control the camera. More recent works, including Free3D (Zheng & Vedaldi, 2024), EscherNet (Kong et al., 2024) CAT3D (Gao* et al., 2024), SV3D (Voleti et al., 2025), etc, jointly predict multiple target views with accurate camera control while ensuring view consistency by integrating cross-view attention. However, these methods guarantee consistency only for the finite set of jointly predicted views.

In contrast, our generalizable deterministic models do not possess the same inherent inconsistency issues of probabilistic models. As demonstrated in the video results on our project webpage, after being trained on large-scale multi-view data, our models can independently generate each target image with precise camera control while maintaining view consistency—without relying on the cross-view attention mechanisms employed by previous generative models. This capability enables our models to generate an unlimited number of consistent views for the observed regions of reconstructed scenes, unlike prior generative models. Nonetheless, our deterministic models have their own inherent limitations, i.e., they can't hallucinate unseen regions, which are discussed in Appendix A.7.

A.7 LIMITATIONS

Our models are deterministic, and like all prior deterministic approaches (Chen et al., 2021; Wang et al., 2021a; Sajjadi et al., 2022; Wang et al., 2023; Zhang et al., 2024), they struggle to produce high-quality results in unseen regions. Previous 3D-based deterministic models typically generate blurry artifacts for those regions due to uncertainty, whereas our model often generates noisy and flickering artifacts with fixed patterns. To illustrate this, we provide video examples of related failure cases on our webpage. Incorporating generative techniques or combining generative methods with our model could help solve this issue, which we leave as a promising future direction.

Additionally, our model's performance degrades when provided with images with aspect ratios and resolutions different from those seen during training. For instance, when trained on 512×512 images and tested on 512×960 input images, we observe high-quality novel view synthesis at the center of the output but blurred regions at the horizontal boundaries that extend beyond the training aspect ratio. We hypothesize that this limitation arises because our model is trained on center-cropped images. Specifically, the Plücker ray density is smaller at the boundaries of the image's longer side, and since our model is not trained on such data, it struggles to generalize. Expanding the training dataset to include more diverse image resolutions and aspect ratios could help address this issue.