

# Long-LRM: Long-sequence Large Reconstruction Model for Wide-coverage Gaussian Splats

Chen Ziwen<sup>1\*</sup> Hao Tan<sup>2</sup> Kai Zhang<sup>2</sup> Sai Bi<sup>2</sup> Fujun Luan<sup>2</sup> Yicong Hong<sup>2</sup>  
Li Fuxin<sup>1</sup> Zexiang Xu<sup>3</sup>  
<sup>1</sup>Oregon State University <sup>2</sup>Adobe Research <sup>3</sup>Hillbot

## Abstract

We propose *Long-LRM*, a feed-forward 3D Gaussian reconstruction model for instant, high-resolution, 360° wide-coverage, scene-level reconstruction. Specifically, it takes in 32 input images at a resolution of 960×540 and produces the Gaussian reconstruction in just 1 second on a single A100 GPU. To handle the long sequence of **250K** tokens brought by the large input size, *Long-LRM* features a mixture of the recent Mamba2 blocks and the classical transformer blocks, enhanced by a light-weight token merging module and Gaussian pruning steps that balance between quality and efficiency. We evaluate *Long-LRM* on the large-scale DL3DV benchmark and Tanks&Temples, demonstrating reconstruction quality comparable to the optimization-based methods while achieving an **800×** speedup w.r.t. the optimization-based approaches and an input size at least **60×** larger than the previous feed-forward approaches. We conduct extensive ablation studies on our model design choices for both rendering quality and computation efficiency. We also explore *Long-LRM*'s compatibility with other Gaussian variants such as 2D GS, which enhances *Long-LRM*'s ability in geometry reconstruction. Project page: <http://arthurhero.github.io/projects/llrm/>

## 1. Introduction

3D reconstruction from multi-view images is a fundamental problem in computer vision, with applications ranging from 3D content creation, VR/AR, to autonomous driving and robotics. Recently, NeRF [36] and various radiance field-based methods [2, 6, 37, 56] have shown great potential in reconstructing high-quality 3D scenes from a set of posed images via differentiable rendering. However, these models are slow to reconstruct and not generalizable to unseen scenes, as they require optimization for each scene independently. While 3D Gaussian splatting (GS) [23] has significantly advanced reconstruction and rendering efficiency, it still typically requires at least 10 minutes to optimize for each scene

\*Research partially done when Chen Ziwen was an intern at Adobe Research



Figure 1. Long-LRM reconstructs a real-world scene with wide viewing coverage from 32 images at a resolution of  $960 \times 540$  in just 1 second. As a feed-forward generalizable model, Long-LRM achieves instant Gaussian reconstruction with high rendering quality comparable to the optimization-based 3D Gaussian splatting methods, which require over 10 minutes to converge on each scene.

and cannot achieve instant reconstruction.

To address this, generalizable 3D GS models [46, 48] have been proposed to enable fast feed-forward GS reconstruction, avoiding per-scene optimization. Several methods [4, 7, 30, 64] have shown promising scene-level reconstruction results on real 3D captures by regressing per-pixel Gaussian primitives. In particular, GS-LRM [64], following the principles of 3D large reconstruction models (LRMs) [17, 25, 52] and using transformer blocks [50] without 3D inductive biases such as epipolar attention or sweeping volumes, has achieved state-of-the-art novel-view rendering quality on multiple challenging datasets. However, these pre-

vious feed-forward GS models are designed to handle only a small number of input images (typically 1-4) with limited viewing coverage, and thus are incapable of reconstructing large real-world scenes, which require at least dozens of images to capture a wide view span. In such cases, optimization-based methods were still the only viable option.

To this end, we propose Long-LRM, which achieves *high-resolution, 360° wide-coverage scene-level Gaussian reconstruction from 32 input images at 960×540 resolution in just 1 second* on a single A100 80G GPU. The photorealistic novel-view renderings produced by our approach have a quality comparable to or even better than 3D GS [23] that takes over 10 minutes for per-scene optimization (Fig. 1).

As inspired by GS-LRM, we patchify the multi-view input images into a sequence of tokens and consider the task of GS reconstruction as a sequence-to-sequence translation to regress pixel-aligned Gaussian primitives. However, unlike GS-LRM that focuses on 2-4 input images, our input setting with 32 960×540 images corresponds to an extremely long token sequence – **about 250K tokens** (considering a patch size of 8×8) – even longer than many modern large language models, such as LLama3 [13] with a context length of 128K.

This long token length is highly challenging for transformer blocks due to their quadratic time complexity. To address this challenge, we leverage recent advancements in state space models (SSMs) [15], designed to handle long-context reasoning efficiently with linear complexity. In particular, we propose a novel LRM architecture that combines Mamba2 [11] blocks with transformer blocks, enabling efficient sequential long-context reasoning while preserving critical global context. Additionally, we introduce a lightweight token merging module to further reduce the number of tokens in the middle of the network processing, along with a Gaussian pruning step to encourage efficient use of the dense per-pixel Gaussians. These combined designs allow us to train our Long-LRM using similar computational resources to GS-LRM, while successfully scaling up the token length by more than 60×, enabling fast, high-quality, wide-coverage reconstruction of real-world scenes.

We train our Long-LRM on the recent DL3DV10K dataset [29], which comprises approximately 10K diverse indoor and outdoor scenes. We evaluate our model on both the DL3DV140 benchmark and the Tanks and Temples dataset [24], using 32 input images for each scene. Our direct feed-forward reconstruction achieves comparable novel view synthesis quality to the optimization-based 3D GS approaches, while substantially reducing the reconstruction time (1 second vs. 13 minutes). We conduct extensive ablation studies on our model design choices for both rendering quality and computation efficiency. We also explore Long-LRM’s compatibility with other GS variants such as 2D GS [18]. **Long-LRM is the first feed-forward GS solution for wide-coverage scene-level reconstruction in a few seconds.**

## 2. Related Work

**3D Reconstruction.** Many traditional 3D reconstruction methods have focused on pure geometry reconstruction, where surface meshes [3, 38, 43, 45] or depth maps [8, 14, 21, 40, 41, 59, 63] are the target output. These methods usually involve explicit feature matching along the epipolar lines, followed by TSDF or depth prediction performed by neural networks. In contrast, we leverage the 3D GS representation to simultaneously reconstruct geometry and color, enabling photo-realistic novel view synthesis.

**Neural reconstruction and rendering.** Instead of directly predicting surface geometry, NeRF [36] proposes to leverage differentiable volume rendering to regress novel view images, supervised with a rendering loss. This implicit way of reconstruction eliminates the need for hard-to-obtain ground-truth 3D supervision while producing visually pleasing reconstruction results. However, NeRF requires optimizing an MLP network for each scene independently, which takes hours or even days. Follow-up works have introduced advanced neural scene representations [1, 2, 6, 37, 47, 56], significantly improving time and memory efficiency. Among these, 3D Gaussian splatting [23] stands out for reducing reconstruction time to just dozens of minutes while maintaining high reconstruction quality and enabling real-time rendering. Variants of 3D GS [28, 33, 39] further extend its capabilities to large-scale city-level reconstruction, but they still require minutes to optimize on each scene.

**Generalizable NeRF and 3D GS.** Previous attempts to develop generalizable NeRF models have primarily relied on classical projective geometric structures, such as epipolar lines [31, 44, 53, 61] or plane-sweep cost volumes [5, 20, 27, 65], to aggregate multi-view features from nearby views for local NeRF estimation. Recently, similar designs have been adapted to enable feed-forward scene-level 3D GS reconstruction with generalizable models [4, 7, 30]. However, since both epipolar geometry and plane-sweep volumes depend on significant overlap between input views, these GS-based methods (as well as most prior NeRF-based methods) are limited to local reconstructions from a small number (1-4) of narrow-span inputs. On the other hand, GS-LRM [64] avoids these 3D-specific structural designs and adopts global-attention transformer blocks, achieving state-of-the-art performance in this domain. However, constrained by the quadratic complexity of transformers, GS-LRM focuses on local reconstruction from only 2-4 views. On the other hand, Gamba [42] and MVGamba [60] utilize Mamba-based architectures with linear time complexity for object-level GS reconstruction. But they cannot perform scene-level reconstruction and still focus on only 1-4 input views. Our model is instead a novel hybrid model that combines transformer and Mamba2 blocks, designed for 360° wide-coverage, scene-level reconstruction from 32 high-resolution images.

**Efficient models for long sequences.** Transformer-based 3D large reconstruction models (LRMs) [17, 25, 52, 54, 55, 57, 64] have emerged for enabling high-quality 3D reconstruction from sparse-view inputs. While transformers dominate various AI fields due to their flexibility with input modalities and scalability in model sizes, their quadratic time complexity makes them extremely slow when handling long sequences, often requiring thousands of GPUs for parallel computing [13]. Efficient architectures such as linear attention [22] and structured state space model (SSM) [16] are thus proposed in NLP to deal with a large corpus of text. Mamba [15], a variant of SSM, offers significant performance improvements by computing state parameters from each input in the sequence and has been successfully extended to tackle vision tasks [12, 19, 26, 32, 42, 60, 67]. Mamba2 [11] further restricts the state matrix  $A$  and expands the state dimensions, showing performance comparable to transformers on multiple language tasks. However, empirical studies [51] indicate that transformers still outperform Mamba2 in in-context learning and long-context reasoning, both critical for 3D reconstruction. Inspired by [51] and Jamba [26], we propose a hybrid architecture combining transformer and Mamba2 blocks for long token-sequence 3D GS reconstruction, achieving a balance between training efficiency and reconstruction quality.

### 3. Method

In this section, we present our Long-LRM method. We give an overview in Sec. 3.1, the implementation details of the Mamba2 blocks in Sec. 3.2 and additional efficiency improvement designs (e.g., token merging) in Sec. 3.3. We end with a discussion of the training objectives in Sec. 3.4 that help the model to effectively converge.

#### 3.1. Overall Architecture

As shown in Fig. 2, we first tokenize the channel-wise concatenated RGB images and Plücker rays. Similar to GS-LRM [64], we view the per-pixel GS prediction as a sequence-to-sequence mapping. But crucially, we use a hybrid of Mamba2 blocks and transformer blocks, following the studies in [51] and [26], for better scalability to higher resolution and denser views.

In our implementation, each hybrid block consists of 7 Mamba blocks and one transformer block, which we empirically observe to be a balanced configuration and easier to converge. For the transformer blocks, we use global self-attention. We detail our implementation of Mamba2 blocks in Sec. 3.2. A token merging module is injected between the hybrid blocks to further speed up the processing (Sec. 3.3).

We decode per-pixel Gaussian parameters from the output tokens using a linear layer and apply training-time and test-time Gaussian pruning to improve efficiency at high resolution settings.

#### 3.2. Mamba2 Block

A Mamba block [15], similar to a transformer block, processes a token sequence of shape  $L \times D$  by mixing the token information, and outputs a token sequence of the same shape. For a sequence of length  $L$ , the transformer block has a computational complexity of  $O(L^2)$  while Mamba effectively reduces it to  $O(L)$ . Thus, it is suitable for the dense reconstruction task in our Long-LRM.

As a variant of SSM, Mamba processes each input token  $x_t$  by formula  $h_t = \mathbf{A}h_{t-1} + \mathbf{B}x_t$  and  $y_t = \mathbf{C}h_t$ , where  $h_t$  is the hidden state,  $y_t$  is the output token,  $t$  is the sequence index, and  $\mathbf{A}, \mathbf{B}, \mathbf{C}$  are parameters. Different from previous work [16], Mamba computes  $\mathbf{A}, \mathbf{B}, \mathbf{C}$  from the input instead of storing them as model parameters. The novel Mamba2 [11] block improves over Mamba by restricting the state matrix  $\mathbf{A}$  to be a scalar times an identity matrix, allowing the usage of efficient block multiplication and expansion to larger state dimensionality, showing performance comparable to transformers on multiple language tasks. However, since the Mamba2 block is designed for language tasks, it only scans through the tokens in one direction, which is sub-optimal for images. Following [67], we take bi-directional scans over the concatenated token sequence. Specifically, we first compute the state parameters from the input using one linear layer; then we run the SSM block in both forward and backward directions on the token sequence. Finally, we sum up the output tokens from the two scans before going through another linear layer. We have conducted some preliminary exploration of more complex scan patterns as in VMamba [32] and LocalMamba [19], but decided not to adopt those due to a substantial decrease in speed.

#### 3.3. Token Merging and Gaussian pruning

With 32  $960 \times 540$  input images and patch size 8, the sequence length can reach 250K, highly challenging even for linear-complexity models like Mamba. Empirically, we find even the all-Mamba2 variant of our model runs out of memory under the highest resolution setting (Tab. 4). To further reduce memory usage, we propose to merge the tokens in the middle of the network.

Our light-weight token merging module is inspired by the traditional multi-level CNN encoders and effectively reduces token sequence length down to 1/4. We first reshape the token sequence from  $L \times D$  back to  $N \times \frac{H}{p} \times \frac{W}{p} \times D$  where  $p$  is the patch size. Then, we apply a channel-wise  $2 \times 2$  2D convolution with stride 2, resulting in output shape  $N \times \frac{H}{2p} \times \frac{W}{2p} \times D'$ , where  $D'$  is the new token dimensionality that can differ from the original one. Finally, we reshape it back to  $\frac{L}{4} \times D'$  where each token now has an ‘effective’ patch size of  $2p$ . We conduct ablation studies on the position of token merging in the network (Tab. 5) and the effect of token dimensionality change (Tab. 4), showing a balance between computational efficiency and reconstruction quality.

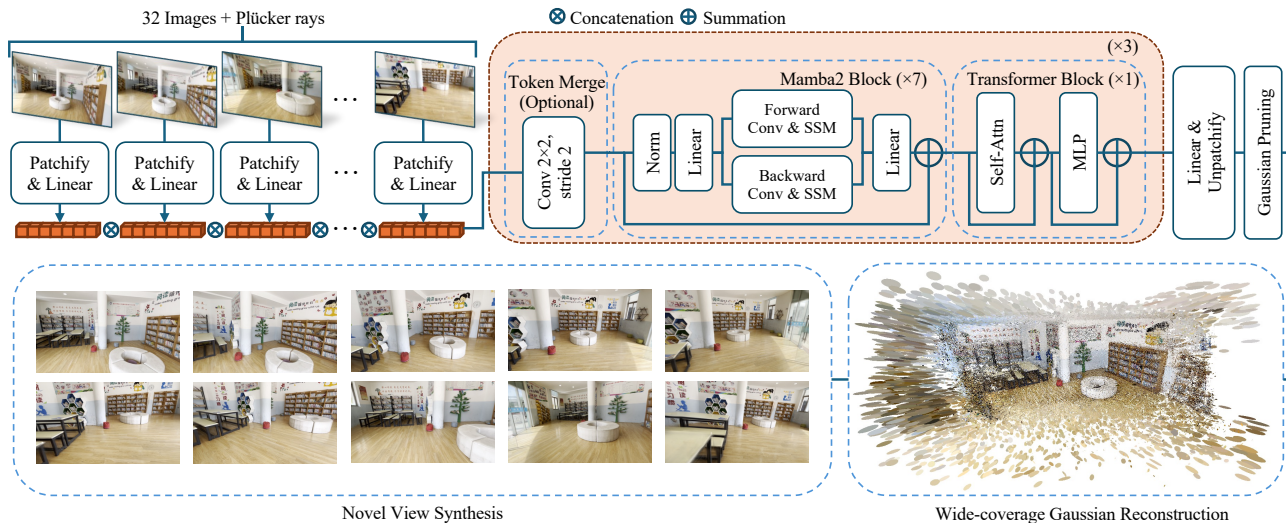


Figure 2. Long-LRM takes 32 input images along with their Plücker ray embeddings as model input, which are then patchified into a token sequence. These tokens are processed through a series of Mamba2 and transformer blocks ( $\{7\text{MIT}\} \times 3$ ). Fully processed, the tokens are decoded into per-pixel Gaussian parameters, followed by a Gaussian pruning step. The bottom section illustrates the resulting wide-coverage Gaussian reconstruction and photo-realistic novel view synthesis.

However, even with token merging, the per-pixel Gaussian prediction still brings an enormous quantity of Gaussians ( $\sim 17$  million for 32 images with resolution  $960 \times 540$ ), posing intense pressure on memory and speed. There is likely a huge redundancy because of the high resolution of inputs and the overlap between the input camera frustums. Thus, a Gaussian pruning step is necessary.

Naively pruning away the most transparent Gaussians, even the ones with only 5% opacity, results in incomplete *opaque* objects in the reconstructed scene, hinting that the model tends to diffuse an opaque color into multiple translucent Gaussians. To encourage the model to use a compact set of Gaussians, we apply a punishment on the opacity of all Gaussians (detailed in Sec. 3.4). With the reduced number of visible Gaussians, we can thus simply prune away Gaussians with low opacity. Empirically, we find no difference in rendering quality if removing Gaussians with opacity below 0.001. We apply Gaussian pruning before rendering in both training and inference at the  $960 \times 540$  resolution setting. We use a fixed-percentage pruning instead of an opacity threshold to ensure constant training memory usage.

### 3.4. Training Objectives

Below, we list the training objectives for Long-LRM:

**Rendering loss.** Following previous work [64], we use a combination of Mean Squared Error (MSE) loss and Perceptual loss

$$\mathcal{L}_{\text{image}} = \frac{1}{M} \sum_{i=1}^M \left( \text{MSE} \left( \mathbf{I}_i^{\text{gt}}, \mathbf{I}_i^{\text{pred}} \right) + \lambda \cdot \text{Perceptual} \left( \mathbf{I}_i^{\text{gt}}, \mathbf{I}_i^{\text{pred}} \right) \right) \quad (1)$$

to supervise the quality of the rendered images, where  $\lambda$  is set to 0.5. While training solely with rendering loss can

already achieve competitive visual quality (see Sec. 5.2), we further introduce two regularization terms to improve training stability and computational efficiency.

**Depth regularization for training stability.** Training instability is a well-known curse for large-scale training. In our task, we observe that the instability comes from the difficulty in optimizing the Gaussian positions. With rendering loss only, the model will produce ill-posed Gaussians known as “floaters”, which do not lie on the actual 3D surface – a common issue for novel view synthesis (see the black “floaters” in Fig. 1). To stabilize training, we add a scale-invariant depth loss

$$\mathcal{L}_{\text{depth}} = \frac{1}{M} \sum_{i=1}^M \text{Smooth-L1} \left( \mathbf{D}_i^{\text{da}}, \mathbf{D}_i^{\text{pred}} \right) \quad (2)$$

where  $\mathbf{D}_i^{\text{da}}$  is the disparity map predicted by DepthAnything [58], and  $\mathbf{D}_i^{\text{pred}}$  is the disparity map obtained from the predicted position of the per-pixel Gaussians. Following [58], we normalize the disparity maps by subtracting their medians  $t(d_i)$  and then dividing by their mean absolute deviation  $\frac{1}{HW} \sum |d_i - t(d_i)|$ . This soft depth supervision effectively helps reduce the chance of the training divergence.

**Opacity regularization for inference efficiency.** Since our per-pixel prediction strategy renders a dense set of Gaussians, to encourage an efficient use of the Gaussians, we apply a small L1 regularization on the opacity

$$\mathcal{L}_{\text{opacity}} = \frac{1}{N} \sum_{i=1}^N |o_i| \quad (3)$$

where the opacity values are between 0 and 1. Intuitively, L1 can encourage the sparsity of the regularized terms [49].

We empirically observe that adding this loss can drastically push the percentage of Gaussians with opacity above 0.001 from 99% down to around 40% (see Tab. 7). With these near-zero opacity Gaussians, we can perform Gaussian pruning as discussed in Sec. 3.3 and reduce both the Gaussian rendering time and the backpropagation time.

**Overall training loss.** Our total loss is thus the rendering loss and the weighted regularization loss terms:

$$\mathcal{L} = \mathcal{L}_{\text{image}} + \lambda_{\text{opacity}} \cdot \mathcal{L}_{\text{opacity}} + \lambda_{\text{depth}} \cdot \mathcal{L}_{\text{depth}} \quad (4)$$

where we set  $\lambda_{\text{opacity}} = 0.1$  and  $\lambda_{\text{depth}} = 0.01$ .

## 4. Experiments

### 4.1. Implementation and Experiment Details

**Architecture Details.** Our model consists of 24 blocks, with every 7 Mamba2 blocks followed by 1 transformer block, repeating 3 times. We start with a patch size of 8 and a token dimensionality of 256. We perform token merging at the beginning of the 9th block, with patch size expanded to 16 and token dimensionality expanded to 1024. For Mamba2 blocks, we use a 256-dimensional state, an expansion rate of 2 and a head dimensionality of 64. For transformer blocks, we use a head dimensionality of 64 and an MLP dimension ratio of 4. We use the FlashAttentionV2 [10] implementation that optimizes GPU IO utilization for long sequences.

**Training Settings.** Directly training the model on high-resolution images is extremely inefficient; therefore, we opt for a low-to-high-resolution curriculum training schedule, with three training stages, using image resolutions of  $256 \times 256$ ,  $512 \times 512$  and  $960 \times 540$ .

Specifically, in the first stage, training images are resized so that the shorter side is 256 and then center-cropped to a square shape. For the training view selection, we first randomly pick a consecutive subsequence ranging from 64 frames to 128 frames, then uniformly sample 32 images as input and randomly sample 8 images as target. Input and target are sampled independently and thus can overlap. We randomly shuffle or reverse the input view order with probability 0.5. We train with a peak learning rate of  $4E-4$  and the AdamW optimizer [34] with a weight decay of 0.05. The learning rate is linearly warmed up in the first 2K steps and then cosine decayed. We use a batch size of 256 and train for 60K steps.

In the 2nd stage, we resize and crop the images to  $512 \times 512$ , decrease the peak learning rate to  $4E-5$ , and train the model for 10K steps at batch size 64. The view selection protocol remains the same.

In the last stage, we resize the images to  $960 \times 540$  without square cropping, expand the view sampling range to the entire sequence (about 200~300 frames for DL3DV), and keep training the model for another 10K steps with batch

size 64. We perform Gaussian pruning at this stage to save GPU memory usage, where we only keep top 40% Gaussians ranked by opacity, plus 10% randomly sampled from the rest. We augment the FOV of the images by randomly center-cropping the images to  $0.77 \sim 1.0$  of the original size and resize back, in order to fit a broader range of camera models.

**Evaluation Settings.** During evaluation, our goal is to reconstruct the scene captured by the entire video sequence. Following the protocol of previous novel-view synthesis works [1, 23], we uniformly pick every 8-th image of the sequence as the test split. From the rest of the sequence, we use  $K$ -means clustering (based on camera positions and viewing directions) to pick a set of input views that can best cover the entire scene. The number of clusters is set to the number of input views. The cameras closest to the cluster centers are chosen as the input split. We use an image resolution of  $960 \times 540$  during the evaluation. We perform Gaussian pruning during evaluation by only keeping the top 50% of the Gaussians with highest opacity values, which is an empirically safe range with negligible quality loss.

**Post-prediction Optimization.** We provide a post-prediction optimization option to further improve reconstruction quality. After Long-LRM predicts the initial set of Gaussians, we render at the input camera views, compute MSE loss between the rendered images and the input images, and back-propagate gradients to the Gaussian parameters. We set a learning rate of  $5E-4$  for position,  $1E-3$  for color, and keep opacity, scale and rotation unchanged.

**Training with 2D GS.** We additionally explore Long-LRM’s compatibility with 2D GS [18], a variant of 3D GS known for its strength in geometry reconstruction and depth map rendering. Specifically, after completing the first two training stages, we adapt the original Long-LRM to 2D GS by directly interpreting the predicted Gaussian parameters as 2D Gaussian parameters. We then fine-tune the model for an additional 10K steps at a resolution of  $512 \times 512$ , followed by the same third-stage training as in the original pipeline.

### 4.2. Datasets

DL3DV [29] is a large-scale, real-world scene dataset for 3D reconstruction and novel view synthesis. It features diverse scene types, with both indoor and outdoor captures. The DL3DV-10K split consists of 10,510 high-resolution videos, each accompanied by 200~300 keyframes with camera pose annotation obtained from COLMAP [41]; the DL3DV-140 Benchmark split contains 140 test scenes. We train our model on DL3DV-10K and evaluate on the DL3DV-140 Benchmark. We also perform zero-shot inference on Tanks&Temples [24], another challenging real-world scene dataset, to show the generalization capability of Long-LRM.

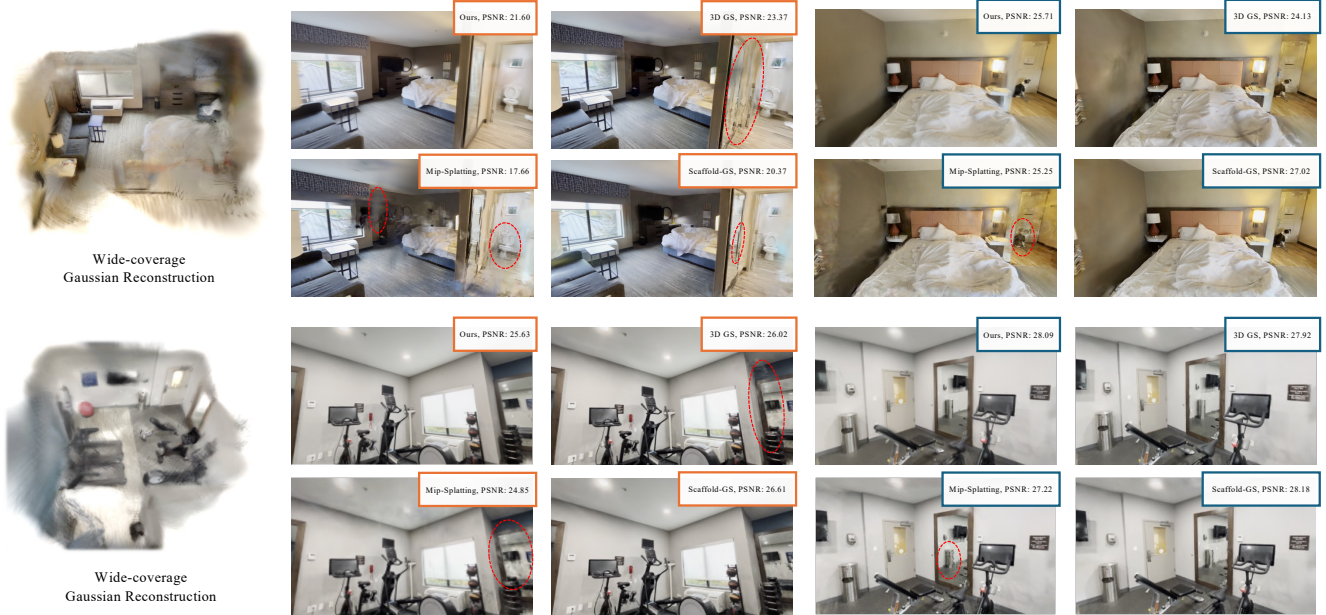


Figure 3. Qualitative comparisons between Long-LRM (1 second) and optimization-based 3D GS methods (13+ minutes), reconstructed from 32 images at  $960 \times 540$  resolution. The leftmost column shows the overlook of our wide-coverage Gaussian reconstruction, while the rest columns show the color renderings from the reconstructed 3D Gaussians. Our approach maintains high-quality reconstruction with competitive PSNR, demonstrating the ability to generate accurate details and fewer artifacts in challenging regions. Red ellipses highlight areas where optimization-based methods struggle with artifacts or inaccuracies, whereas Long-LRM produces cleaner outputs.

Following previous work [23, 30], we use the `train` and the `truck` scene. To compare with previous feed-forward GS methods, we also evaluate Long-LRM under a low-resolution, two-view setting on RealEstate10K [66], a real-world indoor scene dataset, following the same train-test split and evaluation setting introduced by pixelSplat [4]. To test Long-LRM’s instant geometry reconstruction capability after equipped with 2D GS, we perform zero-shot depth map evaluation on ScanNetv2 [9] (see supplementary), a classic indoor scene dataset for multi-view stereo.

### 4.3. Results

Long-LRM achieves wide-coverage, scene-level 3D Gaussian splatting reconstruction from 32 high-resolution input images, which, to the best of our knowledge, no other method have accomplished. Recent work pixelSplat [4], MVSpLat [7], MVSGaussian [30], and GS-LRM [64] are limited to processing 1–4 input images, with pixelSplat and MVSpLat showing results only at  $256 \times 256$  resolution. Most of these methods rely on traditional 3D inductive biases, such as epipolar projection and cost volumes, which are suited for narrow-view inputs with large overlaps but struggle with wide-coverage, high-resolution settings. Moreover, naively extending these methods to handle more input views and higher resolutions leads to out-of-memory issues and requires significant architectural changes. Therefore, we compare Long-LRM with the *optimization-based 3D GS methods in the high-resolution, wide-coverage setting*, and compare

Input Views	Method	Time↓	DL3DV-140			Tanks&Temples		
			PSNR↑	SSIM↑	LPIS↓	PSNR↑	SSIM↑	LPIS↓
16	3D GS <sub>30k</sub>	13min	21.20	0.708	0.264	16.76	0.598	0.334
	Mip-Splatting <sub>30k</sub>	13min	20.88	0.712	0.274	16.82	0.616	0.332
	Scaffold-GS <sub>30k</sub>	16min	22.13	0.738	<b>0.250</b>	17.02	<b>0.634</b>	<b>0.321</b>
	Ours	<b>0.4sec</b>	<b>22.66</b>	<b>0.740</b>	0.292	<b>17.51</b>	0.555	0.408
32	3D GS <sub>30k</sub>	13min	23.60	0.779	0.213	18.10	0.688	0.269
	Mip-Splatting <sub>30k</sub>	13min	23.32	0.784	0.217	18.39	<b>0.700</b>	<b>0.262</b>
	Scaffold-GS <sub>30k</sub>	16min	24.77	0.805	<b>0.205</b>	18.41	0.691	0.290
	Ours	<b>1sec</b>	24.10	0.783	0.254	18.38	0.601	0.363
	Ours <sub>3</sub>	12sec	24.99	0.809	0.243	18.69	0.623	0.360
	Ours <sub>10</sub>	37sec	<b>25.60</b>	<b>0.826</b>	0.233	<b>18.90</b>	0.642	0.350

Table 1. Quantitative comparison to optimization-based 3D GS approaches on full-scene reconstruction. Subscripts indicate number of optimization steps. Our post-prediction optimization step (3.6sec) optimizes over all input images. ‘Time’ refers to the total inference/optimization time for each scene. Image resolution is  $960 \times 540$ .

Method	PSNR↑	SSIM↑	LPIS↓	Input Views	Method	Time↓	PSNR↑	SSIM↑	LPIS↓
pixelSplat	25.89	0.858	0.142	32	2D GS <sub>30k</sub>	13min	24.35	<b>0.813</b>	<b>0.203</b>
MVSpLat	26.39	0.869	0.128		Ours	<b>1sec</b>	23.24	0.755	0.287
GS-LRM	28.10	0.892	0.114		Ours <sub>10</sub>	37sec	<b>24.74</b>	0.800	0.263
Ours	<b>28.54</b>	<b>0.895</b>	<b>0.109</b>						

Table 3. Comparison between Long-LRM (2D GS ver.) and optimization-based 2D GS [18]

Table 2. Quantitative comparison on RealEstate10K under 2-view setting. Image resolution is  $256 \times 256$ .

Table 3. Comparison between Long-LRM (2D GS ver.) and optimization-based 2D GS [18] on color reconstruction quality on DL3DV-140 at resolution  $960 \times 540$ .

with *previous feed-forward methods in a low-resolution, two-input setting*.

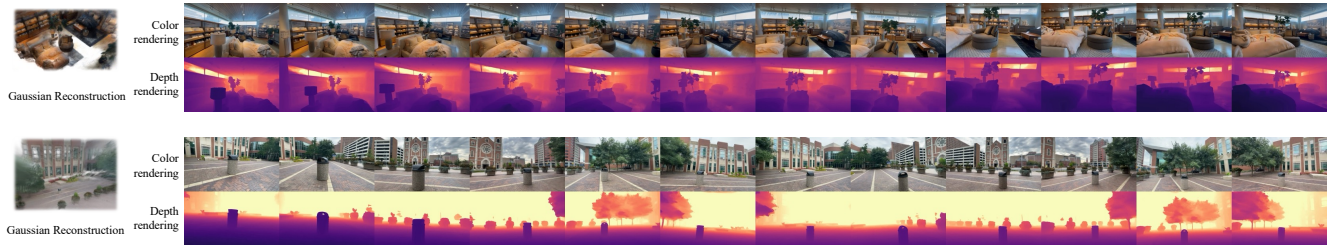


Figure 4. Qualitative results showcasing Long-LRM’s color and depth reconstruction capabilities on two wide-span scenes (the second being 360°). At a sequence of novel camera views, we render RGB images from the 3D Gaussians predicted by Long-LRM, and depth maps from the 2D Gaussians predicted by Long-LRM finetuned with 2D GS. The depth maps exhibit sharp object edges and smooth surfaces, demonstrating Long-LRM’s ability to generalize and inherit strengths from different GS variants.

**High-resolution, wide-coverage reconstruction.** In Table 1, we show a quantitative comparison with the optimization-based 3D GS methods [23, 35, 62] on two real-world wide-span scene datasets: DL3DV-140 [29] and Tanks& Temples [24]. We show results under a sparser 16 input-view setting and the 32 input-view setting. Our feed-forward model achieves an 800× speedup for reconstructing an unseen novel scene (1 second vs. 13 minutes), while our reconstruction quality is comparable with the 30K-step optimization results, and can be further boosted with just a few post-prediction optimization steps. Our model takes the lead in terms of PSNR (+2dB with Ours<sub>10</sub> vs. 3D GS<sub>30k</sub>), while optimization-based methods perform better in LPIPS. Qualitative results (Fig. 1 and 3) show that Long-LRM significantly reduces floater artifacts, which can be attributed to the prior knowledge distilled from a large training dataset and our regularization terms: the opacity loss and the soft depth supervision. More visualization and interactive results can be found on our project page and in the supplementary.

**Low-resolution, sparse-view reconstruction.** In Table 2, we present a quantitative comparison with the feed-forward GS methods [4, 7, 64] on RealEstate10K at a 256 × 256 resolution with 2 input views, a setting commonly used in prior works. Long-LRM achieves the best overall quality, outperforming [4] and [7] by a margin of over 2dB PSNR and slightly surpassing the transformer-based GS-LRM, highlighting the effectiveness of our hybrid architecture.

**Compatibility with other GS variants.** In Table 3, we show that after being finetuned for 2D GS prediction, the color rendering quality of Long-LRM is still comparable with the optimization-based counterpart, while Fig. 4 shows the rendered depth maps from the predicted 2D Gaussians, highlighting Long-LRM’s ability to generalize and inherit strengths from different GS variants. We also perform a zero-shot depth map and mesh quality evaluation on the ScanNetv2 dataset to quantitatively assess the geometry reconstruction ability of Long-LRM (see supplementary).

Overall, Long-LRM not only leads to state-of-the-art rendering quality in the classical sparse-view setting but also enables instant, wide-coverage, high-resolution, scene-level

reconstruction that no previous methods can achieve.

## 5. Analysis

### 5.1. Ablation Studies of Model Designs

In Table 4, we show how the model architecture variants scale with the input size in terms of both training efficiency and reconstruction quality. We consider 4 experimental setups: 1. sparse low-resolution (‘Input Views’=4, ‘Image Size’=256), 2. dense low-resolution (‘Input Views’=32, ‘Image Size’=256), 3. dense high-resolution (‘Input Views’=32, ‘Image Size’=512), 4. dense ultra-resolution (‘Input Views’=32, ‘Image Size’=960 × 540)<sup>1</sup>. The results of different model architecture under the same setup are presented within a Table block. We evaluate five model variants: (1) all Transformer blocks (equivalent to GS-LRM), (2) all Mamba2 blocks, (3) a hybrid model without token merging, (4) a hybrid model with token merging and a constant token dimensionality of 1024, and (5) our final model—a hybrid approach with token merging, starting with a token dimensionality of 256. All variants consist of 24 blocks in total. For each variant, we report the number of parameters (#Param), training iteration time, GPU memory usage, and PSNR reconstruction performance. The detailed experimental setup is provided in the supplementary.

**All transformer blocks.** The performance of transformer is comparable to the hybrid model under the 4-view 256-resolution setting. However, its training time explodes for larger inputs, either with denser views or higher resolutions. Under the 32-view 512-resolution setup, the per-iteration time with batch size=1 reaches the unaffordable 50.5 seconds due to the quadratic time complexity.

**All Mamba2 blocks.** The Mamba2 variant shows a more manageable increase in time but leads to a noticeable decline in reconstruction quality. In the 256-resolution, 4-view setup, the Mamba2 variant exhibits a 1.8 PSNR drop compared to our hybrid model, and the gap widens with longer sequences, reaching 2.5 PSNR for 32 views and 3.3 PSNR at 512 reso-

<sup>1</sup>Note that here the terminology of ‘sparse’, ‘dense’, ‘low’, ‘high’, ‘ultra’ are all relative. We use these terminology for simplicity and clarity.

Input Views	Image Size	Batch Size / GPU	Train Step	Block Type	Token Merge	Patch Size	Token Dimensionality	#Param	Iteration Time (sec)	GPU Memory (GB)	PSNR $\uparrow$			
4	256	16	100K	Transformer	/	8	1024	307M	2.3	44	21.13			
				Mamba2	/	8	1024	170M	2.8	35	19.82			
				{7MIT} $\times$ 3	/	8	1024	187M	2.6	35	<b>21.58</b>			
				{7MIT} $\times$ 3	@9	8 $\rightarrow$ 16	1024	198M	2.0	23	20.96			
				{7MIT} $\times$ 3	@9	8 $\rightarrow$ 16	256 $\rightarrow$ 1024	<b>142M</b>	<b>1.9</b>	<b>20</b>	20.43			
32	256	4	60K	Transformer	/	8	1024	307M	14.5	68	too slow			
				Mamba2	/	8	1024	170M	6.0	70	24.28			
				{7MIT} $\times$ 3	/	8	1024	187M	7.1	70	<b>26.82</b>			
				{7MIT} $\times$ 3	@9	8 $\rightarrow$ 16	1024	198M	5.4	35	25.81			
				{7MIT} $\times$ 3	@9	8 $\rightarrow$ 16	256 $\rightarrow$ 1024	<b>142M</b>	<b>3.5</b>	<b>25</b>	25.62			
32	512	1	10K*	Transformer	/	8	1024	307M	50.5	44	too slow			
				Mamba2	/	8	1024	170M	7.4	62	24.83			
				{7MIT} $\times$ 3	/	8	1024	187M	11.5	64	<b>28.16</b>			
				{7MIT} $\times$ 3	@9	8 $\rightarrow$ 16	1024	198M	7.0	60	27.69			
				{7MIT} $\times$ 3	@9	8 $\rightarrow$ 16	256 $\rightarrow$ 1024	<b>142M</b>	<b>4.0</b>	<b>23</b>	27.46			
32	960 $\times$ 540	1	10K*	<b>All other variants are out of memory.</b>			{7MIT} $\times$ 3	@9	8 $\rightarrow$ 16	256 $\rightarrow$ 1024	142M	12.6	53	27.32

Table 4. **Ablation studies on model architecture.** We study how the model architecture affects training time and memory efficiency as well as the reconstruction quality. **{7MIT} $\times$ 3** refers to our “7 Mamba2 blocks + 1 transformer block, repeating 3 times” model architecture. **@9** refers to the placement of token merging in front of the 9th block. “ $\rightarrow$ ” indicates the change accompanying the token merging step. \*The 512-resolution models are finetuned from the checkpoints of their 256-resolution counterparts, and the 960-resolution from the 512-resolution checkpoints.

lution, which is possibly due to Mamba’s state-based design, which struggles to capture long-range dependencies.

**Impact of Token Merging.** Our hybrid model gets the best of both worlds – the reconstruction quality of transformer and the speed of Mamba2. However, a token merging step is essential for training in the most challenging dense ultra-resolution setup. Reducing the token length to 1/4 while keeping a constant token dimensionality of 1024 reduces training time to 2/3 in the dense-view setup. Further reducing the initial token dimensionality to 256—scaling up to 1024 only after merging—saves an additional 1/3 of training time and memory, making ultra-resolution training feasible, where all other variants run out of memory. We analyze the effect of token merging placement in the network in Tab. 5, aiming at a balance between model size and reconstruction quality.

## 5.2. Ablation Studies of Training Objectives

**Impact of the regularization terms.** Table 6 presents the impact of the two regularization terms introduced in Sec. 3.4: opacity loss and depth supervision. We find that adding opacity loss significantly reduces the number of visible Gaussians (measured as the % of Gaussians with opacity above 0.001) while having a negligible effect on rendering performance. Depth supervision further improves rendering quality and helps prevent gradient explosions by guiding “floater” Gaussians toward the true surface. This encourages the model to use SH coefficients to represent color variations based on the viewing direction, rather than relying on translucent floater

Token Merge	Time (sec)	Mem (GB)	#Param	PSNR $\uparrow$
@1	2.0	21G	198M	<b>21.35</b>
@9	<b>1.9</b>	20G	142M	21.25
@17	<b>1.9</b>	<b>19G</b>	<b>86M</b>	20.99

Table 5. **Ablation studies on placement of token merging in the network** with 4-view 256-resolution setup.

Loss Type	PSNR $\uparrow$	% Gaussians w/ opacity > 0.001
rendering-only	20.43	99.2
+opacity	20.96	68.3
+opacity+depth	21.25	70.1

Table 6. **Ablation studies on training objectives** with 4-view 256-resolution setup. We study how the opacity loss and the depth supervision affect the reconstruction quality and the Gaussian usage.

Input Views	Image Size	Input Sampling Range (frame)	w/ opacity loss	% Gaussians w/ opacity > 0.001
4	256 $\times$ 256	16	$\times$	99.2
4	256 $\times$ 256	16	$\checkmark$	68.3
32	256 $\times$ 256	64 $\sim$ 128	$\checkmark$	41.8
32	512 $\times$ 512	64 $\sim$ 128	$\checkmark$	34.1
32	960 $\times$ 540	200 $\sim$ 300	$\checkmark$	33.3

Table 7. **Gaussian usage impacted by opacity loss and input size.**

Gaussians that act as lens filters in front of the cameras.

**Gaussian Usage.** Table 7 presents the impact of the opacity loss and input size on Gaussian usage (measured as the % of Gaussians with opacity above 0.001). As the input resolution increases—leading to more per-pixel Gaussians being predicted—the chance that multiple pixels can be covered by one Gaussian increases as well, causing a decrease in Gaussian usage. Meanwhile, when the viewing span increases and the overlap between input views decreases, the model needs to retain more Gaussians to maintain reconstruction quality. This counteracts the effect of higher resolution, resulting in only a negligible drop in Gaussian usage in the last row.

## 6. Conclusions

In this work, we present Long-LRM, a novel approach for fast and scalable 3D Gaussian reconstruction. By combining Mamba2 and transformer blocks, along with token merging and Gaussian pruning, Long-LRM achieves instant wide-span scene reconstruction from 32 images at a high resolution of 960  $\times$  540 in just 1 second, delivering rendering quality comparable to optimization-based 3D GS methods. Additionally, Long-LRM outperforms previous feed-forward GS methods in low-resolution, sparse-view setups and demonstrates compatibility with other GS variants, such as 2D GS. Through extensive ablation studies, we highlight the advantages of our hybrid architecture, the efficiency gains from token merging, and the effectiveness of the proposed regularization terms.

## Acknowledgements

We thank Nathan Carr and Kalyan Sunkavalli for their support and helpful discussions. We thank Lu Ling for the support in using the DL3DV dataset. Chen Ziwen and Li Fuxin are partially supported by Oregon State University seed grant AGD010-AS06 and NSF 2321851.

## References

- [1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 2, 5
- [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. 1, 2
- [3] Aljaz Bozic, Pablo Palafox, Justus Thies, Angela Dai, and Matthias Nießner. Transformerfusion: Monocular rgb scene reconstruction using transformers. *Advances in Neural Information Processing Systems*, 34:1403–1414, 2021. 2
- [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. 1, 2, 6, 7
- [5] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14124–14133, 2021. 2
- [6] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 2
- [7] Yuedong Chen, Haoifei 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, 2025. 1, 2, 6, 7
- [8] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. Deep stereo using adaptive thin volume representation with uncertainty awareness. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2524–2534, 2020. 2
- [9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 6, 12
- [10] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*, 2024. 5
- [11] Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024. 2, 3
- [12] Haoye Dong, Aviral Chharia, Wenbo Gou, Francisco Vicente Carrasco, and Fernando De la Torre. Hamba: Single-view 3d hand reconstruction with graph-guided bi-scanning mamba. *arXiv preprint arXiv:2407.09646*, 2024. 3
- [13] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 2, 3
- [14] Arda Duzceker, Silvano Galliani, Christoph Vogel, Pablo Speciale, Mihai Dusmanu, and Marc Pollefeys. Deepvideomvs: Multi-view stereo on video with recurrent spatio-temporal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15324–15333, 2021. 2
- [15] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 2, 3
- [16] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021. 3
- [17] 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. 1, 3
- [18] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024. 2, 5, 6
- [19] Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. Localmamba: Visual state space model with windowed selective scan. *arXiv preprint arXiv:2403.09338*, 2024. 3
- [20] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18365–18375, 2022. 2
- [21] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. *Advances in neural information processing systems*, 30, 2017. 2
- [22] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are mns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020. 3
- [23] 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. 1, 2, 5, 6, 7
- [24] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 2, 5, 7
- [25] 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. In

*The Twelfth International Conference on Learning Representations*, 2023. 1, 3

- [26] Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meir, Yonatan Belinkov, Shai Shalev-Shwartz, et al. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*, 2024. 3
- [27] Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Efficient neural radiance fields for interactive free-viewpoint video. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 2
- [28] Jiaqi Lin, Zhihao Li, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Jiayue Liu, Yangdi Lu, Xiaofei Wu, Songcen Xu, Youliang Yan, et al. Vastgaussian: Vast 3d gaussians for large scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5166–5175, 2024. 2
- [29] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. D13dv-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, 7
- [30] Tianqi Liu, Guangcong Wang, Shoukang Hu, Liao Shen, Xinyi Ye, Yuhang Zang, Zhiguo Cao, Wei Li, and Ziwei Liu. Fast generalizable gaussian splatting reconstruction from multi-view stereo. *arXiv preprint arXiv:2405.12218*, 2024. 1, 2, 6
- [31] 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*, pages 7824–7833, 2022. 2
- [32] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model, 2024. 3
- [33] Yang Liu, Chuanchen Luo, Lue Fan, Naiyan Wang, Junran Peng, and Zhaoxiang Zhang. Citygaussian: Real-time high-quality large-scale scene rendering with gaussians. In *European Conference on Computer Vision*, pages 265–282. Springer, 2025. 2
- [34] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5
- [35] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 7
- [36] 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. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2
- [37] 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. 1, 2
- [38] Zak Murez, Tarrence Van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 414–431. Springer, 2020. 2, 12
- [39] Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *arXiv preprint arXiv:2403.17898*, 2024. 2
- [40] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. Simplerecon: 3d reconstruction without 3d convolutions. In *European Conference on Computer Vision*, pages 1–19. Springer, 2022. 2
- [41] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part III 14*, pages 501–518. Springer, 2016. 2, 5, 12
- [42] Qihong Shen, Zike Wu, Xuanyu Yi, Pan Zhou, Hanwang Zhang, Shuicheng Yan, and Xinchao Wang. Gamba: Marry gaussian splatting with mamba for single view 3d reconstruction. *arXiv preprint arXiv:2403.18795*, 2024. 2, 3
- [43] Noah Stier, Alexander Rich, Pradeep Sen, and Tobias Höllerer. Vortex: Volumetric 3d reconstruction with transformers for voxelwise view selection and fusion. In *2021 International Conference on 3D Vision (3DV)*, pages 320–330. IEEE, 2021. 2, 12
- [44] 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
- [45] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15598–15607, 2021. 2
- [46] Stanislaw Szymonowicz, Christian Ruppert, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10208–10217, 2024. 1
- [47] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023. 2
- [48] Jiayang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2025. 1
- [49] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996. 4

- [50] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 1
- [51] Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norrick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, et al. An empirical study of mamba-based language models. *arXiv preprint arXiv:2406.07887*, 2024. 3
- [52] 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. In *The Twelfth International Conference on Learning Representations*, 2023. 1, 3
- [53] 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. 2
- [54] 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. 3
- [55] 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. 3
- [56] 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*, pages 5438–5448, 2022. 1, 2
- [57] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. In *The Twelfth International Conference on Learning Representations*, 2023. 3
- [58] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10371–10381, 2024. 4
- [59] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European conference on computer vision (ECCV)*, pages 767–783, 2018. 2
- [60] Xuanyu Yi, Zike Wu, Qihong Shen, Qingshan Xu, Pan Zhou, Joo-Hwee Lim, Shuicheng Yan, Xinchao Wang, and Hanwang Zhang. Mvgamba: Unify 3d content generation as state space sequence modeling. *arXiv preprint arXiv:2406.06367*, 2024. 2, 3
- [61] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4578–4587, 2021. 2
- [62] 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*, pages 19447–19456, 2024. 7
- [63] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17, 2016. 2
- [64] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pages 1–19. Springer, 2025. 1, 2, 3, 4, 6, 7
- [65] Xiaoshuai Zhang, Sai Bi, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Nerfusion: Fusing radiance fields for large-scale scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5449–5458, 2022. 2
- [66] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018. 6
- [67] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024. 3