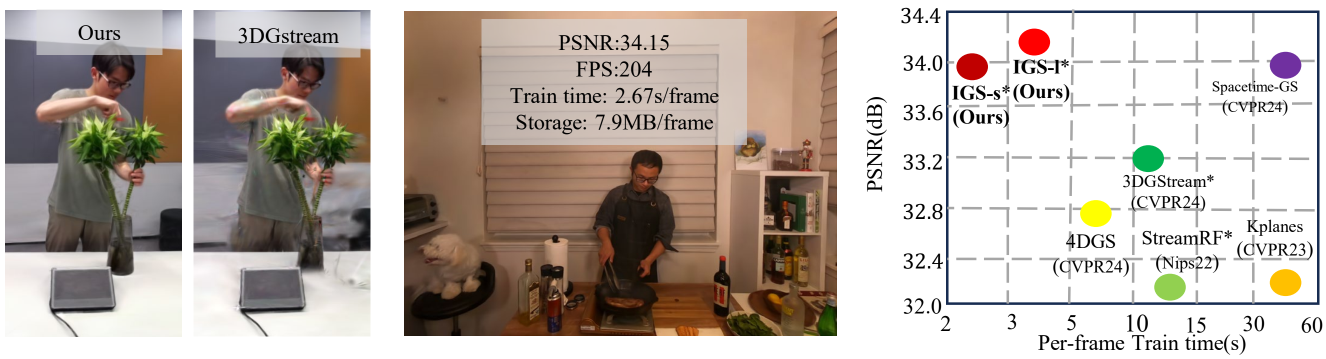
Instant Gaussian Stream: Fast and Generalizable Streaming of Dynamic Scene Reconstruction via Gaussian Splatting

Jinbo Yan,Rui Peng , Zhiyan Wang, Luyang Tang, Jiayu Yang, Jie Liang, Jiahao Wu, Ronggang Wang
Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology,
Shenzhen Graduate School, Peking University
yjb@stu.pku.edu.cn



(a)High Quality Result (b)Realtime Rendering (c)Comparison on Quality and Train Speed

Figure 1. Performance comparison with pervious SOTA[17, 29, 30, 50, 61]. Our method achieves a per-frame reconstruction time of 2.67s, delivering high-quality rendering results in a streaming fashion (a)(b), with a noticeable improvement in performance (c). * denotes a streamable method.

Abstract

Building Free-Viewpoint Videos in a streaming manner offers the advantage of rapid responsiveness compared to offline training methods, greatly enhancing user experience. However, current streaming approaches face challenges of high per-frame reconstruction time (10s+) and error accumulation, limiting their broader application. In this paper, we propose Instant Gaussian Stream (IGS), a fast and generalizable streaming framework, to address these issues. First, we introduce a generalized Anchor-driven Gaussian Motion Network, which projects multi-view 2D motion features into 3D space, using anchor points to drive the motion of all Gaussians. This generalized Network generates the motion of Gaussians for each target frame in the time required for a single inference. Second, we propose a Key-frame-guided Streaming Strategy that refines each key frame, enabling accurate reconstruction of temporally complex scenes while mitigating error accumulation. We conducted extensive in-domain and cross-domain evaluations, demonstrating that our approach can achieve streaming with a average per-frame reconstruction time of 2s+, alongside a enhancement in view synthesis quality.

1. Introduction

Reconstructing Free-Viewpoint Videos (FVV) from multi-view images is a valuable area of research, with applications spanning immersive media such as VR, AR, and sports broadcasting. By enabling interactive, photorealistic visuals, FVVs from dynamic scenes hold the potential to become a next-generation visual medium, offering experiences that go beyond traditional video formats. To enhance user experience, streaming-based FVV construction—where dynamic scenes are reconstructed frame by frame—offers a low-delay response compared to traditional offline training approaches, making it better suited for real-time, interactive applications.

With advancements in real-time rendering and high-quality view synthesis powered by 3D Gaussian Splatting (3DGS)[26], dynamic scene reconstruction has seen rapid progress. Some offline training methods[23, 30, 61, 65, 68, 70] achieve high-quality view synthesis but require complete video sequences for extensive offline training, limiting their practical application, e.g., living streaming. To address these challenges, some methods[29, 50] adopt a streaming framework that reconstructs dynamic scenes frame by frame by modeling inter-frame differences. 3DGStream, with a 12-second per-frame reconstruction time, enables online reconstruction of dynamic scenes, significantly expanding the applications of dynamic scene reconstruction. However, streaming-based dynamic scene reconstruction still faces significant challenges. First, current methods typically require per-frame optimization, resulting in high per-frame latencies(10s+), which severely impact the real-time usability of these systems. Additionally, error accumulation across frames degrades the reconstruction quality of later frames, making it difficult for streaming methods to scale effectively to longer video sequences.

To promote the streaming framework to be more practical, we introduce Instant Gaussian Stream (IGS), a streaming approach for dynamic scene reconstruction that achieves a per-frame reconstruction time of 2s+, mitigates error accumulation, and enhances view synthesis quality. First, to tackle the issue of high per-frame reconstruction time, we developed a generalized Anchor-driven Gaussian Motion Network (AGM-Net). This network employs anchor points to carry motion features that guide Gaussian transformations and enables inference to compute the motion of Gaussian primitives between frames in a single feed forward, eliminating the need for per-frame optimization. Second, to further improve view synthesis quality and minimize error accumulation, we propose a Key-frame-guided Streaming strategy. By establishing key-frame sequences and performing max-point-bounded refinement on key frames, our method mitigates the impact of error accumulation and enhances rendering quality in temporally complex scenes.

We conducted extensive validation in both in-domain and cross-domain scenarios, and the experimental results demonstrate the strong generalization capability of our model, with significant improvements over current state-of-the-art methods in terms of per-frame reconstruction time and rendering quality. To the best of our knowledge, this is the first approach to use a generalized method for streaming reconstruction of dynamic scenes. Our contributions are summarized below.

- We propose a generalized Anchor-driven Gaussian Motion Network that captures Gaussian motion between adjacent frames with a single inference, eliminating the need for frame-by-frame optimization.
- We designed a Key-frame-guided Streaming strategy to enhance our method's capability in handling temporally complex scenes, improving overall view synthesis quality within the streaming framework and mitigating error accumulation.
- The evaluation results in both in-domain and cross-domain scenarios demonstrate the generalization capability of our method and its state-of-the-art performance. We achieve a 2.7 s per-frame reconstruction time for streaming, representing a significant improvement over previous methods. Additionally, we improve view synthesis quality, enabling real-time rendering at 204 FPS while maintaining comparable storage overhead.

2. Related work

2.1. 3D Reconstruction and View Synthesis

Novel view synthesis (NVS) has always been a hot topic in the field of computer vision. By using MLP to implicitly represent the scene, Neural Radiance Fields (NeRF) [37] achieves realistic rendering. Subsequent works have imporved NeRF to enhance rendering quality [1, 2, 59], reduce the number of training views [39, 57, 62, 66], lessen dependence on camera poses [4, 10, 31, 55], and improve both training and inference speeds [3, 8, 16, 20, 21, 38, 43, 44, 49]. 3D Gaussian Splatting (3DGS) [26] employs anisotropic Gaussian primitives to represent scenes and introduces rasterization-based splatting rendering algorithm, enhancing both speed and rendering quality. Some methods focus on various aspects of improving Gaussian field representations, including rendering quality[28, 36, 45, 69, 73, 78], enhancing geometric accuracy[22, 74, 75], and increasing compression efficiency, [11, 13, 36, 67], joint optimization of camera pose and gaussian fields [14, 18, 46], as well 3D generation [9, 53, 54, 80].

2.2. Generalizable 3D Reconstruction for Acceleration

3DGS requires per-scene optimization to achieve realistic rendering results. To accelerate this time-consuming process, some works [24, 27, 51, 52, 76, 79], inspired by generalizable NeRF [7, 25, 60, 64, 72], have proposed to train generalizable Gaussian models on large-scale datasets to enable fast reconstruction. PixelSplat [6] utilizes an Transformer to encode features and decode them into Gaussian attributes. Other generalizable models [12, 15, 34, 77] utilize Transformers or Multi-View Stereo (MVS) [71] techniques to construct cost volumes followed by a decoder, achieving real-time rendering speeds and excellent generalizability. To the best of our knowledge, our work is the first to apply generalizable models to dynamic streaming scenes, utilizing their rapid inference capabilities to accelerate the processing of dynamic scenes reconstruction.

2.3. Dynamic Scene Reconstruction and View Synthesis

There have been numerous efforts to extend static scene reconstruction to dynamic scenes based on NeRF[5, 17, 19, 32, 33, 40, 42, 47, 56]. Since the advent of 3D Gaussian Splatting (3DGS)[26], researchers have explored incorporating its real-time rendering capabilities into dynamic scene reconstruction[23, 30, 61, 65, 68, 70]. For instance, 4D Gaussian Splatting (4DGS)[61] continues the approach of canonical and deformation fields. Other methods[30, 65, 70] lift 3D Gaussian primitives into 4D space by adding a temporal position, achieving high-quality view synthesis.

However, these approaches rely on offline training with full video sequences, making them unsuitable for applications requiring real-time interaction and fast response. To address this issue, existing methods such as StreamRF[29], NeRFPlayer[48], ReRF[58], and 3DGStream[50] reformulates the dynamic modeling problem using an Streaming method. Notably, 3DGStream[50], based on Gaussian Splatting, optimizes a Neural Transformation Cache to model Gaussian movements between frames, further improving the performance. Although these methods achieve promising results, they still rely on per-frame optimization, resulting in significant delays (with current SOTA methods requiring over 10 seconds per frame[50]). Our approach offers a new perspective for streaming dynamic scene modeling: by training a generalized network, we eliminate the need for per-frame optimization, achieving low per-frame reconstruction time alongside high rendering quality.

3. Method

In this section, we begin with an overview of the pipeline in Sec.3.1. Then we introduce the Anchor-driven Gaussian Motion Network (AGM-Net) in Sec.3.2, which is a generalized model that drives Gaussian motion from the previous frame using anchor points. Following this, we present our Key-frame-guided Streaming strategy in Sec. 3.3. Finally, in Sec. 3.4, we outline the loss function used in our training.

3.1. Overview

Our goal is to model dynamic scenes in a streaming manner with minimal per-frame reconstruction time. To achieve this, we adopt a generalized AGM-Net that extracts 3D motion features from the scene using anchor points and drives the motion of Gaussian primitives between frames in a single inference step. And we propose a key-frame-guided Streaming strategy to further improve view synthesis quality and handle temporally complex scenes while addressing error accumulation in streaming reconstruction.

Starting with the Gaussians from the previous key frame as the initial set, we sample M anchor points from them and use these anchor points to drive the motion of all Gaussians, a process achieved by the AGM-Net. Specifically, we construct multi-view motion feature maps and employ a Projection-aware 3D Motion Lift module to project the 2D motion features into 3D space, which are then stored at each anchor point. The motion feature for each Gaussian primitive is interpolated from the neighboring anchor points and decoded into the corresponding motion for each Gaussian between the previous key frame and the target frame. To further enhance the rendering quality during streaming and mitigate error accumulation, we propose a Key-frame-guided Streaming strategy, which involves constructing a key-frame sequence and refining the key frames. The overall pipeline is illustrated in Fig. 2.

3.2. Anchor-driven Gaussian Motion Network

Motion Feature Maps: Given multi-view images of current frames $\mathbf{I}^{'} = (I_1^{'}, ..., I_V^{'})$ with camera parameters , We can first construct a multi-view image pair, which contains the current frame and the previous frame $\mathbf{I}$ from corresponding viewpoints. Then, we use a optical flow model to obtain the intermediate flow embeddings. Next, a modulation layer[9, 41] is applied to inject the viewpoint and depth information into the embeddings, ultimately resulting in 2D motion feature maps $\mathbf{F} \in \mathbb{R}^{V\times C\times H\times W}$.
Anchor Sampling: To deform the Gaussian primitives $\mathcal{G}$ from the previous frame, we need to compute the motion of each Gaussian. However, directly computing the motion for each Gaussian is computationally expensive and memory-intensive due to the large number of Gaussian points. To address this, we employ an anchor-point-based approach to represent the motion features of the entire scene in 3D space. The anchor-driven approach supports batch processing during training, reducing computational overhead while preserving the geometric information of the Gaussian primitives. Specifically, we use **Farthest Point Sampling** (**FPS**) to sample M anchor points from the N Gaussian primitives

$$\mathcal{C} = \mathbf{FPS}(\{\mu_i\}_{i\in N}) \tag{1}$$

where $\mathcal{C} \in \mathbb{R}^{M\times 3}$ represents the sampled anchor points with M set to 8192 in our experiments, and μ_i denotes the position of $\mathcal{G}_i$
Projection-aware 3D Motion Feature Lift: We adopt a projection-aware approach to lift multiview 2D motion features into 3D space. Specifically, we project sampled anchor points onto each motion feature map based on the camera poses, obtaining high-resolution motion features:

$$f_i = \frac{1}{V}\sum_{j\in V}\Psi(\Pi_j(\mathcal{C}_i), F_j) \tag{2}$$

where $\Pi_j(\mathcal{C}_i)$ represents the projection of $\mathcal{C}_i$ onto the image plane of F_j using the camera parameters of F_j, and Ψ denotes bilinear interpolation. By projection, each anchor point can accurately obtain its feature $f_i \in \mathbb{R}^C$ from the multi-view feature map, effectively lifting the 2D motion map into 3D space.

We then use these features $\{f_i\}_{i\in M}$, stored at each anchor point, as input to a Transformer block using self-attention to further capture motion information within the 3D scene.

$$\{z_i : z_i \in \mathbb{R}^C\}_{i\in M} = \mathbf{Transformer}(\{f_i\}_{i\in M}) \tag{3}$$

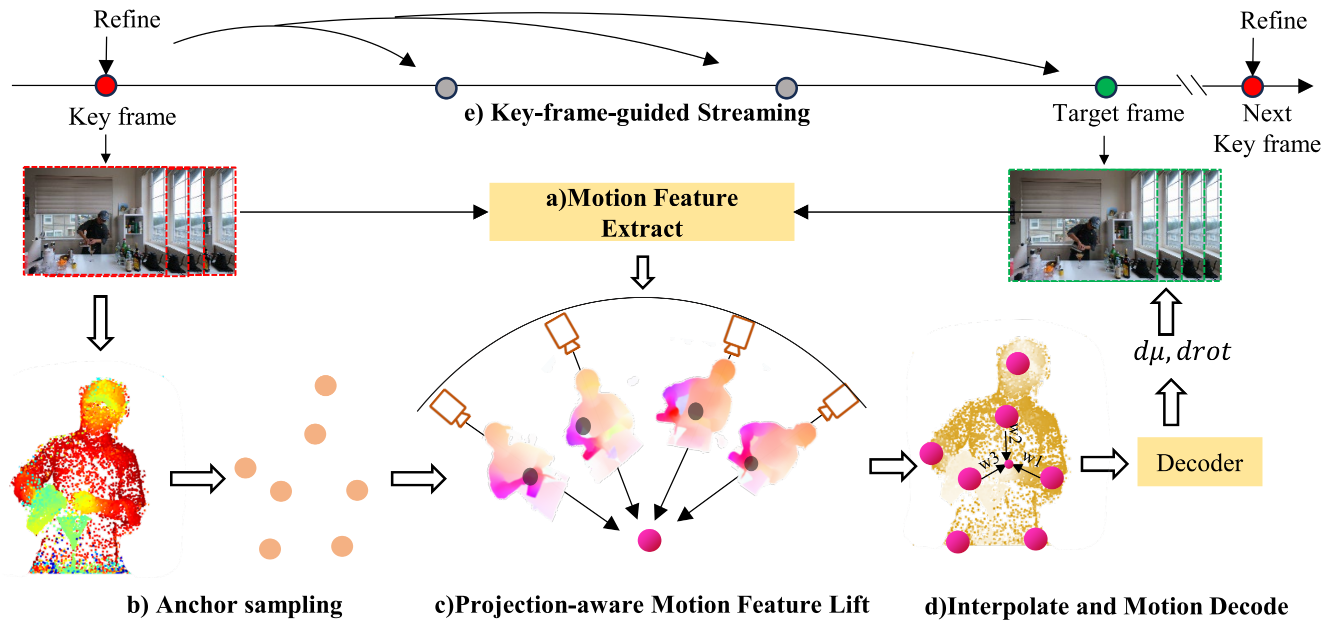


Figure 2. The overall pipeline of IGS. (a) Starting from the key frame and moving towards the target frame, we extract the 2D Motion Feature Map. (b) Then we sample M anchor points from the Gaussian primitives of the key frame, (c) and the anchor points are projected onto these feature maps to obtain 3D motion features through Projection-aware Motion Feature Lift. (d) Each Gaussian point interpolates its own motion feature from neighboring anchors and applies a weighted aggregation of features, which is then decoded into the motion of the Gaussian between the key frame and the target frame. (e) The entire streaming reconstruction process is guided by the Key-frame-guided Streaming strategy, where the key frame directly infers subsequent candidate frames until the next key-frame is reached, at which point max-point bounded refinement is applied to the key-frame.

The output of the Transformer block $\{z_i\}_{i\in M}$ represents the final 3D motion features we obtain. Now, we can use these 3D motion features to represent the motion information of an anchor and its neighborhood, and drive the motion of the neighboring Gaussian points based on these motion features.

Interpolate and Motion Decode: Using the 3D motion features stored at anchor points, we can assign each Gaussian point a motion feature by interpolating from its K nearest anchors in the neighborhood:

$$z_i = \frac{\sum_{k\in\mathcal{N}(i)} e^{-d_k} z_k}{\sum_{k\in\mathcal{N}(i)} e^{-d_k}} \tag{4}$$

where $\mathcal{N}(i)$ represents the set of neighboring anchor points of Gaussian point $\mathcal{G}_i$, and d_k represents the Euclidean distance from Gaussian point G_i to anchor $\mathcal{C}_k$. Then we can use a Linear head to decode the Motion feature to the movement of a Gaussian primitive:

$$d\mu_i, drot_i = \mathbf{Linear}(z_i) \tag{5}$$

here, we use the deformation of the Gaussian's position $d\mu_i$, and the deformation of the rotation $drot_i$, to represent the movement of a Gaussian primitive. The new position and rotation of the Gaussian are as follows:

$$\mu_i^{'} = \mu_i + d\mu_i, \tag{6}$$

$$rot_i^{'} = norm(rot_i) \times norm(drot_i). \tag{7}$$

here $'$ refers to the new attributes. $norm$ denotes to quaternion normalization and $\times$ represents quaternion multiplication, as used in previous work[50].

3.3. Key-frame-guided Streaming

Using AGM-Net, we can transition Gaussian primitives from the previous frame to the current frame within a single forward inference pass. However, this process only adjusts the position and rotation of Gaussian primitives, making it effective for capturing rigid motion but inadequate for accurately representing non-rigid motion. Furthermore, the number of Gaussian points remains constant, limiting its capacity to model temporally dynamic scenes where objects may appear or disappear. These limitations result in challenges in capturing scene dynamics and can lead to error accumulation across frames.

To better model object changes and reduce error accumulation, we propose a Key-frame-guided Streaming strategy that uses key frames as the initial state for deforming Gaussians in subsequent frames. We also introduce a Max points bounded refinement strategy, enabling efficient key frame reconstruction without redundant points and preventing point count growth across frames. This approach helps avoid overfitting in sparse-viewpoint scenes by effectively managing point density.

Key-frame-guided strategy: Starting from frame 0, we designate a key frame every w frames, forming a key-frame sequence$\{K_0, K_w, ..., K_{nw}\}$. The remaining frames serve as candidate frames. During streaming reconstruction, for example, beginning with a key frame K_{iw}, we deform the Gaussians forward across successive candidate frames using AGM-Net until reaching the next key frame $K_{(i+1)w}$. At this point, we refine the deformed Gaussians of key frame $K_{(i+1)w}$. Then, we continue deforming from key frame $K_{(i+1)w}$ to process subsequent frames.

This key-frame-guided strategy offers several advantages. First, when AGM-Net is applied to candidate frames, it is always start from the most recent key frame, preventing error propagation across candidate frames between key frames and eliminating cumulative error. Second, candidate frames do not require optimization-based refinement, as their Gaussians are generated through a single model inference with AGM-Net, ensuring low per-frame reconstrution time. Additionally, we can batch process up to w frames following each key frame, which further accelerates our pipeline.

Max points bounded Key-frame Refinement: During the refinement of each key frame, we optimize all parameters of the Gaussians and support cloning, splitting, and filtering, which is same to 3DGS[26]. This approach allows us to handle object deformations as well as the appearance and disappearance of objects in temporally complex scenes, effectively preventing error accumulation from key frame to subsequent frames. However, this optimization strategy can lead to a gradual increase in Gaussian primitives at each key frame, which not only raises computational complexity and storage requirements but also risks overfitting in sparse viewpoints, particularly in dynamic scenes where viewpoints are generally limited.

To address this, we adopt a Max Points Bounded Refine method. When densifying Gaussian points, we control the number of Gaussians allowed to densify by adjusting each point's gradient, ensuring that the total number of points does not exceed a predefined maximum.

3.4. Loss Function

Our training process consists of two parts: offline training the generalized AGM-Net and performing online training for the key frames. The generalized AGM-Net only needs to be trained once, and it can generalize to multiple scenes. We train the AGM-Net across scenes using gradient descent, relying solely on a view synthesis loss between our predicted views and the ground truth views, which includes an $\mathcal{L}_1$ term and an $\mathcal{L}_{D-SSIM}$ term and can be formulated as:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{D-SSIM} \tag{8}$$

When performing online training on the Gaussians in key frames, we use the same loss function as in Eq.8. However, this time, we optimize the attributes of the Gaussian primitives rather than the parameters of the neural network.

4. Implementation details

In this Section, we first introduce the datasets we used, along with the partitioning and preprocessing of training data, in Sec. 4.1. Next, we provide a detailed explanation of the configuration of the AGM network and the training hyperparameters in Sec. 4.2. Finally, we describe the detailed setup for streaming in Sec. 4.3.

4.1. Datasets

The Neural 3D Video Datasets (N3DV) includes 6 dynamic scenes recorded using a multi-view setup featuring 21 cameras, with a resolution of 2704×2028. Each multi-view video comprises 300 frames.

Meeting Room Datasets includes 3 dynamic scenes recorded with 13 cameras at a resolution of 1280 × 720. Each multi-view video also contains 300 frames.

Dataset Preparation: We split four sequences from the N3DV dataset into the training set, with the remaining two sequences, $\{cut\ roasted\ beef, sear\ steak\}$, used as the test set. For the training set, we constructed 3D Gaussians for all frames in the four training sequences, totaling 1200 frames, , which required 192 GPU hours. For each frame's 3D Gaussian, we performed motions forward and backward for five frames, creating 12,000 pairs for training. To evaluate our model's generalization ability, we fine-tune it on the $discussion$ scene from the Meeting Room dataset for 5 epochs, accounting for variations in scale and camera parameters across different datasets, before conducting cross-domain evaluation. For testing, we selected one viewpoint for evaluation for both datasets, consistent with previous methods.

4.2. AGM Network

We use GM-Flow[63] to extract optical flow embeddings and add a Swin-Transformer[35] block for fine-tuning while keeping the other parameters of GM-Flow fixed. Our AGM model accepts an arbitrary number of input views. To balance computational complexity and performance, we use $V = 4$ views, each producing a motion map with $C = 128$ channels and a resolution of 128 x 128. We sample $M = 8192$ anchor points from Gaussian Points, which sufficiently captures dynamic details. The Transformer block in 3D motion feature lift module comprises 4 layers, yielding a 3D motion feature with $C = 128$ channels. For rendering, we adopt a variant of Gaussian Splatting Rasterization from Rade-GS[75] to obtain more accurate depth maps and geometry.

During training, we randomly select 4 views as input and use 8 views for supervision. Training is conducted on four A100 GPUs with 40GB of memory each, running for a total of 15 epochs with a batch size of 16. Fine-tuning on the Meeting Room dataset takes 5 epochs with a batch size of 32. The parameter γ in Eq. 8 is set to 0.2. We use the Adam optimizer with a weight decay of 0.05, and β values of (0.9, 0.95). The learning rate is set to 4×10^{-4} for training on the N3DV dataset and 4×10^{-5} for fine-tuning on the Meeting Room dataset.

4.3. Streaming Inference

We set $w = 5$ to construct key frame sequences, resulting in 60 keyframes from a 300-frame video, and conduct an ablation study to assess the impact of different w values in Sec. 5.3. We designed two versions for keyframe optimization: a smaller version **IGS**-s (Ours-s) with 50 iterations refinement for Key frames, providing lower per-frame latency, and a larger version **IGS**-l (Ours-l) with 100 iterations, which achieves higher reconstruction quality. In both versions, densification and pruning are performed every 20 iterations. For the test sequences, we construct the Gaussians for the 0th frame using the compression method provided by Lightgaussian[13], which reduces storage usage and mitigates overfitting due to sparse viewpoints. We employ 6,000 iterations for training the first frame of the N3DV dataset, compressing the number of Gaussians at 5,000 iterations. For the Meeting Room dataset, we train the Gaussians of the first frame using 15,000 iterations, compressing the number of Gaussians at 7,000 iterations. For more details, please refer to the Supp..

5. Experiments

5.1. Metrics and Baselines

Baselines: We compare our approach to current state-of-the-art methods for dynamic scene reconstruction, covering both offline and online training methods. Offline methods[17, 30, 61, 65, 70] rely on a set of Gaussian primitives or Hex-planes to represent entire dynamic scenes. Online training methods[29, 50] employ per-frame optimization to support streaming reconstruction. Specifically, 3DGStream[50] models the movement of Gaussian points across frames by optimizing a Neural Transform Cache, creating a 3DGS-based pipeline for free-viewpoint video streaming that enables high-quality, real-time reconstruction of dynamic scenes.

Metrics: Following prior work, we evaluate and report **PSNR**, **Storage** usage, **Train** time, and **Render** Speed to compare with previous methods. All metrics are averaged over the full 300-frame sequence, including frame 0.

Table 1. Comparison on the N3DV dataset, with results measured at a resolution of 1352 x 1014. † indicates that the evaluation was performed using the official code in the same experimental environment as ours, including the same initial point cloud. Highlights denote the **best** and <u>second best</u> results.

Method	PSNR↑ (dB)	Train ↓ (s)	Render↑ (FPS)	Storage↓ (MB)
Offline training				
Kplanes[17]	32.17	48	0.15	1.0
Realtime-4DGS[70]	33.68	-	114	-
4DGS[61]	32.70	7.8	30	**0.3**
Spacetime-GS[30]	33.71	48	140	<u>0.7</u>
Saro-GS[65]	<u>33.90</u>	-	40	1.0
Online training				
StreamRF[29]	32.09	15	8.3	31.4
3DGStream[50]	33.11	12	**215**	7.8
3DGStream[50]†	32.75	16.93	<u>204</u>	7.69
Ours-s	33.89	**2.67**	<u>204</u>	7.90
Ours-l	**34.15**	<u>3.35</u>	<u>204</u>	7.90

5.2. Comparisons

In-domain evaluation: We present our in-domain evaluation on two test sequences from the N3DV dataset, with results shown in Tab. 1. For a fair comparison of performance, we tested 3DGStream using the same Gaussians from the 0th frame and applied the same variant of Gaussian Splatting Rasterization as used in our approach (denoted with † in the table). Compared to 3DGStream and StreamRF, our method achieves a 6x reduction in train time, with an average delay of 2.67 seconds per frame, while maintaining comparable rendering speed and storage usage. Our approach also achieves enhanced rendering quality. Compared to offline training methods, our approach provides low-delay streaming capabilities while achieving state-of-the-art rendering quality and reducing training time. A qualitative comparison of rendering quality can be seen in Fig. 5. It is evident that our method outperforms others in rendering details, such as the transition between the knife and fork, and in modeling complex dynamic scenes, like the moving hand and the shifting reflection on the wall.

We also conducted a PSNR trend comparison with 3DGStream to verify the effectiveness of our method in mitigating error accumulation. The comparison results and the smoothed trends are shown in Fig. 2. As seen, our rendering quality does not degrade with increasing frame number, while 3DGStream suffers from error accumulation, with a noticeable decline in quality as the frame number increases. This confirms the effectiveness of our approach in addressing error accumulation. However, it is also apparent that our method exhibits more fluctuation in per-frame PSNR. This is because 3DGStream assumes small inter-frame motion, leading to smaller adjustments and smoother differences between frames.

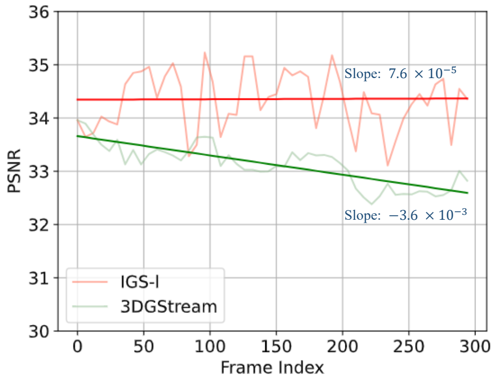


Figure 3. The PSNR trend comparison on the sear steak .

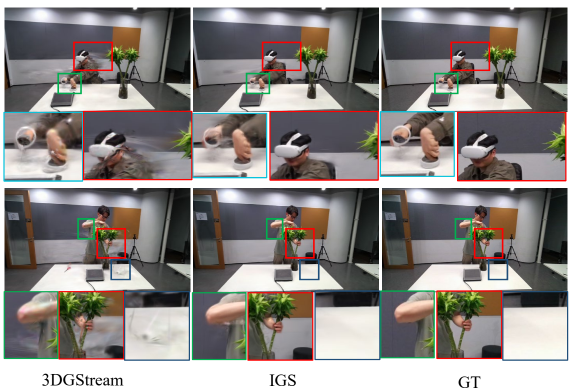


Figure 4. Qualitative comparison from the Meeting Room dataset.

Cross-domain evaluation with fine-tuning: We conducted a cross-domain evaluation on the Meeting Room Dataset, fine-tuning our model for 5 epochs on a single sequence and leaving the other two sequences for testing. The evaluation results are presented in Tab. 2. Our method outperforms 3DGStream in rendering quality, train time, and storage efficiency, achieving streaming with just 2.77s of per-frame reconstruction time, a significant improvement over 3DGStream. This demonstrates the effectiveness and generalizability of our approach, as it enables efficient dynamic scene modeling with streaming capabilities in new environments after only a brief fine-tuning, without requiring per-frame optimization. A qualitative comparison of rendering quality can be seen in Fig. 4. Compared to 3DGStream, which produces artifacts near moving objects, our method yields more accurate motion during large displacements, resulting in improved performance in temporally complex scenes.

5.3. Ablation Study

The use of the pretrained optical flow model: We used a pretrained optical flow model to extract flow embeddings from image pairs, which are then lifted into 3D space. To validate its effectiveness, we replaced the pretrained model with a 4-layer UNet without pretrained parameters and trained it jointly with the overall model. The results in Tab. 3 highlight the benefit of using the 2D prior.

Table 2. Comparison on the Meeting Room dataset. † indicates that the evaluation was performed using the official code in the same experimental environment as ours, including the same initial point cloud.

Method	PSNR↑ (dB)	Train ↓ (s)	Render↑ (FPS)	Storage↓ (MB)
3DGStream[50]†	28.36	11.51	252	4.0
Ours-s	29.35	**2.77**	252	**1.26**
Ours-l	**30.16**	3.20	252	**1.26**

Table 3. Ablation Study Results

Method	PSNR↑ (dB)	Train↓ (s)	Storage↓ (MB)
No-pretrained optical flow model	31.07	2.65	7.90
No-projection-aware feature lift	32.95	**2.38**	7.90
No-points bounded refinement	33.23	3.02	110.26
Ours-s(full)	**33.62**	2.67	7.90

Projection-aware 3D Motion Feature Lift: We use a projection-based approach to lift multi-view 2D motion feature maps into 3D space, accurately linking 3D anchor points to 2D features. To evaluate its effectiveness, we replaced this method with a Transformer-based approach using cross-attention between image features and anchor points, enhanced with positional embeddings through a 4-layer Transformer block. As shown in Tab. 3, Projection-aware Feature Lift is crucial for IGS performance, with only a slight increase in training time.

Key-frame guided Streaming: We employ a key-frame-guided strategy to address error accumulation in streaming and to enhance reconstruction quality. Keyframes are selected and refined through Max-points-bounded Refine. Without this refinement, AGM-Net would rely solely on Gaussians propagated from the last keyframe, resulting in accumulated errors that significantly impact performance, as shown in Fig. 6 (a). We also evaluate the effect of max-points bounding during refinement, as shown in Tab. 3. Without point limits, storage requirements increase substantially, and overfitting causes a decline in view quality.

Key-frame selection: We conducted an ablation study on the interval w for setting keyframes, testing values of $w = 1$, $w = 5$, and $w = 10$, with results shown in Tab. 4. When $w = 1$, every frame becomes a keyframe, leading to excessive optimization that overfits Gaussians to training views, degrading test view quality and increasing training time and storage. Conversely, with $w = 10$, each keyframe drives the next 10 frames, but this distance weakens model performance, as it relies on assumptions about adjacent-frame similarity. The setting $w = 5$ strikes the best balance across view synthesis quality, train time, and storage, and is thus our final choice.

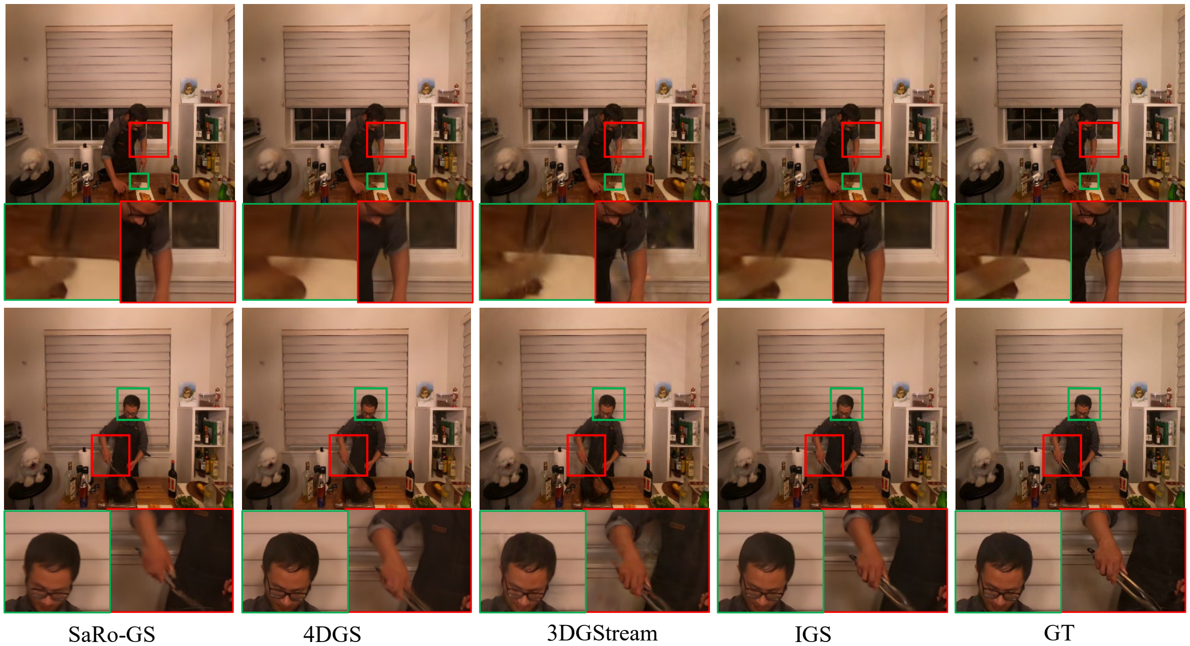
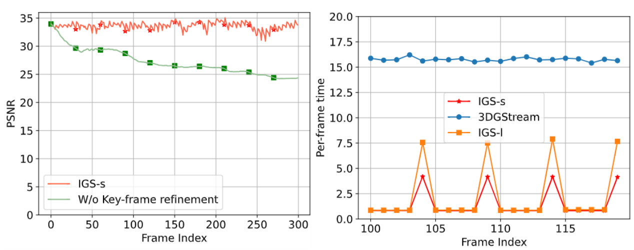


Figure 5. Qualitative comparison from the N3DV dataset.



(a) Ablation Study on Key-frame Refinement (b) Per-frame reconstruction time

Figure 6. (a)Ablation Study on Key-frame Refinement. (b)Per-frame reconstruction time.

Table 4. The impact of different keyframe intervals w.

Method	PSNR(dB)↑	Train(s)↓	Storage(MB)↓
w=1	33.55	6.38	36.0
w=5	**33.62**	**2.67**	7.90
w=10	30.14	2.75	**1.26**

6. Discussion

6.1. Independent per-frame reconstruction time

We further evaluate the performance of IGS by analyzing the independent per-frame reconstruction time, as shown in Fig. 6 (b). The reconstruction time for each frame exhibits a periodic pattern: for candidate frames, it takes 0.8s, while for key frames, it takes 4s and 7.5s for the small and large versions, respectively, which are significantly smaller than the 16s required by 3DGStream.

6.2. Limitation

IGS is the first to use a generalized method for streaming dynamic scene reconstruction, but it has limitations that can be addressed in future work. As shown in Fig. 3, our results exhibit jitter between adjacent frames, caused by the lack of temporal dependencies in the current framework. This makes the model more sensitive to noise. In contrast, 3DGStream assumes minimal motion between frames, yielding smoother results, but it fails in scenes with large motion (Fig. 4). To reduce jitter, we plan to incorporate temporal dependencies into IGS, modeling them as a time series for more robust performance.

7. Conclusion

In this paper, we propose IGS as a novel streaming-based method for modeling dynamic scenes. By adopting a generalized approach, IGS enables frame-by-frame modeling with a per-frame reconstruction time of just over 2 seconds, while maintaining or even surpassing state-of-the-art rendering quality, all while keeping storage requirements low. We introduce a generalized AGM-Net that lifts 2D multi-view motion features to 3D anchor points, using these anchors to drive Gaussian motion. This allows the model to infer the motion of Gaussians between adjacent frames in a single step. Additionally, we propose a Key-frame-guided Streaming strategy, where key frame sequences are selected and refined to mitigate error accumulation, further enhancing rendering quality. Extensive in-domain and cross-domain experiments demonstrate the strong generalization capabilities of our model, reducing significant streaming average cost while achieving state-of-the-art rendering quality, render speed, and storage efficiency.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5855–5864, 2021. 2

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

[4] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4160–4169, 2023. 2

[5] Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023. 2

[6] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19457–19467, 2024. 2

[7] 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

[8] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, pages 333–350. Springer, 2022. 2

[9] Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger. Lara: Efficient large-baseline radiance fields. In *European Conference on Computer Vision (ECCV)*, 2024. 2, 3

[10] Yue Chen, Xingyu Chen, Xuan Wang, Qi Zhang, Yu Guo, Ying Shan, and Fei Wang. Local-to-global registration for bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8264–8273, 2023. 2

[11] 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, 2025. 2

[12] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pages 370–386. Springer, 2025. 2

[13] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, Dejia Xu, and Zhangyang Wang. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. *arXiv preprint arXiv:2311.17245*, 2023. 2, 6

[14] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. *arXiv preprint arXiv:2403.20309*, 2, 2024. 2

[15] Xin Fei, Wenzhao Zheng, Yueqi Duan, Wei Zhan, Masayoshi Tomizuka, Kurt Keutzer, and Jiwen Lu. Pixelgaussian: Generalizable 3d gaussian reconstruction from arbitrary views. *arXiv preprint arXiv:2410.18979*, 2024. 2

[16] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5501–5510, 2022. 2

[17] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023. 1, 2, 6

[18] Yang Fu, Sifei Liu, Amey Kulkarni, Jan Kautz, Alexei A. Efros, and Xiaolong Wang. Colmap-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20796–20805, 2024. 2

[19] Xiang Guo, Jiadai Sun, Yuchao Dai, Guanying Chen, Xiaoqing Ye, Xiao Tan, Errui Ding, Yumeng Zhang, and Jingdong Wang. Forward flow for novel view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16022–16033, 2023. 2

[20] Peter Hedman, Pratul P. Srinivasan, Ben Mildenhall, Jonathan T. Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. *ICCV*, 2021. 2

[21] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19774–19783, 2023. 2

[22] 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

[23] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. *arXiv preprint arXiv:2312.14937*, 2023. 1, 3

[24] 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

[25] 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

[26] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 1, 2, 5

[27] Hao Li, Yuanyuan Gao, Dingwen Zhang, Chenming Wu, Yalun Dai, Chen Zhao, Haocheng Feng, Errui Ding, Jingdong Wang, and Junwei Han. Ggrt: Towards generalizable 3d gaussians without pose priors in real-time. *arXiv preprint arXiv:2403.10147*, 2024. 2

[28] Haolin Li, Jinyang Liu, Mario Sznaier, and Octavia Camps. 3d-hgs: 3d half-gaussian splatting. *arXiv preprint arXiv:2406.02720*, 2024. 2

[29] Lingzhi Li, Zhen Shen, Zhongshu Wang, Li Shen, and Ping Tan. Streaming radiance fields for 3d video synthesis. *Advances in Neural Information Processing Systems*, 35: 13485–13498, 2022. 1, 2, 3, 6

[30] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. *arXiv preprint arXiv:2312.16812*, 2023. 1, 3, 6

[31] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5741–5751, 2021. 2

[32] Haotong Lin, Sida Peng, Zhen Xu, Tao Xie, Xingyi He, Hujun Bao, and Xiaowei Zhou. High-fidelity and real-time novel view synthesis for dynamic scenes. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–9, 2023. 2

[33] Jia-Wei Liu, Yan-Pei Cao, Weijia Mao, Wenqiao Zhang, David Junhao Zhang, Jussi Keppo, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Devrf: Fast deformable voxel radiance fields for dynamic scenes. *Advances in Neural Information Processing Systems*, 35:36762–36775, 2022. 2

[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, 2025. 2

[35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 5

[36] 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. 2

[37] 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. 2

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

[39] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 2

[40] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. *arXiv preprint arXiv:2106.13228*, 2021. 2

[41] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 3

[42] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 2

[43] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14335–14345, 2021. 2

[44] Christian Reiser, Rick Szeliski, Dor Verbin, Pratul Srinivasan, Ben Mildenhall, Andreas Geiger, Jon Barron, and Peter Hedman. Merf: Memory-efficient radiance fields for real-time view synthesis in unbounded scenes. *ACM Transactions on Graphics (TOG)*, 42(4):1–12, 2023. 2

[45] 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

[46] Christian Schmidt, Jens Piekenbrinck, and Bastian Leibe. Look gauss, no pose: Novel view synthesis using gaussian splatting without accurate pose initialization. In *IROS*, 2024. 2

[47] Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16632–16642, 2023. 2

[48] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2732–2742, 2023. 3

[49] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 2

[50] Jiakai Sun, Han Jiao, Guangyuan Li, Zhanjie Zhang, Lei Zhao, and Wei Xing. 3dgstream: On-the-fly training of 3d gaussians for efficient streaming of photo-realistic free-

viewpoint videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20675–20685, 2024. 1, 2, 3, 4, 6, 7

[51] Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, João F Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. *arXiv preprint arXiv:2406.04343*, 2024. 2

[52] 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*, pages 10208–10217, 2024. 2

[53] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023. 2

[54] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024. 2

[55] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4190–4200, 2023. 2

[56] Chaoyang Wang, Ben Eckart, Simon Lucey, and Orazio Gallo. Neural trajectory fields for dynamic novel view synthesis, 2021. 2

[57] Guangcong Wang, Zhaoxi Chen, Chen Change Loy, and Ziwei Liu. Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9065–9076, 2023. 2

[58] Liao Wang, Qiang Hu, Qihan He, Ziyu Wang, Jingyi Yu, Tinne Tuytelaars, Lan Xu, and Minye Wu. Neural residual radiance fields for streamably free-viewpoint videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 76–87, 2023. 3

[59] Peng Wang, Yuan Liu, Zhaoxi Chen, Lingjie Liu, Ziwei Liu, Taku Komura, Christian Theobalt, and Wenping Wang. F2-nerf: Fast neural radiance field training with free camera trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4150–4159, 2023. 2

[60] 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

[61] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20310–20320, 2024. 1, 3, 6

[62] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21551–21561, 2024. 2

[63] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8121–8130, 2022. 5

[64] Han Xu, Jiteng Yuan, and Jiayi Ma. Murf: Mutually reinforcing multi-modal image registration and fusion. *IEEE transactions on pattern analysis and machine intelligence*, 45(10):12148–12166, 2023. 2

[65] Jinbo Yan, Rui Peng, Luyang Tang, and Ronggang Wang. 4d gaussian splatting with scale-aware residual field and adaptive optimization for real-time rendering of temporally complex dynamic scenes. In *Proceedings of the 32nd ACM International Conference on Multimedia*, page 7871–7880, New York, NY, USA, 2024. Association for Computing Machinery. 1, 3, 6

[66] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8254–8263, 2023. 2

[67] Runyi Yang, Zhenxin Zhu, Zhou Jiang, Baijun Ye, Xiaoxue Chen, Yifei Zhang, Yuantao Chen, Jian Zhao, and Hao Zhao. Spectrally pruned gaussian fields with neural compensation. *arXiv preprint arXiv:2405.00676*, 2024. 2

[68] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023. 1, 3

[69] Ziyi Yang, Xinyu Gao, Yangtian Sun, Yihua Huang, Xiaoyang Lyu, Wen Zhou, Shaohui Jiao, Xiaojuan Qi, and Xiaogang Jin. Spec-gaussian: Anisotropic view-dependent appearance for 3d gaussian splatting. *arXiv preprint arXiv:2402.15870*, 2024. 2

[70] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. In *International Conference on Learning Representations (ICLR)*, 2024. 1, 3, 6

[71] 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

[72] 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

[73] 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. 2

[74] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics*, 2024. 2

[75] Baowen Zhang, Chuan Fang, Rakesh Shrestha, Yixun Liang, Xiaoxiao Long, and Ping Tan. Rade-gs: Rasterizing depth in gaussian splatting. *arXiv preprint arXiv:2406.01467*, 2024. 2, 5

[76] Chuanrui Zhang, Yingshuang Zou, Zhuoling Li, Minmin Yi, and Haoqian Wang. Transplat: Generalizable 3d gaussian splatting from sparse multi-view images with transformers. *arXiv preprint arXiv:2408.13770*, 2024. 2

[77] 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. 2

[78] Zheng Zhang, Wenbo Hu, Yixing Lao, Tong He, and Hengshuang Zhao. Pixel-gs: Density control with pixel-aware gradient for 3d gaussian splatting. In *ECCV*, 2024. 2

[79] 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

[80] Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10324–10335, 2024. 2