

# RelayGS: Reconstructing Dynamic Scenes with Large-Scale and Complex Motions via Relay Gaussians

Qiankun Gao<sup>1, 2</sup>, Yanmin Wu<sup>1</sup>, Chengxiang Wen<sup>1</sup>, Jiarui Meng<sup>1</sup>, Luyang Tang<sup>1, 2, 3</sup>,  
Jie Chen<sup>1, 2</sup>✉, Ronggang Wang<sup>1, 2, 3</sup>, Jian Zhang<sup>1, 2, 3</sup>✉

<sup>1</sup>School of Electronic and Computer Engineering, Peking University    <sup>2</sup>Peng Cheng Laboratory

<sup>3</sup>Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology,  
Peking University Shenzhen Graduate School

## Abstract

*Reconstructing dynamic scenes with large-scale and complex motions remains a significant challenge. Recent techniques like Neural Radiance Fields and 3D Gaussian Splatting (3DGS) have shown promise but still struggle with scenes involving substantial movement. This paper proposes RelayGS, a novel method based on 3DGS, specifically designed to represent and reconstruct highly dynamic scenes. Our RelayGS learns a complete 4D representation with canonical 3D Gaussians and a compact motion field, consisting of three stages. First, we learn a fundamental 3DGS from all frames, ignoring temporal scene variations, and use a learnable mask to separate the highly dynamic foreground from the minimally moving background. Second, we replicate multiple copies of the decoupled foreground Gaussians from the first stage, each corresponding to a temporal segment, and optimize them using pseudo-views constructed from multiple frames within each segment. These Gaussians, termed Relay Gaussians, act as explicit relay nodes, simplifying and breaking down large-scale motion trajectories into smaller, manageable segments. Finally, we jointly learn the scene's temporal motion and refine the canonical Gaussians learned from the first two stages. We conduct thorough experiments on two dynamic scene datasets featuring large and complex motions, where our RelayGS outperforms state-of-the-arts by more than 1 dB in PSNR, and successfully reconstructs real-world basketball game scenes in a much more complete and coherent manner, whereas previous methods usually struggle to capture the complex motion of players.*

## 1. Introduction

Dynamic scene reconstruction plays a pivotal role in a wide range of applications requiring immersive and interactive environments, such as virtual reality, metaverse, and free-viewpoint videos. However, achieving high-fidelity recon-

struction of dynamic scenes with large-scale and complex motions from multi-view videos remains highly challenging.

The recently emerged Gaussian Splatting (3DGS)[14] has advanced 3D reconstruction, enhancing efficiency and quality compared to its predecessor, Neural Radiance Fields (NeRF)[26]. Using Gaussian ellipsoids as explicit 3D primitives, 3DGS achieves real-time 1080p rendering via a rasterized pipeline. Dynamic extensions of 3DGS [5, 9, 11, 15, 21–23, 25, 37, 38] typically combine canonical representations with implicit motion fields, similar to dynamic NeRFs [28–30]. While effective for small-scale motions, these methods face challenges with large, complex motions in real-world scenarios, such as sports events with fast-moving players. The primary limitation stems from the coupling of canonical Gaussian learning with neural motion fields, which complicates optimization. Neural networks not only find it challenging to predict large motions but also tend to overfit the dominant small motions in the scene, limiting their ability to model extensive complex movements.

A key approach to addressing large-scale and complex motion is to decouple the highly dynamic foreground from the minimally moving background. By isolating the foreground, we can better capture complex motion trajectories, while minimizing background interference. MLPs efficiently represent the background's motion dynamics; however, the main challenge lies in modeling large, non-rigid foreground motions, which we tackle by decomposing these trajectories into shorter, simpler segments.

In this paper, we propose **RelayGS** to reconstruct dynamic scenes with large-scale, complex motions from multi-view videos. Our goal is to achieve a complete 4D representation, comprising a set of explicit canonical 3D Gaussians and a compact motion field. The core idea is to simplify complex motion trajectories during the learning of canonical 3D Gaussians, laying the foundation for the subsequent joint learning of the 3D Gaussians and the motion field. Specifically, our method unfolds in three progressive stages:

- **I**) We begin by learning a static initial 3DGS from all frames without considering temporal changes, primarily capturing the shared background. To also represent the foreground, we introduce a *learnable mask* to distinguish high-dynamic foreground Gaussians from low-dynamic background Gaussians. All Gaussians are used to render the first frame, while only those with mask = 1 are used in subsequent frames, yielding a coarse representation of both the background and initial foreground while effectively decoupling the two (see Sec. 4.1 for details).

- **II**) Ideally, each foreground Gaussian would follow a complex motion trajectory over time, but achieving this directly is challenging. Instead, we replicate multiple copies of the decoupled foreground Gaussians from the first stage, each corresponding to a temporal segment. To further optimize and densify these copies, we construct pseudo-views using selected frames from the corresponding segment. These foreground Gaussians, which we term **Relay Gaussians**, serve as explicit, discrete nodes along the idealized motion trajectory, effectively simplifying and approximating the complex, large-scale trajectory by breaking it down into smaller, manageable segments. (see Sec. 4.2 for details). *This can be regarded as temporal densification, analogous to the spatial densification in 3DGS (refer to Sec. 7 in supp.).*

- **III**) Finally, we jointly optimize the canonical Gaussians learned in the previous stages together with a compact motion field to achieve a complete 4D representation. Though our method is not limited to a specific motion field model, we follow 4D-GS [36] in this paper by adopting HexPlane [3] and lightweight MLPs, with several key modifications to better capture large, complex motions. Specifically, for the foreground **Relay Gaussians**, we employ an additional set of MLPs and introduce a learnable scaling factor for position offsets as they may require a larger range that cannot be fully captured by the MLP's predictions alone. These improvements allow the foreground Relay Gaussians to more accurately represent the dynamic and complex motions in the scene. (see Sec. 4.3 for details).

We conducted thorough experiments to validate the effectiveness of our **RelayGS**. On the PanopticSports dataset [12], featuring large-scale motions, our method achieves a **1 dB** PSNR improvement over previous state-of-the-arts. On the more challenging VRU Basketball Games dataset [34], our method reconstructs scenes with greater completeness and coherence, where prior methods struggle to capture the dynamic foreground content with complex motions. The contributions of this paper are summarized as follows:

- We introduce a simple learnable mask that effectively decouples high dynamic foreground and low dynamic background Gaussians without relying on additional priors, while learning a more accurate and complete fundamental 3DGS representation of the dynamic scene.

- We propose the temporal Relay Gaussians to decompose

large-scale and complex motion trajectories into smaller, more manageable motion segments, simplifying the representation and learning of complex dynamics.

- We utilize distinct MLPs to predict motion changes for background Gaussians and foreground Relay Gaussians, along with a learnable scaling factor for the position changes of Relay Gaussians, enabling accurate capture of larger and more complex motions.

- We conduct experiments on real-world dynamic scene datasets featuring large-scale, complex motions, where our RelayGS significantly outperforms previous state-of-the-art methods, achieving a **1 dB** improvement in PSNR on PanopticSports dataset and delivering more complete and coherent reconstructions of complex, large-scale foreground motions.

## 2. Related Work

**Dynamic Scene Modeling.** Early methods [3, 7, 18, 28–30, 32] based on Neural Radiance Fields (NeRF) model deformation fields to map canonical spaces to dynamic frames but suffer from high computational costs due to dense sampling. Recent approaches have shifted toward the more efficient 3D Gaussian Splatting (3DGS). A straightforward strategy expands Gaussian primitives to 4D, as seen in 4DGS [38] and Rotor-4DGS [6]. Other methods decouple the dynamic scene into a canonical 3DGS and a temporal motion field. Deformable3DGS [37] uses deep MLPs to predict Gaussian motion, while 4D-GS [36] enhances this framework with multi-resolution HexPlane and lightweight MLPs for improved efficiency. SC-GS [11] assumes motion is driven by key points, predicting time-varying transformations through a deformation MLP and interpolating them to generate a coherent motion field. Additionally, online frame-by-frame learning methods incrementally model dynamic changes. Dynamic3DGS [24] updates Gaussian positions and rotations at each timestamp, while 3DGStream [33] employs a NGP [27] to manage transformations efficiently. HiCoM [8] leverages the non-uniform distribution and local consistency to enable fast and accurate motion learning across frames.

**Dynamic-Static Decoupling.** S4D [10], EgoGaussian [39], and SC-4DGS [17] use pre-trained segmentation models to generate 2D motion masks for separating dynamic and static content. Compact-D3DGS [13] and GauFRe [20] rely on optical flow. In contrast, our method uses a learnable mask to decouple high-dynamic foreground from low-dynamic background, eliminating reliance on pre-trained motion priors and enhancing adaptability to complex motions.

## 3. Preliminaries

**3D Gaussian Splatting.** 3D Gaussian Splatting [14] explicitly represents scenes using anisotropic 3D Gaussian primitives, mathematically formulated as:

$$\mathcal{G}(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}, \quad \boldsymbol{\Sigma} = \mathbf{R} \mathbf{S} \mathbf{S}^T \mathbf{R}^T, \quad (1)$$

where the mean vector  $\mu$  and covariance matrix  $\Sigma$  respectively characterize the central position and geometric shape. The matrix  $\Sigma$  is decomposed into a scaling matrix  $\mathbf{S} = \text{diag}(s_x, s_y, s_z)$  and a rotation matrix  $\mathbf{R} \in SO(3)$ , further simplified as a vector  $\mathbf{s} \in \mathbb{R}^3$  and a quaternion  $\mathbf{q} \in \mathbb{R}^4$ , to ensure physical meaning and facilitate optimization. Each Gaussian is associated with an opacity  $o$  and spherical harmonics  $\mathbf{h}$  representing color.

Rendering is performed by blending the contributions of  $N$  overlapping Gaussian primitives at each pixel, taking into account their depth-ordering to ensure correct compositing, expressed as:

$$C = \sum_{i \in N} \mathbf{c}_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (2)$$

where  $\mathbf{c}_i, \alpha_i$  represents the color and blending weight of the  $i^{th}$  Gaussian, respectively. The training alternates between parameter optimization and density control. Parameter optimization is supervised by the  $\mathcal{L}_1$  loss and D-SSIM term:

$$\mathcal{L} = (1 - \lambda) \mathcal{L}_1 + \lambda \mathcal{L}_{\text{D-SSIM}}. \quad (3)$$

**4D Gaussian Splatting.** 4D-GS [36] extends 3D Gaussian Splatting (3DGS) by incorporating a deformation field to model dynamic scenes. The deformation field is implemented through a HexPlane [3] encoding module  $\mathcal{H}$  and a set of lightweight MLPs. Based on the Gaussian center position  $\mu = (x, y, z)$  and a given time  $t$ ,  $\mathcal{H}$  outputs a feature encoding  $\mathbf{f}$ , which is then fed into separate MLPs to predict changes in Gaussian attributes such as position  $\mu$ , scale  $\mathbf{s}$ , rotation  $\mathbf{r}$ , and opacity  $o$ , represented as follows:

$$\begin{aligned} \Delta\mu &= \phi_\mu(\mathbf{f}), & \Delta\mathbf{r} &= \phi_\mathbf{r}(\mathbf{f}), \\ \Delta\mathbf{s} &= \phi_\mathbf{s}(\mathbf{f}), & \Delta\mathbf{o} &= \phi_\mathbf{o}(\mathbf{f}). \end{aligned} \quad (4)$$

The deformed 3D Gaussian is expressed as:

$$\mathcal{G}' = \{\mu + \Delta\mu, \mathbf{s} + \Delta\mathbf{s}, \mathbf{r} + \Delta\mathbf{r}, o + \Delta o, \mathbf{h}\}. \quad (5)$$

At time  $t$ , the 3D Gaussian  $\mathcal{G}$  in the scene will be replaced by the deformed 3D Gaussian  $\mathcal{G}'$  for rendering.

The training process consists of two stages. The first stage serves as a warm-up, optimizing static scenes using only 3D Gaussians. In the second stage, the HexPlane, MLPs, and 3D Gaussians are jointly optimized. The loss function includes an  $\mathcal{L}_1$  loss and a grid-based total variation loss  $\mathcal{L}_{tv}$ :

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_{tv}. \quad (6)$$

## 4. Methodology

The proposed method, **RelayGS**, is designed to effectively tackle the challenge of reconstructing dynamic scenes with large-scale and complex motions. The goal, similar to our

baseline method 4D-GS [36], is to achieve a complete 4D representation consisting of a set of explicit canonical 3D Gaussians and a compact motion field. Our core idea is to simplify complex motion trajectories during the learning of canonical 3D Gaussians (detailed in Secs. 4.1 and 4.2), thereby laying a strong foundation for the subsequent joint learning of canonical 3D Gaussians and the motion field (detailed in Sec. 4.3), as illustrated in Fig. 1. While our method is not limited to a specific motion field, we adopt HexPlane and lightweight MLPs, following 4D-GS, with several improvements to better accommodate complex motions. We introduce the three progressive stages in detail below.

### 4.1. Stage 1: Initial Representation and Foreground-Background Decoupling

The primary goal of this first stage is to construct the fundamental 3D structure of the dynamic scene. Previous method [36] initialize a set of static Gaussians from sparse point clouds and jointly optimize them using all given frames without considering temporal scene changes, *i.e.*, treating it as a static scene for initialization. This approach effectively captures the relatively static background of the scene, but struggles with the highly dynamic foreground.

The highly dynamic foreground, due to its significant positional variations across frames, cannot be easily initialized. For instance, even if some Gaussians can model dynamic foreground objects in a specific frame, due to the large motion of the objects, they may cause inconsistencies in another frame, resulting in large rendering errors. Under this initialization paradigm, the Gaussians representing such foreground objects would be noisy or automatically pruned.

To address this limitation and learn the highly dynamic foreground simultaneously, we introduce a “*learnable mask*” for each Gaussian primitive to indicate whether it belongs to the highly dynamic foreground or the relatively static background. The implementation of this mask follows the straight-through estimator [2], a technique widely adopted in previous works [4, 16] to assess the importance of each Gaussian primitive for rendering quality in static scenes, enabling effective pruning and compression to reduce storage overhead. However, we are the first to leverage this approach in the context of dynamic scene reconstruction, using it to distinguish between foreground and background Gaussians. The formulation is expressed as:

$$\begin{aligned} \mathbf{M}_n &= \text{sg}(\mathbf{1}[\sigma(\mathbf{m}_n) > \epsilon] - \sigma(\mathbf{m}_n)) + \sigma(\mathbf{m}_n) \\ &= \begin{cases} 1, & \text{if } \sigma(\mathbf{m}_n) > \epsilon \\ 0, & \text{otherwise} \end{cases}, \end{aligned} \quad (7)$$

where  $n$  is the index among all  $N$  Gaussians,  $\epsilon$  is the masking threshold,  $\mathbf{m} \in \mathbb{R}^N$  is the learnable mask parameter,  $\mathbf{M} \in \{0, 1\}^N$  is the generated binary masks,  $\text{sg}(\cdot)$  is the stop gradient operator, and  $\mathbf{1}[\cdot]$  and  $\sigma(\cdot)$  are indicator and

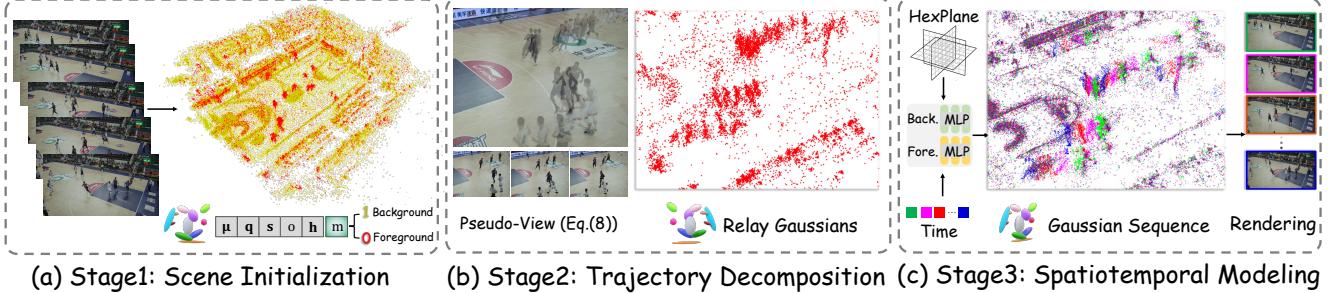


Figure 1. Framework of the proposed RelayGS. (a) Initialize the scene with all images and separate the relatively static background and dynamic foreground using a learnable mask (visualized as yellow and red). (b) Construct pseudo-GT views through multi-view blending to optimize Relay Gaussians for decomposing complex trajectories. (c) Based on the HexPlane 4D representation, using different MLPs for foreground and background Gaussians to obtain temporal deformation, and then render through the differentiable pipeline of 3DGS.

sigmoid function, respectively. It is important to note that, although  $\mathbf{M}_n$  is binary, gradients can still be backpropagated to  $\mathbf{m}_n$ , allowing for optimization through gradient descent.

We use all Gaussians to render views for the first frame, ignoring the mask. However, when rendering other frames, we replace each Gaussian’s opacity with the following:

$$\hat{\mathbf{o}}_n = \mathbf{M}_n \mathbf{o}_n, \quad (8)$$

where,  $\mathbf{o}_n$  and  $\hat{\mathbf{o}}_n$  are the opacity before and after applying the mask, respectively. The Gaussians representing highly dynamic foreground in the first frame incur a larger loss in other frames due to their movement, resulting in higher gradients that progressively decrease  $\hat{\alpha}_n$ . To preserve a high  $\alpha_n$  value for the first frame,  $\mathbf{M}_n$  is optimized toward 0.

In this way, we can effectively decouple the canonical Gaussians into two groups, as shown in Fig. 1(a), allowing the separation of the highly dynamic foreground (red points) from the background (yellow points) with minimal motion.

This initial stage not only allows us to learn a better foundational scene representation compared to prior methods like 4D-GS, but the foreground-background decoupling also plays a significant role in subsequent stages, as detailed in the following sections.

#### 4.2. Stage 2: Large Motion Trajectory Decomposition by Relay Gaussians

Ideally, each foreground Gaussian would follow a large-scale and complex motion trajectory over time, but achieving this directly is challenging. To address this, we replicate multiple copies of the decoupled foreground Gaussians from the first stage, each copy corresponding to a specific temporal segment. In our implementation, consecutive  $k$  (*e.g.*, 16) frames are treated as one segment, *i.e.*, the  $1^{st}$ - $k^{th}$  frames form the first segment, followed by subsequent segments.

Since motion trajectories are continuous over time, these copies, once optimize to the right positions, will break down the large motion trajectory into smaller segments, each representing a portion of the overall motion trajectory. We term

them as **Relay Gaussians** since they serve as explicit relay nodes along the large-scale motion trajectory.

To optimize and densify more Relay Gaussians, we construct pseudo-views by blending  $p = 3$  uniformly selected frames (*e.g.*, frames 1, 8, and 16 in the first segment) for supervision. Let the three selected frames in the corresponding segment be denoted as  $I_{t_1}, I_{t_2}, I_{t_3}$ . The pseudo-view  $I_{\text{pseudo}}$  for Relay Gaussians is then constructed as:

$$I_{\text{pseudo}} = \beta_1 I_{t_1} + \beta_2 I_{t_2} + \beta_3 I_{t_3}, \quad (9)$$

where  $\beta_1 + \beta_2 + \beta_3 = 1$  are blending weights applied to the selected frames, typically chosen based on frame importance or uniform blending. In this work, we use the straightforward uniform blending, *i.e.*,  $\beta_1 = \beta_2 = \beta_3 = \frac{1}{3}$ , for conciseness. These pseudo-views capture snapshots of the foreground at different time steps, as shown in Fig. 1 (b), providing a richer representation for optimizing the Relay Gaussians, ensuring they more accurately capture the motion trajectory within each segment.

By leveraging Relay Gaussians to decompose large-scale motion trajectories into smaller, more manageable segments, we reduce the complexity of handling dynamic motions, which will become evident in the final learning stage. In Sec. 7 of supplementary document, we analyze this stage as a process of temporal densification.

#### 4.3. Stage 3: 4D Spatiotemporal Modeliing and Optimization

After the previous two stages, we have established more refined canonical 3D Gaussians. To achieve a complete 4D representation, it is essential to incorporate temporal variation through an motion field. Although our method is not limited to a specific motion field model, in this work, we adopt the HexPlane and MLPs from 4D-GS due to their efficiency and flexibility in spatiotemporal encoding. However, we have made the following improvements to better capture large and complex motions.

**Foreground-background isolation.** To avoid overfitting to small motions due to all Gaussians sharing MLPs, we propose a divide-and-conquer strategy. For the background Gaussians, we utilize a dedicated set of MLPs that predict the temporal changes in their positions and other attributes. For the foreground Relay Gaussians, another set of MLPs models their time-varying positions and attributes throughout the motion trajectory, as shown in Fig. 1 (c).

**Position deformation scaling.** To enhance the ability to capture complex motion patterns, we introduce a learnable scaling factor  $\gamma \in \mathbb{R}^3$  for each Relay Gaussian. This factor adjusts the predicted position deformations, allowing the model to accommodate larger motion ranges that the MLP alone may not fully capture. This addition ensures that Relay Gaussians can adapt flexibly to intricate motions beyond the standard MLP predictions.

$$\mu \leftarrow \mu + (1 + e^\gamma) \cdot \Delta\mu. \quad (10)$$

By jointly optimizing the canonical Gaussians and our improved motion field, we achieve a comprehensive 4D scene reconstruction that integrates both spatial and temporal dynamics, ultimately yielding a coherent and precise representation of the entire dynamic scene.

## 5. Experiment

### 5.1. Experimental Setup

In this work, we primarily focus on addressing large-scale and complex motion in dynamic scenes. We conduct experiments on the following two representative datasets:

**PanopticSports Dataset** is a subset of the CMU Panoptic Studio dataset [12], containing 6 dynamic sports scenes: Joggle, Box, Softball, Tennis, Football and Basketball. Each scene has a resolution of  $640 \times 360$  and spans 150 frames, captured at 30 FPS. The data was collected using 31 static cameras, of which 27 are used for training and 4 for testing (cameras 0, 10, 15, and 30).

**VRU Basketball Games Dataset** [34] contains two real-world basketball game scenes, “GZ” and “DG4”. Each was captured in an indoor basketball court using 34 fixed, synchronized cameras, evenly distributed around the court to cover 360 degrees. The sequences span 10 seconds, with a resolution of  $1920 \times 1080$  at 25 FPS, resulting in 250 frames per sequence. Of the 34 cameras, 30 are used for training, while 4 (cameras 0, 10, 20, and 30) are reserved for testing. More details of datasets can be found in the Appendix.

**Implementation.** Our implementation is based on the open-source 4D-GS [36] code. In the first stage, 3D Gaussians are initialized using sparse point clouds derived from the initial frame, following 3DGS [14] and 4D-GS. Each Gaussian is assigned a learnable mask attribute initialized to 2, resulting in values near 1 after sigmoid activation. Optimization runs for 3,000 steps with periodic densification.

In the second stage, we set  $k = 16$  and train for 14,000 steps. In the third stage, we initialize HexPlane and MLPs following 4D-GS, with the difference that we configure two separate MLP sets: one for background Gaussians and the other for Relay Gaussians. Each set of MLPs is responsible for predicting the temporal changes in the four Gaussian attributes—position, scaling, rotation, and opacity. We exclude the spherical harmonics MLP, as it increases model size and reduces rendering speed without significant performance gains. Additionally, the  $\gamma$  is initialized to 0. This last stage is trained for 20,000 steps. For the PanopticSports dataset, multi-view color inconsistencies are present, so we apply a learnable channel-wise affine color tune for each camera, following Dynamic3DGS [24]. For VRU scenes, we use  $2 \times$  downsampled views during the first two stages to reduce computational time. All experiments were conducted on an NVIDIA RTX 4090 GPU with batch size 4. The learning rate and densification settings are consistent across all three stages, more details can be found in the Appendix.

### 5.2. Experimental Results

**Quantitative Comparison.** We compare our RelayGS with several state-of-the-art methods, including 4D-GS [36], Dynamic3DGS [24], ST-GS [19], E-D3DGS [1], and D-MiSo [35]. The results are shown in Tab. 1 and Tab. 2. (1)

**Quality:** Our RelayGS method consistently outperforms competitors in terms of reconstruction quality (*i.e.*, PSNR) on both datasets. Specifically, on the six scenes of the PanopticSports dataset (see Tab. 2), RelayGS achieves PSNR improvements of 0.27 dB, 0.53 dB, 1.6 dB, 1.19 dB, 1.24 dB, and 1.28 dB, respectively, averaging a gain of 1.02 dB over the previous best methods. Compared to the baseline method 4D-GS, we achieve an average performance gain of 2.47 dB. On the more challenging VRU Basketball Games dataset (see Tab. 1), RelayGS outperforms the previous best method ST-GS and the baseline method 4D-GS by an average of 0.45 dB and 2 dB, respectively. It is worth *noting* that, although the PSNR difference compared with ST-GS appears small, the static floor occupies approximately 70% of the pixels in these VRU view images, meaning the quality improvement is more significant in the dynamic foreground regions. Additionally, ST-GS is heavily dependent on initialization, as it extracts sparse point clouds for each frame and then merges them as the initial scene. Since point clouds for each frame cannot be obtained in the PanopticSports dataset, ST-GS is not applicable.

**(2) Efficiency:** While our method learns corresponding foreground content for each segment via Relay Gaussians, RelayGS strikes a good balance between reconstruction quality and efficiency factors such as storage, training time, and rendering speed compared to competitors, some of which achieve high storage efficiency but fall short in reconstruction quality. In contrast, our method demonstrates a clear advantage in storage efficiency, particularly

Table 1. Quantitative results on the VRU Basketball Games dataset. Our RelayGS and other methods only use the point clouds derived from the initial frame. “ST-GS<sup>16</sup>” uses point clouds of uniformly selected 16 frames, while “ST-GS<sup>250</sup>” utilizes point clouds of all 250 frames, the default setting for their method. Notably, “ST-GS” fails to perform effectively when restricted to the same point clouds as ours.

Method	GZ				DG4			
	PSNR (dB ↑)	Storage (MB ↓)	Train (mins ↓)	Render (fps ↑)	PSNR (dB ↑)	Storage (MB ↓)	Train (mins ↓)	Render (fps ↑)
ST-GS <sup>16</sup> [19]	26.49	35	64	264	25.79	40	64	236
ST-GS <sup>250</sup> [19]	27.32	400	107	143	26.79	360	112	134
4D-GS [36]	25.83	42	63	88	25.17	45	62	80
E-D3DGS [1]	26.14	113	224	35	25.06	136	301	27
<b>RelayGS (Ours)</b>	<b>28.06</b>	200	105	74	<b>26.94</b>	191	107	69

Table 2. Quantitative results on the PanopticSports dataset. “Dynamic3DGS” and “D-MiSo” data are partially taken directly from their original papers or estimated based on the paper and available code. “Dynamic3DGS” is a frame-by-frame learning method, whereas our RelayGS and other methods learn from all frames jointly.

Method	Juggle			Boxes			Softball		
	PSNR (dB ↑)	Storage (MB ↓)	Train (mins ↓)	PSNR (dB ↑)	Storage (MB ↓)	Train (mins ↓)	PSNR (dB ↑)	Storage (MB ↓)	Train (mins ↓)
Dynamic3DGS [24]	29.48	221	107	29.46	221	108	28.43	221	116
4D-GS [36]	28.19	48	30	27.67	47	29	27.41	46	29
E-D3DGS [1]	26.54	36	95	26.78	33	100	26.01	33	80
D-MiSo [35]	29.79	-	-	29.39	-	-	28.60	-	-
<b>RelayGS (Ours)</b>	<b>30.06</b>	31	48	<b>29.99</b>	30	48	<b>30.20</b>	33	48
	Tennis			Football			Basketball		
	PSNR (dB ↑)	Storage (MB ↓)	Train (mins ↓)	PSNR (dB ↑)	Storage (MB ↓)	Train (mins ↓)	PSNR (dB ↑)	Storage (MB ↓)	Train (mins ↓)
Dynamic3DGS [24]	28.11	221	101	28.49	221	114	28.22	221	113
4D-GS [36]	27.49	45	29	26.67	54	33	27.72	37	24
E-D3DGS [1]	27.41	31	74	25.93	33	76	26.48	35	87
D-MiSo [35]	29.02	-	-	28.99	-	-	28.49	-	-
<b>RelayGS (Ours)</b>	<b>30.21</b>	31	48	<b>30.23</b>	37	48	<b>29.77</b>	51	48

on the PanopticSports dataset. Compared to the baseline 4D-GS, RelayGS introduces an additional stage with Relay Gaussians, which increases the training time and slightly reduces the rendering speed in some tend. However, RelayGS still maintains a clear advantage in training time compared to other methods. While achieving high-quality reconstruction, we can also ensure a real-time rendering speed of around 70 fps on RTX 4090 GPU.

**Qualitative Analysis** Fig. 2 and Fig. 3 show frames from two representative scenes with heavily featured foreground dynamic content. As seen, our RelayGS reconstructs the humans with greater clarity and completeness. This improvement is primarily due to the fact that, compared to our

baseline, 4D-GS, our stage I not only learns the background Gaussians but also captures the foreground Gaussians. In our stage II, we further refine the foreground Gaussians by learning additional Gaussians that cover more of the motion trajectories, known as Relay Gaussians. ST-GS, although using point clouds from all 250 frames, obtains a denser sampling of motion trajectories. However, due to its simpler approach to modeling motion changes, it struggles to accurately capture the foreground with complex motions. This issue is more evident in the rendered videos, where ST-GS shows inconsistencies in the motion of the Gaussians associated with the same object, leading to flickering in the foreground. In contrast, our method, leveraging HexPlane encoding following 4D-GS, models temporally and spatially



Figure 2. Qualitative comparisons on GZ scene of VRU Basketball Games dataset.

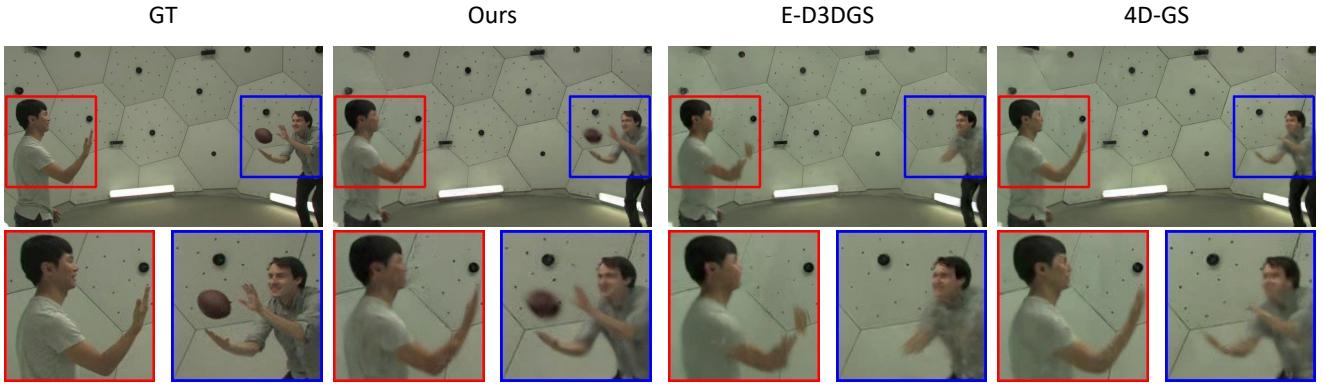


Figure 3. Qualitative comparisons on Football scene of PanopticSports dataset.

consistent motion, resulting in smoother and more coherent reconstructions. Additionally, both 4D-GS and E-D3DGS struggle to handle the large-scale motion of the ball in these scenes. In comparison, our method performs significantly better, although challenges remain. The relatively small and isolated ball with mostly empty space around it makes it difficult to track. Our second stage mitigates this issue to some extent by introducing Relay Gaussians, but it remains a challenging aspect due to the sparse Gaussians learned in the first stage. In summary, RelayGS not only achieves SOTA performance on quantitative metrics but also demonstrates superior spatiotemporal modeling capabilities, particularly on foreground dynamic content. *We encourage readers to view the supplementary rendered videos for a more comprehensive understanding of reconstruction results.*

**3D Gaussian visualization.** We visualize the canonical Gaussians learned at different stages, with the results shown in Fig. 4. As observed in Fig. 4 (b), in the baseline method 4D-GS, the canonical Gaussians learned in the first stage primarily represent the background, with very few Gaussians capturing the foreground. In contrast, in our method, the base Gaussians learned in the first stage include both background and foreground Gaussians, which can be dis-

tinguished by a binary mask, visualized in different colors in Fig. 4 (c). Furthermore, through the learning process in the second stage, our method is able to capture additional Relay Gaussians (red points in Fig. 4 (d)) along the motion trajectories of the foreground, significantly improving the representation of dynamic content.

### 5.3. Ablation Study

In Tab. 3, we present ablation studies on several key components of our method. The case #2 represents the configuration where no foreground Gaussians copies is applied, and only a single global set of foreground Gaussians is used. This results in a significant performance drop, as it cannot effectively handle large-scale motion. In case #3, we remove the second stage of our method, directly replicating a set of foreground Gaussians for each segment and learning them jointly with the implicit motion field. This also leads to a notable performance decrease, especially in the more complex GZ scene. In case #4, we demonstrate the significance of multi-view synthesis pseudo-views, which enable the acquisition of richer Relay Gaussians representing trajectories. In cases #5 and #6, we conduct ablation studies on the setting of different MLPs for foreground-background isolation and the scaling factor  $\gamma$  in the third stage, respectively. These

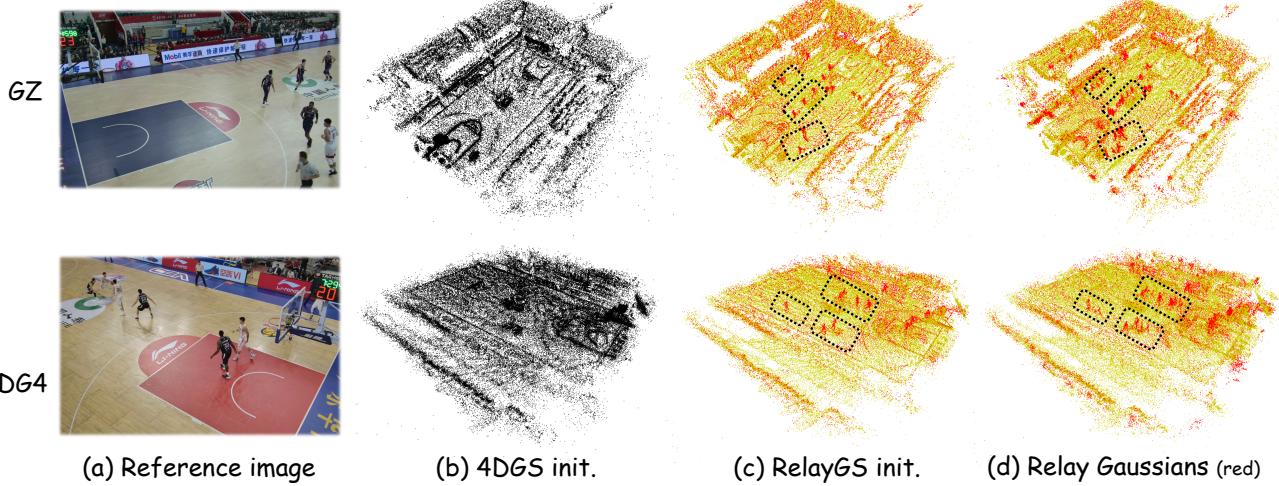


Figure 4. The visualization of canonical 3D Gaussians. (a) Reference image of the scene. (b) Initialization by 4D-GS, with the foreground Gaussian almost eliminated. (c) Initialization by our method achieves separation of background and foreground, visualized in different colors. (d) Relay Gaussians (red) generated in the second stage realize the decomposition of large-scale complex trajectories.

Table 3. Ablation study on key design components. For detailed analysis, please refer to Sec. 5.3.

Case	GZ	Softball
#1 full method	28.06	30.20
#2 w/o Fg Gaussian Copies	26.07 ↓1.99	29.42 ↓0.78
#3 w/o Stage II	27.27 ↓0.79	29.93 ↓0.27
#4 w/o Pseudo-Views	27.80 ↓0.26	30.00 ↓0.20
#5 w/o Fg-Bg Isolation	27.80 ↓0.26	30.07 ↓0.13
#6 w/o Scaling Factor $\gamma$	27.87 ↓0.19	29.73 ↓0.47

Table 4. Ablation on number of frames per segment.

$k$	8	16	32	64	128
PSNR (dB)	27.90	28.06	27.82	27.56	27.10

results highlight the importance of our improvements for 4D spatiotemporal modeling.

In Tab. 4, we perform an ablation study on the length of each segment, *i.e.*, the number of frames included in each segment. As the segment length increases and the number of segments decreases, the motion trajectory within each segment becomes larger, leading to a gradual decline in performance. However, choosing the  $k$  value too small will increase the training cost and not result in a significant performance improvement. Based on experience, we set  $k=16$  as the default selection.

## 6. Conclusion

This paper introduces RelayGS to tackle the challenges of reconstructing dynamic scenes with large-scale and complex motions. We first learn the basic scene structure and, through

a learnable mask, simultaneously capture the shared low-dynamic background and high-dynamic foreground, achieving effective decoupling of foreground and background Gaussians. Then, we replicate the foreground Gaussians and train them with pseudo-views constructed by blending frames within corresponding temporal segments. These foreground Gaussians are referred to as Relay Gaussians, which decompose the complex, large-scale motion trajectories into smaller, manageable segments. Finally, we jointly optimize a compact motion field and the canonical Gaussians to learn a comprehensive 4D representation of the dynamic scene. Experiments on two real-world datasets demonstrate that RelayGS achieves state-of-the-art reconstruction quality for large-scale motions while maintaining a balance between reconstruction fidelity and storage efficiency, making it practical for real-world dynamic scene applications.

**Limitations.** While our method achieves significant performance advantages, it still faces some known challenges. (1) Insufficient motion modeling of small but fast-moving objects due to the limited pixel coverage of these objects, insufficient camera view coverage, and sparse canonical surrounding Gaussians. (2) Our temporal segmentation and pseudo-view construction strategies are relatively straightforward. In the future, we plan to explore more adaptive temporal segmentation methods that align with the motion complexity of the scene. Moreover, we aim to develop more sophisticated frame selection strategies that better capture motion dynamics, enabling Relay Gaussians to more closely follow the ideal motion trajectory and improve the accuracy of motion representation. (3) Our method is tailored for multi-view inputs from stationary cameras and may not be suitable for settings with monocular videos or those involving moving cameras.

## References

- [1] Jeongmin Bae, Seoha Kim, Youngsik Yun, Hahyun Lee, Gun Bang, and Youngjung Uh. Per-gaussian embedding-based deformation for deformable 3d gaussian splatting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 5, 6, 2
- [2] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons. *arXiv preprint arXiv:1308.3432*, 2013. 3
- [3] 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 (CVPR)*, 2023. 2, 3
- [4] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Hac: Hash-grid assisted context for 3d gaussian splatting compression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 3
- [5] Gang Zeng Diwen Wan, Ruijie Lu. Superpoint gaussian splatting for real-time high-fidelity dynamic scene reconstruction. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024. 1
- [6] 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*, 2024. 2
- [7] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Walborg, 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 (CVPR)*, 2023. 2
- [8] Qiankun Gao, Jiarui Meng, Chengxiang Wen, Jie Chen, and Jian Zhang. Hicom: Hierarchical coherent motion for dynamic streamable scenes with 3d gaussian splatting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2
- [9] Zhiyang Guo, Wengang Zhou, Li Li, Min Wang, and Houqiang Li. Motion-aware 3d gaussian splatting for efficient dynamic scene reconstruction. *arXiv preprint arXiv:2403.11447*, 2024. 1
- [10] Bing He, Yunuo Chen, Guo Lu, Li Song, and Wenjun Zhang. S4d: Streaming 4d real-world reconstruction with gaussians and 3d control points. *arXiv preprint arXiv:2408.13036*, 2024. 2
- [11] 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 (CVPR)*, 2024. 1, 2
- [12] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE international conference on computer vision*, 2015. 2, 5
- [13] Kai Katsumata, Duc Minh Vo, and Hideki Nakayama. A compact dynamic 3d gaussian representation for real-time dynamic view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 2
- [14] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 2023. 1, 2, 5
- [15] Agelos Kratimenos, Jiahui Lei, and Kostas Daniilidis. Dynmf: Neural motion factorization for real-time dynamic view synthesis with 3d gaussian splatting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 1
- [16] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian representation for radiance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3
- [17] Fang Li, Hao Zhang, and Narendra Ahuja. Self-calibrating 4d novel view synthesis from monocular videos using gaussian splatting. *arXiv preprint arXiv:2406.01042*, 2024. 2
- [18] 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 (CVPR)*, 2022. 2
- [19] 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 (CVPR)*, 2024. 5, 6, 2, 3
- [20] Yiqing Liang, Numair Khan, Zhengqin Li, Thu Nguyen-Phuoc, Douglas Lanman, James Tompkin, and Lei Xiao. Gaufre: Gaussian deformation fields for real-time dynamic novel view synthesis. *arXiv preprint arXiv:2312.11458*, 2023. 2
- [21] 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 (CVPR)*, 2024. 1
- [22] Qingming Liu, Yuan Liu, Jiepeng Wang, Xianqiang Lv, Peng Wang, Wenping Wang, and Junhui Hou. Modgs: Dynamic gaussian splatting from causally-captured monocular videos. *arXiv preprint arXiv:2406.00434*, 2024.
- [23] Zhicheng Lu, Xiang Guo, Le Hui, Tianrui Chen, Min Yang, Xiao Tang, Feng Zhu, and Yuchao Dai. 3d geometry-aware deformable gaussian splatting for dynamic view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1
- [24] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *International Conference on 3D Vision (3DV)*, 2024. 2, 5, 6
- [25] Marko Mihajlovic, Sergey Prokudin, Siyu Tang, Robert Maier, Federica Bogo, Tony Tung, and Edmond Boyer. Splatfields: Neural gaussian splats for sparse 3d and 4d reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. 1
- [26] 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. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1

- [27] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 2022. [2](#)
- [28] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. [1](#), [2](#)
- [29] 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. *ACM Transactions on Graphics (TOG)*, 2021.
- [30] 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 (CVPR)*, 2021. [1](#), [2](#)
- [31] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [2](#)
- [32] 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 (CVPR)*, 2023. [2](#)
- [33] 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 (CVPR)*, 2024. [2](#)
- [34] VRU. Vru-sequence, 2024. <https://anonymous.4open.science/r/VRU-Sequence/>. [2](#), [5](#)
- [35] Joanna Waczyńska, Piotr Borycki, Joanna Kaleta, Sławomir Tadeja, and Przemysław Spurek. D-miso: Editing dynamic 3d scenes using multi-gaussians soup. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2024. [5](#), [6](#)
- [36] 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)*, 2024. [2](#), [3](#), [5](#), [6](#), [1](#)
- [37] 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 (CVPR)*, 2024. [1](#), [2](#)
- [38] 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](#), [2](#)
- [39] Daiwei Zhang, Gengyan Li, Jiajie Li, Mickaël Bressieux, Otmar Hilliges, Marc Pollefeys, Luc Van Gool, and Xi Wang. Egogaussian: Dynamic scene understanding from egocentric video with 3d gaussian splatting. In *International Conference on 3D Vision (3DV)*, 2025. [2](#)

# RelayGS: Reconstructing Dynamic Scenes with Large-Scale and Complex Motions via Relay Gaussians

## Supplementary Material

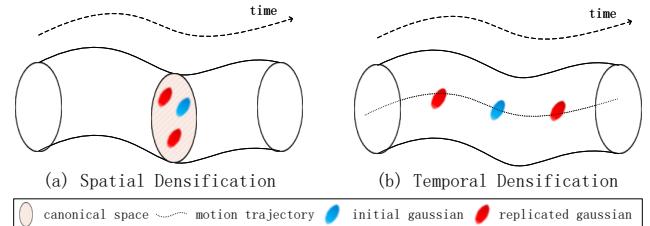
This supplementary document provides additional insights and details to support our main paper. In Section 7, we analyze our Relay Gaussians from the perspective of Gaussian densification. In Section 8, we emphasize that our method adopts a unified reconstruction framework rather than a segment-based approach, aiming to prevent any potential misunderstandings. Section 9 provides detailed information about the datasets used in our experiments. Section 10 presents more implementation details to facilitate reproducibility. Finally, in Section 11, we showcase additional experimental results, including qualitative and quantitative comparisons, along with a description of the accompanying videos for better visualization of our method’s performance.

## 7. Relay Gaussians from Densification Perspective

**Spatial Densification.** In standard 3DGS, regions with insufficient spatial representation are typically addressed by adding Gaussians in those areas, an operation we refer to as *spatial densification*. Most prior 4D reconstruction methods [36, 37] adopt canonical 3D Gaussians combined with a temporal deformation field as the 4D representation framework. Consequently, densification is performed solely in the canonical 3D space, essentially extending the spatial densification strategy of 3DGS into 4D settings, as illustrated in Fig. 5 (a). These methods assume that a single canonical Gaussian corresponds to an entire motion trajectory across time, with the motion field responsible for learning the Gaussian’s position at each time step. However, this assumption is overly idealized and proves challenging in practice. Real-world scenes often involve large-scale, complex motions, and motion fields, typically implicit, may struggle to capture these trajectories accurately, leading to significant errors in both spatial and temporal alignment.

**Temporal Densification.** Analogous to spatial densification in static 3DGS, the idealized motion trajectory of a canonical Gaussian along the time dimension in dynamic scene reconstruction may be underrepresented. An intuitive solution to this issue, as depicted in Fig. 5 (b), is to introduce additional Gaussians along the trajectory and optimize them, progressively achieving a more accurate representation of the intended motion trajectory. This operation is referred to by us as *temporal densification*.

In the second stage of our method, multiple copies of the foreground Gaussians are replicated and optimized to their target positions, which fundamentally constitutes *temporal densification*. This process directly increases the temporal



**Figure 5. Illustrative depiction of two types of densification.** In 3DGS for static scene reconstruction, *spatial densification* is employed to better fit 3D structures. Prior 4D methods, as shown in (a), perform densification within a canonical 3D space, relying on deformation fields to model motion trajectories, but often fail to sufficiently represent these trajectories. As shown in (b), explicitly densifying along the motion trajectory by adding new Gaussians enables a more accurate representation of dynamic motion. Our method introduces Relay Gaussians, fundamentally rooted in the intrinsic combination of spatial and temporal densification, enabling enhanced 4D reconstruction.

density of the Gaussians along their idealized motion trajectories, ensuring they are sufficiently distributed to capture the complex dynamics of highly active regions. By doing so, it lays the groundwork for accurately representing large-scale, dynamic motions.

However, in dynamic scenes, the non-rigid nature of dynamic objects introduces further complexity. The same object may undergo varying transformations at different time steps, sometimes requiring more Gaussians for accurate representation, sometimes fewer, and occasionally none at all—such as when the object moves out of the scene or is occluded or enveloped by other content. Temporal densification must adapt to these variations, ensuring that the Gaussian representations dynamically align with the scene’s temporal and structural changes for optimal fidelity and efficiency.

Thus, the second stage of our method is designed as a dedicated process that goes beyond simple temporal densification by leveraging pseudo-views constructed from multiple temporal frames to refine the process. These pseudo-views serve two critical purposes: first, they enable a more intensive temporal densification by providing additional supervisory signals, ensuring that the replicated Gaussians are further aligned with complex and large-scale motion trajectories. Second, they support enhanced spatial densification by guiding the optimization of Gaussians to adapt to non-rigid transformations. This ensures that the learned canonical Gaussians achieve both temporal precision and spatial

consistency, providing a robust and unified foundation for accurate and adaptable dynamic scene reconstruction.

## 8. Unified vs. Segment-Based Reconstruction

Despite the explicit use of temporal segments in the second stage of our method for learning Relay Gaussians, our approach is fundamentally different from segment-based reconstruction methods such as Deformable3DGS [37] and ST-GS [19], which rely on segment-wise learning for long-term dynamic scenes.

Segment-based methods treat each temporal segment as an independent learning task, reconstructing a motion field and 4D representation for each segment separately. In contrast, our approach leverages temporal segmentation purely as an optimization strategy within a unified framework, where all segments collectively contribute to a single, cohesive 4D representation.

Instead of performing multiple independent reconstructions—e.g., 10 separate processes for a 250-frame sequence divided into 25-frame segments—our method employs a single training pipeline across three fixed stages (3k, 14k, and 20k steps, respectively). This unified process not only ensures temporal coherence across the entire sequence but also significantly reduces reconstruction time and storage requirements compared to segment-based methods.

## 9. Dataset Details

**PanopticSports Dataset.** The cameras are temporally aligned with accurate intrinsic and extrinsic parameters. Positioned in a roughly hemispherical arrangement around the area of interest in the middle of the capture studio, the cameras provide comprehensive coverage of the scene. The images are undistorted using the provided distortion parameters and resized to  $640 \times 360$ . The dataset provides a point cloud generated by 10 available depth cameras for each scene. In our experiments, this point cloud is first downsampled to approximately 35,000 points, which are then used to initialize the Gaussian primitives. Each scene involves one or two moving persons and some moving objects, while the background remains completely static. Additionally, the foreground colors are quite similar to the background, which further increases the reconstruction difficulty due to the reduced contrast between the foreground and background.

**VRU Basketball Games Dataset.** The camera poses and distortion parameters were estimated using the first frame from all 34 views by COLMAP [31], and all frames were undistorted accordingly. After undistortion, the resolution slightly increases, and we did not resize the images back to  $1920 \times 1080$ . Following the 4D-GS [36] method, a point cloud was generated and downsampled to approximately 80,000 points for initializing the Gaussian primitives. Each scene includes multiple basketball players, a basketball,

Table 5. Quantitative results on the VRU Basketball Games dataset at half resolution. “ST-GS” utilizes point clouds of all 250 frames, the default setting for their method.

Method	PSNR (dB ↑)	
	GZ	DG4
ST-GS [19]	27.61	26.87
E-D3DGS [1]	26.33	25.39
<b>RelayGS (Ours)</b>	<b>28.97</b>	<b>27.50</b>

scoreboards, advertisement banners, and thousands of spectators. The basketball players and the basketball exhibit fast and large-scale movements with highly complex motion patterns, including non-rigid deformations. The scoreboards and banners also dynamically change over time, and even the background spectators are not completely static, as some exhibit subtle movements. Additionally, the physical scale of the scene is significantly larger than previously available dynamic scene datasets, making it highly challenging to reconstruct.

## 10. More Implementation Details

Our method employs slightly different settings for learning rates and densification thresholds between the foreground and background Gaussians. The background learning rates are similar to those used in previous methods, with the initial learning rate for position set to  $2e-4$  and the minimum learning rate to  $1e-5$ . For the foreground Gaussians, the initial learning rate for position is set to  $1e-3$ . The gradient threshold for densification is  $1e-4$ , which is half of the threshold used for the background. Additionally, the scaling threshold for densification is set to  $1e-3$  for the foreground, which is 0.1 times that of the background. These settings encourage the foreground Gaussians to be smaller and split faster than the background Gaussians. More detailed experimental settings will be released in our future open-source code to better support reproducible research.

## 11. Additional Experimental Results

The goal of the first two stages of our method is to learn a more robust base Gaussian representation, simplifying complex motion patterns in the scene and preparing for full learning in the final stage. Using low-resolution views during these stages produces comparable results while significantly reducing training time. Additionally, we observed that our method performs more effectively at low resolutions, resulting in a larger performance gap compared to counterpart methods. The results are presented in Tab. 5, further reinforcing the superiority of our approach in motion learning.

We present the quality comparison on other scenes from



Figure 6. Qualitative comparisons on DG4 scene of VRU Basketball Games dataset.

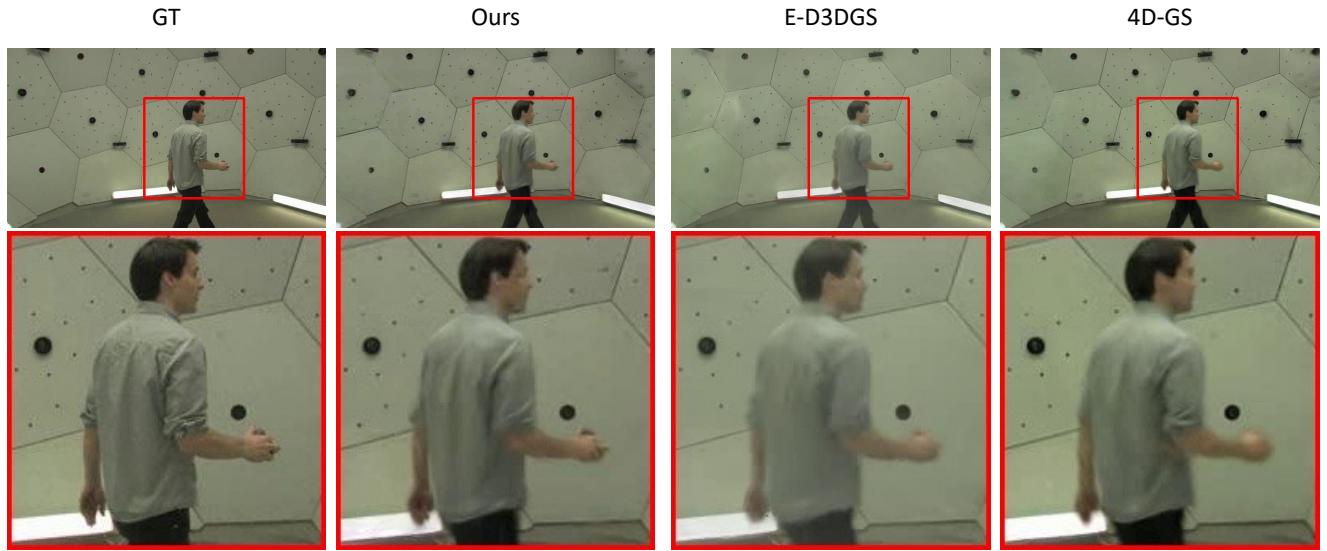


Figure 7. Qualitative comparisons on Juggle scene of PanopticSports dataset.

the two datasets in Figures 6 to 11. To highlight the differences between our method and other methods, we have marked specific foreground regions with red and blue boxes and magnified them for closer inspection. As shown in the magnified views, our method reconstructs the foreground more completely and produces higher-quality details, demonstrating superior performance in preserving fine-grained structures. These qualitative results clearly demonstrate that our method consistently achieves significantly better visual quality compared to competitive counterparts across different scenes from both datasets, highlighting the generalization ability and robustness of our RelayGS method.

In Fig. 12, we provide additional visualization results of Relay Gaussians on the PanopticSports dataset, showcasing how our method learns Relay Gaussians for large-scale dynamic content.

In the zip file of this Supplementary Material, which

includes this document, there are 3 videos (also accessible [online](#)), all composed of 4 test views with 10 seconds per view, resulting in a total duration of 40 seconds:

- VRU\_GZ\_GT.mp4: The ground truth video.
- VRU\_GZ\_RelayGS\_PSNR-28.06.mp4: The video rendered by our RelayGS method.
- VRU\_GZ\_ST-GS\_PSNR-27.32.mp4: The video rendered by the prior SOTA ST-GS [19] method initialized using the sparse point clouds of all 250 frames.

These videos allow a direct comparison of reconstruction quality and motion coherence. As shown, our RelayGS method demonstrates superior performance in both aspects compared to the competitive ST-GS method.



Figure 8. Qualitative comparisons on Boxes scene of PanopticSports dataset.

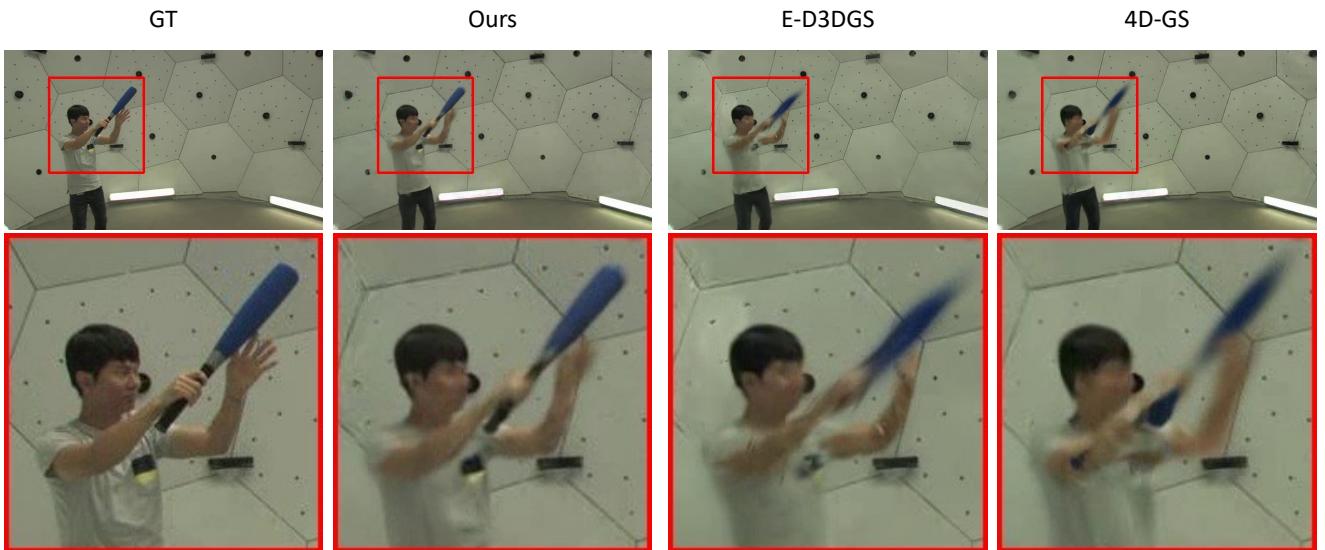


Figure 9. Qualitative comparisons on Softball scene of PanopticSports dataset.

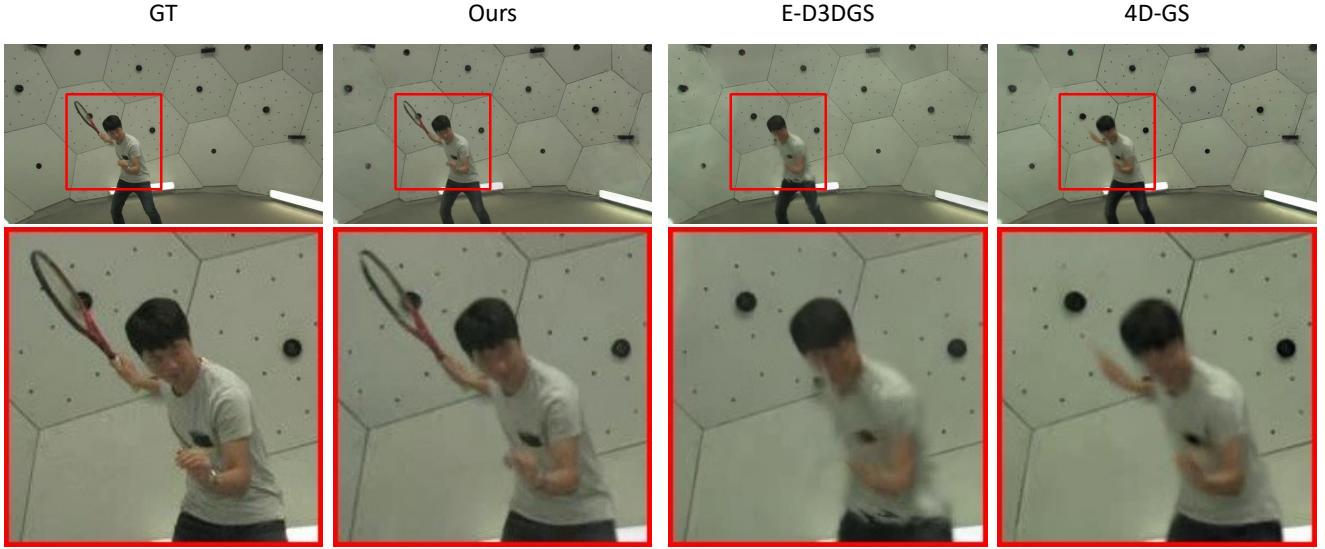


Figure 10. Qualitative comparisons on Tennis scene of PanopticSports dataset.

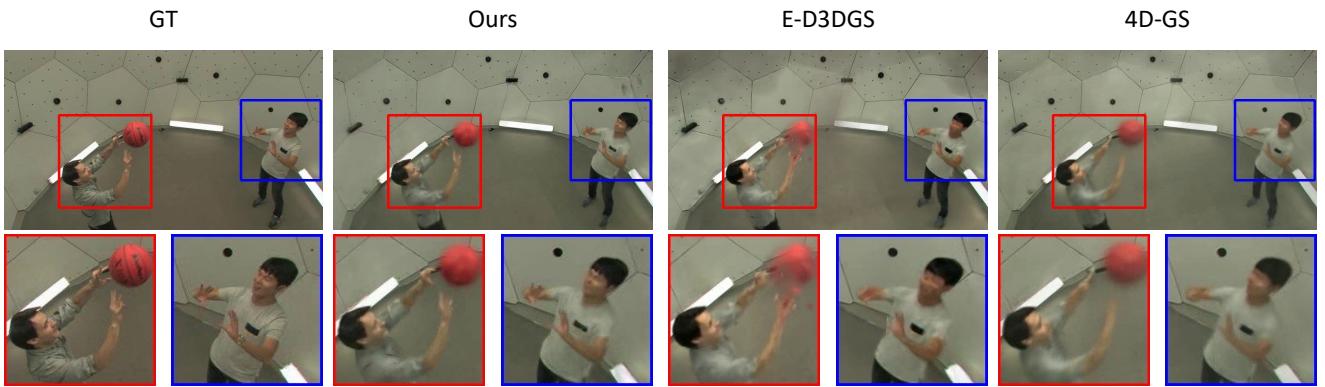


Figure 11. Qualitative comparisons on Basketball scene of PanopticSports dataset.

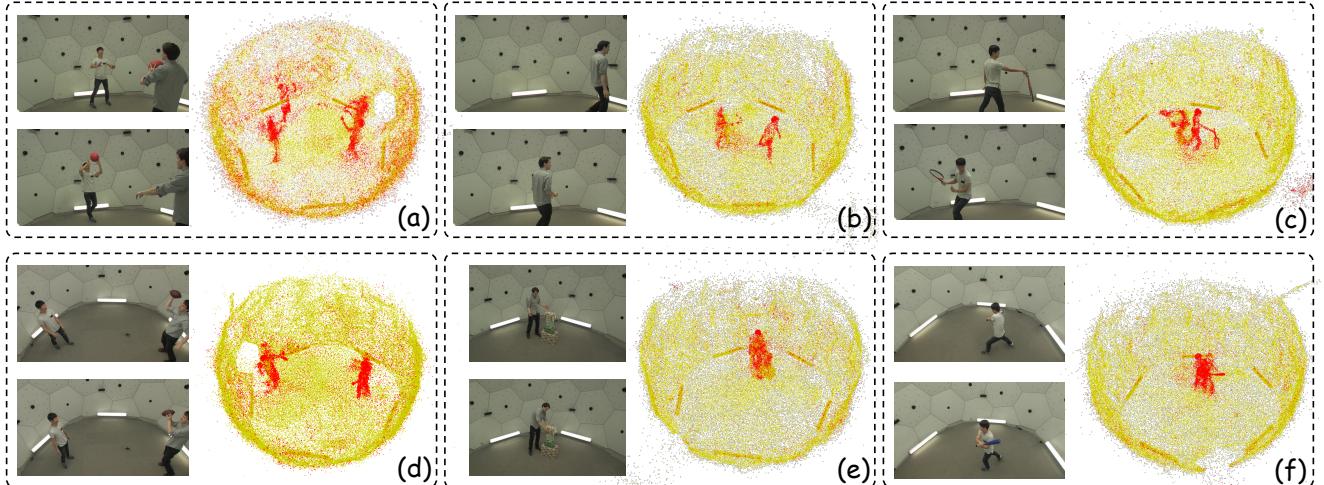


Figure 12. Visualizations of the second-stage dynamic foreground Relay Gaussians (red points) in 6 scenes of the PanopticSports dataset. (a)-(c) show people in the foreground with larger motion amplitudes, generating more dispersed trajectories. (d)-(f) show people in the foreground with smaller motion amplitudes, generating more concentrated trajectories.