

MaGS: Reconstructing and Simulating Dynamic 3D Objects with Mesh-adsorbed Gaussian Splatting

Shaojie Ma¹ Yawei Luo^{1†} Wei Yang² Yi Yang¹

¹ Zhejiang University, China ² Huazhong University of Science and Technology, China

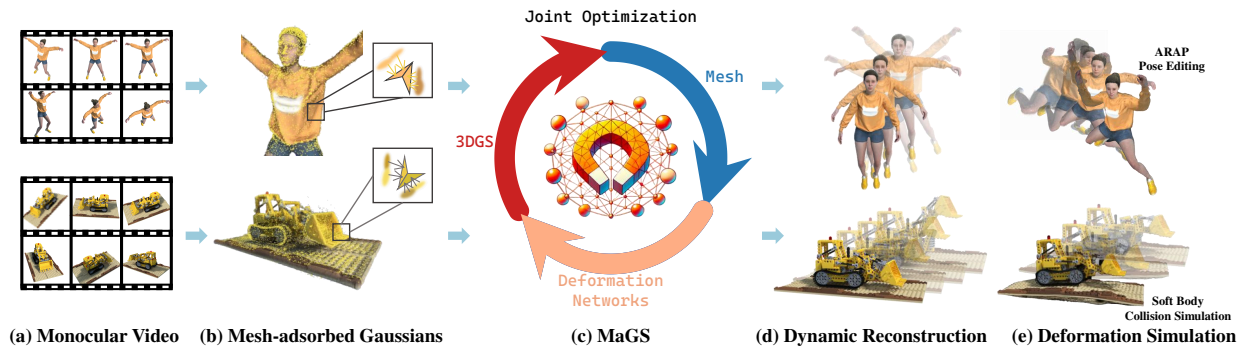


Figure 1. We propose the Mesh-adsorbed Gaussian Splatting (MaGS), a unified framework for reconstructing and simulating a dynamic 3D object from a monocular video.

Abstract

3D reconstruction and simulation, although interrelated, have distinct objectives: reconstruction requires a flexible 3D representation that can adapt to diverse scenes, while simulation needs a structured representation to model motion principles effectively. This paper introduces the *Mesh-adsorbed Gaussian Splatting (MaGS)* method to address this challenge. MaGS constrains 3D Gaussians to roam near the mesh, creating a mutually adsorbed mesh-Gaussian 3D representation. Such representation harnesses both the rendering flexibility of 3D Gaussians and the structured property of meshes. To achieve this, we introduce *RMD-Net*, a network that learns motion priors from video data to refine mesh deformations, alongside *RGD-Net*, which models the relative displacement between the mesh and Gaussians to enhance rendering fidelity under mesh constraints. To generalize to novel, user-defined deformations beyond input video without reliance on temporal data, we propose *MPE-Net*, which leverages inherent mesh information to bootstrap *RMD-Net* and *RGD-Net*. Due to the universality of meshes, MaGS is compatible with various deformation priors such as ARAP, SMPL, and soft physics simulation. Extensive experiments on the *D-NeRF*, *DG-Mesh*, and *PeopleSnapshot* datasets demonstrate that MaGS achieves state-of-the-art performance in both reconstruction and simulation.

[†]Corresponding Author

Project page: <https://cwcac.github.io/MaGS-page/>

1. Introduction

The human visual system can simultaneously capture 3D appearance (reconstruction) and infer dynamic objects' probable motions (simulation) from a monocular video. In contrast, computer vision and graphics typically treat 3D reconstruction and simulation as distinct tasks. Over the years, various reconstruction methods have emerged [12, 24, 35, 46]. These methods represent the geometry and appearance of a 3D scene from multi-view images, either implicitly or explicitly, and subsequently render photo-realistic novel views based on the 3D representation. Neural Radiance Fields (NeRF) [35], an implicit method, utilizes volume rendering techniques to bridge 2D and 3D spaces. Explicit methods concretely represent 3D scenes using Point Clouds, Meshes, and Voxels. Hybrid approaches like PointRF [67] and Pulsar [25] have been developed, effectively combining explicit and implicit representations. 3D Gaussian Splatting (3DGS) has demonstrated significant improvements in rendering quality and speed [5, 24], and has been extended to 4D scene reconstruction [57, 62, 63], recent works further explore 4D scene generation with higher controllability and realism [32–34], enabling potential integration into applications such as video, gaming, and even smart education systems [31].

Simulation involves motion editing [21, 23, 39, 61], physical simulation [14, 43, 58], and other applications [10, 42, 54, 66] on reconstructed results. Implicit reconstruction methods lack surface, skeletal, or physical priors, limiting realistic simulation. Explicit methods, though struc-

tured, struggle with rendering quality and reconstruction speed. Hybrid approaches [22, 61, 64, 70] integrate the strengths of both paradigms to enable simultaneous reconstruction and simulation capabilities. For instance, Huang *et al.* [19] employ sparse control points to guide deformation in a more organized fashion. Furthermore, multiple approaches [14, 16, 52] integrate mesh with 3DGS to enable natural simulations of static objects. Dynamic scenes demand implicit expressiveness for rendering and explicit priors for simulation. DG-Mesh [27] extends mesh with 3DGS for reconstruction. DynaSurfGS [2] and Dynamic2DGS [69] improve the 3DGS representation and can export better meshes of dynamic scenes.

3D reconstruction and simulation, while interrelated, have distinct objectives: reconstruction requires a flexible 3D representation adaptable to diverse scenes, whereas simulation necessitates a structured representation to model deformation policies effectively. This dual requirement poses significant challenges for a unified framework. In this paper, we propose the Mesh-adsorbed Gaussian Splatting (MaGS) method to address this. MaGS constrains 3D Gaussians to roam near the mesh surface, creating a hybrid mesh-Gaussian 3D representation that combines the rendering flexibility of 3D Gaussians with the adaptability of meshes to different geometric priors. To realize this representation, we introduce a learnable Relative Mesh Deformation Network (RMD-Net) that learns motion principles from video data to refine mesh deformations, and a learnable Relative Gaussian Deformation Network (RGD-Net) to model the relative displacement between the mesh and 3D Gaussians to enhance rendering fidelity under mesh constraints. Unlike conventional anchored, fixed mesh-Gaussian methods [16, 27, 52], MaGS allows Gaussians relative displacement through the RGD-Net, effectively bypassing the trade-off between rendering accuracy and deformation rationality during dynamic object reconstruction.

In the simulation, the reconstructed object can undergo customized deformations beyond the video. In this process, MaGS reuses the motion principles learned in RMD-Net to refine customized simulation deformations, ensuring object deformations better conform to physical laws. Additionally, MaGS reuses the mesh-Gaussian relational priors learned in RGD-Net to accurately compute the relative displacement of Gaussians on the customized deformed mesh for better rendering. Moreover, MaGS eliminates dependence on temporal information using Mesh Pose Embedding Network (MPE-Net), which guides RMD-Net & RGD-Net based on mesh inherent information rather than video timestamps. This strategy allows MaGS to generalize effectively to novel deformations beyond the input video. Due to the universality of meshes, MaGS is compatible with mesh-based simulation methods, such as ARAP [47], SMPL [29], and soft physics simulation [49], enabling it to handle com-

plex deformations. Through joint optimization of meshes, 3D Gaussians, and networks, MaGS achieves both high rendering accuracy and realistic deformation. Extensive experiments on the D-NeRF, DG-Mesh, and PeopleSnapshot datasets demonstrate that MaGS surpasses SOTA methods, establishing a new paradigm for unified reconstruction and simulation tasks.

2. Related Work

2.1. Differentiable Rendering for Dynamic Scenes

Since the introduction of NeRF [35], advancements in 3D scene reconstruction have progressed rapidly. Researchers have extended NeRF with temporal encoding to address dynamic scenes [13, 40, 50]. Other approaches have optimized NeRF’s temporal modeling, including using voxel grids for faster training [9, 28] and k -plane representations to improve efficiency [3, 11, 17, 44]. Park *et al.* [38, 39] incorporated geometric priors and hyperspace projections to enhance the interpretability of deformation fields, while Yan *et al.* [60] improved accuracy by modeling specular reflections. More recently, 3DGS [24] has gained increasing attention. It significantly improves rendering speed compared to NeRF while providing a more explicit geometric interpretation. Dynamic field has also been applied to 3DGS, discussing the use of deformation fields to represent dynamic scenes [23, 57, 62]. Some studies have applied 3DGS to dynamic scenes of the human body [26, 41, 45].

2.2. Differentiable Rendering for Mesh Reconstruction

Meshes are a key 3D representation in animation, gaming, autonomous driving, and digital twins. Extracting high-quality meshes from neural rendering is a growing research area. Early methods used Marching Cubes [30] with depth data. NeuS [55] integrates Signed Distance Functions with neural rendering for better meshes. NeRF2Mesh [48] uses an adaptive algorithm for improved accuracy. SuGaR [16] *bind* Gaussians to mesh for optimizing mesh through the position of Gaussians. 2DGS [18] uses flat Gaussians to improve boundary accuracy, enhancing the quality of mesh extraction. PGSR [4] applies Planar-based Gaussian Splatting for high-quality meshes. DynaSurfGS [2] extends this to dynamic scenes. DG-Mesh [27] maps Gaussians to mesh facets with *anchoring* for optimization.

2.3. Differentiable Rendering with Explicit Representations

Recent studies have advanced deformation techniques by integrating explicit priors into neural rendering. NeRF-Editing [64] integrates NeRF with mesh-based deformation using ARAP [47] to simulate deformations, while NeuMesh [61] embeds neural fields onto mesh for improved

simulation. SuGaR [16] *bind* Gaussians to static mesh, enabling simultaneous optimization and stable simulations. GaMeS [52] introduces pseudo-mesh and designs a mesh-Gaussians *binding* algorithm for deformation editing. Similarly, Mani-GS [14] anchors Gaussian points to meshes, exploiting mesh structures to guide Gaussian deformations. For human body simulation, approaches like [7, 36, 41] combine 3DGS with parametric models. SC-GS [19] and SP-GS [53] employs sparse control points for editing Gaussians. D-MiSo [51] extends GaMeS to dynamic scenes for physical simulation.

3. Preliminaries

3.1. 3D Gaussian Splatting

3D Gaussian Splatting [24] employs learnable 3D Gaussians to map spatial coordinates to pixel values, improving rendering quality and efficiency. Each Gaussian is defined by a mean $\mu \in \mathbb{R}^3$, covariance $\Sigma \in \mathbb{R}^{3 \times 3}$, opacity σ , scaling $s \in \mathbb{R}^3$, rotation $q \in \mathbb{R}^4$, and spherical harmonics $sh \in \mathbb{R}^L$ (where L depends on the model). The Gaussian function measures spatial influence, with Σ constructed from scaling S and rotation R derived from s and q , respectively. Opacity σ controls visibility, and sh enables view-dependent shading.

For rendering, 3D Gaussians are projected onto a 2D image plane. The 3D covariance Σ is transformed into a 2D covariance Σ' using the view transformation W and projection Jacobian J . The color at a pixel u , denoted $C(u)$, is computed by blending contributions from overlapping Gaussians, weighted by their transmittance T_i and opacity α_i , where T_i accounts for occlusion by preceding Gaussians, and α_i combines opacity with the projected Gaussian distribution. This approach efficiently balances detail and performance in 3D rendering.

3.2. Dynamic 3D Gaussian Splatting

Modeling dynamic scenes requires handling temporal variations in Gaussian parameters. To model dynamic scenes, recent methods [2, 19, 62] use 3D Gaussians that adapt over time through deformation fields. A network \mathbb{D} predicts temporal updates to Gaussian parameters from an initial frame ($t = 0$) to subsequent frames. Given a Gaussian $G_g = \{\mu_g, q_g, s_g, \sigma_g, c_g\}$, the network outputs adjustments:

$$(\delta\mu, \delta q, \delta s) = \mathbb{D}(\mathbb{E}_p(\mu_g), \mathbb{E}_t(t)), \quad (1)$$

where \mathbb{E}_p and \mathbb{E}_t are spatial and temporal embeddings. The updated Gaussian at time t is:

$$\mu_g(t) = \mu_g + \delta\mu, \quad q_g(t) = q_g \cdot \delta q, \quad s_g(t) = s_g + \delta s. \quad (2)$$

The network aligns these updates with ground-truth frames to optimize both Gaussian and deformation parameters.

3.3. Mesh Extraction from Dynamic 3D Gaussians

Existing methods for mesh extraction from dynamic Gaussians, such as DynaSurfGS [2] and Dynamic2DGS [69], typically employ Marching Cubes [30] or TSDF [37] to fuse multiple RGB-D views into a 3D mesh. To enhance mesh quality and simplify geometry, clustering-based floating-point elimination and quadric decimation [15] are occasionally applied. However, these approaches are limited in that they do not produce meshes with temporal facet-ID correspondence, making them unsuitable for MaGS (as such correspondences are necessary to transfer learned deformation principles from the rendering process to simulation).

Instead, to obtain the initial mesh, we first train a deformation field-based dynamic 3D Gaussian model (Sec. 3.2) to coarsely represent the dynamic target. We render the median depths of the 3D Gaussians at $T = 0$ across 100 view-points, fuse them via TSDF, and reconstruct the initial mesh via TSDF Integration [37]. The mesh vertices are then input into the deformation field to predict per-frame deformations. This deformation process only alters vertex positions, maintaining facet-ID correspondence across frames.

4. Methodology

Figure 2 gives an overview of MaGS. Section 4.1 details the Mesh-adsorbed Gaussian initialization. Section 4.2 describes the network design. Section 4.3 outlines the optimization process for Mesh-adsorbed Gaussian and network, and Section 4.4 explains the simulation process. In Section 4.5, we discuss MaGS with existing methods to illustrate the distinctions of our design. Implementation details of each network can be found in the Appendix.

4.1. Mesh-adsorbed Gaussian

The Mesh-adsorbed Gaussian is a hybrid representation composed of a triangular mesh and 3D Gaussian. The mesh part is a triangular mesh with N vertices and M facets, defined by three attributes: *vertices*, *facets*, and *normals*. Here, *vertices* represents the spatial coordinates of each vertex, denoted as $v_i \in \mathbb{R}^3$ for $i = 1, 2, \dots, N$. The *facets* attribute defines each facet by referencing three vertices of the mesh. The *normals* attribute provides the normal vector $n_i \in \mathbb{R}^3$ for each facet $i = 1, 2, \dots, M$, which can be calculated from the *vertices* and *facets*. Following Shao *et al.* [45], we compute the normal at any position on a facet by linearly interpolating the normals at its vertices.

The 3D Gaussian of Mesh-adsorbed Gaussians extends the standard 3D Gaussian with attributes b and h , along with MeshId. Here, $b = (b_1, b_2, b_3)$ represents the barycentric coordinates on a mesh facet, satisfying $b_1 + b_2 + b_3 = 1$ and $b_k \geq 0$ for $k = 1, 2, 3$, while $h \in [0, 1]$ denotes the offset along the facet’s normal direction. The MeshId attribute records the facet on which the Gaussian resides.

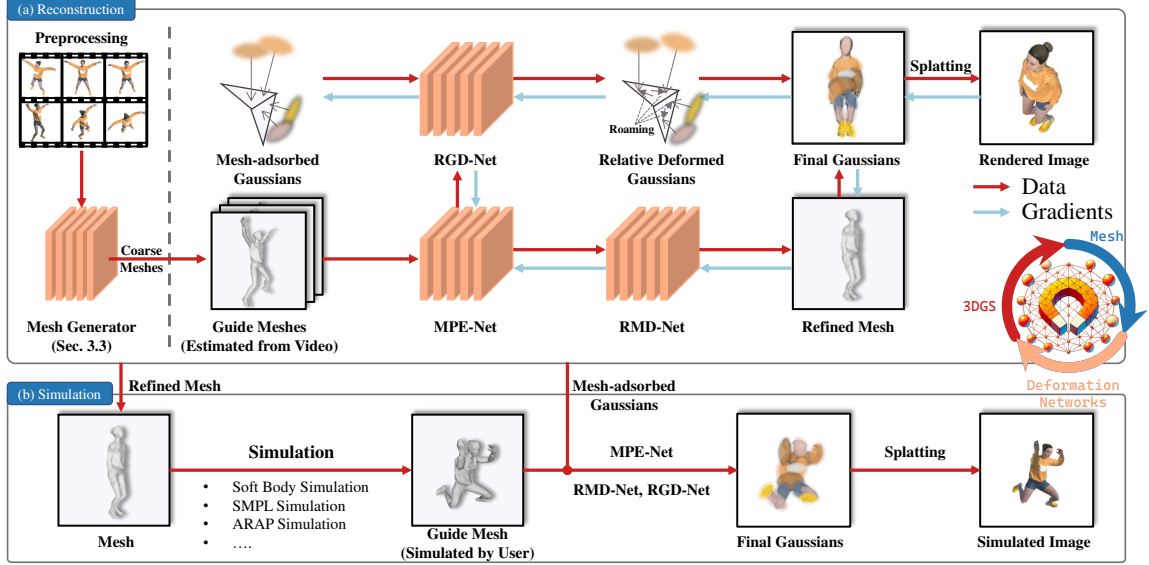


Figure 2. **Pipeline of MaGS.** MaGS begins by extracting a temporally consistent coarse mesh for each frame of video (Sec. 3.3). These meshes, referred to as Guide Meshes, provide the foundation for dynamic reconstruction. During the **reconstruction process**, pose information from the guide meshes is extracted using MPE-Net and forwarded to RMD-Net and RGD-Net. RMD-Net and RGD-Net perform relative deformations on the guide mesh and the Mesh-adsorbed Gaussians, respectively, yielding the refined mesh and relative deformed Gaussian. These two components produce the Final Gaussians. Splating-based rendering is then employed, with the rendering loss used to optimize the Gaussians, MPE-Net, RMD-Net, and RGD-Net via backpropagation. The reconstruction phase not only yields a high-precision mesh and Gaussians but also trains the networks to learn deformation principles from the video, effectively preparing them for simulation. In the **simulation phase**, mesh-based techniques—such as soft body simulation, ARAP, and SMPL—are used to deform the reconstructed meshes, producing new guide mesh. Mesh-adsorbed Gaussians are also inherited (adsorbed to their corresponding facets). The following process resembles the reconstruction, where MPE-Net, RMD-Net, and RGD-Net are again utilized to yield the Final Gaussians, which are then rendered to generate the final image.

Initially, Gaussians are randomly initialized, with b and h values sampled uniformly, and MeshId evenly distributed across facets, ensuring an equal number of Gaussians per facet.

To compute the spatial coordinates of a 3D Gaussian based on its logical coordinates, we define the position μ_i for a Gaussian i on facet j as:

$$l_j = \frac{|v_{j_1} \vec{v}_{j_2}| |v_{j_1} \vec{v}_{j_3}|}{|v_{j_1} \vec{v}_{j_2} \times v_{j_1} \vec{v}_{j_3}|}, \mu_i = b_{i1}v_{j_1} + b_{i2}v_{j_2} + b_{i3}v_{j_3} + h_i n_j l_j, \quad (3)$$

where l_j represents the scaling factor for facet j , $v_{j_1}, v_{j_2}, v_{j_3}$ are the spatial coordinates of the three vertices of facet j , and n_j denotes the normal vector of facet j .

When the offset h_i of the Gaussian i is nonzero, the Gaussian does not lie directly on the mesh surface but instead hovers above it along the normal direction. This offset increases the Gaussian’s freedom to move across the mesh, a design we term **Gaussian Hover**.

4.2. Deformation for Mesh-adsorbed Gaussian

We design three networks to handle deformation: MPE-Net, RMD-Net, and RGD-Net. MPE-Net extracts pose embedding from coarse mesh, which is input to RMD-Net and

RGD-Net. RMD-Net predicts the relative mesh deformation to refine the dynamic mesh, while RGD-Net adjusts the corresponding Gaussian parameters.

4.2.1. Mesh Pose Embedding Network

Since timestamps are not available as input during simulation, relying on temporal cues to guide deformation would hinder simulation performance. Therefore, we abandon timestamps as input for our algorithm. Instead, we design MPE-Net to extract information directly from the guide (coarse) mesh. MPE-Net takes as input a subset of mesh vertices called handle vertices:

$$\mathcal{E}_M, \mathcal{E}_V = \text{MPE}(\mathcal{H}, \mathcal{N}), \quad (4)$$

where \mathcal{H} and \mathcal{N} denote the sets of handle vertices and their corresponding normal vectors, and \mathcal{E}_M and \mathcal{E}_V encode the mesh pose and vertex-specific deformations, respectively.

To ensure that handle vertices capture more pose information of the mesh, we use Poisson disk sampling [65] to sample points uniformly throughout the mesh and map them to mesh vertices using the k-nearest neighbors (KNN).

4.2.2. Relative Mesh Deformation Network

We design RMD-Net to learn motion priors from video within a rendering process, enabling it to refine guide mesh

deformations by predicting the relative displacement between the guide and refined meshes. RMD-Net takes \mathcal{E}_M and \mathcal{E}_V as inputs to predict refined deformation information for each vertex i . This process can be represented as:

$$(\Delta v, \Delta q, \Delta s, \Delta \sigma, \Delta c)_i = \text{RMD}(\mathcal{E}_M, \mathcal{E}_{V_i}), \quad (5)$$

where Δv represents the displacement of the vertex and Δq represents the rotation applied to the vertex coordinates. Although Δs , $\Delta \sigma$, and Δc do not correspond directly to mesh deformation, they are utilized in Mesh-adsorbed Gaussian rendering.

4.2.3. Relative Gaussian Deformation Network

Unlike other methods, MaGS allows Gaussians to **roam on** rather than **anchor to** a facet when the mesh deforms. Specifically, we consider that the barycentric coordinates b and offset h of each Gaussian on the mesh change with the mesh deformations. For this purpose, we design RGD-Net to calculate changes of b and h , denoted as Δb and Δh , during mesh deformation.

RGD-Net takes \mathcal{E}_M , the $\mathcal{E}_{V_1}, \mathcal{E}_{V_2}, \mathcal{E}_{V_3}$ of the three vertices of the facet, and (b_i, h_i) of Mesh-adsorbed Gaussian i as input and outputs $(\Delta b_i, \Delta h_i)$:

$$(\Delta b_i, \Delta h_i) = \text{RGD}(\mathcal{E}_M, \mathcal{E}_{V_1}, \mathcal{E}_{V_2}, \mathcal{E}_{V_3}, b_i, h_i), \quad (6)$$

4.3. Optimization for Mesh-adsorbed Gaussian

This section explains the optimization process, detailing how the mesh vertices and Mesh-adsorbed Gaussian parameters are updated through backpropagation.

When the mesh deforms, the facet area and surface normal change, affecting the Final Gaussian. For facet j , we calculate $\Delta l_j = \frac{l'_j}{l_j}$, where l_j and l'_j are the scaling factors before and after deformation, respectively. Similarly, the change in the surface normal, Δn_j , is computed. Using the methods in Section 4.2, we obtain $\Delta v_j, \Delta q_j, \Delta s_j, \Delta \sigma_j, \Delta c_j$ for facet j , and $(\Delta b_i, \Delta h_i)$ for Gaussian i . The mean position μ'_i of the Final Gaussian is then computed using the following equations:

$$\begin{aligned} b'_i &= b_i + \Delta b_i, & h'_i &= h_i + \Delta h_i, & v'_{j_k} &= v_{j_k} + \Delta v_{j_k}, \\ \mu'_i &= b'_{i1} v'_{j_1} + b'_{i2} v'_{j_2} + b'_{i3} v'_{j_3} + h'_i n'_j l'_j, \end{aligned} \quad (7)$$

where n'_j represents the normal after deformation.

Other properties σ'_i, c'_i, s'_i , and q'_i are updated as follows:

$$\begin{aligned} \sigma'_i &= (b'_{i1} \Delta \sigma_{j_1} + b'_{i2} \Delta \sigma_{j_2} + b'_{i3} \Delta \sigma_{j_3}) \cdot \sigma_i, \\ c'_i &= (b'_{i1} \Delta c_{j_1} + b'_{i2} \Delta c_{j_2} + b'_{i3} \Delta c_{j_3}) + c_i, \\ s'_i &= \Delta l_j \cdot (b'_{i1} \Delta s_{j_1} + b'_{i2} \Delta s_{j_2} + b'_{i3} \Delta s_{j_3}) \cdot s_i, \\ q'_i &= \Delta n_j \cdot (b'_{i1} \Delta q_{j_1} + b'_{i2} \Delta q_{j_2} + b'_{i3} \Delta q_{j_3}) \cdot q_i, \end{aligned} \quad (8)$$

where s_i, q_i, σ_i , and c_i are the undeformed properties.

Through the above steps, we derive the Final Gaussians based on Mesh-adsorbed Gaussians by utilizing the deformation information provided by the mesh. Formally, this can be expressed as:

$$(b_{\text{gs}} + \Delta b, h_{\text{gs}} + \Delta h) \times (v_{\text{mesh}} + \Delta v) \rightarrow \text{Final Gaussians}, \quad (9)$$

where \times denotes the interpolation operation MaGS uses to calculate Gaussian properties.

This formulation captures the adjustment of Mesh-adsorbed Gaussians properties in response to mesh deformations, ensuring that the Gaussians' positions and orientations are accurately updated as the underlying mesh structure deforms. The Final Gaussians incorporate all parameters of a standard 3D Gaussian and can be splatted using a differentiable renderer [24]. We render the Final Gaussians, then compare the rendered image with the ground truth. The loss is computed using the following equation, enabling backpropagation:

$$\mathcal{L} = \mathcal{L}_{L1} \times (1 - \lambda_{\text{ssim}}) + \mathcal{L}_{\text{SSIM}} \times \lambda_{\text{ssim}}, \quad (10)$$

where \mathcal{L}_{L1} represents the pixel-wise L1 difference, and $\mathcal{L}_{\text{SSIM}}$ represents the structural similarity loss. Since the entire process is differentiable, we can jointly optimize MPE-Net, RMD-Net, RGD-Net, and the Mesh-adsorbed Gaussians through backpropagation based on rendering errors.

4.4. Mesh-guided Simulation

Finally, we discuss how the optimized Mesh-adsorbed Gaussians can be leveraged for simulation tasks, taking advantage of the mesh's universality to support various physics-based priors. Generally, we utilize physics simulation tools like Blender or Taichi for soft-body simulation, As-Rigid-As-Possible (ARAP) editing simulation, cloth simulation, etc, to enhance the reconstructed mesh with physical prior. Moreover, for avatar datasets, SMPL is also a feasible simulation method.

The RMD-Net, introduced in Section 4.2, refines the guide mesh using deformation principles from training videos, enhancing deformation reasonability. Since our pipeline retains the original vertex-facet relationships, we can apply the offsets Δv calculated by RMD-Net directly to the vertices v of the guide mesh to obtain the optimized deformed mesh. After deforming the mesh, we adjust the Mesh-adsorbed Gaussians accordingly and render the simulation results with the method described in Section 4.3. This approach enables high-precision rendering of simulation outcomes based on Mesh-adsorbed Gaussians.

4.5. Discussion

Advantages of MaGS. MaGS offers three key advantages over existing methods: **1)** It incorporates a continuous mesh that can be directly utilized for simulation, distinguishing it

Table 1. **Quantitative results on D-NeRF dataset [40]**. We present the average PSNR/MS-SSIM/VGG-LPIPS values for novel view synthesis on dynamic scenes from D-NeRF, with each cell colored to indicate the **best**, **second best** and **third best**.

Type	Method	Bouncingballs			Hellwarrior			Hook			Jumpingjacks		
		PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓
NeRF-Based	D-NeRF [40]	38.18	0.9910	0.0120	28.47	0.9317	0.0638	30.42	0.9820	0.0379	33.73	0.9902	0.0206
	TiNeuVox-B [9]	40.62	0.9969	0.0083	30.68	0.9495	0.0592	32.45	0.9898	0.0374	35.50	0.9944	0.0191
	Tensor4D [44]	25.36	0.9610	0.0411	31.40	0.9250	0.0675	29.03	0.9550	0.4990	24.01	0.9190	0.0768
	K-Planes [11]	40.61	0.9910	0.2970	25.27	0.9480	0.0755	28.59	0.9530	0.5810	32.27	0.9710	0.0389
3DGS Based	Deformable-GS [62]	37.09	0.9974	0.0060	41.17	0.9934	0.0152	36.48	0.9962	0.0117	37.99	0.9978	0.0065
	4D-GS [57]	40.67	0.9968	0.0069	31.84	0.9629	0.0474	33.90	0.9922	0.0201	36.66	0.9962	0.0106
	SC-GS [19]	44.91	0.9980	0.0166	42.93	0.9940	0.0155	39.89	0.9970	0.0076	41.13	0.9980	0.0067
	SP-GS [53]	41.72	0.9970	0.0097	40.25	0.9904	0.0289	35.42	0.9928	0.0202	34.70	0.9926	0.0181
	Grid4D [59]	41.92	0.9981	0.0053	43.43	0.9959	0.0099	39.14	0.9974	0.0067	39.92	0.9984	0.0050
Mesh-3DGS Based	D-MiSo [51]	38.80	0.9957	0.0140	40.69	0.9919	0.0233	37.53	0.9961	0.0116	41.86	0.9988	0.0042
	DG-Mesh [27]	31.66	0.9762	0.0351	27.80	0.9705	0.0546	29.34	0.9613	0.0516	27.54	0.9702	0.1149
	DynaSurfGS [2]	40.92	0.9948	0.0139	29.45	0.9758	0.0360	32.97	0.9773	0.0277	35.49	0.9864	0.0202
	Ours	41.97	0.9976	0.0055	43.69	0.9957	0.0098	41.23	0.9984	0.0049	44.29	0.9993	0.0022

Type	Method	Mutant			Standup			Trex			Average		
		PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓	PSNR↑	MS-SSIM↑	LPIPS↓
NeRF-Based	D-NeRF	32.31	0.9871	0.0256	34.42	0.9896	0.0197	32.07	0.9910	0.0178	32.80	0.9804	0.0282
	TiNeuVox-B	33.75	0.9920	0.0288	35.95	0.9930	0.0187	33.18	0.9955	0.0174	34.59	0.9873	0.0270
	Tensor4D	29.99	0.9510	0.0422	30.86	0.9640	0.0214	23.51	0.9340	0.0640	27.74	0.9441	0.1160
	K-Planes	33.79	0.9820	0.0207	34.31	0.9840	0.0194	31.41	0.9800	0.0234	32.32	0.9727	0.1508
3DGS Based	Deformable-GS	41.02	0.9990	0.0038	42.01	0.9988	0.0036	36.07	0.9978	0.0056	38.83	0.9972	0.0075
	4D-GS	37.16	0.9963	0.0106	37.79	0.9959	0.0102	35.00	0.9968	0.0087	36.15	0.9910	0.0163
	SC-GS	45.19	0.9990	0.0028	47.89	0.9990	0.0023	41.24	0.9980	0.0046	43.31	0.9976	0.0080
	SP-GS	38.69	0.9970	0.0118	42.22	0.9977	0.0095	32.93	0.9930	0.0163	37.99	0.9944	0.0164
	Grid4D	45.33	0.9995	0.0020	47.35	0.9996	0.0015	40.82	0.9992	0.0027	42.56	0.9983	0.0047
Mesh-3DGS Based	D-MiSo	44.45	0.9992	0.0027	47.46	0.9995	0.0017	40.52	0.9990	0.0031	41.62	0.9972	0.0087
	DG-Mesh	31.44	0.9693	0.0378	32.31	0.9791	0.0355	29.10	0.9674	0.0507	29.88	0.9706	0.0543
	DynaSurfGS	38.61	0.9903	0.0150	37.76	0.9884	0.0191	34.21	0.9848	0.0230	35.63	0.9854	0.0221
	Ours	46.42	0.9996	0.0019	49.16	0.9997	0.0010	41.65	0.9993	0.0025	44.06	0.9985	0.0040

Table 2. **Quantitative results on PeopleSnapshot dataset [1]**. We present the average PSNR/SSIM/VGG-LPIPS values for novel pose synthesis on PeopleSnapshot, with each cell colored to indicate the **best**, **second best** and **third best**.

Methods	male-3-casual			male-4-casual			female-3-casual			female-4-casual		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Anim-NeRF [6]	29.37	0.9700	0.0170	28.37	0.9600	0.0270	28.91	0.9740	0.0220	28.90	0.9680	0.0170
InstantAvatar [20]	30.91	0.9770	0.0220	29.77	0.9800	0.0250	29.73	0.9750	0.0250	30.92	0.9770	0.0210
3DGS-Avatar [41]	34.28	0.9724	0.0149	30.22	0.9653	0.0231	30.57	0.9581	0.0209	33.16	0.9678	0.0157
SplattingAvatar [45]	36.48	0.9766	0.0247	33.98	0.9776	0.0340	37.36	0.9754	0.0345	35.25	0.9734	0.0271
Ours	38.94	0.9854	0.0110	35.19	0.9842	0.0175	39.31	0.9829	0.0215	37.83	0.9828	0.0116

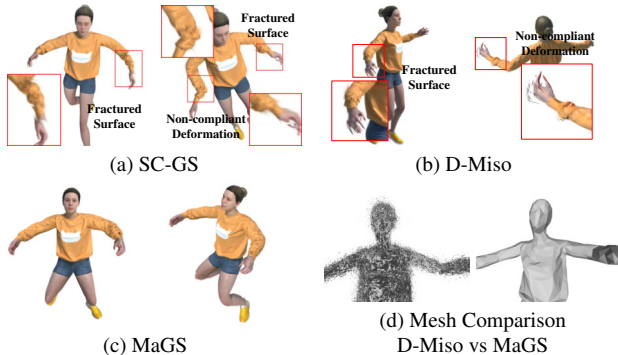


Figure 3. **Simulation comparison on the D-NeRF dataset [40]**.

from methods like Dynamic2DGS (lacks temporal continuity), D-MiSo (lacks spatial continuity), and SC-GS (lacks both). For point cloud-based methods, when large edits are performed, the lack of surface information provided by the mesh leads to the issue shown in Figure 3. 2) MaGS allows Gaussians to roam along the mesh surface. In contrast, DG-Mesh and SplattingAvatar do not support the mesh-

Gaussian displacement. 3) MaGS preserves the continuity of the mesh between frames, ensuring a consistent point-and-facet correspondence. **This continuity is crucial for us to inherit the Mesh-adsorbed Gaussian from reconstruction to simulation**, and represents a notable improvement over methods like DynaSurfGS and Dynamic 2DGS, which generate independent meshes per frame using TSDF without maintaining cross-frame consistency. This enhances MaGS’ suitability for downstream tasks.

MaGS’s Capability in handling Large and OOD Deformation. In MaGS, RMD-Net is trained to calibrate OOD coarse meshes to video-aligned fine meshes. RGD-Net operates in local relative coordinates, which naturally supports rendering generalization to OOD deformations. These two key modules enable MaGS to generalize learned video priors to novel deformations in both simulation and rendering. Except for priors in video, MaGS is highly compatible with external deformation priors (e.g., ARAP, SMPL), which further improves generalization to larger deformations. Increasing motion diversity in the video also en-

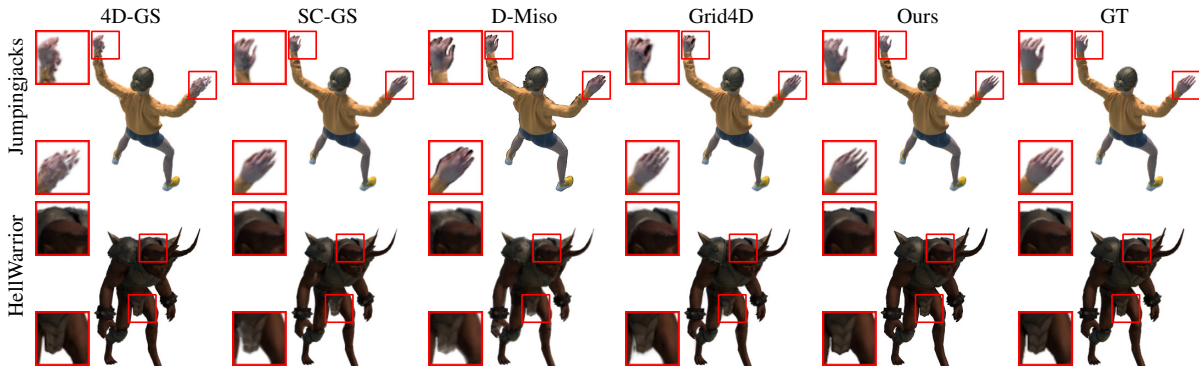


Figure 4. **Qualitative comparisons on D-NeRF [40].** We compare MaGS with 4D-GS [57], SC-GS [19], D-Miso [51], and Grid4D [59].

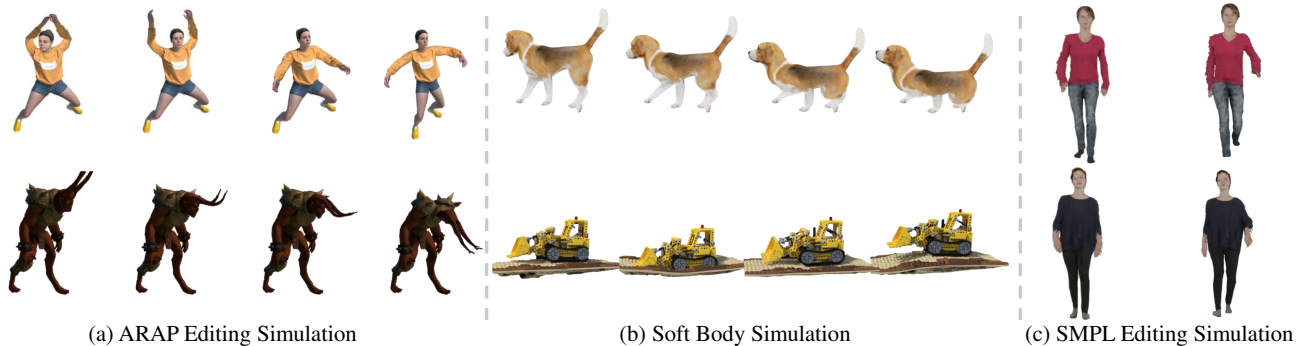


Figure 5. **Mesh-guided Simulation of MaGS.** We utilize user-edited meshes—modified through methods such as dragging, soft-body collisions, or SMPL-based motion editing—to guide deformation and achieve natural rendering results.

hances OOD performance. However, like other mesh-based methods, MaGS does not support changed-topology deformations such as fluid dynamics.

Trade-off in Mesh Complexity. The facet number of the mesh is a hyperparameter in TSDF integration, which we set to $\sim 1\text{K}$, a relative small value that makes MaGS more efficient. While we can easily increase it, doing so is unnecessary: **1)** In MaGS, the rendering/simulation is essentially performed by Gaussians, while mesh only guides Gaussians motion adjustments, which is not the decisive factor in the final performance; **2)** A simple mesh is capable of guiding deformations, as the rank required for deformations is significantly lower than the rank of even a simple mesh [8]; **3)** The mesh-GS adaptive displacement design further minimizes the MaGS’s reliance on complex meshes.

5. Experiments

5.1. Experimental Settings

We evaluated MaGS on four datasets: the D-NeRF dataset [40], the DG-Mesh dataset [27], the PeopleSnapshot dataset [1], and the Nerfies dataset [38]. The D-NeRF dataset comprises eight synthetic monocular scenes. The DG-Mesh dataset provides ground-truth meshes for each frame, enabling the evaluation of dynamic reconstruction

accuracy. The PeopleSnapshot dataset includes monocular videos of individuals. Following previous work [45], we use optimized SMPL models from InstantAvatar [20] as guide meshes for PeopleSnapshot. The Nerfies dataset contains real-world dynamic scenes and is commonly used to benchmark dynamic novel view synthesis. Following previous studies [19, 27, 45], the D-NeRF and DG-Mesh datasets are evaluated at a resolution of 400×400 , while the PeopleSnapshot dataset is evaluated at 1080×1080 . For Nerfies, we use the standard resolution provided by the dataset. Evaluation metrics include PSNR, SSIM, MS-SSIM [56], and LPIPS [68]. Following DG-Mesh, we also use EMD and CD to evaluate reconstruction accuracy. All experiments were performed on an NVIDIA RTX 4090.

5.2. Quantitative Comparisons

D-NeRF. We evaluate MaGS against state-of-the-art methods on the D-NeRF dataset. Table 1 presents the evaluation results across seven scenes, excluding Lego¹. MaGS achieves higher PSNR, SSIM, and LPIPS metrics across most scenes, surpassing SC-GS [19] (the second-best method) with an average PSNR improvement of 0.7 dB.

¹Yang *et al.* [62] noted inconsistencies in D-NeRF’s Lego scene and provided a corrected version. See Appendix for details.

Table 3. **Quantitative results on DG-Mesh [27].** We use * to indicate the data provided by DynaSurfGS.

Methods	CD↓	EMD↓	PSNR↑	Time↓	FacesNum
D-NeRF* [40]	1.1506	0.1710	28.44	/	/
K-Plane* [11]	0.9224	0.1440	31.13	/	/
HexPlane* [3]	1.9072	0.1474	30.18	/	/
TiNeuVox-B* [9]	2.5186	0.1666	31.96	/	/
DG-Mesh [27]	0.6022	0.1192	31.43	89.3	170,232
DynaSurfGS* [2]	0.7570	0.1136	33.18	/	/
Dynamic 2D Gaussians [69]	0.5254	0.1260	36.40	72.7	1,419,454
Ours	0.6662	0.1106	40.76	47.6	981

Table 4. **PSNR (↑) comparisons on the Nerfies dataset [38].**

Methods	Tobysit↑	Broom↑	Curls↑	Tail↑
D-3DGS [62]	21.0	20.4	18.2	24.6
SC-GS [19]	16.3	15.3	12.3	17.0
Grid4D [59]	21.0	20.2	18.6	24.3
Ours	22.1	21.3	18.7	24.2

DG-Mesh. Table 3 shows that MaGS achieves leading performance in PSNR and EMD on the DG-Mesh dataset². MaGS performs closely to the state-of-the-art in Chamfer Distance (CD), using only $\sim 1k$ mesh faces compared to $\sim 170k$ for DG-Mesh and $\sim 1.4m$ for Dynamic 2D Gaussians. Moreover, MaGS requires only 47.6 minutes for optimization, which is considerably faster than DG-Mesh (89.3 min) and Dynamic 2D Gaussians (72.7 min).

PeopleSnapshot. We evaluate MaGS on the PeopleSnapshot dataset [1]. Table 2 shows that MaGS achieves the highest PSNR and SSIM metrics across all scenarios, surpassing SplattingAvatar. MaGS demonstrates an approximately 2 dB average PSNR improvement over the next best method and also outperforms in LPIPS.

Nerfies. Table 4 shows PSNR results on the Nerfies dataset. MaGS achieves the highest average PSNR, outperforming prior methods on three out of four scenes. It surpasses SC-GS [19] by 1.1 dB on average, demonstrating strong generalization across diverse scenes.

5.3. Qualitative Comparisons and Simulation

We present qualitative comparisons on two datasets: D-NeRF and DG-Mesh. Figure 4 illustrates the results on D-NeRF, with magnified images highlighting the finer details of the synthesized outputs. Our method achieves superior visual quality, producing sharper and more accurate reconstructions. Similarly, Figure 6 shows a comparison on DG-Mesh, where we compare our method with the state-of-the-art dynamic mesh extraction techniques. Our method generates meshes that more closely resemble the ground truth, demonstrating improved accuracy in reconstruction.

In terms of simulation, Figure 5 showcases the results produced by MaGS. As shown, our method effectively pre-

²Includes beagle, bird, duck, girlwalk, horse. See Appendix for details.

Table 5. **Ablation experiments on the D-NeRF [40].**

Method	PSNR↑	SSIM↑	LPIPS↓
MaGS w/o RMD-Net and RGD-Net	41.14	0.9974	0.0064
MaGS w/o Gaussian Hover	41.87	0.9977	0.0059
MaGS w/o RGD-Net	42.98	0.9982	0.0047
MaGS Full	44.06	0.9985	0.0040

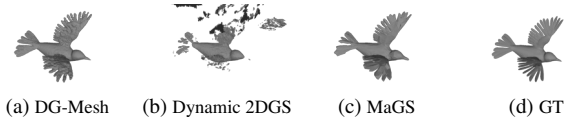


Figure 6. **Mesh Comparison on DG-Mesh dataset [27].**

serves texture throughout the deformation process. Additionally, Figure 3 compares MaGS with SC-GS, a leading simulation method based on sparse guide points. MaGS avoids the surface fracture problems in SC-GS when subjected to large deformations.

5.4. Ablation Studies

We ablate MaGS to understand the contribution of key design choices with the default settings, including the RMD-Net, Gaussians Hover, and RGD-Net, as shown in Table 5, removing both RMD-Net and RGD-Net results in a PSNR drop of 2.92, which is a 6.6% decrease. Disabling Gaussians Hover reduces PSNR by 2.19, a 5.0% reduction while excluding RGD-Net alone leads to a PSNR drop of 1.08, or 2.4%. The full MaGS configuration achieves the highest PSNR of 44.06, highlighting the importance of each component in enhancing the model’s performance.

6. Conclusion

This paper introduces the MaGS method, which addresses 3D reconstruction and simulation within a unified framework. It creates a novel adsorbed mesh-Gaussian 3D representation by constraining 3D Gaussians to roam near the mesh surface, which combines the rendering flexibility of 3D Gaussians with the adaptability of meshes to different geometric priors. MaGS is compatible with various deformation priors like ARAP, SMPL, and soft physics simulation. Extensive experiments on D-NeRF, DG-Mesh, and PeopleSnapshot demonstrate that MaGS achieves SOTA performance in both reconstruction and simulation.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (62293554, U2336212), National Key R&D Program of China (SQ2023AAA01005), Natural Science Foundation of Zhejiang Province, China (LZ24F020002), Ningbo Innovation ”Yongjiang 2035” Key Research and Development Programme (2024Z292), and Young Elite Scientists Sponsorship Program by CAST (2023QNRC001).

References

- [1] Thiemo Alldieck, Marcus A. Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3D people models. *CoRR*, abs/1803.04758, 2018. 6, 7, 8
- [2] Weiwei Cai, Weicai Ye, Peng Ye, Tong He, and Tao Chen. DynaSurfGS: Dynamic surface reconstruction with planar-based gaussian splatting. *CoRR*, abs/2408.13972, 2024. 2, 3, 6, 8
- [3] Ang Cao and Justin Johnson. HexPlane: A fast representation for dynamic scenes. *CoRR*, abs/2301.09632, 2023. 2, 8
- [4] Danpeng Chen, Hai Li, Weicai Ye, Yifan Wang, Weijian Xie, Shangjin Zhai, Nan Wang, Haomin Liu, Hujun Bao, and Guofeng Zhang. PGSR: planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *CoRR*, abs/2406.06521, 2024. 2
- [5] Guikun Chen and Wenguan Wang. A survey on 3D gaussian splatting. *CoRR*, abs/2401.03890, 2024. 1
- [6] Jianchuan Chen, Ying Zhang, Di Kang, Xuefei Zhe, Linchao Bao, Xu Jia, and Huchuan Lu. Animatable neural radiance fields from monocular RGB videos, 2021. arXiv:2106.13629. 6
- [7] Yufan Chen, Lizhen Wang, Qijing Li, Hongjiang Xiao, Shengping Zhang, Hongxun Yao, and Yebin Liu. Mono-GaussianAvatar: Monocular gaussian point-based head avatar. *CoRR*, abs/2312.04558, 2023. 3
- [8] Yuchao Dai, Hongdong Li, and Mingyi He. A simple prior-free method for non-rigid structure-from-motion factorization. *Int. J. Comput. Vis.*, 107(2):101–122, 2014. 7
- [9] 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. *CoRR*, abs/2205.15285, 2022. 2, 6, 8
- [10] Yutao Feng, Xiang Feng, Yintong Shang, Ying Jiang, Chang Yu, Zeshun Zong, Tianjia Shao, Hongzhi Wu, Kun Zhou, Chenfanfu Jiang, and Yin Yang. Gaussian splashing: Unified particles for versatile motion synthesis and rendering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pages 518–529. Computer Vision Foundation / IEEE, 2025. 1
- [11] Sara Fridovich-Keil, Giacomo Meanti, Frederik Warburg, Benjamin Recht, and Angjoo Kanazawa. K-Planes: Explicit radiance fields in space, time, and appearance. *CoRR*, abs/2301.10241, 2023. 2, 6, 8
- [12] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(8):1362–1376, 2010. 1
- [13] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. *CoRR*, abs/2105.06468, 2021. 2
- [14] Xiangjun Gao, Xiaoyu Li, Yiyu Zhuang, Qi Zhang, Wenbo Hu, Chaopeng Zhang, Yao Yao, Ying Shan, and Long Quan. Mani-GS: Gaussian splatting manipulation with triangular mesh. *CoRR*, abs/2405.17811, 2024. 1, 2, 3
- [15] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1997, Los Angeles, CA, USA, August 3-8, 1997*, pages 209–216. ACM, 1997. 3
- [16] Antoine Guédon and Vincent Lepetit. SuGaR: Surface-aligned gaussian splatting for efficient 3D mesh reconstruction and high-quality mesh rendering. *CoRR*, abs/2311.12775, 2023. 2, 3
- [17] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-MipRF: Tri-mip representation for efficient anti-aliasing neural radiance fields. *CoRR*, abs/2307.11335, 2023. 2
- [18] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2D gaussian splatting for geometrically accurate radiance fields. *CoRR*, abs/2403.17888, 2024. 2
- [19] 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. *CoRR*, abs/2312.14937, 2023. 2, 3, 6, 7, 8
- [20] Tianjian Jiang, Xu Chen, Jie Song, and Otmar Hilliges. InstantAvatar: Learning avatars from monocular video in 60 seconds. *CoRR*, abs/2212.10550, 2022. 6, 7
- [21] Wei Jiang, Kwang Moo Yi, Golnoosh Samei, Oncel Tuzel, and Anurag Ranjan. NeuMan: Neural human radiance field from a single video. *CoRR*, abs/2203.12575, 2022. 1
- [22] Ying Jiang, Chang Yu, Tianyi Xie, Xuan Li, Yutao Feng, Huamin Wang, Minchen Li, Henry Y. K. Lau, Feng Gao, Yin Yang, and Chenfanfu Jiang. VR-GS: A physical dynamics-aware interactive gaussian splatting system in virtual reality. *CoRR*, abs/2401.16663, 2024. 2
- [23] HyunJun Jung, Nikolas Brasch, Jifei Song, Eduardo Pérez-Pellitero, Yiren Zhou, Zhihao Li, Nassir Navab, and Benjamin Busam. Deformable 3D gaussian splatting for animatable human avatars. *CoRR*, abs/2312.15059, 2023. 1, 2
- [24] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *CoRR*, abs/2308.04079, 2023. 1, 2, 3, 5
- [25] Christoph Lassner and Michael Zollhöfer. Pulsar: Efficient sphere-based neural rendering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 1440–1449. Computer Vision Foundation / IEEE, 2021. 1
- [26] Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. *CoRR*, abs/2311.16096, 2023. 2
- [27] Isabella Liu, Hao Su, and Xiaolong Wang. Dynamic gaussians mesh: Consistent mesh reconstruction from dynamic scenes. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. 2, 6, 7, 8
- [28] Jiawei 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. *CoRR*, abs/2205.15723, 2022. 2

- [29] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):248:1–248:16, 2015. 2
- [30] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987, Anaheim, California, USA, July 27-31, 1987*, pages 163–169. ACM, 1987. 2, 3
- [31] Yawei Luo and Yi Yang. Large language model and domain-specific model collaboration for smart education. *Frontiers Inf. Technol. Electron. Eng.*, 25(3):333–341, 2024. 1
- [32] Siwei Meng, Yawei Luo, and Ping Liu. Grounding creativity in physics: A brief survey of physical priors in AIGC. *CoRR*, abs/2502.07007, 2025. 1
- [33] Qiaowei Miao, Yawei Luo, and Yi Yang. PLA4D: pixel-level alignments for Text-to-4D gaussian splatting. *CoRR*, abs/2405.19957, 2024.
- [34] Qiaowei Miao, Kehan Li, Jinsheng Quan, Zhiyuan Min, Shaojie Ma, Yichao Xu, Yi Yang, and Yawei Luo. Advances in 4D generation: A survey. *CoRR*, abs/2503.14501, 2025. 1
- [35] 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. *CoRR*, abs/2003.08934, 2020. 1, 2
- [36] Arthur Moreau, Jifei Song, Helisa Dharmo, Richard Shaw, Yiren Zhou, and Eduardo Pérez-Pellitero. Human gaussian splatting: Real-time rendering of animatable avatars. *CoRR*, abs/2311.17113, 2023. 3
- [37] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew W. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011, Basel, Switzerland, October 26-29, 2011*, pages 127–136. IEEE Computer Society, 2011. 3
- [38] 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 *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 5845–5854. IEEE, 2021. 2, 7, 8
- [39] 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. *CoRR*, abs/2106.13228, 2021. 1, 2
- [40] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. *CoRR*, abs/2011.13961, 2020. 2, 6, 7, 8
- [41] Zhiyin Qian, Shaofei Wang, Marko Mihajlovic, Andreas Geiger, and Siyu Tang. 3DGS-Avatar: Animatable avatars via deformable 3D gaussian splatting. *CoRR*, abs/2312.09228, 2023. 2, 3, 6
- [42] Yi-Ling Qiao, Alexander Gao, Yiran Xu, Yue Feng, Jia-Bin Huang, and Ming C. Lin. Dynamic mesh-aware radiance fields. *CoRR*, abs/2309.04581, 2023. 1
- [43] Jinsheng Quan, Chunshi Wang, and Yawei Luo. ParticleGS: Particle-based dynamics modeling of 3D gaussians for prior-free motion extrapolation. *CoRR*, abs/2505.20270, 2025. 1
- [44] 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. *CoRR*, abs/2211.11610, 2022. 2, 6
- [45] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. SplattingAvatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. *CoRR*, abs/2403.05087, 2024. 2, 3, 6, 7
- [46] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3D. *ACM Trans. Graph.*, 25(3):835–846, 2006. 1
- [47] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing, Barcelona, Spain, July 4-6, 2007*, pages 109–116. Eurographics Association, 2007. 2
- [48] Jiayang Tang, Hang Zhou, Xiaokang Chen, Tianshu Hu, Errui Ding, Jingdong Wang, and Gang Zeng. Delicate textured mesh recovery from NeRF via adaptive surface refinement. *CoRR*, abs/2303.02091, 2023. 2
- [49] Demetri Terzopoulos, John C. Platt, Alan H. Barr, and Kurt W. Fleischer. Elastically deformable models. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987, Anaheim, California, USA, July 27-31, 1987*, pages 205–214. ACM, 1987. 2
- [50] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 12939–12950. IEEE, 2021. 2
- [51] Joanna Waczynska, Piotr Borycki, Joanna Kaleta, Slawomir Konrad Tadeja, and Przemyslaw Spurek. D-MiSo: Editing dynamic 3D scenes using multi-gaussians soup. *CoRR*, abs/2405.14276, 2024. 3, 6, 7
- [52] Joanna Waczynska, Piotr Borycki, Slawomir Konrad Tadeja, Jacek Tabor, and Przemyslaw Spurek. GaMeS: Mesh-based adapting and modification of gaussian splatting. *CoRR*, abs/2402.01459, 2024. 2, 3
- [53] Diwen Wan, Ruijie Lu, and Gang Zeng. Superpoint gaussian splatting for real-time high-fidelity dynamic scene reconstruction. *CoRR*, abs/2406.03697, 2024. 3, 6
- [54] Chunshi Wang, Hongxing Li, and Yawei Luo. SonicGauss: Position-aware physical sound synthesis for 3D gaussian representations, 2025. 1
- [55] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *CoRR*, abs/2106.10689, 2021. 2

- [56] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, pages 1398–1402 Vol.2, 2003. [7](#)
- [57] 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. *CoRR*, abs/2310.08528, 2023. [1](#), [2](#), [6](#), [7](#)
- [58] Tianyi Xie, Zeshun Zong, Yuxing Qiu, Xuan Li, Yutao Feng, Yin Yang, and Chenfanfu Jiang. PhysGaussian: Physics-integrated 3D gaussians for generative dynamics. *CoRR*, abs/2311.12198, 2023. [1](#)
- [59] Jiawei Xu, Zexin Fan, Jian Yang, and Jin Xie. Grid4D: 4D decomposed hash encoding for high-fidelity dynamic gaussian splatting. *CoRR*, abs/2410.20815, 2024. [6](#), [7](#), [8](#)
- [60] Zhiwen Yan, Chen Li, and Gim Hee Lee. NeRF-DS: Neural radiance fields for dynamic specular objects. *CoRR*, abs/2303.14435, 2023. [2](#)
- [61] Bangbang Yang, Chong Bao, Junyi Zeng, Hujun Bao, Yinda Zhang, Zhaopeng Cui, and Guofeng Zhang. NeuMesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. *CoRR*, abs/2207.11911, 2022. [1](#), [2](#)
- [62] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3D gaussians for high-fidelity monocular dynamic scene reconstruction. *CoRR*, abs/2309.13101, 2023. [1](#), [2](#), [3](#), [6](#), [7](#), [8](#)
- [63] Zeyu Yang, Hongye Yang, Zijie Pan, Xiatian Zhu, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4D gaussian splatting. *CoRR*, abs/2310.10642, 2023. [1](#)
- [64] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuewen Ma, Rongfei Jia, and Lin Gao. NeRF-Editing: Geometry editing of neural radiance fields. *CoRR*, abs/2205.04978, 2022. [2](#)
- [65] Cem Yuksel. Sample elimination for generating poisson disk sample sets. *Comput. Graph. Forum*, 34(2):25–32, 2015. [4](#)
- [66] Junyi Zeng, Chong Bao, Rui Chen, Zilong Dong, Guofeng Zhang, Hujun Bao, and Zhaopeng Cui. Mirror-NeRF: Learning neural radiance fields for mirrors with whitted-style ray tracing. *CoRR*, abs/2308.03280, 2023. [1](#)
- [67] Qiang Zhang, Seung-Hwan Baek, Szymon Rusinkiewicz, and Felix Heide. Differentiable point-based radiance fields for efficient view synthesis. *CoRR*, abs/2205.14330, 2022. [1](#)
- [68] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *CoRR*, abs/1801.03924, 2018. [7](#)
- [69] Shuai Zhang, Guanjun Wu, Xinggang Wang, Bin Feng, and Wenyu Liu. Dynamic 2D gaussians: Geometrically accurate radiance fields for dynamic objects. *CoRR*, abs/2409.14072, 2024. [2](#), [3](#), [8](#)
- [70] Licheng Zhong, Hong-Xing Yu, Jiajun Wu, and Yunzhu Li. Reconstruction and simulation of elastic objects with spring-mass 3D gaussians. *CoRR*, abs/2403.09434, 2024. [2](#)