H3D-DGS: Exploring Heterogeneous 3D Motion Representation for Deformable 3D Gaussian Splatting

Bing He [1] *& Yunuo Chen [1] *& Guo Lu [1] & Qi Wang [2] & Qunshan Gu [2]
& Rong Xie [1] & Li Song [1] †& Wenjun Zhang [1]
[1] Shanghai Jiao Tong University, [2] Alibaba Group
{sandwich_theorem}@sjtu.edu.cn
{cyril-chenyn}@sjtu.edu.cn

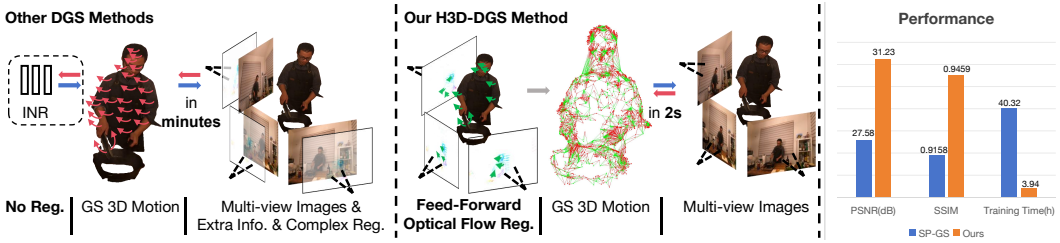


Figure 1: **H3D-DGS**: Previous deformable 3DGS methods heavily depend on gradient-based methods, often introducing complex loss regularizations to recover 3D motion. We challenge this convention by observing that, the observability of 3D motion varies across spatial directions. Specifically, the motion components observable in camera's image plane can be directly estimated using optical flow. In contrast, motion components orthogonal to the image plane must be inferred from additional viewpoints. We propose to directly incorporate these observable components into a fixed, feed-forward 3D motion representation, allowing the neural network to focus solely on learning the unobservable motion. To this end, we introduce a heterogeneous 3D motion representation—**H3D control points**—which decouple observable and learnable motion components. This heterogeneous structure serves as an effective inductive bias, promoting physically plausible 3D motion estimation. our H3D-DGS achieves both superior performance and faster convergence compared to existing methods.

Abstract

Dynamic scene reconstruction poses a persistent challenge in 3D vision. Deformable 3D Gaussian Splatting has emerged as an effective method for this task, offering real-time rendering and high visual fidelity. This approach decomposes a dynamic scene into a static representation in a canonical space and time-varying scene motion. Scene motion is defined as the collective movement of all Gaussian points, and for compactness, existing approaches commonly adopt implicit neural fields or sparse control points. However, these methods predominantly rely on gradient-based optimization for all motion information. Due to the high degree of freedom, they struggle to converge on real-world datasets exhibiting complex motion. To preserve the compactness of motion representation and address convergence challenges, this paper proposes heterogeneous 3D control points, termed **H3D control points**, whose attributes are obtained using a hybrid strategy combining optical flow back-projection and gradient-based methods. This design decouples directly observable motion components from those that are geometrically occluded. Specifically, components of 3D motion that project onto the

*Authors contributed equally to this work.
†Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

https://doi.org/10.52202/085713-3803

image plane are directly acquired via optical flow back projection, while unobservable portions are refined through gradient-based optimization. Experiments on the Neu3DV and CMU-Panoptic datasets demonstrate that our method achieves superior performance over state-of-the-art deformable 3D Gaussian splatting techniques. Remarkably, our method converges within just 100 iterations and achieves a per-frame processing speed of 2 seconds on a single NVIDIA RTX 4070 GPU.

1 Introduction

Reconstructing real-world scenes is a long-standing challenge in the field of 3D vision. Recently, 3D Gaussian Splatting (3D-GS) Kerbl et al. (2023) has demonstrated remarkable success in producing high-quality reconstructions for static scenes. This technique utilizes Gaussians to model the scene, assigning them with physically meaningful properties, and renders image by "splatting" these Gaussians onto the image plane.

Why control points? Compared to NeRF Mildenhall et al. (2021), a key advantage of 3D-GS lies in its discrete structure. This structure ensures that scene representation—via Gaussians—is concentrated at occupied regions within the scene, avoiding the inefficiencies of a global field that allocates unnecessary representational capacity to empty space. Similarly, the global implicit neural field used for deformation faces a similar issue: only a small part of the scene is dynamic, while the majority remains static. Therefore, a discrete, localized motion representation holds promise, as it offers precise and flexible modeling of 3D motions at a local level.

In our approach, we adopt control points as the motion representation because, despite being compact, they can efficiently represent local motion. We further distinguish between the background and the moving objects in the 3D scene, such that our control points are applied only to the latter.

Why heterogeneous? Realizing accurate 3D motion representation requires reconstructing motion unobservable from a single camera. In a multi-view system, traditional graphics methods Vedula et al. (1999) struggle to align complex 3D points. While gradient-based methods Luiten et al. (2023) circumvent the 3D alignment problem, they suffer from poor convergence due to their high degrees of freedom (DoF) and require extensive regularization, making them time-consuming.

Our heterogeneous approach differs by combining graphics-based techniques with gradient-based methods. Recognizing that optical flow is the 2D projection of scene flow, we introduce a local decoupling strategy, dividing local 3D motion into observable and unobservable components. Components of 3D motion that project onto the camera plane are directly acquired via optical flow back projection. In contrast, the unobservable portion is refined through gradient-based methods. Since we have incorporated regularization into the structural design of the control point, backpropagation in our method focuses only on the complex portion of the 3D motion. Our method mitigates convergence issues and achieves fast optimization. This new form of heterogeneous 3D motion representation is referred to as the "H3D control points" approach.

Streaming framework. Building on the introduced 3D motion representation, we present a novel generalized streaming framework for dynamic 3D real-world reconstruction with multiple camera setups. Beginning with an initial 3D reconstruction, our workflow decomposes the dynamic reconstruction process into distinct submodules: 3D segmentation, H3D control point generation, object-wise motion manipulation, and residual compensation. This structured approach minimizes accumulated errors and ensures a compact and robust representation.

Our key contributions are as follows:

- We propose a strategy to split 3D motion into observable and unobservable components. Observable motion (projected to the camera plane) is estimated directly via optical flow back-projection, while unobservable components are optimized using gradients backpropagation.
- We introduce **H3D control points**, a heterogeneous motion representation that significantly improves convergence and accuracy. Our method converges within 100 iterations and achieves a processing speed of 2 seconds per frame on a single NVIDIA 4070 GPU.
- We develop a streaming framework for real-world dynamic scene reconstruction, setting a new benchmark on the Neu3DV and CMU-Panoptic datasets.

2 Related Work

2.1 Dynamic Scene Reconstruction

Neural Radiance Field (NeRF) Mildenhall et al. (2021) have demonstrated strong performance in novel view synthesis by modeling scenes with global continuous implicit functions. Numerous

extensions have adapted NeRF to dynamic scenes. Mainstream methods Xian et al. (2021); Wang et al. (2021); Park et al. (2021a,b); Pumarola et al. (2021); Du et al. (2021); Li et al. (2021); Liu et al. (2022); Fang et al. (2022); Shao et al. (2023); Cao and Johnson (2023); Fridovich-Keil et al. (2023); Liu et al. (2023); Song et al. (2023) modeled dynamic scenes by learning a deformation field that warps a canonical 3D representation over time. Other research Wang et al. (2023b); Li et al. (2023); Lin et al. (2022, 2023) improved reconstruction quality by incorporating camera pose priors. Additional supervision information, such as depth Attal et al. (2021) and optical flow Wang et al. (2023a), were also employed to guide training. NeRFPlayer Song et al. (2023), used self-supervised learning to segment dynamic scenes into static, deforming, and newly appearing regions, applying tailored strategies to each.

Recently, 3D-GS Kerbl et al. (2023) introduced an elegant point-based rendering approach with efficient CUDA implementations. Many 4D Gaussian methods parallel NeRF-based approaches by incorporating temporal dynamics into spatial representations. For instance, Yang et al. (2023b) incorporated time-variant attributes into Gaussians, while Dynamic-GS Luiten et al. (2023) learned dense Gaussian motion directly. Subsequent works Guo et al. (2024); Zhu et al. (2024) incorporated optical flow to enhance motion accuracy. 3DGStream Sun et al. (2024) proposed a Neural Transformation Cache to model per-frame motion. Gaussian-Flow Lin et al. (2024) introduced a Dual-Domain Deformation Model for point-wise motion representation. Spacetime Gaussian Li et al. (2024) proposed a feature splatting and rendering approach. Several studies Wu et al. (2023); Yang et al. (2023a); Huang et al. (2023); Lin et al. (2024); Das et al. (2024); Yu et al. (2024) deformed canonical Gaussians using global implicit fields to capture 4D dynamics.

2.2 3D Control Points

Real-world scenes typically consist of a large static background and a smaller dynamic foreground. While global neural fields are compact in representation, they often lack the flexibility to selectively model only the dynamic regions, and require architectural redesigns to adapt across different scenes. In contrast, 3D control point methods are promising due to their ability to flexibly capture localized motion and scale effectively. Traditional graphics approaches Sorkine (2005); Yu et al. (2004) have long offered flexible deformation techniques that preserve geometric details. Among these, Sumner et al. (2007) introduced Embedded Deformation (ED) graphs, which use sparse control points to represent the motion of dense surfaces, achieving a balance between compactness and flexibility. HiFi4G Jiang et al. (2023) directly adopted ED-graphs, but its surface reconstruction requires dense camera coverage and is computationally expensive. Moreover, unlike meshes, the volumetric radiance representation of 3D Gaussians is not well-suited for the thin nature of surfaces Huang et al. (2024). Recent gradient-based methods have explored the use of control points for compact motion representation in 3D. SP-GS Wan et al. (2024) clustered Gaussians into superpoints, primitive motion groups, while SC-GS Huang et al. (2023) also adopted the concept of control points, although its direct optimization is challenged by high degrees of freedom (DoFs).

Table 1: Method Comparison. Rep. and Manip. are abbreviations for representation and manipulation, respectively.

	Large Move	Fast Train	Compact Rep.	Motion Manip.
Dy-GS	✓	✗	✗	✗
MA-GS	✗	✗	✓	✗
4D-GS	✗	✓	✓	✗
SP-GS	✗	✗	✓	✗
SC-GS	✗	✗	✓	✓
Ours	✓	✓	✓	✓

Table 2: Average reconstruction results for the Neu3DV dataset. Training time is reported in hours.

Metrics	PSNR↑	SSIM↑	LPIPS↓	Training Time.
Dy-GS	27.65	0.9232	0.1313	57.35
MA-GS	28.76	0.9299	0.1146	45.38
4D-GS	30.49	0.9401	0.0998	6.88
SP-GS	27.03	0.9109	0.1225	40.32
Ours	**30.91**	**0.9437**	**0.0941**	**1.89**

Our method introduces four key distinctions from existing gradient-based approaches. First, our method robustly handles real-world datasets with large-scale motion Joo et al. (2017), where prior methods Guo et al. (2024); Huang et al. (2023); Wan et al. (2024); Wu et al. (2023) struggle. Second, we introduce structural regularization that improves training efficiency, enabling our model to train as fast as 2 seconds per frame. Third, our H3D control points offer a compact motion representation, requiring only 0.2% as many control points as Gaussians. In comparison, methods like Dynamic-GSLuiten et al. (2023) and GS-Flow Gao et al. (2024) use redundant motion representations, applying residuals to every Gaussian. Fourth, our control points influence nearby Gaussians spatially,

https://doi.org/10.52202/085713-3803

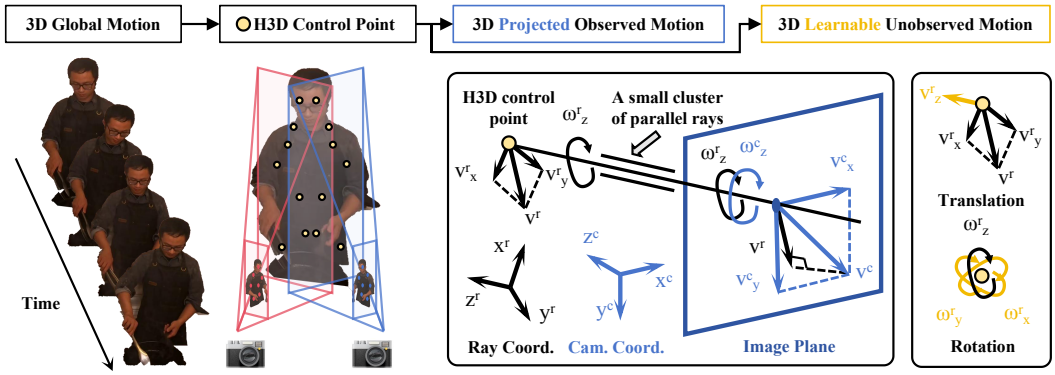


Figure 2: **H3D Control Points.** To predict dense 3D motion in a sparse manner, we propose H3D control points containing local translation and rotation information. Unlike previous works which learn all motion information with gradient-based method, we exploit 2D motion priors derived from the optical flow. Both translation and rotation are divided into projected observable part and learnable unobservable part. We model the localized light as parallel rays and make a detailed derivation.

inheriting the desirable properties of traditional control points and supporting user-driven motion manipulation Huang et al. (2023).

3 Method

In this section, we introduce our method for dynamic scene reconstruction with Gaussians. First, we present the notations and symbols used throughout the section to facilitate understanding. We then begin with the key idea of our approach—motion decoupling—and provide its detailed formulation. Next, we introduce the control point action pattern, where motion information is constructed using the proposed motion decoupling strategy. Based on the resulting H3D control points, we describe the full pipeline for dynamic scene reconstruction. Finally, we present the associated loss functions used to optimize the model.

3.1 Mathematical Notation

The symbols used in our method are defined as follows:
Scalar values are denoted in standard font, while vectors and matrices are bolded. The superscripts indicate reference systems, and the subscripts denote point-specific information. Euclidean distance is represented by $\|\cdot\|$, and learnable parameters are marked with a hat symbol $\hat{\cdot}$.
Positions x, y and optical flow values u, v are measured in pixels, while 3D positions X, Y, Z and velocities $\mathbf{V}$ are measured in meters. The focal length of the camera is denoted as f, in pixels in Eq. 1, in meters in Eqs. 2 3, (c_x, c_y) indicates the center of projection in pixels, $\mathbf{K}$ represents the intrinsic matrix of the camera, $\mathbf{R}$ is the rotation matrix, $\mathbf{q}$ denotes the unit rotation quaternion, and $\mathbf{t}$ represents translation.
For simplicity, linear velocity $\mathbf{V}$ and translation $\mathbf{t}$, as well as angular velocity $\boldsymbol{\omega}$ and angular variation represented by the unit quaternion $\mathbf{q}$, are treated analogously in our calculations, as both represent changes in motion attributes between two neighboring frames.

3.2 Local 3D Motion Decoupling and Heterogeneous 3D Control Points Generation

To effectively represent local translation and rotation, the attributes of a 3D control point encompass its 3D position, local spatial translation, and rotation.
Typically, the 3D position of a point is obtained by back projection. Given a point located at (x_0^c, y_0^c) on the image plane with depth Z^c, its 3D position in the camera coordinate system is defined as:

$$\mathbf{X}^c = \frac{Z^c}{f}(x_0^c - c_x, y_0^c - c_y, f)^{\mathrm{T}}. \tag{1}$$

Now we focus on translation and rotation. At any given viewpoint, a portion of the 3D rigid motion is projected onto the image plane. This motion component can therefore be extracted directly from optical flow and projected into space with appropriate design.

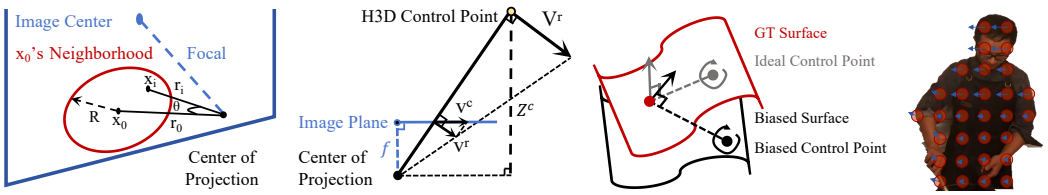


Figure 3: **Auxiliary Diagrams for Local Motion Mapping.** (a) An illustration of angles, points, and rays within the neighborhood of $\mathbf{x}_0$. (b) A quantitative depiction of motion projection. (c) A comparison between Gaussians distributed on the ground truth surface and control points located on a biased surface. (d) Illustration for grid sampling. Grid sampling is performed independently for each camera to provide a 2D motion prior for H3D control points.

To facilitate this, we define a "ray coordinate system," where the z-axis aligns with the H3D control point, extending from the camera toward the scene. This motion decoupling process is illustrated in Fig. 2. The components v_x^r, v_y^r, ω_z^r are derived from optical flow, while $v_z^r, \omega_x^r, \omega_y^r$ remain learnable. To extend the motion representation to the local space rather than just a single point, we introduce a local ray approximation assumption. This allows any point x_i in the neighborhood $\mathcal{N}$ around x_0 (see Fig. 3(a)) to share the same local coordinate system. The derivation, given in the Appendix, demonstrates that parallel light retains only motion components perpendicular to the ray direction. The relationship between image plane optical flow and true motion information is thus projective:

$$\mathbf{v}^r = f * \frac{1}{N} \sum_{\mathbf{x}_i \in \mathcal{N}} (\frac{\mathrm{d}x_i^r}{\mathrm{d}t}, \frac{\mathrm{d}y_i^r}{\mathrm{d}t})^{\mathrm{T}} = f * \frac{1}{N} \left[\mathbf{R}_{cr}\right]_{:2,:2} \left[\mathbf{K^{-1}}\right]_{:2,:2} \sum_{\mathbf{x}_i \in \mathcal{N}} (u_i, v_i)^{\mathrm{T}} \tag{2}$$

where (u_i, v_i) denotes the optical flow of points $\mathbf{x}_i$. The inverse intrinsic matrix $\mathbf{K^{-1}}$ converts the optical flow to meters, and rotation matrix $\mathbf{R}_{cr}$ transforms the system from the camera to the ray coordinate system. The notation $[\cdot]_{:2,:2}$ selects the matrix's first two rows and columns, and N is the number of pixels in $\mathcal{N}$. To better interpret scene dynamics $\mathbf{v}_r$, we use the focal length f to adjust translation from normalized to physical space.

As illustrated in Fig. 3(b), the formulas for translational and rotational information can be expressed respectively as:

$$\mathbf{V}^r = (\frac{\mathrm{d}X^r}{\mathrm{d}t}, \frac{\mathrm{d}Y^r}{\mathrm{d}t})^{\mathrm{T}} = \frac{Z^c}{f}\mathbf{v}^r, \tag{3}$$

$$\omega_z^r = \frac{1}{N} \sum_{\mathbf{x}_i \in \mathcal{N}} \frac{(\mathbf{x}_i^r - \mathbf{x}_0^r) \times \mathbf{v}_i^r}{\|\mathbf{x}_i^r - \mathbf{x}_0^r\|^2}. \tag{4}$$

Thus, in the ray coordinate system, 3D position $\mathbf{X}^c$, 2D translation $\mathbf{V}^r$, and 1D rotation ω_z^r of the H3D control point are determined, while the learnable parts $\hat{V}_z^r, \hat{\omega}_x^r, \hat{\omega}_y^r$ complete the motion information, represented as:

$$\mathbf{t^r} = (V_x^r, V_y^r, \hat{V}_z^r), \tag{5}$$

$$\mathbf{q^r} = \mathrm{Euler2Quat}((\hat{\omega}_x^r, \hat{\omega}_y^r, \omega_z^r)), \tag{6}$$

The function $\mathrm{Euler2Quat}$ converts Euler angles to quaternions. Euler angles are more intuitive and interpretable for decoupling and controlling rotation components, while quaternions are used in the underlying 3D Gaussian Splatting framework operates to represent rotation.

At this stage, all attributes of the H3D control point are derived, completing its construction from the optical flow. For each camera, H3D control points are independently sampled using a uniform grid pattern, as illustrated in Fig.3(d). The sampling interval determines the density of H3D control points, and the radius of the circular sampling area represents the range of motion cues. The H3D control points from all cameras are aggregated to represent the motion across the entire scene. An intuitive schematic is also provided in Appendix Fig. 7(b).

We also experimented with the incorporation of local macro-rotation following the ED-graph method Sumner et al. (2007), but observed a decline in performance. We attribute this to structural differences between Gaussians and meshes: while meshes are densely distributed along object

https://doi.org/10.52202/085713-3803

surfaces, Gaussians exhibit a looser spatial distribution due to their higher degrees of freedom. Consequently, we omit the component concerning local macro rotation to minimize biases introduced by rough depth estimations, as illustrated in Fig. 3(c). Although this adjustment compromises some sparsity, it optimizes our motion representation approach for volumetric radiance representations like Gaussians.

3.3 Motion Manipulation

The translation $\mathbf{t}$ and rotation $\mathbf{q}$ of each Gaussian G are computed by interpolating from its K-nearest Cover and Hart (1967) control points C_i within its spatial neighborhood $\mathcal{N}_G$. The interpolation weights w_i are inversely proportional to the Euclidean distances to the control points.

$$(\mathbf{t}, \mathbf{q}) = \sum_{C_i \in \mathcal{N}_G} w_i * (\mathbf{t}_i, \mathbf{q}_i), \tag{7}$$

$$w_i = \frac{1/\left\|\mathbf{X}_G - \mathbf{X}_{C_i}\right\|}{\sum_{C_i \in \mathcal{N}_G} 1/\left\|\mathbf{X}_G - \mathbf{X}_{C_i}\right\|}. \tag{8}$$

The position $\mathbf{X}_t$ and rotation $\mathbf{q}_t$ of the Gaussian at time t are given by:

$$\mathbf{X}_t = \mathbf{t} + \mathbf{X}_{t-1} \tag{9}$$

$$\mathbf{q}_t = \mathbf{q}\mathbf{q}_{t-1} \tag{10}$$

In practice, we set K = 3, meaning that each moving Gaussian derives its motion from the three nearest control points.

3.4 Streaming workflow

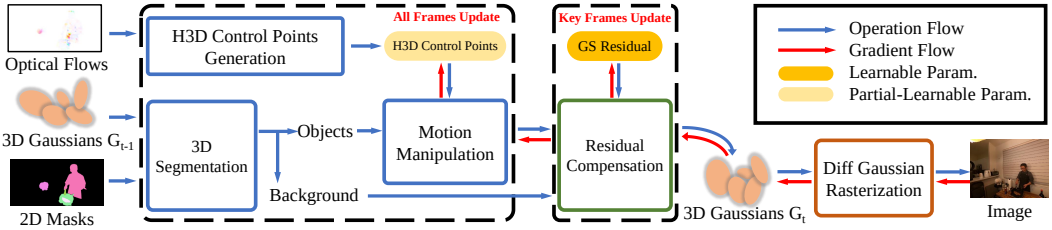


Figure 4: **Streaming workflow.** The workflow starts by segmenting the scene into a static background and moving objects using 3D segmentation algorithm. Optical flow is then applied to generate H3D control points. Motion-related attributes of the Gaussians are manipulated on an object-wise basis. To prevent reconstruction failures from error accumulation, Gaussian attributes are periodically updated in a keyframe manner, capturing additional scene information as attribute residuals of the Gaussians.

The streaming workflow is depicted in Fig. 4. The proposed method comprises four independent modules: 3D segmentation, H3D control point generation, object-wise motion manipulation, and residual compensation. Inputs include 3D Gaussians, 2D masks, and optical flow. 3D Gaussians are obtained either from the initial static scene reconstruction or the previous frame. The 2D object masks are generated via the SAM-track method Cheng et al. (2023). Optical flows are derived from the DIS method Kroeger et al. (2016).

3D Segmentation. The purpose of 3D segmentation is to label each Gaussian as either part of a moving object or the static background. The control points act on their spatially proximate Gaussians. To ensure they only influence moving objects, it is essential to accurately define the spatial regions where each local representation applies. To achieve this, we utilize multiview masks and employ a Gaussian category voting algorithm to segment the scene into dynamic objects and static background regions, following an approach similar to SA-GS Hu et al. (2024). Specifically, each Gaussian is projected onto the image planes of all training viewpoints. By tallying the category labels across views, the Gaussian is assigned the category with the most votes. H3D control points, which also retain category labels when projected back into 3D, only manipulate Gaussians of the same category. This framework inherently supports topological transformations between objects with different categories but similar spatial locations.

Residual Compensation. This module is designed to mitigate error accumulation and maintain stable long-term reconstruction. Inspired by video coding techniques like residual coding Wiegand et al. (2003); Sze et al. (2014), we integrate a keyframe-based update mechanism. The system

performs full optimization of both Gaussians and H3D control points at keyframes, whereas during non-keyframe intervals, Gaussian attributes remain fixed and only control point parameters are updated. To formalize this structure, we introduce the concept of a Group of Scenes (GOS), where GOS-N denotes a sequence with one keyframe followed by N-1 non-keyframes. This design achieves a balance between computational efficiency and temporal reconstruction stability.

3.5 Loss Function

We adopt the same single-image reconstruction loss used in the original 3D Gaussian Splatting framework, consisting of an L1 loss combined with a D-SSIM term, which are computed between the rendered image I_{render} and the ground truth image I_{gt}:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_1(I_{render}, I_{gt}) + \lambda\mathcal{L}_{\text{D-SSIM}}(I_{render}, I_{gt}), \tag{11}$$

with λ set to 0.2. Our motion modeling and streaming pipeline are designed for both computational efficiency and robustness, enabling effective convergence without the need for additional loss constraints.

4 Experiment

Figure 5: **Left:** Frame 20 of the "coffee_martini" sequence from the Neu3DV dataset. **Right:** Frame 74 of the "softball" sequence from the CMU-Panoptic dataset.

4.1 Datasets and Implementation Details

Neu3DV Dataset Li et al. (2021). The Neural 3D Video Synthesis Dataset includes six sequences, originally captured at a resolution of 2704 × 2028, which were downsampled to 1352 × 1014 for training. The sequence 'flame_salmon_1' contains 1200 frames, while the remaining five sequences consist of 300 frames each. All sequences were recorded using 15 to 20 static cameras, all placed to form a fanned-out arrangement in front of the scene.

CMU-Panoptic Dataset Joo et al. (2017). The CMU-Panoptic Dataset comprises three sequences featuring complex, dynamic object motions. Each sequence is recorded at a resolution of 640 × 360 and consists of 150 frames. The data was collected using 31 static cameras evenly distributed in a spherical arrangement around the scene, with 27 used for training and 4 reserved for testing,

Implementation Details. All experiments were conducted on NVIDIA RTX 4070 GPUs. To evaluate the effectiveness of our H3D control point method for motion representation, we compared it with SP-GS Wan et al. (2024), SC-GS Huang et al. (2023), Dynamic-GS Luiten et al. (2023), MA-GS Guo et al. (2024) and 4D-GS Wu et al. (2023), all maintaining a constant number of Gaussian points. Using the official implementations provided by each baseline, we ensured identical 3D point initialization. In accordance with the 4D-GS setup, we employed COLMAP Schönberger and Frahm (2016); Schönberger et al. (2016) to initialize 3D points from the first training frames and followed the 10k-iteration reconstruction strategy of Dynamic-GS. For subsequent frames, we trained keyframes with 500 iterations and non-keyframes with 100 iterations, using a single Adam optimizer with fixed learning rates as in Dynamic-GS. Since our method requires both object masks and optical flow as inputs, we specify the sources for these data. For the Neu3DV Dataset, we obtained the object masks using the SAM-track method Cheng et al. (2023). For the CMU Dataset, we reused the foreground masks provided by Dynamic-GS Luiten et al. (2023). Unless otherwise specified, we used DIS Kroeger et al. (2016) as the default optical flow model. The H3D control point sampling configurations were tailored to each dataset based on resolution and motion complexity. We set the

https://doi.org/10.52202/085713-3803

Table 3: Per-scene results for the Neu3DV dataset. Each cell is color-coded to denote performance ranking: best for the top performance, second for the second best, and third for the third best.

Scene	*sear_steak*			*cook_spinach*			*cut_roasted_beef*		
Metrics	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Dynamic-GS [27]	31.38	0.9469	0.1119	29.98	0.9388	0.1179	29.64	0.9360	0.1248
MA-GS [11]	30.36	0.9508	0.0854	31.15	0.9378	0.1053	31.17	0.9401	0.1157
4D-GS [48]	31.62	0.9569	0.0808	32.79	0.9522	0.0926	32.13	0.9467	0.0959
SP-GS [43]	30.75	0.9474	0.0931	31.32	0.9445	0.0914	30.44	0.9457	0.0942
SC-GS [14]	31.60	0.9510	0.1345	-	-	-	-	-	-
Ours-GoS1	33.23	0.9654	0.0719	33.20	0.9586	0.0796	33.00	0.9609	0.0795
Ours-GoS5	33.72	0.9661	0.0704	32.91	0.9579	0.0819	33.23	0.9592	0.0835
Ours-GoS10	33.64	0.9655	0.0716	32.65	0.9553	0.0861	32.47	0.9555	0.0890
Scene	*flame_steak*			*flame_salmon_1*			*coffee_martini*		
Metrics	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Dynamic-GS [27]	30.41	0.9429	0.1121	20.19	0.8875	0.1583	24.29	0.8870	0.1630
MA-GS [11]	29.14	0.9456	0.1008	25.05	0.9075	0.1274	25.72	0.8979	0.1533
4D-GS [48]	29.28	0.9545	0.0836	28.27	0.9106	0.1289	28.87	0.9198	0.1168
SP-GS [43]	25.59	0.8934	0.1248	25.13	0.9057	0.1320	19.26	0.8291	0.1996
SC-GS [14]	-	-	-	-	-	-	24.82	0.8972	0.2239
Ours-GoS1	32.84	0.9645	0.0723	28.00	0.9173	0.1083	26.90	0.9140	0.1170
Ours-GoS5	33.18	0.9649	0.0707	27.65	0.9155	0.1127	26.71	0.9119	0.1242
Ours-GoS10	32.94	0.9631	0.0733	27.17	0.9127	0.1165	26.51	0.9100	0.1283

Table 4: Per-scene results for the CMU-Panoptic dataset.

Scene	*softball*			*boxes*			*basketball*		
Metrics	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Dynamic-GS [27]	26.93	0.9076	0.1804	27.79	0.9069	0.1769	28.54	0.9032	0.1812
Ours-GoS2	27.48	0.9264	0.1374	27.88	0.9227	0.1413	27.72	0.9203	0.1423

grid interval to 64 pixels with a 16-pixel radius for the Neu3DV dataset, and 32 pixels with an 8-pixel radius for the CMU dataset.

4.2 Reconstruction Results

Quantitative Results. We first report average quantitative results including PSNR, SSIM, and LPIPS in Tab. 2. Our method consistently outperformed SP-GS, MA-GS, Dynamic-GS, 4D-GS across all metrics and demonstrated a significant advantage in training time. While MA-GS also leveraged optical flow, it did so via a gradient-based method with motion represented as an implicit neural field. Our approach achieved superior visual quality and more efficient training. Per-scene quantitative results are provided in Tab. 3 and Tab. 4. Our approach achieved state-of-the-art (SOTA) PSNR performance in most sequences and yields even better SSIM and LPIPS scores across more scenes, indicating improved perceptual reconstruction quality. In the Neu3DV dataset, SC-GS demonstrates vulnerability when dealing with real-world dynamic scenarios, failing to converge efficiently across multiple sequences. In the CMU-Panoptic dataset, MA-GS, SP-GS, SC-GS, 4D-GS all encounter difficulties handling dynamic motion. For instance, 4D-GS fails under rapid scene movement, resulting in object disappearance as depicted in Fig. 5.

Subjective Assessment. In the Neu3DV dataset, the only case where our method did not surpass 4D-GS in objective metrics is shown in Fig. 5. This is primarily due to suboptimal static scene initialization, which is decoupled from the subsequent motion modeling in our pipeline. The resulting background artifacts lowered the overall score. In dynamic regions, however, our method still delivered superior detail fidelity and accurate scene reconstruction. In contrast, 4D-GS benefited from global optimization of the static background, boosting its overall metric but producing inferior quality in motion areas. Notably, 4D-GS reconstructed non-existent coffee liquid. SP-GS amplified background artifacts by using unreliable clustering centers as motion control points. SC-GS suffered

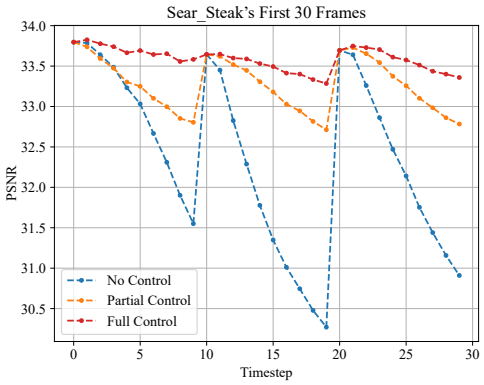


Figure 6: An example illustrating reconstruction quality degradation across frames under three different settings of 3D control points.

Table 5: Comparison between Gaussians and control points.

Points Category	Points Num.	Attr. Dim.	Param. Num.
Scene GS	$> 100k$	≥ 13	$> 1000k$
Obj. GS	$\sim 10k$	≥ 13	$> 100k$
Obj. Ctrl	$0.2k - 2.5k$	9	$1.8k - 22.5k$

Table 6: Comparison between different optical flow methods.

O.F. Model	2D MSE↓	PSNR↑	SSIM↑	LPIPS↓
PWC [39]	4.553e-5	33.39	0.9644	0.0737
SpyNet [32]	1.509e-5	33.55	0.9649	0.0725
DIS [18]	1.230e-5	33.64	0.9655	0.0716

from blurry dynamic regions, as its high-degree-of-freedom control points are challenging to train via gradient descent. Dynamic-GS produced blurry results because their complex loss functions lack generalization to dynamic scenarios.
In the CMU-Panoptic dataset, Dynamic-GS reconstructed noisy backgrounds and blurred feet, while 4D-GS struggled with large motion due to limitations in its global deformation field. Our method, by contrast, preserved fine textures and structural details without over-smoothing or object disappearance. Subjective comparisons clearly show our method excels in handling challenging motion. Further qualitative results are provided in the **supplementary video**.

Covergence Speed. Our non-keyframes converged within 2 seconds per frame, and keyframes took approximately 10 seconds each. We anticipate further reductions in processing time as we continue refining the implementation.

4.3 Ablation Study

Points Parameter Comparison. In Tab. 5, we compare 3D points across various categories to highlight the compactness of our control point method. Each Gaussian is characterized by 13 attributes. The number of parameters increases with higher harmonic degrees. In contrast, the number of 3D control points is significantly smaller, and their attributes remain fixed, indicating a much more compact representation and greater potential for streaming transmission.

Effectiveness of H3D Control Points. The rendering quality of three methods was evaluated under the GoS-10 configuration: **No Control**, which lacks motion modeling and simply duplicates keyframes to subsequent non-keyframes; **Partial Control**, which manipulates the scene using only the projected motion attributes in H3D control points; and **Full Control**, which manipulates the scene using the complete set of motion attributes in H3D control points.
The PSNR values for the first 30 frames of the sear_steak sequence are presented in Fig. 6. The **No Control** method experienced rapid quality degradation over time, while **Partial Control** showed moderate improvement by modeling partial motion. In contrast, **Full Control** consistently maintained higher quality throughout the sequence.
By examining the gap between curves over each segment (e.g., steps 0–9, 10–19, 20–29), one can observe the progressive contribution of each component. Moreover, focusing solely on the **Full Control** setting, which corresponds to the GoS-10 setting in our main experiment, one can evaluate how reconstruction quality evolves over time in a streaming setup. As small errors accumulate frame-by-frame (e.g., 0–9, 10–19, 20–29), the residual compensation module applied at keyframes (e.g., 9–10, 19–20) effectively corrects these accumulated errors and restores high-quality reconstruction.

More Advanced Optical Flow Method Leads to Better Reconstruction Result. We further analyzed the impact of different optical flow algorithms Kroeger et al. (2016); Ranjan and Black (2017); Sun et al. (2018) on reconstruction performance. Among them, DIS Kroeger et al. (2016) provided the most accurate 2D flow estimates on the Neu3DV dataset. Optical flow accuracy was assessed via the average MSE between each frame and its warped predecessor, where lower 2D alignment error correlated strongly with improved 3D reconstruction quality. This highlights the potential for enhancing the pipeline by adopting more advanced optical flow methods. Furthermore,

https://doi.org/10.52202/085713-3803

the observed positive correlation between optical flow accuracy and reconstruction quality validates the effectiveness of our 3D motion modeling approach.

5 Conclusion and Discussion

We propose a novel heterogeneous 3D motion representation framework to address the challenges of dynamic scene reconstruction. By integrating discrete local motion modeling and leveraging H3D control points, our approach effectively decouples observable and learnable components of 3D motion, enabling precise and flexible representation. The framework further establishes a streaming pipeline that incorporates key innovations in 3D segmentation, motion manipulation, and residual compensation, ensuring robust and efficient reconstruction.

Our method surpasses existing state-of-the-art 4D Gaussian splatting approaches on real-world datasets. However, it still has limitations. The quality of 4D reconstruction is influenced by the initial static reconstruction, which remains an underexplored area with room for improvement. Additionally, the current method requires multi-view inputs from initial frames and does not support monocular video, a limitation we aim to address in future work.

Another limitation arises from background artifacts such as jittering, which are particularly noticeable in static regions (e.g., windows in some video sequences). These artifacts primarily stem from the incremental nature of our streaming reconstruction framework. Unlike global optimization methods that jointly optimize across all frames, our method updates each frame sequentially based on the previous reconstruction and the current input. While this design improves flexibility and enables low-latency streaming reconstruction, it also introduces susceptibility to cumulative noise and temporal drift, leading to observable floaters and jitter in the background. Addressing these artifacts presents an important direction for future work, for example by incorporating lightweight temporal smoothing or consistency mechanisms, while maintaining the efficiency of online reconstruction.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China under Grant Nos. 62431015 and 62471290.

References

David Arthur, Sergei Vassilvitskii, et al. k-means++: The advantages of careful seeding. In *Soda*, pages 1027–1035, 2007.

Benjamin Attal, Eliot Laidlaw, Aaron Gokaslan, Changil Kim, Christian Richardt, James Tompkin, and Matthew O'Toole. Törf: Time-of-flight radiance fields for dynamic scene view synthesis. *Advances in neural information processing systems*, 34:26289–26301, 2021.

Ang Cao and Justin Johnson. Hexplane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 130–141, 2023.

Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023.

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

Devikalyan Das, Christopher Wewer, Raza Yunus, Eddy Ilg, and Jan Eric Lenssen. Neural parametric gaussians for monocular non-rigid object reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10715–10725, 2024.

Yilun Du, Yinan Zhang, Hong-Xing Yu, Joshua B Tenenbaum, and Jiajun Wu. Neural radiance flow for 4d view synthesis and video processing. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14304–14314. IEEE Computer Society, 2021.

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.

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

Quankai Gao, Qiangeng Xu, Zhe Cao, Ben Mildenhall, Wenchao Ma, Le Chen, Danhang Tang, and Ulrich Neumann. Gaussianflow: Splatting gaussian dynamics for 4d content creation. *arXiv preprint arXiv:2403.12365*, 2024.

Zhiyang Guo, Wengang Zhou, Li Li, Min Wang, and Houqiang Li. Motion-aware 3d gaussian splatting for efficient dynamic scene reconstruction. *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

Xu Hu, Yuxi Wang, Lue Fan, Junsong Fan, Junran Peng, Zhen Lei, Qing Li, and Zhaoxiang Zhang. Semantic anything in 3d gaussians. *arXiv preprint arXiv:2401.17857*, 2024.

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. *arXiv preprint arXiv:2403.17888*, 2024.

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. *arXiv preprint arXiv:2312.14937*, 2023.

Yuheng Jiang, Zhehao Shen, Penghao Wang, Zhuo Su, Yu Hong, Yingliang Zhang, Jingyi Yu, and Lan Xu. Hifi4g: High-fidelity human performance rendering via compact gaussian splatting. *arXiv preprint arXiv:2312.03461*, 2023.

Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Scott Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. Fast optical flow using dense inverse search. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 471–488. Springer, 2016.

Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021.

Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4273–4284, 2023.

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.

Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Efficient neural radiance fields for interactive free-viewpoint video. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022.

Haotong Lin, Sida Peng, Zhen Xu, Tao Xie, Xingyi He, Hujun Bao, and Xiaowei Zhou. Im4d: High-fidelity and real-time novel view synthesis for dynamic scenes. *arXiv preprint arXiv:2310.08585*, 2023.

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.

Jia-Wei Liu, Yan-Pei Cao, Weijia Mao, Wenqiao Zhang, David Junhao Zhang, Jussi Keppo, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Devrf: Fast deformable voxel radiance fields for dynamic scenes. *Advances in Neural Information Processing Systems*, 35:36762–36775, 2022.

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.

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023.

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. *Communications of the ACM*, 65(1): 99–106, 2021.

 https://doi.org/10.52202/085713-3803

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, 2021a.

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. *arXiv preprint arXiv:2106.13228*, 2021b.

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.

Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4161–4170, 2017.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16632–16642, 2023.

Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2732–2742, 2023.

Olga Sorkine. Laplacian mesh processing. *Eurographics (State of the Art Reports)*, 4(4):1, 2005.

Robert W Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. In *ACM siggraph 2007 papers*, pages 80–es. 2007.

Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018.

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.

Vivienne Sze, Madhukar Budagavi, and Gary J Sullivan. High efficiency video coding (hevc). In *Integrated circuit and systems, algorithms and architectures*, page 40. Springer, 2014.

Sundar Vedula, Simon Baker, Peter Rander, Robert Collins, and Takeo Kanade. Three-dimensional scene flow. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 722–729. IEEE, 1999.

Diwen Wan, Ruijie Lu, and Gang Zeng. Superpoint gaussian splatting for real-time high-fidelity dynamic scene reconstruction. *arXiv preprint arXiv:2406.03697*, 2024.

Chaoyang Wang, Ben Eckart, Simon Lucey, and Orazio Gallo. Neural trajectory fields for dynamic novel view synthesis. *arXiv preprint arXiv:2105.05994*, 2021.

Chaoyang Wang, Lachlan Ewen MacDonald, Laszlo A Jeni, and Simon Lucey. Flow supervision for deformable nerf. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21128–21137, 2023a.

Feng Wang, Sinan Tan, Xinghang Li, Zeyue Tian, Yafei Song, and Huaping Liu. Mixed neural voxels for fast multi-view video synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19706–19716, 2023b.

Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576, 2003.

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. *arXiv preprint arXiv:2310.08528*, 2023.

Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9421–9431, 2021.

Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. *arXiv preprint arXiv:2309.13101*, 2023a.

Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. *arXiv preprint arXiv:2310.10642*, 2023b.

Heng Yu, Joel Julin, Zoltán Á Milacski, Koichiro Niinuma, and László A Jeni. Cogs: Controllable gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21624–21633, 2024.

Yizhou Yu, Kun Zhou, Dong Xu, Xiaohan Shi, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Mesh editing with poisson-based gradient field manipulation. In *ACM SIGGRAPH 2004 Papers*, pages 644–651. 2004.

Ruijie Zhu, Yanzhe Liang, Hanzhi Chang, Jiacheng Deng, Jiahao Lu, Wenfei Yang, Tianzhu Zhang, and Yongdong Zhang. Motiongs: Exploring explicit motion guidance for deformable 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 37:101790–101817, 2024.

 https://doi.org/10.52202/085713-3803

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.
Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.
The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.
IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: Abstract and Introduction

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Conclusion

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.

- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: Method 3.1,3.2 Appendix A.1

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Method, Experiment

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.

https://doi.org/10.52202/085713-3803

- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification: Code will be released after the paper is published.

   Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: Experiment 4.1 Appendix A.3

Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Not common in the field

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Abstract, Experiment 4.1

   Guidelines:

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

9. **Code of ethics**

https://doi.org/10.52202/085713-3803

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

Answer: [Yes]

Justification: Yes.

Guidelines:

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

10. **Broader impacts**

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

Answer: [NA]

Justification: no societal impact.

Guidelines:

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

11. **Safeguards**

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

Answer: [NA]

Justification: no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Yes.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: Yes.

    Guidelines:

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

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

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

    Answer: [NA]

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

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

 https://doi.org/10.52202/085713-3803

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [NA]

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

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [NA]

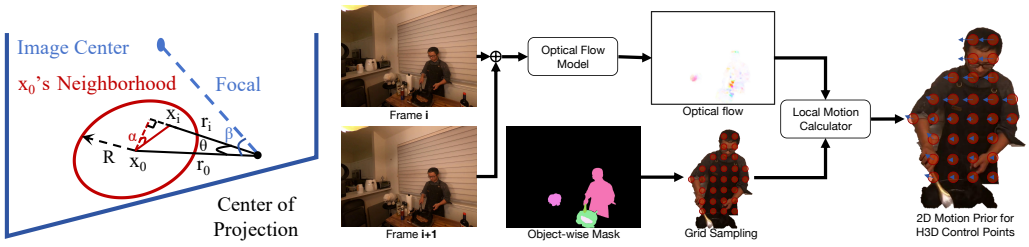
    Justification: the core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

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

A Appendix

A.1 Near-Parallel Light Hypothesis



(a) (b)

Figure 7: **Detailed diagrams for near-parallel light hypothesis proof and H3D control points sampling:** (a) Illustration for angles, points, rays in $\mathbf{x}_0$'s neighborhood. (b) Workflow for acquiring 2D motion prior.

In the image plane, we consider a small region centered at $\mathbf{x}_0$ with radius R. Within this region, the rays from the camera projection center to each pixel can be approximated as nearly parallel. A detailed illustration is provided in Fig. 7(a). For an arbitrary point $\mathbf{x}_i$ in the neighborhood of $\mathbf{x}_0$, we aim to show that the angle θ between the reference ray $\mathbf{r}_0$ (passing through $\mathbf{x}_0$) and the neighboring ray $\mathbf{r}_i$ is a first-order small quantity.

To begin, we drop a perpendicular (plumb line) from $\mathbf{x}_0$ to the ray $\mathbf{r}_i$, representing the shortest distance between the point and the ray. The length of this perpendicular segment can be expressed as:

$$\|\mathbf{x}_i - \mathbf{x}_0\| \cdot \cos\alpha, \tag{12}$$

where $|\cdot|$ denotes the Euclidean distance, and α is the angle between the plumb line and the line connecting $\mathbf{x}_0$ to $\mathbf{x}_i$.

Next, applying the cosine theorem, we represent the distance from the projection center to $\mathbf{x}_0$ as

$$f \cdot 1/\cos\beta, \tag{13}$$

where f is the camera focal length, and β is the angle between the ray $\mathbf{r}_0$ and the principal optical axis. Then, using the sine theorem, we can express the angle θ between rays $\mathbf{r}_0$ and $\mathbf{r}_i$:

$$\theta_i = \mathbf{arcsin}\left(\frac{\|\mathbf{x}_i - \mathbf{x}_0\|}{f} \cdot \frac{\cos\alpha}{1/\cos\beta}\right). \tag{14}$$

Since $\|\mathbf{x}_i - \mathbf{x}_0\|$ is always less than R, and the trigonometric terms involved are bounded by 1, the angle θ remains small as long as the focal length f is significantly larger than the neighborhood radius R.

Therefore, we conclude the proof of local ray near-parallelism. In practical applications, using a normalized focal length greater than 1000 pixels, it is sufficient to restrict the local region to a radius of 50 pixels to ensure the validity of this approximation.

A.2 2D Motion Prior Acquisition

We propose an intuitive workflow for acquiring 3D motion priors, as illustrated in Fig. 7(b). The inputs are two consecutive frames and object-wise masks.

First, an optical flow network estimates dense motion between the two frames. we generate a sampling grid for each object within the view. This grid is then passed to the local motion calculator—an abstracted version of the method described in Sec. 3.2—which processes the grid and associates the resulting 2D motion priors with the corresponding 3D control points.

It is worth noting that the 3D control points are acquired independently from all training viewpoints. This multi-view acquisition results in a denser and more redundant distribution of control points compared to single-view methods, thereby reducing the effective influence range of each point. To compensate for this, we adopt a larger sampling interval when selecting control points and use a smaller neighborhood around each point for computing motion priors.

https://doi.org/10.52202/085713-3803

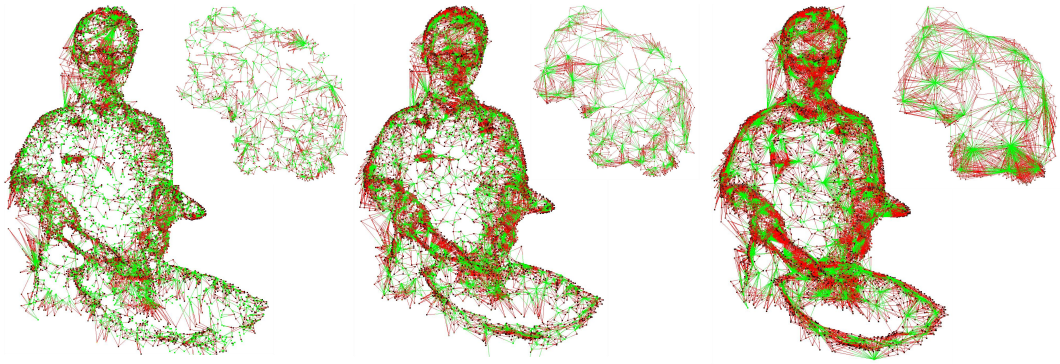


Figure 8: **Schematic comparison of Gaussians and control points for humans and dogs:** We visualized the topology using red and green line segments—red lines connect Gaussian points, while green lines connect control points. Additionally, we report the number of control points in the first frame for both human and dog subjects at various pruning rates, along with the corresponding reconstruction quality over the full sequence.

A.3 H3D Control Points Sparsification

H3D control point pruning is an optional step that further demonstrates the advantages of discrete motion representation. As shown in Fig.8, we visualize the correspondence between Gaussians and control points under varying pruning rates. By sparsifying the control points, we achieve substantial gains in representation compactness while maintaining comparable reconstruction quality. For stable clustering initialization—especially when using a large number of cluster centers—we recommend adopting the k-means++ strategyArthur et al. (2007). This method offers a good trade-off between computational efficiency and reconstruction accuracy, making it well-suited for scenarios requiring densely distributed control points.

A.4 3D Motion Visualization.

Figure 9: **3D Motion Visualization:** Visualization of Gaussian 3D motion in the "sear_steak" and "boxes" sequences.

We visualized the 3D motion of Gaussians in Fig. 9. The motion trajectories accurately depict the subject turning steaks and lifting boxes.

A.5 Significance of Initial 3D Scene.

The suboptimal static scene reconstruction in the first frame of the *coffee_martini* sequence noticeably impacted the quality of the subsequent 4D reconstruction. To further analyze this dependency, we evaluated PSNR, SSIM, and LPIPS-VGG metrics across sequences in the Neu3DV dataset. As shown in Fig. 10, the scatter plots exhibit a strong correlation between the quality of the initial 3D reconstruction and the final 4D results, with points closely following the y = x line. This alignment suggests that the initial static reconstruction defines an upper bound on the achievable dynamic reconstruction quality. Importantly, our method is designed to fully exploit this potential: it builds on

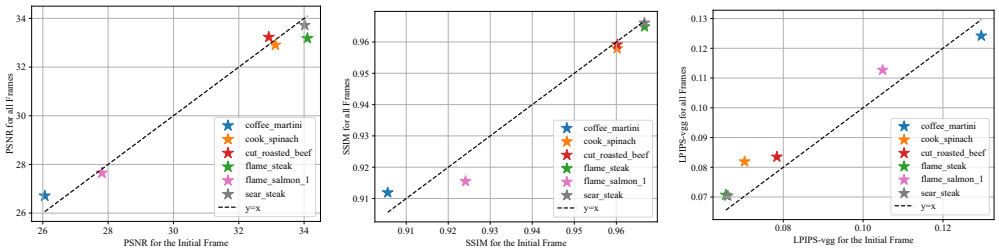


Figure 10: Reconstruction quality correlation between the initial frame and the entire video.

the static scene without introducing new Gaussians during the sequence, making efficient use of the available representation regardless of its initial fidelity.

A.6 More Detailed Settings for Fair Comparison

SP-GS Wan et al. (2024) and SC-GS Setting. We used the official codebases and initialized Gaussians from the same point cloud to ensure consistency.

MA-GS Guo et al. (2024) Setting. We followed the official implementation and used the same initial point cloud. We selected the deformation-based pipeline, which employs an implicit neural network for motion representation and incorporates optical flow via gradient-based optimization.

Dynamic-GS Luiten et al. (2023) Setting. This approach requires 2D foreground masks and segmentation labels for the initial 3D points. For a fair comparison, we merged our objects' 2D masks and applied the labeling strategy described in Sec. 3.4 to assign semantic categories to the initial 3D points. Additionally, we reduced the training iterations from 2k to 0.5k per frame when evaluating on the CMU-Panoptic dataset.

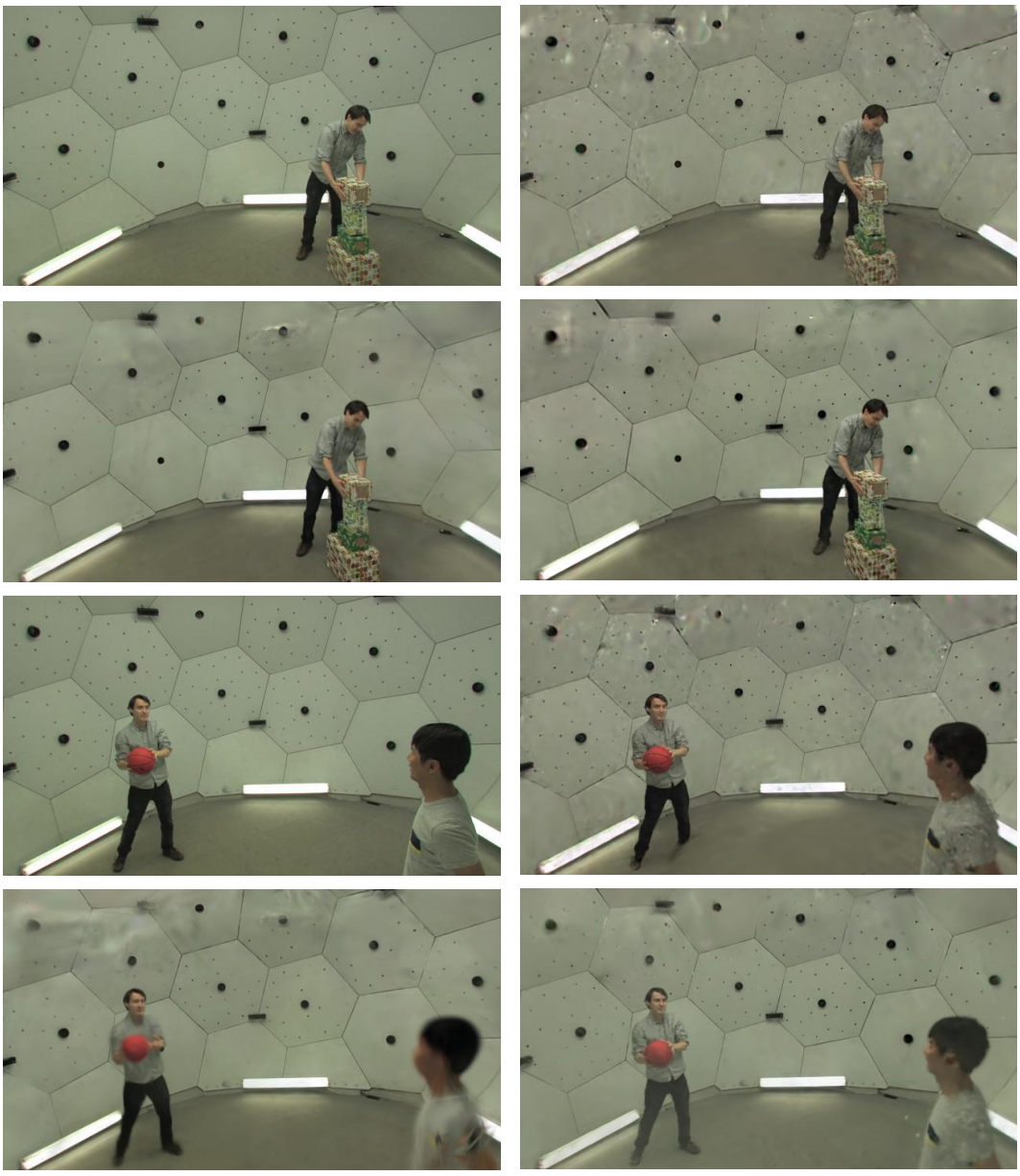
4D-GS Wu et al. (2023) Setting. For the "flame_salmon_1" sequence, four times longer than the other sequences, we expanded the training iterations from 17k to 68k to ensure a fair comparison.

A.7 Additional Subjective Results at Novel Viewpoints

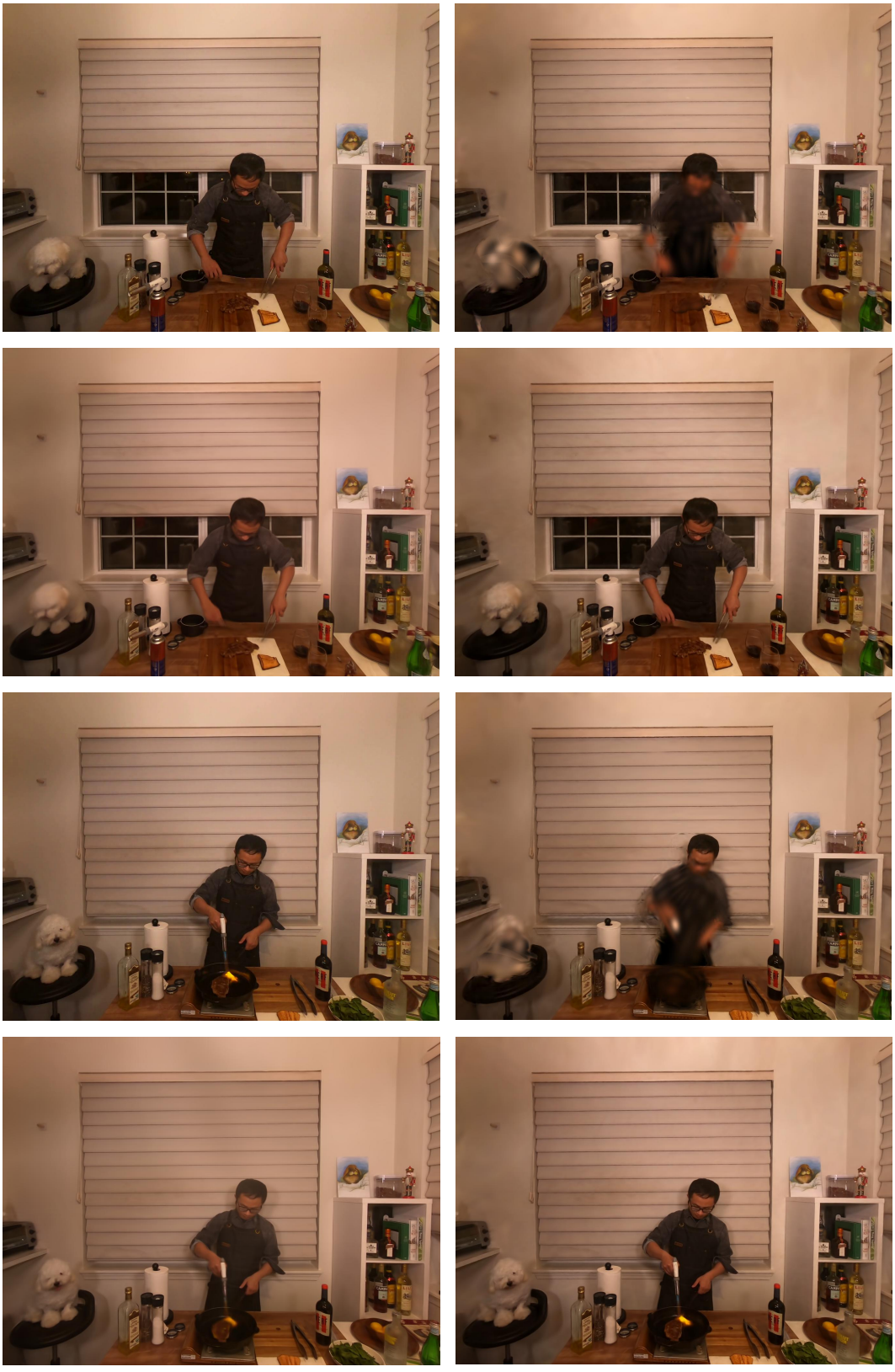
We present additional qualitative results from various sequences in both image and video formats to support an intuitive evaluation of our method. The images are organized in two-row groups, arranged from left to right and top to bottom in the following order: GT, Dynamic-GS, 4D-GS, and Ours. Corresponding video results are provided in the accompanying MP4 files.

https://doi.org/10.52202/085713-3803

Figure 11: More subjective outputs.



https://doi.org/10.52202/085713-3803



https://doi.org/10.52202/085713-3803