iLRM: An Iterative Large 3D Reconstruction Model

Gyeongjin Kang[1] Seungtae Nam[2] Seungkwon Yang[2] Xiangyu Sun[1]
Sameh Khamis[3] Abdelrahman Mohamed[4*] Eunbyung Park[2]

[1]Sungkyunkwan University [2]Yonsei University [3]Rembrand [4]Meta

https://gynjn.github.io/iLRM/

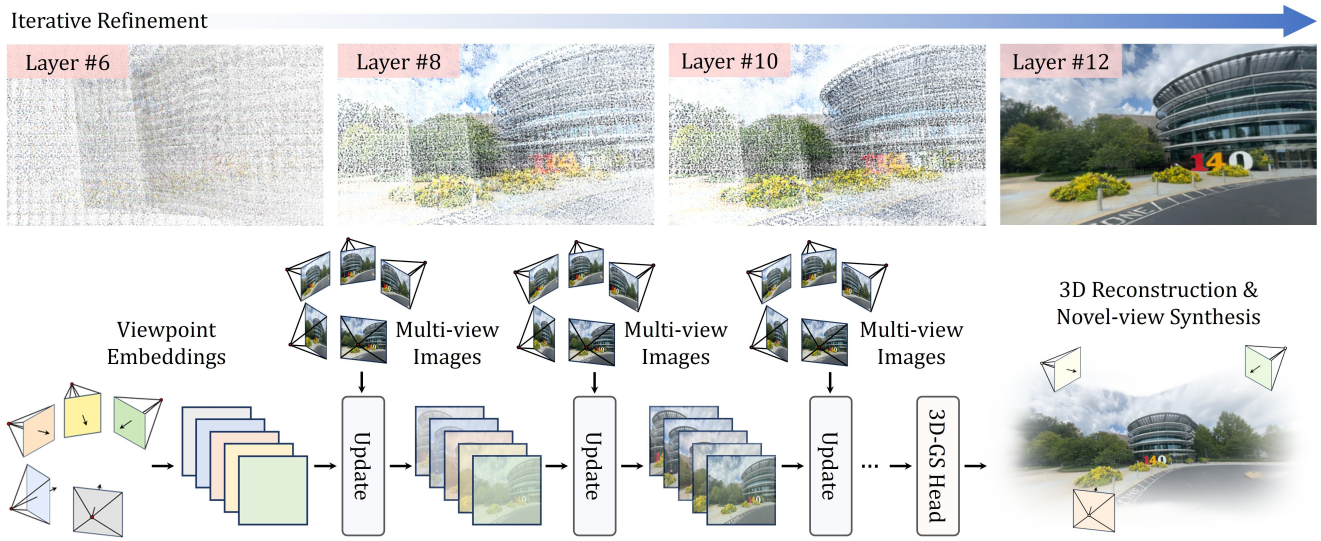


Figure 1. The overall architecture of the *iLRM*. As the layer index increases, compact viewpoint tokens are iteratively refined by attending to multi-view image tokens and are finally decoded into 3D Gaussian primitives, enabling efficient and high quality 3D reconstruction.

Abstract

Feed-forward 3D modeling has emerged as a promising approach for rapid and high-quality 3D reconstruction. In particular, directly generating explicit 3D representations, such as 3D Gaussian splatting, has attracted significant attention due to its fast and high-quality rendering. However, many state-of-the-art methods, primarily based on transformer architectures, suffer from severe scalability issues because they rely on full attention across image tokens from multiple input views, resulting in prohibitive computational costs as the number of views or image resolution increases. Toward a scalable and efficient feed-forward 3D reconstruction, we introduce an iterative Large 3D Reconstruction Model (iLRM) that generates 3D Gaussian representations through an iterative refinement mechanism, guided by three core principles: (1) decoupling the scene representation from input images to enable compact 3D representations; (2) decomposing global multi-view interactions into a two-stage attention scheme to reduce computational costs; and (3) injecting high-resolution information at every layer to achieve high-fidelity reconstruction. Experimental results on widely used datasets, such as RE10K and DL3DV, demonstrate that iLRM outperforms existing methods in both reconstruction quality and speed.

1. Introduction

Since the recent success of 3D Gaussian Splatting (3D-GS) [28], significant progress has been made in leveraging this 3D representation for building generalizable feed-forward 3D reconstruction models [4, 9, 10, 44, 51, 52, 56]. These methods typically train large neural networks to transform multi-view input images into feature representations, then regress Gaussian attributes. In contrast to per-scene 3D-GS optimization approaches [16, 28, 35, 36], these feed-forward models can reconstruct 3D scenes in a

*Work done while at Rembrand.

single forward pass, offering near real-time performance. Moreover, the prior knowledge learned from large-scale datasets [13, 14, 32, 59] allows them to effectively generalize to unseen scenes. While their reconstruction quality often lags behind that of per-scene optimization methods, fast reconstruction speed and generalization capability mark a promising step toward the long-standing goal of achieving accurate and real-time 3D scene reconstruction.

Among the promising approaches, pixel-aligned Gaussian models [4, 43, 58] have emerged as the de facto standard, leveraging decades of advances developed for numerous image-based tasks. While these models have proven effective, they also exhibit certain limitations. In particular, since they generate per-pixel Gaussians directly from the input images, the image resolution determines the number of Gaussians produced, which can lead to an excessive number of redundant Gaussians. For example, given input images at 1K resolution across 200 viewpoints (a scale comparable to the bicycle scene in the mip-NeRF 360 dataset [2]), these methods would produce 200 million Gaussians, despite prior studies [8, 15, 29, 30] demonstrating that such scenes can be efficiently represented with around 0.5 million Gaussians. To mitigate this issue, several techniques have been proposed, such as Gaussian regularization [60] and feature fusion [48]. Alternatively, the network can also be designed to generate fewer Gaussians, for example by downsampling the output resolutions. However, these strategies still require processing high-resolution images and therefore do not address another fundamental limitation of these models: high computational and memory demands.

A significant portion of computational and memory overhead arises from modeling interactions across multiple input views in feed-forward 3D reconstruction models. For instance, GS-LRM [56] performs full attention over all image tokens from every input view, leading to a quadratic increase in complexity with respect to both the number of views and image resolution. MVSplat [10] and DepthSplat [51] construct and process cost volumes for each view, further contributing to the computational demands. While one might attempt to alleviate this burden by reducing the input image resolution or using a sparser set of views, such strategies risk discarding essential geometric and appearance information required for accurate reconstruction.

Beyond the computational complexity and the inefficiency of the generated representations, we also question whether the prevailing formulation, casting 3D reconstruction as a sequence-to-sequence problem that maps entire sets of image tokens to dense, pixel-aligned Gaussians, is fundamentally well-suited to the nature of the task. While this formulation has achieved impressive results [25, 56, 60], it remains primarily a one-shot 3D scene generation process. In contrast, the recent optimization-based methods [28, 36] follow a fundamentally different strategy: they treat reconstruction as an iterative refinement process, where each iteration involves rendering the current scene estimate, measuring reconstruction error, and updating the representation accordingly. This loop enables the model to progressively capture finer geometric and appearance details while ensuring strong 3D consistency. The success of these methods suggests that high-quality 3D reconstruction may benefit not only from expressive representations but also from feedback-driven iterative refinement, a trait largely absent in existing feed-forward 3D models.

In this paper, we introduce *iLRM*[*], an iterative large 3D reconstruction model that effectively 1) incorporates the principles of feedback-driven refinement, while also 2) addressing the computational burden and representational inefficiencies inherent in existing feed-forward approaches. As illustrated in Fig. 1, the network (acting as an optimizer) transforms the embedding features (analogous to updating the 3D-GS representation) at each layer (analogous to each optimization step), based on multi-view image tokens (serving as gradient-like signals). This design allows the model to iteratively update the scene representation at every layer based on feedback from the multi-view input images, effectively mimicking the optimization process within a feed-forward architecture. Through this process, the learned neural network jointly examines the input view images and the evolving scene representation to identify where and how to make targeted updates that improve reconstruction quality.

Another core design principle of our approach is to decouple the representation, later transformed into 3D Gaussians, from direct dependence on input images, addressing the computational complexity and redundancy that arise in architectures that generate pixel-aligned Gaussians directly from multi-view inputs. By decoupling the representation and input images, we can use low-resolution representations to produce a compact set of Gaussians while still leveraging high-resolution input images for detailed guidance.

In addition, we propose an efficient mechanism for modeling the interaction between the representations and the input images. A naïve approach would involve computing full attention between all tokens across views, which quickly becomes computationally prohibitive. To overcome this, we initialize our representation using viewpoint embeddings, each tied to a specific input view. Interaction modeling is then split into two stages. First, we perform cross-attention between each viewpoint embedding and its corresponding image, which is highly efficient due to the one-to-one mapping. Next, we apply self-attention across all viewpoint embeddings. Importantly, since this second stage operates over a low-resolution representation space, it remains computa-

[*]By "feedback-driven," we mean that at every update layer the current scene tokens are explicitly revised via cross-attention with each image tokens. We call this "iterative" because the scene representation is repeatedly updated layer by layer under static (unchanged) image evidence, rather than merely transformed by stacked self-attention blocks [23].

tionally tractable while facilitating global information exchange across views. Overall, this scalable design significantly reduces computational and memory overhead and allows for the incorporation of more viewpoints, thereby improving reconstruction fidelity.

We comprehensively evaluate the proposed method on large-scale datasets, RealEstate10K [59] and DL3DV [32]. The experimental results demonstrate that *iLRM* achieves superior rendering quality while substantially reducing both computational and memory overhead compared to recently proposed feed-forward Gaussian models. Moreover, in high-resolution and wide-coverage settings (**540×960, 32 views**), our method completes inference in **0.5 seconds**, achieving comparable performance to an optimization-based approach, which takes about **8 minutes**.

2. Related Works

2.1. Feed-forward 3D Gaussian Splatting

Feed-forward 3D Gaussian Splatting [4, 10, 37, 43, 44, 51, 52, 56] capitalizes on robust priors learned from extensive datasets to estimate Gaussian primitive parameters and synthesize novel view images using sparse input data. PixelSplat [4] and LatentSplat [49] predict Gaussians from image features using an epipolar line sampling method to enhance geometric accuracy, while MVSplat [10] and MVSGaussian [34] construct cost volumes through a plane-sweep stereo approach. In a further development, Flash3D [42] and DepthSplat [51] introduce a pre-trained depth estimation model [39, 53], which improves the robustness of the spatial positions of 3D Gaussians. In contrast, GS-LRM [56] and Long-LRM [60] minimize reliance on explicit 3D priors by leveraging large-scale data-driven priors.

While demonstrating strong results, a major limitation of all the aforementioned approaches lies in their non-scalable architectural design, which restricts their ability to effectively leverage a large number of input views. Moreover, the one-shot generation strategy, which produces 3D representations in a single forward pass, often fails to capture complex geometric details and fine 3D consistency, making them suboptimal for high-quality 3D reconstruction. We address these limitations by proposing an iterative 3D reconstruction framework and scalable architectural designs.

Iterative refinements. Our work is also closely related to recent methods that adopt iterative refinement strategies, such as G3R [7] and Gen-Den [37]. Both utilize actual gradients to update their representations more precisely. While promising, these approaches require additional computational burden for rendering multiple images per training iteration, and relying solely on gradients may risk overlooking valuable information present in the raw input images. Nonetheless, exploring how to incorporate gradient information remains an interesting direction for future work.

2.2. 3D Representations from Embeddings

Inspired by previous generative approaches [3, 18, 27], recent works [5, 17, 22] have investigated the synthesis of 3D representations directly from learnable embeddings, guided by input image supervision. This paradigm leverages the expressive capacity of latent spaces to encode rich geometric priors, which act as structural templates that guide the reconstruction process. Such approaches offer notable flexibility, allowing rendering from arbitrary viewpoints and adaptation to varying space scales and camera poses. However, both LRM [22] and Lara [5] are limited to object-centric representations, restricting scalability to complex scenes involving multiple objects or large spatial layouts. The recently proposed Quark [17] also utilizes learnable embeddings to fuse visual cues from multiple images, demonstrating compelling results, but its representation is confined to the target view [25, 34, 50], lacking an explicit and persistent 3D reconstruction.

In contrast to previous works, we construct scene-level explicit 3D representations from viewpoint embeddings by decoupling the generation of Gaussians from the input images. This separation enables iterative refinement of the embeddings using low-level visual features and provides flexible control over the density of the 3D representation, independent of the input image resolution.

3. Method

3.1. Motivation and Problem Statement

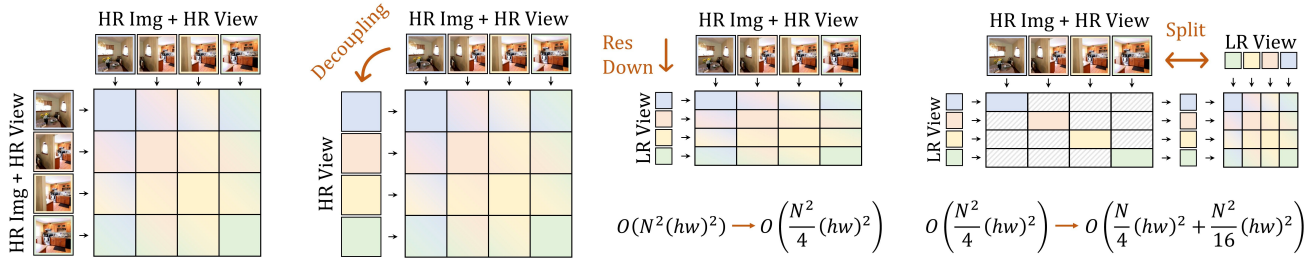
Existing generalizable 3D Gaussian reconstruction methods process multi-view images in an end-to-end fashion, often employing techniques such as epipolar line sampling [4], plane-sweep stereo [9, 10, 51], or full-resolution attention [52, 56, 60] to enforce multi-view consistency. While effective, these strategies introduce significant computational and memory overhead, limiting their scalability.

To address these challenges, we propose *iLRM*, a novel feed-forward 3D reconstruction framework that decouples Gaussian generation from direct dependence on input images. Instead of generating pixel-aligned Gaussians, *iLRM* initializes viewpoint-centric embeddings as the basis for constructing the 3D scene. These embeddings are then iteratively refined via cross-attention with multi-view image features, enabling the model to efficiently fuse geometric and appearance cues across views.

We start with N multi-view images $\{I_i\}_{i=1}^N$ and camera poses $\{C_i\}_{i=1}^N$. Based on this setup, our goal is to train a model f_θ that maps a set of viewpoints to 3D Gaussians, leveraging the associated multi-view images as visual cues throughout the reconstruction pipeline. More formally,

$$f_\theta : \{(C_i, I_i)\}_{i=1}^N \mapsto \{(\mu_k, \alpha_k, \Sigma_k, c_k)\}_{k=1}^{H^v W^v N}, \quad (1)$$

where f_θ is modeled as a feed-forward network with the model parameter θ. $\mu_k, \alpha_k, \Sigma_k, c_k$ are attributes of 3D



(a) HR Img & HR View (GS-LRM) (b) Decoupled HR Img & HR View (c) Decoupled HR Img & LR View (d) Cross-attn + Self-attn

Figure 2. The proposed scalable architectural designs by decoupling viewpoint and image tokens, and modeling the global interactions via cross- and self-attentions (N: # views, $h = H/p, w = W/p$).

Gaussians, representing the mean, opacity, covariance, and color, respectively, while H^v and W^v denote the height and width of the generated Gaussians for each camera viewpoint. It is important to note that they do not correspond to the resolution of the input images. We train our model using held-out target images along with their corresponding camera poses, enabling high-quality novel view synthesis.

3.2. Architectural Design

We propose an end-to-end transformer that directly regresses 3D Gaussian parameters from viewpoint embeddings. To compensate for the absence of direct image input, we enrich these embeddings at each layer via cross-attention with multi-view image features. The resulting embeddings are further refined through self-attention to capture global dependencies across viewpoints.

Viewpoint tokenization. Following previous works [25, 44, 56], we employ a Plücker ray embedding for each input view using the camera poses. Specifically, given the intrinsic, rotation, and translation, we construct the Plücker ray embeddings for each viewpoint. We then divide these viewpoint embeddings into non-overlapping patches of size $p \times p$, and reshape each patch into a 1D vector, resulting in a tensor of shape $H^v W^v / p^2 \times 6p^2$. Then, we encode it using a single linear layer to produce viewpoint tokens, $V_i^{(0)} \in \mathbb{R}^{H^v W^v / p^2 \times d}$. Plücker coordinates inherently capture spatial variations across pixels and views, allowing them to effectively differentiate between patches. As a result, we do not utilize additional positional embeddings.

Multi-view image tokenization. For each input view image, which provides visual guidance to the reconstruction process, we extract both image features and corresponding pose information. Specifically, we divide an input image into non-overlapping patches and obtain two sets: RGB image patches and Plücker ray patches. These are then concatenated and linearly projected to construct the image patch tokens, $S_i \in \mathbb{R}^{HW/p^2 \times d}$,

$$S_{ij} = \text{Linear}(\text{concat}(I_{ij}, P_{ij})) \in \mathbb{R}^d, \tag{2}$$

where $I_{ij} \in \mathbb{R}^{3p^2}, P_{ij} \in \mathbb{R}^{6p^2}$ represent the flattened j-th image and ray patches for the i-th view, respectively, and HW/p^2 is the number of tokens for each input view image.

Scalable multi-view context modeling. Fig. 2-(a) shows the typical feed-forward 3D methods [10, 51, 56] using transformer architecture, which perform full attention across multi-view images, incurring a quadratic increase in computational cost with respect to both the number of views and the image resolution. Fig. 2-(b) depicts our decoupling approach. Thanks to the decoupling technique, we can reduce the viewpoint resolution while still leveraging high-resolution multi-view images (Fig. 2-(c)). We further decrease the computation cost by two-stage multi-view context modeling, per-view cross-attention and viewpoint self-attention (Fig. 2-(d)). For example, given 16 input images with a resolution of 256 $\times$ 256 and a patch size of 8, the relative computational cost follows the ratio (1:1:0.25:0.08, Fig. 2-(a):(b):(c):(d)), highlighting that our proposed method can accommodate more input views with significantly less computational burden.

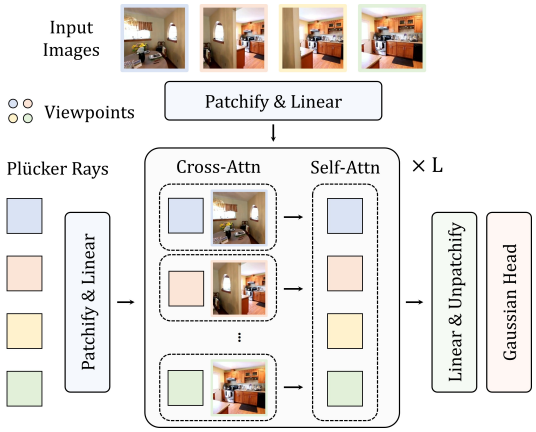
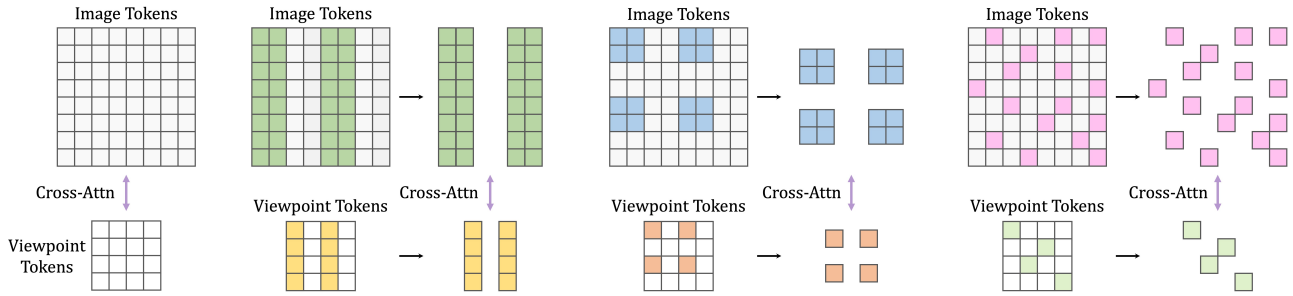


Figure 3. Network architecture.

Update block. Given a set of viewpoint tokens, we formulate the problem as an iterative refinement process, where the viewpoint tokens are progressively updated through in-



(a) Full Cross-Attn (b) Half Cross-Attn (c) Quarter Cross-Attn (d) Random Quarter Cross-Attn

Figure 4. Various mini-batch cross-attention schemes. We primarily adopt "Quarter Cross-attention" in our experiments.

teractions with multi-view image tokens, ultimately leading to more accurate and spatially consistent 3D Gaussian Splatting. As shown in Fig. 3, our model consists of multiple transformer modules, each comprising one cross-attention layer followed by one self-attention layer.

$$\tilde{V}_i^{(l-1)} = \text{cross-attn}^{(l)}(V_i^{(l-1)}, S_i), \quad (3)$$

$$\{V_i^{(l)}\}_{i=1}^N = \text{self-attn}^{(l)}(\{\tilde{V}_i^{(l-1)}\}_{i=1}^N), \quad (4)$$

where the superscript (l) denotes the layer index. In the cross-attention layers, the viewpoint tokens are refined by the visual information from their corresponding image tokens. In the self-attention layers, the viewpoint tokens interact with each other to enhance their representations using global contextual information. Note that we use separate model parameters for the update blocks at different layers.

Token uplifting. Standard cross-attention is typically applied between token sets of similar spatial resolutions. In our setting, however, we intentionally use lower-resolution (LR) viewpoint tokens compared to HR image tokens to improve scalability and efficiency, which may limit their ability to fully capture rich visual information. To bridge this gap, we propose a token uplifting strategy. Each LR viewpoint token is lifted by a linear query layer that expands its feature dimension by a factor of k, yielding a tensor of shape $H^v W^v/p^2 \times dk$, which is then reshaped to $H^v W^v k/p^2 \times d$ so that each original token corresponds to k finer-grained query tokens for better visual correspondence during cross-attention. After cross-attention with HR image tokens as keys and values, the resulting tensor is reshaped back to $H^v W^v/p^2 \times dk$ and projected to the original dimension d via a linear layer, yielding refined viewpoint tokens of shape $H^v W^v/p^2 \times d$. We set $k = 2$ to balance representational capacity and efficiency.

Mini-batch cross-attention. In our architecture, viewpoint tokens are iteratively updated at each layer based on the image tokens via cross-attention. The proposed decoupling design allows us to arbitrarily reduce the number of viewpoint tokens for improved scalability, whereas the resolution of image tokens remains fixed due to their spatial nature. Consequently, the primary computational bottleneck in cross-attention lies in the high-resolution image tokens.

To address this, we propose several efficient cross-attention schemes, as illustrated in Fig. 4, aimed at improving scalability without sacrificing performance. Our design is conceptually inspired by mini-batch gradient descent in optimization, where only a subset of data points is sampled in each iteration to reduce computational cost. Similarly, our mechanism selectively samples subsets of both image tokens and viewpoint tokens during cross-attention. While random token sampling (Fig. 4-(d)) is ideal in theory, it complicates efficient implementation. To mitigate this, we design structured sampling strategies that are simple to implement and demonstrate strong empirical performance.

Decoding into 3D Gaussians. After the final self-attention layer, we decode the final layer's viewpoint tokens, $V_i^{(L)}$, into Gaussian parameters through a single linear layer and apply post-activation functions. For a detailed description, please refer to our supplementary materials.

Interpretation. Compared to standard self-attention, $S^{(l)} = S^{(l-1)} + f(S^{(l-1)})$, our method applies evidence-conditioned updates, $V^{(l)} = V^{(l-1)} + F(V^{(l-1)}, S)$, where the image tokens S are fixed and provide detailed visual guidance. This resembles a gradient descent iteration, $V^{(l)} \approx V^{(l-1)} - \eta \nabla_V \mathcal{E}(V^{(l-1)}; S)$, where $\mathcal{E}$ is an implicit objective function, making each layer a feedback correction step rather than a pure feature transformation. Our mini-batch variant further extends this view as $V_{\text{mb}}^{(l)} \approx V_{\text{mb}}^{(l-1)} - \eta \nabla_{V_{\text{mb}}} \mathcal{E}(V_{\text{mb}}^{(l-1)}; S_{\text{mb}})$, where V_{mb} and S_{mb} denote a subset of viewpoint tokens and their corresponding image tokens, respectively.

3.3. Training Objectives

We rasterize 3D Gaussians from viewpoint tokens to obtain rendered images $\hat{I}_t$, supervised against ground-truth images I_t via MSE and perceptual loss [6, 31]:

$$\mathcal{L}_{\text{total}} = \sum_{t \in \mathcal{T}} \mathcal{L}_{\text{MSE}}(\hat{I}_t, I_t) + \lambda \mathcal{L}_{\text{perceptual}}(\hat{I}_t, I_t), \quad (5)$$

where $\mathcal{T}$ is the set of target view indices and λ=0.5 balances the two loss terms.

4. Experiments

Datasets. We train our model on two large-scale datasets: RealEstate10K (RE10K) [59] and DL3DV [32], and evaluate it on three datasets, including ACID [33]. We adopt the RE10K split following [4] and the official split for DL3DV. We use an image resolution of 256×256 for the RE10K and ACID datasets, while for the DL3DV dataset, we use a resolution of 256×448 and 512×960. In addition, we employ the undistorted version of the DL3DV dataset at a resolution of 540×960, which originates from Long-LRM [60].

Implementation details. Our model consists of 12 update layers, each containing one cross-attention and one self-attention block. Inside each attention module, we adopt a pre-normalization method with LayerNorm [1] and QK-Norm [20] method with an RMSNorm [55] layer. Also, each block utilizes 12-head attention [45] and two GELU [19]-activated linear layers. We set the hidden dimension to $d = 768$, and use a patch size of $p = 8$. For more details, please refer to the supplementary material.

Evaluation. We compare our model against recent generalizable 3D reconstruction methods [4, 10, 37, 51, 56, 60] as well as optimization-based approach [28]. For evaluation, we follow the settings from [4, 54] for RE10K and [51, 60] for DL3DV. We denote our various viewpoint settings as $(V, H/F, F)$, where V is the number of viewpoints, and the following entries indicate the resolutions of viewpoint and image tokens (H: half-resolution, F: full-resolution). For example, a setting of $(2, H, F)$ indicates two viewpoints tokens with half-resolution and full-resolution image tokens. MC refers to our quarter mini-batch cross-attention (Fig. 4-(c)). Note that our 2-view full-resolution setting $(2, F, F)$ does not include token uplifting, as resolutions are identical. Additionally, when using more views than is required in the evaluation protocol, we sample extra views, ensuring there is no overlap with the target indices.

Method	#Param (M)	PSNR ↑	SSIM ↑	LPIPS ↓	# Gaussians	Time (s)
pixelSplat	125	25.89	0.858	0.142	131,072	0.101
MVSplat	12	26.39	0.869	0.128	131,072	0.047
GS-LRM*	300	28.10	0.892	0.114	131,072	-
DepthSplat	354	27.47	0.889	0.114	131,072	0.065
Gen-Den	28	27.08	0.879	0.120	347,072	0.224
Ours $(2, F, F)$	171	**28.65**	**0.900**	0.110	131,072	**0.025**
Ours $(4, H, F)$	185	30.37	0.923	0.095	65,536	0.027
Ours-MC $(4, H, F)$	185	30.10	0.919	0.098	65,536	0.027
Ours $(8, H, F)$	185	31.57	0.935	0.082	131,072	0.028
Ours-MC $(8, H, F)$	185	31.24	0.933	0.084	131,072	0.029

Table 1. Quantitative comparisons on the RE10K dataset with various view configurations. Inference time is measured on a RTX 4090 GPU. * indicates closed-source methods. The time difference in the MC variant is negligible due to the short sequence length in inference. For a more comprehensive analysis of our mini-batch cross-attention, see Tab. 8.

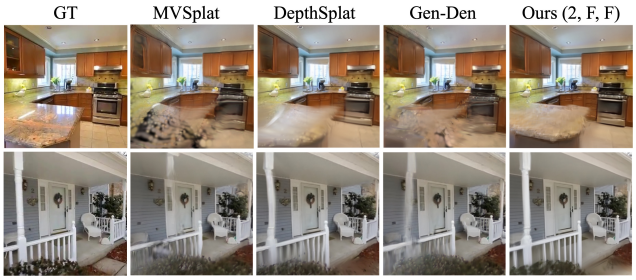


Figure 5. Qualitative comparison on the RE10K dataset.

Method	ACID			DL3DV		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
MVSplat	28.15	0.841	0.147	22.65	0.737	0.191
DepthSplat	28.37	0.847	**0.141**	24.28	0.813	0.147
Gen-Den	28.61	0.847	**0.141**	22.92	0.750	0.188
Ours $(2, F, F)$	**29.24**	**0.856**	0.143	**25.35**	**0.826**	**0.144**
Ours $(4, H, F)$	30.10	0.877	0.138	27.90	0.877	0.122
Ours-MC $(4, H, F)$	29.90	0.873	0.141	27.68	0.881	0.127
Ours $(8, H, F)$	30.96	0.894	0.122	29.56	0.907	0.101
Ours-MC $(8, H, F)$	30.72	0.890	0.125	29.33	0.904	0.102

Table 2. Cross-dataset generalization on the ACID and DL3DV (256×256) using a model trained on the RE10K dataset.

Method	Views	PSNR ↑	SSIM ↑	LPIPS ↓	# Gaussians	Time (s)	Memory (GB)
MVSplat	6	22.93	0.775	0.193	688,128	0.279	5.87
DepthSplat	6	24.19	0.823	**0.147**	688,128	0.102	3.55
	11	24.28	0.833	0.141	1,261,568	0.170	6.01
	24	22.37	0.781	0.195	2,752,512	0.371	12.39
Ours	$(6, H, F)$	**25.60**	**0.830**	0.168	**172,032**	**0.031**	**1.40**
	$(11, H, F)$	**26.99**	**0.865**	**0.140**	**315,392**	**0.051**	**1.59**
	$(24, H, F)$	**27.38**	**0.882**	**0.126**	**688,128**	**0.123**	**2.01**

Table 3. Quantitative comparisons on the DL3DV dataset under the 50-frame baseline setting (256×448). Inference time and memory consumption are measured on a RTX 4090 GPU.

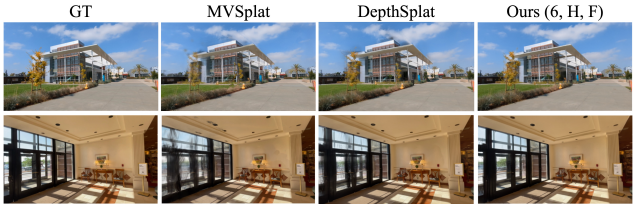


Figure 6. Qualitative comparison on DL3DV dataset (256×448).

4.1. Results

In Tab. 1, 2 and Fig. 5, we compare our approach with feed-forward methods on the RE10K dataset and cross-dataset generalization on ACID and DL3DV. Furthermore, we report results with an increased number of input views (4 and 8), which incur less than half of the computation time compared to the baseline (DepthSplat) while achieving superior performance. For the DL3DV dataset, our method consistently outperforms the baselines across various viewpoint and resolution configurations, including inference speed and memory usage, while achieving efficient scene representation with fewer Gaussians, as shown in Tab. 3, 4 and Fig. 6. While DepthSplat and our method are both trained under varying numbers of input views, our

Figure 7. Qualitative comparison on undistorted DL3DV dataset under the wide-baseline setting (32 input images, 540×960, zero-shot).

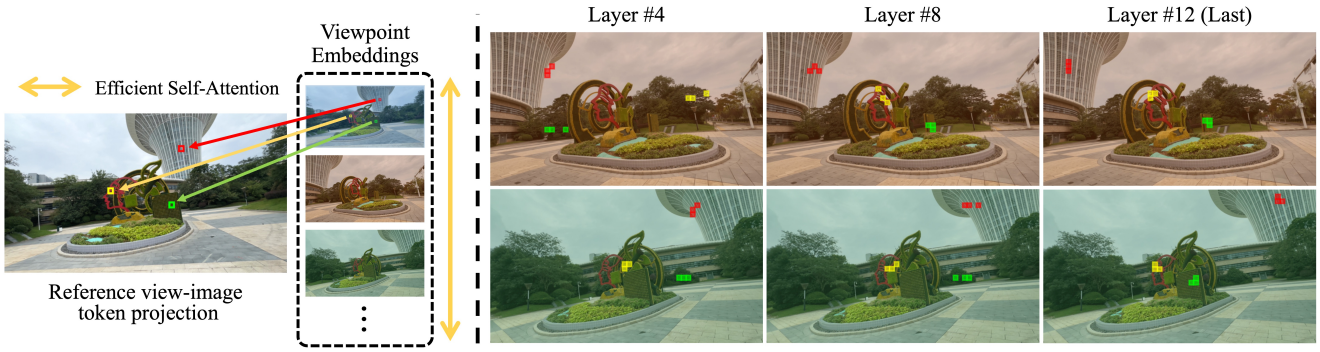


Figure 8. For the colored query patches in the reference viewpoint (red, yellow, green), we visualize top-3 attended tokens from other viewpoints throughout the iterative refinement process.

approach demonstrates enhanced scalability with respect to the increasing number of views in Tab. 3.

Method	Views	PSNR ↑	SSIM ↑	LPIPS ↓	# Gaussians	Time (s)	Memory (GB)
DepthSplat	12	21.38	0.739	0.265	5,898,240	-	OOM
Ours	$(12, H, F)$	**24.35**	**0.781**	**0.256**	**1,474,560**	**0.415**	**3.53**

Table 4. Quantitative comparisons on the DL3DV dataset under the 100-frame baseline setting (512×960). Inference time and memory consumption are measured on a single RTX 4090 GPU. Since DepthSplat encounters out-of-memory (OOM) issue on the device, we evaluate its performance using a H100 GPU.

Method	Views	Time ↓	PSNR ↑	SSIM ↑	LPIPS ↓
3D-GS	16	8min	21.48	0.753	0.252
Long-LRM	16	0.50sec	22.66	0.740	**0.292**
Ours	$(16, H, F)$	**0.19sec**	**22.91**	**0.766**	0.295
3D-GS	32	8min	24.43	0.827	0.191
Long-LRM	32	0.84sec	23.97	0.778	0.267
Ours	$(32, H, F)$	**0.53sec**	**24.30**	**0.803**	**0.256**
Long-LRM$_{10}$	32	11sec	25.56	0.826	0.237
Ours$_{10}$	$(32, H, F)$	**4.5sec**	**25.67**	**0.844**	**0.230**
Long-LRM (Unseen)	40	1.05sec	24.18	0.787	0.260
Ours (Unseen)	$(40, H, F)$	**0.76sec**	**24.54**	**0.811**	**0.248**
Long-LRM (Unseen)	48	1.38sec	24.30	0.797	0.252
Ours (Unseen)	$(48, H, F)$	**1.04sec**	**24.78**	**0.820**	**0.240**

Table 5. Quantitative comparisons on the undistorted DL3DV dataset (540×960). We utilized flash attention v3 [41] using a H100 GPU. We re-evaluate Long-LRM [60] with their official checkpoint except for 16-view metrics (16-view weights are not released).

In the wide-coverage setting (Tab. 5), we evaluate performance using various numbers of high-resolution input images under full-frame coverage. For comparison, we also include optimization-based 3D-GS [28] trained for 30k iterations using the input images and camera poses. Long-LRM$_{10}$ means finetuning 10 epochs initialized from the Long-LRM's generated Gaussians. Since our approach produces more compact 3D Gaussian representations (4× fewer), the finetuning process is significantly faster than the baseline. We further evaluate longer-context generalization ability (40 and 48 views) using a model trained with 32 views. Our method achieves better performance and faster inference across all metrics and scales more favorably with the number of views while maintaining compact scenes.

4.2. Attention Visualization

We investigate how our method achieves global consistency throughout the iterative refinement process. Using the first input view as the reference, we select three query patches from its viewpoint embedding, and visualize top-3 attended tokens in the other viewpoint embeddings. For ease of visualization, we project the selected tokens onto the corresponding images via spatial upsampling. As shown in Fig. 8, attended tokens from other viewpoints gradually shifts toward geometrically and semantically corresponding regions as the layers go deeper, demonstrating the progressive, iterative refinement of the multi-view scene representation, which aligns our proposed design motivation.

	# Params	PSNR ↑	SSIM ↑	LPIPS ↓
12 layers (base)	185M	**29.24**	**0.907**	**0.109**
9 layers	139M	29.01	0.903	0.112
6 layers	94M	28.68	0.898	0.116
3 layers	48M	28.04	0.887	0.126

Table 6. Ablations on model size.

	PSNR ↑	SSIM ↑	LPIPS ↓
Baseline (12 layers)	**29.24**	**0.907**	**0.109**
w/o iter. refinement	28.58	0.893	0.127
w/o resolution decoupling	28.47	0.891	0.123
w/o token uplifting	28.90	0.901	0.113

Table 7. Ablations on model architecture.

4.3. Computational Costs of Training

We report detailed comparisons of computational costs during training in Tab. 8. The iteration time is measured under the same setting: half-resolution 8 viewpoints $(8, H, F)$, and a batch size of 16 on a single RTX 4090 GPU. For memory comparison, to provide a clearer analysis, all models are run without gradient checkpointing on a single H100 GPU. Lastly, we present a theoretical comparison of FLOPs that further underscores the efficiency of our method, with only a marginal drop in performance. For detailed calculations of FLOPs, please refer to our supplementary material.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Iteration (s)	Memory (GB)	GFLOPs
Baseline	**30.39**	**0.923**	**0.095**	1.51	62.5	3.83
w/ Half Cross-attn	30.25	0.922	0.096	1.13	47.4	1.71
w/ Quarter Cross-attn	30.08	0.919	0.098	**0.94**	**39.0**	**0.81**

Table 8. Quantitative comparison of our mini-batch cross-attention on the RE10K dataset, with iteration time and memory consumption measured during training.

4.4. Ablations and Analysis

Tab. 6 presents the ablations on the number of update layers. All variants are trained under half-resolution 4 viewpoints setting $(4, H, F)$, on the RE10K dataset with batch size 16. The results demonstrate consistent performance gains as the number of layers increases. From the perspective of the iterative refinement procedure, increasing the number of layers can be interpreted as introducing more optimization steps, which aligns with our intuition that deeper refinement leads to more accurate 3D representations.

In Tab. 7, we report the ablation results on architectural components. All experiments follow the model-size ablation training setup with a 12-layer baseline. More extensive ablation studies are provided in the supplementary material.

1) Iterative refinement. The cross-attention blocks in our model keep providing visual evidence (image) into the viewpoint tokens as part of the iterative refinement process. We validate this by replacing per-layer cross-attention with a single cross-attention in the first layer: the baseline has 12 layers (each with cross- then self-attention), while the variant has 1 cross-attention followed by 23 self-attention layers. The result shows that our consecutive cross-attention with image features plays a critical role in refining the viewpoint embeddings especially in terms of the LPIPS metric.

2) Resolution decoupling. Our design decouples image resolution from the viewpoint representation, so cross-attention consumes high-resolution image features while the scene tokens remain lightweight. When image features are constrained to the viewpoint resolution (prior approaches [56, 60]), performance drops, indicating that resolution decoupling is essential for simultaneously preserving compactness and high-fidelity reconstructions.

3) Token uplifting. Removing the token uplifting mechanism leads to a drop in performance across all metrics compared to baseline. This validates the importance of expanding low-resolution view tokens before cross-attention with high-resolution image tokens. Without this step, the model struggles to capture fine-grained spatial correspondences, resulting in a degraded reconstruction quality.

5. Limitations

One limitation of this work is the self-attention bottleneck across many input views. While our compact viewpoint embeddings substantially reduce the computational cost, challenges may arise as the number of input views increases considerably. In this study, aiming for scalable feed-forward 3D models, we present the first implementation of the framework that iteratively refines 3D representations by leveraging high-resolution image information at every layer. Further development of more scalable alternatives [11, 12] would be a valuable direction for future research.

A second limitation is the reliance on accurate camera poses [38, 40] in static scenarios. Furthermore, because our primary goal is high fidelity novel view synthesis with rendering supervision, the recovered geometry may be less accurate than explicit geometry-supervised methods [46, 47]. Even so, our scalable and flexible structure provides a natural basis for relaxing these assumptions, as joint pose refinement [57] or even pose-free variants [21, 24, 26, 54] could be realized by training on suitable datasets and supervisions without modifying the core architectural design.

6. Conclusion

In this work, we present an iterative Large 3D Reconstruction Model (*iLRM*), a feed-forward architecture that reflects per-scene optimization-based schemes, by stacking multiple update layers composed of cross- and self-attention modules. By decoupling Gaussian representations from input images and splitting the update mechanism into per-view interactions with image features and global aggregation over compact viewpoint embeddings, *iLRM* enables efficient, scalable, and high-quality 3D reconstruction across diverse scenes. We believe that *iLRM* lays a strong foundation for future research in feed-forward 3D reconstruction.

Acknowledgements

This work was supported by Samsung Research Funding & Incubation Center of Samsung Electronics under Project Number SRFC-IT2401-01, the Artificial Intelligence Industrial Convergence Cluster Development Project funded by the Ministry of Science and ICT (MSIT, Korea) & Gwangju Metropolitan City, and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) under the following projects: (No. RS-2024-00457882, AI Research Hub Project); (No. RS-2025-25441838, Development of a human foundation model for human-centric universal artificial intelligence and training of personnel); (No. RS-2020-II201361, Artificial Intelligence Graduate School Program (Yonsei University)); and (No. RS-2025-02653113, High-Performance Research AI Computing Infrastructure Support at the 2 PFLOPS Scale).

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 6

[2] 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

[3] 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*, pages 16123–16133, 2022. 3

[4] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 1, 2, 3, 6

[5] Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger. Lara: Efficient large-baseline radiance fields. In *European Conference on Computer Vision*, pages 338–355. Springer, 2024. 3

[6] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1511–1520, 2017. 5

[7] Yun Chen, Jingkang Wang, Ze Yang, Sivabalan Manivasagam, and Raquel Urtasun. G3r: Gradient guided generalizable reconstruction. In *European Conference on Computer Vision*, pages 305–323. Springer, 2024. 3

[8] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Hac: Hash-grid assisted context for 3d gaussian splatting compression. In *European Conference on Computer Vision*, pages 422–438. Springer, 2024. 2

[9] Yuedong Chen, Chuanxia Zheng, Haofei Xu, Bohan Zhuang, Andrea Vedaldi, Tat-Jen Cham, and Jianfei Cai. Mvsplat360: Feed-forward 360 scene synthesis from sparse views. *arXiv preprint arXiv:2411.04924*, 2024. 1, 3

[10] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, 2025. 1, 2, 3, 4, 6

[11] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 8

[12] Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *International Conference on Machine Learning*, 2024. 8

[13] 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*, 2023. 2

[14] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 2024. 2

[15] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, Zhangyang Wang, et al. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *Advances in neural information processing systems*, 37: 140138–140158, 2024. 2

[16] Guangchi Fang and Bing Wang. Mini-splatting: Representing scenes with a constrained number of gaussians. In *European Conference on Computer Vision*, 2024. 1

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

[18] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 3

[19] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 6

[20] Alex Henry, Prudhvi Raj Dachapally, Shubham Pawar, and Yuxuan Chen. Query-key normalization for transformers. *arXiv preprint arXiv:2010.04245*, 2020. 6

[21] Sunghwan Hong, Jaewoo Jung, Heeseong Shin, Jisang Han, Jiaolong Yang, Chong Luo, and Seungryong Kim. Pf3plat: Pose-free feed-forward 3d gaussian splatting. *arXiv preprint arXiv:2410.22128*, 2024. 8

[22] 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. *arXiv preprint arXiv:2311.04400*, 2023. 3

[23] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021. 2

[24] Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, and Georgios Pavlakos. Rayzer: A self-supervised large view synthesis model. *arXiv preprint arXiv:2505.00702*, 2025. 8

[25] Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. *arXiv preprint arXiv:2410.17242*, 2024. 2, 3, 4

[26] Gyeongjin Kang, Jisang Yoo, Jihyeon Park, Seungtae Nam, Hyeonsoo Im, Sangheon Shin, Sangpil Kim, and Eunbyung Park. Selfsplat: Pose-free and 3d prior-free generalizable 3d gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22012–22022, 2025. 8

[27] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 3

[28] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 2023. 1, 2, 6, 7

[29] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21719–21728, 2024. 2

[30] Joo Chan Lee, Jong Hwan Ko, and Eunbyung Park. Optimized minimal 3d gaussian splatting. *arXiv preprint arXiv:2503.16924*, 2025. 2

[31] Zhengqi Li, Wenqi Xian, Abe Davis, and Noah Snavely. Crowdsampling the plenoptic function. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 178–196. Springer, 2020. 5

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

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

[34] Tianqi Liu, Guangcong Wang, Shoukang Hu, Liao Shen, Xinyi Ye, Yuhang Zang, Zhiguo Cao, Wei Li, and Ziwei Liu. Mvsgaussian: Fast generalizable gaussian splatting reconstruction from multi-view stereo. In *European Conference on Computer Vision*, pages 37–53. Springer, 2024. 3

[35] Tao Lu, Ankit Dhiman, R Srinath, Emre Arslan, Angela Xing, Yuanbo Xiangli, R Venkatesh Babu, and Srinath Sridhar. Turbo-gs: Accelerating 3d gaussian fitting for high-quality radiance fields. *arXiv preprint arXiv:2412.13547*, 2024. 1

[36] Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De La Torre. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*, 2024. 1, 2

[37] Seungtae Nam, Xiangyu Sun, Gyeongjin Kang, Younggeun Lee, Seungjun Oh, and Eunbyung Park. Generative densification: Learning to densify gaussians for high-fidelity generalizable 3d reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26683–26693, 2025. 3, 6

[38] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L Schönberger. Global structure-from-motion revisited. In *European Conference on Computer Vision*, pages 58–77. Springer, 2024. 8

[39] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10106–10116, 2024. 3

[40] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 8

[41] Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *Advances in Neural Information Processing Systems*, 37: 68658–68685, 2024. 7

[42] Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, Joao F Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. *arXiv preprint arXiv:2406.04343*, 2024. 3

[43] Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2, 3

[44] 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. In *European Conference on Computer Vision*, 2025. 1, 3, 4

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 6

[46] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 8

[47] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vi-

sion made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 8

[48] Yunsong Wang, Tianxin Huang, Hanlin Chen, and Gim Hee Lee. Freesplat: Generalizable 3d gaussian splatting towards free view synthesis of indoor scenes. *Advances in Neural Information Processing Systems*, 37, 2025. 2

[49] Christopher Wewer, Kevin Raj, Eddy Ilg, Bernt Schiele, and Jan Eric Lenssen. latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. In *European Conference on Computer Vision*, pages 456–473. Springer, 2024. 3

[50] Haofei Xu, Anpei Chen, Yuedong Chen, Christos Sakaridis, Yulun Zhang, Marc Pollefeys, Andreas Geiger, and Fisher Yu. Murf: Multi-baseline radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[51] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16453–16463, 2025. 1, 2, 3, 4, 6

[52] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. *arXiv preprint arXiv:2403.14621*, 2024. 1, 3

[53] 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*, 2024. 3

[54] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. *arXiv preprint arXiv:2410.24207*, 2024. 6, 8

[55] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 2019. 6

[56] 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*, 2025. 1, 2, 3, 4, 6, 8

[57] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21936–21947, 2025. 8

[58] Shunyuan Zheng, Boyao Zhou, Ruizhi Shao, Boning Liu, Shengping Zhang, Liqiang Nie, and Yebin Liu. Gps-gaussian: Generalizable pixel-wise 3d gaussian splatting for real-time human novel view synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19680–19690, 2024. 2

[59] 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. 2, 3, 6

[60] Chen Ziwen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Li Fuxin, and Zexiang Xu. Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. 2, 3, 6, 7, 8