

Motion Matters: Compact Gaussian Streaming for Free-Viewpoint Video Reconstruction

Jiacong Chen^{1,2} Qingyu Mao³ Youneng Bao⁴ Xiandong Meng⁵ Fanyang Meng⁵
Ronggang Wang^{5,6} Yongsheng Liang^{1,2}✉

¹College of Applied Technology, Shenzhen University, Shenzhen, China

²College of Big Data and Internet, Shenzhen Technology University, Shenzhen, China

³College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China

⁴Department of Computer Science, City University of Hong Kong, Hong Kong, China

⁵Pengcheng Laboratory, Shenzhen, China

⁶School of Electronic and Computer Engineering, Peking University, Shenzhen, China

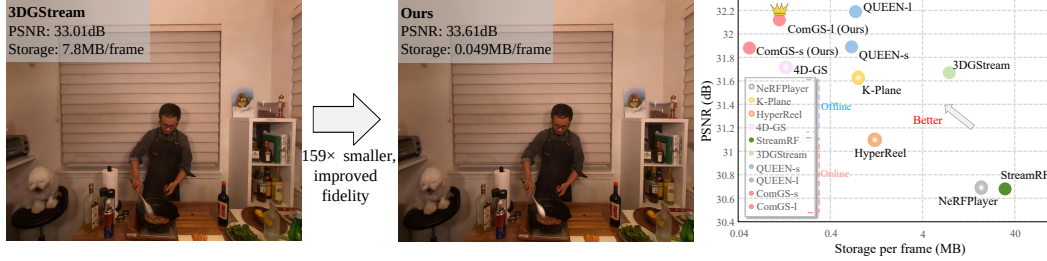


Figure 1: **Left:** Experimental results on N3DV dataset [1] showcase the effectiveness of our method, which reduces the storage requirement of 3DGSStream [2] by $159\times$, with enhanced visual quality. **Right:** Comparison with existing methods in storage and reconstruction fidelity. Hollow circles denote offline methods, while solid circles represent online methods.

Abstract

3D Gaussian Splatting (3DGS) has emerged as a high-fidelity and efficient paradigm for online free-viewpoint video (FVV) reconstruction, offering viewers rapid responsiveness and immersive experiences. However, existing online methods face challenge in prohibitive storage requirements primarily due to point-wise modeling that fails to exploit the motion properties. To address this limitation, we propose a novel Compact Gaussian Streaming (ComGS) framework, leveraging the locality and consistency of motion in dynamic scene, that models object-consistent Gaussian point motion through keypoint-driven motion representation. By transmitting only the keypoint attributes, this framework provides a more storage-efficient solution. Specifically, we first identify a sparse set of motion-sensitive keypoints localized within motion regions using a view-space gradient difference strategy. Equipped with these keypoints, we propose an adaptive motion-driven mechanism that predicts a spatial influence field for propagating keypoint motion to neighboring Gaussian points with similar motion. Moreover, ComGS adopts an error-aware correction strategy for key frame reconstruction that selectively refines erroneous regions and mitigates error accumulation without unnecessary overhead. Overall, ComGS achieves a remarkable storage reduction of over $159\times$ compared to 3DGSStream and $14\times$ compared to the SOTA method QUEEN, while maintaining competitive visual fidelity and rendering speed. Project page: <https://chenjiacong-1005.github.io/ComGS/>.

1 Introduction

Reconstructing free-viewpoint video (FVV) from multi-view videos captured by cameras with known poses has attracted growing interest in the field of computer vision and graphics. FVV exhibits great potential as a next-generation visual medium that enables immersive and interactive experiences, with broad application in virtual and augmented reality (VR/AR) applications [3].

Recently, 3D Gaussian Splatting (3DGS) has become a promising method for FVV reconstruction, due to its significant advancements in real-time rendering and high-fidelity view synthesis. These approaches typically fall into two categories: 1) incorporating temporal function into Gaussian primitives and optimizing directly [4–6], and 2) applying a deformation field to capture the spatio-temporal transformations of canonical Gaussians [7–11]. While these FVV reconstructions accurately represent dynamic scenes, they are trained in an offline manner and require transmitting the full set of reconstructed parameters prior to rendering.

In contrast, by enabling per-frame training and progressive transmission, online FVV reconstruction allows immediate playback without the overhead of full-scene preloading. As a pioneer work, 3DGStream [2] extends 3DGS to online FVV reconstruction using InstantNGP [12] to model the geometric transformation frame-by-frame. While achieving impressive rendering speed, its structural constraint hinders the volumetric representation performance and degrades the visual quality. Building on this paradigm, subsequent works [13, 14] enhance model expressiveness through explicitly optimizing Gaussian attribute residuals, achieving competitive synthesis quality and higher robustness. However, the storage demands of these methods remain prohibitively high for real-time transmission, with reconstructed data typically exceeding 20 MB per second.

In this paper, we aim to design a storage-efficient solution for FVV streaming that minimizes bandwidth requirements and enables real-time transmission. In online FVV reconstruction, since dynamic scenes contain a large proportion of static regions, the key to efficient reconstruction lies in motion modeling. Our first insight, therefore, is to only model the Gaussian attribute residuals in the motion regions, which eliminates the unnecessary updates in static regions. Building on motion modeling, we note that scene motion tends to be consistent, where Gaussian points associated with the same object typically exhibit the same or similar motion in dynamic scene representation. Our second insight, based on this observation, is to use a shared motion representation to model the attribute residuals with similar motion. This contrasts with existing online methods [2, 13] that utilize point-wise strategy to update the attribute residuals in motion regions, and the result is motion redundancy elimination and more compact storage. Lastly, we exploit a key frame fine-tune strategy to handle the error accumulation brought by non-rigid motion and novel objects emergence.

Specifically, to accomplish this, we propose a Compact Gaussian Streaming (ComGS) framework that leverages a set of keypoints ($= 200$), significantly fewer than the full set of Gaussian points ($\approx 200K$), to holistically model motion regions at each timestep. ComGS begins with a motion-sensitive keypoints selection through a view-space gradient difference strategy. This ensures that the selected keypoints are accurately positioned within motion regions and prevents redundant or incorrect modeling of static areas. Subsequently, we design an adaptive motion-driven mechanism that defines a keypoint-specific spatial influence field, with which neighboring Gaussian points can share the motion of the keypoint. Unlike conventional k-nearest neighbor (KNN) methods [15, 16], the spatial influence field can accommodate the complexity and variability of motion structure in dynamic scenes, so that keypoints can more accurately drive the motion of the surrounding region. Finally, to mitigate error accumulation in a compact and effective manner, we propose an error-aware correction strategy for key frame reconstruction that selectively updates only those Gaussians with reconstruction errors.

Our major contributions can be summarized as follow:

- We introduce a motion-sensitive keypoint selection to accurately identify keypoints within motion regions, and an adaptive motion-driven mechanism that effectively propagates motion to neighboring points. These leverage the locality and consistency of motion and achieve a more storage-efficient solution for online FVV reconstruction.
- We propose an error-aware correction strategy for key frame reconstruction that mitigates error accumulation over time by selectively updating Gaussian points with reconstruction errors, which ensures long-term consistency and minimizes redundant correction.

- Experiments on two benchmark datasets show that the effectiveness of our method and its individual components. Our method achieves a compression ratio of $159\times$ over the 3DGSStream and $14\times$ over state-of-the-art model QUEEN, enabling real-time transmission while preserving competitive reconstruction quality and rendering speed.

2 Related work

2.1 Dynamic Gaussian Splatting

Recently, 3D Gaussian Splatting (3DGS) [17–22] has attracted great attention in Free-viewpoint video (FVV) reconstruction for its high photorealistic performance and real-time rendering speed. Several works [4–6, 23, 24] expand temporal variation as a function and optimize directly for modeling Gaussian attributes across frames. For instance, 4D Gaussian Splatting [4] incorporates time-conditioned 3D Gaussians and auxiliary components into 4D Gaussians, while ST-GS [6] models the transformation of structural attributes and opacity as a temporal function to represent scene motions. To support long FVVs representation, TGH [23] introduces a multi-level hierarchy of 4D Gaussian primitives that exploits various degrees of temporal redundancy in dynamic scenes. While these time variant-based methods achieve superior rendering efficiency, they often suffer from prohibitive storage requirements. Other works [8, 11, 25–27] employ vanilla 3D Gaussians as a canonical space and a deformation field to represent the dynamic scene. In this category, 4D-GS [8] utilizes hexplanes [28], six orthogonal planes, as latent embeddings and deliver them into a small MLP to deform temporal transformation of Gaussian points, achieving efficient computational complexity and lightweight storage requirement. Building upon this, GD-GS [11] further improves scene modeling accuracy by incorporating geometric priors, which provides a more structured and precise representation of dynamic scene. Among them, both SC-GS [15] and SP-GS [16] adopt sparse control points to control scene motion using a k-nearest neighbor (KNN) [29] strategy for motion modeling. While these methods achieve notable improvements in computational efficiency and rendering speed, they are designed for offline FVV reconstruction and do not support frame-by-frame delivering. Additionally, motion-insensitive control point selection and scale-agnostic KNN motion modeling lead to redundant representation of static regions and reduced deformation accuracy in dynamic scenes. Our online method addresses these limitations by selecting keypoints from motion regions at each timestep and modeling motion with awareness of local motion scales, which enables more accurate and efficient modeling of online FVV.

2.2 Online Free-Viewpoint Video Reconstruction

Compared to the offline methods, online reconstruction enables FVV to be incrementally trained and transmitted in a per-frame manner, which allows users to preview or interact immediately with the video content. Leveraging the high-fidelity view synthesis capabilities of Neural Radiance Field (NeRF) [12, 30–36], a set of studies have explored NeRF-based methods [37–41] for online FVV reconstruction, such as StreamRF [37], VideoRF [38] and TeTriRF [40]. Despite advanced visual quality, NeRF-based methods are hindered by their limited rendering speeding of implicit structure, which limits their practical applications.

With the utilization of 3DGS [17], 3DGSStream [2] introduces a hash-based MLP to encode the position and rotation transformation of Gaussian points at each frame, and designs an adaptive Gaussians addition strategy for novel objects across frames. Based on this paradigm, QUEEN [13] proposes a Gaussian residual-based framework for model expressiveness enhancement and a learned quantization-sparsity framework for residuals compression. HiCoM [14] designs a hierarchical coherent motion mechanism to effectively capture and represent scene motion for fast and accurate training. To deploy into mobile device, V^3 [42] presents a novel approach that compresses Gaussian attributes as a 2D video to facilitate hardware video codecs. IGS [43] proposes a generalized anchor-driven Gaussian motion network that learns residuals with a single step, achieving a significant improvement of training speed. Nevertheless, these methods face challenge in real-time transmission, due to their substantial storage requirements. This overhead mainly stems from redundant updates of static Gaussian points across frames, as well as repeated modeling of Gaussian points with similar motion. Our study exploits the locality and consistency of motion by leveraging motion-sensitive keypoints to adaptively drive motion regions, and this avoids redundant storage and transmission.

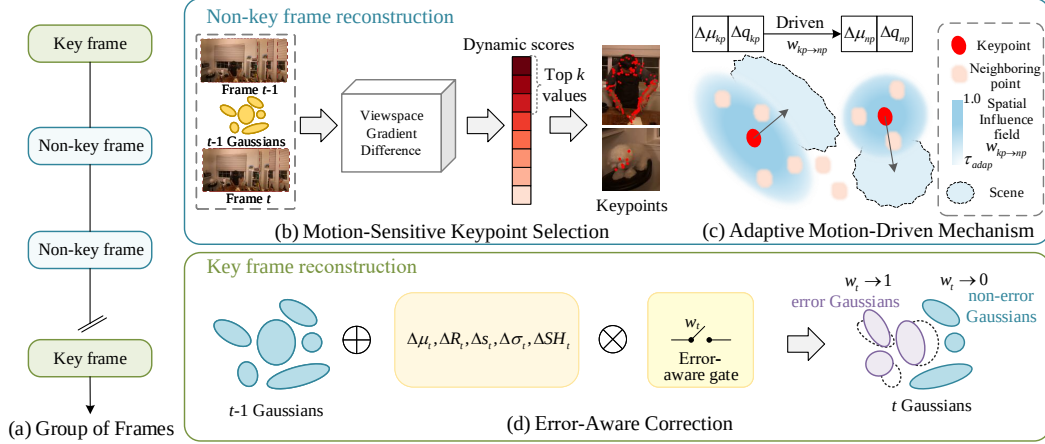


Figure 2: The overall pipeline of ComGS framework. (a) The reconstruction process starts from the first frame initialized using vanilla 3DGS [17]. Subsequent frames are organized into groups of frames (GoFs). For non-key frames, (b) we begins with a motion-sensitive keypoint selection using a viewspace gradient difference strategy, (c) and utilizes an adaptive motion-driven mechanism to control neighboring points motion. For key frames, (d) an error-aware correction strategy is introduced to mitigate the error accumulation across frames.

2.3 3D Gaussian Representation Compression

Despite 3DGS-based methods achieve impressive performance in novel view synthesis [17, 44], the massive size of Gaussian points hinder them for efficient storage and transmission. Several studies propose a variety of compression techniques for reducing the required storage, which can be categorized into either post-processing-based [19–21, 45, 13] or neural contextual coding-based methods [10, 46–49]. Post-processing-based approaches include removing unimportant Gaussian points [19, 20], pruning spherical harmonic coefficients [19, 21], and applying vector quantization [13, 45] to compress the parameter representation. The latter methods utilize sophisticated entropy modeling to accurately predict probability distributions that exploit global context for compressing 3D Gaussian representation more effectively. In this paper, we focus on leveraging the locality and consistency of motion in dynamic scene and mitigating the redundancy reconstruction on static and similar motion regions, introducing a novel and more compact method for online FVV reconstruction.

3 Methods

Our goal is to reconstruct and transmit FVV in a storage-efficient and streaming manner. To achieve it, we propose a Compact Gaussian Streaming (ComGS) framework for online FVV reconstruction, as illustrated in Fig. 2. First, ComGS begins with a motion-sensitivity keypoint selection using a viewspace gradient difference, ensuring subsequent motion control learning (Sec. 3.2). Second, we develop an adaptive motion-driven mechanism that applies a spatial influence field to control neighboring point motion (Sec. 3.3). Third, we devise an error-aware correction strategy for key frame reconstruction to mitigate error accumulation brought by non-rigid motion and novel objects emergence in online reconstruction (Sec. 3.4). Finally, we introduce our compression techniques and optimization process in Sec. 3.5.

3.1 Preliminary

3DGS [17] models a 3D scene as a large amount of anisotropic 3D Gaussian points in world space as an explicit representation. The central position and geometric shape of each Gaussian point i in world space are defined by a mean vector μ_i and covariance matrix Σ_i , mathematically represented as:

$$G_i(x) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma_i^{-1}(\mathbf{x} - \mu_i)\right) \quad (1)$$

For differentiable optimization, the covariance matrix Σ_i is decoupled into a scaling matrix S_i and a rotation matrix R_i . Each Gaussian point is characterized by its color c_i and opacity σ_i . For novel view synthesis, the covariance matrix Σ'_i in camera coordinate is given as:

$$\Sigma'_i = \mathbf{J}\mathbf{W}\Sigma_i\mathbf{W}^T\mathbf{J}^T \quad (2)$$

where $\mathbf{J}$ is the Jacobian of the affine approximation of the perspective projection and $\mathbf{W}$ represents the view transformation matrix mapping world coordinates.

During rendering, the Gaussian points are initially projected into viewing plane, and the final color C can be obtained by α -blending of the N ordered 3D Gaussian points overlapping the pixel as:

$$C = \sum_{i=1}^N c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \quad (3)$$

where α_i represents the blending weight of the i^{th} Gaussian point.

3.2 Motion-Sensitive Keypoint Selection

Establishing an effective keypoint-driven motion representation necessitates to select appropriate keypoints. Considering motion locality, keypoints should be located in motion regions, which avoids redundant modeling in static areas and enables accurate modeling of complex motions

Thus, inspired by [13], we propose a motion-sensitive keypoint selection based on view-space gradient difference (Fig. 2 (b)). The core idea is to identify the dynamic Gaussian by the gradient change of rendering loss in inter-frames, and based on the gradient values, the k Gaussian points with the largest gradient are selected as keypoints. Specifically, following the gradient computation in 3DGS [17], we compute gradients using the previous Gaussian positions p_{t-1} , the rendered images $\hat{I}_{t-1}$, the reconstruction loss $\mathcal{L}_{recon}$, and the ground-truth images I_{t-1} and I_t :

$$\mathcal{G}_{t-1} = \frac{\partial \mathcal{L}_{recon}^{t-1}}{\partial p_{t-1}}, \quad \mathcal{L}_{recon}^{t-1} = \mathcal{L}_{recon}(\hat{I}_{t-1}, I_{t-1}) \quad (4)$$

$$\mathcal{G}_t = \frac{\partial \mathcal{L}_{recon}^t}{\partial p_{t-1}}, \quad \mathcal{L}_{recon}^t = \mathcal{L}_{recon}(\hat{I}_t, I_t) \quad (5)$$

Dynamic significance scores $\Delta\mathcal{G}_t \in \mathbb{R}^N$ (N is the number of Gaussians) were calculated by means of absolute values of gradient differences:

$$\Delta\mathcal{G}_t = \frac{1}{V} \sum_{v=1}^V |\mathcal{G}_t^{(v)} - \mathcal{G}_{t-1}^{(v)}| \quad (6)$$

where V is the number of the training viewpoints. Finally, we select the top k high dynamic significance scores from all Gaussian points as keypoints $\mathcal{K}_t$ at timestamp t . Selecting the top- k Gaussian points with the highest dynamic scores not only identifies those located in motion regions, but also naturally allocates more keypoints to the areas with complex motion, facilitating more accurate modeling of such regions.

In this paper, for a balance of training efficiency and reconstructed quality, we set $k = 200$.

3.3 Adaptive Motion-Driven Mechanism

Equipped with the selected keypoints $\mathcal{K}_t$ at current timestep, the next step is to determine which neighboring points are controlled by these keypoints, and apply their transformations to drive the motion of the controlled neighboring points. Previous works [15, 16] employ k-nearest neighbor (KNN) [29] search to predict the motion of each Gaussian points, showing advanced results in monocular synthetic video reconstruction, but they do not fully consider unnecessary modeling in static region and motion scale difference, which leads to computational redundancy and inaccurate representation.

In contrast, we propose an adaptive motion-driven mechanism that enables each keypoint to drive neighboring points through a spatial influence field, as illustrated Fig. 2 (c). Specifically, motivated

by [17], for each keypoints $\mathcal{K}_t^i$ at t timestep, we initialize a quaternion $q_{adap}^i \in \mathbb{R}^4$ and a scaling vector $s_{adap}^i \in \mathbb{R}^3$ to compute the spatial influence field $\Sigma_{adap}^i \in \mathbb{R}^{3 \times 3}$. For a neighboring Gaussian point G_j with position μ_j , its distance to keypoint $\mathcal{K}_t^i$ is given by $d_{ij} = \mu_j - \mu_{\mathcal{K}_t^i}$. The influence weight is then computed as:

$$w_{ij} = \exp\left(-\frac{1}{2}d_{ij}^\top(\Sigma_{adap}^i)^{-1}d_{ij}\right) \quad (7)$$

If w_{ij} exceeds a predefined threshold τ_{adap} , the Gaussian G_j is considered to be controlled by keypoint $\mathcal{K}_t^i$:

$$\mathcal{C}_t^i = \{G_j \mid w_{ij} \geq \tau_{adap}\} \quad (8)$$

where $\mathcal{C}_t^i$ denotes the set of Gaussian points controlled by keypoint $\mathcal{K}_t^i$.

To model motion, each keypoint $\mathcal{K}_t^i$ is further assigned a learnable translation offset $\Delta\mu_{\mathcal{K}_t^i} \in \mathbb{R}^3$ and a rotation represented by a quaternion $\Delta q_{\mathcal{K}_t^i} \in \mathbb{R}^4$. For a Gaussian G_j controlled by multiple keypoints $\{\mathcal{K}_t^i\}_{i \in \mathcal{I}_t^j}$, its overall motion is computed by aggregating the motions of its associated keypoints, weighted by their influence scores w_{ij} :

$$\Delta\mu_t^j = \sum_{i \in \mathcal{I}_t^j} w_{ij} \cdot \Delta\mu_{\mathcal{K}_t^i}, \quad \Delta q_t^j = \sum_{i \in \mathcal{I}_t^j} w_{ij} \cdot \Delta q_{\mathcal{K}_t^i} \quad (9)$$

where $\Delta\mu_t^j$ and Δq_t^j indicate the transformation of Gaussian j at t timestep, and $\mathcal{I}_t^j$ represents the set of keypoints that control the motion of Gaussian j .

By leveraging a compact set of keypoints with spatial influence fields, our method enables accurate and efficient control of Gaussian motions at each frame. Since Gaussians share motion attributes through keypoints, only 14 parameters per keypoint are required, significantly reducing storage demands and mitigating data redundancy.

3.4 Error-Aware Corrector

By using keypoints to drive scene motion, we model the transformation of Gaussian points from the previous frame to the current frame with an extremely compact parameters. Nevertheless, keypoint-based motion controlling only supports to represent rigid motion effectively and faces challenge to handle non-rigid motion and novel objects emergence, which results in error accumulation across frames.

A straightforward solution to mitigate error accumulation and ensure accurate long-term FVV representations is to separate the video into frame groups and update the attributes of all Gaussian points at key frames. However, this strategy would lead to a substantial of unnecessary parameters updating, since most of parameters are already correctly representing the scene and do not require modification. To mitigate error accumulation in a compact and efficient manner, we propose an error-aware corrector strategy that only finetunes the Gaussian points with detected errors, significantly decreasing storage demands and promoting more accurate scene reconstruction, as illustrated in Fig. 2 (d).

Specifically, given a video sequence, we select a key frame every s frames, forming the key frame sequence $\{f_s, f_{2s}, \dots, f_{ns}\}$, as shown in Fig. 2 (a). Note that in this paper, key frames are used for error correction, and only the first frame of the video sequence is independently reconstructed. The remaining frames are reconstructed by keypoints driven. During key frame reconstruction, given the attributes of a Gaussian point at previous timestep $\theta_i^{t-1} : (\mu_i^{t-1}, q_i^{t-1}, s_i^{t-1}, \sigma_i^{t-1}, c_i^{t-1})$, we introduce a set of learnable parameters $\Delta\theta_i^t$ to model the attribute residuals. To identify which Gaussian points require correction, we predict a learnable mask m_i for each point. A sigmoid function is used to map m_i to the range $(0, 1)$, which refers as a soft mask:

$$m_i^{soft} = \text{Sigmoid}(m_i), m_i^{soft} \in (0, 1) \quad (10)$$

Similar to [20, 21], the soft mask is binarized into a hard mask using a predefined threshold ϕ_{thres} , where the non-differentiable binarization is handled with the straight-through estimator (STE) to enable gradient flow, represented as:

$$m_i^{hard} = sg(\mathbb{1}(m_i^{soft} > \phi_{thres}) - m_i^{soft}) + m_i^{soft}, m_i^{hard} \in \{0, 1\} \quad (11)$$

Table 1: Quantitative comparisons on Neural 3D Video (N3DV) [1] and MeetRoom datasets [37].

Dataset	Category	Method	PSNR (dB) $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Storage (MB) $\downarrow$	Training (sec) $\downarrow$	Rendering (FPS) $\uparrow$
N3DV	Offline	NeRFPlayer [41]	30.69	0.932	0.209	17.10	72	0.05
		HyperReel [52]	31.10	0.928	-	1.20	104	2.00
		4D-GS [8]	31.15	0.964	0.149	0.13	8	34
		SpaceTime [6]	32.05	0.948	-	0.67	20	140
	Online	StreamRF [37]	30.68	-	-	31.4	15	8.3
		TeTriRF [40]	30.43	0.906	0.248	0.06	39	4
		3DGStream [2]	31.67	0.941	0.140	7.80	8.5	261
		QUEEN-s [13]	31.89	0.945	0.139	0.68	4.65	345
		QUEEN-l [13]	32.19	0.946	0.136	0.75	7.9	248
		ComGS-s (ours)	31.87	0.943	0.132	0.049	37	91
		ComGS-l (ours)	32.12	0.945	0.129	0.106	43	147
	Static	I-NGP [53]	28.10	-	-	48.2	66	4.1
		3DG-S [17]	31.31	-	-	21.1	156	571
MeetRoom	Online	StreamRF [37]	26.72	-	-	9.0	10.2	10
		3DGStream [2]	30.79	0.950	0.188	4.1	4.9	350
		QUEEN-s [13]	31.14	0.954	0.173	0.45	3.8	421
		ComGS-s (ours)	31.49	0.955	0.171	0.028	28.3	98

where $\mathbb{1}$ is the indicator function and sg indicates the stop gradient operation. Then, the m_i^{hard} is applied to the attribute residuals before rendering, followed as:

$$\theta_i^t = \theta_i^{t-1} + m_i^{hard} \Delta \theta_i^t \quad (12)$$

Meanwhile, we define a optimized function to regulate the perceptual error while encouraging sparse residual updates:

$$\mathcal{L}_{error} = \frac{1}{N} \sum_i m_i^{soft} \quad (13)$$

where N is the number of all Gaussian points. After optimization for the current key frame, only the attribute residuals $\Delta \hat{\theta}^t = \{\Delta \theta_i^t | m_i^{hard} = 1\}$ and the hard mask set $\mathcal{M}^{hard} = \{m_i^{hard} | i = 1, 2, \dots, N\}$ need to be stored and transmitted, minimizing the required data redundancy and transmission overhead.

3.5 Optimization and Compression

For the first frame optimization, we employ COLMAP [50] to generate the initial point cloud and follow the pipeline of 3DGStream [2]. The optimization for both the first frame and non-key frames is supervised by the reconstruction loss $\mathcal{L}_{recon}$, which is composed by an L_1 -norm loss $\mathcal{L}_1$ and a D-SSIM loss $\mathcal{L}_{D-SSIM}$ [51]:

$$\mathcal{L}_{recon} = (1 - \lambda_{D-SSIM})\mathcal{L}_1 + \lambda_{D-SSIM}\mathcal{L}_{D-SSIM} \quad (14)$$

For key frame optimization, we minimize a combined loss consisting of $\mathcal{L}_{recon}$ and $\mathcal{L}_{error}$:

$$\mathcal{L}_{total} = \mathcal{L}_{recon} + \lambda_{error}\mathcal{L}_{error} \quad (15)$$

where λ_{error} controls the degree of error awareness, thereby balancing reconstruction quality and memory efficiency. We set $\lambda_{D-SSIM} = 0.2$ and $\lambda_{error} = 0.001$ in this paper.

After optimization, the initialized Gaussians θ^0 and the residuals $\Delta \hat{\theta}^t$ for key frame error correction are further compressed through quantization and entropy coding, enabling compact storage without performance degradation. More details are provided in the **Appendix**.

4 Experiments

4.1 Experimental Setup

We evaluate our method on two widely-used public benchmark datasets. (1) **Neural 3D Video (N3DV)** [1] consists of six indoor video sequences captured by 18 to 21 viewpoints. (2) **Meet**

Table 2: Quantitative comparisons on the long video sequence *flame salmon* from the N3DV dataset [1].

Method	PSNR (dB) $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	Storage (MB) $\downarrow$	Training (sec) $\downarrow$	Rendering (FPS) $\uparrow$
E-NeRF [54]	23.48	0.89	0.260	0.692	13.8	5
4DGS [4]	28.89	0.952	0.197	2.23	31.2	90
TGH [23]	29.44	0.945	0.214	0.075	6.3	550
ComGS-s (ours)	29.56	0.920	0.140	0.053	45.4	91



Figure 3: Quantitative comparison. We visualize our method and other online FVV methods on N3DV [1] and MeetRoom [37] dataset.

Room [37] comprises four indoor scenes recorded with a 13 cameras multi-view system. In both of two datasets, we employ the first view for testing. Our method is implemented on an NVIDIA A100 GPU. We train 150 epochs for non-key frames reconstruction and 1000 epochs for key frames fine-tuning. We measure the visual quality of rendered images by average PSNR, required storage, rendering FPS and training time. More implement details are provided in the **Appendix**.

4.2 Quantitative Comparisons

We conduct quantitative comparisons on existing online methods including StreamRF [37], TeTriRF [40], 3DGStream [2] and QUEEN [13], as well as the SOTA offline FVV approaches [6, 8, 41, 52] on N3DV and Meetroom (Tab. 1). Our method is evaluated in two variants: ComGS-s (small) and ComGS-l (large), using key frame intervals of $s=10$ and $s=2$, respectively.

Tab. 1 shows that our ComGS achieves competitive results among existing online FVV methods on N3DV dataset. Notably, ComGS-s achieves a substantial reduction in storage by $159\times$ compared to 3DGStream and $14\times$ compared to QUEEN. This advantage enables real-time transmission in limited bandwidth and enhances the overall user viewing experience. On MeetRoom dataset, our method outperforms 3DGStream, obtaining $+0.7\text{dB}$ PSNR and $146\times$ smaller size. Our advantages are mainly due to two factor: 1) using keypoint as a shared representation requires transmitting only a small number of keypoint attributes; and 2) the error-aware correction module effectively rectifies regions with scene inaccuracies using minimal additional parameters. In the **Appendix**, the quantitative results are provided for each scene to offer a more detailed comparison.

We further evaluate the effectiveness of ComGS on handling long videos. We compare our method with the TGH [23] (which is the most recently proposed for handling long video sequences) on the *Flame Salmon* sequence (1200 frames) from the N3DV dataset [1]. Tab. 2 shows that our method achieves competitive results on rendering quality and required storage.

4.3 Qualitative Comparisons

As shown in Fig. 3, we compare our reconstructed results to other online FVV methods on N3DV and MeetRoom. ComGS effectively reconstructs both motion and static regions and provides more closer results to the ground truth. Fig. 3 shows that 3DGStream introduces noticeable artifacts due to its global update of Gaussian points across the entire scene, which often leads to incorrect updates

Table 3: Ablation study on proposed components. *Flame Steak* and *Flame Salmon* scenes are from the N3DV dataset.

Experiments	Selection	Adaptive	Correction	Flame Steak		Flame Salmon	
				PSNR (dB)↑	Storage (KB)↓	PSNR (dB)↑	Storage (KB)↓
1	×	✓	✓	33.27	46.7	29.22	56.7
2	✓	×	✓	32.82	36.4	28.96	45.7
3	×	×	✓	31.26	37.9	27.75	46.4
4	✓	✓	×	31.67	26.9	28.74	26.9
5	✓	✓	✓	33.49	46.5	29.32	53.4



Figure 4: Visualization of our keypoint-driven motion representation. **Top**: selected keypoints are concentrated in motion regions. **Bottom**: adaptive control of neighboring points also focuses on motion-intensive areas, enabling accurate and efficient motion modeling.

in static regions. In contrast, our method restricts modeling to motion regions and applies targeted corrections in error-prone areas, resulting in more accurate and robust scene reconstruction. More qualitative results are offered in **Appendix**.

4.4 Ablation Study

To validate the effectiveness of our proposed methods, we ablate three components of ComGS framework in Tab. 3.

In the **Experiment 1**, we ablate the motion-sensitive keypoint selection and instead select keypoints randomly. Since the random selection is not guided by motion regions, it may result in ineffective modeling of static areas and inadequate representation on motion regions (Fig. 5 (b)), which leads to a slight degradation in PSNR. **Experiment 2** removes the adaptive motion-driven mechanism and models scene motion only using the selected keypoints, without incorporating any neighboring points. The resulting drop in reconstruction quality demonstrates that effective motion modeling relies not only on accurately keypoint selection, but also on the selection of their neighboring points. In the **Experiment 3**, we reconstruct FVV only relying on the first frame reconstruction and key frame correction, without modeling non-key frames by keypoint reconstruction, which results in a significant performance drop. We emphasize that although the parameters of keypoints are few, the keypoint-based modeling plays a crucial role in FVV reconstruction. **Experiment 4** ablates the error-aware correction in key frame reconstruction. The performance degradation demonstrates that the error-aware correction in key frames would solve the error accumulation across frames.

Table 4: Ablation study on comparing control strategies for neighboring points.

Control tech	PSNR (dB)	Storage (KB)
KNN	31.39	44.1
Adaptive	31.87	49.0

Table 5: Ablation study of the error-aware correction strategy.

Configuration	PSNR (dB)	Storage (KB)
w/o error-aware	31.65	373
with error-aware	31.87	49.0

To further investigate the role of keypoint-driven motion representation, we visualize the selection and driven process in Fig. 4. The top row shows that keypoints are predominantly selected in motion



Figure 5: Visualization of different selection methods and corresponding updated regions.



Figure 6: (a) PSNR comparison over time. Visualizations of (b) w/o key frames correction. (c) ComGS-s. (d) ComGS-l.

regions, such as the human body and moving objects. The bottom row highlights the adaptively controlled areas for neighboring points, which similarly focus on regions with significant motion (e.g., the person and the dog). Fig. 5 visualizes Gaussians updated region using farthest keypoint selection [16], random keypoint selection and our method, respectively, which demonstrates that our method accurately captures motion-intensive areas. These results indicate that ComGS can effectively leverage the locality and consistency of scene motion.

We also evaluate a KNN-based method [29] for selecting neighboring points around keypoints (Tab. 4). This approach shows inferior performance, as it does not distinguish between static and motion regions, leading to redundant modeling and poor adaptation to varying motion scales.

Fig. 6 evaluates the effect of key frame correction. The visual results in Fig. 6 (b–d) further highlight that key frame correction significantly reduces artifacts in motion regions such as flames, helping to maintain finer temporal consistency throughout the sequence. Tab. 5 shows that correction without error-aware leads to significantly higher storage due to redundant Gaussians updating. Moreover, without focusing on high-error regions, updates may affect error-free areas and result in suboptimal performance. Therefore, enabling error-awareness improves both accuracy and efficiency.

5 Conclusion

In this paper, we proposed ComGS, a storage-efficient framework for online FVV real-time transmission. We utilized a keypoint-driven motion representation to models scene motion by leveraging the locality and consistency of motion. This approach significantly reduces storage requirements through motion-sensitive keypoint selection and an adaptive motion driven mechanism. To address error accumulation over time, we further introduce an error-aware correction strategy that mitigates these error in an efficient manner. Experiments demonstrate the surpassing storage efficiency, competitive visual fidelity and rendering speed of our method.

Limitations: Notably, our method still remains a few limitations. First, as the first frame serves as the foundation for subsequent frame updates, poor initialization would lead to error propagation and degraded performance. Developing a robust and efficient initialized strategy for first frame could further improve the visual quality and storage efficiency of online FVV. Second, our method relies on the dense view videos as inputs, which is expensive for practical applications. Future work will explore extending the framework to sparse-view or monocular inputs for real-world scenarios. Additionally, this method does not fully consider the training time in the encoding stage, leaving room for further improvements in training efficiency. In future work, we aim to design a practical solution on novel applications, such as 3D video conference and volumetric live streaming, providing viewers with immersive and interactive experiences.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No.62031013), the Guangdong Province Key Construction Discipline Scientific Research Capacity Improvement Project (Grant No.2022ZDJS117), Engineering Technology R&D Center of Guangdong Provincial Universities (Grant No.2024GCZX004) and the Pengcheng Laboratory.

References

- [1] T. Li, M. Slavcheva, M. Zollhoefer, S. Green, C. Lassner, C. Kim, T. Schmidt, S. Lovegrove, M. Goesele, R. Newcombe *et al.*, “Neural 3d video synthesis from multi-view video,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5521–5531.
- [2] J. Sun, H. Jiao, G. Li, Z. Zhang, L. Zhao, and W. Xing, “3dstream: 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*, 2024, pp. 20 675–20 685.
- [3] Y. Chen, Q. Wang, H. Chen, X. Song, H. Tang, and M. Tian, “An overview of augmented reality technology,” in *Journal of Physics: Conference Series*, vol. 1237, no. 2. IOP Publishing, 2019, p. 022082.
- [4] Z. Yang, H. Yang, Z. Pan, and L. Zhang, “Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting,” *arXiv preprint arXiv:2310.10642*, 2023.
- [5] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan, “Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis,” in *2024 International Conference on 3D Vision (3DV)*. IEEE, 2024, pp. 800–809.
- [6] Z. Li, Z. Chen, Z. Li, and Y. Xu, “Spacetime gaussian feature splatting for real-time dynamic view synthesis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 8508–8520.
- [7] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin, “Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 331–20 341.
- [8] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, “4d gaussian splatting for real-time dynamic scene rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 310–20 320.
- [9] Z. Lu, X. Guo, L. Hui, T. Chen, M. Yang, X. Tang, F. Zhu, and Y. Dai, “3d geometry-aware deformable gaussian splatting for dynamic view synthesis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 8900–8910.
- [10] M. Liu, Q. Yang, H. Huang, W. Huang, Z. Yuan, Z. Li, and Y. Xu, “Light4gs: Lightweight compact 4d gaussian splatting generation via context model,” *arXiv preprint arXiv:2503.13948*, 2025.
- [11] J. Bae, S. Kim, Y. Yun, H. Lee, G. Bang, and Y. Uh, “Per-gaussian embedding-based deformation for deformable 3d gaussian splatting,” in *European Conference on Computer Vision*. Springer, 2024, pp. 321–335.
- [12] T. Müller, A. Evans, C. Schied, and A. Keller, “Instant neural graphics primitives with a multiresolution hash encoding,” *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.
- [13] S. Girish, T. Li, A. Mazumdar, A. Shrivastava, S. De Mello *et al.*, “Queen: Quantized efficient encoding of dynamic gaussians for streaming free-viewpoint videos,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 43 435–43 467, 2025.
- [14] Q. Gao, J. Meng, C. Wen, J. Chen, and J. Zhang, “Hicom: Hierarchical coherent motion for dynamic streamable scenes with 3d gaussian splatting,” in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- [15] Y.-H. Huang, Y.-T. Sun, Z. Yang, X. Lyu, Y.-P. Cao, and X. Qi, “Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 4220–4230.
- [16] D. Wan, R. Lu, and G. Zeng, “Superpoint gaussian splatting for real-time high-fidelity dynamic scene reconstruction,” in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 49 957–49 972.

- [17] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering,” *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.
- [18] K. Navaneet, K. P. Meibodi, S. A. Koohpayegani, and H. Pirsivash, “Compact3d: Compressing gaussian splat radiance field models with vector quantization,” *arXiv preprint arXiv:2311.18159*, 2023.
- [19] Z. Fan, K. Wang, K. Wen, Z. Zhu, D. Xu, and Z. Wang, “Lightgaussian: Unbounded 3d gaussian compression with 15x reduction and 200+ fps,” *arXiv preprint arXiv:2311.17245*, 2023.
- [20] J. C. Lee, D. Rho, X. Sun, J. H. Ko, and E. Park, “Compact 3d gaussian representation for radiance field,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21 719–21 728.
- [21] H. Wang, H. Zhu, T. He, R. Feng, J. Deng, J. Bian, and Z. Chen, “End-to-end rate-distortion optimized 3d gaussian representation,” in *European Conference on Computer Vision*. Springer, 2025, pp. 76–92.
- [22] P. Papantonakis, G. Kopanas, B. Kerbl, A. Lanvin, and G. Drettakis, “Reducing the memory footprint of 3d gaussian splatting,” *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 7, no. 1, pp. 1–17, 2024.
- [23] Z. Xu, Y. Xu, Z. Yu, S. Peng, J. Sun, H. Bao, and X. Zhou, “Representing long volumetric video with temporal gaussian hierarchy,” *ACM Transactions on Graphics (TOG)*, vol. 43, no. 6, pp. 1–18, 2024.
- [24] Y. Wang, P. Yang, Z. Xu, J. Sun, Z. Zhang, Y. Chen, H. Bao, S. Peng, and X. Zhou, “Freetimegs: Free gaussian primitives at anytime anywhere for dynamic scene reconstruction,” in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 21 750–21 760.
- [25] W. O. Cho, I. Cho, S. Kim, J. Bae, Y. Uh, and S. J. Kim, “4d scaffold gaussian splatting for memory efficient dynamic scene reconstruction,” *arXiv preprint arXiv:2411.17044*, 2024.
- [26] D. Sun, H. Guan, K. Zhang, X. Xie, and S. K. Zhou, “Sdd-4dgs: Static-dynamic aware decoupling in gaussian splatting for 4d scene reconstruction,” *arXiv preprint arXiv:2503.09332*, 2025.
- [27] J. Yan, R. Peng, L. Tang, and R. Wang, “4d gaussian splatting with scale-aware residual field and adaptive optimization for real-time rendering of temporally complex dynamic scenes,” in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 7871–7880.
- [28] A. Cao and J. Johnson, “Hexplane: A fast representation for dynamic scenes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 130–141.
- [29] L. E. Peterson, “K-nearest neighbor,” *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.
- [30] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [31] C. Sun, M. Sun, and H.-T. Chen, “Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5459–5469.
- [32] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, “Tensorf: Tensorial radiance fields,” in *European conference on computer vision*. Springer, 2022, pp. 333–350.
- [33] A. Chen, Z. Xu, X. Wei, S. Tang, H. Su, and A. Geiger, “Dictionary fields: Learning a neural basis decomposition,” *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–12, 2023.
- [34] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, “Plenoxels: Radiance fields without neural networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 5501–5510.
- [35] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, “Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 5855–5864.
- [36] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, “Ref-nerf: Structured view-dependent appearance for neural radiance fields,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 5481–5490.
- [37] L. Li, Z. Shen, Z. Wang, L. Shen, and P. Tan, “Streaming radiance fields for 3d video synthesis,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 13 485–13 498, 2022.

- [38] L. Wang, K. Yao, C. Guo, Z. Zhang, Q. Hu, J. Yu, L. Xu, and M. Wu, "Videorf: Rendering dynamic radiance fields as 2d feature video streams," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 470–481.
- [39] L. Wang, Q. Hu, Q. He, Z. Wang, J. Yu, T. Tuytelaars, L. Xu, and M. Wu, "Neural residual radiance fields for streamably free-viewpoint videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 76–87.
- [40] M. Wu, Z. Wang, G. Kouros, and T. Tuytelaars, "Tetrirf: Temporal tri-plane radiance fields for efficient free-viewpoint video," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 6487–6496.
- [41] L. Song, A. Chen, Z. Li, Z. Chen, L. Chen, J. Yuan, Y. Xu, and A. Geiger, "Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 5, pp. 2732–2742, 2023.
- [42] P. Wang, Z. Zhang, L. Wang, K. Yao, S. Xie, J. Yu, M. Wu, and L. Xu, "V³: Viewing volumetric videos on mobiles via streamable 2d dynamic gaussians," *ACM Transactions on Graphics (TOG)*, vol. 43, no. 6, pp. 1–13, 2024.
- [43] J. Yan, R. Peng, Z. Wang, L. Tang, J. Yang, J. Liang, J. Wu, and R. Wang, "Instant gaussian stream: Fast and generalizable streaming of dynamic scene reconstruction via gaussian splatting," *arXiv preprint arXiv:2503.16979*, 2025.
- [44] T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, and B. Dai, "Scaffold-gs: Structured 3d gaussians for view-adaptive rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 654–20 664.
- [45] S. Niedermayr, J. Stumpfegger, and R. Westermann, "Compressed 3d gaussian splatting for accelerated novel view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 349–10 358.
- [46] Y. Chen, Q. Wu, W. Lin, M. Harandi, and J. Cai, "Hac: Hash-grid assisted context for 3d gaussian splatting compression," in *European Conference on Computer Vision*. Springer, 2025, pp. 422–438.
- [47] Y.-T. Zhan, C.-Y. Ho, H. Yang, Y.-H. Chen, J. C. Chiang, Y.-L. Liu, and W.-H. Peng, "Cat-3dgs: A context-adaptive triplane approach to rate-distortion-optimized 3dgs compression," *arXiv preprint arXiv:2503.00357*, 2025.
- [48] Z. Chen, Z. Chen, W. Jiang, W. Wang, L. Liu, and D. Xu, "4dgs-cc: A contextual coding framework for 4d gaussian splatting data compression," *arXiv preprint arXiv:2504.18925*, 2025.
- [49] L. Tang, J. Yang, R. Peng, Y. Zhai, S. Shen, and R. Wang, "Compressing streamable free-viewpoint videos to 0.1 mb per frame," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 7, 2025, pp. 7257–7265.
- [50] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.
- [51] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [52] B. Attal, J.-B. Huang, C. Richardt, M. Zollhoefer, J. Kopf, M. O'Toole, and C. Kim, "Hyperreel: High-fidelity 6-dof video with ray-conditioned sampling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 610–16 620.
- [53] Y. Jiang, K. Yao, Z. Su, Z. Shen, H. Luo, and L. Xu, "Instant-nvr: Instant neural volumetric rendering for human-object interactions from monocular rgb-d stream," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 595–605.
- [54] H. Lin, S. Peng, Z. Xu, Y. Yan, Q. Shuai, H. Bao, and X. Zhou, "Efficient neural radiance fields for interactive free-viewpoint video," in *SIGGRAPH Asia 2022 Conference Papers*, 2022, pp. 1–9.
- [55] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.
- [56] S. Fridovich-Keil, G. Meanti, F. R. Warburg, B. Recht, and A. Kanazawa, "K-planes: Explicit radiance fields in space, time, and appearance," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 479–12 488.

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: We show comprehensive experiments and ablation studies on the datasets that are broadly used in this area. Our claims accurately reflect the contribution and scope of our work.

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: We discuss the limitations of our work in the conclusions section.

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: [NA]

Justification: Our study is not a theory work. Sec. 3 explains all components and equations in detail, and we cite relevant theoretical works in the related work and method sections.

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: We provide detailed explanations of each component of our method in Sec. 3, and additional implementation details are presented in Appendix A.

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: [No]

Justification: We would release the code after acceptance.

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: We follow benchmark and evaluation metric which are widely used by existing works in this area. More experimental details and hyperparameters are provided in 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: We follow benchmark and evaluation metrics that are widely used by existing work in the area. To our knowledge, most of the existing work in this area do not provide statistical significance.

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: We provide experimental details comprise of computational hardware, training times, storage requirements and dataset specification in the implementation details in the main paper and Appendix.

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: Yes, our 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: [Yes]

Justification: We discuss the potential positive societal impacts in Appendix.

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: This 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: Our method is evaluated on several public datasets and we followed their license, as well as credited and cited their work and dataset.

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 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: N/A.

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: N/A.

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: N/A.

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: N/A.

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

We provide more material to supplement our main paper. This appendix first introduces more implementation details in Sec. A. Then, we provide additional experimental results in Sec. B, and broader impact in Sec. C.

A More Implementation details

Training: Our code is based on the open-source code of 3DGStream [2]. On both N3DV and MeetRoom dataset, we utilize COLMAP [50] to generate the initial point cloud and vanilla 3DGS [17] to initialize the Gaussians for 3000 epochs at first frame. Subsequently, our ComGS reconstructs the non key frames for 150 epochs and key frames for 1000 epochs. For the balance of visual quality and storage requirements, we set spherical harmonics (SH) degree to 1. During training, the learning rate for Gaussian attributes is set to 0.002, for the attributes of the adaptive influence region to 0.02, and for the learnable mask m_i to 0.01. Other learning rates follow the setting of 3DGStream [2].

Compression: For the reconstruction process, the uncompressed Gaussian attributes and their residuals have substantial memory requirements. We employ quantization and entropy coding to further compress them. Specifically, for the first frame reconstruction, we apply 16-bit quantization to the position attributes due to their higher sensitivity, while the other attributes are quantized to 8 bits. For the correction in key frame reconstruction, we quantize all attribute residuals using 8 bits. Notably, the attributes of a keypoint play a crucial role in guiding the motion of nearby non-keypoints. As a result, even minor quantization errors in keypoints may be amplified throughout the scene. To preserve modeling accuracy, we thus refrain from quantizing keypoint attributes. Finally, we deliver these quantized values to entropy coding [55].

Datasets: (1) **Neural 3D Video (N3DV) dataset** [1] comprises of six indoor scenes captured by a multi-view system of 18 to 21 cameras at a resolution of 2704×2028 and 30 FPS. Following the previous works [2, 1, 8], we downsample the videos by a factor of 2 for training and testing and employ the central view for testing view. (2) **MeetRoom dataset** [37] is captured by a 13-camera multi-view system, including four dynamic scenes at 1280×720 resolution and 30 FPS. The center reference camera is also used for testing. As the aforementioned two datasets contain 300 frames, we also conduct long video reconstruction evaluation on the *Flame Salmon* scene with 1200 frames from the N3DV dataset. We perform distortion for this dataset following the settings of the 3DGS [17] to improve the reconstruction quality.

Table 6: Quantitative results of the random access version on N3DV dataset [1].

Metric	Coffee Martini	Cook Spinach	Cut Beef	Flame Salmon	Flame Steak	Sear Steak
PSNR (dB $\uparrow$)	28.52	32.31	32.97	29.19	33.01	33.51
Storage (KB $\downarrow$)	177.4	115.3	119.3	168.6	114.7	105.3

Table 7: Ablation study on Number of keypoints.

#Keypoints	50	100	200	300	400	500
PSNR (dB $\uparrow$)	31.77	31.85	31.87	31.84	31.86	31.80
Storage (KB $\downarrow$)	44.4	46.2	50.1	50.2	54.4	57.3

B Additional Experimental Results

B.1 Random Access

Random access is crucial for video streaming and interactive user experiments. However, existing online FVV reconstruction methods [2, 13, 14] rely on the Gaussian points of the previous frame during each current frame reconstruction, thus only supporting forward playback from the first frame.

In contrast, our method enables random access by simply modifying a small part of the system configuration. Specifically, compared to the original setting, we instead reconstruct non-key frames

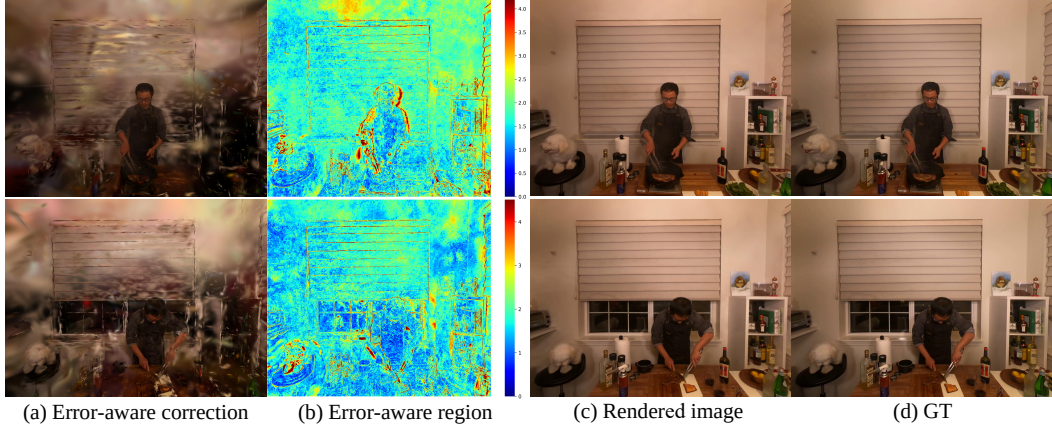


Figure 7: (a) Visualization of error-aware Gaussians. (b) Visualization of error regions between key frame and previous frame. (c)(d) Comparison on rendered images and original images.

using the keypoints from their nearest preceding key frame. Key frames are reconstructed based on the previous key frame using error-aware correction. Additionally, to further decouple key frames from earlier ones, we introduce periodic I-frames (e.g., every 60 frames), in which all Gaussian primitives are either saved or re-optimized independently. With this adaptation, accessing a specific frame only requires access to its nearest preceding key frame and the associated keypoints, making random access feasible. Tab. 6 presents the quantitative results of the random access version.

Table 8: Ablation study on Group of Frames.

#Frames	2	5	10	15	20
PSNR (dB $\uparrow$)	32.12	32.01	31.87	31.78	31.66
Storage (KB $\downarrow$)	108.3	66.6	50.1	43.2	40.0

B.2 More Ablation Study

In this section, we further investigate the hyperparameters and analyze the impact of the proposed components on N3DV [1] dataset, to achieve a balance between performance and efficiency.

Effect of the keypoint numbers: To investigate the impact of the number of keypoints on reconstruction quality and compression efficiency, we conduct an ablation study by varying the number of keypoints from 50 to 500. As shown in Tab. 7, the reconstruction performance peaks when using 200 keypoints. This observation aligns with the nature of dynamic scenes, where motion typically occurs in a limited spatial region. Using 200 keypoints is sufficient to capture these areas for effective reconstruction. Increasing the number of keypoints beyond this leads to redundant or incorrect representation in static regions. Therefore, using 200 keypoints strikes a good balance between performance and storage, and is adopted as the default configuration in our method.

Effect of group of frames: We evaluate how the size of the Group of Frames (GoF) affects reconstruction quality and storage, as shown in Tab. 8. These results indicate that shorter GoFs can better handle non-rigid motions and novel objects, which are difficult to be reconstructed by keypoint-driven motion. Larger GoFs exploit temporal redundancy for better compression, but may accumulate errors in the presence of motion and scene changes. In our setting, we use $\text{GoF} = 2$ as our *large* model for high-fidelity reconstruction, and $\text{GoF} = 10$ as our *small* model for compact representation.

Effect of error-aware correction: We explore the effect of the parameter λ_{error} on reconstruction quality and storage, as shown in Tab. 9. While a larger λ_{error} improves compression by focusing only on perceptually salient errors, it may overlook subtle regions, which leads to degraded reconstruction. In contrast, smaller values retain more points, which helps suppress error accumulation across frames, albeit with higher storage costs.

Table 9: Effect of λ_{error} on reconstructed quality and storage.

λ_{error}	0	0.0001	0.001	0.01
PSNR (dB $\uparrow$)	31.91	31.91	31.87	31.79
Storage (KB $\downarrow$)	183.0	96.3	50.1	29.2

Table 10: **Per-scene quantitative results on the N3DV dataset.** Offline and online methods are separated for clarity.

Method	Coffee Martini		Cook Spinach		Cut Beef	
	PSNR (dB $\uparrow$)	Storage (MB $\downarrow$)	PSNR (dB $\uparrow$)	Storage (MB $\downarrow$)	PSNR (dB $\uparrow$)	Storage (MB $\downarrow$)
KPlanes [56]	29.99	1.0	32.60	1.0	31.82	1.0
NeRFPlayer [41]	31.53	18.4	30.56	18.4	29.35	18.4
HyperReel [52]	28.37	1.2	32.30	1.2	32.92	1.2
4DGS [4]	28.33	29.0	32.93	29.0	33.85	29.0
4D-GS [8]	27.34	0.3	32.46	0.3	32.49	0.3
Spacetime-GS [6]	28.61	0.7	33.18	0.7	33.52	0.7
E-D3DGS [11]	29.33	0.5	33.19	0.5	33.25	0.5
StreamRF [37]	27.84	31.84	31.59	31.84	31.81	31.84
3DGStream [2]	27.75	7.80	33.31	7.80	33.21	7.80
QUEEN-I [13]	28.38	1.17	33.40	0.59	34.01	0.57
ComGS-s (ours)	28.63	0.058	32.94	0.047	33.30	0.051
ComGS-l (ours)	28.76	0.154	33.26	0.094	33.53	0.104
Method	Flame Salmon		Flame Steak		Sear Steak	
	PSNR (dB $\uparrow$)	Storage (MB $\downarrow$)	PSNR (dB $\uparrow$)	Storage (MB $\downarrow$)	PSNR (dB $\uparrow$)	Storage (MB $\downarrow$)
KPlanes [56]	30.44	1.0	32.38	1.0	32.52	1.0
NeRFPlayer [41]	31.65	18.4	31.93	18.4	29.12	18.4
HyperReel [52]	28.26	1.2	32.20	1.2	32.57	1.2
4DGS [4]	29.38	29.0	34.03	29.0	33.51	29.0
4D-GS [8]	29.20	0.3	32.51	0.3	32.49	0.3
Spacetime-GS [6]	29.48	0.7	33.40	0.7	33.46	0.7
E-D3DGS [11]	29.72	0.5	33.55	0.5	33.55	0.5
StreamRF [37]	28.26	31.84	32.24	31.84	32.36	31.84
3DGStream [2]	28.42	7.80	34.30	7.80	33.01	7.80
QUEEN-I [13]	29.25	1.00	34.17	0.59	33.93	0.56
ComGS-s (ours)	29.31	0.052	33.42	0.045	33.59	0.040
ComGS-l (ours)	29.58	0.129	33.84	0.083	33.74	0.0704

Fig. 7 (a) visualizes the error-aware Gaussian points identified by error-aware correction, while (b) shows a heatmap of differences between the key frame and the previous frame, which highlights the error regions. We observe that the error-aware points in (a) align well with the high-error regions in (b), which indicates that our method effectively captures areas likely to suffer from error accumulation. Fig. 7 (c) and (d) compare our rendered images with the ground truth. The results show that our method significantly reduces artifacts in dynamic regions, confirming the effectiveness of our error-aware correction.

B.3 More Results

To offer a more comprehensive comparison, the per-scene quantitative results are presented on N3DV [1] and MeetRoom [37] in Tab. 10 and Tab. 11, respectively. Moreover, we also provide the experimental results of existing offline and online methods in Tab. 10 as a reference. Further qualitative results with StreamRF [37] and 3DGStream [2] are indicated in Fig. 8 and Fig. 9.

C Broader Impact

Our work is a positive technology. This method reconstructs free-viewpoint videos from multi-view 2D videos in a streaming manner, which can improve the immersive and interactive experience of viewers. As discussed in the introduction, this technology has potential to benefit various aspects of daily life, including applications in remote diagnosis and 3D video conferencing.

Table 11: **Per-scene quantitative results on the MeetRoom dataset.**

Metrics	Discussion	Stepin	Trimming	VrHeadset
PSNR (dB $\uparrow$)	31.72	30.17	32.12	31.95
Storage (KB $\downarrow$)	37.5	24.2	27.0	24.5

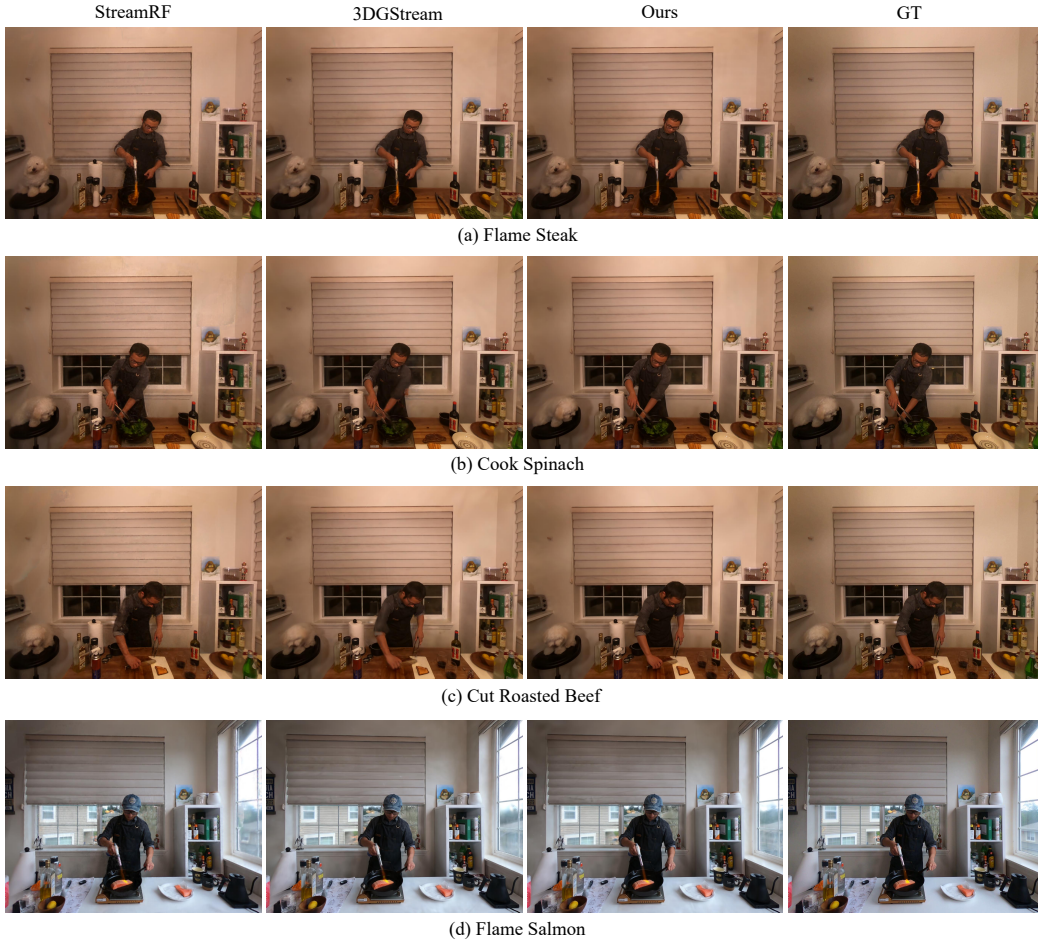


Figure 8: Comparison on N3DV [1] dataset.



Figure 9: Comparison on MeetRoom [37] dataset.