FreeGaussian: Annotation-free Control of Articulated Objects via 3D Gaussian Splats with Flow Derivatives

Qizhi Chen[1, 2*], Delin Qu[2, 3*], Junli Liu[2, 4], Yiwen Tang[2], Haoming Song[2], Dong Wang[2], Yuan Yuan[4], Bin Zhao[2, 4†]

[1]Zhejiang University
[2]Shanghai AI Laboratory
[3]Fudan University
[4]Northwestern Polytechnical University

Abstract

Reconstructing controllable Gaussian splats for articulated objects from monocular video is especially challenging due to its inherently insufficient constraints. Existing methods address this by relying on dense masks and manually defined control signals, limiting their real-world applications. In this paper, we propose an annotation-free method, **FreeGaussian**, which mathematically disentangles camera egomotion and articulated movements via flow derivatives. By establishing a connection between 2D flows and 3D Gaussian dynamic flow, our method enables optimization and continuity of dynamic Gaussian motions from flow priors without any control signals. Furthermore, we introduce a 3D spherical vector controlling scheme, which represents the state as a 3D Gaussian trajectory, thereby eliminating the need for complex 1D control signal calculations and simplifying controllable Gaussian modeling. Extensive experiments on articulated objects demonstrate the state-of-the-art visual performance and precise, part-aware controllability of our method.

Code — https://github.com/Tavish9/freegaussian

1 Introduction

Controllable view synthesis (CVS) aims to recover scenes containing multiple articulated objects and interactable motions of each object given a set of input views, it demands the recovered geometry, appearance, and motion faithfully respect the kinematic constraints of each articulated object while remaining photorealistic under novel viewpoints, which distinguishes it from conventional 4D reconstruction. Recently, CVS has attracted growing interest in content creation (Liao, Cao, and Shan 2024; Tang et al. 2023; Gao et al. 2024), virtual reality (Steuer 1992; Huang et al. 2025; Kang, Song, and Huang 2024; Steuer 1992; Kerbl et al. 2023; Waisberg et al. 2023) and robotic manipulation (Song et al. 2025; Qu et al. 2025b,a).

Recent advances leverage 3D Gaussian splatting (Kerbl et al. 2023) to achieve real-time, high-fidelity rendering of dynamic scenes (Yu et al. 2023; Yang et al. 2023) and have been scaled to scene-level datasets with dense annotations (Qu et al. 2024; Gao et al. 2025; Zeng et al. 2025). Yet, these methods remain fundamentally tied to manual supervision: they either require pixel-accurate part masks for each articulated link (Yu et al. 2023) or rely on pre-defined control signals in neural radiance fields (Kania et al. 2022; Qu et al. 2024). Without mask or control signal supervision, the model collapses, failing to decode features to color and losing scene control capabilities. Thus, dense part masks and control signal annotations have become a prerequisite for current articulated-object CVS, severely limiting real-world deployment.

To address this challenge, we propose **FreeGaussian**, a annotation-free but effective Gaussian splatting method for controllable scene reconstruction, which automatically explores interactable structures and restores scenes from successive frames, without any manual annotations. Dynamic Gaussian flow under instantaneous motion can be analytically derived from optical flow and camera egomotion via differential analysis. It enables us to localize controllable structures without masks and estimates joint-angle trajectories without any control signals. These consistent constraints are folded into training, enabling high-fidelity rendering and fine-grained manipulation of articulated objects while eliminating the need for manual supervision and extending practical applicability to real-world scenes.

More specifically, in the training stage, FreeGaussian directly derive dynamic Gaussians flow from optical flow and camera-induced camera flow, accumulated with Gaussian projection displacements. By tracking the dynamic Gaussian flow, we highlight interactive dynamic Gaussians and obtain their trajectories via HDBSCAN clustering, eliminating the dependence on manual mask annotations. To overcome the reliance on 1D control signal inputs, we introduce a 3D spherical vector controlling scheme that exploits 3D Gaussian scene representations bypassing dynamic Gaussian trajectories as state representations, aligning with the splatting rasterization pipeline and greatly simplifying the control process. During the control stage, the Gaussian dynamics are retrieved from the network, given the 3D control vector as input. Beyond localizing interactive Gaussians, the dynamic Gaussian flow constraints 3DGS motion between

*These authors contributed equally.
†Corresponding author.

frames, guaranteeing smooth motion and eliminating ghosting artifacts to improve rendering quality.

Extensive evaluations show that our method outperforms existing methods significantly in both novel view synthesis and articulated object controlling, enabling more accurate and efficient modeling of interactable content with no annotations. Contributions can be summarized as follows:

- We propose **FreeGaussian**, a novel annotation-free Gaussian Splatting method for controllable scene reconstruction, which automatically explores interactable scene objects with flow priors, and restores scene interactivity without any manual annotations.
- FreeGaussian analytically derive the **dynamic Gaussian flow constraints** via differential analysis with alpha composition, which draws the mathematical link among optical flow, camera motion, and dynamic Gaussian flow. The flow constraints refine Gaussian optimization enabling unsupervised interactive structure localization and the training of continuous Gaussian motion variations.
- Exploiting 3D Gaussian explicitness, we introduce a **3D spherical vector controlling scheme**, avoiding traditional complex 1D control variable calculations bypassing 3DGS trajectory as state representation, further simplifying and accelerating interactive Gaussian modeling.

2 Related Work

4D Novel View Synthesis. Neural Radiance Fields (NeRF) (Mildenhall et al. 2020) has innovated great progress in dynamic scene reconstruction. The existing methods can be categorized into three primary categories: time-varying methods (Du et al. 2021; Fang et al. 2022; Li et al. 2021; Park et al. 2021a; Pumarola et al. 2021; Tretschk et al. 2021; Yuan et al. 2021) that append temporal embeddings and scene-flow to the radiance MLP; deformable-canonical approaches (Gao et al. 2021; Li et al. 2022; Park et al. 2021b; Xian et al. 2021) warp query points from a dynamic space to a static canonical volume; and hybrid representations (Shao et al. 2023; Fridovich et al. 2023; Cao and Johnson 2023; Song et al. 2023) have accelerated training and rendering via time-space feature planes, dynamic voxels, or 4D hash encodings. More recently, 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) has gained prominence due to its superior training efficiency and real-time rendering. Subsequent 3DGS extensions for dynamic scenes learn dense Gaussian trajectories directly (Yang et al. 2023; Luiten et al. 2024), augmenting 3DGS with 4D feature planes (Wu et al. 2023) or learnable motion bases (Kratimenos, Lei, and Daniilidis 2023), and incorporating flow-based regularisation losses to enforce temporal consistency.

Controllable Scene Representation. Decoupling appearance, geometry, and time has unlocked controllable avatars (Rivero et al. 2024; Liu et al. 2023) and interactive simulators (Qu et al. 2024; Wang et al. 2024). CoNeRF (Kania et al. 2022) pioneered this effort by extending HyperNeRF (Park et al. 2021b) and regressing the attribute and the mask to enable few-shot attribute control. CoGS (Yu et al. 2023) leveraged 3D Gaussians to achieve real-time control of dynamic scenes without requiring explicit control signals. LiveScene (Qu et al. 2024) scales to scene level via factorized interactive space. But all these methods remain limited by dense manual annotations. More recently, MotionGS (Zhu et al. 2025) explores explicit motion priors to guide the deformation of 3D Gaussian.

3 Methodology

As depicted in Figure. 1, our approach exploits the underlying connections among dynamic Gaussian flow, optical flow, and camera motion to achieve annotation-free interactive scene reconstruction. The dynamic Gaussian flow autonomously segments interactable objects, forming the basis for downstream articulated object control. This enables trajectory-guided clustering and integrates with a 3D spherical vector control framework, resulting in a streamlined and scalable Gaussian modeling pipeline for dynamic scenes.

We first review 3DGS basics in Section. 3.1, then formulate the connection between optical flow, camera motion, and dynamic Gaussian flow in Section. 3.2. Based on this, we introduce a 3D spherical vector control scheme in Section. 3.3, which discovers and clusters dynamic Gaussians via trajectory analysis. The full pipeline is optimized with joint loss functions detailed in Section. 3.4.

3.1 Preliminary of 3DGS Rasterization

3D Gaussian Splatting (Kerbl et al. 2023) explicitly represents scenes with millions of Gaussians and emerges ultra high-quality rendering performance recently. Given a set of images capture with corresponding camera poses, 3DGS models scenes by learning a set of 3D Gaussians $\mathbf{G} = \{G_i : (\mathbf{X}_i, \mathbf{\Sigma}_i, \mathbf{o}_i, \mathbf{H}_i) | i = 1, ..., N\}$, where $\mathbf{X}_i \in \mathbb{R}^3$, $\mathbf{\Sigma}_i \in \mathbb{R}^{3\times3}$, $\mathbf{o}_i \in \mathbb{R}$, and $\mathbf{H}_i \in \mathbb{R}^{48}$ are the center position, 3D covariance, opacity, and spherical harmonics of the i-th Gaussian, respectively. With the rasterization pipeline, 3DGS projects $\mathbf{G}$ to image planes as 2D Gaussians $\mathbf{g} = \{g_i : (\boldsymbol{\mu}_i, \mathbf{\Sigma}'_i, \mathbf{o}_i, \mathbf{c}_i) | i = 1, ..., N\}$ and blender pixel colors $\hat{\mathbf{C}}$ via alpha composition:

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

where $\boldsymbol{\mu}_i \in \mathbb{R}^2$, $\mathbf{\Sigma}'_i \in \mathbb{R}^{2\times2}$, $\mathbf{c}_i \in \mathbb{R}^3$, $\alpha_i \in [0, 1]$ and $T_i \in [0, 1]$ are the 2d center, 2d covariance, color, alpha value and transmittance of 2D Gaussian g_i. The alpha value α_i at pixel coordinate $\mathbf{m}$ can be obtained by:

$$\alpha_i = \mathbf{o}_i \exp(-\frac{1}{2}(\mathbf{m} - \boldsymbol{\mu}_i)^T \mathbf{\Sigma}'^{-1}_i (\mathbf{m} - \boldsymbol{\mu}_i)). \tag{2}$$

With the supervision of observations, 3DGS optimizes parameters to minimize the photometric loss between rendered and ground-truth images.

3.2 Dynamic Gaussian Flow Analysis

Our insight is that the dynamic Gaussian flow under instantaneous motion can be analytically decoupled from optical flow and camera motion via differential analysis with alpha composition. Considering a dynamic scene with interactive

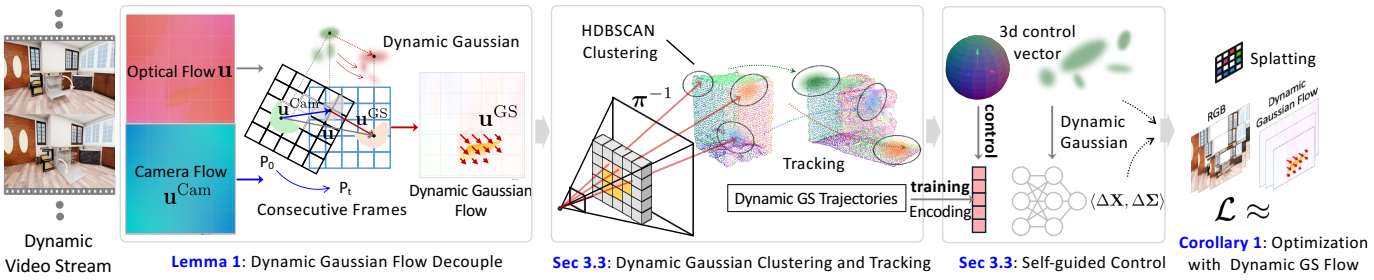

Figure 1: The overview of FreeGaussian. Given a set of video stream $\{\mathbf{P}(t), \mathbf{I}(t)\}$, our method recovers controllable 3D Gaussians $\mathbf{G}^*$ with two stages. First, we pre-train a deformable 3DGS and calculate dynamic Gaussian flow $\mathbf{u}^{\text{GS}}$ via Eq. (3). Then, we reproject dynamic Gaussian flow maps and cluster the active Gaussians with HDBSCAN algorithm, followed by trajectory calculation. In the controllable training stage, we optimize Gaussians $\mathbf{G}$ and network $\mathbf{\Theta}$ under the rasterisation loss in Eq. (7), which jointly aligns rendered images with input views and enforces consistency in the predicted dynamic flows.

objects as shown in Figure. 2, the camera and 3D Gaussians hold separate velocities in consecutive frames 0 and t. Assuming a dynamic 3D Gaussian G_i with velocity $\boldsymbol{v}^{\text{GS}}$, it is projected as image measurement g_i under the constant camera instantaneous motion by translation velocity $\boldsymbol{v}$ and rotational velocity $\boldsymbol{\omega}$. The optical flow $\mathbf{u}$ induced by $(\boldsymbol{v}, \boldsymbol{\omega})$ of a pixel $\mathbf{m} = (x, y)^\top$ can be obtained by *Lemma 1*:

Lemma 1: *Dynamic Gaussian flow* $\mathbf{u}^{GS}$ *under instantaneous motion can be derived from optical flow* $\mathbf{u}$ *and camera flow* $\mathbf{u}^{Cam}$ *with the following transform Eq.* (3).

$$\mathbf{u} = \mathbf{u}^{\text{Cam}} + \mathbf{u}^{\text{GS}} + \boldsymbol{\Delta}, \quad \mathbf{u}^{\text{Cam}} = \frac{\mathbf{A}\boldsymbol{v}}{Z} + \mathbf{B}\boldsymbol{\omega},$$

$$\begin{aligned} \mathbf{u}^{\text{GS}} &= \mathbf{A}\sum_{i=1}^{M} T_i \alpha_i \frac{\boldsymbol{v}^{\text{GS}}}{Z_i}, \boldsymbol{\Delta} = \mathbf{A}\sum_{i=1}^{M} T_i \alpha_i \boldsymbol{v}(\frac{1}{Z_i} - \frac{1}{Z}), \\ \mathbf{A} &= \begin{bmatrix} -f_x & 0 & x - c_x \\ 0 & -f_y & y - c_y \end{bmatrix}, \\ \mathbf{B} &= \begin{bmatrix} \frac{(x-c_x)(y-c_y)}{f_y} & -f_x - \frac{(x-c_x)^2}{f_x} & \frac{(y-c_y)f_x}{f_y} \\ f_y + \frac{(y-c_y)^2}{f_y} & -\frac{(x-c_x)(y-c_y)}{f_x} & -\frac{(x-c_x)f_y}{f_x} \end{bmatrix}. \end{aligned} \tag{3}$$

where f_x, f_y, c_x, c_y are camera intrinsics, M denotes the number of Gaussian projections sorted with Gaussian depth Z_i intersecting the pixel $\mathbf{m}$. Flow residual term $\boldsymbol{\Delta}$ are preserved to guarantee accuracy, even when it approaches zero after refined optimization.

The expression Eq. (3) elucidates the triadic relationship, yet Gaussian flow is not amenable to joint 3DGS training. For flexibility, we consider a pixel $\mathbf{m}_{i,t}$ following 2D Gaussian distribution g_i at time t, and obtain $\mathbf{m}_{i,t} \sim \mathcal{N}(\boldsymbol{\mu}_{i,t}, \boldsymbol{\Sigma}'_{i,t})$, with 2D mean $\boldsymbol{\mu}_{i,t}$ and covariance $\boldsymbol{\Sigma}'_{i,t} = \mathbf{B}_{i,t}\mathbf{B}_{i,t}^\top$. The following *Corollary* describes the dynamic Gaussian flow with 2D Gaussian means.

Corollary 1: *The dynamic Gaussian flow* $\tilde{\mathbf{u}}^{GS}$ *on image plane can be accumulated with 2D Gaussian means displacement* $\boldsymbol{\mu}_{i,t} - \boldsymbol{\mu}_{i,0}$.

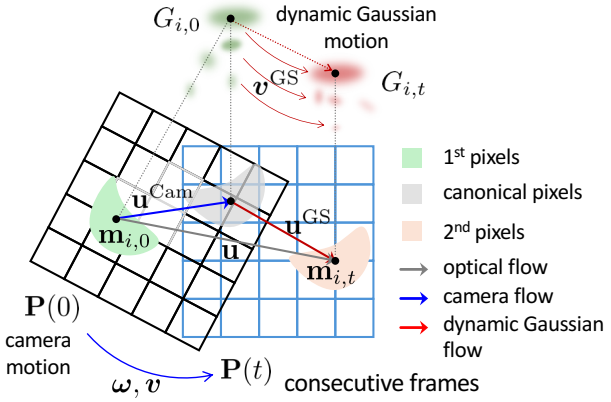

Figure 2: Dynamic Gaussian flow illustration. In interactive scenes, consider an instantaneous motion model, where the camera and 3D Gaussian hold separate velocities in consecutive frames. The projected optical flow $\mathbf{u}$ can be decomposed into camera flow $\mathbf{u}^{\text{Cam}}$ and dynamic Gaussian flow $\mathbf{u}^{\text{GS}}$, as described in Eqs. (3) and (4).

$$\begin{aligned} \mathbf{u} &= \mathbf{u}^{\text{Cam}} + \tilde{\mathbf{u}}^{\text{GS}} + \boldsymbol{\Delta}, \\ \tilde{\mathbf{u}}^{\text{GS}} &= \sum_{i=1}^{M} T_i \alpha_i (\boldsymbol{\mu}_{i,t} - \boldsymbol{\mu}_{i,0}). \end{aligned} \tag{4}$$

Discussion. The expression in Eqs. (3) and (4) reveals dynamic gaussian flow can be directly derived from 2D image flow $\mathbf{u}$ and camera-induced camera flow $\mathbf{u}^{\text{Cam}}$, accumulated with 2DGS projection displacement $\boldsymbol{\mu}_{i,t} - \boldsymbol{\mu}_{i,0}$. This naturally aligns with the 3D Gaussian rasterization pipeline, providing continuous motion constraints for dynamic Gaussian optimization. Besides, in static Gaussian scenes, the equation degenerates to camera flow with $\mathbf{u} = \mathbf{u}^{\text{Cam}}$. Hence, the resulting dynamic Gaussian flow map will highlight interactive 3D Gaussians, as illustrated in Figure. 3.
Compared with GaussianFlow (Gao et al. 2024), which lacks explicit camera motion modeling, and MotionGS (Zhu et al.

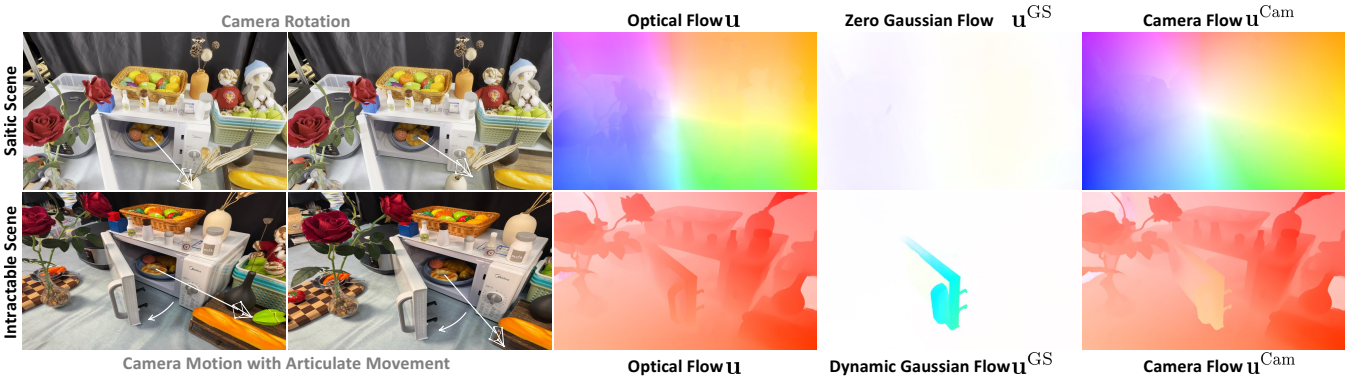

Figure 3: Illustration of dynamic Gaussian flow map under static and dynamic scenes. a) In static scenes with solely camera motion, Eq. (4) degenerate to pure camera flow, yielding zero dynamic Gaussian flow. b) In contract, when articulated object moves, the dynamic Gaussian flow map will highlight interactive 3D Gaussians.

2025), which relies on back-projection from known camera poses, our method is more general and flexible, benefiting from a principled formulation under instantaneous motion.

3.3 Self-guided Control with Dynamic 3DGS

Based on the discussion in Section. 3.2, dynamic Gaussian flow constraint Eq. (4) provides continuous Gaussian constraints and, critically, exposes the position of interactive areas, whose changing topological structures in dynamic scenes are reflected in varying Gaussian. To overcome the severe dependence on mask annotations in existing methods, we propose leveraging dynamic Gaussian flow to explore dynamic Gaussians of interactive objects and extract their trajectories for joint training.

Dynamic Gaussian clustering and tracking. With the formulations in Eq. (4), we first pretrain a deformable 3DGS $\mathbf{G}'$ with a set of camera streams. Then dynamic Gaussian flow $\mathbf{u}^{\mathrm{GS}}$ from Eq. (4) can be extracted frame-by-frame and binaried to obtain flow maps. By back-projecting the flow maps to identify dynamic 3D Gaussians, we highlight Gaussians $\mathcal{D} = \{g_i \mid i = 1, 2, \ldots, Q\}$ with sharp dynamics, as illustrated in Figure. 1. Next, we use unsupervised clustering algorithm **HDBSCAN** to group dynamic Gaussians into clusters $\mathcal{C} = \{c_i \mid i = 1, 2, \ldots, K\}$, where K is the number of interactive objects. The cluster centers move over time, generating continuous trajectories $\boldsymbol{\varsigma}(t, k)$, where k indexing which object the trajectory belongs to.

3D Spherical Vector Control. Prior works compress control signals into 1D vector, which introduces fundamental limitations: the 1D vector in CoGS (Yu et al. 2023) fails to capture complex Gaussian motions like rotations, while CoNeRF (Kania et al. 2022) requires the number of controllable regions and their corresponding signal ranges to be specified in advance. We overcome these limitations by representing the Gaussian states with 3D spherical vectors, which can be directly obtained from dynamic Gaussian tracking trajectory. This technique eliminates the requirement of control signals and curve fitting while increasing control flexibility.

Specifically, in the training stage, we represent the Gaussian dynamics state using cluster trajectory coordinates $\mathbf{v}_c^i = \boldsymbol{\varsigma}(t, k) - \boldsymbol{\varsigma}(0, k)$, concatenated with Gaussian centers $\mathbf{X}_i$. Then, we encode the coordinates with $\mathbf{E}(\mathbf{v}_c^i, \mathbf{X}_i)$ and jointly train the model Θ to recover Gaussian dynamics $\langle \Delta\mathbf{X}_i, \Delta\boldsymbol{\Sigma}_i \rangle$:

$$\boldsymbol{f}_\Theta \left(\mathbf{E}(\mathbf{v}_c^i, \mathbf{X}_i) \right) \mapsto \langle \Delta\mathbf{X}_i, \Delta\boldsymbol{\Sigma}_i \rangle . \tag{5}$$

After that, we perform splatting rasterization in Eq. (1) with the Gaussian combining with predicted dynamics. During the control stage, we manually input interactive 3D vector $\mathbf{v}_c'$, which is mapped to the nearest point in the original trajectory, to retrive the Gaussian dynamics from the network through $\boldsymbol{f}_\Theta \left(\mathbf{E}(\mathbf{v}_c', \mathbf{X}_i) \right)$.

3.4 Loss Functions

Loss with dynamic Gaussian flow. The expression in Eq. (4) suggests that incorporating optical flow and camera flow prior to the loss function can improve 3DGS optimization and maintain dynamic Gaussian smooth transitions between frames. Hence, we propose a dynamic Gaussian flow loss $\mathcal{L}_{\text{uGS}}$ to optimize the dynamic Gaussian field $\mathbf{G}$ and network $\boldsymbol{\Theta}$ with the following formulation:

$$\mathcal{L}_{\text{uGS}} = \left\| \mathbf{u} - \mathbf{u}^{\text{Cam}} - \sum_{i=1}^{M} T_i \alpha_i (\boldsymbol{\mu}_{i,t} - \boldsymbol{\mu}_{i,0}) \right\|^2 , \tag{6}$$

where $\mathbf{u}$ and $\mathbf{u}^{\text{Cam}}$ can be calculated with optical flow estimator (Contributors 2021) and Eq. (4), respectively. Dynamic Gaussians $\mathbf{G}$ and $\boldsymbol{\Theta}$ are optimized via the proposed dynamic gaussian flow supervision $\mathcal{L}_{\text{uGS}}$ in Eq. (6) with the fundamental per-frame photometric supervision $\mathcal{L}_{\text{RGB}}$, and $\mathcal{L}_{\text{D-SSIM}}$. The loss function for FreeGaussian optimization can be formulated as:

$$\mathcal{L} = \lambda \mathcal{L}_{\text{RGB}} + (1 - \lambda) \mathcal{L}_{\text{D-SSIM}} + \beta \mathcal{L}_{\text{uGS}}. \tag{7}$$

4 Experiment

4.1 Experimental Setup

Datasets. We benchmark FreeGaussian on three publicly-available datasets. We adopt CoNeRF dataset (Kania et al.

Method	CoNeRF Synthetic			CoNeRF Controllable			GT	InterReal #Medium			InterReal #Challenging			InterReal #Avg		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
HyperNeRF (Park et al. 2021b)	25.963	0.854	0.158	32.520	0.981	0.169	✗	25.283	0.671	0.467	25.261	0.713	0.517	25.277	0.682	0.480
K-Planes (Fridovich et al. 2023)	33.301	0.933	0.150	31.811	0.912	0.262	✗	27.999	0.813	0.177	26.427	0.756	0.331	27.606	0.799	0.215
CoNeRF (Kania et al. 2022)	32.394	0.972	0.139	32.342	0.981	0.168	✓	27.501	0.745	0.367	26.447	0.734	0.472	27.237	0.742	0.393
CoGS (Yu et al. 2023)	33.455	0.960	0.064	32.601	**0.983**	**0.164**	✓	30.774	0.913	0.100	—	—	—	**30.774**	0.913	0.100
LiveScene (Qu et al. 2024)	43.349	0.986	**0.011**	32.782	0.932	0.186	✓	30.815	0.911	**0.066**	28.436	0.846	0.185	30.220	0.895	0.096
MotionGS (Zhu et al. 2025)	35.057	0.981	0.052	28.363	0.882	0.273	✗	29.193	0.903	0.105	—	—	—	29.193	0.903	0.105
FreeGaussian (Ours)	**43.939**	**0.993**	**0.011**	**33.247**	0.941	0.218	✗	**31.310**	**0.938**	0.072	**29.133**	**0.899**	**0.161**	30.765	**0.928**	**0.094**

Table 1: Quantitative results on CoNeRF and InterReal datasets. FreeGaussian ranks first on CoNeRF synthetic scene and outperforms all competing methods across various settings on InterReal datasets.

Method	Type	GT	#Easy Sets				#Medium Sets				#Avg (all 20 sets)			
			M-PSNR↑	PSNR↑	SSIM↑	LPIPS↓	M-PSNR↑	PSNR↑	SSIM↑	LPIPS↓	M-PSNR↑	PSNR↑	SSIM↑	LPIPS↓
HyperNeRF (Park et al. 2021b)	4D-NeRF	✗	20.870	30.708	0.908	0.316	22.093	31.621	0.936	0.265	21.679	30.748	0.917	0.299
K-Planes (Fridovich et al. 2023)	4D-NeRF	✗	24.211	32.841	0.952	0.093	24.312	32.548	0.954	0.100	24.810	32.573	0.952	0.097
CoNeRF (Kania et al. 2022)	Con-NeRF	✓	26.561	32.104	0.932	0.254	27.716	33.256	0.951	0.207	27.013	32.477	0.939	0.234
MK-Planes*	Con-NeRF	✓	23.509	31.630	0.948	0.098	25.860	31.880	0.951	0.104	24.561	31.477	0.946	0.106
MK-Planes	Con-NeRF	✓	23.872	31.677	0.948	0.098	25.217	32.165	0.952	0.099	24.743	31.751	0.949	0.099
CoGS (Yu et al. 2023)	Con-GS	✓	25.208	32.315	0.961	0.108	26.332	32.447	0.965	0.086	26.103	32.187	0.963	0.097
LiveScene (Qu et al. 2024)	Con-NeRF	✓	26.680	**33.221**	0.962	**0.072**	27.985	33.262	0.965	0.072	27.310	33.158	0.962	0.072
MotionGS (Zhu et al. 2025)	Flow-GS	✗	26.306	31.907	0.961	0.111	25.391	30.904	0.969	0.083	25.706	31.282	0.926	0.100
FreeGaussian (Ours)	Flow-GS	✗	**27.655**	33.205	**0.967**	**0.072**	**28.281**	**33.922**	**0.972**	**0.071**	**27.838**	**33.249**	**0.969**	**0.071**

Table 2: Quantitative results on OmniSim Dataset. FreeGaussian surpasses prior works on nearly all metrics. "Con-*" indicates Controllable methods, "GT" refers to control signals and M-PSNR denotes mask-weighted PSNR for dynamic region.

2022) for single-object evaluation and OmniSim and InterReal datasets (Qu et al. 2024) for multiple-object setting. A self-captured toy-kitchen sequence is included for visualization. Throughout all experiments the training pipeline remains entirely **NO Ground Truth** for control signals.

Baselines. Comparison spans three distinct techniques, including 3D deformable methods (Fridovich et al. 2023; Park et al. 2021b), controllable scene reconstruction methods (Kania et al. 2022; Yu et al. 2023; Qu et al. 2024) and flow-based controllable method (Zhu et al. 2025).

Implementation details. FreeGaussian is built on 4DGS (Yang et al. 2023). We use RAFT (Teed and Deng 2020) for optical flow prediction and perform HDBSCAN clustering for dynamic Gaussian flow with Euclidean metric. Training proceeds for 60k steps on a single RTX 4090 with the Adam optimizer at learning rate $1.6e^{-4}$ in roughly 30 minutes: 30k steps of deformable pre-training followed by 30k steps of flow training.

4.2 Evaluation of Novel View Synthesis

Results on CoNeRF Datasets. The quantitative results of our approach on the CoNeRF Synthetic and Controllable scenes are presented in Table. 1. Notably, our method surpasses all existing approaches in terms of PSNR, SSIM, and LPIPS metrics on CoNeRF Synthetic scenes. Furthermore, on CoNeRF Controllable scenes, our method attains the highest PSNR of 33.247, while demonstrating comparable SSIM and LPIPS scores to the SOTA methods. These results underscore the success of the guidance-free paradigm. Figure. 4 visualizes the rendering result of our method on the CoNeRF dataset. Our method handles the controllable objects well and retains the details of the moving area, demonstrating its effectiveness in modeling interactive scenes.

Metric on LiveScene Dataset. As reported in Tables. 1 and 2, FreeGaussian leads across both OmniSim and InterReal while remaining fully annotation-free. On OmniSim, it achieves the highest scores on `#medium` subset, surpassing sparse-label baselines (Zhu et al. 2025) by nearly 2dB in PSNR. Although PSNR is slightly inferior to the dense-labele LiveScene on the `#easy` subset, its advantage is decisive whenever manual labels are unavailable. M-PSNR metric further confirms superior reconstruction qaulity of dynamic regions. On InterReal, CoGS and MotionGS underperform on `#medium` and collapses on the `#challenging` subset, where prolonged trajectories and dense interaction expose the limits of prior controllable or flow-based methods. FreeGaussian not only converges robustly but also posts the best `#challenging` results and the top `#medium` PSNR and SSIM, demonstrating robust fidelity and stability in large-scale, real-world interactive scenarios with incomplete supervision.

Individual Object Control Visualiztion. Figure. 7 presents a example to demonstrate case of per-object control. During manipulation, each object is assigned an independent 3D spherical vector which controls its instantaneous motion. This disentanglement removes cross-object constraints, and allows the model to compose attribute combinations absent from the training set. The example demonstrates a sequence where two cabinets always open or close together. By independently setting their control vectors, we generate a configuration in which the top cabinet is open while the bottom cabinet remains closed (top-right), confirming that the model can extrapolate novel scene arrangements with both diversity and fidelity.

Flow Decouple Visualization. Figure. 5 contrasts the flow-decomposition quality of FreeGaussian and MotionGS on

Figure 4: View Synthesis Visualization on CoNeRF Dataset. In comparison with other methods, FreeGaussian achieves more realistic and detailed rendering quality, whereas other methods suffer from ghosting artifacts.

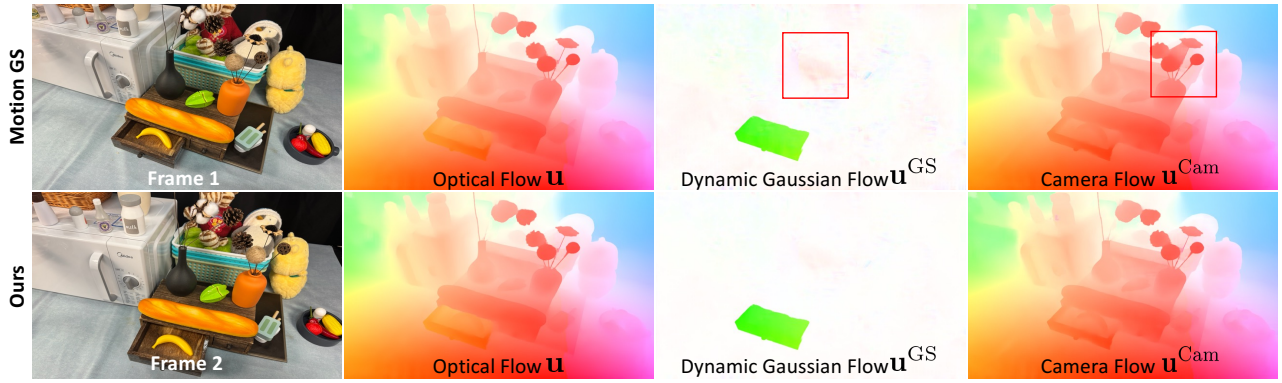

Figure 5: Flow Decoupling Comparison. FreeGaussian (row 2) cleanly separates camera egomotion from the microwave's self-motion, producing artifact-free dynamic Gaussian flow.

real-life toy-kitchen scene, in which an automatically opening drawer and a moving camera jointly generate complex optical flow. Although both methods estimate the optical flow accurately, MotionGS decouples camera flow that is both noisy and partially aliased (top-right), thus the residual dynamic gaussian flow inherits substantial artifacts. In contrast, FreeGaussian cleanly disentangles the camera-induced flow from the drawer's independent motion, yielding a per-object component that is significantly cleaner. This precise separation supplies downstream constraints with more reliable guidance, thus improving the render fidelity.

Setting		PSNR↑	SSIM↑	LPIPS↓
Sim	FreeGaussian	**35.31**	**0.975**	**0.062**
	#1.HDBSCAN → KMeans	31.33	0.966	0.065
	#2.HDBSCAN → MeanShift	30.95	0.959	0.068
	#3.3D Vector → 1D Vector	33.22	0.969	0.064
Real	FreeGaussian	**32.45**	**0.951**	0.092
	#4.HDBSCAN → KMeans	32.33	0.949	**0.091**
	#5.HDBSCAN → MeanShift	31.86	0.932	0.100
	#6.3D Vector → 1D Vector	30.33	0.918	0.107

Table 3: Ablation Study. Ablations on two components of our proposed method.

4.3 Ablation and Analysis

We conduct ablation studies to examine the contribution of two components in FreeGaussian. Following previous work (Qu et al. 2024), we select three representative subsets from the OmniSim dataset: `#seq001`, `#seq004`, and `#seq0015` and a self-captured toy-kitchen dataset. Table. 3 shows the results of each ablation experiment.

Effectiveness of 3D Vector Control. we conduct ablation using directly 1D vector adopted by CoGS while keeping all other settings identical. As shown in the Table. 3 (#3, #6), this change degrades rendering quality since PCA only approximates the dominant direction, leaving detailed trajectories misaligned, shown in the middle of Figure. 8. Consequently, the model reconstructs coarse structures in the control stage. In contrast, 3D vector provides per-Gaussian clusters, fine-grained control (right); the explicit motion cues tightly constrain the Gaussian flow, ensuring consistent motion guidance between training and controlling.

Effectiveness of HDBSCAN Clustering. Clustering is essential in the control stage as the number of controllable objects is not a prior of our approach.Compared with widely used clustering methods like KMeans, HDBSCAN is more robust to noise with outliner handling and more flexible

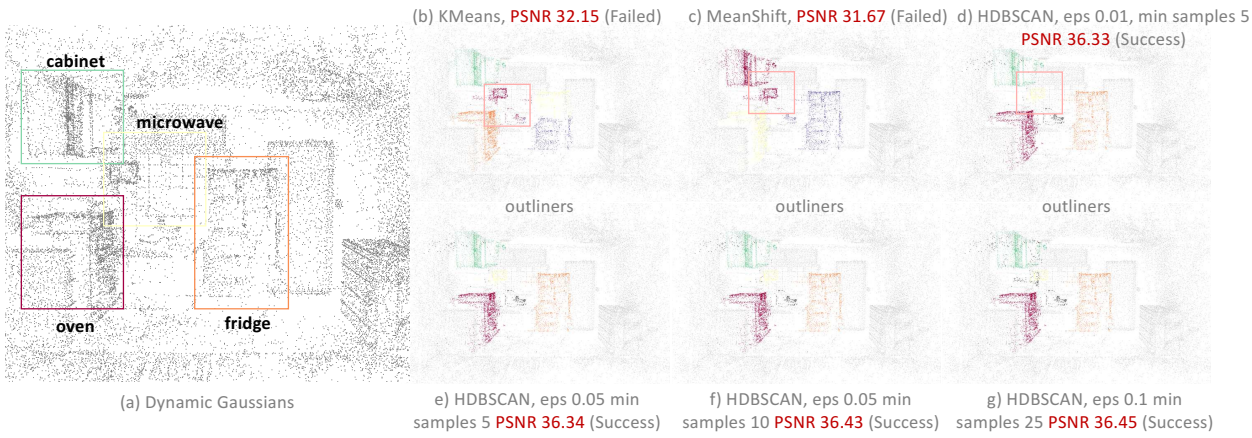

Figure 6: Ablation of clustering results among KMeans, MeanShift and HDBSCAN on #seq001 of OmniSim.

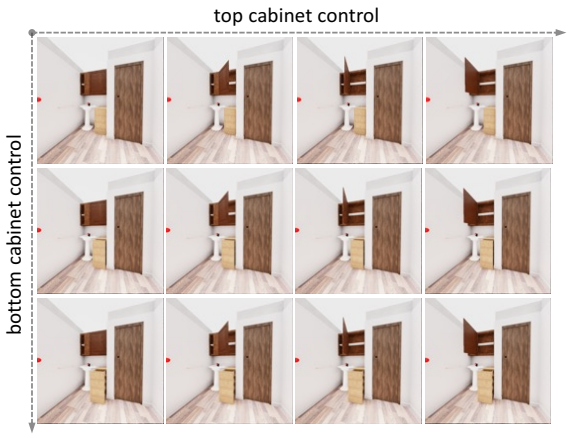

Figure 7: Individual Object Control. Our method supports per-object manipulation, enabling the synthesis of previously unseen views for each scene without retraining.

without predefined cluster numbers. Besides, MeanShift may converge to local optima depending on the cluster landscape and initial window locations. Figure. 6 illustrates the remarkable stability and accuracy of HDBSCAN. (d)-(g) show that, HDBSCAN delineates object geometry cleanly and isolates outliers, whereas K-Means introduces a large number of noisy points (b) and Meanshift yields an inappropriate cluster cardinality (c). For real life scene clustering visualizations, please refer to the supplementary materials.

5 Conclusion

In this work, we derive a mathematical link among optical flow, camera flow, and dynamic Gaussian flow with differential analysis, yielding an annotation-free Gaussian-splatting pipeline for controllable view synthesis. Flow-based constraints refine optimization, ensuring smooth motion, high fidelity, and automatically highlighting interactable Gaussians. After obtaining each interactable object, a 3D spherical vector encodes its state, eliminating explicit trajectory computation. Extensive experiments show superior performance in view synthesis and scene controlling, enabling more accurate and efficient modeling of articulated objects.

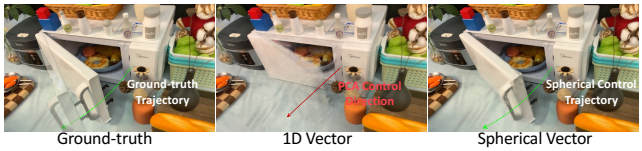

Figure 8: Ablation of 3D spherical vector. 1D vector PCA could not match arc trajectory, while 3D spherical vectors recover fine structure and motion.

Acknowledgements

This work is supported by the Shanghai AI Laboratory.

References

Cao, A.; and Johnson, J. 2023. HexPlane: A Fast Representation for Dynamic Scenes. *CVPR*.

Contributors, M. 2021. MMFlow: OpenMMLab Optical Flow Toolbox and Benchmark. https://github.com/open-mmlab/mmflow.

Du, Y.; Zhang, Y.; Yu, H.-X.; Tenenbaum, J. B.; and Wu, J. 2021. Neural radiance flow for 4d view synthesis and video processing. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 14304–14314. IEEE Computer Society.

Fang, J.; Yi, T.; Wang, X.; Xie, L.; Zhang, X.; Liu, W.; Nießner, M.; and Tian, Q. 2022. Fast dynamic radiance fields with time-aware neural voxels. In *SIGGRAPH Asia 2022 Conference Papers*, 1–9.

Fridovich, S.; Meanti, G.; Warburg, F. R.; Recht, B.; and Kanazawa, A. 2023. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12479–12488.

Gao, C.; Saraf, A.; Kopf, J.; and Huang, J.-B. 2021. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5712–5721.

Gao, Q.; Xu, Q.; Cao, Z.; Mildenhall, B.; Ma, W.; Chen, L.; Tang, D.; and Neumann, U. 2024. Gaussianflow: Splatting gaussian dynamics for 4d content creation. *arXiv preprint arXiv:2403.12365*.

Gao, Y.; Li, C.; You, Z.; Liu, J.; Li, Z.; Chen, P.; Chen, Q.; Tang, Z.; Wang, L.; Yang, P.; Tang, Y.; Tang, Y.; Liang, S.; Zhu, S.; Xiong, Z.; Su, Y.; Ye, X.; Li, J.; Ding, Y.; Wang, D.; Wang, Z.; Zhao, B.; and Li, X. 2025. OpenFly: A Comprehensive Platform for Aerial Vision-Language Navigation. arXiv:2502.18041.

Huang, S.; Shen, G.; Kang, Y.; and Song, Y. 2025. Immersive Augmented Reality Music Interaction through Spatial Scene Understanding and Hand Gesture Recognition. *Preprints*.

Kang, Y.; Song, Y.; and Huang, S. 2024. Revolutionizing Engagement: The Fusion of Personalization and Augmented Reality. *Preprints*.

Kania, K.; Yi, K. M.; Kowalski, M.; Trzciński, T.; and Tagliasacchi, A. 2022. Conerf: Controllable neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18623–18632.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4): 139–1.

Kratimenos, A.; Lei, J.; and Daniilidis, K. 2023. DynMF: Neural Motion Factorization for Real-time Dynamic View Synthesis with 3D Gaussian Splatting. *arXiV*.

Li, T.; Slavcheva, M.; Zollhoefer, M.; Green, S.; Lassner, C.; Kim, C.; Schmidt, T.; Lovegrove, S.; Goesele, M.; Newcombe, R.; et al. 2022. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5521–5531.

Li, Z.; Niklaus, S.; Snavely, N.; and Wang, O. 2021. 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*, 6498–6508.

Liao, J. Z. Z. L. J.; Cao, Y.-P.; and Shan, Y. 2024. Advances in 3D Generation: A Survey. *arXiv preprint arXiv:2401.17807*.

Liu, X.; Zhan, X.; Tang, J.; Shan, Y.; Zeng, G.; Lin, D.; Liu, X.; and Liu, Z. 2023. HumanGaussian: Text-Driven 3D Human Generation with Gaussian Splatting. *arXiv preprint arXiv:2311.17061*.

Luiten, J.; Kopanas, G.; Leibe, B.; and Ramanan, D. 2024. Dynamic 3D Gaussians: Tracking by Persistent Dynamic View Synthesis. In *3DV*.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Park, K.; Sinha, U.; Barron, J. T.; Bouaziz, S.; Goldman, D. B.; Seitz, S. M.; and Martin-Brualla, R. 2021a. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5865–5874.

Park, K.; Sinha, U.; Hedman, P.; Barron, J. T.; Bouaziz, S.; Goldman, D. B.; Martin-Brualla, R.; and Seitz, S. M. 2021b. HyperNeRF: A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields. *ACM Trans. Graph.*, 40(6).

Pumarola, A.; Corona, E.; Pons-Moll, G.; and Moreno-Noguer, F. 2021. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10318–10327.

Qu, D.; Chen, Q.; Zhang, P.; Gao, X.; Zhao, B.; Wang, D.; and Li, X. 2024. LiveScene: Language Embedding Interactive Radiance Fields for Physical Scene Rendering and Control. *ArXiv*, abs/2406.16038.

Qu, D.; Song, H.; Chen, Q.; Chen, Z.; Gao, X.; Ye, X.; Lv, Q.; Shi, M.; Ren, G.; Ruan, C.; et al. 2025a. EO-1: Interleaved Vision-Text-Action Pretraining for General Robot Control. *arXiv preprint arXiv:2508.21112*.

Qu, D.; Song, H.; Chen, Q.; Yao, Y.; Ye, X.; Ding, Y.; Wang, Z.; Gu, J.; Zhao, B.; Wang, D.; et al. 2025b. SpatialVLA: Exploring Spatial Representations for Visual-Language-Action Model. *arXiv preprint arXiv:2501.15830*.

Rivero, A.; Athar, S.; Shu, Z.; and Samaras, D. 2024. Rig3DGS: Creating Controllable Portraits from Casual Monocular Videos. *ArXiv*, abs/2402.03723.

Shao, R.; Zheng, Z.; Tu, H.; Liu, B.; Zhang, H.; and Liu, Y. 2023. Tensor4D: Efficient Neural 4D Decomposition for High-fidelity Dynamic Reconstruction and Rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Song, H.; Qu, D.; Yao, Y.; Chen, Q.; Lv, Q.; Tang, Y.; Shi, M.; Ren, G.; Yao, M.; Zhao, B.; et al. 2025. Hume: Introducing System-2 Thinking in Visual-Language-Action Model. *arXiv preprint arXiv:2505.21432*.

Song, L.; Chen, A.; Li, Z.; Chen, Z.; Chen, L.; Yuan, J.; Xu, Y.; and Geiger, A. 2023. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5): 2732–2742.

Steuer, J. 1992. Defining virtual reality: dimensions determining telepresence.

Tang, J.; Ren, J.; Zhou, H.; Liu, Z.; and Zeng, G. 2023. DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation. *arXiv preprint arXiv:2309.16653*.

Teed, Z.; and Deng, J. 2020. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*.

Tretschk, E.; Tewari, A.; Golyanik, V.; Zollhöfer, M.; Lassner, C.; and Theobalt, C. 2021. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 12959–12970.

Waisberg, E.; Ong, J.; Masalkhi, M.; Zaman, N.; Sarker, P.; Lee, A. G.; and Tavakkoli, A. 2023. The future of ophthalmology and vision science with the Apple Vision Pro. *Eye*, 38: 242–243.

Wang, G.; Pan, L.; Peng, S.; Liu, S.; Xu, C.; Miao, Y.; Zhan, W.; Tomizuka, M.; Pollefeys, M.; and Wang, H. 2024. NeRF in Robotics: A Survey. *ArXiv*, abs/2405.01333.

Wu, G.; Yi, T.; Fang, J.; Xie, L.; Zhang, X.; Wei, W.; Liu, W.; Tian, Q.; and Xinggang, W. 2023. 4D Gaussian Splatting for Real-Time Dynamic Scene Rendering. *arXiv preprint arXiv:2310.08528*.

Xian, W.; Huang, J.-B.; Kopf, J.; and Kim, C. 2021. Spacetime neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9421–9431.

Yang, Z.; Gao, X.; Zhou, W.; Jiao, S.; Zhang, Y.; and Jin, X. 2023. Deformable 3D Gaussians for High-Fidelity Monocular Dynamic Scene Reconstruction. *arXiv preprint arXiv:2309.13101*.

Yu, H.; Julin, J.; Milacski, Z. Á.; Niinuma, K.; and Jeni, L. A. 2023. Cogs: Controllable gaussian splatting. *arXiv preprint arXiv:2312.05664*.

Yuan, W.; Lv, Z.; Schmidt, T.; and Lovegrove, S. 2021. Star: Self-supervised tracking and reconstruction of rigid objects in motion with neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13144–13152.

Zeng, S.; Qi, D.; Chang, X.; Xiong, F.; Xie, S.; Wu, X.; Liang, S.; Xu, M.; and Wei, X. 2025. JanusVLN: Decoupling Semantics and Spatiality with Dual Implicit Memory for Vision-Language Navigation. *arXiv preprint arXiv:2509.22548*.

Zhu, R.; Liang, Y.; Chang, H.; Deng, J.; Lu, J.; Yang, W.; Zhang, T.; and Zhang, Y. 2025. Motiongs: Exploring explicit motion guidance for deformable 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 37: 101790–101817.