MoDec-GS: Global-to-Local Motion Decomposition and Temporal Interval Adjustment for Compact Dynamic 3D Gaussian Splatting

Sangwoon Kwak[1,2], Joonsoo Kim[1], Jun Young Jeong[1], Won-Sik Cheong[1],
Jihyong Oh[3†], Munchurl Kim[2†]
[1]Electronics and Telecommunications Research Institute,
[2]Korea Advanced Institute of Science and Technology, [3]Chung-Ang University
{s.kwak, joonsookim, jyj0120, wscheong}@etri.re.kr
{sw.kwak, mkimee}@kaist.ac.kr jihyongoh@cau.ac.kr
https://kaist-viclab.github.io/MoDecGS-site/

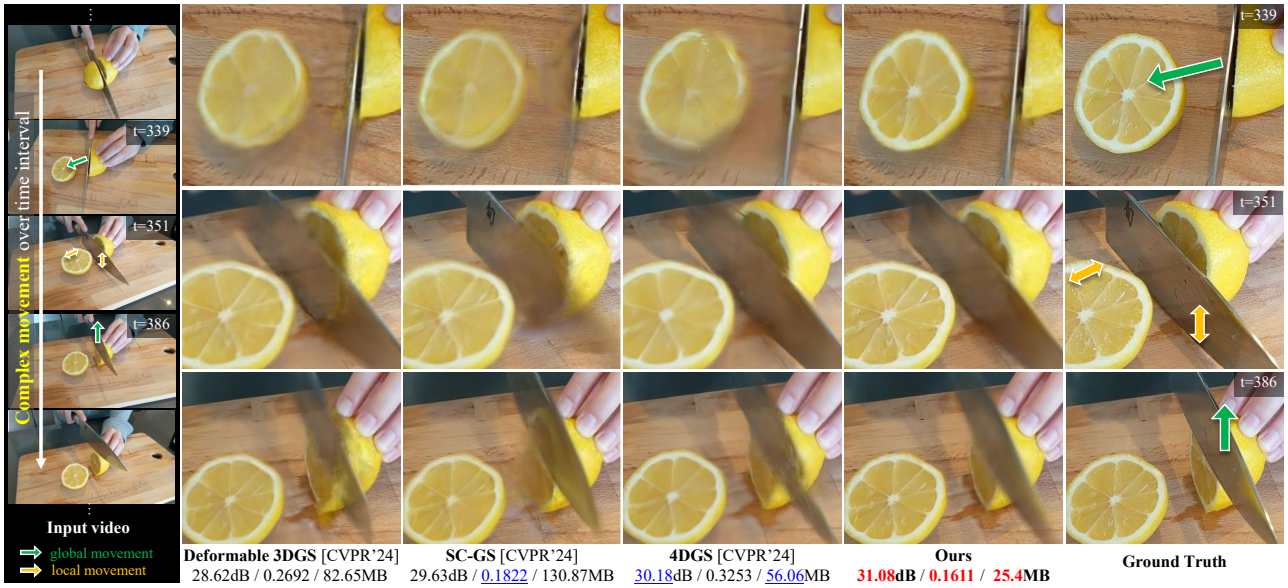

Figure 1. **Novel view synthesis results on** [49]. We introduce **MoDec-GS**, a novel framework for learning compact dynamic 3D Gaussians from real-world videos with complex motion. While existing SOTA methods [21, 60, 63] have difficulty modeling such complex combination of global and local motions, our approach effectively handles them thanks to GLMD (Sec. 4.1), and outperforms the prior methods in rendering quality even with a *compact model size*. The metrics under each framework are, PSNR (dB)↑ / LPIPS [65] ↓ / Storage (MB)↓.

Abstract

3D Gaussian Splatting (3DGS) has made significant strides in scene representation and neural rendering, with intense efforts focused on adapting it for dynamic scenes. Despite delivering remarkable rendering quality and speed, existing methods struggle with storage demands and the representation of complex real-world motions. To address these challenges, we propose MoDec-GS, a memory-efficient Gaussian splatting framework designed to reconstruct novel views in challenging scenarios with complex motions. We introduce Global-to-Local Motion Decomposition (GLMD) to effectively capture dynamic motions in a coarse-to-fine manner. This approach leverages Global Canonical Scaffolds (Global CS) and Local Canonical Scaffolds (Local CS), which extend static Scaffold representation to dynamic video reconstruction. For Global CS, we propose Global Anchor Deformation (GAD) to efficiently represent global dynamics along complex motions, by directly deforming the implicit Scaffold attributes which are anchor position, offset, and local context features. Next, we finely adjust local motions via the Local Gaussian Deformation (LGD) of Local CS explicitly. Additionally, we introduce Temporal Interval Adjustment (TIA) to automatically control the tem-

†Co-corresponding authors.

poral coverage of each Local CS during training, enabling MoDec-GS to find optimal interval assignments based on the specified number of temporal segments. Extensive evaluations demonstrate that MoDec-GS achieves an average 70% reduction in model size over state-of-the-art methods for dynamic 3D Gaussians from real-world dynamic videos while maintaining or even improving rendering quality.

1. Introduction

Novel view synthesis (NVS) generates new perspectives of a scene from a limited set of images, closely approximating real footage. NVS has long been a key research area for many years, with advancements in techniques such as depth-image-based rendering [12, 44, 67]. This ongoing interest is largely driven by the broad applicability of NVS in areas such as virtual reality, augmented reality, and immersive media, where natural viewpoint transitions that mimic real-life experiences are essential for enhancing user realism [1, 2].

Approaching NVS as the task of modeling the radiance field has taken the computer vision community by storm. This paradigm shift, led by Neural Radiance Field (NeRF) [42], has set a new photorealism standard that surpasses conventional methods. The original NeRF represents the radiance field as an implicit function linked to volume rendering, achieving remarkable visual fidelity. However, it faces challenges with its slow training and, more critically, rendering speed, which is far from real-time. Despite various optimization efforts [14, 36, 45], achieving real-time rendering on consumer-level devices remains difficult, largely due to NeRF's reliance on pixel-wise volumetric rendering.

Recently, 3D Gaussian Splatting (3DGS) [26] has emerged as a compelling alternative, offering exceptional rendering speed without compromising visual quality. By representing the radiance field as a collection of 3D Gaussian ellipsoids, 3DGS enables efficient patch-wise rasterization through Gaussian projection and alpha compositing. This patch-level rasterization pipeline, fully leveraging GPU parallel computation, allows 3DGS to achieve unrivaled rendering speed. Subsequently, 3DGS has inspired diverse research trajectories, with extension to video inputs and compression emerging as key areas of focus [28, 32, 40, 43, 60, 62].

Current approaches to dynamic adaptation often pair a static canonical 3DGS with implicit [9, 21, 27, 33, 54, 60, 63] or explicit [24, 34] components to manage the temporal deformation of the attributes within the canonical 3DGS. Another approach extends 3D Gaussian primitives into 4D by incorporating a temporal dimension [8, 62]. While both approaches preserve solid rendering quality for dynamic scenes, they face storage issues due to the multi-dimensional attributes assigned to numerous 3D Gaussians in canonical 3DGS or 4D Gaussians [29]. Handling long-duration content with complex motion also poses difficulties, as representing all frames with a unified model causes blurring due to its limited capacity, as noted by Shaw et al. [54]. To address this, they segment sequences based on scene motion and train a separate model for each segment. However, this approach requires an extra step to compute motion vectors, which diminishes the usability of 3DGS.

Early methods to address the storage demands of 3DGS had focused on compressing the original 3DGS representation, employing techniques like vector quantization [10, 28, 46, 47], Gaussian pruning [10, 28], and implicit encoding of Gaussians' attributes [18, 28, 61]. Some recent studies have successfully introduced more memory-efficient representations based on the 3DGS framework, with a notable example being the anchor-based representation [4, 38, 40, 58], which assigns implicit features at sparse anchor points to predict attributes for a broader set of neighboring 3D Gaussians. However, extending these methods, originally designed for static scenes, to dynamic videos may be challenging, as dynamic scene modeling typically requires substantial additional components. A few recent methods have aimed to unify dynamic extension and memory efficiency, but one [56] is limited to multi-view sequences, while another [29] still struggles with long-duration content.

To address the limitations of existing methods, we propose a novel dynamic 3D Gaussian splatting framework, enhancing model compactness and rendering quality while preserving real-time capability. Our framework employs a deformation-based approach with an anchor-based representation [40] for the canonical 3DGS due to its compactness. Building on this, we introduce a two-stage deformation process inspired by the observation that natural motion involves both global and local components. In the first stage, called Global Anchor Deformation (GAD), the canonical representation is deformed to a specific time interval using an anchor deformation encoder that captures global motion across the entire sequence. In the second stage, called Local Gaussian Deformation (LGD), the deformed representation is refined via local deformation encoder capturing finer motion within the specific time interval around the chosen time point. A final feature of our framework is Temporal Interval Adjustment (TIA), which assigns the temporal interval to each deformation encoder and is automatically determined during training. This does not require any precomputed external information such as optical flow, and effectively localizes the dynamic motion of the scene. In short, our contributions are as follows:

- We propose a novel framework MoDec-GS based on Global-to-Local Motion Decomposition (GLMD), which effectively handles real world's complex motions composed of global and local movements.
- We introduce the TIA to adaptively control the temporal intervals of each local canonical anchor during training,

enabling our MoDec-GS to achieve optimal visual quality even with a compactly limited model size.
- Extensive experiments on three widely used monocular datasets show that our method significantly reduces storage while maintaining or even improving visual quality, specifically, on iPhone dataset [16], it shows a PSNR gain of + 0.7dB and a storage reduction of -94% compared to the second best method in terms of quality, SC-GS [21].

2. Related Works

2.1. 3D Gaussian Splatting for Dynamic Scenes

A natural evolution of 3DGS [26] for static scenes is its extension to dynamic scenes, with recent research in this area generally split into two main categories: deformation-based and 4D Gaussian-based methods. Deformation-based methods rely on a static canonical 3DGS, paired with a component that captures the temporal deformation of the attributes within this canonical representation. This deformation can be modeled implicitly using structures like MLP [21, 33, 63] or feature grids [9, 60], or explicitly via polynomial [32, 34], Fourier [24, 34] or learned basis functions [27]. In contrast, 4D Gaussian-based methods [8, 62] introduce time as an extra dimension in the 3D Gaussian formulation. Both works aim to integrate the 4D Gaussian paradigm within the established 3DGS training and rendering framework, with a particular focus on efficiently representing rotations in 4D space.

Although both categories achieve decent rendering quality for dynamic scenes, they still demand substantial storage, to handle the multi-dimensional attributes of millions of 3D Gaussians in canonical 3DGS or 4D Gaussians [29, 66]. Additionally, representing long-duration or large-motion contents with a compact model is challenging [54]. This difficulty arises from the need to maintain fast rendering speed while ensuring reliable training on a limited set of sampled videos. A straightforward approach is to train a separate model for each empirically determined interval; however, determining the optimal interval empirically is inefficient, as the flow of motion information can vary significantly depending on the content. Shaw et al. [54] proposed a method for temporal segmentation to address this issue, but this considers only the magnitude of motion. In contrast, we integrates a temporal segmentation method directly into the training process, enabling MoDec-GS to explore optimal intervals by itself to achieve the best rendering quality.

2.2. Compact 3D Gaussian Splatting

To address the substantial memory demands of 3DGS, various strategies have been proposed. The first category focuses on compressing the original 3DGS representation, with key approaches including vector quantization [10, 28, 46, 47], pruning redundant Gaussians [10, 28, 57], implicit encoding of high-dimensional attributes [18, 28, 61], using standardized compression pipelines [10, 43, 61] and applying entropy constraint [57]. The second category explores more efficient Gaussian representations to mitigate storage challenges [20, 40]. A prominent example is Scaffold-GS, which introduces a unique method by assigning learnable features to a sparse set of anchor points that predict attributes for a broader set of neighboring 3D Gaussians. Recent advancements in the Scaffold-GS framework have further enhanced memory efficiency by organizing anchor points hierarchically across multiple levels [58] or using a binary hash grid to model context for unstructured anchor attributes [4].

However, adapting methods from both categories to 4DGS may not be straightforward, as most 4D extensions require substantial architectural modifications to extend 3DGS for dynamic scene modeling. A few recent approaches have covered both dynamic extension and memory efficiency; for instance, Sun et al. [56] propose a framework for on-the-fly training, where adaptive control over the quantity of 3D Gaussians is employed, allowing the model size to remain moderate for streaming. However, this method assumes multi-view inputs. Lee et al. [29] implement a combination of compression techniques, including residual vector quantization and hash grid-based encoding, on top of the Spacetime Gaussian proposed by Li et al. [32]. However, this approach doesn't provide an efficient solution for handling long-duration, large-motion content.

3. Preliminary

3.1. Splatting of Gaussian primitives

Gaussian primitives, or Gaussians, are characterized using their respective rotation matrices R and scaling matrices $S = diag([s_i])$. To render the primitives for a viewport that corresponds to a viewing transformation matrix W, one can calculate the following covariance matrix:

$$\Sigma' = J^T W^T \Sigma W J, \tag{1}$$

where $\Sigma = R^T S^T S R$ and J is an affine approximation of the projective transformation [26]. The covariance matrix represents the approximate shape of the projected Gaussian. Once the 2D covariance matrices and the projected central positions for each Gaussians are calculated, the primitives are sorted in the order of the depth values. Lastly, the colors for each pixel in the viewport can be calculated by the following alpha-blending procedure:

$$\boldsymbol{C} = \sum_{i=1}^{n} \mathbf{c}_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \tag{2}$$

where $\mathbf{c}_i$ is the color of the i-th primitive that is calculated from spherical harmonic coefficients. The opacity α_i is obtained by evaluating the 2D Gaussian distribution function at the pixel position.

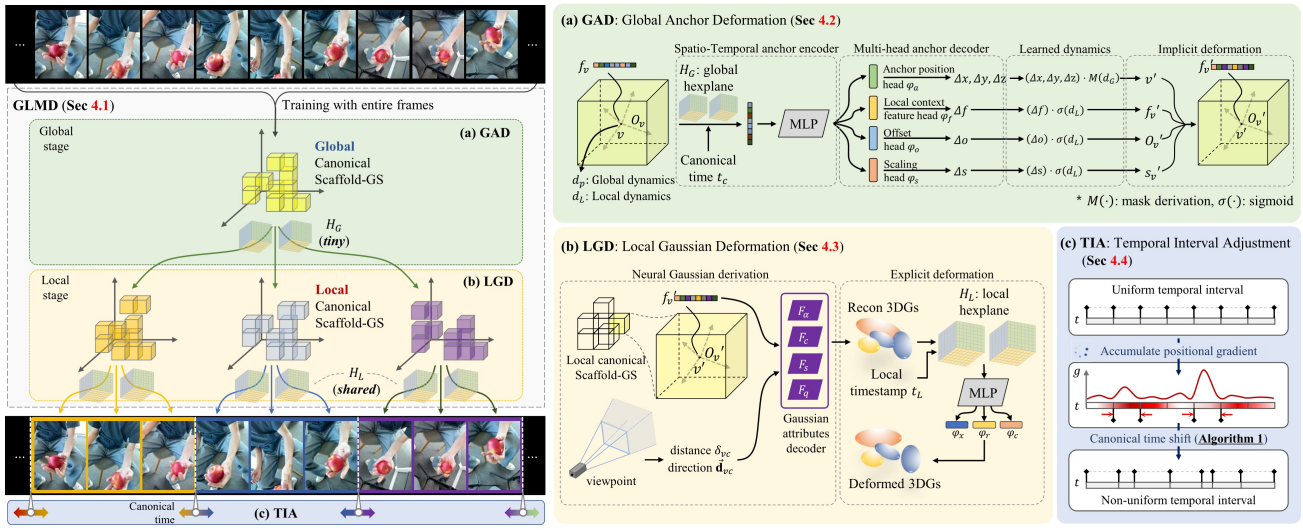

Figure 2. **Overview of our MoDec-GS framework.** To effectively train dynamic 3D Gaussians with complex motion, we introduce Global-to-Local Motion Decomposition (GLMD) (Sec 4.1). We first train a Global Canonical Scaffold-GS (Global CS) with entire frames, and apply a Global Anchor Deformation (GAD) to Local Canonical Scaffold-GS (Local CS) dedicated to represent its corresponding temporal segment (Sec 4.2). Next, to finely adjust the remaining local motion, we apply Local Gaussian Deformation (LGD) which explicitly deforms the reconstructed 3D Gaussians with a shared hexplane (Sec 4.3). During the training, Temporal Interval Adjustment (TIA) is performed, optimizing the temporal interval into a non-uniform interval that adopts to the scene's level of motion (Sec 4.4).

3.2. Scaffold-GS

Scaffold-GS representation consists of a set of anchor points which are the central points of voxels of a predefined size, a set of neural Gaussians associated with the anchor points, and a set of neural networks that predict the attributes of the neural Gaussians. There are k-number of neural Gaussians that are associated with each anchor point, and such a group of Gaussians that are softly bound to a spatial point work as a local representation. The centers of the neural Gaussians are given as follows:

$$\mathbf{m}_i = \mathbf{x}_v + \mathbf{o}_i, \tag{3}$$

where $i \in \{0, ..., k-1\}$ and $\mathbf{x}_v$ is the position of an anchor point v. $\mathbf{m}_i$ and $\mathbf{o}_i$ are the center position and the learnable offset for the i-th neural Gaussian. The opacities of the neural Gaussians are predicted as follows:

$$\{\text{attr}_{v,0}, \cdots, \text{attr}_{v,k-1}\} = F_{\text{attr}}(\hat{f}_v, \delta_{v,\text{cam}}, \vec{\mathbf{d}}_{v,\text{cam}}) \tag{4}$$

where $\text{attr}_{v,i}$ is the attribute of the i-th neural Gaussian associated with the anchor point v. The attributes include opacity, color, quaternion, and scale. Separate neural networks F_{attr} are used to predict the attributes where the networks take inputs including learnable anchor feature $\hat{f}_v$ and the displacement $(\delta_{v,\text{cam}}, \vec{\mathbf{d}}_{v,\text{cam}})$ from the viewing position to the anchor v. Similar to the densification in 3DGS, the anchor points are added or removed based on the gradient accumulation and the opacity. Please see *Suppl.* C.1 for detail.

4. Proposed Method

4.1. Overview of MoDec-GS

We adopt a deformation-based real-time dynamic scene rendering method [60], but use anchor-based representation [40] as a canonical 3DGS due to its compactness. To effectively capture real-world videos with a complex combination of global and local motions, we introduce Global-to-Local Motion Decomposition (GLMD) as illustrated in Fig. 2. GLMD consists of two stages: the first models global motion, while the second refines local motion. In the first stage, we apply Global Anchor Deformation (GAD), which deforms the position and attributes of anchors with a tiny global hexplane, transforming the Global Canonical Scaffold-GS (Global CS) into the Local Canonical Scaffold-GS (Local CS) (Sec. 4.2). As shown in Fig. 3, this anchor-based transformation effectively captures global motion. Additionally, we embed learnable parameters into anchors to reflect motion characteristics, enabling effective control over both anchor-wise global motion and local motion within each anchor. In the second stage, the Local CS is reconstructed into 3DGs through neural Gaussian derivation and then explicitly deformed to each target timestamp using a shared local hexplane (Sec. 4.3). To optimize the temporal interval assigned to each Local CS based on the scene motion, we propose Temporal Interval Adjustment (TIA) (Sec. 4.4). This method dynamically re-balances temporal intervals during training, efficiently utilizing limited representation capability, without requiring any precomputed external information such as optical flow or tracking [30, 35, 37].

4.2. Global Anchor Deformation (GAD)

Anchor Deformation. One approach to representing dynamic motion based on deformation is learning a hexplane that deforms 3DGs attributes after reconstruction [60]. While intuitive and efficient, it may struggle to handle a complex combinations of global and local motions due to the hexplane's limited capacity. In contrast, another method to achieve this involves directly deforming the anchor's position and attributes in anchor-based representation [40]. As shown in Fig 3, the method of deforming the anchor itself is more efficient for representing a global motion of relatively large objects, rather than learning deformation fields for each reconstructed individual 3DGs. Therefore, as shown in Fig. 2-(a), we deform the anchor's position and its attributes in GAD stage. For a given anchor point v, the anchor position x_v, y_v, z_v is queried in a tiny global hexplane H_G along with a timestamp. Here, the timestamp corresponds to the canonical time t_c, which is a time representing each divided temporal segment, determining temporal interval of the Local CS. The queried feature is decoded by a tiny MLP and a multi-head anchor decoder, producing the deformations for the position and attributes associated with the anchor: $(\Delta x, \Delta y, \Delta z)$, Δf_v, ΔO_v, Δs_v. For example, a deformation of local context feature can be obtained by

$$\Delta f_v = \varphi_f[F_G(H_G(x_v, y_v, z_v, t_c)], \tag{5}$$

where F_G is a light MLP and φ_f is a local context feature head among the multi-head anchor decoders. The deformation values are added to the anchor attributes, producing a deformed anchor, at which point the proposed novel term, learnable *anchor dynamics*, are incorporated.

Anchor Dynamics. Even for a long-range video, a considerable portion of the scene is still static. Rather than separately generating static and dynamic parts [33] or utilizing a precomputed external dynamic mask [41], we aim to learn motion dynamics by assigning additional learnable attributes to the anchor, allowing them to be optimized during the training process. To separately model the global movement characteristics of the anchor and the local movements within the anchor, we applied and trained d_G and d_L independently. Global dynamics d_G learns whether the entire anchor moves globally and applies binary masking based on a threshold. This learnable masking inspired by [28] is derived as follows:

$$M(d_G) = \text{sg}(\mathcal{I}[\sigma(d_G) > \epsilon] - \sigma(d_G)) + \sigma(d_G), \tag{6}$$

where $\text{sg}(\cdot)$ is the stop gradient operator, $\mathcal{I}$ is an indicator, $\sigma(\cdot)$ is sigmoid function, and ϵ is the masking threshold. Local dynamics d_L is simply activated and then multiplied to the attributes of the corresponding anchor. Finally, the attributes of the deformed anchor are given by

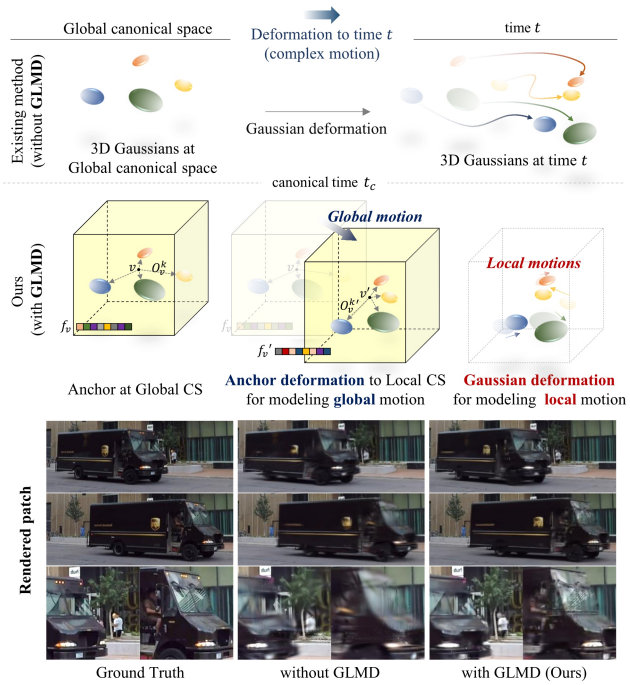

Figure 3. **Concept and effect of 2-stage deformation.** For representing a complex motion of 3D Gaussians, a global movement over time intervals can be more efficiently handled through deformation of anchor itself. In contrast, subtle motions of individual 3D Gaussians within a time interval can be effectively addressed by explicit deformation of each Gaussian.

$$x_{v'}, y_{v'}, z_{v'} = (x_v, y_v, z_v) + M(d_G) \cdot (\Delta x, \Delta y, \Delta z) \tag{7}$$

$$f_{v'} = f_v + \Delta f \cdot \sigma(d_L), \tag{8}$$

$$o_{v'} = o_v + \Delta o \cdot \sigma(d_L), \tag{9}$$

$$s_{v'} = s_v + \Delta s \cdot \sigma(d_L). \tag{10}$$

4.3. Local Gaussian Deformation (LGD)

Once global motion over a time interval is captured in the first stage, the remaining local motion of individual 3D Gaussians are relatively minor and simplified, as shown in Fig. 3. This claim is also supported by *Suppl.* G.3. Representing such movements can be effectively handled by the explicit deformation [60] of the reconstructed Gaussians, rather than anchor deformation, since it would require learning feature changes capable of generating the attributes of the displaced Gaussians (See Sec. 3 and Tab. 3). Based on this reasoning, a deformed Local CS is first reconstructed into 3D Gaussians through neural Gaussian derivation.

Neural Gaussian Derivation. Within a given view frustum, k neural Gaussians are spawned from the deformed anchor, and each Gaussian's attributes are reconstructed using the deformed feature along with the viewing direction and distance. For instance, an opacity set of k Gaussians is spawned as follows:

$$\{\alpha_0, \cdots, \alpha_{k-1}\} = F_\alpha(\hat{f}'_v, \delta_{v',\text{cam}}, \vec{\mathbf{d}}_{v',\text{cam}}), \tag{11}$$

where F_α is MLP decoder, $\hat{f'}_v$ is a feature bank constructed from the deformed feature on anchor v', $\delta_{v',\text{cam}}$ and $\overrightarrow{\mathbf{d}}_{v',\text{cam}}$ are relative distance and viewing direction from viewpoint to the anchor, respectively [40].

Gaussian Deformation. The spawned neural Gaussians are then explicitly deformed to the target timestamp [60]. For example, positional deformations of k-th neural Gaussian in a Local CS is given by:

$$\Delta x_k, \Delta y_k, \Delta z_k = \varphi_p[F_L(H_L(x_k, y_k, z_k, t_L)] \quad (12)$$

where H_L is a local multi-resolution hexplane, F_L is a MLP decoder, φ_p is a position head, and t_L is a target timestamp. Note that the local hexplane and corresponding MLP decoder are shared across each Local CS for compactness.

4.4. Temporal Interval Adjustment (TIA)

To adaptively localize the degree of motion and guide the temporal scope of each Local CS, we divided the total frames N into l segments, where $1 < l < N$. Initially, segments have uniform temporal intervals. However, depending on the scene motion characteristics, their optimal sizes that each Local CS can effectively represent may vary. To account for this, we propose Temporal Interval Adjustment (TIA) to adjust the temporal intervals to fit the scene during the training process. The TIA re-balances the complexity of deformation required for each local canonical Gaussians, enabling effective scene representation even with *compactly* limited size of hexplane.

Canonical time shift. For the temporal interval adjustment, we employ the canonical time-based shift method, as illustrated in Fig. 2-(c). Basically, temporal intervals are managed by a canonical time list $T_c = [t_1, t_2, \cdots, t_{l-1}]$, which represents the lowest timestamp of each interval and serves as the boundary between temporal segments. During the training process, this list is fixed at equal intervals from the normalized entire time range $[0, 1]$, until a preset iteration for starting temporal adjustment, $T^{\text{TIA}}_{\text{from}}$. After the starting iteration, the TIA process, as described in Algo. 1, is repeated until a preset iteration for ending the process, $T^{\text{TIA}}_{\text{until}}$. During each adjustment period $T^{\text{TIA}}_{\text{period}}$, positional gradients are accumulated in the temporal interval to where the timestamp of each training view belongs. The accumulated gradient list $\mathbf{G}^{\text{acc}} = [g^{\text{acc}}_0, g^{\text{acc}}_1, \cdots, g^{\text{acc}}_{l-1}]$ and the accumulation count list $\nu^{\text{acc}} = [\nu^{\text{acc}}_0, \nu^{\text{acc}}_1, \cdots, \nu^{\text{acc}}_{l-1}]$, are initially set to zero, and updated in each iteration as follows:

$$g^{\text{acc}}_c = g^{\text{acc}}_c + g^{pos}_t, \quad (13)$$

$$\nu^{\text{acc}}_c = \nu^{\text{acc}}_c + 1, \quad (14)$$

where g^{pos}_t is the Frobenius norm of positional gradient for a certain iteration where the training view has timestamp t within $[t_c, t_{c+1})$. Please note that the left boundary of the first temporal interval should not be represented as an adjustable canonical time but always be fixed at 0. Therefore, T_c is one element shorter in length than $\mathbf{G}^{\text{acc}}$. Based on the statistics of the accumulated gradients, we identify Local CS with insufficient expressiveness, and shrink the corresponding temporal intervals. The idea behind this approach is that temporal segments with significantly high accumulated positional gradient indicate regions where the Local CS struggles to represent efficiently; therefore, we reduce their assigned time intervals. Each temporal interval with an accumulated gradient greater than the preset threshold τ_{TIA} is shrunk by a step size s_{TIA} on both sides.

Algorithm 1 Temporal Interval Adjustment (Fig. 2-(c))

procedure TIA(T_c, $\mathbf{G}^{\text{acc}}$, ν^{acc}, g^{pos}_t, τ_{TIA}, s_{TIA})
  if $T^{\text{TIA}}_{\text{from}} \leq$ iter $\leq T^{\text{TIA}}_{\text{until}}$ **then**
    Update $\mathbf{G}^{\text{acc}}$ with g^{pos}_t (Eq. 13)
    if iter $\%\ T^{\text{TIA}}_{\text{period}} = 0$ **then**
      $\mu = \sum_{c=0}^{l-1}(g^{\text{acc}}_c / \nu^{\text{acc}}_c)/l$ ▷ acc. grad. mean
      $\sigma = \sqrt{\sum_{c=0}^{l-1}[(g^{\text{acc}}_c / \nu^{\text{acc}}_c) - \mu]^2/l}$ ▷ std.
      for $j = 0$ to $l-1$ **do**
        if $g^{\text{acc}}_j \geq \mu + \tau_{\text{TIA}} \cdot \sigma$ **then** ▷ shrink
          if $j \neq 0$ **and** $t_j \leq t_{j+1} - s_{\text{TIA}}$ **then**
            $t_j \leftarrow t_j + s_{\text{TIA}}$
          if $j \neq l-1$ **and** $t_j \leq t_{j+1} - s_{\text{TIA}}$ **then**
            $t_{j+1} \leftarrow t_{j+1} - s_{\text{TIA}}$
      Init $\mathbf{G}^{\text{acc}}$, ν^{acc}, μ, σ

5. Experiments

5.1. Experimental Setup

Implementation Details. Our framework is built upon 4DGS [60] and Scaffold-GS [40], retaining most hyperparameters. The parameters related to newly designed modules are empirically derived (see *Suppl.* B for the details).

Datasets and Metrics. There are various and well-validated multi-view videos datasets [31, 52], but achieving high rendering quality especially in *monocular* reconstruction remains *challenging*, which is the most accessible real-life application due to the prevalence of camera-equipped mobile devices. To focus on the complex motion present in such real-world videos, we evaluate our method on recent monocular video benchmark, Dycheck-iPhone [16], which closely reflects the real-life video characteristics without teleporting. We also used HyperNeRF [49] and Nvidia-monocular dataset [64], which are widely used for monocular evaluation, employed to assess the generalization performance of our method. For all three datasets, initial point cloud data was manually generated by COLMAP [53] using the script provided in [60]. The image quality of our approach is evaluated using three metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) [65]. Each metric is computed per frame and subsequently averaged across all test frames. Storage efficiency is measured in megabytes (MB) by summing the size of the trained

Method	Apple		Block		Paper-windmill		Space-out	
SC-GS [21]	14.96 / 0.692 / 0.508	173.3	13.98 / 0.548 / 0.483	115.7	14.87 / **0.221** / 0.432	446.3	**14.79** / 0.511 / **0.440**	114.2
Deformable 3DGS [63]	15.61 / 0.696 / **0.367**	87.71	14.87 / 0.559 / **0.390**	118.9	14.89 / 0.213 / **0.341**	160.2	14.59 / 0.510 / 0.450	42.01
4DGS [60]	15.41 / 0.691 / 0.524	61.52	13.89 / 0.550 / 0.539	63.52	14.44 / 0.201 / 0.445	123.9	14.29 / 0.515 / 0.473	52.02
MoDec-GS (Ours)	**16.48** / **0.699** / 0.402	**23.78**	**15.57** / **0.590** / 0.478	**13.65**	**14.92** / 0.220 / 0.377	**17.08**	14.65 / **0.522** / 0.467	**18.24**

	Spin		Teddy		Wheel		**Average**	
SC-GS [21]	14.32 / 0.407 / 0.445	219.1	12.51 / 0.516 / **0.562**	318.7	11.90 / 0.354 / 0.484	239.2	13.90 / 0.464 / 0.479	232.4
Deformable 3DGS [63]	13.10 / 0.392 / 0.490	133.9	11.20 / 0.508 / 0.573	117.1	11.79 / 0.345 / **0.394**	106.1	13.72 / 0.461 / **0.430**	109.4
4DGS [60]	14.89 / 0.413 / 0.441	71.80	12.31 / 0.509 / 0.605	80.44	10.83 / 0.339 / 0.538	96.50	13.72 / 0.460 / 0.509	78.54
MoDec-GS (Ours)	**15.53** / **0.433** / **0.366**	**26.84**	**12.56** / **0.521** / 0.598	**12.28**	**12.44** / **0.374** / 0.413	**16.68**	**14.60** / **0.480** / 0.443	**18.37**

Table 1. **Quantitative results comparison on the iPhone datasets [16].** **Red** and blue denote the best and the second best performances, respectively. Each block element of 4-performance denotes (mPSNR(dB)↑ / mSSIM↑ / mLPIPS↓ Storage(MB)↓).

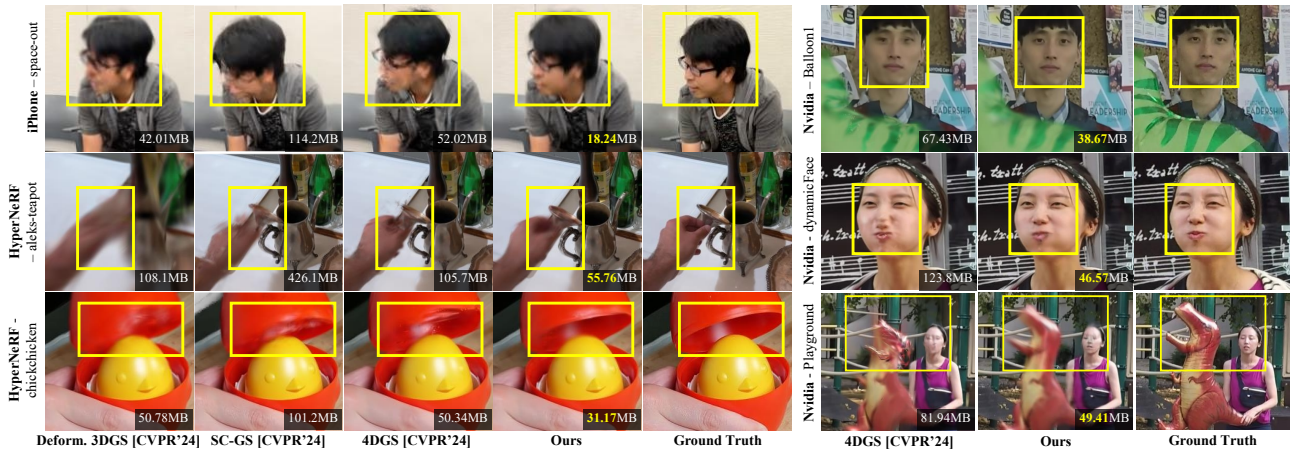

Figure 4. **Qualitative results comparison on three datasets [16, 49, 64].** The yellow boxes highlight areas where the proposed method achieves notable visual quality improvements, and the storage for the corresponding sequence is displayed below each rendered patch.

model. For iPhone dataset, we used the masked metrics based on the official co-visible mask provided by [16].

Comparison Methods. We compare our approach with recent dynamic 3D Gaussian representation methods that can reconstruct dynamic scenes from a monocular video footage: Deformable-3DGS [63], SC-GS [21], and 4DGS [60]. The official codes are used for the methods, and we adjusted several parameters to achieve reasonable rendering quality, with details provided in the *Suppl.* B.

5.2. Results

Quantitative Comparison. As detailed in Tab. 1, our method significantly reduces storage while achieving the best or second-best visual quality performance across almost all sequences of iPhone dataset [16]. On average, it maintains or even improves visual quality with only about 6% of the storage compared to the second-best method in terms of quality, SC-GS [21]. On HyperNeRF dataset [49], we present only the average performance shown as Tab. 2-(a). For this dataset, our method attains the best performance in PSNR, SSIM and second-best in LPIPS, while having approximately 18% of SC-GS's storage [21] and around 57% of 4DGS's [60]. For the Nvidia dataset [64], we compare our method only with the second-best method in terms of visual quality relative to storage, 4DGS [60]. Tab. 2-(b) shows that our method reduces storage while simultaneously improving visual quality in all metrics. Supplementary materials includes all per-sequence results (Tab. 9), additional results on synthetic (Tab. 5) and real-world (Tab. 6) datasets, and comparison with NeRF-extension frameworks (Tab. 4). Please refer to it for further details.

Qualitative Comparison. To evaluate the visual quality of the proposed method, we conducted qualitative assessments on three datasets [16, 49, 64] shown in Fig. 4. We focused particularly on regions where dynamic objects are in motion. As introduced in Fig. 1, while comparison methods struggle with handling complex motion, our method demonstrates better quality in regions with such motion, thanks to GLMD. Furthermore, as shown in the NVIDIA dataset results, our method maintains fine visual quality even when static objects exhibit only local changes (e.g., facial expressions). This is due to TIA effectively localizing the coverage of Local CS. Not only does our method achieve these quality improvements, but it also maintains storage requirements at about half the average size of compared methods.

Methods	(a) HyperNeRF PSNR↑	SSIM↑	LPIPS↓	Storage↓
SC-GS [CVPR'24] [21]	26.95	0.815	**0.213**	226.0
Deformable 3DGS [CVPR'24] [63]	25.96	0.766	0.294	87.13
4DGS [CVPR'24] [60]	27.44	0.797	0.302	72.65
Ours	**27.78**	**0.827**	0.219	**40.82**

Methods	(b) Nvidia PSNR↑	SSIM↑	LPIPS↓	Storage↓
4DGS [CVPR'24] [60]	25.82	0.844	0.219	67.44
Ours	**26.65**	**0.876**	**0.171**	**39.64**

Table 2. **Quantitative results comparison on (a) HyperNeRF [49] and (b) Nvidia monocular [64] dataset**.

Variant	mPSNR↑	mSSIM↑	mLPIPS↓	Storage↓
(a) 1stage, Gaussian deform ([60])	13.73	0.460	0.509	78.54
(b) 1stage, anchor deform	13.56	0.449	0.510	36.92
(c) 2stage, all anchor deform	13.93	0.453	0.492	55.29
(d) 2stage, GAD + LGD (**GLMD**)	14.48	0.475	0.455	49.70
(e) (d) with smaller hexplane	14.46	0.475	0.451	22.67
(f) (e) with d_G and d_L (anchor dynamics)	14.51	0.478	0.447	22.72
(g) (f) with **TIA** (our final MoDec-GS)	**14.60**	**0.480**	**0.443**	**18.37**

Table 3. **Ablation studies on MoDec-GS components**. Each row evaluates the impact of a specific design choice. Yellow-green cells highlight configurations with substantial storage reduction.

5.3. Analysis

Ablation studies. We analyze the effectiveness of the components in our MoDec-GS through comprehensive ablation studies as shown in Tab 3. All results are averaged over all the iPhone sequences [16]. Note that our baseline is a single-stage deformation method that explicitly deforms Gaussians, as in [60]. We first examine the effectiveness of our anchor deformation - (b). Leveraging anchor-based representation [40] slightly reduces performance but significantly cuts storage (52% reduction). Configuring the Local CS by adding a global hexplane - (c), we observe that the performance improvement outweighs the increase in storage due to the additional grid, compared to (a). We then show the effectiveness of LGD - (d). Instead of applying consistent anchor deformation for both global and local deformation, performing LGD after reconstructing the neural Gaussian noticeably improves performance. It also reduces storage by allowing smaller Global CS via anchor adjustment, as the regions represented by adjacent Gaussians no longer require anchors. We further analyzed GLMD's handling of complex motion via optical flow in *Suppl*. G.3 and G.4. The efficient design of GLMD allows for a reduction in the size of the global and local hexplanes with minimal impact on visual quality - (e) (55% reduction). By adding the proposed learnable parameters that control anchor's motion dynamics, the increase is almost negligible while improving visual quality to some extent. Finally, applying TIA to this variant - our final MoDec-GS - enables both quality improvement and storage reduction. When temporal intervals are appropriately adapted to the degree of motion in the scene through TIA, the representational capacity of the limited-size hexplane can be utilized more efficiently, and also the Gaussian movements represented by each LGD become simpler (See Fig. 3), which further enhances the efficiency of anchor deformation. Additional experiments verified TIA's effects. We precomputed optical flow [55] to measure the degree of motion in the scene, and evaluated how the TIA responds to it. As shown in Fig. 5, initially uniform temporal intervals (black dotted line) are adjusted into non-uniform intervals (blue solid line) during the training process, shrinking and shifting toward regions with relatively higher normalized optical flow magnitude. Examining accumulated flow confirms TIA effectively *rebalances* the degree of motion across intervals. Overall, each additional component in MoDec-GS contributes to either improved visual fidelity or enhanced memory efficiency, with our final configuration achieving the best balance.

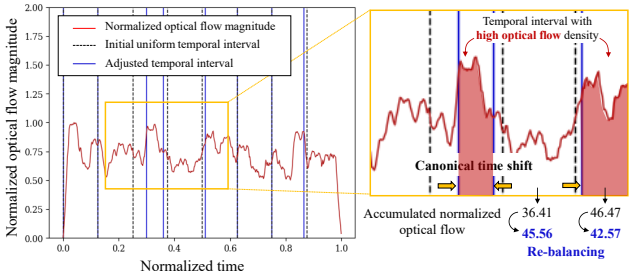

Figure 5. **Effectiveness of TIA.** Initially uniform intervals (black dotted lines) are adaptively reallocated based on motion complexity (blue lines), as indicated by normalized optical flow magnitude.

6. Conclusion

We propose MoDec-GS, a novel compact framework for high-quality dynamic 3D Gaussian splatting, addressing storage demands and complex motion challenges in dynamic scene reconstruction. By utilizing Global-to-Local Motion Decomposition (GLMD), which incorporates Global Anchor Deformation (GAD) for global motion and Local Gaussian Deformation (LGD) for fine-grained local adjustments, MoDec-GS effectively captures complex motions while minimizing storage use. Additionally, our Temporal Interval Adjustment (TIA) allows adaptive temporal segmentation, across dynamic intervals without requiring external motion data. Extensive evaluations confirm that MoDec-GS significantly reduces model size—up to average 70%—while either preserving or enhancing rendering quality across challenging datasets, offering a compact yet powerful solution for real-world dynamic 3D reconstruction.

Acknowledgements

This work was supported by the IITP grant funded by the Korea government (MSIT): Development of immersive video spatial computing technology for ultra-realistic metaverse services (No.2022-0-00022, RS-2022-II220022) and the Graduate School of Metaverse Convergence support program (RS-2024-00418847)

References

[1] Jill M Boyce, Renaud Doré, Adrian Dziembowski, Julien Fleureau, Joel Jung, Bart Kroon, Basel Salahieh, Vinod Kumar Malamal Vadakital, and Lu Yu. Mpeg immersive video coding standard. *Proceedings of the IEEE*, 109(9):1521–1536, 2021. 2

[2] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics (TOG)*, 39(4):86–1, 2020. 2

[3] Brian Chao, Hung-Yu Tseng, Lorenzo Porzi, Chen Gao, Tuotuo Li, Qinbo Li, Ayush Saraf, Jia-Bin Huang, Johannes Kopf, Gordon Wetzstein, et al. Textured gaussians for enhanced 3d scene appearance modeling. *arXiv preprint arXiv:2411.18625*, 2024. 5

[4] Yihang Chen, Qianyi Wu, Weiyao Lin, Mehrtash Harandi, and Jianfei Cai. Hac: Hash-grid assisted context for 3d gaussian splatting compression. In *European Conference on Computer Vision*, pages 422–438. Springer, 2025. 2, 3

[5] 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. 2

[6] Mengyu Chu, You Xie, Laura Leal-Taixé, and Nils Thuerey. Temporally coherent gans for video super-resolution (tecogan). *arXiv preprint arXiv:1811.09393*, 1(2):3, 2018. 4

[7] Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixé, and Nils Thuerey. Learning temporal coherence via self-supervision for gan-based video generation. *ACM Transactions on Graphics (TOG)*, 39(4):75–1, 2020. 3, 6

[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 *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2, 3

[9] Bardienus P Duisterhof, Zhao Mandi, Yunchao Yao, Jia-Wei Liu, Mike Zheng Shou, Shuran Song, and Jeffrey Ichnowski. Md-splatting: Learning metric deformation from 4d gaussians in highly deformable scenes. *arXiv preprint arXiv:2312.00583*, 2023. 2, 3

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

[11] Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 2, 3, 4

[12] Christoph Fehn. Depth-image-based rendering (dibr), compression, and transmission for a new approach on 3d-tv. In *Proceedings of SPIE 5291, Stereoscopic Displays and Virtual Reality Systems XI*, pages 93–104, San Jose, California, United States, 2004. International Society for Optics and Photonics. 2

[13] L Franke, D Rückert, L Fink, and M Stamminger. Trips: Trilinear point splatting for real-time radiance field rendering. arxiv abs/2401.06003 (2024). 5

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

[15] Wanshui Gan, Hongbin Xu, Yi Huang, Shifeng Chen, and Naoto Yokoya. V4d: Voxel for 4d novel view synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 2023. 3, 4

[16] Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. pages 33768–33780, 2022. 3, 6, 7, 8, 2, 4

[17] Qiankun Gao, Yanmin Wu, Chengxiang Wen, Jiarui Meng, Luyang Tang, Jie Chen, Ronggang Wang, and Jian Zhang. Relaygs: Reconstructing dynamic scenes with large-scale and complex motions via relay gaussians. *arXiv preprint arXiv:2412.02493*, 2024. 2

[18] Sharath Girish, Kamal Gupta, and Abhinav Shrivastava. Eagles: Efficient accelerated 3d gaussians with lightweight encodings. *arXiv preprint arXiv:2312.04564*, 2023. 2, 3

[19] Xiang Guo, Jiadai Sun, Yuchao Dai, Guanying Chen, Xiaoqing Ye, Xiao Tan, Errui Ding, Yumeng Zhang, and Jingdong Wang. Forward flow for novel view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16022–16033, 2023. 2, 3

[20] Abdullah Hamdi, Luke Melas-Kyriazi, Jinjie Mai, Guocheng Qian, Ruoshi Liu, Carl Vondrick, Bernard Ghanem, and Andrea Vedaldi. Ges: Generalized exponential splatting for efficient radiance field rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19812–19822, 2024. 3, 5

[21] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4220–4230, 2024. 1, 2, 3, 7, 8, 5, 6

[22] Saqib Javed, Ahmad Jarrar Khan, Corentin Dumery, Chen Zhao, and Mathieu Salzmann. Temporally compressed 3d gaussian splatting for dynamic scenes. *arXiv preprint arXiv:2412.05700*, 2024. 4

[23] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE international conference on computer vision*, pages 3334–3342, 2015. 4

[24] Kai Katsumata, Duc Minh Vo, and Hideki Nakayama. An efficient 3d gaussian representation for monocular/multi-view dynamic scenes. *arXiv preprint arXiv:2311.12897*, 2023. 2, 3

[25] Kai Katsumata, Duc Minh Vo, and Hideki Nakayama. A compact dynamic 3d gaussian representation for real-time

dynamic view synthesis. In *European Conference on Computer Vision*, pages 394–412. Springer, 2024. 4

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

[27] Agelos Kratimenos, Jiahui Lei, and Kostas Daniilidis. Dynmf: Neural motion factorization for real-time dynamic view synthesis with 3d gaussian splatting. In *European Conference on Computer Vision*, pages 252–269. Springer, 2025. 2, 3

[28] 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*, pages 21719–21728, 2024. 2, 3, 5

[29] Joo Chan Lee, Daniel Rho, Xiangyu Sun, Jong Hwan Ko, and Eunbyung Park. Compact 3d gaussian splatting for static and dynamic radiance fields. *arXiv preprint arXiv:2408.03822*, 2024. 2, 3

[30] Jiahui Lei, Yijia Weng, Adam Harley, Leonidas Guibas, and Kostas Daniilidis. Mosca: Dynamic gaussian fusion from casual videos via 4d motion scaffolds. *arXiv preprint arXiv:2405.17421*, 2024. 4

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

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

[33] 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 *ArXiv*, 2024. 2, 3, 5

[34] Youtian Lin, Zuozhuo Dai, Siyu Zhu, and Yao Yao. Gaussian-flow: 4d reconstruction with dynamic 3d gaussian particle. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21136–21145, 2024. 2, 3

[35] Bangya Liu and Suman Banerjee. Swings: Sliding window gaussian splatting for volumetric video streaming with arbitrary length. *arXiv preprint arXiv:2409.07759*, 2024. 4

[36] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 2

[37] Qingming Liu, Yuan Liu, Jiepeng Wang, Xianqiang Lv, Peng Wang, Wenping Wang, and Junhui Hou. Modgs: Dynamic gaussian splatting from causually-captured monocular videos. *arXiv preprint arXiv:2406.00434*, 2024. 4

[38] 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 32nd ACM International Conference on Multimedia*, pages 2936–2944, 2024. 2

[39] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13–23, 2023. 2

[40] Tao Lu, Mulin Yu, Linning Xu, Yuanbo Xiangli, Limin Wang, Dahua Lin, and Bo Dai. Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20654–20664, 2024. 2, 3, 4, 5, 6, 8, 1

[41] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pages 800–809. IEEE, 2024. 5, 4

[42] 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 *Communications of the ACM*, pages 99–106, 2021. 2

[43] Wieland Morgenstern, Florian Barthel, Anna Hilsmann, and Peter Eisert. Compact 3d scene representation via self-organizing gaussian grids. *arXiv preprint arXiv:2312.13299*, 2023. 2, 3

[44] Yuji Mori, Norishige Fukushima, Tomohiro Yendo, Toshiaki Fujii, and Masayuki Tanimoto. View generation with 3d warping using depth information for ftv. *Signal Processing: Image Communication*, 24(1-2):65–72, 2009. 2

[45] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. In *ACM Transactions on Graphics (TOG)*, pages 102:1–102:15, 2022. 2

[46] KL Navaneet, Kossar Pourahmadi Meibodi, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. Compact3d: Compressing gaussian splat radiance field models with vector quantization. *arXiv preprint arXiv:2311.18159*, 2023. 2, 3

[47] 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*, pages 10349–10358, 2024. 2, 3

[48] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 2, 3

[49] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M Seitz. Hypernerf: A higher-dimensional representation for topologically varying neural radiance fields. In *SIGGRAPH Aisa*, 2021. 1, 6, 7, 8, 2, 3

[50] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10318–10327, 2021. 4

[51] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 5

[52] 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 Workshops*, pages 30–40, 2017. 6

[53] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 6, 3

[54] Richard Shaw, Michal Nazarczuk, Jifei Song, Arthur Moreau, Sibi Catley-Chandar, Helisa Dhamo, and Eduardo Pérez-Pellitero. Swings: sliding windows for dynamic 3d gaussian splatting. In *European Conference on Computer Vision*. Springer, 2024. 2, 3

[55] Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1599–1610, 2023. 8, 4

[56] Jiakai Sun, Han Jiao, Guangyuan Li, Zhanjie Zhang, Lei Zhao, and Wei Xing. 3dgstream: On-the-fly training of 3d gaussians for efficient streaming of photo-realistic free-viewpoint videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20675–20685, 2024. 2, 3

[57] Henan Wang, Hanxin Zhu, Tianyu He, Runsen Feng, Jiajun Deng, Jiang Bian, and Zhibo Chen. End-to-end rate-distortion optimized 3d gaussian representation. *arXiv preprint arXiv:2406.01597*, 2024. 3

[58] Yufei Wang, Zhihao Li, Lanqing Guo, Wenhan Yang, Alex C Kot, and Bihan Wen. Contextgs: Compact 3d gaussian splatting with anchor level context model. *arXiv preprint arXiv:2405.20721*, 2024. 2, 3

[59] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 3, 6

[60] 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 *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 1, 2, 3, 4, 5, 6, 7, 8

[61] Minye Wu and Tinne Tuytelaars. Implicit gaussian splatting with efficient multi-level tri-plane representation. *arXiv preprint arXiv:2408.10041*, 2024. 2, 3

[62] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023. 2, 3

[63] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20331–20341, 2024. 1, 2, 3, 7, 8, 5, 6

[64] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5336–5345, 2020. 6, 7, 8, 2

[65] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 1, 6, 3

[66] Xinjie Zhang, Zhening Liu, Yifan Zhang, Xingtong Ge, Dailan He, Tongda Xu, Yan Wang, Zehong Lin, Shuicheng Yan, and Jun Zhang. Mega: Memory-efficient 4d gaussian splatting for dynamic scenes. *arXiv preprint arXiv:2410.13613*, 2024. 3

[67] Sveta Zinger, Luat Do, and Phn De With. Free-viewpoint depth image based rendering. *Journal of Visual Communication and Image Representation*, 21(5–6):533–541, 2010. 2