

# ReCon-GS: Continuum-Preserved Gaussian Streaming for Fast and Compact Reconstruction of Dynamic Scenes

Jiaye Fu<sup>1,2</sup>, Qiankun Gao<sup>1</sup>, Chengxiang Wen<sup>1</sup>, Yanmin Wu<sup>1</sup>,  
Siwei Ma<sup>2</sup>, Jiaqi Zhang<sup>2\*</sup>, Jian Zhang<sup>1,3\*</sup>

<sup>1</sup>School of Electronic and Computer Engineering, Peking University

<sup>2</sup>National Engineering Research Center of Visual Technology, Peking University

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



Figure 1: The proposed ReCon-GS framework for streamable dynamic scene reconstruction achieves superior rendering quality with substantially reduced storage. The left figures show the high-quality rendering results of ReCon-GS in a streaming fashion. The right figure is the performance comparison with previous SOTA [1, 2, 3, 4, 5, 6], where the radius of circle corresponds to the rendering speed.

## Abstract

Online free-viewpoint video (FVV) reconstruction is challenged by slow per-frame optimization, inconsistent motion estimation, and unsustainable storage demands. To address these challenges, we propose the **Reconfigurable Continuum** Gaussian Stream, dubbed **ReCon-GS**, a novel storage-aware framework that enables high-fidelity online dynamic scene reconstruction and real-time rendering. Specifically, we dynamically allocate multi-level Anchor Gaussians in a density-adaptive fashion to capture inter-frame geometric deformations, thereby decomposing scene motion into compact coarse-to-fine representations. Then, we design a dynamic hierarchy reconfiguration strategy that preserves localized motion expressiveness through on-demand anchor re-hierarchization, while ensuring temporal consistency through intra-hierarchical deformation inheritance that confines transformation priors to their respective hierarchy levels. Furthermore, we introduce a storage-aware optimization mechanism that flexibly adjusts the density of Anchor Gaussians at different hierarchy levels, enabling a controllable trade-off between reconstruction fidelity and memory usage. Extensive experiments on three widely used datasets demonstrate that, compared to state-of-the-art methods, **ReCon-GS** improves training efficiency by approximately 15% and achieves superior FVV synthesis quality with enhanced robustness and stability. Moreover, at equivalent rendering quality, **ReCon-GS** slashes memory requirements by over 50% compared to leading state-of-the-art methods. Code is available at: <https://github.com/jyfu-vcl/ReCon-GS/>.

\*corresponding to: zhangjian.sz@pku.edu.cn, jqzhang@pku.edu.cn

# 1 Introduction

Free-viewpoint Video (FVV) reconstruction has emerged as a cornerstone of next-generation immersive media systems, driven by the escalating demands of XR, AR, and VR applications. While this technology enables dynamic scene exploration from arbitrary viewpoints, its inherent spatiotemporal complexity poses persistent challenges in balancing real-time performance with reconstruction accuracy. Offline approaches [7, 1, 8, 9], though capable of high-quality reconstruction, fail to meet the low-latency requirements of real-time applications such as streaming media and immersive live broadcasting. To address this limitation, recent methods [10, 6] adopt streaming reconstruction frameworks that progressively model inter-frame deformation for frame-by-frame reconstruction.

Despite these advancements, critical bottlenecks persist in achieving practical real-time FVV reconstruction. First, current streaming 3D Gaussian splatting methods incur redundant storage due to deformation fields, primarily attributed to the fundamental incompatibility between the uniform spatial motion encoding strategy of Gaussian primitives and the inherent hierarchical characteristics of physical motion. This mismatch leads to inaccurate and inefficient deformation modeling for dominant macroscopic motions, triggering redundant deformation parameter storage and severe error accumulation issues. While some approaches [11, 12] mitigate error accumulation in streaming motion modeling by introducing auxiliary mechanisms (*e.g.*, optical flow estimation networks), such solutions inevitably increase training costs and significantly compromise practical applicability. Furthermore, Industrial deployment necessitates dynamically balancing storage efficiency and fidelity to accommodate fluctuating resources. However, existing frameworks employ static strategies, thereby lacking the flexibility necessary for practical deployment.

To address the aforementioned challenges, we propose the **Reconfigurable Continuum** Gaussian Stream, referred to as **ReCon-GS**, a novel streaming framework that jointly optimizes rendering fidelity and storage adaptability. Our method introduces *Adaptive Hierarchical Motion Representation*, an anchor-driven multi-scale motion encoding paradigm that dynamically allocates Anchor Gaussians through grid-based farthest point sampling to decompose scene dynamics into coarse-to-fine motion structures. This paradigm ensures alignment between Anchor Gaussian distributions and spatial geometric features. Deformation parameters are propagated to clustered General Gaussians via hierarchical mechanisms, enabling effective multi-scale rigid motion representation with minimal storage requirements. Then, **ReCon-GS** implements a *Dynamic Hierarchy Reconfiguration* strategy to resolve the degradation of local motion representation caused by Anchor Gaussian drift during spatial deformation. Specifically, within each hierarchy level, we perform trajectory matching of Anchor Gaussians across frames to track their movement throughout the sequence, while constraining deformation field propagation to operate exclusively within these matched trajectories. Reorganized anchors inherit deformation priors only from their original predecessors at the same hierarchy level, thereby preserving both motion inheritance granularity and temporal motion coherence. Consequently, leveraging these architectural innovations, we reformulate 4D reconstruction as a storage-aware fidelity optimization problem, achieving paradigm-level advancements analogous to rate-distortion optimization theory. This enables dynamic balance between memory efficiency and rendering quality according to application demands without compromising real-time performance, as our framework adaptively adjusts the density of Anchor Gaussians at different hierarchy levels based on scene complexity.

Extensive experiments validate our framework’s efficacy across diverse scenarios, demonstrating that **ReCon-GS** achieves compact yet high-fidelity reconstruction under varying motion patterns while outperforming state-of-the-art streaming methods in three well-known datasets. Our key contributions are summarized as follows:

- We propose Adaptive Hierarchical Motion Representation, an anchor-driven multi-scale motion encoding paradigm that adaptively align Anchor Gaussians with scene geometry to decompose scene dynamics into coarse-to-fine structures, achieving an efficient yet compact motion representation.
- We design a Dynamic Hierarchy Reconfiguration strategy to address anchor drift-induced motion degradation, combining re-hierarchization with intra-level deformation inheritance to propagate priors exclusively within original hierarchy layers, eliminating cross-layer interference while preserving motion fidelity.
- Comprehensive experiments across multiple datasets validate **ReCon-GS**’s superiority over state-of-the-art streaming methods. Our framework achieves superior performance with 15%

faster training convergence while reducing storage requirements by over 50% at equivalent rendering quality. Even under constrained storage budgets, **ReCon-GS** maintains significant rendering quality improvements, demonstrating robust practical efficacy.

## 2 Related Work

### 2.1 3D Gaussian Splatting for Static Scenes

With the rise of novel view synthesis (NVS) tasks, 3D Gaussian Splatting (3DGS) [13] has gradually replaced Neural Radiance Fields (NeRF)-based methods [14, 15, 16, 17, 18] as the mainstream approach for static scene reconstruction due to its high rendering quality and fast training speed. However, its massive storage requirements significantly hinder the practical application of 3DGS. Consequently, numerous studies have focused on achieving more compact 3DGS representations while maintaining high-quality rendering. Some methods [19, 20, 21] concentrate on pruning Gaussian primitives, such as removing unimportant Gaussians through trainable mask mechanisms [22] or gradient thresholding approaches [23]. Additionally, other approaches [24] aim to reduce the parameters of 3D Gaussians to decrease per-primitive storage. Meanwhile, 3DGS methods [19, 22, 24, 25, 26] based on Vector Quantization (VQ) principles have also proven effective in reducing storage demands. Alternative strategies [27, 28, 29] explore compact spatial structure representations to minimize 3DGS storage. Scaffold-GS [30] achieves efficient storage reduction by clustering Gaussians to anchor points and implicitly representing Gaussian attributes for each cluster through MLPs. Subsequent works [31, 32, 33] further compress 3DGS by incorporating spatial structure priors into entropy encoders and other compression techniques.

### 2.2 3D Gaussian Splatting of Dynamic Scenes

3D dynamic scene reconstruction stands as one of the most challenging tasks in computer vision and graphics. With the rise of 3D Gaussian Splatting (3DGS), dynamic reconstruction approaches have diverged into two main paradigms. The first involves offline methods [7, 1, 8, 34, 35] that implicitly model scene dynamics through canonical Gaussian fields coupled with spatiotemporal deformation fields. While these methods achieve compact representations, they struggle with efficient motion parameterization. Recent advancements address this by embedding Gaussians into sparse motion representations using anchor points [36, 37, 38] or control points [39, 40, 41], enabling motion propagation through sparse keyframes while preserving reconstruction quality. The second paradigm adopts streaming frameworks that first reconstruct a high-quality 3DGS representation for the initial frame and then model subsequent frames via deformation fields [42, 6, 10]. However, the massive storage overhead of per-frame motion parameters poses significant challenges. Gao et al. [3] mitigate this through a hierarchical motion representation framework, decomposing deformations into coarse-to-fine transformations to reduce parameter dimensionality. Despite this, streaming frameworks inherently suffer from error accumulation due to incremental updates. To address this, some methods [43, 44, 11, 12, 45] leverage pre-trained motion estimation models (e.g., optical flow networks) to guide 3D Gaussian deformations via 2D motion priors, enhancing both rendering quality and temporal consistency. These innovations highlight the critical balance between storage efficiency, computational tractability, and reconstruction fidelity in dynamic 3D modeling.

## 3 Preliminaries

3DGS models scenes using anisotropic 3D Gaussians defined by a mean  $\mu$  and a covariance matrix  $\Sigma = \mathbf{R}\mathbf{S}\mathbf{S}^\top\mathbf{R}^\top$ , where  $\mathbf{S} = \text{diag}(s_x, s_y, s_z) \in \mathbb{R}^3$  represents axis-aligned scaling, and  $\mathbf{R} \in SO(3)$  is parameterized by a quaternion  $\mathbf{q}$ . Each Gaussian’s view-dependent color uses spherical harmonics  $\mathbf{SH}$  and opacity  $\alpha \in [0, 1]$ .

The mathematical formulation of Gaussian is:

$$\mathcal{G}(\mathbf{x}) = e^{-\frac{1}{2}(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)}. \quad (1)$$

For novel view synthesis, Gaussians are projected onto the imaging plane using a view transformation matrix  $\mathbf{W}$ . The projected 2D covariance  $\Sigma'$  is computed through the Jacobian  $\mathbf{J}$  of the projective transformation:

$$\Sigma' = \mathbf{J}\mathbf{W}\Sigma\mathbf{W}^\top\mathbf{J}^\top. \quad (2)$$

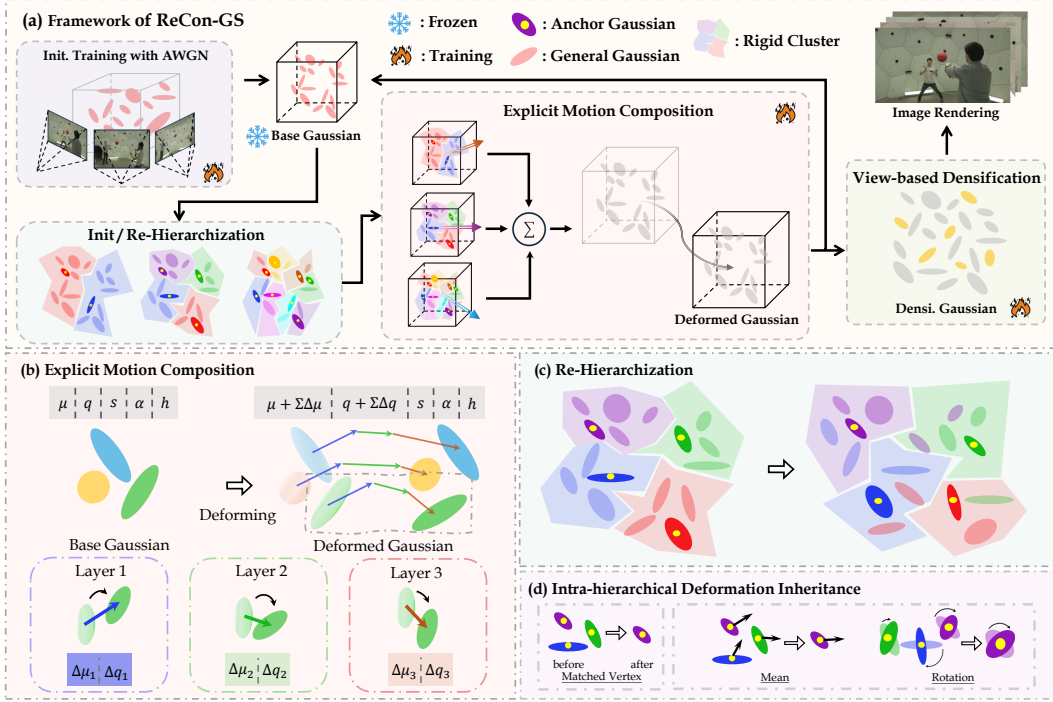


Figure 2: **Illustration of our ReCon-GS framework.** (a) We begin by generating a high-quality base 3D Gaussian Splatting (3DGS) representation for the first frame, then embed it into an adaptively hierarchical motion presentation framework via grid-based farthest-point sampling. (b) Explicit Motion Composition updates the base 3DGS across successive frames. (c) and (d) Periodically, a Re-Hierarchization stage accommodates complex object motion while preserving temporal consistency through Intra-hierarchical Deformation Inheritance. Finally, a view-based densification further refines the 3DGS for high-quality rendering.

Rendering proceeds by alpha-blending  $N$  depth-ordered Gaussians at each pixel, with the final color calculated as:

$$C = \sum_{i \in N} \mathbf{c}_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \quad (3)$$

where  $\mathbf{c}_i$  and  $\alpha_i$  correspond to the color and blending weight of the  $i^{th}$  Gaussian, respectively.

The learning pipeline iteratively alternates between gradient-based parameter updates and topological adaptation. Parameter optimization is supervised by the  $\mathcal{L}_1$  loss and D-SSIM term:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_1 + \lambda\mathcal{L}_{D-SSIM}, \quad (4)$$

where  $\lambda$  is typically set to 0.2.

## 4 Methodology

### 4.1 Overview

Our goal is to achieve high-quality dynamic scene reconstruction from multi-view videos under varying storage constraints. To accomplish this, our framework first generates compact Base Gaussians  $\mathcal{G}_B$  through an additive noise-injected 3DGS training strategy [3] during initial frame reconstruction. Subsequently, to efficiently parameterize per-frame Gaussian deformations, we propose an adaptive hierarchical motion representation framework that explicitly models scene dynamics through scale-aware motion decomposition. Specifically, each Gaussian’s motion is parameterized as a composition of layer-wise rigid transformations across multiple geometric scales, with each deformation layer governed by a distinct set of Anchor Gaussians  $\mathcal{G}_A$  spatially distributed via grid-based farthest-point sampling. General Gaussians within each layer hierarchically inherit blended motion parameters from

their nearest anchor counterparts through cross-scale deformation coefficients, enabling progressive motion aggregation from coarse to fine hierarchies.

To enhance the adaptability of our anchor-based motion representation for scenes with varying motion magnitudes, we introduce dynamic hierarchy reconfiguration - a self-optimizing mechanism that redistributes Anchor Gaussians to match evolving scene geometry while maintaining temporal motion consistency through Intra-hierarchical Deformation Inheritance. Finally, to further refine local details, ReCon-GS optimizes 3DGS representation through a view-adaptive densification process, enhancing scene rendering fidelity. The complete pipeline is illustrated in Figure 2 (a).

## 4.2 Adaptively Hierarchical Motion Presentation

Inspired by the hierarchical decomposition of real-world object motion into multi-scale rigid-body components, we propose an adaptive hierarchical motion representation framework. Our framework begins with the **Base Gaussians**  $\mathcal{G}_B$ , which is the static 3D Gaussians reconstructed from the first frame of multi-view inputs. Building upon  $\mathcal{G}_B$ , we establish an adaptive hierarchical motion representation through three core stages:

**Anchor Gaussian Initialization.** Given  $\mathcal{G}_B = \{g_i\}_{i=1}^N$  with position  $\mu_i \in \mathbb{R}^3$ , we first compute the scene’s spatial bounds as  $\mu_{\min} = \min_i \mu_i$  and  $\mu_{\max} = \max_i \mu_i$ .  $\mathcal{G}_B$  is then divided into a uniform 3D grid with resolution  $M = \lceil N_{\text{anchor}}^{1/3} \rceil$ , where  $N_{\text{anchor}}$  is the number of anchor. For each cell  $\mathcal{C}_{i,j,k}$ , we select the **Anchor Gaussians**  $g_a \in \mathcal{G}_A$  with position  $\mu_a$  closest to its geometric center:

$$\mathbf{g}_a = \arg \min_{\mathbf{g} \in \mathcal{C}_{i,j,k}} \|\mu - \mathbf{c}_{i,j,k}\|_2, \text{ where } \mathbf{c}_{i,j,k} = \mu_{\min} + \left(i + \frac{1}{2}, j + \frac{1}{2}, k + \frac{1}{2}\right) \odot \frac{\Delta}{M}, \quad (5)$$

with  $\Delta = \mu_{\max} - \mu_{\min}$ . This *grid-based farthest point sampling* ensures each anchor locally represents its cell while maintaining global uniformity. Compared to conventional farthest point sampling approaches, our grid-based strategy achieves significant computational acceleration through structured sampling constraints, effectively reducing the time complexity while preserving sampling quality.

**Hierarchical Rigid-Cluster Formation.** With  $\mathcal{G}_A$  initialized, each **General Gaussians**  $g_n \in \mathcal{G}_N$  is assigned to its nearest anchor via L1-distance minimization:

$$g_a^* = \arg \min_{g_a \in \mathcal{G}_A} \|\mu_n - \mu_a\|_1. \quad (6)$$

forming primary rigid clusters  $\{\mathcal{R}_a\}$ . To address residual motions within rigid clusters, we recursively subdivide the hierarchy into two finer levels within the original spatial coordinate system. At each subsequent level, we increase the density of Anchor Gaussians by adapting the grid resolution  $M = \lceil (N_{\text{anchor}} \cdot 3^{l-1})^{1/3} \rceil$ , where  $l = 2, 3$  denotes the hierarchy level.

**Explicit Motion Composition.** As shown in Figure 2 (b), each hierarchy level  $l \in \{1, 2, 3\}$  associates clusters with rigid transformations  $\mathbf{T}^{(l)} = (\Delta\mu^{(l)}, \Delta\mathbf{q}^{(l)}, .)$ , where  $\Delta\mu^{(l)} \in \mathbb{R}^3$  and  $\Delta\mathbf{q}^{(l)} \in \mathbb{R}^4$  denote incremental translation and rotation of Anchor Gaussians  $\mathcal{G}_A$ . Thus, Anchor Gaussians  $\mathcal{G}_A$  act as rigid motion keypoints that govern the affine transformation of all General Gaussians  $\mathcal{G}_N$  within their associated cluster. By propagating these parameters hierarchically, the motion of each Anchor dictates the collective deformation of its rigid sub-region, emulating real-world rigid-body dynamics. The total deformation of a Gaussian is the cumulative sum across levels:

$$\Delta\mu_h = \sum_{l=1}^3 \Delta\mu^{(l)}, \quad \Delta\mathbf{q}_h = \sum_{l=1}^3 \Delta\mathbf{q}^{(l)}. \quad (7)$$

Upper level model global rigid motions shared across clusters, while the finest level resolves residual deformations unique to individual Gaussians. Crucially, we decouple geometric attributes (optimized via  $\Delta\mu_h$  and  $\Delta\mathbf{q}_h$ ) from static appearance parameters (sphere harmonic coefficient  $SH$  and opacity  $\alpha$ ), freezing the latter to avoid parameter explosion from separately modeling appearance deformations, which is critical for ReCon-GS’s storage efficiency. Since pixel intensities rely on  $\alpha$ -blended Gaussian contributions, precise geometric deformation alone ensures photorealistic rendering. By optimizing geometry while freezing appearance attributes, ReCon-GS achieves parameter efficiency without sacrificing visual quality.

**Motion Parameter Preservation.** ReCon-GS can store motion parameters at each level more compactly compared to previous streaming reconstruction method [3]. This is primarily due to the use of a carefully designed *grid-based farthest point sampling* algorithm for anchor Gaussian assignment, instead of a uniform anchor sampling strategy. As a result, ReCon-GS can store motion parameters for Anchor Gaussians based solely on their relative positions, without the need for additional indices. This significantly reduces the memory required for storing motion parameters.

### 4.3 Dynamic Hierarchy Reconfiguration

Using Adaptive Hierarchical Motion Representation framework, we efficiently allocates explicit motion parameters according to scene density and geometric structures. However, ReCon-GS also inherits a critical challenge common to streaming reconstruction methods [6, 3]: persistent scene motion induces severe deformations in local rigid clusters, causing Anchor Gaussians to gradually lose their representativeness as rigid motion bases. This degradation propagates errors through the motion hierarchy, accumulating artifacts in rendered sequences. To address this, we propose Dynamic Hierarchy Reconfiguration strategy—a self-correcting mechanism that periodically reinitializes the hierarchical structure while preserving temporal motion coherence through intra-hierarchical deformation inheritance.

**Periodic Re-Hierarchization.** At each reconfiguration step  $t = kT$  ( $k \in \mathbb{N}^+$ ), the framework regenerate Anchor Gaussians  $\mathcal{G}_A$  using grid-based farthest point sampling on the current Gaussian distribution, ensuring updated anchors align with the latest scene geometry. Then, General Gaussian  $\mathcal{G}_N$  is reassigned to new anchors via L1-distance minimization. The process is described in Figure 2 (c).

**Intra-hierarchical Deformation Inheritance.** As present in Figure 2 (d), after re-hierarchization, we implement intra-hierarchical deformation inheritance to enforce temporal motion consistency between  $\mathcal{G}_A$  across hierarchy levels. For each hierarchy level  $l \in \{1, 2, 3\}$ , each new Anchor Gaussian inherits deformation parameters from its three nearest legacy Anchors to ensure temporal continuity. Specifically, The translational deformation is governed by:

$$\Delta \mu'_a{}^{(l)} = \frac{1}{3} \sum_{i=1}^3 \Delta \mu_{a_i}^{(l)}, \quad (8)$$

where  $\Delta \mu'_a{}^{(l)}$  denotes the translational deformation parameter of new Anchor Gaussian at level  $l$ , and  $\Delta \mu_{a_i}^{(l)}$  represents the translational deformation parameter of the  $i$ -th matched Anchor Gaussian. Rotational deformation follows an eigenvector-based formulation:

$$\Delta \mathbf{q}'_a{}^{(l)} = \frac{v_{\max}(\mathcal{M})}{\|v_{\max}(\mathcal{M})\|}, \quad \mathcal{M} = \sum_{i=1}^3 \Delta \mathbf{q}_{a_i}^{(l)} (\Delta \mathbf{q}_{a_i}^{(l)})^\top, \quad (9)$$

where  $\Delta \mathbf{q}'_a{}^{(l)}$  is the rotational deformation parameter of new Anchor Gaussian, and  $\Delta \mathbf{q}_{a_i}^{(l)}$  represents the rotational deformation parameter of the  $i$ -th matched Anchor Gaussian.  $v_{\max}(\mathcal{M})$  denotes the eigenvector corresponding to the largest eigenvalue of  $\mathcal{M}$ .

### 4.4 Storage-aware Optimization

Thanks to the streaming architecture of ReCon-GS, the storage overhead of Gaussian primitives constitutes only a minor fraction of the total storage consumption, with the majority allocated to per-frame deformation field parameters. By adopting an anchor-driven adaptive hierarchical motion representation, ReCon-GS operates as a storage-aware framework—the hierarchical anchor resolution  $\{N_{anchor}^{(l)}\}_{l=1}^3$  dynamically adapts to varying scene complexity and storage constraints. Moreover, ReCon-GS adopts a decoupled two-stage optimization pipeline to balance motion fidelity and computational efficiency. In Phase 1, we exclusively train the hierarchical deformation fields while freezing the geometric and appearance attributes of the Base Gaussians  $\mathcal{G}_B$ . Once the deformation fields converge, Phase 2 activates the view-based densification to address persistent discrepancies between learned motion and scene dynamics.

Table 1: **Quantitative comparison** on N3DV dataset. The storage metric includes the size without and with the initial frame, separated by “/”. The training time metric includes the first frame training. The method with <sup>†</sup> is reproduced by us through official code in the same experimental environment.

Category	Method	PSNR (dB)↑	SSIM↑	LPIPS↓	Storage (MB)↓	Train (sec)↓	Render (FPS)↑
Offline	4DGS <sup>†</sup> [1]	31.36	0.950	0.131	0.3	7.8	30
	STG [9]	32.05	0.948	-	0.67	20	140
	SaRO-GS [5]	32.15	-	-	1.0	-	40
	Swift4D [46]	32.23	-	-	0.4	5.0	125
	SplineGS [47]	32.60	-	-	-	11	76
Online	Dynamic 3DGS [42]	30.67	-	-	-9.2	560	-
	StreamRF [48]	30.68	0.930	-	17.7/31.5	15	12
	3DGStream <sup>†</sup> [6]	31.35	0.948	0.130	7.6/7.8	8.1	245
	4DGC [2]	31.58	0.943	-	-0.5	50	168
	QUEEN-I [4]	32.19	0.946	0.136	-0.75	7.9	248
	HiCoM <sup>†</sup> [3]	32.08	0.953	0.130	0.48/0.69	6.6	255
	<b>ReCon-GS (ours)</b>	32.66	0.957	0.123	0.40/0.44	6.4	250

Table 2: **Quantitative comparison** on Meet Room and PanopticSports datasets. The storage metric includes the size without and with the initial frame, separated by “/”. The training time metric includes the first frame training. The method with <sup>†</sup> is reproduced by us through official code in the same experimental environment. The method marked with \* adopts the *Discussion* scene from the Meet Room dataset as the training set, with the remaining scenes serving as the test set.

Method	Meet Room					PanopticSports				
	PSNR (dB)↑	LPIPS↓	Storage (MB)↓	Train (sec)↓	Render (FPS)↑	PSNR (dB)↑	LPIPS↓	Storage (MB)↓	Train (sec)↓	Render (FPS)↑
3DGStream <sup>†</sup> [6]	29.30	0.188	4.0/4.1	4.77	260	23.02	0.187	7.9/8.1	5.87	369
IGS-I* [12]	30.13	-	-1.26	2.67	252	-	-	-	-	-
HiCoM <sup>†</sup> [3]	29.57	0.182	0.30/0.39	3.91	236	29.17	0.142	1.33/2.11	8.60	358
<b>ReCon-GS (ours)</b>	30.84	0.163	0.28/0.30	3.86	256	29.33	0.136	0.64/0.8	7.14	410

## 5 Experiments

### 5.1 Dataset

**The Nerual 3D Video Dataset (N3DV)** [51] comprises 6 indoor dynamic scenes. Each scene contains 18 to 21 free-viewpoint videos, with each video spanning 300 frames captured at 30 FPS and a resolution of  $2704 \times 2028$ .

**Meeting Room Dataset** [48] comprises 3 dynamic scenes across diverse real-world scenarios. Each scene contains 13 free-viewpoint videos, with each video spanning 300 frames captured at 30 FPS and a resolution of  $1280 \times 720$ .

**PanopticSports Dataset** [42], derived from sports sequences in the Panoptic Studio dataset [52], includes 6 sports-oriented scenarios. Each scenario consists of 31 free-viewpoint videos, each containing 150 frames recorded at 30 FPS and a resolution of  $640 \times 360$ .

**Technicolor Dataset** [53] consists of video recordings captured by a  $4 \times 4$  camera rigs. Each video has a spatial resolution of  $2048 \times 1088$  and a frame rate of 30 FPS. Following the Ex4DGS [34], evaluation is performed on five distinct scenes (*Birthday*, *Fabien*, *Painter*, *Theater*, and *Trains*) using their original full resolution.

### 5.2 Implementation Details

Our framework initializes first-frame reconstruction using 3DGS [13] with the maximum degree of spherical harmonics (SH) to 1 and a noise-injected coefficient  $\lambda_{noise} = 0.01$ , training on three datasets for 10,000 to 15,000 steps. During the construction of hierarchical motion representation,

Table 3: **Quantitative comparison** on Technicolor dataset. The storage metric includes the size with the initial frame. The training time metric includes the first frame training.

Category	Method	PSNR (dB)↑	SSIM↑	Storage (MB)↓	Render (FPS)↑
NeRF-based	HyperReel [49]	31.80	0.906	1.20	4
Offline	STG [9]	33.60	-	1.10	87
	Ex4DGS [34]	33.62	0.916	2.81	72
Online	E-D3DGS [50]	33.24	0.907	1.54	79
	<b>ReCon-GS (ours)</b>	<b>33.83</b>	<b>0.932</b>	<b>0.82</b>	<b>207</b>

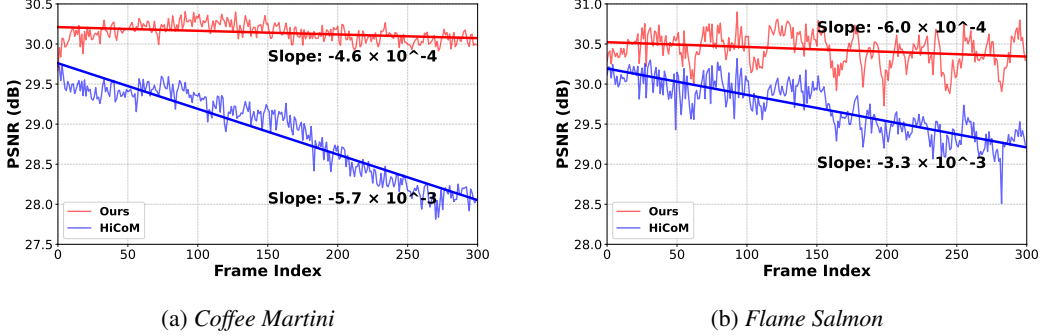


Figure 3: The PSNR Trend Comparison between Ours and HiCoM [3] across different scenarios.

we set the finest-level Anchor Gaussians to 1/24 of the total Gaussian primitive count, with each subsequent layer containing one-third the anchors of its predecessor.

For per-frame optimization, all datasets undergo an initial 100 steps for deformation training. Subsequently, N3DV and Meet Room datasets receive an additional 100 refinement steps for Gaussian densification and pruning, while PanopticSports employs 400 steps to address its higher motion complexity. All experiments are conducted on an NVIDIA RTX 4090 GPU, with results averaged over 3 independent runs to ensure statistical reliability. Extended implementation details are provided in the **Appendix A**.

### 5.3 Experiment Results

**Quantitive comparisons.** We conducted a comprehensive evaluation of our method against existing state-of-the-art (SOTA) online and offline approaches across three datasets using multiple quantitative metrics, including PSNR, SSIM, LPIPS, storage consumption, training time, and rendering speed. To ensure statistical validity, storage and rendering speed measurements were calculated as averaged values across all video frames. As demonstrated in Table 1, Table 2 and Table 3, our method achieves SOTA performance in rendering quality and training efficiency compared to online baselines. Specifically, our method outperforms the recent SOTA streaming methods in rendering quality by **more than 0.5dB** on the N3DV [51], MeetRoom [48], and Technicolor [53] datasets. This improvement underscores the effectiveness of our approach in achieving superior visual fidelity in dynamic scene reconstructions. In the case of the PanopticSports [42] dataset, our method achieves the best rendering quality while inducing a substantial reduction in storage requirements, decreasing storage by approximately 60% compared to previous streaming methods. In comparison to offline methods, ReCon-GS requires slightly more storage, primarily due to the inherent storage advantages of offline methods in implicit motion representation.

Furthermore, as evidenced in Figure 3, ReCon-GS effectively alleviates temporal performance degradation observed in recent SOTA method. Through the Dynamic Hierarchy Reconfiguration Strategy, ReCon-GS successfully circumvents the degradation in motion representation capability of Anchor Gaussians. More quantitive evaluations are provided in the **Appendix C.1**.

**Qualitative comparisons.** While our ReCon-GS framework primarily focuses on optimizing the efficiency of dynamic scene reconstruction, Table 1 and Table 2 substantiate the superiority of our



Figure 4: **Qualitative results** on the *coffee martini* scene in the N3DV Dataset. To ensure a fair comparison, we retrained their official code with the same initial sparse points.

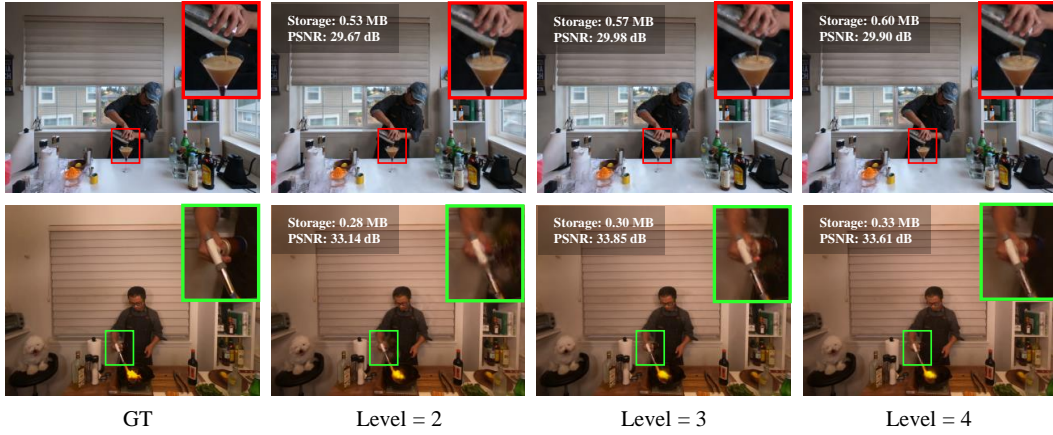


Figure 5: Qualitative results of our ReCon-GS under different hierarchical levels

method in rendering quality. As illustrated in Figure 4, compared to HiCoM [3] and 3DGStream [6], our approach achieves more complete and temporally consistent reconstruction of dynamic scene details across multiple datasets. Moreover, the proposed method demonstrates more enhanced dynamic reconstruction capabilities when handling prolonged temporal intervals. We have also provided an additional subjective quality evaluations in **Appendix C.3**.

#### 5.4 Ablation Study

As evidenced in Table 4, we conducted ablation studies on key strategies of ReCon-GS, including Adaptively Hierarchical Motion Presentation paradigm, Dynamic Hierarchy Reconfiguration strategy and view-based densification process. The experimental results demonstrate that the Adaptively Hierarchical Motion Presentation paradigm, despite introducing some memory overhead, significantly enhances the performance of ReCon-GS, proving the paradigm’s effective motion representation

Table 4: Ablation on key components of our ReCon-GS framework.

Method	PSNR (dB)↑	SSIM ↑	Storage (MB)↓	Train (sec)↓	Render (FPS) ↑
w/o Hierarchical Motion Representation	31.43	0.9525	<b>0.32</b>	<b>6.37</b>	260
w/o Dynamic Hierarchy Reconfiguration	32.00	0.9521	0.45	6.45	248
w/o View-based Densification	32.15	0.9559	0.40	6.52	<b>287</b>
Ours (full)	<b>32.66</b>	<b>0.9571</b>	0.44	6.44	250

Table 5: Ablation on hierarchy depth of our ReCon-GS framework.

Hierarchy Depth	N3DV			Meet Room		
	PSNR (dB) ↑	Storage (MB) ↓	Train (sec) ↓	PSNR (dB) ↑	Storage (MB) ↓	Train (sec) ↓
2	32.422	<b>0.40</b>	<b>6.44</b>	30.20	<b>0.28</b>	3.87
3	<b>32.662</b>	0.44	6.49	30.84	0.30	<b>3.86</b>
4	32.658	0.46	6.54	<b>31.01</b>	0.32	4.01

capabilities. Furthermore, the implementation of Dynamic Hierarchy Reconfiguration strategy further improves ReCon-GS’s rendering quality, indicating enhanced spatiotemporal localization accuracy of Gaussian primitives and ensured temporal motion consistency. While the view-based densification process increases per-frame training time, it substantially improves the representation of high-frequency scene details. This suggests that merely deforming base Gaussian primitives is insufficient for comprehensive dynamic scene modeling.

Furthermore, Table 5 presents ablation experiments on hierarchy depth, motivated by our investigation into motion representation efficiency through ReCon-GS. As shown in Figure 5, experimental findings reveal that a 3-level hierarchy optimally balances high-fidelity scene reconstruction with storage efficiency. This configuration enables ReCon-GS to achieve superior fidelity-storage trade-offs compared to other hierarchical configurations. See **Appendix B** for more ablation studies.

## 6 Conclusion

This paper introduces **ReCon-GS**, a novel storage-aware framework for high-fidelity multi-view video reconstruction. ReCon-GS achieves density-aware multi-layer decoupled compact motion modeling of Gaussian primitives through an Adaptive Hierarchical Motion Representation framework. Furthermore, we propose a Dual-level Deformation Strategy combining inter-frame temporal deformation and intra-frame spatial deformation, accurately capture spatio-temporal positions of Gaussian primitives while avoiding reconstruction failures in high-dynamic motions and emerging objects. Notably, ReCon-GS innovatively transforms dynamic scene reconstruction from conventional single-objective optimization into a storage-aware fidelity optimization framework. This paradigm shift allows dynamic balancing between memory efficiency and rendering quality according to application requirements, significantly enhancing the practical applicability of 3D dynamic reconstruction method. Extensive experiments demonstrate that ReCon-GS achieves state-of-the-art performance in both rendering quality and training efficiency while maintaining exceptional storage effectiveness. Compared with existing SOTA methods, ReCon-GS attains over 50% memory reduction while preserving same rendering fidelity.

**Limitations.** While maintaining alignment with prior methodologies in focusing on multi-view general scene reconstruction tasks, our approach is not specifically optimized for monocular datasets or specialized reconstruction scenarios. Similar to existing streaming-based architectures, we face challenges in dynamic allocation/updating of Gaussians to handle real-time object emergence/dissolution, as abrupt scene changes require instant 3DGS structural adaptations.

## 7 Acknowledge

This work was supported in part by the Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology (No. 2024B1212010006) and Beijing Natural Science Foundation under grant no. L242014, in part by the PCL-CMCC Foundation for science and innovation grant 2024ZY1C0040, which are gratefully acknowledged.

## References

- [1] 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] Qiang Hu, Zihan Zheng, Houqiang Zhong, Sihua Fu, Li Song, Guangtao Zhai, Yanfeng Wang, et al. 4DGC: Rate-Aware 4D Gaussian Compression for Efficient Streamable Free-Viewpoint Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
- [3] Qiankun Gao, Jiarui Meng, Chengxiang Wen, Jie Chen, and Jian Zhang. HiCoM: Hierarchical coherent motion for dynamic streamable scenes with 3d gaussian splatting. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [4] Sharath Girish, Tianye Li, Amrita Mazumdar, Abhinav Shrivastava, Shalini De Mello, et al. QUEEN: QUAntized Efficient ENcoding of Dynamic Gaussians for Streaming Free-viewpoint Videos. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [5] Jinbo Yan, Rui Peng, Luyang Tang, and Ronggang Wang. 4D Gaussian Splatting with Scale-aware Residual Field and Adaptive Optimization for Real-time rendering of temporally complex dynamic scenes. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, 2024.
- [6] 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.
- [7] 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.
- [8] 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 *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH)*, 2024.
- [9] 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.
- [10] 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.
- [11] Zhiyang Guo, Wengang Zhou, Li Li, Min Wang, and Houqiang Li. Motion-aware 3d gaussian splatting for efficient dynamic scene reconstruction. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2024.
- [12] 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
- [13] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics (TOG)*, 2023.
- [14] 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.
- [15] Kai Zhang, Gernot Riegler, Noah Snaveley, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.
- [16] 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 (ICCV)*, 2021.
- [17] 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 (ICCV)*, 2023.

- [18] 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 (CVPR)*, 2022.
- [19] Zhiwen Fan, Kevin Wang, Kairun Wen, Zehao Zhu, DeJia Xu, Zhangyang Wang, et al. Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [20] Zhaoliang Zhang, Tianchen Song, Yongjae Lee, Li Yang, Cheng Peng, Rama Chellappa, and Deliang Fan. LP-3DGS: Learning to Prune 3D Gaussian Splatting. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [21] Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. Eagles: Efficient accelerated 3d gaussians with lightweight encodings. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.
- [22] 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.
- [23] Muhammad Salman Ali, Sung-Ho Bae, and Enzo Tartaglione. ElmgS: Enhancing memory and computation scalability through compression for 3d gaussian splatting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.
- [24] Wieland Morgenstern, Florian Barthel, Anna Hilsman, and Peter Eisert. Compact 3d scene representation via self-organizing gaussian grids. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.
- [25] Simon Niedermayr, Josef Stumpfegger, and Rüdiger Westermann. Compressed 3d gaussian splatting for accelerated novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
- [26] Henan Wang, Hanxin Zhu, Tianyu He, Runsen Feng, Jiajun Deng, Jiang Bian, and Zhibo Chen. End-to-end rate-distortion optimized 3d gaussian representation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.
- [27] Xiangrui Liu, Xinju Wu, Pingping Zhang, Shiqi Wang, Zhu Li, and Sam Kwong. CompGS: Efficient 3d scene representation via compressed gaussian splatting. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, 2024.
- [28] Seungjoo Shin, Jaesik Park, and Sunghyun Cho. Locality-aware Gaussian Compression for Fast and High-quality Rendering. In *International Conference on Learning Representations (ICLR)*, 2025.
- [29] Yu-Ting Zhan, Cheng-Yuan Ho, Hebi Yang, Yi-Hsin Chen, Jui Chiu Chiang, Yu-Lun Liu, and Wen-Hsiao Peng. CAT-3DGS: A Context-Adaptive Triplane Approach to Rate-Distortion-Optimized 3DGS Compression. In *International Conference on Learning Representations (ICLR)*, 2025.
- [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 (CVPR)*, 2024.
- [31] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtaash 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.
- [32] Yufei Wang, Zhihao Li, Lanqing Guo, Wenhan Yang, Alex Kot, and Bihan Wen. ContextGS: Compact 3d gaussian splatting with anchor level context model. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [33] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtaash Harandi, and Jianfei Cai. HAC++: Towards 100X Compression of 3D Gaussian Splatting. *arXiv preprint arXiv:2501.12255*, 2025.
- [34] Junoh Lee, ChangYeon Won, Hyunjun Jung, Inhwan Bae, and Hae-Gon Jeon. Fully explicit dynamic gaussian splatting. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [35] Jiahao Lu, Jiacheng Deng, Ruijie Zhu, Yanzhe Liang, Wenfei Yang, Tianzhu Zhang, and Xu Zhou. Dn-4dgs: Denoised deformable network with temporal-spatial aggregation for dynamic scene rendering. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

- [36] Woong Oh Cho, In Cho, Seoha Kim, Jeongmin Bae, Youngjung Uh, and Seon Joo Kim. 4D Scaffold Gaussian Splatting for Memory Efficient Dynamic Scene Reconstruction. *arXiv preprint arXiv:2411.17044*, 2024.
- [37] Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
- [38] Sangwoon Kwak, Joonsoo Kim, Jun Young Jeong, Won-Sik Cheong, Jihyong Oh, and Munchurl Kim. MoDec-GS: Global-to-Local Motion Decomposition and Temporal Interval Adjustment for Compact Dynamic 3D Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
- [39] 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.
- [40] Diwen Wan, Ruijie Lu, and Gang Zeng. Superpoint Gaussian splatting for real-time high-fidelity dynamic scene reconstruction. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.
- [41] Diwen Wan, Yuxiang Wang, Ruijie Lu, and Gang Zeng. Template-free Articulated Gaussian Splatting for Real-time Reposable Dynamic View Synthesis. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [42] 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.
- [43] Ruijie Zhu, Yanzhe Liang, Hanzhi Chang, Jiacheng Deng, Jiahao Lu, Wenfei Yang, Tianzhu Zhang, and Yongdong Zhang. Motiongs: Exploring explicit motion guidance for deformable 3d gaussian splatting. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [44] 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. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.
- [45] 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.
- [46] 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 *International Conference on Learning Representations (ICLR)*, 2025.
- [47] Jihwan Yoon, Sangbeom Han, Jaeseok Oh, and Minsik Lee. SplineGS: Learning Smooth Trajectories in Gaussian Splatting for Dynamic Scene Reconstruction. In *International Conference on Learning Representations (ICLR)*, 2025.
- [48] Lingzhi Li, Zhen Shen, Zhongshu Wang, Li Shen, and Ping Tan. Streaming Radiance Fields for 3D Video Synthesis. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [49] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [50] 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.
- [51] 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.
- [52] 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/CVF International Conference on Computer Vision (ICCV)*, 2015.
- [53] Neus Sabater, Guillaume Boisson, Benoit Vandame, Paul Kerbiriou, Frederic Babon, Matthieu Hog, Remy Gendrot, Tristan Langlois, Olivier Bureller, Arno Schubert, et al. Dataset and Pipeline for Multi-view Light-field Video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The main claims are made in the abstract and introduction accurately reflect the paper's contributions and scope.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The paper discusses the limitations of the work in Sec. 6.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: The paper provides the full set of assumptions and a complete (and correct) proof.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: The paper will disclose all the code and data needed to reproduce the main experimental results of the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

#### 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have released the code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) in Sec. 5.2 and Appendix A.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: All experimental results of our proposed method and some baseline methods are obtained by averaging three experiments conducted on our sources to ensure validity. However, since some baseline methods are not open-sourced, their results are directly taken from the reported results in the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

#### 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments. The details can be found at Sec. 5.2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

#### 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

#### 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: There is no societal impact of the work performed. This research is fundamentally academic-oriented, aimed at advancing theoretical understanding and promoting disciplinary development within the field. As the work currently lacks substantive practical

applications beyond scholarly contexts, it does not demonstrate measurable societal impact at this stage of investigation.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

#### 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

#### 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The creators or original owners of assets (e.g., code, data, models), used in the paper, are properly credited, and the license and terms of use explicitly are mentioned and properly respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

### 13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

### 14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

### 15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

#### 16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The paper does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

## Appendix

### A More Implementation Details

We implement ReCon-GS upon the open-source codebase of 3D Gaussian Splatting (3DGS) [13]. During initial frame training, we set the spherical harmonic coefficients (SH) to degree 1 to ensure the storage compactness. For all datasets, we terminate the densification process of 3DGS at 5,000 iterations, which significantly alleviates the overfitting issue in streaming 3DGS during initial frame training. Additionally, we adopt the opacity preservation strategy from 4DGS to effectively eliminate oversized Gaussian primitives. The hyperparameter  $\lambda_{noise}$  follows the first-frame configuration from HiCoM [3], while other 3DGS parameters maintain their default settings in 3DGS. Specifically, we train the first frame for 10,000 iterations on the Meet Room dataset, and extend this to 15,000 iterations for both N3DV and PanopticSports datasets.

For subsequent frame optimization, our approach differs from HiCoM’s implementation which uses 250 iterations for deformation field training on Panoptic Sports versus 100 iterations on N3DV and Meet Room datasets. We uniformly apply 100 iterations for deformation field training across all datasets due to our superior ability in capturing the motion of Gaussian primitives. In Phase 2, the refinement of deformed Gaussian primitives, considering the more pronounced surface deformation in PanopticSports dataset compared to other datasets, we implement 400 iterations for intensive refinement on PanopticSports dataset versus 100 iterations for N3DV and MeetRoom datasets. All 3DGS-related hyperparameters remain consistent with those used in initial frame training to ensure methodological coherence.

### B More Ablation Studies

As ReCon-GS represents a novel storage-aware framework designed for high-fidelity real-time rendering in dynamic scene reconstruction, the quantity of Anchor Gaussians per layer in our adaptive hierarchical motion representation structure fundamentally governs the critical trade-off between storage efficiency and rendering quality. Concurrently, the iteration counts allocated to each training stage in subsequent frame training directly determine the training time of our optimization process.

Table 6 investigates the impacts of different iteration configurations across training phases on ReCon-GS performance. The results indicate that prolonging the deformation field training process in Phase 1 can effectively improve rendering quality, however, this training configuration significantly increases computational time. The view-based densification process in Phase 2 demonstrates optimal performance and best storage quality at 100 steps. As evidenced by Table 4 and Table 6, view-based densification effectively address persistent discrepancies between learned motion and scene dynamics, though their impact on overall rendering quality remains less substantial compared to the training of adaptively hierarchical deformation.

We also performed an ablation study on the performance of ReCon-GS, varying the training duration of the initial frame (Frame 0). The detailed results is shown in Table 7. At 2,500 iterations, the suboptimal quality and sparse representation of the initial reconstruction lead to the steepest performance degradation slope. Increasing the training to 5,000 iterations substantially mitigates this quality degradation. Extending the training to 10,000 iterations yields a converged initial frame, which further minimizes quality decay while achieving state-of-the-art performance. Importantly, with the converged initial frame, ReCon-GS requires only a minimal number of incremental Gaussians per frame to ensure high-quality reconstruction. This result strongly validates the superior robustness of

Table 6: Ablation on training steps of our ReCon-GS framework.

Phase 1	Phase 2	PSNR (dB)↑	SSIM ↑	Storage (MB)↓	Train (sec)↓	Rendering (FPS) ↑
50	100	32.40	0.9560	<b>0.44</b>	<b>5.20</b>	248
150		<b>32.76</b>	0.9568	<b>0.44</b>	7.66	249
100	50	32.56	0.9564	0.46	5.22	246
	150	32.64	0.9568	0.49	7.70	249
100	100	32.66	<b>0.9571</b>	<b>0.44</b>	6.44	<b>250</b>

Table 7: **Ablation on the Training Steps of Frame 0.** The Init. 3DG refers to the initial 3D Gaussians constructed from the Frame 0. The Incr. 3DG denotes the incremental Gaussians added in Phase 2 for all following frames.

Training Steps	PSNR (dB) ↑	Frame 0 PSNR (dB) ↑	Slope ↓	Storage (MB) ↓	Init. 3DG Storage (MB) ↓	Incr. 3DG Storage (MB) ↓
2500	31.03	31.59	0.0077	0.46	8.96	42.17
5000	32.24	32.51	0.0029	0.48	11.14	11.57
10000	32.63	32.75	0.0016	0.46	11.12	4.11

Table 8: **First-frame reconstruction results across various scenario configurations.** 3DGStream uses standard 3DGS training, Our Method and HiCoM use standard 3DGS training that incorporates with positional noise injection.

Method	Coffee Martini				Flame Steak				Sear Steak			
	PSNR (dB) ↑	GS Num (k) ↓	Storage (MB) ↓	Train (Sec) ↓	PSNR (dB) ↑	GS Num (k) ↓	Storage (MB) ↓	Train (Sec) ↓	PSNR (dB) ↑	GS Num (k) ↓	Storage (MB) ↓	Train (Sec) ↓
3DGStream	27.62	416	7.60	340	33.85	208	7.60	304	29.25	214	7.60	317
HiCoM	29.73	348	83.00	368	34.27	187	44.55	334	33.96	191	45.60	317
<b>Ours</b>	29.67	353	16.17	354	31.20	189	8.65	324	33.90	191	8.76	338

ReCon-GS against error accumulation and underscores the critical impact of initial frame quality in streaming frameworks.

## C More Detailed Results

### C.1 Quantitative Results

#### C.1.1 Initial 3DGS Comparison

For streaming 3DGS reconstruction frameworks, the initial frame training plays a pivotal role in determining the reconstruction quality of subsequent frames. Table 8 presents comparative metrics of average initial frame training quality and corresponding storage consumption across three benchmark datasets. While both ReCon-GS and HiCoM implement positional noise injection during initial Gaussian optimization – a strategic commonality that differentiates them from 3DGStream – our framework demonstrates superior rendering fidelity. This configuration, combined with our first-frame spherical harmonic (SH) coefficient restriction to degree 1, enables ReCon-GS to achieve more compact Gaussian representations compared to HiCoM’s SH degree 3 implementation, significantly enhancing storage efficiency without compromising visual quality.

#### C.1.2 Per-Scene Results

Table 9, Table 10, and Table 11 present comprehensive comparative results between ReCon-GS and recent state-of-the-art methods across three benchmark datasets under specific scene.

#### C.1.3 More Results

Figure 6 demonstrates the superior rate-distortion performance of our method compared to existing SOTA approaches across multiple scenarios. Table 12 presents the detailed results under different hierarchical anchor densities. By modulating hierarchical anchor density  $\{N_{anchor}^{(l)}\}_{l=1}^3$  and leveraging the high-efficiency characteristics of the Adaptively Hierarchical Motion Presentation paradigm, ReCon-GS maintains robust rendering quality under varying storage constraints.

### C.2 Anchor Gaussian Visualization

We visualize the distribution of Anchor Gaussians at different levels for ReCon-GS and HiCoM, with the results shown in Figure 7. As shown in the Figure 7, compared to the baseline method HiCoM, ReCon-GS assigns more Anchor Gaussians to motion-rich and texture-rich regions (*e.g.*, people, VR glasses, flowers). This is primarily due to the density-based Anchor Gaussian assignment mechanism used by ReCon-GS. As a result, ReCon-GS is able to perform more refined rigid motion representation in areas with rich textures, thus improving the overall rendering quality.

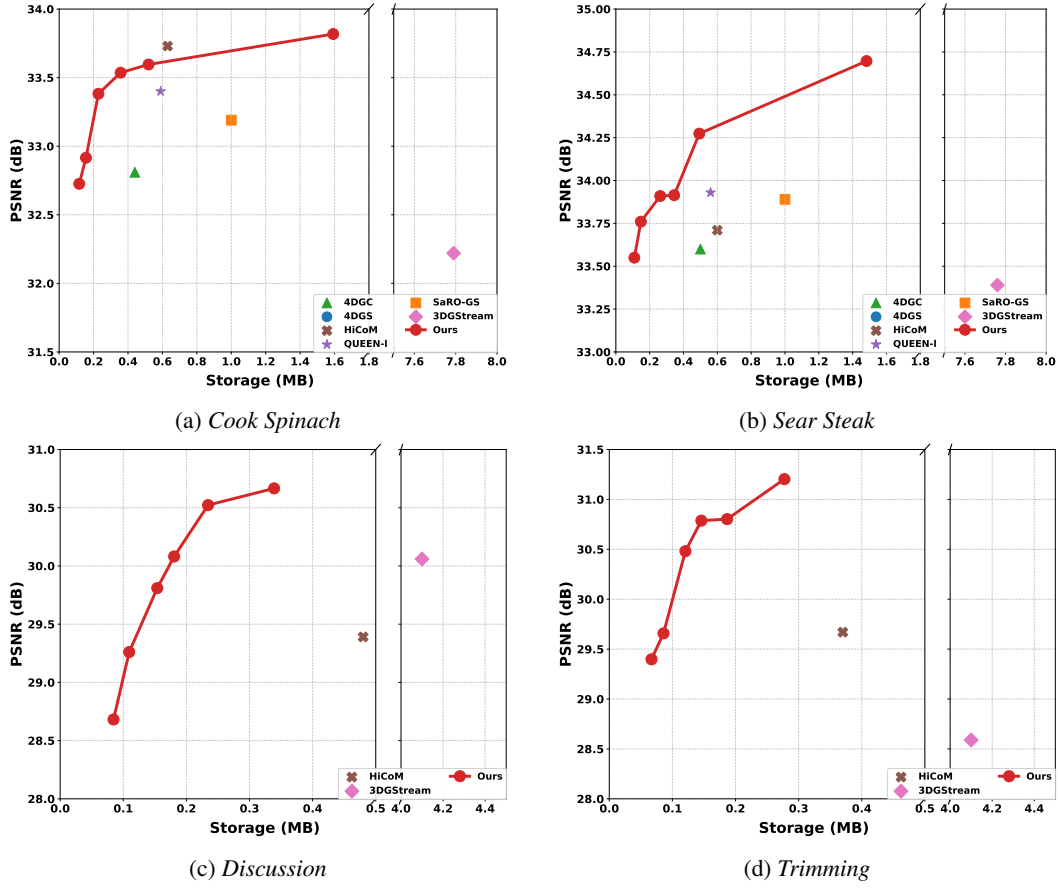


Figure 6: **Rate-distortion performance across diverse scenarios.** As evidenced by comparative evaluations, our framework not only achieves superior compression efficiency compared to state-of-the-art approaches [1, 6, 3, 4, 5, 2], but also demonstrates unique adaptability in storage-aware fidelity optimization - a critical capability for practical deployment scenarios where resource constraints and reconstruction quality must be jointly optimized.

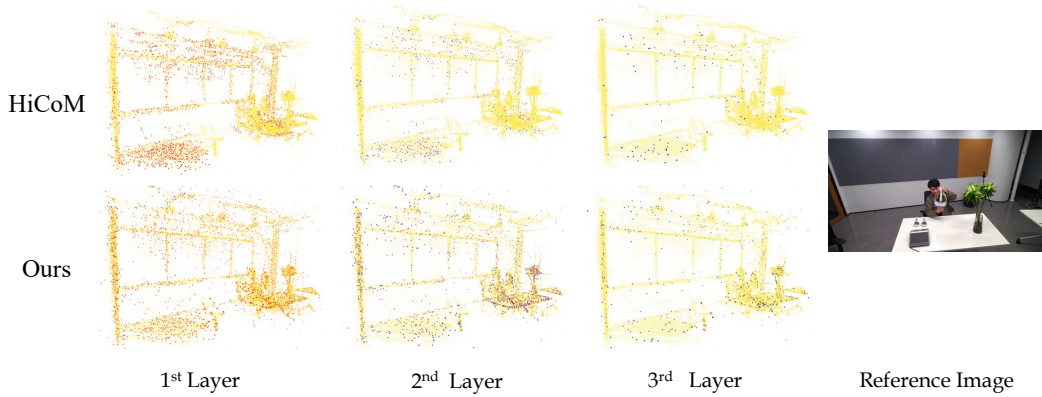


Figure 7: **The Visualization of Anchor Gaussian Distribution.** 1<sup>st</sup> Layer is the finest layer, and 3<sup>rd</sup> Layer is the coarsest layer. The yellow points represent the General Gaussian, while the red, purple, and blue points represent the Anchor Gaussians for their respective layers.

Table 9: **Per-scene quantitative results on the N3DV dataset.** The method with  $\dagger$  is reproduced by us through official code in the same experimental environment.

Category	Method	Coffee Martini			Cook Spinach			Cut Roasted Beef		
		PSNR $\uparrow$	SSIM $\uparrow$	Storage $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	Storage $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	Storage $\downarrow$
Offline	4DGS $^\dagger$ [1]	30.10	0.935	0.30	31.15	0.958	0.30	32.60	0.951	0.30
	SaRO-GS [5]	28.96	-	1.0	33.19	-	1.0	33.91	-	1.0
Online	3DGStream $^\dagger$ [6]	27.75	0.917	7.85	32.22	0.957	7.79	32.67	0.957	7.76
	4DGC [2]	27.98	-	0.58	32.81	-	0.44	33.03	-	0.47
	QUEEN-I [4]	28.38	0.915	1.17	33.40	0.956	0.59	34.01	0.959	0.57
	HiCoM $^\dagger$ [3]	28.67	0.925	0.82	33.73	0.962	0.63	32.73	0.963	0.60
	<b>Ours</b>	30.14	0.938	0.64	33.54	0.961	0.35	33.92	0.966	0.36
	Method	Flame Salmon			Flame Steak			Sear Steak		
		PSNR $\uparrow$	SSIM $\uparrow$	Storage $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	Storage $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	Storage $\downarrow$
Offline	4DGS $^\dagger$ [1]	30.21	0.934	0.30	33.49	0.961	0.30	30.64	0.961	0.30
	SaRO-GS [5]	29.14	-	1.0	33.83	-	1.0	33.89	-	1.0
Online	3DGStream $^\dagger$ [6]	28.61	0.924	7.83	33.47	0.966	7.79	33.39	0.965	7.76
	4DGC [2]	28.49	-	0.51	33.58	-	0.44	33.60	-	0.50
	QUEEN-I [4]	29.25	0.923	1.00	34.17	0.962	0.59	33.93	0.962	0.56
	HiCoM $^\dagger$ [3]	29.70	0.932	0.81	33.92	0.969	0.60	33.71	0.968	0.60
	<b>Ours</b>	30.43	0.938	0.61	34.00	0.969	0.33	33.91	0.966	0.35

Table 10: **Per-scene quantitative results on the Meet Room dataset.** The method with  $\dagger$  is reproduced by us through official code in the same experimental environment.

Category	Method	Discussion			Trimming			Vrheadset		
		PSNR $\uparrow$	SSIM $\uparrow$	Storage $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	Storage $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	Storage $\downarrow$
Online	3DGStream $^\dagger$ [6]	30.06	0.945	4.10	28.59	0.939	4.10	29.25	0.941	4.10
	HiCoM $^\dagger$ [3]	29.39	0.941	0.48	29.67	0.945	0.37	29.65	0.947	0.30
	<b>Ours</b>	30.67	0.955	0.35	31.20	0.956	0.28	30.64	0.953	0.30

Table 11: **Per-scene quantitative results on the PanopticSports dataset.** The method with  $\dagger$  is reproduced by us through official code in the same experimental environment.

Category	Method	Basketball			Boxes			Football		
		PSNR $\uparrow$	SSIM $\uparrow$	Storage $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	Storage $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	Storage $\downarrow$
Offline	4DGS $^\dagger$ [1]	26.99	0.912	0.34	27.76	0.922	0.32	27.20	0.922	0.36
Online	Dynamic3DGS [42]	28.22	0.910	0.74	29.46	0.910	0.74	28.49	0.910	0.74
	3DGStream $^\dagger$ [6]	20.08	0.815	8.10	23.76	0.861	8.10	22.80	0.856	8.10
	HiCoM $^\dagger$ [3]	28.49	0.915	2.06	29.03	0.922	2.05	29.64	0.926	2.01
	<b>Ours</b>	28.78	0.921	0.81	28.93	0.926	0.80	30.01	0.931	0.77
	Method	Juggle			Softball			Tennis		
		PSNR $\uparrow$	SSIM $\uparrow$	Storage $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	Storage $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	Storage $\downarrow$
Offline	4DGS $^\dagger$ [1]	26.72	0.923	0.32	26.88	0.923	0.31	27.64	0.925	0.31
Online	Dynamic3DGS [42]	29.48	0.920	0.74	28.43	0.910	0.74	28.11	0.910	0.74
	3DGStream $^\dagger$ [6]	26.25	0.891	8.10	22.22	0.858	8.10	22.99	0.862	8.10
	HiCoM $^\dagger$ [3]	29.28	0.932	2.02	29.61	0.928	1.97	28.97	0.925	2.05
	<b>Ours</b>	29.71	0.938	0.79	29.61	0.933	0.77	28.97	0.929	0.79

Table 12: Quantitative Results under different hierarchical anchor density  $\{N_{anchor}^{(l)}\}_{l=1}^3$ .

	Hierarchical Anchor Density					
	1/16	1/24	1/32	1/48	1/64	1/96
Storage (MB/Frame)	0.63	0.44	0.37	0.28	0.19	0.14
PSNR (dB)	32.72	32.66	32.50	32.39	32.31	32.11

### C.3 Qualitative Results

Figure 8 provides expanded quantitative comparisons between ReCon-GS and HiCoM on the N3DV dataset, highlighting critical performance differentials in dynamic scene reconstruction.

Figure 9 presents extended visual comparisons on the MeetRoom dataset, contrasting geometric fidelity and temporal consistency across competing frameworks.

Figure 10 details supplementary visual evaluations on the PanopticSports dataset, emphasizing our method’s enhanced capability in handling complex motion patterns under real-world capture conditions.

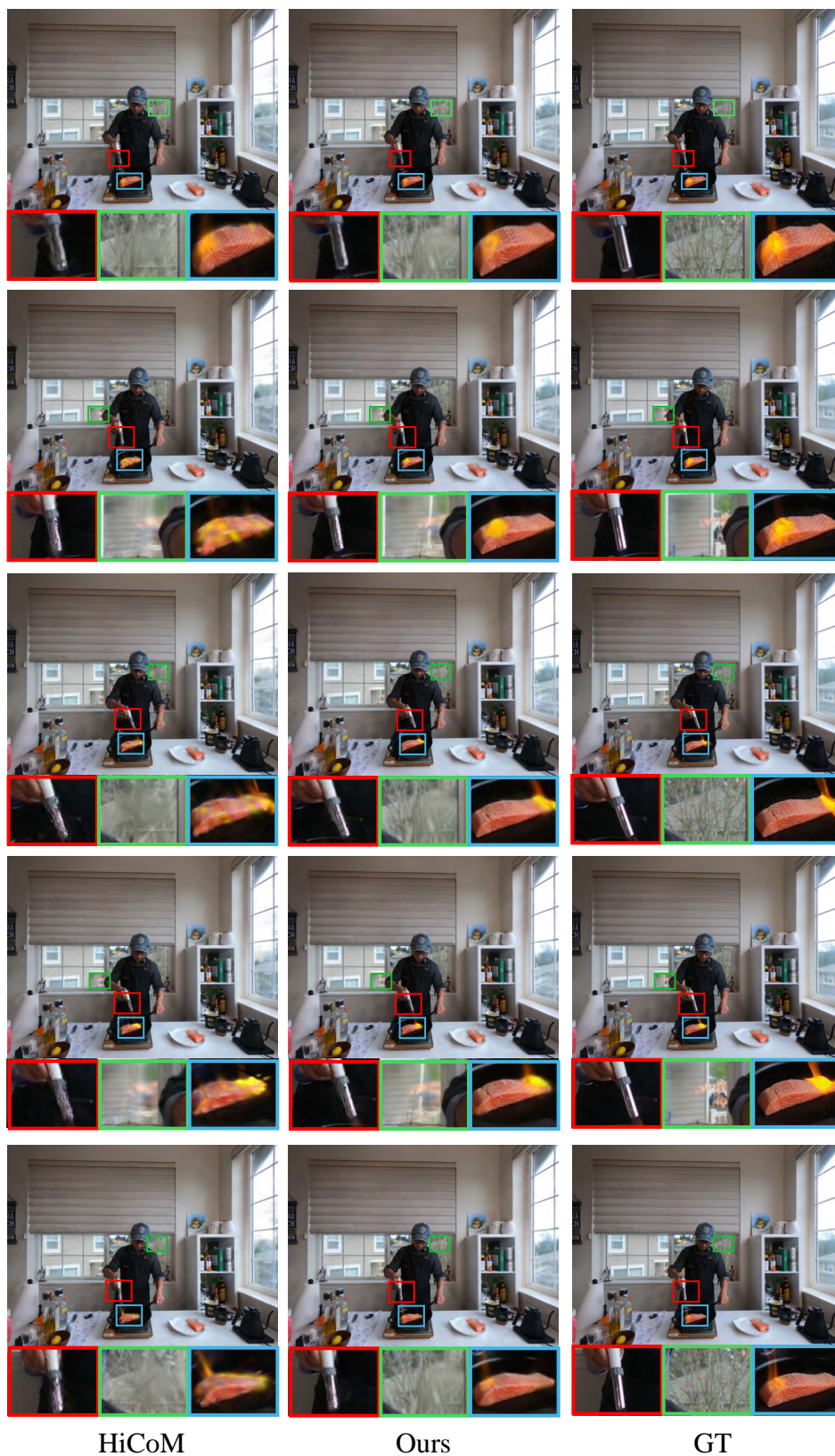


Figure 8: Qualitative results on N3DV Dataset.

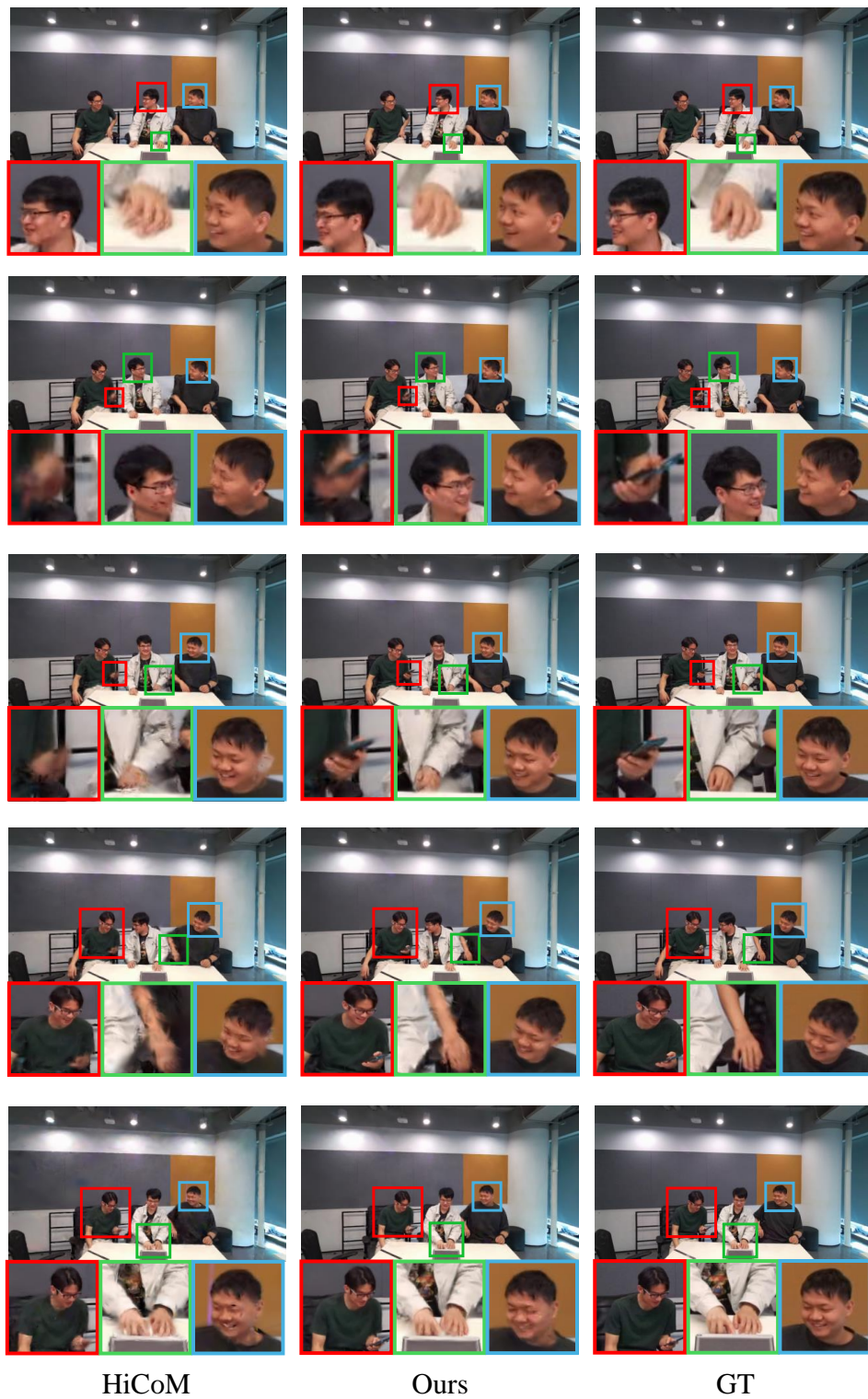


Figure 9: Qualitative results on Meet Room Dataset.

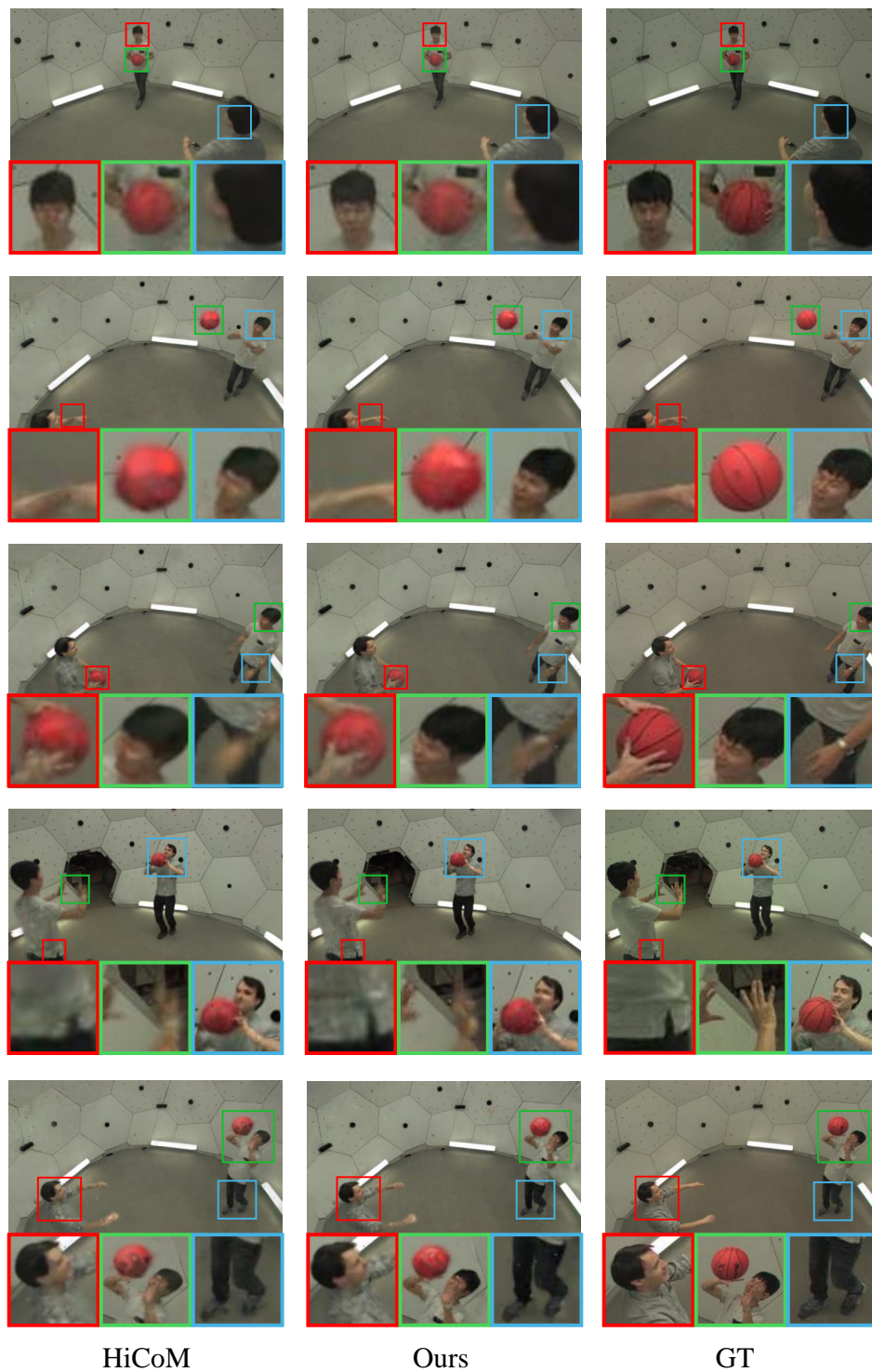


Figure 10: Qualitative results on PanopticSports Dataset.