TRiGS: Temporal Rigid-Body Motion for Scalable 4D Gaussian Splatting

Suwoong Yeom[1*], Joonsik Nam[1*], Seunggyu Choi[1*], Lucas Yunkyu Lee[3], Sangmin Kim[4], Jaesik Park[4], Joonsoo Kim[5], Kugjin Yun[5], Kyeongbo Kong[2†], and Suk-Ju Kang[1†]

[1] Sogang University
[2] Pusan National University
[3] POSTECH
[4] Seoul National University
[5] Electronics and Telecommunications Research Institute
[*] Indicates Equal Contribution [†] Corresponding Authors

Project Page: https://wwwjjn.github.io/TRiGS-project_page/

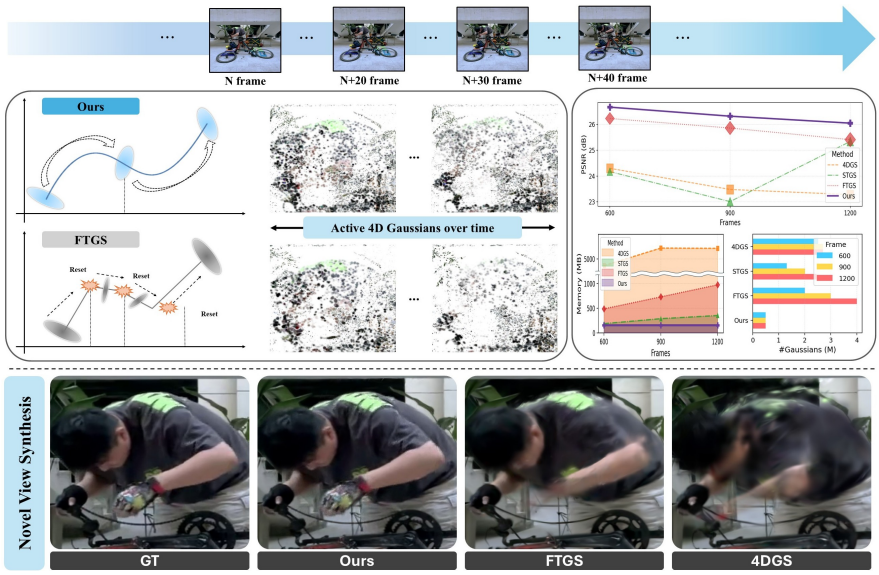


Fig. 1: Comparison of temporal modeling for extended dynamic scenes. (Left) While previous methods like FTGS rely on fragmented piecewise linear approximations that force primitives to repeatedly reset, our continuous rigid-body transformations preserve the long-term temporal identity of active Gaussians. **(Right)** By eliminating the need for unnecessary proliferation, TRiGS maintains a strictly constant, compact memory footprint while sustaining high rendering quality across extended sequences (up to 1200 frames), in stark contrast to baselines that suffer from severe memory bloat and performance drops. **(Bottom)** Consequently, our approach delivers temporally stable, high-fidelity novel view synthesis without visual degradation or motion artifacts.

Abstract. Recent 4D Gaussian Splatting (4DGS) methods achieve impressive dynamic scene reconstruction but often rely on piecewise linear velocity approximations and short temporal windows. This disjointed modeling leads to severe temporal fragmentation, forcing primitives to be repeatedly eliminated and regenerated to track complex nonlinear dynamics. This makeshift approximation eliminates the long-term temporal identity of objects and causes an inevitable proliferation of Gaussians, hindering scalability to extended video sequences. To address this, we propose TRiGS, a novel 4D representation that utilizes unified, continuous geometric transformations. By integrating $SE(3)$ transformations, hierarchical Bézier residuals, and learnable local anchors, TRiGS models geometrically consistent rigid motions for individual primitives. This continuous formulation preserves temporal identity and effectively mitigates unbounded memory growth. Extensive experiments demonstrate that TRiGS achieves high fidelity rendering on standard benchmarks while uniquely scaling to extended video sequences (e.g., 600 to 1200 frames) without severe memory bottlenecks, significantly outperforming prior works in temporal stability.

Keywords: 4D Gaussian Splatting · Continuous Motion Modeling · Scalable Representation · Novel View Synthesis

1 Introduction

Reconstructing dynamic scenes from video is a fundamental problem in computer vision, enabling immersive applications such as virtual reality, augmented reality, and free-viewpoint rendering. Recently, 3D Gaussian Splatting (3DGS) [10, 11, 15, 39] has advanced static scene rendering by achieving real-time performance while maintaining high visual fidelity. Building upon this success, recent works [2, 12, 20, 32, 33, 36, 37] have extended 3DGS into the temporal domain to model complex 4D dynamic environments, leading to rapid progress in dynamic scene representation and rendering.

One representative approach for 4D dynamic scene reconstruction learns implicit deformation fields [2,4,12,14,27,32,33,35,37] that describe each frame with respect to a canonical space. However, these methods struggle to maintain stable deformations as the temporal distance increases and displacements become large, a common issue in scenes with complex motions. To handle extended sequences, some approaches [18, 33] divide the video into sequential streaming chunks; yet, this strategy incurs severe memory overhead and often leads to visual quality degradation. To overcome the limitations in modeling complex dynamics, recent works [17, 20, 31, 36] have shifted toward directly optimizing 4D primitives or parameterizing motion with explicit functions. These explicit approaches currently achieve state-of-the-art performance by granting Gaussians extreme temporal flexibility. For example, they allow primitives to freely appear and disappear over time, or they approximate local motion using piecewise linear velocity and short temporal opacity windows.

Despite achieving high rendering quality on short video clips, these flexible formulations suffer from a fundamental limitation known as temporal fragmentation, as illustrated in Fig. 1. Since complex non-linear motion is constrained to linear approximations or short temporal windows, Gaussian primitives cannot accurately track continuous dynamic changes. Because a linear assumption inherently diverges from a true curved trajectory over time, the primitives gradually drift away from the actual geometry. To compensate for this growing spatial mismatch, the model is driven to repeatedly eliminate, split, and regenerate primitives. As visually evident in the left panel of Fig. 1, this disjointed modeling leads to an inevitable proliferation of Gaussians. This makeshift not only degrades the object's long-term temporal identity but also incurs substantial memory overhead. Consequently, such fragmented representations are difficult to scale to extended video sequences beyond hundreds of frames under practical memory constraints.

To address these limitations, we propose TRiGS, a new dynamic rendering framework that integrates continuous geometric transformations into 4D Gaussian Splatting to model temporally robust rigid motions. TRiGS moves beyond fragmented formulations that approximate continuous nonlinear motion using discontinuous linear segments or impose short lifetimes on primitives. Instead, it directly optimizes the motion so that the Gaussians composing an object undergo a geometrically consistent, continuous rigid transformation over time.

Concretely, we achieve this continuous representation through three key components. First, we employ the exponential map of $SE(3)$ to unify complex rotation and translation into a single continuous rigid transformation. Second, to accurately capture nonlinear dynamics over time, we formulate hierarchical motion decomposition where continuous time varying offsets, parameterized by a quadratic Bézier curve, smoothly refine the base transformation parameters. Finally, to faithfully capture independent local motions rather than relying on a shared global center of rotation, we introduce learnable local anchors. Each Gaussian is associated with its own anchor that acts as an independent center of rotation, enabling the stable optimization of geometrically consistent and fine grained local rigid motions.

As a result, unlike piecewise linear modeling which easily loses track of complex dynamics, our continuous motion and anchor modeling consistently preserves the long-term temporal identity of Gaussians. It also reduces the need to compensate for motion approximation errors by repeatedly splitting and rearranging primitives, thereby suppressing unnecessary Gaussian proliferation and excessive re-initialization. Overall, TRiGS prevents temporal fragmentation while lowering memory usage, making dynamic scene reconstruction feasible for long sequences beyond hundreds of frames without severe memory bottlenecks.

In summary, our main contributions are:

- We propose TRiGS, a novel 4D representation that utilizes unified, continuous geometric transformations rather than fragmented, piecewise linear velocity approximations. By integrating $SE(3)$ representations with learnable

local anchors, it models geometrically consistent rigid motions, effectively overcoming the severe temporal fragmentation seen in existing methods.
– We introduce a robust dynamic rendering pipeline designed to preserve the long-term temporal identity of primitives. Rather than repeatedly eliminating and regenerating Gaussians to handle complex nonlinear dynamics, we capture these continuous changes using time varying offsets. Coupled with a motion guided relocation strategy, it actively recycles redundant Gaussians, effectively mitigating unbounded memory growth.
– We demonstrate through extensive experiments that TRiGS achieves high fidelity rendering on standard dynamic benchmarks while uniquely scaling to extended video sequences (e.g., 600 to 1200 frames). By maintaining a compact representation without severe memory bottlenecks, it significantly outperforms prior works in temporal stability and scalability.

2 Related Work

2.1 Implicit Deformation for Dynamic 3D Gaussians

Following 3D Gaussian Splatting [15], recent efforts have extended it to dynamic scenes. Representative approaches [4, 16, 24, 32, 37, 38, 40] involve defining a canonical space and implicitly predicting the deformations of Gaussians over time using Neural Fields. Subsequent studies have improved performance by dividing the scene into multiple local spaces [7, 18, 33] or guiding the deformation based on sparse control points [12, 21, 23]. However, these implicit deformation field based strategies often struggle to maintain structural rigidity over time. Consequently, they demand careful tuning of initialization and regularizers to prevent drift. In long and complex scenes, errors accumulate, causing the model to lose geometric consistency and redundantly proliferate Gaussians to mask tearing and distortion artifacts.

2.2 4D Attributes and Explicit Motion Modeling

Parallel to these implicit approaches, 4D attributes [8, 9, 20, 36] were introduced to ensure spatiotemporal consistency by adding a temporal dimension to Gaussians. Subsequent studies utilized explicit functions [13, 17, 25, 26, 30] to optimize spatiotemporal displacements. Recently, methods [14, 31] with compact velocity equations have been proposed, achieving high performance. Despite these advancements, existing explicit modeling approaches face a shared challenge: they optimize transformation parameters independently and struggle to correctly track the natural, curved motions of actual rigid bodies. Ultimately, these models patch the gaps by excessively duplicating Gaussians to cover rendering errors.

In this work, we argue that structurally coupled motion modeling is a principled approach for the 4DGS framework. Thus, departing from independent parameter optimization, we introduce the geometric coupling of parameters and a

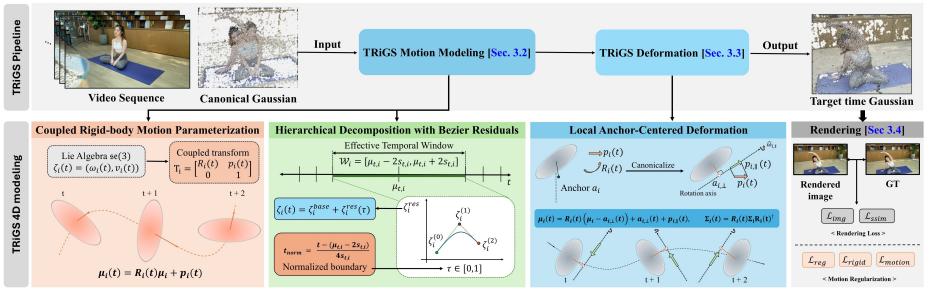


Fig. 2: Overview of our framework. We comprehensively model the scene dynamics through three key components. First, the motion is parameterized via a hierarchical decomposition using Bézier residuals within an effective temporal window. Second, this Lie algebra representation is mapped to a coupled $SE(3)$ transformation. Third, the transformation is applied relative to a gauge-fixed local anchor ($a_{i,\perp}$) to ensure stable, articulate deformation. Finally, the deformed primitives are rendered to optimize both photometric and motion regularization objectives.

representation based on the Gaussian's inherent local coordinate system. By fundamentally preventing unnatural shape distortions and performing precise motion estimation, the proposed model effectively represents long dynamic scenes with only a small number of Gaussians.

3 Method

We consider a calibrated video capturing a dynamic 3D scene. Our goal is to reconstruct a time-varying scene representation that supports novel-view rendering. To this end, we propose TRiGS, which comprehensively models complex scene dynamics through three key components: Coupled $SE(3)$ transformation for robust rigid-body motion modeling, Hierarchical Decomposition with Bézier Residuals, and a Local Anchor-Centered Deformation to capture local motion.

3.1 Preliminaries

We represent the scene as a set of 3D Gaussian primitives $\{g_i\}$. Each primitive g_i has a canonical mean $\mu_i \in \mathbb{R}^3$, covariance $\Sigma_i \in \mathbb{R}^{3\times3}$, and canonical opacity α_i. Time t is treated as a continuous scalar, and each primitive is associated with a central time $\mu_{t,i}$.

Temporal Opacity Temporal opacity is used to restrict the visibility of each primitive to a local neighborhood in time. We introduce a temporal visibility function $\gamma_i : \mathbb{R} \to (0, 1]$, parameterized by a temporal scale $s_{t,i} > 0$, to define the time-dependent opacity as:

$$\gamma_i(t) \;=\; \exp\Big(-\frac{(t-\mu_{t,i})^2}{2s_{t,i}^2}\Big), \qquad \alpha_i(t) \;=\; \alpha_i \cdot \gamma_i(t). \tag{1}$$

At query time t, the rendering process utilizes the time-dependent parameters $\mu_i(t), \Sigma_i(t), \alpha_i(t)$, where $\mu_i(t)$ and $\Sigma_i(t)$ are governed by the motion model.

Linear Motion Modeling Like FreeTimeGS (FTGS) [31], a simple baseline models the primitive center using translation-only linear motion:

$$\mu_i(t) \;=\; \mu_i + v_i\,(t - \mu_{t,i}), \tag{2}$$

where $v_i \in \mathbb{R}^3$ is a per-primitive velocity. To represent complex motions, a translation-only model often needs to decompose the trajectory with piecewise-linear representation, which typically requires many primitives as the motion becomes more complex. Under this formulation, the more complex the motion becomes, the more primitives are needed to maintain fidelity, inevitably causing severe temporal fragmentation and a significant surge in memory usage.

3.2 Motion Modeling

Coupled Rigid-body Motion Parameterization To avoid allocating many time-localized primitives for complex motion, we model the motion of each Gaussian primitive with an $SE(3)$ formulation:

$$\mu_i(t) \;=\; R_i(t)\,\mu_i \;+\; p_i(t), \tag{3}$$

where $R_i(t) \in SO(3)$ and $p_i(t) \in \mathbb{R}^3$. However, optimizing $R_i(t)$ and $p_i(t)$ as independent parameters is often ill-conditioned, because rotation and translation can partially compensate each other under the rendering objective. This creates a continuum of near-equivalent solutions, yielding an entangled motion decomposition and accumulated drift over time. We therefore parameterize motion directly in the Lie algebra $\mathfrak{se}(3)$ and map it to $SE(3)$ via the exponential map, which enforces a coupled rigid transform and helps stabilize optimization without introducing additional disentanglement-specific losses. Concretely, we optimize a time-conditioned coefficient $\zeta_i(t) = (\omega_i(t), \nu_i(t)) \in \mathfrak{se}(3)$ and define the relative log-parameter $\mathbf{u}_i(t) := \zeta_i(t)\Delta t$ with $\Delta t := t - \mu_{t,i}$. The relative transform from the central time $\mu_{t,i}$ to t is then

$$\mathbf{T}_i(\mu_{t,i} \to t) = \exp\big(\hat{\mathbf{u}}_i(t)\big) = \begin{bmatrix} R_i(t) & p_i(t) \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{u}_i(\mu_{t,i}) = \mathbf{0}. \tag{4}$$

where $\hat{\mathbf{u}}_i(t)$ denotes the standard wedge operator in $\mathfrak{se}(3)$. This construction satisfies $\mathbf{T}_i(\mu_{t,i}) = \mathbf{I}$ by design. Moreover, $R_i(t)$ and $p_i(t)$ are jointly determined by the same closed-form exponential map. Let $\mathbf{u}_i(t) = (\phi_i(t), \upsilon_i(t))$ with $\phi_i(t) = \omega_i(t)\Delta t$ and $\upsilon_i(t) = \nu_i(t)\Delta t$. We compute $R_i(t)$ via Rodrigues' formula and obtain $p_i(t) = J(\phi_i(t))\,\upsilon_i(t)$ using the $SO(3)$ left Jacobian $J(\cdot)$, rather than optimizing translation independently.

Hierarchical Decomposition with Bézier Residuals To capture complex, non-linear motion over time, we model the time dependence of the Lie algebra coefficient $\zeta_i(t)$ using a hierarchical decomposition into a base term and a Bézier residual. Specifically, we decompose it into learnable base terms $\zeta_i^{\text{base}} = (\omega_i^{\text{base}}, \nu_i^{\text{base}})$ and time-varying residuals $\zeta_i^{\text{res}} = (\omega_i^{\text{res}}(\tau), \nu_i^{\text{res}}(\tau))$ represented by a quadratic Bézier curve. To concentrate modeling capacity where primitive i is effectively supervised, we localize the parameterization to an effective temporal window implied by temporal visibility $\gamma_i(t)$. Since $\gamma_i(t)$ rapidly decays away from $\mu_{t,i}$, primitive i receives almost no gradients at distant times. Restricting the domain stabilizes optimization and avoids drift in weakly supervised regions. We define the effective window as

$$\mathcal{W}_i \;=\; [\,\mu_{t,i} - 2s_{t,i},\;\; \mu_{t,i} + 2s_{t,i}\,]. \tag{5}$$

To evaluate the Bézier residual within $\mathcal{W}_i$, we map the continuous global time t to a local normalized coordinate $\tau \in [0, 1]$. We compute the relative time progress t_{norm} and clamp it to ensure the temporal offset remains bounded outside this active temporal extent:

$$t_{\text{norm}} \;=\; \frac{t - (\mu_{t,i} - 2s_{t,i})}{4s_{t,i}}, \qquad \tau \;=\; \begin{cases} 0, & \text{if } t_{\text{norm}} < 0 \\ t_{\text{norm}}, & \text{if } 0 \le t_{\text{norm}} \le 1 \\ 1, & \text{if } t_{\text{norm}} > 1 \end{cases} \tag{6}$$

We model the residual $\zeta_i^{\text{res}}(\tau)$ as a quadratic Bézier curve with learnable coefficients, providing a smooth fine-grained refinement on top of the base motion. The final motion coefficient is obtained by adding the residual to the base:

$$\zeta_i^{\text{res}}(\tau) = (1-\tau)^2\zeta_i^{(0)} + 2(1-\tau)\tau\zeta_i^{(1)} + \tau^2\zeta_i^{(2)}, \tag{7}$$

$$\zeta_i(t) = \zeta_i^{\text{base}} + \zeta_i^{\text{res}}(\tau). \tag{8}$$

where $\zeta_i^{\text{base}}, \zeta_i^{(k)} \in \mathfrak{se}(3)$ for $k \in \{0, 1, 2\}$ are learnable parameters.

3.3 Local Anchor-Centered Deformation

Applying an $SE(3)$ transform in Eq. 3 with a single shared anchor implicitly assumes that all primitives rotate around the same center. Dynamic scenes contain multiple parts undergoing independent local motions with distinct centers of rotation. To faithfully capture these independent motions, we introduce a learnable per-primitive local anchor $a_i \in \mathbb{R}^3$ and parameterize the $SE(3)$ transform around it.

However, a_i is not uniquely identifiable from translation. The same deformation can be explained by multiple $(a_i, p_i(t))$ pairs leading to optimization ambiguity. To alleviate this ambiguity, we canonicalize the a_i as $a_{i,\perp}(t)$ by projecting a_i onto the plane orthogonal to the rotation axis, and use only the axis-aligned translational component $v_{i,\parallel}(t)$:

$$a_{i,\perp}(t) = a_i - \langle a_i, \bar{\omega}_i(t)\rangle\bar{\omega}_i(t), \qquad v_{i,\parallel}(t) = \langle v_i(t), \bar{\omega}_i(t)\rangle\,\bar{\omega}_i(t), \tag{9}$$

where $\bar{\omega}_i(t)$ denotes the unit rotation axis direction. The anchor projection $a_{i,\perp}(t)$ acts as a gauge fixing, since $(I - R_i(t))\bar{\omega}_i(t) = 0$ implies that the axis-parallel component of a_i lies in the null space of $(I - R_i(t))$ and is thus unidentifiable. Consequently, we keep only the axis-aligned translation induced by the exponential map and denote it as $p_{i,\parallel}(t) = J(\phi_i(t))\, v_{i,\parallel}(t)$. With this local anchor parameterization, the deformed Gaussian primitives at time t are given by

$$\mu_i(t) = R_i(t)\,(\mu_i - a_{i,\perp}(t)) + a_{i,\perp}(t) + p_{i,\parallel}(t), \qquad \Sigma_i(t) = R_i(t)\,\Sigma_i\,R_i(t)^\top. \tag{10}$$

Please see the supplementary material for the closed-form $SE(3)$ exponential map, the gauge-fixing derivation, and small-angle numerical stabilization.

3.4 Training

Similar to 3DGS, we optimize Gaussian primitive parameters by minimizing a rendering objective between the rendered images and the ground-truth images:

$$\mathcal{L} = \lambda_{img}\,\mathcal{L}_{img} + \lambda_{ssim}\,\mathcal{L}_{ssim} + \lambda_{reg}\,\mathcal{L}_{reg} + \lambda_{motion}\,\mathcal{L}_{motion} + \lambda_{rigid}\,\mathcal{L}_{rigid}. \tag{11}$$

Motion regularization Although our primary supervision is the photometric rendering loss, the optimization remains ill-posed due to the large number of primitive parameters and the ambiguity of explaining image residuals, especially in dynamic scenes. Therefore, we regularize primitive parameters with $\mathcal{L}_{reg}$ following FTGS, enforce temporal smoothness with $\mathcal{L}_{motion}$, and encourage local motion coherence with $\mathcal{L}_{rigid}$. We denote the stop-gradient operator by $sg[\cdot]$.

$$b_i = \zeta_i^{(0)} - 2\zeta_i^{(1)} + \zeta_i^{(2)}, \qquad \mathcal{L}_{motion} = \|b_i\|_2^2. \tag{12}$$

$\mathcal{L}_{motion}$ regularizes the quadratic Bézier residual by penalizing its acceleration term in both translation and rotation components. This discourages abrupt curvature in $\zeta_i(t)$ and promotes temporally smooth trajectories within the effective window.

$$K_{ij} = \exp\Big(-\lambda_c\,\|\mathbf{c}_i - \mathbf{c}_j\|_2^2\Big), \tag{13}$$

$$\mathcal{L}_{rigid} = \sum_i \sum_{j\in\mathcal{N}(i)} K_{ij}\left(\left\|\nu_i^{\text{base}} - \nu_j^{\text{base}}\right\|_2^2 + \left\|\omega_i^{\text{base}} - \omega_j^{\text{base}}\right\|_2^2\right), \tag{14}$$

where $\mathbf{c}_i$ denotes the DC spherical-harmonics color of primitive i in canonical space. $\mathcal{L}_{rigid}$ promotes locally coherent motion by aligning the base motions of spatial neighbors. The appearance-based weight K_{ij} serves as a soft affinity that suppresses incorrect coupling across object boundaries. Concretely, we compute the penalty over k-nearest-neighbor Gaussian primitives $j \in \mathcal{N}(i)$ based on their canonical centers μ_i in 3D. This encourages smooth motion within each region while preserving motion discontinuities at boundaries, reducing local drift and motion fragmentation.

Table 1: Quantitative comparison on the SelfCap dataset across 600, 900, and 1200 frame budgets. best, second-best, and third-best results are highlighted in the table. (* Results from our reproduction, as the official code is not publicly available.)

Method	600 frames			900 frames			1200 frames		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
GIFStream [18]	23.76	0.877	0.134	23.80	0.871	0.124	24.27	0.874	0.117
4DGS [36]	24.30	0.860	0.227	23.48	0.851	0.240	23.28	0.851	0.233
Ex-4DGS [17]	21.61	0.801	0.233	20.93	0.793	0.248	20.45	0.787	0.248
STGS [20]	24.17	0.844	0.161	23.01	0.864	0.136	25.34	0.734	0.100
LocalDyGS [33]	25.49	0.858	0.240	23.51	0.838	0.280	24.24	0.884	0.116
FTGS* [31]	26.23	0.913	0.121	25.86	0.898	0.132	25.41	0.871	0.146
Ours	**26.67**	**0.937**	**0.113**	**26.32**	**0.916**	**0.120**	**26.05**	**0.904**	**0.099**

Motion-guided Relocation To reuse capacity under a fixed primitive budget, we periodically recycle low-opacity primitives by cloning informative active ones. We split primitives using an opacity threshold τ_α:

$$\mathcal{I} = \{i \mid \sigma(\alpha_i) < \tau_\alpha\}, \qquad \mathcal{A} = \{i \mid \sigma(\alpha_i) \geq \tau_\alpha\}. \tag{15}$$

We then sample a source primitive $i \in \mathcal{A}$ with probability $q_i \propto s_i$, where

$$s_i = \sum_{u \in \{\alpha, 2D, t, d\nu\}} p_i^{(u)}, \qquad q_i = \frac{s_i}{\sum_{k \in \mathcal{A}} s_k}. \tag{16}$$

Here $\{p_i^{(u)}\}$ are normalized difficulty cues defined in the supplement. The sampled primitive is cloned to replace an element in $\mathcal{I}$, reallocating primitives to hard regions while keeping the total count unchanged.

4 Experiments

4.1 Implementation details

We implement our approach in PyTorch and optimize all parameters using Adam with the same settings as 3DGS [15]. The model is trained for 30k iterations. The weights for motion regularization are set to $\lambda_{reg} = 0.01$, $\lambda_{motion} = 0.0001$, $\lambda_{rigid} = 1.0$, $\lambda_c = 50$, and $K = 3$. We perform motion-guided relocation every $N = 100$ iterations using an opacity threshold $\tau_\alpha = 0.005$. To improve rendering quality, we use RoMa [5] to obtain a point cloud initialization. All experiments are conducted on a single RTX 4090 GPU.

4.2 Experimental settings

We evaluate TRiGS on the The Neural 3D Video(N3V) [19] and SelfCap [31] datasets against recent state-of-the-art dynamic synthesis methods [17, 18, 20, 31, 33, 36]. We assess rendering quality (PSNR, SSIM, LPIPS) and efficiency (FPS, Gaussian count, memory usage). For a fair comparison, all methods are initialized with the same RoMa-derived point cloud. Since the official code for FTGS is unavailable, we report results from our reproduction (∗), which we validated by achieving comparable or superior performance to the original paper on the N3V dataset (Table 2).

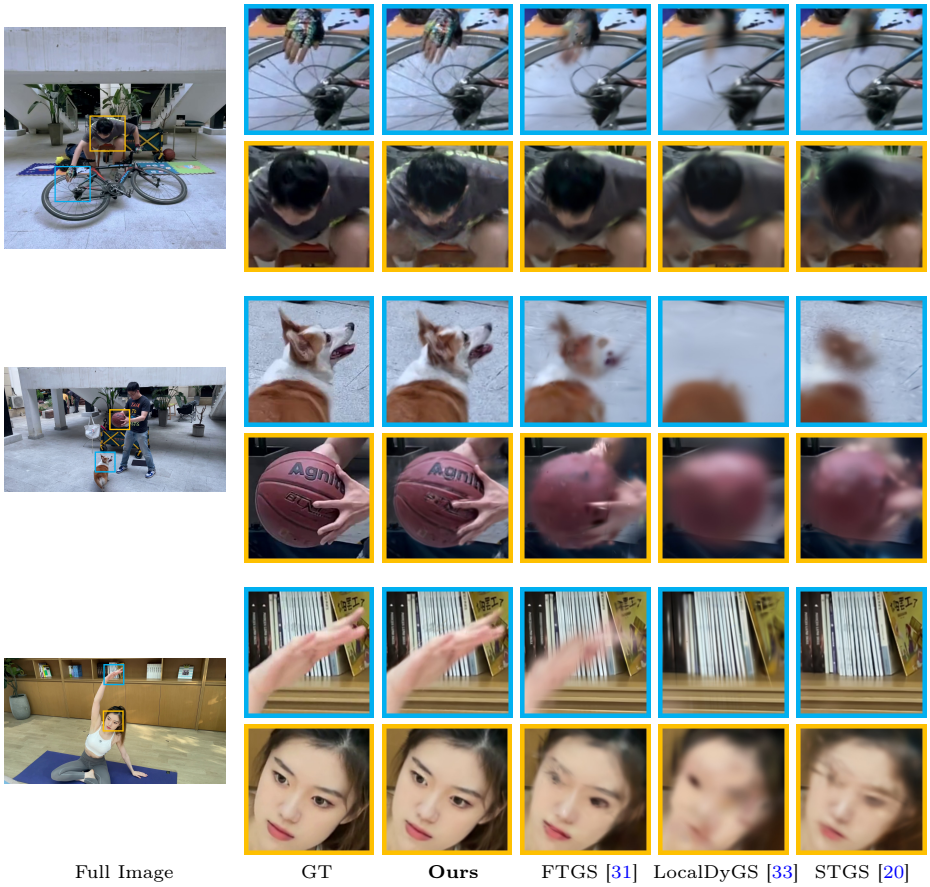


Fig. 3: Qualitative comparison on the SelfCap 1200-frame scenes.

SelfCap The SelfCap dataset provides challenging dynamic sequences characterized by fast and complex motions. It is captured using a multi-view system comprising 22 to 24 cameras at 60 FPS. For our evaluation, we utilize six distinct scenes: *bike1*, *bike2*, *corgi1*, *corgi2*, *dance*, and *yoga*. To rigorously assess long-term temporal stability and scalability, we edit the original videos to construct three variations of extended datasets consisting of 600, 900, and 1200 frames. The standard resolution is 3840×2160, except for the bike scenes, which are provided at 1080×1080.

Neural3DV The Neural 3D Video (N3V) dataset is a widely adopted standard benchmark for dynamic scene reconstruction. Following established protocols, we downsample the original 2704×2028 videos (recorded at 30 FPS) and apply the official camera split for training and testing.

4.3 Comparisons

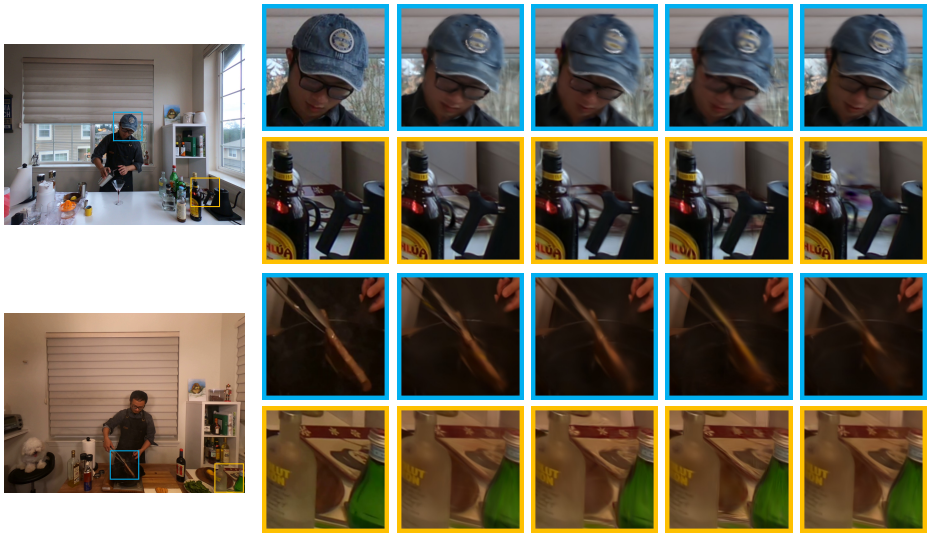


Fig. 4: Qualitative results on the N3V dataset.

Quantitative comparisons We first evaluate our method on the extended sequences of the SelfCap dataset to demonstrate its robustness over long-term frame budgets. As shown in Table 1, TRiGS achieves the highest rendering quality across 600, 900, and 1200 frames. Notably, while explicit 4DGS baselines like FTGS suffer from significant quality degradation (e.g., PSNR dropping from 26.23 to 25.41) as the sequence lengthens, TRiGS maintains high fidelity (PSNR of 26.05 at 1200 frames) without severe performance drops.

This stable performance is further highlighted in our efficiency analysis (Table 3). When processing extended videos, existing explicit 4D primitive models consistently suffer from severe memory bottlenecks. For instance, FTGS doubles its primitive count and memory footprint (up to 977 MB) when scaling from 600 to 1200 frames. Furthermore, while streaming-based models like GIFStream are designed to handle extended frames, they incur a massive memory overhead (up to 2825 MB) and still yield suboptimal rendering quality. In contrast, by effectively capturing continuous geometric transformations, TRiGS maintains a highly compact and constant representation of only 0.5M Gaussians and 160 MB of memory, maintaining over 110 FPS across all frame budgets.

Finally, to verify that our method is not only effective for long videos but also excels on standard-length sequences, we evaluate TRiGS on the N3V dataset (typically ∼300 frames). As reported in Table 2, TRiGS achieves state-of-the-art performance with the highest PSNR (33.36) and lowest LPIPS (0.031). This confirms that our continuous motion modeling generally surpasses existing approaches in capturing complex dynamic scenes, regardless of the video length. We attribute this to the increased per-primitive motion capacity, which enables

Table 2: Quantitative comparison on N3V datasets. best, second-best, and third-best results are highlighted in the table. († Results reported in the original paper, using 0.5M initial points.)

Method	PSNR↑	$DSSIM_1$ ↓	$DSSIM_2$ ↓	LPIPS↓
$HexPlane^2$ [3]	31.71	-	0.014	0.075
K-Planes [6]	31.63	-	0.018	-
MixVoxels [29]	31.73	-	0.015	0.064
HyperReel [1]	31.10	0.036	-	0.096
NeRFPlayer [28]	30.96	0.034	-	0.111
Deformable-3DGS [37]	31.15	0.030	-	0.049
C-D3DGS [14]	30.46	-	0.022	0.150
SWinGS [22]	31.10	0.030	-	0.096
Ex4DGS [17]	32.11	0.030	0.015	0.048
4DGS [36]	32.01	-	0.014	0.055
STGS [20]	32.05	0.026	0.014	0.044
GIFStream [18]	31.75	-	-	0.051
LocalDyGS [33]	32.28	0.028	0.014	0.043
DASH [4]	32.22	-	-	-
Swift4D [34]	32.23	-	0.014	0.043
FTGS† [31]	32.97	0.028	0.014	0.043
FTGS*	32.80	0.021	-	0.040
Ours	**33.36**	**0.019**	**0.010**	**0.031**

Table 3: Efficiency comparison on SelfCap across 600, 900, and 1200 frame budgets.

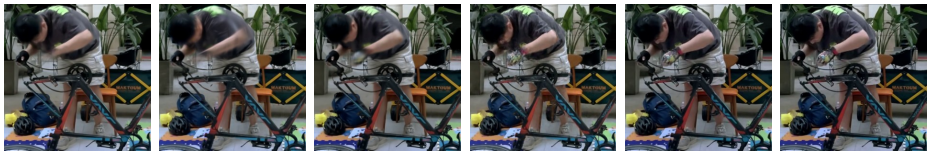
Method	600 frames			900 frames			1200 frames		
	FPS↑	#Gaussians↓	MB↓	FPS↑	#Gaussians↓	MB↓	FPS↑	#Gaussians↓	MB↓
GIFStream [18]	94	6.47M	1586	98	8.85M	2169	102	11.53M	2825
4DGS [36]	43	2.51M	4665	43	2.70M	5021	44	2.70M	5015
STGS [20]	90	1.31M	190	83	2.01M	291	90	2.45M	355
FTGS* [31]	113	2M	490	110	3M	733	106	4M	977
Ours	**116**	**0.5M**	**160**	**111**	**0.5M**	**160**	**120**	**0.5M**	**160**

each Gaussian to accurately capture local dynamics, successfully representing the entire sequence with drastically fewer primitives.

Qualitative comparisons Consistent with our quantitative results, visual comparisons in Fig. 3 and Fig. 4 confirm the superiority of TRiGS in rendering complex dynamic scenes, particularly over extended sequences. While baseline methods suffer from error accumulation over time—often resulting in severe ghosting, blurriness, and structural degradation in fast-moving regions (e.g., articulating limbs or rotating wheels)—TRiGS consistently preserves sharp details and accurate geometries. By leveraging continuous geometric transformations with local anchors, our approach effectively captures intricate local dynamics without the need for massive primitive duplication. This translates to highly realistic, temporally stable novel-view synthesis, which is further demonstrated in our supplementary video.

Table 4: Ablation study of core components in TRiGS on the 1200-frame extended sequence. We evaluate the impact of trajectory modeling, anchor modes, and base parameterization on rendering quality and memory efficiency.

Configuration	Modeling Choices			Quality Metrics			Efficiency	
	Trajectory	Anchor	Base Param.	PSNR↑	SSIM↑	LPIPS↓	#Gaussians↓	MB↓
Baseline	Linear	None	$\mathbb{R}^3$	24.83	0.865	0.162	2.10M	512
Naive SE(3)	Rigid Body	Origin	$SO(3) \times \mathbb{R}^3$	23.95	0.842	0.198	2.85M	694
Local SE(3)	Rigid Body	Local	$\mathfrak{se}(3)$	25.52	0.891	0.125	0.60M	185
Linear + Bézier	Non-linear	None	$\mathbb{R}^3$	25.10	0.873	0.145	1.85M	450
Full TRiGS	**Non-linear**	**Local**	$\mathfrak{se}(3)$	**26.05**	**0.904**	**0.099**	**0.50M**	**160**



Baseline NaiveSE3 Linear+Bèzier LocalSE3 Full Ground Truth

Fig. 5: Visual ablation on the SelfCap *bike2* scene. Compared to incomplete configurations that suffer from motion artifacts and blurriness, **Full TRiGS** successfully preserves sharp geometric details and structural integrity.

4.4 Analysis and Ablation Study

Ablation of Core Components To validate the necessity of each structural design in TRiGS, we conduct an ablation study on the challenging 1200-frame sequence, as summarized in Table 4.

The *Baseline* relies on a simple linear velocity model without geometric constraints. Because it fails to track continuous curved trajectories, it naturally suffers from spatial mismatches. A straightforward attempt to apply geometric constraints is *Naive SE(3)*, which models rigid transformations around a global origin with independent rotation and translation parameters ($SO(3) \times \mathbb{R}^3$). Interestingly, this configuration yields the lowest performance (23.95 PSNR) among all settings. This empirically validates our theoretical analysis in Sec. 3: Utilizing a single shared anchor misrepresents independent motions, and optimizing uncoupled parameters leads to entangled optimization and accumulated drift.

In contrast, introducing learnable local anchors and coupled Lie algebra parameterization (*Local SE(3)*) yields a quantum jump in performance. The PSNR significantly improves to 25.52, confirming that decoupling global motions into geometrically consistent local rigid transformations is crucial for stable optimization. Furthermore, we evaluate *Linear + Bézier*, which applies non-linear Bézier curves only to unconstrained translations. While it slightly improves upon the baseline, it demonstrates that non-linear trajectories alone cannot guarantee geometric consistency without rigid-body constraints. Ultimately, our **Full TRiGS** model, which synergistically combines local SE(3) transformations with hierarchical decomposition with Bézier Residual, achieves the highest visual fidelity (26.05 PSNR).

Table 5: Ablation study on optimization components in TRiGS. We evaluate the impact of gauge fixing, regularization losses, and relocation strategy on the 1200-frame sequence.

Configuration	PSNR↑	SSIM↑	LPIPS↓
w/o Gauge Fixing ($a_{i,\perp}$)	25.15	0.880	0.134
w/o Rigid Reg. ($\mathcal{L}_{rigid}$)	25.48	0.885	0.128
w/o Motion Smoothness ($\mathcal{L}_{motion}$)	25.66	0.892	0.115
w/o Motion-guided Relocation	25.80	0.897	0.108
Full TRiGS Architecture	**26.05**	**0.904**	**0.099**

Analysis of Representation Capacity It is important to clarify the significant differences in Gaussian counts across the configurations in Table 4. TRiGS intentionally operates under a strict, fixed memory budget (0.50M primitives) by utilizing a relocation strategy without standard densification. However, for the ablated baselines (e.g., *Baseline* and *Naive SE(3)*), we permitted standard gradient-based densification. This was a deliberate experimental design to evaluate the upper-bound rendering capability of their motion models; restricting these baseline configurations to 0.50M primitives would result in catastrophic rendering collapse due to their inability to track complex trajectories.

Interestingly, even when allowed to aggressively proliferate (up to 2.85M primitives) to compensate for spatial mismatches, these configurations still fail to reach the visual fidelity of TRiGS. Because they lack accurate geometric constraints, they attempt to patch trajectory errors by excessively splitting primitives, leading to severe memory bloat without resolving the underlying structural misalignment. This confirms that our highly expressive motion formulation effectively resolves complex dynamics under a minimal memory budget, demonstrating the advantage of structured motion parameterization over brute-force capacity scaling.

Ablation of Optimization Components Beyond the core architecture, we ablate our specific optimization and regularization strategies in Table 5. Removing the gauge fixing step (*w/o Gauge Fixing*) causes the largest quality drop (25.15 PSNR). Without it, the translational component suffers from unidentifiability, which severely destabilizes the optimization process.

Motion regularizers are also crucial for rendering fidelity. Removing rigid regularization (*w/o Rigid Reg.*) disrupts the spatial alignment of neighboring motions, causing subtle surface tearing and a drop in SSIM (0.885). Omitting the motion smoothness loss (*w/o Motion Smoothness*) leads to jittery and temporally inconsistent trajectories. Finally, disabling opacity-based recycling (*w/o Motion-guided Relocation*) prevents the active reallocation of redundant Gaussians to complex dynamic regions, lowering the overall representational capacity. Ultimately, integrating these strategies allows the full TRiGS architecture to achieve the highest visual quality (26.05 PSNR, 0.099 LPIPS).

5 Conclusion

We presented TRiGS, a new dynamic rendering framework that increases per-primitive temporal modeling capacity for novel-view synthesis. TRiGS mitigates the temporal fragmentation and memory growth from translation-only motion or short temporal opacity via three designs: (i) Coupled rigid-body parameterization with a closed-form exponential map, (ii) Hierarchical Bézier residual within a visibility-driven temporal window, and (iii) Local anchor-centered deformation to capture faithful local motions. Moreover, our motion-aware training improves long-sequence stability by reducing drift while maintaining effective primitive coverage under a fixed budget. Experiments show that TRiGS achieves higher rendering quality with fewer Gaussians and lower memory overhead, and remains stable on longer videos where prior methods lose temporal identity and exhibit failure modes.

Acknowledgements

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2022-0-00022, RS-2022-II220022, Development of immersive video spatial computing technology for ultra-realistic metaverse services) (30%); the Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2026 (Project Name: Multimodal Storyverse: A Multi-Agent Collaborative Platform for Expansive Narrative Creation, Project Number: RS-2026-25527029, Contribution Rate: 30%); the Artificial Intelligence Innovation Graduate School (Sogang University) grant funded by the Korea government (MSIT) (RS-2026-25547954) (20%); the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00414230) (15%); and the IITP grant funded by MSIT (RS-2021-II211343, AI Graduate School Program (SNU)) (5%).

References

1. Attal, B., Huang, J.B., Richardt, C., Zollhoefer, M., Kopf, J., O'Toole, M., Kim, C.: Hyperreel: High-fidelity 6-dof video with ray-conditioned sampling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16610–16620 (2023)
2. Bae, J., Kim, S., Yun, Y., Lee, H., Bang, G., Uh, Y.: Per-gaussian embedding-based deformation for deformable 3d gaussian splatting. In: European Conference on Computer Vision. pp. 321–335. Springer (2024)
3. Cao, A., Johnson, J.: Hexplane: A fast representation for dynamic scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 130–141 (2023)

4. Chen, J., Hu, Z., Wu, P., Zhu, H., Li, H., Sun, X.: Dash: 4d hash encoding with self-supervised decomposition for real-time dynamic scene rendering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 26349–26359 (2025)
5. Edstedt, J., Sun, Q., Bökman, G., Wadenbäck, M., Felsberg, M.: Roma: Robust dense feature matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19790–19800 (2024)
6. Fridovich-Keil, S., Meanti, G., Warburg, F.R., Recht, B., Kanazawa, A.: K-planes: Explicit radiance fields in space, time, and appearance. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12479–12488 (2023)
7. Fu, J., Gao, Q., Wen, C., Wu, Y., Ma, S., Zhang, J., Zhang, J.: Recon-gs: Continuum-preserved gaussian streaming for fast and compact reconstruction of dynamic scenes. arXiv preprint arXiv:2509.24325 (2025)
8. Gao, Z., Planche, B., Zheng, M., Choudhuri, A., Chen, T., Wu, Z.: 6dgs: Enhanced direction-aware gaussian splatting for volumetric rendering. arXiv preprint arXiv:2410.04974 (2024)
9. Gao, Z., Planche, B., Zheng, M., Choudhuri, A., Chen, T., Wu, Z.: 7dgs: Unified spatial-temporal-angular gaussian splatting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 26316–26325 (2025)
10. Guédon, A., Lepetit, V.: Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5354–5363 (2024)
11. Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2d gaussian splatting for geometrically accurate radiance fields. In: ACM SIGGRAPH 2024 conference papers. pp. 1–11 (2024)
12. Huang, Y.H., Sun, Y.T., Yang, Z., Lyu, X., Cao, Y.P., Qi, X.: Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4220–4230 (2024)
13. Jin, I.H., Mun, H., Kim, J., Yun, K., Kong, K.: Moe-gs: Mixture of experts for dynamic gaussian splatting. arXiv preprint arXiv:2510.19210 (2025)
14. Katsumata, K., Vo, D.M., Nakayama, H.: A compact dynamic 3d gaussian representation for real-time dynamic view synthesis. In: European Conference on Computer Vision. pp. 394–412. Springer (2024)
15. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., et al.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
16. Kratimenos, A., Lei, J., Daniilidis, K.: Dynmf: Neural motion factorization for real-time dynamic view synthesis with 3d gaussian splatting. In: European Conference on Computer Vision. pp. 252–269. Springer (2024)
17. Lee, J., Won, C., Jung, H., Bae, I., Jeon, H.G.: Fully explicit dynamic gaussian splatting. Advances in Neural Information Processing Systems **37**, 5384–5409 (2024)
18. Li, H., Li, S., Gao, X., Batuer, A., Yu, L., Liao, Y.: Gifstream: 4d gaussian-based immersive video with feature stream. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 21761–21770 (2025)
19. Li, T., Slavcheva, M., Zollhoefer, M., Green, S., Lassner, C., Kim, C., Schmidt, T., Lovegrove, S., Goesele, M., Newcombe, R., et al.: Neural 3d video synthesis from multi-view video. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5521–5531 (2022)

20. Li, Z., Chen, Z., Li, Z., Xu, Y.: Spacetime gaussian feature splatting for real-time dynamic view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8508–8520 (2024)
21. Li, Z., Chen, Y., Liu, P.: Dreammesh4d: Video-to-4d generation with sparse-controlled gaussian-mesh hybrid representation. Advances in Neural Information Processing Systems **37**, 21377–21400 (2024)
22. Liu, B., Banerjee, S.: Swings: Sliding window gaussian splatting for volumetric video streaming with arbitrary length. arXiv preprint arXiv:2409.07759 (2024)
23. Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., Dai, B.: Scaffold-gs: Structured 3d gaussians for view-adaptive rendering. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20654–20664 (2024)
24. Lu, Z., Guo, X., Hui, L., Chen, T., Yang, M., Tang, X., Zhu, F., Dai, Y.: 3d geometry-aware deformable gaussian splatting for dynamic view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8900–8910 (2024)
25. Luiten, J., Kopanas, G., Leibe, B., Ramanan, D.: Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In: 2024 International Conference on 3D Vision (3DV). pp. 800–809. IEEE (2024)
26. Park, J., Bui, M.Q.V., Bello, J.L.G., Moon, J., Oh, J., Kim, M.: Splinegs: Robust motion-adaptive spline for real-time dynamic 3d gaussians from monocular video. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26866–26875 (2025)
27. Shaw, R., Nazarczuk, M., Song, J., Moreau, A., Catley-Chandar, S., Dhamo, H., Pérez-Pellitero, E.: Swings: sliding windows for dynamic 3d gaussian splatting. In: European Conference on Computer Vision. pp. 37–54. Springer (2024)
28. Song, L., Chen, A., Li, Z., Chen, Z., Chen, L., Yuan, J., Xu, Y., Geiger, A.: Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. IEEE Transactions on Visualization and Computer Graphics **29**(5), 2732–2742 (2023)
29. Wang, F., Tan, S., Li, X., Tian, Z., Song, Y., Liu, H.: Mixed neural voxels for fast multi-view video synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19706–19716 (2023)
30. Wang, Q., Ye, V., Gao, H., Zeng, W., Austin, J., Li, Z., Kanazawa, A.: Shape of motion: 4d reconstruction from a single video. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9660–9672 (2025)
31. Wang, Y., Yang, P., Xu, Z., Sun, J., Zhang, Z., Chen, Y., Bao, H., Peng, S., Zhou, X.: Freetimegs: Free gaussian primitives at anytime anywhere for dynamic scene reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21750–21760 (2025)
32. Wu, G., Yi, T., Fang, J., Xie, L., Zhang, X., Wei, W., Liu, W., Tian, Q., Wang, X.: 4d gaussian splatting for real-time dynamic scene rendering. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20310–20320 (2024)
33. Wu, J., Peng, R., Jiao, J., Yang, J., Tang, L., Xiong, K., Liang, J., Yan, J., Liu, R., Wang, R.: Localdygs: Multi-view global dynamic scene modeling via adaptive local implicit feature decoupling. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9519–9529 (2025)
34. Wu, J., Peng, R., Wang, Z., Xiao, L., Tang, L., Yan, J., Xiong, K., Wang, R.: Swift4d: Adaptive divide-and-conquer gaussian splatting for compact and efficient reconstruction of dynamic scene. arXiv preprint arXiv:2503.12307 (2025)

35. Yan, Y., Lin, H., Zhou, C., Wang, W., Sun, H., Zhan, K., Lang, X., Zhou, X., Peng, S.: Street gaussians: Modeling dynamic urban scenes with gaussian splatting. In: European Conference on Computer Vision. pp. 156–173. Springer (2024)
36. Yang, Z., Yang, H., Pan, Z., Zhang, L.: Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. arXiv preprint arXiv:2310.10642 (2023)
37. Yang, Z., Gao, X., Zhou, W., Jiao, S., Zhang, Y., Jin, X.: Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20331–20341 (2024)
38. Yoon, J., Han, S., Oh, J., Lee, M.: Splinegs: Learning smooth trajectories in gaussian splatting for dynamic scene reconstruction. In: The Thirteenth International Conference on Learning Representations (2025)
39. Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A.: Mip-splatting: Alias-free 3d gaussian splatting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19447–19456 (2024)
40. Zhu, R., Liang, Y., Chang, H., Deng, J., Lu, J., Yang, W., Zhang, T., Zhang, Y.: Motiongs: Exploring explicit motion guidance for deformable 3d gaussian splatting. Advances in Neural Information Processing Systems **37**, 101790–101817 (2024)