This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

LocalDyGS: Multi-view Global Dynamic Scene Modeling via Adaptive Local Implicit Feature Decoupling

Jiahao Wu[1,2], Rui Peng[1,2], Jianbo Jiao[3], Jiayu Yang[2], Luyang Tang[1,2]
Kaiqiang Xiong[1,2], Jie Liang[1] Jinbo Yan[1], Runling Liu[1] Ronggang Wang[1,2]†
[1] Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology
Shenzhen Graduate School, Peking University
[2] Pengcheng Lab [3] University of Birmingham

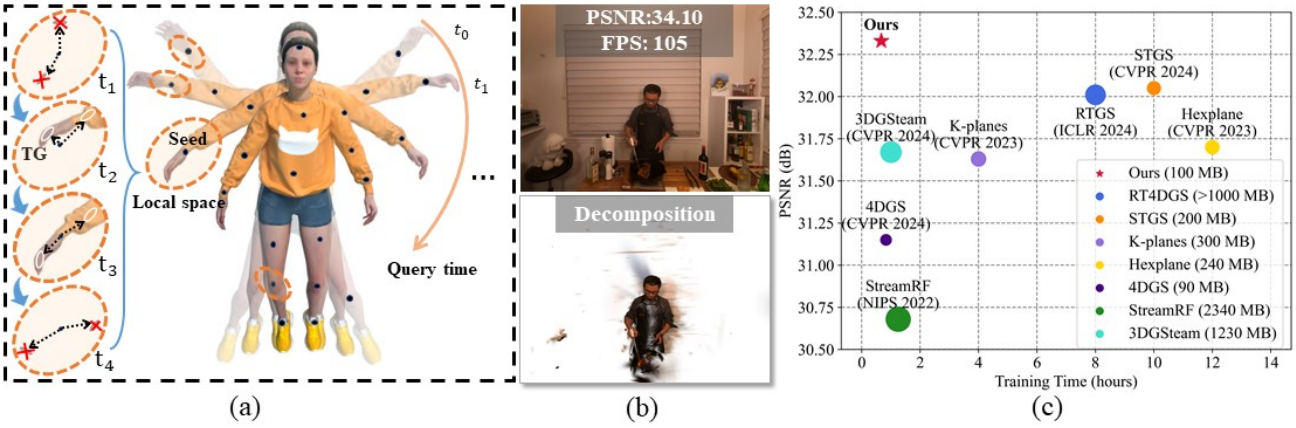


Figure 1. (a) shows our foundational idea: Decomposing a globally complex dynamic scene into a series of streamlined local spaces. The Temporal Gaussian (TG) activates only when the arm enters the local space, generating varying TGs to represent the motion of the arm, and deactivates once the arm exits. (b) displays our high-quality rendering results and the accuracy of dynamic-static decoupling, while (c) demonstrates the superior performance of our method compared to other approaches on the N3DV [24] dataset.

Abstract

Due to the complex and highly dynamic motions in the real world, synthesizing dynamic videos from multi-view inputs for arbitrary viewpoints is challenging. Previous works based on neural radiance field or 3D Gaussian splatting are limited to modeling fine-scale motion, greatly restricting their application. In this paper, we introduce LocalDyGS, which consists of two parts to adapt our method to both large-scale and fine-scale motion scenes: 1) We decompose a complex dynamic scene into streamlined local spaces defined by seeds, enabling global modeling by capturing motion within each local space. 2) We decouple static and dynamic features for local space motion modeling. A static feature shared across time steps captures static information, while a dynamic residual field provides time-specific features. These are combined and decoded to generate Temporal Gaussians, modeling motion within each local space. As a result, we propose a novel dynamic scene reconstruction framework to model highly dynamic real-world scenes more realistically. Our method not only demonstrates competitive performance on various fine-scale datasets compared to state-of-the-art (SOTA) methods, but also represents the first attempt to model larger and more complex highly dynamic scenes. Project page: https://wujh2001.github.io/LocalDyGS/.

1. Introduction

Multi-view dynamic scene reconstruction is a crucial and challenging problem with a wide range of applications, such as free-viewpoint control for sports events and movies, AR, VR, and gaming. For monocular dynamic scene reconstruction, due to the lack of accurate geometric information in monocular input, it is often limited to reconstructing relatively simple scenes [33, 34, 54] and struggles to model highly dynamic and complex scenes, thus failing to provide users with a more immersive visual experience. To model the complex and dynamic real-world scenes with high quality, a widely adopted solution is

†Corresponding author.

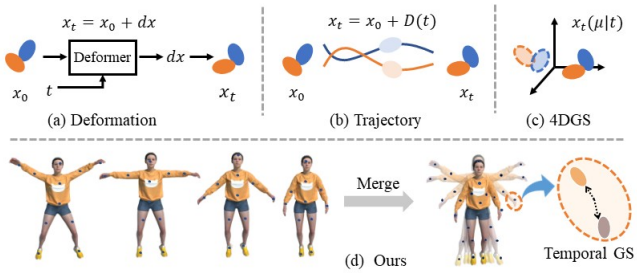


Figure 2. Overview of existing dynamic methods.

to use multi-view synchronized videos to provide dense spatiotemporal supervision [4, 24, 28, 42, 63].

Many researchers have explored multi-view dynamic scene reconstruction from different perspectives to enhance visual quality. For example, 3DGStream [39] utilizes a Neural Transformation Cache (NTC) to model each frame individually, enabling streaming dynamic scene reconstruction. SpaceTimeGS [26] employs polynomials to control the motion trajectories and opacity of Gaussian points, thereby representing the entire dynamic scene. More recently, Swift4D [49] leverages pixel variance as a supervisory signal to decouple static and dynamic Gaussian points. Meanwhile, they validate their approach using a basketball court dataset [40] with larger motion scales and a more complex environment. Despite significant progress, challenges remain: 1) flickering and blurring issues with large-scale complex motion datasets, and 2) high training time and storage requirements.

Therefore, we propose *LocalDyGS*, a multi-view dynamic method adaptable to both fine and large-scale motion, comprising two-fold: 1) decomposing the global space into local spaces, and 2) generating Temporal Gaussians to model motion within each local space (*Local space* is defined as the space surrounding a seed). Specifically, our method no longer explicitly models the longtime motion of each Gaussian point. Instead, as shown in Fig. 1 (a), we use seeds to decompose the complex 3D space into an array of independent local spaces. Local motion modeling is then achieved by generating Temporal Gaussian within each local space, enabling global dynamic scene representation. When seeds cover all regions where a moving object appears, this local space motion modeling approach has the potential to handle large-scale dynamic scenes. To ensure complete coverage, we use a fused Structure from Motion (SfM) [37] point cloud from multiple frames, positioning seed points across all areas with moving objects.

For specific details on local space motion modeling, we assign a learnable static feature shared across all time steps to represent time-invariant static information. As for the dynamic information within each local space, we also construct a dynamic residual field that provides unique dynamic residual features at each time step. With the static feature as a base, dynamic features at each time step generate distinct Temporal Gaussians to model motion over time. This decoupled design helps reduce the load on the dynamic residual field. Next, an adaptive weight field is designed to balance static and dynamic residual features. These features are combined through a weighted linear sum and decoded by a dedicated multilayer perceptron (MLP) to produce the corresponding Temporal Gaussian, capturing motion within the local space. Finally, we propose an adaptive error-based seed growth strategy to alleviate incomplete coverage in the initial point cloud, thereby improving the model's robustness to the SfM [37] initialized point cloud. In summary, our contributions can be outlined as follows:

- We propose to decompose the 3D space into seed-based local spaces, enabling global dynamic scene modeling with the capacity for multi-scale motion.
- We propose decomposing scene features into static and dynamic components to simplify local dynamic modeling and enhance rendering quality.
- We designed a unique Adaptive Seed Growing strategy to address the issue of incomplete coverage of dynamic scenes by the initial point cloud.
- We are the first to extend dynamic reconstruction to a large scale, and extensive experiments validate our superior performance across various metrics.

2. Related Work

2.1. Novel View Synthesis for Static Scenes.

Synthesizing novel views for static scenes is a classical and well-studied problem. Previous years, NeRF has emerged as a groundbreaking work in novel view synthesis, inspiring a series of new view synthesis approaches aimed at improving training speed and rendering quality [5–7, 10, 13, 17], surface reconstruction [44, 46], autonomous driving [48, 51], SLAM [61] etc.. Recently, 3DGS [20] has garnered significant attention in the community for its rapid model training and real-time inference, achieving SOTA visual quality. Many advanced works have emerged, focusing on surface reconstruction [18, 35, 58, 59], few-shot [12, 57, 62] and pose-free methods [8, 15], HDR [50, 52]. In particular, recent works [30, 36] suggest that world space is sparse and can be represented using a set of structural points to represent a class of points to achieve more compact 3D scene representation [19, 30, 41].

2.2. Novel View Synthesis for Dynamic Scenes.

Synthesizing novel views for dynamic scenes is a more challenging and applicable problem. A variety of NeRF-based dynamic scene methods, such as deformation field [16, 32, 34], scene flow [25], and multi-plane [9, 14], have been proposed. Among these, [38] and [43] specifically decouple dynamic and static elements to achieve higher

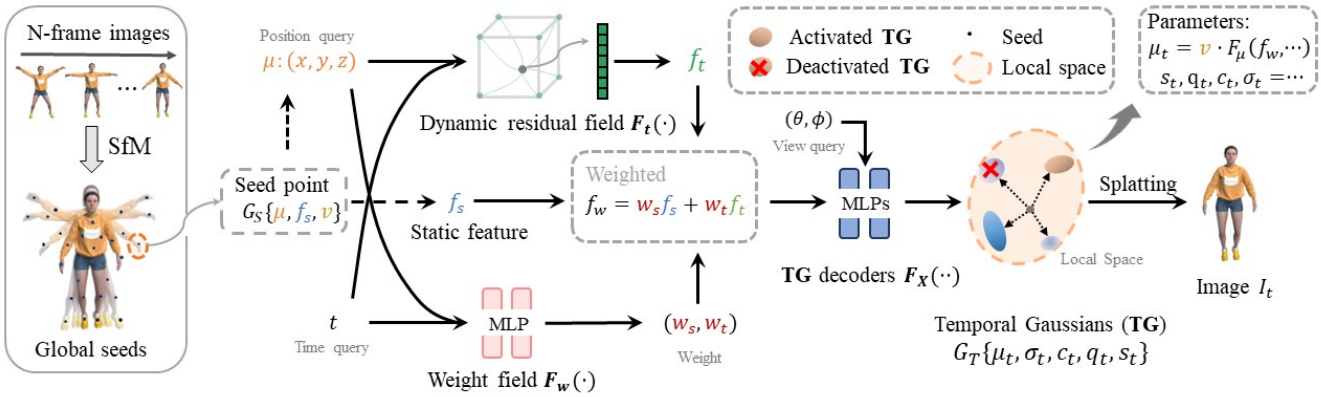


Figure 3. **Overview of LocalDyGS.** We sample N frames across the time domain to extract the SfM [37] point cloud, using it to initialize seeds and local spaces, with each seed assigned two learnable parameters: a static feature f_s shared across all time steps, and a scale v defining the local space range. Additionally, we construct a global dynamic residual field and a weighting field to provide temporal information for the local space. The two are combined through weighted linear summation to obtain the weighted feature f_w, which is then passed through a dedicated Temporal Gaussian (TG) decoder to predict parameters such as mean and color of the Temporal Gaussians. Finally, we perform a deactivation operation to remove Temporal Gaussians that do not belong to the query time t for rasterization. *We use the 'jumping' sequence from the D-NeRF monocular dataset for demonstration, but our method is based on multi-view reconstruction.*

rendering quality. A more related topic to our work, 3DGS-based dynamic methods, has emerged in recent literature and can be roughly categorized into three types: As shown in Fig. 2 (a), deformation field methods [3, 19, 41, 47, 53, 56, 60], represented by [47], which map Gaussian points in a canonical field to a deformation field to represent dynamic scenes at each timestamp. As shown in Fig. 2 (b), trajectory tracking-based solutions [22, 27] typically use polynomials or Fourier series to represent the motion trajectory of each Gaussian. As shown in Fig. 2 (c), methods that extend 3DGS to 4DGS [11, 55] require a large number of Gaussian points for fitting, resulting in high storage requirements and slower training speeds.

Monocular and multi-view dynamic scene. Although many recent monocular dynamic reconstruction works [19, 33, 45, 56] have advanced the field, relying solely on monocular video as input remains challenging for reconstructing complex real-world scenes. Current methods are still limited to synthetic datasets [34] or simple motion scenarios [33, 54]. In contrast, for complex real-world reconstruction, leveraging multi-view synchronized videos to provide dense spatiotemporal supervision appears to be more promising. Dynerf [24], 3DGStream [39], and SpaceTimeGS [26] etc. have explored multi-view dynamic scenes, demonstrating the potential of free-viewpoint outputs from multi-view inputs. However, as shown in our experiments, they suffer from blurring and flickering in real-world scenes with complex, large-scale motion, limiting their applicability. To address this, we propose *LocalDyGS*, which handles larger-scale and fine-scale motion scenes with a more compact structure, faster training speed, and higher-quality rendering.

3. Method

In this section, we introduce LocalDyGS , which consists of two main components: 1) decomposing the global space into local spaces and 2) generating Temporal Gaussians to model motion within each local space, as shown in Fig. 3. In the following subsections, we first describe the initialization of our seeds and local spaces. Next, we introduce the spatio-temporal fields, which equip each local space with essential temporal information for dynamic modeling. Finally, we explain the densification process and the training of our method.

3.1. 3DGS and ScaffoldGS Preliminary

As an emerging popular technique for novel view synthesis, 3DGS [20] uses 3D Gaussians as rendering primitives. Each primitive is defined as $G\{\mu, q, s, \sigma, c\}$, where the parameters represent mean (μ), rotation (q), scaling (s), opacity (σ), and color (c), respectively. A 3D Gaussian point $G(x)$ can be mathematically defined as:

$$G(x) = e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \tag{1}$$

where Σ is the covariance of the 3D Gaussian, typically represented by q and s. During the rendering stage, as described in [64], the 3D Gaussian is projected into a 2D Gaussian $G'(x)$. Then, the rasterizer sorts the 2D Gaussians and applies α-blending.

$$C(p) = \sum_{i \in K} c_i \alpha_i(p) \prod_{j=1}^{i-1}(1 - \alpha_j(p)), \quad \alpha_i(p) = \sigma_i G'_i(p). \tag{2}$$

Here, p represents the position of the pixel, and K denotes the number of 2D Gaussians intersecting with the queried

pixel. Finally, end-to-end training can be achieved through supervised views.

A work closely related to ours is the static reconstruction method ScaffoldGS [30], in which the scene is represented using anchors. Each anchor is associated with the following attributes: a mean position $\mu_a \in \mathbb{R}^3$, a static feature vector $f_a \in \mathbb{R}^{32}$, a scale factor $l_a \in \mathbb{R}^3$, and offsets $O_a \in \mathbb{R}^{k\times 3}$ corresponding to k Gaussian points. The positions of neural Gaussians are calculated as:

$$\{\mu_0, ..., \mu_{k-1}\} = \mu_a + \{O_0, ...O_{k-1}\} \cdot l_a. \tag{3}$$

In addition, the other Gaussian parameters are also decoded using MLPs. To distinguish from the Scaffold *anchor*, we use the *Seed* to refer to the anchor in our dynamic method.

3.2. Global Seeds Initialization

In our framework, as shown in Fig. 3, LocalDyGS fuses SfM point clouds from N frames across the time domain to initialize seed positions μ, providing prior knowledge of where dynamic objects appear. One of our core ideas is that each sparse seed point models the temporal dynamics of only its surrounding 3D scene (referred to as local space), rather than performing long-term motion tracking as in previous methods [26, 39]. This means we allow a moving object to be represented by multiple seeds. As shown in Fig. 3, the moving arm is modeled by a series of different seeds. This significantly reduces the complexity of motion modeling.

For local space modeling, since static information occupies a large portion of the scene and varies significantly across each local space, we assign each local space an independently optimized static feature $f_s \in \mathbb{R}^{64}$ to more accurately capture the static information. This feature is shared across all time steps and initialized to 0. Additionally, we assign each local space a scale parameter $v \in \mathbb{R}^3$ to control the spatial range of its influence. It is initialized as the average distance between the three nearest seed points. In areas with sparser seed points, the local space for each seed becomes larger. Finally, a seed in local space can be defined as $G_S\{\mu, f_s, v\}$.

- the position of seed (global parameter)
- static feature of local space (shared across all time steps)
- the scale of local space

3.3. Feature-Decoupled Spatio-Temporal Fields

At first, we attempted to model the entire scene using a single spatio-temporal structure (without static features), following previous methods [48]. However, we found that this approach causes blurring in both dynamic and static regions, as shown in Fig. 9. We speculate that a single model struggles to store such vast scene information. Inspired by the Deformable-based method [48, 49, 56], they use the canonical field as a base and introduce a time-aware deformation field to reconstruct dynamic scenes. At a more fundamental feature level, we decouple scene information into static and dynamic residual features. Specifically, for each local space, we use independent static features $f_s \in \mathbb{R}^{64}$ as the foundation, capturing most of the local space's static information, while a shared dynamic residual field F_d encodes temporal variations for each local space, enabling dynamic scene reconstruction.

Since motion often exhibits local similarity, we need a compact, adaptive structure to deliver dynamic residual features while preserving locality to ensure neighboring seeds share similar features. Inspired by [31], we construct the dynamic residual field by combining multi-resolution four-dimensional hash encoding with a shallow, fully-fused MLP. Specifically, each voxel grid node at different resolutions is mapped to a hash table storing d-dimensional learnable feature vectors. Given a seed point and query time $(\mu, t) \in \mathbb{R}^4$, its hash encoding at the resolution level l can be written as: $h_{4d}(\mu, t; l) \in \mathbb{R}^m$. This encoded feature is a linear interpolation of the feature vectors corresponding to the vertices of the grid surrounding the insertion point. Therefore, the hash-encoded feature across L resolutions can be expressed as:

$$f_h(\mu, t) = [h_{4d}(\mu, t; 1), h_{4d}(\mu, t; 2), ..., h_{4d}(\mu, t; L)]. \tag{4}$$

We then employ a shallow, fully-fused MLP ϕ to cross-fuse the hash features from each resolution level.

$$f_d = F_d(\mu, t) = \phi(f_h(\mu, t)). \tag{5}$$

To enable the model to adaptively balance its learning of static and dynamic residual features and accelerate convergence [33], we designed a weight field F_w, implemented with a shallow MLP, to predict the weights w_s and w_d for these features: $w_s, w_d = F_w(\mu, t)$. Given a seed and query time t, we collect the outputs from the above fields and compute the weighted feature vector f_w for this seed as follows:

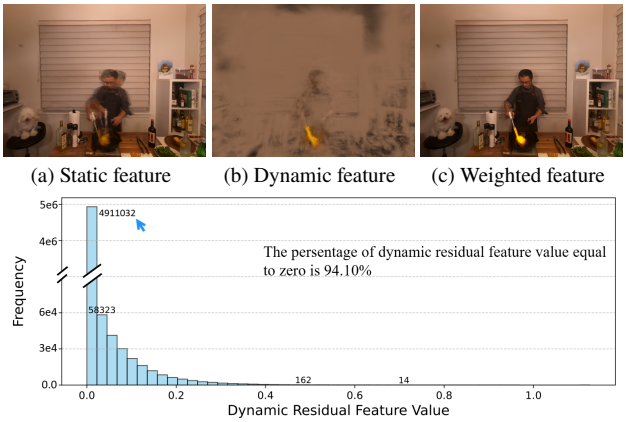
$$f_w = w_s f_s + w_d f_d \tag{6}$$

where weighted feature vector f_w represents the geometric information of the scene at position μ at time t.

In summary, the dynamic residual field supplies each local space with dynamic residual features to represent motion, which often approach zero, as shown in Fig. 4 (d). When decoded and rendered, these features effectively capture the temporal details, as shown in Fig. 4 (b). The predominant static information is provided by the static feature f_s, depicted in Fig. 4 (a). Together, these elements accurately represent the entire scene, as shown in Fig. 4 (c). This decoupling enables more effective modeling of static and dynamic components, improving rendering quality, especially in large-scale motion scenes.

3.4. Local Temporal Gaussian Derivation

In this section, we explain how to generate Temporal Gaussians from each seed as the final rendering



(a) Static feature (b) Dynamic feature (c) Weighted feature

(d) Distribution of dynamic residual feature value

Figure 4. (a) and (b) show the results decoded with static and dynamic residual features, demonstrating a clear separation effect; (c) shows weighted feature results; (d) shows the distribution of dynamic residual values across all local spaces. Scene information is primarily represented by static features, while dynamic residual features only capture temporal residual details; therefore, dynamic residual features tend to approach zero.

primitives. Each Temporal Gaussian is parameterized as $G_t\{\mu_t, q_t, s_t, \sigma_t, c_t\}$, where t denotes the query time, allowing the Temporal Gaussian to have varying parameters over time. Each seed produces k Temporal Gaussians, with their means given by:

$$\{\mu_t^i\}_{i=0}^{k-1} = \mu + v \cdot F_\mu(f_w) \tag{7}$$

where μ and v denote the position and scale parameters of the local space, as described in Sec. 3.2. μ_t^i represents the i-th Temporal Gaussian generated from the seed at time t, and $F_\mu(\cdot)$ is a shallow MLP that outputs a vector of size $k \times 3$. Inspired by [30], the other Temporal Gaussian parameters are similarly predicted using individual MLPs F_*. For instance, the opacity can be represented as:

$$\{\sigma_t^i\}_{i=0}^{k-1} = \text{Sigmoid}(F_o(f_w, \mathbf{d})), \quad \mathbf{d} = \frac{\mu - \mu_c}{\|\mu - \mu_c\|_2} \tag{8}$$

where μ_c is the center of the observed camera coordinates, with quaternions $\{q_t^i\}$ and scales $\{s_t^i\}$ similarly derived.

Temporal Gaussian deactivation. Through experiments, we find that some local spaces only model moving objects at the query time t_a. At other times t_b ($b \neq a$), most Temporal Gaussians in these local spaces exhibit low opacity σ_t^i, contributing minimally to the scene representation while increasing computational load. Therefore, we set a threshold τ_α to deactivate these Temporal Gaussians, reducing computational load without affecting rendering quality. The specific ablation experiments can be found in ablation studies.

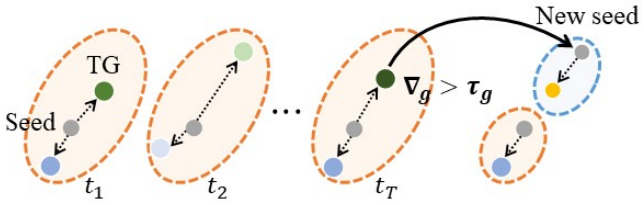


Figure 5. We add seeds where the 2D projection gradient ∇_g of Temporal Gaussian exceeds the threshold τ_g over time $\{t_1, .., t_T\}$.

3.5. Adaptive Seed Growing

The sparse point cloud initialized by SfM often suffers from incomplete scene coverage, especially in areas with weak textures and limited observations [20, 30]. This lack of coverage makes it challenging to construct precise local spaces for scene modeling, which in turn hinders convergence to high rendering quality. To address this challenge, we propose an Adaptive Seed Growing (ASG), an error-based seed growth approach where new seeds are added in important regions identified by the Temporal Gaussians. As shown in Fig. 5, within each local space, we record the maximum 2D projection gradient ∇_{max}^i and its 3D position μ_{max}^i for the i-th Temporal Gaussian during the n iterations. This is mathematically expressed as:

$$\{\nabla_{max}^i, \mu_{max}^i\} = \max_{t \in T}\{\nabla_t^i, \quad \mu_t^i\} \tag{9}$$

where T represents the set of query times corresponding to n iterations. If $\nabla_{max}^i > \tau_g$, additional seed filling is needed, so a seed is added at μ_{max}^i to model motion in that local space. This gradient-based growth method helps to address the limitations of the initial point cloud, enhancing the model's robustness in scene modeling. For detailed ablation studies, refer to Tab. 4.

3.6. Loss Function

To encourage generating small Temporal Gaussians at each query time t, making each responsible only for its corresponding local space, we apply a volume regularization L_v, similar to that in [29, 30], defined as:

$$L_v = \sum_{i=1}^{M} \text{Prod}(s_t^i) \tag{10}$$

where M denotes the number of active Temporal Gaussians from all local spaces, $\text{Prod}(\cdot)$ represents the product of the vector values, and s_t^i is the scaling of each active Temporal Gaussian at query time t. Following the 3DGS approach, we incorporate L_1 and L_{SSIM} losses to enhance reconstruction quality. The total loss function is defined as:

$$L = (1 - \lambda_{SSIM})L_1 + \lambda_{SSIM}L_{SSIM} + \lambda_v L_v. \tag{11}$$

4. Experiments

4.1. Implementation

Our method is primarily compared with current open-source SOTA methods, including SpacetimeGS [26], 4DGS [47], and 3DGStream [39]. We maintain the same training iterations as 3DGS, using 30,000 iterations. For our method, we set $k = 10$ for all experiments, and all MLPs are 2-layer networks with ReLU activation, with the output activation function using Sigmoid or normalization. The dimensions of the dynamic and static features are set to 64. The hash table size is set to 2^{17}, with other settings consistent with INGP [31]. For the ASG method, we start from 3,000 iterations to 15,000 iterations, implementing the seed point growth strategy every 100 iterations, with $\tau_g = 0.001$. The deactivation threshold of Temporal Gaussian is set to $\tau_\alpha = 0.01$. The two loss weights, λ_{SSIM} and λ_{vol}, are set to 0.2 and 0.001, respectively, with the optimizer being Adam [21], following the learning rate of 3DGS [20]. All experiments are conducted on an NVIDIA RTX 3090 GPU.

4.2. Datasets

We primarily evaluate our method on the fine-scale motion datasets N3DV [24] and MeetRoom [23], consistent with most multi-view methods [23, 39, 43]. To further evaluate the robustness of our method in large-scale dynamic scenes, we test our method on more challenging VRU basketball court dataset [49].

The N3DV dataset [24] is a widely used benchmark, captured by a multi-view system of 21 cameras, recording dynamic scenes at a resolution of 2704×2028 and 30 FPS. Following previous work [24, 39, 47], we downsample the dataset and split cameras for training and testing.

The MeetRoom dataset [23] is even more challenging, captured by a multi-view system with only 13 cameras, recording dynamic scenes at a resolution of 1280×720 and 30 FPS. In line with prior work [23, 39], we use 12 cameras for training and reserve one for testing.

The VRU Basketball Court dataset [40] is captured using a 34-camera multi-view system, recording real-world basketball games *GZ*, *DG4* at 1920×1080 resolution and 25 FPS. We use 30 cameras for training, reserving 4 cameras (cameras 0, 10, 20, 30) for testing. This dataset is used for the first time in Swift4D [49], and is provided by AVS-VRU [2] for academic use. Compared to previous **fine-scale** motion datasets [23, 24], it features **larger motion scales** and better evaluates the dynamic modeling capability of the methods.

4.3. Comparisons

Quantitative comparisons. We benchmark LocalDyGS by quantitatively comparing it across the three datasets mentioned above and against a range of SOTA methods, including offline methods like 4DGS [47] and SpaceTimeGS [26], as well as the online method 3DGStream [39]. To verify our method's outstanding performance, we extract the reported quantitative results on the N3DV dataset from their respective papers, and present the average rendering speed, training time, required storage, PSNR, SSIM, and LPIPS for all scenes in the N3DV dataset in Tab. 1. The results show that our method surpasses previous SOTA methods in multiple aspects, achieving the current SOTA level in quality, while delivering over 10x the speed and requiring only half the storage of the previous SOTA method [26]. To demonstrate the generalizability of LocalDyGS, we also conduct experiments on the MeetRoom dataset introduced in StreamRF [23]. As shown in Tab. 2, our method is competitive with the current SOTA streaming method, 3DGStream, particularly excelling in model storage and image quality. Finally, as shown in Tab. 3, our method demonstrates robust quantitative performance on the VRU basketball dataset [40], which involves larger-scale motion.

Qualitative comparisons. We compare scenes from the N3DV dataset and the Meet Room dataset with current

Table 1. **Quantitative comparisons on the Neural 3D Video Dataset [25].** "Size" is the total model size for 300 frames. DSSIM_1 sets data range to 1.0 while DSSIM_2 to 2.0 [26]. $*$ indicates online method.

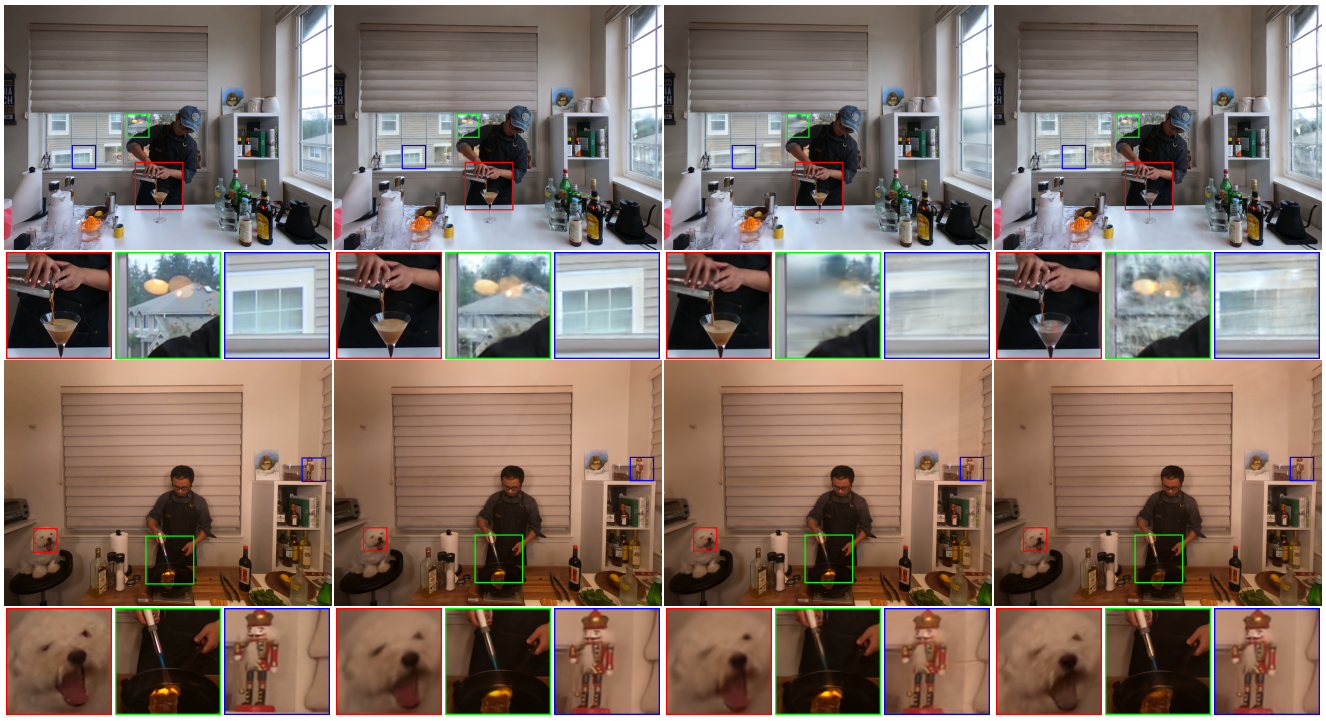
Method	PSNR↑	DSSIM_1↓	DSSIM_2↓	LPIPS↓	FPS↑	Time↓	Size↓
StreamRF * [23]	28.26	-	-	-	10.9	-	5310 MB
NeRFPlayer [38]	30.69	0.034	-	0.111	0.05	6.0 h	5130 MB
HyperReel [1]	31.10	0.036	-	0.096	2	-	360 MB
K-Planes [14]	31.63	-	0.018	-	0.3	5.0 h	311 MB
HexPlane [9]	31.70	-	0.014	0.075	0.21	12.0 h	240 MB
MixVoxels [42]	31.73	-	0.015	0.064	4.6	-	500 MB
4DGaussian [47]	31.02	0.030	-	0.150	30	0.67 h	90 MB
3DGStream[1] [39]	31.67	-	-	-	215	1.0 h	1230 MB
RealTimeGS [55]	32.01	-	0.014	0.055	114	9.0 h	> 1000 MB
SpaceTimeGS [26]	32.05	0.026	0.014	0.044	140	> 5 h	200 MB
LocalDyGS(Ours)	32.28	0.028	0.014	0.043	105	0.58 h	100 MB

Table 2. **Quantitative comparison on the MeetRoom dataset [23].** PSNR is averaged across all 300 frames, while training time and storage requirements accumulate over the entire sequence.

Method	PSNR ↑	Time(hours) ↓	Size(MB) ↓
Plenoxel [13]	27.15	70	304500
I-NGP [31]	28.10	5.5	14460
3DGS [20]	31.31	13	6330
StreamRF [23]	26.72	0.85	2700
3DGStream [39]	30.79	0.6	1230
LocalDyGS(Ours)	**32.45**	**0.36**	**90**

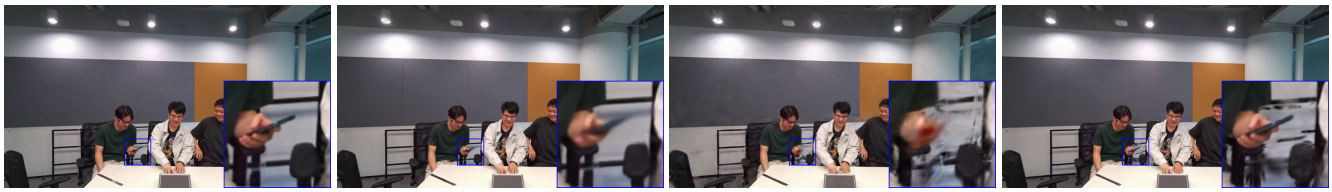
Table 3. **Quantitative comparison on VRU (GZ) basketball court dataset [40].** Static methods are tested on frame 0.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
GOF [58]	30.39	0.949	0.141
2DGS [18]	30.78	0.949	0.187
3DGS [20]	30.50	0.949	0.171
4DGS [47]	28.32	0.930	0.186
SpaceTimeGS [26]	27.42	0.926	0.193
LocalDyGS(Ours)	**30.58**	**0.944**	**0.173**



(a) GT (b) Ours (c) SpaceTimeGS [26] (d) 3DGStream [39]

Figure 6. Qualitative results of *coffee martini* and *sear steak* from the N3DV dataset [24] (a dataset featuring fine-scale motion). We compare our method with SOTA approaches, including STGS [26] and 3DGStream [39]. Our method produces fewer floaters and preserves more details in the dynamic scene, such as newly appearing objects (e.g., coffee liquid and flame), distant background elements, and the dog's face.



(a) GT (b) Ours (c) 3DGStream [39] (d) 3DGS [20]

Figure 7. Qualitative result on the *discussion* of Meetroom dataset [23] (a dataset featuring sparse views and large textureless regions).

mainstream SOTA methods, including the streaming method 3DGStream [39] and non-streaming methods 4DGS [47] and SpaceTimeGS [26]. As shown in Fig. 6, we particularly highlight the modeling of motion areas in certain scenes, such as hands and claws, as well as complex objects like distant branches and plates. Our method can faithfully capture scene information for both dynamic and complex static objects. Fig. 7 demonstrates the subjective effects in the MeetRoom dataset, where our method outperforms 3DGStream in capturing both dynamic hands and static backgrounds. As shown in Fig. 8, we also compare our method with the non-streaming 4DGS and streaming method 3DGStream on the VRU basketball court dataset, which features larger-scale motion. Our approach provides a more faithful representation of large-scale motion.

Table 4. ASG Ablation study on MeetRoom dataset[23].

Method	PSNR ↑	$SSIM_1$↓	$SSIM_2$↓	Size↓
w/o ASG	31.81	0.021	0.011	68 MB
w/ ASG	33.02	0.019	0.009	96 MB

Table 5. **Ablation study of proposed components.** Conducted on the N3DV dataset [24].

Method	*Coffee*	*Sear Steak*	Mean ↑	FPS ↑	Time ↓
w/o static	26.24	32.68	29.46	96	42 mins
w/o deactivation	29.20	33.18	31.19	89	40 mins
Full	29.03	33.77	31.40	105	36 mins

(a) GT (b) Ours (c) 4DGS[47] (d) 3DGStream[39]

Figure 8. Qualitative results on *VRU GZ* [40] (a dataset featuring large-scale, complex motion). Compared to current SOTA dynamic methods, our approach is particularly effective at adapting to large-scale, complex motion scenes. More results can be seen in our videos.

Table 6. Ablation study on the number of frames whose SfM point clouds are used in initialization, conducted on N3DV.

N Frames	PSNR↑	$DSSIM_1$↓	LPIPS↓	Time↓
N = 30	32.30	0.028	0.043	0.75 h
N = 6	32.28	0.028	0.043	0.58 h
N = 1	31.84	0.041	0.058	0.52 h

Table 7. Ablation study on different values of k (N3DV dataset).

k value	PSNR↑	DSSIM↓	Time↓	FPS↑
k = 5	32.15	0.028	31.2 m	118
k = 10	32.28	0.028	34.8 m	105
k = 20	31.96	0.032	40.5 m	86

4.4. Ablation Studies

Decoupling of dynamic-static feature. To validate our static-dynamic feature decoupling approach, we remove the static feature and retain only the dynamic residual feature for training. As shown in Fig. 9 and Tab. 5, removing the static feature leads to a noticeable decline in rendering quality. We hypothesize that this limitation arises from the dynamic residual field's inability to encode the full scene information, leading to noticeable blurring and distortion effects.

Adaptive seed growing (ASG). We conduct ablation experiments on our ASG (Sec. 3.5) in the *discussion* scene. As shown in Fig. 10 and Tab. 4, the ASG technique demonstrates a clear improvement in the accuracy of dynamic region reconstruction.

The number of frames used for initialization. As shown in Tab. 6, the performance of our method increases with the number of initialization frames. In the experiments, we set $N = 6$ to balance performance and training time.

The deactivation of temporal Gaussians. As shown in Tab. 5, this approach effectively reduces redundant Temporal Gaussians without compromising rendering quality, leading to a substantial improvement in inference speed.

Learning with different k value. We apply different k values in our method as same as ScaffoldGS [30]. The results as shown in Tab. 7.

(a) w/o static feature in training (b) Full

Figure 9. A comparison of (a) to (b) shows that training using only dynamic features leads to significant blurring issues.

(a) w/o ASG (b) w/ ASG

Figure 10. Ablation study conducted on the *discussion* scene.

5. Conclusion

In this paper, we first introduce a method that can adapt not only to fine-scale dynamic scenes but also to large-scale scenes. Specifically, we propose decomposing the 3D space into local space based on seeds. For motion modeling within each local space, we assign a static feature shared across all time steps to represent static information, while a global dynamic residual field provides time-specific dynamic residual features to capture dynamic information at each time step. Finally, these features are combined and decoded to produce time-varying Temporal Gaussians, which serve as the final rendering primitives. Extensive experiments show that LocalDyGS effectively adapts to dynamic scenes across various motion scales, performing well on both large-scale motion datasets, such as the VRU [40], and fine-scale motion datasets, like N3DV and MeetRoom.

Acknowledgment

This work was partially supported by Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology(Grant No. 2024B1212010006), Shenzhen Science and Technology Program-Shenzhen Cultivation of Excellent Scientific and Technological Innovation Talents project(Grant No. RCJC20200714114435057), National Natural Science Foundation of China U21B2012, this work is also financially supported for Outstanding Talents Training Fund in Shenzhen, the Royal Society grants (SIF\R1\231009, IES\R3\223050) and an Amazon Research Award.

References

[1] Benjamin Attal, Jia-Bin Huang, Christian Richardt, Michael Zollhoefer, Johannes Kopf, Matthew O'Toole, and Changil Kim. HyperReel: High-fidelity 6-DoF video with ray-conditioned sampling. *arXiv preprint arXiv:2301.02238*, 2023. 6

[2] AVS. https://www.avs.org.cn/. 2024. 6

[3] Jeongmin Bae, Seoha Kim, Youngsik Yun, Hahyun Lee, Gun Bang, and Youngjung Uh. Per-gaussian embedding-based deformation for deformable 3d gaussian splatting. *arXiv preprint arXiv:2404.03613*, 2024. 3

[4] Aayush Bansal, Minh Vo, Yaser Sheikh, Deva Ramanan, and Srinivasa Narasimhan. 4d visualization of dynamic events from unconstrained multi-view videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5366–5375, 2020. 2

[5] 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

[6] 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.

[7] 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

[8] Matteo Bortolon, Theodore Tsesmelis, Stuart James, Fabio Poiesi, and Alessio Del Bue. 6dgs: 6d pose estimation from a single image and a 3d gaussian splatting model. *arXiv preprint arXiv:2407.15484*, 2024. 2

[9] 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

[10] 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

[11] Yuanxing Duan, Fangyin Wei, Qiyu Dai, Yuhang He, Wenzheng Chen, and Baoquan Chen. 4d-rotor gaussian splatting: towards efficient novel view synthesis for dynamic scenes. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 3

[12] 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*, 2024. 2

[13] 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, 6

[14] 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. 2, 6

[15] 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

[16] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5712–5721, 2021. 2

[17] 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

[18] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. *arXiv preprint arXiv:2403.17888*, 2024. 2, 6

[19] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4220–4230, 2024. 2, 3

[20] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 2, 3, 5, 6, 7

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[22] Agelos Kratimenos, Jiahui Lei, and Kostas Daniilidis. Dynmf: Neural motion factorization for real-time dynamic view synthesis with 3d gaussian splatting. *arXiv preprint arXiv:2312.00112*, 2023. 3

[23] 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. 6, 7

[24] Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, Richard Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5521–5531, 2022. 1, 2, 3, 6, 7

[25] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021. 2, 6

[26] Zhan Li, Zhang Chen, Zhong Li, and Yi Xu. Spacetime gaussian feature splatting for real-time dynamic view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8508–8520, 2024. 2, 3, 4, 6, 7

[27] Youtian Lin, Zuozhuo Dai, Siyu Zhu, and Yao Yao. Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21136–21145, 2024. 3

[28] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019. 2

[29] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. Mixture of volumetric primitives for efficient neural rendering. *ACM Transactions on Graphics (ToG)*, 40(4):1–13, 2021. 5

[30] 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, 4, 5, 8

[31] 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. 4, 6

[32] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 2

[33] 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. 1, 3, 4

[34] 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. 1, 2, 3

[35] Lukas Radl, Michael Steiner, Mathias Parger, Alexander Weinrauch, Bernhard Kerbl, and Markus Steinberger. Stopthepop: Sorted gaussian splatting for view-consistent real-time rendering. *ACM Transactions on Graphics (TOG)*, 43(4):1–17, 2024. 2

[36] 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

[37] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 2, 3

[38] 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. 2, 6

[39] 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. 2, 3, 4, 6, 7, 8

[40] VRU. https://anonymous.4open.science/r/vru-sequence/. 2024. 2, 6, 8

[41] Diwen Wan, Ruijie Lu, and Gang Zeng. Superpoint gaussian splatting for real-time high-fidelity dynamic scene reconstruction. *arXiv preprint arXiv:2406.03697*, 2024. 2, 3

[42] Feng Wang, Sinan Tan, Xinghang Li, Zeyue Tian, and Huaping Liu. Mixed neural voxels for fast multi-view video synthesis. *arXiv preprint arXiv:2212.00190*, 2022. 2, 6

[43] Feng Wang, Sinan Tan, Xinghang Li, Zeyue Tian, Yafei Song, and Huaping Liu. Mixed neural voxels for fast multi-view video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19706–19716, 2023. 2, 6

[44] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021. 2

[45] Qianqian Wang, Vickie Ye, Hang Gao, Weijia Zeng, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. 2024. 3

[46] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3295–3306, 2023. 2

[47] 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*, pages 20310–20320, 2024. 3, 6, 7, 8

[48] Hanfeng Wu, Xingxing Zuo, Stefan Leutenegger, Or Litany, Konrad Schindler, and Shengyu Huang. Dynamic lidar re-simulation using compositional neural fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19988–19998, 2024. 2, 4

[49] Jiahao Wu, Rui Peng, Zhiyan Wang, Lu Xiao, Luyang Tang, Jinbo Yan, Kaiqiang Xiong, and Ronggang Wang. Swift4d: Adaptive divide-and-conquer gaussian splatting for compact and efficient reconstruction of dynamic scene. In *The Thirteenth International Conference on Learning Representations*. 2, 4, 6

[50] Jiahao Wu, Lu Xiao, Rui Peng, Kaiqiang Xiong, and Ronggang Wang. Hdrgs: High dynamic range gaussian splatting. *arXiv preprint arXiv:2408.06543*, 2024. 2

[51] Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, et al. Mars: An instance-aware, modular and realistic simulator for autonomous driving. In *CAAI International Conference on Artificial Intelligence*, pages 3–15. Springer, 2023. 2

[52] Lu Xiao, Jiahao Wu, Zhanke Wang, Guanhua Wu, Runling Liu, Zhiyan Wang, and Ronggang Wang. Multi-view image enhancement inconsistency decoupling guided 3d gaussian splatting. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025. 2

[53] Jinbo Yan, Rui Peng, Zhiyan Wang, Luyang Tang, Jiayu Yang, Jie Liang, Jiahao Wu, and Ronggang Wang. Instant gaussian stream: Fast and generalizable streaming of dynamic scene reconstruction via gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16520–16531, 2025. 3

[54] Zhiwen Yan, Chen Li, and Gim Hee Lee. Nerf-ds: Neural radiance fields for dynamic specular objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8285–8295, 2023. 1, 3

[55] Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023. 3, 6

[56] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024. 3, 4

[57] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024. 2

[58] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient and compact surface reconstruction in unbounded scenes. *arXiv preprint arXiv:2404.10772*, 2024. 2, 6

[59] 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

[60] Boming Zhao, Yuan Li, Ziyu Sun, Lin Zeng, Yujun Shen, Rui Ma, Yinda Zhang, Hujun Bao, and Zhaopeng Cui. Gaussianprediction: Dynamic 3d gaussian prediction for motion extrapolation and free view synthesis. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–12, 2024. 3

[61] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12786–12796, 2022. 2

[62] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In *European Conference on Computer Vision*, pages 145–163. Springer, 2025. 2

[63] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM transactions on graphics (TOG)*, 23(3):600–608, 2004. 2

[64] Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002. 3