This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

4DSurf: High-Fidelity Dynamic Scene Surface Reconstruction

Renjie Wu[1] Hongdong Li[1,3] Jose M. Alvarez[2] Miaomiao Liu[1]
[1]Australian National University [2]NVIDIA [3]Amazon

{renjie.wu, hongdong.li, miaomiao.liu}@anu.edu.au josea@nvidia.com

Abstract

This paper addresses the problem of dynamic scene surface reconstruction using Gaussian Splatting (GS), aiming to recover temporally consistent geometry. While existing GS-based dynamic surface reconstruction methods can yield superior reconstruction, they are typically limited to either a single object or objects with only small deformations, struggling to maintain temporally consistent surface reconstruction of large deformations over time. We propose "4DSurf", a novel and unified framework for generic dynamic surface reconstruction that does not require specifying the number or types of objects in the scene, can handle large surface deformations and temporal inconsistency in reconstruction. The key innovation of our framework is the introduction of Gaussian deformations induced Signed Distance Function Flow Regularization that constrains the motion of Gaussians to align with the evolving surface. To handle large deformations, we introduce an Overlapping Segment Partitioning strategy that divides the sequence into overlapping segments with small deformations and incrementally passes geometric information across segments through the shared overlapping timestep. Experiments on two challenging dynamic scene datasets, Hi4D and CMU Panoptic, demonstrate that our method outperforms state-of-the-art surface reconstruction methods by 49% and 19% in Chamfer distance, respectively, and achieves superior temporal consistency under sparse-view settings.

1. Introduction

Dynamic surface reconstruction aims to recover temporally consistent 3D geometry from video sequences, serving as a foundation for numerous applications such as digital avatars and virtual reality. Unlike the reconstruction of static surfaces that recovers a single shape, dynamic surface reconstruction must faithfully model continuous deformations for a scene of multiple shapes over time. Here, we address dynamic surface reconstruction from sparse-view video sequences (fewer than 10 views), a practical setup, while maintaining large scene coverage (shown in Fig. 1).

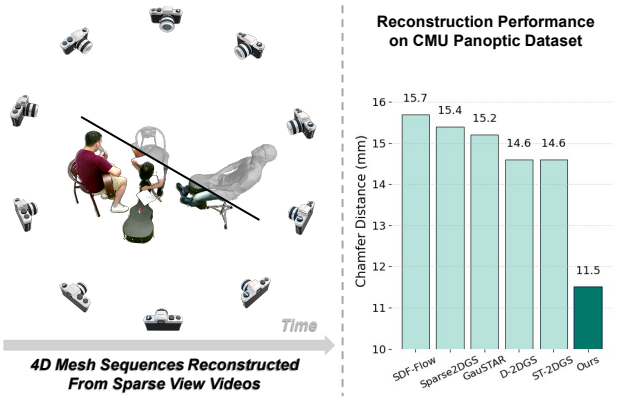

Figure 1. *Left*: Using only a sparse set of input videos. *Right*: Our approach sets a new state-of-the-art benchmark in surface reconstruction for dynamic scenes on the CMU Panoptic dataset [13], compared with recent dynamic surface reconstruction methods (Neural SDF-Flow [26], Sparse2DGS [46], GauSTAR [56], D-2DGS [55], and ST-2DGS [42]).

Recent advances in neural implicit and Gaussian-based representations have significantly advanced dynamic surface reconstruction. Dynamic Neural Radiance Fields (NeRFs) [1, 2, 26, 40, 47] can capture detailed geometry and view-dependent effects, yet suffer from slow optimization and limited scalability. In contrast, Gaussian Splatting (GS)-based approaches [10, 43–45] enable real-time rendering and efficient optimization, but often struggle to reconstruct accurate geometry. To achiever better geometry, several GS–based dynamic surface reconstruction methods [3, 20, 24, 25, 55] are proposed, but they only perform well on the case of a single object with small deformations. Therefore, they struggle with large deformations, which often result in surface jitter and inconsistent geometry over time. Some methods [12, 16, 17] introduce priors such as SMPL-X [32] for human-specific dynamic surface reconstruction or depth and normal priors from pre-trained models [17] to regularize the surface reconstruction.

By contrast, we present "*4DSurf*", a prior-free method to handle an unconstrained scene that may contain multiple (non-) rigid shapes and undergo large deformations. Furthermore, we address the issues of surface jittering and temporal inconsistency in existing methods by constraining the temporal evolution of the surface. It is well known that a static Signed Distance Function (SDF) field can represent arbitrary surface. When extended to dynamic scenes, a time-dependent SDF field can be used to represent the surface that changes consistently over time. Therefore, we can leverage SDF flow [26] to characterize the temporal surface evolution, which is the temporal derivative of time-dependent SDF field and represents how signed distances evolve over the time. The SDF flow constrains that, for any locally smooth surface point, its temporal change equals the negative projection of its scene flow onto the surface normal [26]. To achieve Gaussian-based temporal SDF field representation, we introduce Gaussian Velocity Field to describe the deformation of Gaussians from canonical shape to any timestep by predicting the motion of Gaussians. It explicitly models the scene flow of any surface point on each Gaussian. We then can derive SDF flow from the motion of Gaussians. Moreover, our dynamic Gaussian representation enables the approximation of SDF flow from rendered depth maps, providing an additional constraint from geometry changes in 3D space. Together, these constraints promote temporally consistent surface reconstruction.

To tackle large deformations over long sequence, we adopt an Overlapping Segment Partitioning strategy by dividing the sequence into overlapping segments trained incrementally. The geometry of each segment is modeled by a canonical shape and its associated learnable Gaussian Velocity Field. Furthermore, we develop a Low-Rank Adaptation (LoRA) [8]-based incremental motion tuning to model the Gaussian Velocity Field for each segment, efficiently adapting motion parameters with less storage. This makes our method scalable for handling long motion sequences without losing performance, as shown in our experiments. Our main contributions are summarized as follows:

- We propose a prior-free, generic geometry- and motion-consistent surface reconstruction method "*4DSurf*" of dynamic scenes from sparse view videos.
- We enforce temporally consistent reconstruction by regularizing the surface evolution so that the SDF flow matches that induced by Gaussian deformations.
- We develop an Overlapping Segment Partitioning strategy to handle large deformations, which can mitigate error accumulation and enhance scalability, and an incremental motion tuning variant that further reduces storage usage while maintaining competitive performance.

Extensive experiments on two challenging dynamic scene datasets [13, 54] with multiple shapes and varying deformations, demonstrating our method achieves state-of-the-art performance and strong generality on reconstructing dynamic scene surface from sparse view videos.

2. Related Work

Novel View Synthesis for Dynamic Scenes. NeRFs [27] have greatly advanced novel view synthesis and been extended to dynamic scenes. Dynamic NeRFs can be roughly divided into three types: (1) those accelerating training and rendering via representation decoupling or hash encoding [11, 22, 35, 39]; (2) those using deformation fields with a canonical space [30, 31, 34, 38]; and (3) incremental ones reusing a static NeRF from previous frames [18, 36]. Recent 3D-GS [15] achieves a better efficiency–quality trade-off and inspires dynamic GS. Some model dynamics via Gaussian-based canonical spaces and deformation fields [10, 52], while others incorporate hash encoding or decoupled representations for faster optimization [45, 48]. Incremental schemes reuse Gaussians from prior timesteps [5, 37, 50, 56]. Other works extend Gaussians into higher-dimensional space [6, 53], enrich Gaussian attributes [21, 43], or leverage priors such as optical flow or depth from foundation models [23, 44]. However, most of them neglect geometric representation over time, resulting in temporally inaccurate and inconsistent geometry.

Dynamic Surface Reconstruction. Early works [14, 19, 29, 41, 49] deform predefined templates, while NeRF-based methods [1, 2, 26, 40, 47] learn implicit fields. However, template-based methods rely heavily on templates, and NeRF-based methods suffer from slow training and limited scalability. Recently, many GS-based dynamic surface reconstruction methods have been proposed. DG-Mesh [24] integrates deformable Gaussians with a differentiable Poisson solver [33], while later methods adopt 2D-GS [42, 55] or planar-based Gaussians [3]. MaGS [25] jointly optimizes mesh vertices and Gaussians, DGNS [20] couples SDF-based NeRF with dynamic GS, and GauSTAR [56] adaptively regenerates Gaussians for finer surfaces. Human-specific reconstruction methods [12, 16, 17] often depend on priors such as SMPL-X [32]. Despite recent progress, above methods are still confined to specific scenarios (e.g., single-object or multi-human setups with dense views) and rely on priors like depth, optical flow, normals, or SMPL-X. Some recent works [26, 42] relax such foundation priors but still struggle with large deformations. In contrast, we pursue a generic dynamic scene surface reconstruction method from sparse view videos, which is not dedicated to any specific objects and can handle large deformations.

3. Preliminary: 2D Gaussian Splatting

Our goal is to model geometry, therefore we adopt 2DGS [9] in our framework, as it can model better geometry compared to 3DGS [15]. Each primitive is defined by

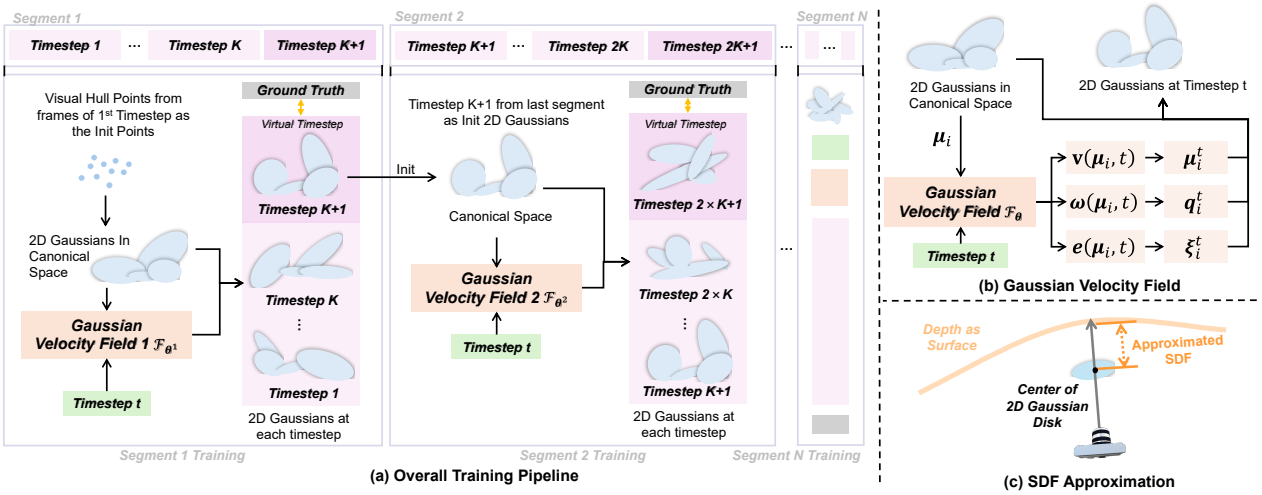

Figure 2. Overview: **(a) Overall Training Pipeline.** We first divide the sequence into N segments, each containing $K+1$ timesteps with one overlapping virtual timestep. For the 1st segment, the initialization is derived from the visual hull reconstructed from all frames of its first timestep. After training the first segment, the Gaussians of virtual timestep serves as the initialization for training the next segment. Each segment maintains its own canonical space and Gaussian Velocity Field. **(b) Gaussian Velocity Field.** Given the Gaussian center $\boldsymbol{\mu}_i$ in the canonical space and a specific timestep t, the Gaussian Velocity Field $\mathcal{F}_{\boldsymbol{\theta}}(\cdot)$ predicts its velocity $\mathbf{v}(\boldsymbol{\mu}_i,t)$, angular velocity $\boldsymbol{\omega}(\boldsymbol{\mu}_i,t)$ and expansion velocity $\boldsymbol{e}(\boldsymbol{\mu}_i,t)$ at timestep t. These are then converted to position $\boldsymbol{\mu}_i^t$, rotation $\boldsymbol{q}_i^t$, and scale $\boldsymbol{\xi}_i^t$, which are fed into the differentiable rasterizer for image rendering. **(c) SDF Approximation.** Following previous works [7, 28], we compute the distance between the center and its corresponding depth point to estimate the signed distance.

a center $\boldsymbol{\mu} \in \mathbb{R}^3$, two orthogonal tangent vectors $\mathbf{t}_u, \mathbf{t}_v$ (parameterized via quaternion $\boldsymbol{q}$), and a scale vector $\boldsymbol{\xi} = (\xi_u, \xi_v)$. The primitive is defined in the local tangent plane as $P(u,v) = \boldsymbol{\mu} + \xi_u \mathbf{t}_u u + \xi_v \mathbf{t}_v v$, and the Gaussian value can be calculated by $\mathcal{G}(\mathbf{u}) = \exp(-(u^2+v^2)/2)$. Each primitive also carries an opacity α and a view-dependent color $\mathbf{c}$ represented by spherical harmonics. The pixel color is rendered via alpha blending: $\hat{\mathbf{c}}(\mathbf{p}) = \sum_i \mathbf{c}_i \mathrm{w}_i, \mathrm{w}_i = \alpha_i \mathcal{G}_i(\mathbf{u}(\mathbf{p})) \prod_{j=1}^{i-1} (1 - \alpha_j \mathcal{G}_j(\mathbf{u}(\mathbf{p})))$. Depth is rendered as: $\hat{D}(\mathbf{p}) = \sum_{i=1} \mathrm{w}_i \, \mathrm{d}_i / (\sum_{i=1} \mathrm{w}_i + \epsilon)$, where d_i is the intersection depth between the ray and the i-th Gaussian disk.

4. Method

We present our overall training pipeline in Fig. 2(a). In this section, we first describe our SDF Flow Regularization (Sec. 4.1) induced by Gaussian Velocity Field. Then, we introduce our Overlapping Segment Partitioning strategy that can handle large deformations (Sec. 4.2). Next, we introduce the incremental motion tuning to minimize storage while maintaining competitive performance (Sec. 4.3). Lastly, we present the overall training objective (Sec. 4.4).

4.1. SDF Flow Regularization

Our method enforces consistency between SDF flow derived from the motion of the Gaussians and SDF flow estimated from 3D geometric changes, jointly yielding temporally consistent surface evolution. Below we first revisit the definition of SDF flow in [26] followed by our defined Gaussian Velocity Field and the derived regularization.

Revisit SDF Flow from Point-Wise Motion.

SDF field for static scene defines a scalar field $s(\hat{\mathbf{x}}) : \mathbb{R}^3 \rightarrow \mathbb{R}$, where $s(\hat{\mathbf{x}})$ denotes the signed distance from point $\hat{\mathbf{x}}$ to the closest surface. The surface is given by the zero-level set $\{\hat{\mathbf{x}} \mid s(\hat{\mathbf{x}}) = 0\}$, and the normal is defined as $\mathbf{n}(\hat{\mathbf{x}}) = \nabla_{\hat{\mathbf{x}}} s(\hat{\mathbf{x}})$. When extended to dynamic scenes, the SDF field becomes time-dependent, $s(\hat{\mathbf{x}}, t) : \mathbb{R}^3 \times \mathbb{R} \rightarrow \mathbb{R}$, and its temporal derivative $\frac{\partial s}{\partial t}$ characterizes how the signed distance evolves. Following [26], when $\Delta t \rightarrow 0$, for a locally rigid point with line velocity $\mathbf{v}$ and angular velocity $\boldsymbol{\omega}$, its SDF flow is given by:

$$\frac{\partial s}{\partial t} = \lim_{\Delta t \to 0} \frac{\Delta s}{\Delta t} = -\frac{\partial \hat{\mathbf{x}}}{\partial t}^{\top} \mathbf{n}(\hat{\mathbf{x}}) = -(\boldsymbol{\omega} \times \hat{\mathbf{x}} + \mathbf{v})^{\top} \mathbf{n}(\hat{\mathbf{x}}), \tag{1}$$

which directly links SDF changes to the underlying scene flow $\frac{\partial \hat{\mathbf{x}}}{\partial t}$, demonstrating how the surface evolves over time.

However, Eq. (1) only considers point-wise motion independently on the surface and its relations to its SDF changes, which cannot be directly applied for the points on the Gaussians in the canonical space. Therefore, we introduce the Gaussian Velocity Field to describe the deformations of Gaussians from canonical shape to any timestep by modeling the motion of Gaussians. With such Gaussian Velocity Field, we can explicitly model the scene flow of any surface point on each Gaussian. Then we can derive the

relationship between the motion of the Gaussian and its induced SDF changes to all surface points of the Gaussian. In the following, we first describe the Gaussian Velocity Field and then derive the SDF flow from the motion of Gaussians.

Gaussian Velocity Field

To explicitly describe the scene flow of Gaussians, we use Gaussian Velocity Field to predict motions of Gaussians to model deformations at each timestep instead of directly predicting the absolute deformation. The Fig. 2(b) shows the overall procedure. Specifically, given the center $\boldsymbol{\mu}_i$ of the i^{th} Gaussian in the canonical space and the timestep $t \in \mathbb{R}$, the Gaussian Velocity Field $\mathcal{F}_{\boldsymbol{\theta}}$ predicts three types of motion parameters: velocity $\mathbf{v}(\boldsymbol{\mu}_i, t) \in \mathbb{R}^3$, angular velocity $\boldsymbol{\omega}(\boldsymbol{\mu}_i, t) \in \mathbb{R}^3$, and expansion velocity $\boldsymbol{e}(\boldsymbol{\mu}_i, t) \in \mathbb{R}^2$ (velocity of scale changes). Formally, $\mathbf{v}(\boldsymbol{\mu}_i, t),\ \boldsymbol{\omega}(\boldsymbol{\mu}_i, t),\ \boldsymbol{e}(\boldsymbol{\mu}_i, t) = \mathcal{F}_{\boldsymbol{\theta}}\big(\gamma(\boldsymbol{\mu}_i), \gamma(t)\big)$, where $\gamma(\cdot)$ denotes the positional encoding function. Then, we can obtain the following parameters describing the deformation of the i^{th} Gaussian at timestep t: $\boldsymbol{\mu}_i^t = \boldsymbol{\mu}_i + \mathbf{v}(\boldsymbol{\mu}_i, t)\, t, \boldsymbol{q}_i^t = \phi\big(\boldsymbol{\omega}(\boldsymbol{\mu}_i, t)\, t\big) \otimes \boldsymbol{q}_i, \boldsymbol{\xi}_i^t = \boldsymbol{\xi}_i + \boldsymbol{e}(\boldsymbol{\mu}_i, t)\, t$, where $\phi(\cdot)$ represents a function that can convert rotation vectors to quaternions, and $\otimes$ denotes quaternion multiplication. These Gaussian parameters can enable differentiable rasterization rendering at arbitrary timestep within each segment.

SDF Flow from Motion of Gaussians

Given the definition of above Gaussian Velocity Field, we then can derive the SDF flow induced by the motion of Gaussians, which can be introduced as a constraint for ensuring temporally consistent reconstruction of surfaces.

Assumption. *At timestep t, the motion of a 3D point $\mathbf{x} \in \mathbb{R}^3$ in the canonical space of a Gaussian can be approximated by a rigid transformation parameterized by rotation $\mathrm{R}^t \in SO(3)$ and translation $\mathrm{T}^t \in \mathbb{R}^3$. After an interval $\Delta t \to 0$, we can get the displacement of the point:*

$$\mathbf{x}^{t+\Delta t} - \mathbf{x}^t = \Delta\mathrm{R}\,\mathrm{R}^t\mathbf{x} + \Delta\mathrm{T}, \tag{2}$$

where $\Delta\mathrm{R} \in SO(3)$ and $\Delta\mathrm{T} \in \mathbb{R}^3$ denote the incremental rotation and translation.

Theorem. *For any 3D point $\mathbf{x}$ in the canonical space of a Gaussian, the temporal derivative $\mathbf{f}$ of its SDF equals the negative projection of the induced scene flow onto the surface normal $\mathbf{n}$ (normal of the Gaussian):*

$$\mathbf{f} = -(\boldsymbol{\omega} \times \mathrm{R}^t\mathbf{x} + \mathbf{v})^\top \mathbf{n}(\mathrm{R}^t\mathbf{x}), \tag{3}$$

where $\mathbf{f} \in \mathbb{R}$ denotes the SDF flow, $\boldsymbol{\omega} \in \mathbb{R}^3$ and $\mathbf{v} \in \mathbb{R}^3$ are the angular and linear velocities of the points on the Gaussian, and $\mathbf{n}(\mathrm{R}^t\mathbf{x})$ is the surface normal at $\mathrm{R}^t\mathbf{x}$.

Please refer to Supplementary Material Sec. 7 for a detailed derivation. Note that, we compute the SDF flow at center of each Gaussian in practice for efficient regularization.

SDF Flow from Geometry Changes

On the other hand, SDF flow can be derived by measuring the changes of SDF values of each point in the 3D space. The rendered depth can be interpreted as a pseudo-surface of a time-dependent SDF. Here we look at the Gaussian centers at time t and use these signed distances as our approximated SDF values, for convenient computation and compatibility with the existing GS rasterizer. Indeed, ideally, the SDF values of Gaussian centers are zero for time t. However, SDF flow measures their changes due to the evolving of the surface induced by the motion of Gaussians. We follow previous works [7, 28] to do the SDF approximation efficiently, which also leverages depth map as a pseudo-surface for SDF estimation. We exploit the temporal derivative of those SDF values. Thus, given a Gaussian center $\boldsymbol{\mu}_i^t$, we can approximate the SDF value $\tilde{s}(\boldsymbol{\mu}_i^t, t)$ at timestep t as:

$$\tilde{s}(\boldsymbol{\mu}_i^t, t) = \hat{D}(\mathbf{p}^*, t) - d(\boldsymbol{\mu}_i^t, t), \tag{4}$$

where $d(\boldsymbol{\mu}_i^t, t) \in \mathbb{R}$ denotes the distance from the camera origin to the $\boldsymbol{\mu}_i^t$ along the optical axis, and $\hat{D}(\mathbf{p}^*, t)$ represents the corresponding surface depth point at the projected pixel $\mathbf{p}^*$ on the depth map. The process can be referred to Fig. 2(c). Then we can get its temporal derivative to obtain the SDF flow $\tilde{\mathbf{f}}_i \in \mathbb{R}$ from geometry changes:

$$\tilde{\mathbf{f}}_i^t = \frac{\partial \tilde{s}(\boldsymbol{\mu}_i^t, t)}{\partial t} = \frac{\partial \hat{D}(\mathbf{p}^*, t)}{\partial t} - \frac{\partial d(\boldsymbol{\mu}_i^t, t)}{\partial t}. \tag{5}$$

See Supplementary Material Sec. 8 for more details of above formula. By matching SDF flow from the motion of Gaussians and geometry changes, we can achieve temporally consistent surface evolution. Formally, SDF flow regularization $\mathcal{L}_{\text{flow}} = \sum_i |\mathbf{f}_i^t - \tilde{\mathbf{f}}_i^t|$, where $\mathbf{f}_i^t$ and $\tilde{\mathbf{f}}_i^t$ denote the SDF flow from the motion of Gaussians and geometry changes of the i^{th} Gaussian at timestep t, respectively.

4.2. Segment Partition

Existing works [20, 25, 55] can reconstruct dynamic scenes with small deformations using one deformation field and canonical shape. However, they struggle with large deformations. To address this issue, we subdivide the video sequence into consecutive segments train the the model incrementally. We allow overlaps between neighboring segments, where each segment shares a virtual timestep with the following one (the first timestep of the next segment). Thus, each segment contains $K+1$ timesteps. We name this strategy as Overlapping Segment Partitioning. Within each segment, we model the geometry using a canonical shape along with a Gaussian Velocity Field that describes the deformation. After training one segment, the Gaussians at its virtual timestep are used as the initialization for training the next segment, ensuring consistent geometry can pass to later segments. Also, new Gaussians will be created in previously unseen regions. This process is shown in Fig. 2(a).

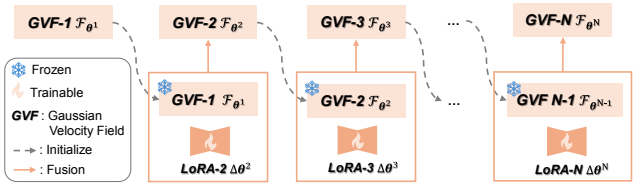

Figure 3. Incremental Motion Tuning (IMT). After training the Gaussian Velocity Field of the 1st segment, for the later N^{th} segment ($N \geq 2$), its Gaussian Velocity Field $\mathcal{F}_{\boldsymbol{\theta}^N}$ is initialized from the $\boldsymbol{\theta}^{N-1}$ of previous segment and fine-tuned using LoRA $\Delta\boldsymbol{\theta}^N$.

During mesh extraction, the reconstructed mesh of the virtual timestep is discarded to avoid duplication.

4.3. Incremental Motion Tuning

The Overlapping Segment Partitioning-based incremental training strategy can mitigate error accumulation and handle large deformations, but it increases the number of Gaussian velocity fields and canonical shapes, leading to higher storage costs. Merging canonical shapes of all segments is non-trivial, so that we aim to reduce storage of Gaussian Velocity Fields. Since all segments depict the same dynamic scene, their primary difference lies in motion. We propose Incremental Motion Tuning (IMT), which incrementally adapts the Gaussian Velocity Field from the previous segment in a parameter-efficient manner. To be specific, after training the Gaussian Velocity Field of the first segment, for later N^{th} segment ($N \geq 2$), we fine-tune the Gaussian Velocity Field from its previous segment by using LoRA [8] instead of learning a new one from scratch (see Fig. 3). Formally, the Gaussian Velocity Field parameters of the N^{th} ($N \geq 2$) segment is defined as: $\boldsymbol{\theta}^N = \boldsymbol{\theta}^{N-1} + \Delta\boldsymbol{\theta}^N$, where $\boldsymbol{\theta}^N$ denotes the Gaussian Velocity Field parameters of segment N, and $\Delta\boldsymbol{\theta}^N = \mathbf{A}^N\mathbf{B}^N$ represents the low-rank update, with $\mathbf{A}^N \in \mathbb{R}^{\mathbf{d}\times\mathbf{r}}$ and $\mathbf{B}^N \in \mathbb{R}^{\mathbf{r}\times\mathbf{d}}$, where $\mathbf{r} \ll \mathbf{d}$. By storing only $\Delta\boldsymbol{\theta}^N$ for each segment, IMT can reduce the storage cost while maintain competitive performance.

4.4. Optimization

We employ a photometric loss $\mathcal{L}_{\text{img}}$, three regularization losses, and a mask loss $\mathcal{L}_{\text{m}}$. $\mathcal{L}_{\text{img}}$ follows prior works [10, 55] and combines an $\mathcal{L}_1$ with D-SSIM term. We also adopt the normal alignment loss $\mathcal{L}_{\text{n}}$ and depth distortion loss $\mathcal{L}_{\text{d}}$ from 2DGS [9], which align splat normals with depth-derived normals for coherent surface and regularize geometry by encouraging concentrated ray–splat intersections to reduce depth ambiguity. Then, followed by our SDF flow regularization $\mathcal{L}_{\text{flow}}$. Finally, we incorporate a mask loss to reduce background artifacts. Formally, $\mathcal{L}_{\text{m}} = \mathcal{L}_1(\mathbf{M}^*, \mathbf{M})$, where $\mathbf{M}^*$ and $\mathbf{M}$ denote rendered and ground-truth alpha masks. Here is the total training objective $\mathcal{L}_{\text{total}}$:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{img}} + \lambda_1\mathcal{L}_{\text{n}} + \lambda_2\mathcal{L}_{\text{d}} + \lambda_3\mathcal{L}_{\text{flow}} + \lambda_4\mathcal{L}_{\text{m}}, \quad (6)$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ denote the weights of each loss term.

5. Experiments

5.1. Settings

Datasets. We conduct experiments on two public datasets: CMU Panoptic [13] and Hi4D [54]. Following [26], we use four scenes from the CMU Panoptic: `Ian3`, `Haggling b2`, `Band1`, and `Pizza1`, each captured with a circular rig of 10 RGB-D cameras at 1920 × 1080 resolution. Each scene spans 24 timesteps and provides ground-truth point clouds. From the Hi4D, we select six scenes: `Backhug02`, `Basketall13`, `Fight17`, `Football18`, `Talk22`, and `Cheers37`, each is captured with 8 RGB cameras at 940 × 1280 resolution. On average, each sequence contains 118 timesteps and each timestep is annotated with a high-quality textured 3D mesh. Compared with CMU Panoptic, Hi4D features larger motions, longer sequences, and multi-human scene. Since our goal is surface reconstruction, we use all RGB views for training and only evaluate the meshes.

Evaluation Metrics. We follow prior works [26, 42, 46] and adopt Chamfer Distance (CD) that measured in terms of Accuracy ↓ (Acc), Completeness ↓ (Comp), and Overall ↓ (the average of Acc and Comp) as our comparison metrics.

Baselines. We focus on generic dynamic scene surface reconstruction without relying on external priors, with the emphasis on GS-based methods. We mainly compare our method against GS-based dynamic surface reconstruction methods, including Space-Time-2DGS [42], Dynamic-2DGS [55], and GauSTAR [56]. Note that GauSTAR reconstructs dynamic scenes from dense-view videos using optical flow and depth priors, and performs incremental learning at each timestep. For completeness, we also compare some NeRF-based dynamic surface reconstruction methods on CMU Panoptic dataset, they are NDR [2] and Neural SDF Flow [26], as they require long training time. Some dynamic novel view synthesis methods are also included, such as Tensor4D [35], 4DGS [45], SC-GS [10], and Free-TimeGS [43], which demonstrate strong novel-view synthesis performance, with FreeTimeGS representing the state-of-the-art in novel view synthesis from multi-view dynamic scenes. We further consider Sparse2DGS [46], a state-of-the-art static surface reconstruction method from sparse views, which we apply independently to each timestep for fair comparison. We exclude Space-Time-2DGS on the Hi4D due to unavailable code, and also omit MonoFusion [44], DG-Mesh [24], and MaGS [25] for their reliance on strong priors or task-specific assumptions.

Implementation Details. We build our model upon [55] and use same parameters for the optimization of Gaussians. The initial point cloud is obtained by constructing a visual hull [51] from all-view foreground masks at the

Table 1. CMU Panoptic [13] comparisons. We evaluate performance with Chamfer Distance (unit: mm). The top three results for each metric are highlighted with , , and , respectively. Ours consistently achieves the best performance on the Overall metric.

Methods	Band1			Ian3			Haggling b2			Pizza1		
	Acc ↓	Comp ↓	Overall ↓	Acc ↓	Comp ↓	Overall ↓	Acc ↓	Comp ↓	Overall ↓	Acc ↓	Comp ↓	Overall ↓
NDR [2]	15.9	23.7	19.8	21.8	20.7	21.3	12.5	22.8	17.7	17.7	25.0	21.3
Tensor4D [35]	17.1	29.2	23.2	15.4	22.8	19.1	13.7	25.3	19.5	18.3	23.5	22.9
Neural SDF-Flow [26]	13.0	21.4	17.2	14.1	17.5	15.8	8.3	18.6	13.5	11.5	20.6	16.1
4DGS [45]	12.4	22.4	17.3	8.8	17.2	13.0	9.0	19.9	14.4	12.1	22.1	17.1
SC-GS [10]	12.2	22.2	17.2	8.4	17.3	12.8	8.3	19.2	13.8	11.7	22.1	16.9
FreeTimeGS [43]	12.8	22.9	17.9	8.8	17.3	13.1	10.6	19.9	15.3	12.9	22.5	17.7
Sparse2DGS [46]	12.6	22.7	17.7	7.8	17.2	12.5	8.4	20.3	14.4	11.1	22.7	16.9
Space-Time-2DGS [42]	11.9	20.9	16.4	8.6	16.5	12.6	8.6	18.9	13.7	11.2	20.4	15.8
GauSTAR [56]	14.2	21.1	17.6	10.1	17.3	13.7	11.0	18.7	14.8	11.4	17.9	14.7
Dynamic-2DGS [55]	12.1	19.9	16.0	9.5	15.4	12.5	9.3	18.0	13.7	12.6	19.8	16.2
Ours w IMT-64	11.0	14.5	12.8	8.2	12.6	10.4	8.5	13.5	11.0	10.8	13.5	12.1
Ours wo IMT	11.1	14.4	12.7	8.3	12.6	10.5	8.4	13.3	10.8	10.9	13.4	12.2

Table 2. Hi4D [54] comparisons. We evaluate performance using Chamfer Distance (unit: cm). The top three results for each metric are highlighted in , , and , respectively. Ours significantly outperforms all baselines on the Overall metric.

Methods	Cheers37			Talk22			Football18			Fight17			Basketball13			Backhug02		
	Acc ↓	Comp ↓	Overall ↓	Acc ↓	Comp ↓	Overall ↓	Acc ↓	Comp ↓	Overall ↓	Acc ↓	Comp ↓	Overall ↓	Acc ↓	Comp ↓	Overall ↓	Acc ↓	Comp ↓	Overall ↓
4DGS [45]	3.05	1.43	2.24	3.73	3.75	3.74	3.05	2.22	2.63	2.53	2.94	2.73	2.96	2.53	2.74	3.05	5.36	4.21
SC-GS [10]	1.58	0.97	1.27	1.62	3.21	2.41	1.47	1.81	1.64	1.58	2.65	2.11	1.72	1.90	1.81	1.87	5.16	3.51
FreeTimeGS [43]	2.34	1.22	1.78	3.21	2.21	2.71	2.20	1.64	1.92	3.71	2.14	2.93	3.04	3.10	3.07	2.61	3.81	3.21
Sparse2DGS [46]	2.66	0.96	1.81	3.21	1.86	2.54	1.94	0.98	1.46	1.27	0.75	1.01	4.87	2.10	3.48	2.00	1.59	1.79
GauSTAR [56]	2.10	2.52	2.31	2.74	1.96	2.35	2.74	0.49	1.62	3.32	2.25	2.79	4.15	1.04	2.59	2.18	1.99	2.09
Dynamic-2DGS [55]	2.62	1.96	2.29	2.75	0.88	1.82	1.76	0.44	1.10	2.33	1.20	1.77	3.55	0.98	2.27	1.59	0.55	1.07
Ours w IMT-64	0.71	0.25	0.48	1.09	0.59	0.84	0.81	0.30	0.56	0.82	0.46	0.64	1.09	0.69	0.89	0.93	0.60	0.76
Ours wo IMT	0.70	0.24	0.47	1.06	0.62	0.84	0.73	0.29	0.51	0.91	0.35	0.63	0.90	0.50	0.70	1.01	0.68	0.84

first timestep. The segment size is set to 5 with one virtual timestep, and each segment is trained for 30K iterations (about 30 mins). The network structure of Gaussian Velocity Field is similar to the deformation network in [52]. If with IMT-64, only the three heads are trainable, while other linear layers are fine-tuned using LoRA (rank=64). All experiments are conducted on one NVIDIA RTX 3090Ti GPU. For mesh extraction, a TSDF volume [4] is constructed by fusing RGB-D images from all training views following [9, 55]. More details of datasets, baselines and implementation are in Supplementary Material Sec. 10.

5.2. Comparisons

We compare our methods against recent GS-based dynamic surface reconstruction baselines on CMU Panoptic [13] in Tab. 1 and Hi4D [54] in Tab. 2. We consider two versions of our method: one learns Gaussian Velocity Fields from scratch for each segment (Ours wo IMT), while another version (Ours w IMT-64) employs IMT-64 by fine tuning the Gaussian Velocity Fields. We present qualitative comparisons on both CMU in Fig. 4 and Hi4D in Fig. 5. Baselines used for qualitative comparisons are selected from three categories introduced in Sec. 5.1, with one representative baseline chosen from each category, they are: Dynamic-2DGS [55], Sparse 2DGS [46], and FreeTimeGS [43].

CMU Panoptic Dataset. As shown in Tab. 1, our two methods consistently outperforms all the baselines and achieves the lowest Overall values across all four scenes. Compared with second-best results, Ours w/o IMT surpasses them by over 19%. We visualize the scene `Band1` and `Ian3` in Fig. 4. Since `Band1` involves a more complex scene, we show two different viewpoints at two distinct timesteps. Our methods consistently produce smoother and more detailed reconstructions, whereas the baselines show uneven, noisy, and incomplete surfaces as highlighted in Fig. 4.

Hi4D Dataset. As shown in Tab. 2, our two methods also achieve lowest Overall values on all six scenes. Ours wo IMT improves upon second-best results by over 49%. We visualize reconstructed scenes `Basketball13` and `Fight17` at two long-range timesteps from the same viewpoint in Fig. 5. It can be clearly observed that our methods reconstruct smoother surfaces and more complete meshes than other baselines. Last but not least, both Dynamic-2DGS and FreeTimeGS exhibit severe geometric inconsistencies and accumulated errors over timesteps. Also, their reconstructed surfaces become noticeably coarse and jittering. While our methods maintain best geometric consistency and smooth surface across long-range timesteps.

5.3. Ablation Studies & Discussion

We perform ablation studies and discussion on Hi4D [54] because it has longer sequences with larger deformations.

Ablation Studies. We conduct ablation studies on each component and present quantitative and corresponding qualitative results in Tab. 3 and Fig. 6. We first replace the deformation field in Dynamic-2DGS [55] with our proposed *Gaussian Velocity Field* while keeping the depth and normal regularization, photorealistic and mask losses unchanged (row **a**, Fig. 6(**a**)). Subsequently, we incorporate the *SDF Flow Regularization* (row **b**, Fig. 6(**b**)) to en-

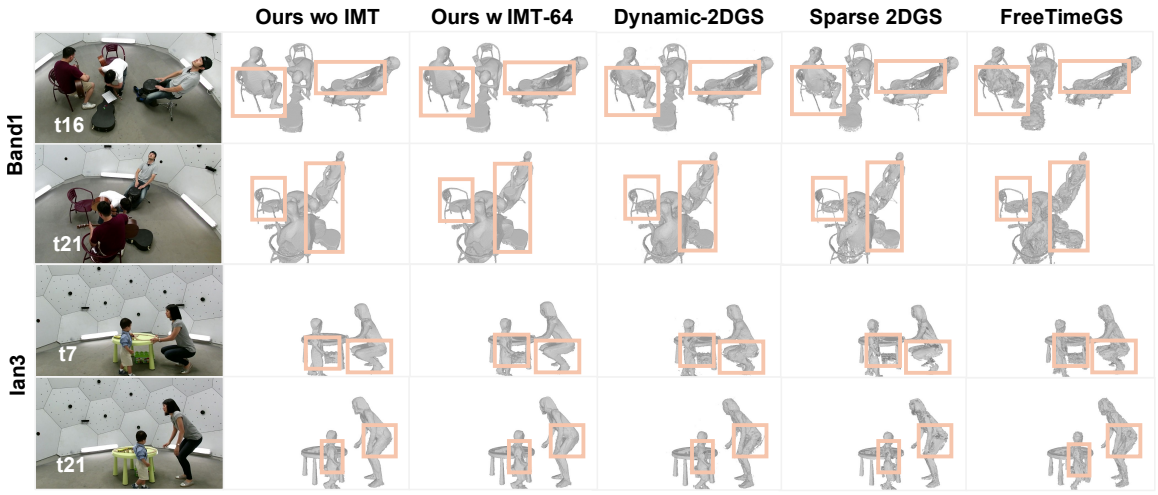

Figure 4. Qualitative results on CMU Panoptic [13]. We compare our methods with three baselines (Dynamic-2DGS [55], Sparse2DGS [46], FreeTimeGS [43]) at two timesteps of the `Band1` and `Ian3` scene. Bounding boxes highlight major differences.

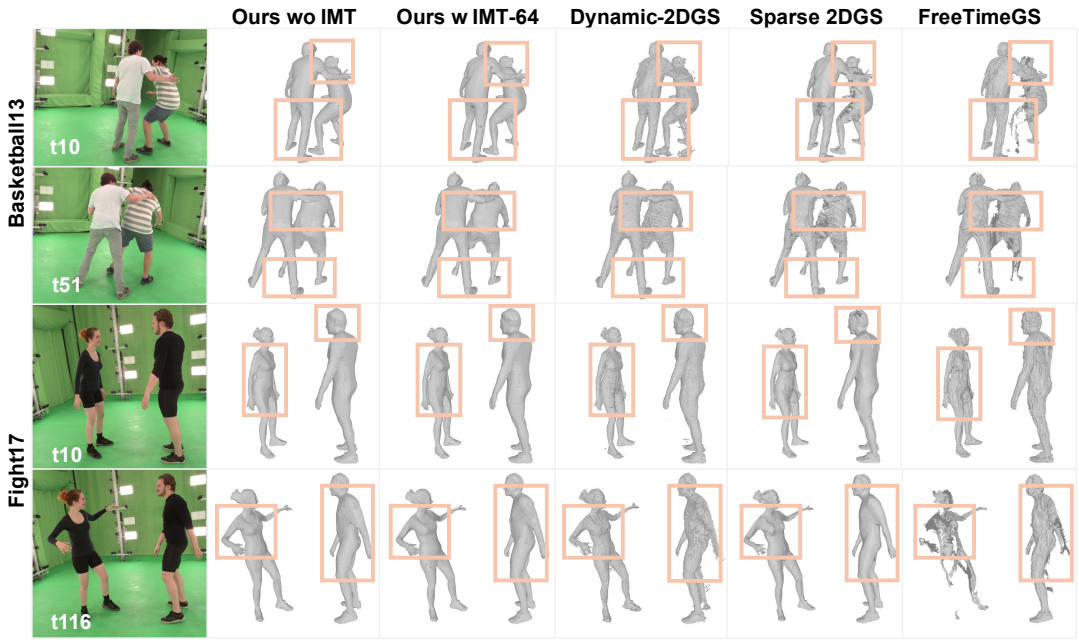

Figure 5. Qualitative results on Hi4D [54]. We compare our methods with three baselines (Dynamic-2DGS [55], Sparse2DGS [46], FreeTimeGS [43]) at two timesteps of the `Basketball13` and `Fight17` scene. Bounding boxes highlight major differences.

courage consistent surface evolution from motion of Gaussians and geometry changes across time, which leads to a noticeable improvement in reconstruction performance. As shown in Fig. 6(**b**), using regularization can be much smoother than not using this regularization term and artifacts are reduced. We then apply *Independent Segment Partitioning*, dividing the sequence into independent segments of size 5 (row **c**, Fig. 6(**c**)). This design effectively reduces error accumulation in both Acc, Comp and Overall. However, lacking geometric information passing across segments, we introduce the *Overlapping Segment Partitioning* strategy (row **d**, Fig. 6(**d**)), which further improves performance. Comparing Fig. 6(**c**) and (**d**), it is easy to see that with the Gaussians of the previous segment's virtual timestep as the initialization of the next segment, the geometric quality and consistency are improved, and the surface becomes smoother. Finally, adopting *IMT-64* (row **e**, Fig. 6(**e**)) achieves comparable performance quantitatively

Table 3. Ablation Studies on the Hi4D dataset [54]. We calculate the average of the three metrics (Acc, Comp, and Overall) for the six scenes. GVF: Gaussian Velocity Field. SF-Reg: SDF-Flow Regularization. I-Segment: Independent Segment Partitioning. O-Segment: Overlapping Segment Partitioning. IMT-64: Incremental Motion Tuning with LoRA rank 64.

	GVF	SF-Reg	I-Segment	O-Segment	IMT-64	Acc ↓	Comp ↓	Overall ↓
(a)	√					1.75	1.23	1.49
(b)	√	√				1.18	0.86	1.02
(c)	√	√	√			1.06	0.47	0.77
(d)	√	√		√		0.89	0.45	0.67
(e)	√	√		√	√	0.91	0.48	0.70

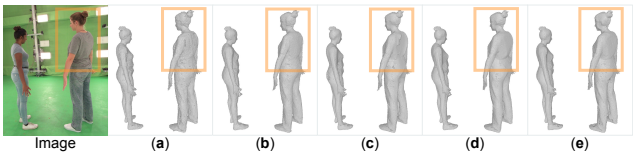

Figure 6. Qualitative comparison of each ablation study on scene `Cheers37`. Subscripts correspond to rows in Tab. 3. (**a**): Gaussian Velocity Field with normal & depth regularization from [9], photorealistic and mask losses. (**b**): Add SDF Flow Regularization based on (**a**). (**c**): Training (**b**) with Independent Segment Partitioning. (**d**): Training (**b**) with Overlapping Segment Partitioning. (**e**): Adopting Incremental Motion Tuning (LoRA rank 64) on (**d**).

Table 4. Temporal stability comparison on the Hi4D dataset [54]. STD: standard deviation. It shows the average STD of the three metrics (Acc, Comp, and Overall) across six scenes. The best and the second-best are highlighted in **bold** and underlined.

Methods	Acc STD ↓	Comp STD ↓	Overall STD ↓
Dynamic-2DGS [55]	1.50	1.30	1.19
GauSTAR [56]	0.53	4.39	2.70
Sparse2DGS [46]	0.95	0.50	0.68
Ours w IMT-64	<u>0.28</u>	<u>0.37</u>	<u>0.28</u>
Ours wo IMT	**0.22**	**0.17**	**0.18**

Table 5. Different LoRA ranks comparison on Hi4D dataset [54]. It shows the average of the three metrics (Acc, Comp, and Overall) in six scenes under different LoRA ranks.

Method	Acc ↓	Comp ↓	Overall ↓
Ours w IMT-16	1.07	0.70	0.89
Ours w IMT-32	1.00	0.64	0.82
Ours w IMT-64	0.91	0.48	0.70
Ours wo IMT	0.89	0.45	0.67

and qualitative while reducing storage. Below, we compare the stability of our method with that of other baselines and the performance of IMT with varying ranks. Additionally, please see Supplementary Material for more experiments.

Temporal Stability. We evaluate the temporal stability using the average standard deviation (STD) of three metrics on six scenes of Hi4D dataset [54]. Tab. 4 shows that our methods achieve the lowest average STD across all metrics, demonstrating superior temporal stability compared with other baselines. Specifically, our method without IMT-64 reduces the STD of CD (Overall metric) to only 0.18, significantly lower than Sparse2DGS [46] (0.68) and Dynamic-2DGS [55] (1.19). This indicates that the reconstructed surfaces from our methods exhibit substantially fewer surface jitter and are more temporally consistent.

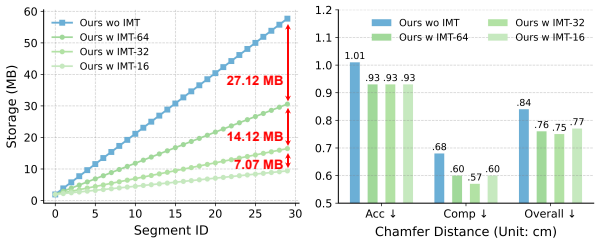

Figure 7. LoRA rank and storage analysis on scene `Backhug02`. *Left*: As the number of segments increases, *Ours wo IMT* exhibits growing storage of Gaussian velocity fields, while *Ours w IMT-*{*LoRA-rank*} effectively curb the storage growth. *Right*: For different LoRA ranks, *Ours w IMT* maintains strong performance (even at rank 16), achieving competitive results to *Ours wo IMT*.

LoRA Rank & Storage Analysis. We try different LoRA ranks (16, 32, 64) on fine-tuning Gaussian Velocity Fields using IMT, excluding rank 128 as its parameters approach full fine-tuning. Results in Tab. 5 show that lower ranks slightly reduce performance due to limited capacity but still maintain competitive results. Take the scene `Backhug02` as an example. While canonical shape storage per segment remains stable on all settings (38–39 MB), Fig. 7(left) shows all Gaussian Velocity Fields' storage drops from 57.7 MB (Ours wo IMT) to 30.6, 16.5, and 9.4 MB for IMT-64, IMT-32, and IMT-16, respectively. Fig. 7(right) demonstrates that IMT maintains well reconstruction quality with different ranks, making IMT scalable for long sequences.

6. Conclusion

In this paper, we introduce a novel method *"4DSurf"* for generic dynamic scene surface reconstruction from sparse view videos. Our key idea is to regularize geometry evolution by matching the SDF flow from motion of Gaussians and geometry changes, which enforces temporally consistent surface evolution. To further alleviate deformation error accumulation and storage usage, we propose Overlapping Segment Partitioning and Incremental Motion Tuning, respectively. Extensive experiments on two challenging datasets demonstrate that our method achieves the lowest and temporally-stablest in Chamfer distance.

Acknowledgement. ML is funded in part by an ARC Discovery Grant: DP200102274. HL holds concurrent appointments with both ANU and Amazon. This paper describes work performed at ANU and is not associated with Amazon. HL is also funded in part by an ARC Discovery Grant: DP220100800.

References

[1] Sandika Biswas, Qianyi Wu, Biplab Banerjee, and Hamid Rezatofighi. Tfs-nerf: Template-free nerf for semantic 3d reconstruction of dynamic scene. *Advances in Neural Information Processing Systems*, 37:114458–114477, 2024. 1, 2

[2] Hongrui Cai, Wanquan Feng, Xuetao Feng, Yan Wang, and Juyong Zhang. Neural surface reconstruction of dynamic scenes with monocular rgb-d camera. *Advances in Neural Information Processing Systems*, 35:967–981, 2022. 1, 2, 5, 6

[3] Weiwei Cai, Weicai Ye, Peng Ye, Tong He, and Tao Chen. Dynasurfgs: Dynamic surface reconstruction with planar-based gaussian splatting. *arXiv preprint arXiv:2408.13972*, 2024. 1, 2

[4] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996. 6

[5] Qiankun Gao, Jiarui Meng, Chengxiang Wen, Jie Chen, and Jian Zhang. Hicom: Hierarchical coherent motion for dynamic streamable scenes with 3d gaussian splatting. *Advances in Neural Information Processing Systems*, 37: 80609–80633, 2024. 2

[6] Zhongpai Gao, Benjamin Planche, Meng Zheng, Anwesa Choudhuri, Terrence Chen, and Ziyan Wu. 7dgs: Unified spatial-temporal-angular gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 26316–26325, 2025. 2

[7] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5354–5363, 2024. 3, 4

[8] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 2, 5

[9] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024. 2, 5, 6, 8

[10] Yi-Hua Huang, Yang-Tian Sun, Ziyi Yang, Xiaoyang Lyu, Yan-Pei Cao, and Xiaojuan Qi. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4220–4230, 2024. 1, 2, 5, 6

[11] Mustafa Işık, Martin Rünz, Markos Georgopoulos, Taras Khakhulin, Jonathan Starck, Lourdes Agapito, and Matthias Nießner. Humanrf: High-fidelity neural radiance fields for humans in motion. *ACM transactions on graphics (TOG)*, 42(4):1–12, 2023. 2

[12] Zeren Jiang, Chen Guo, Manuel Kaufmann, Tianjian Jiang, Julien Valentin, Otmar Hilliges, and Jie Song. Multiply: Reconstruction of multiple people from monocular video in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 109–118, 2024. 1, 2

[13] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE international conference on computer vision*, pages 3334–3342, 2015. 1, 2, 5, 6, 7

[14] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European conference on computer vision (ECCV)*, pages 371–386, 2018. 2

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

[16] Hansol Lee, Tackgeun You, Hansoo Park, Woohyeon Shim, Sanghyeon Kim, and Hwasup Lim. Contactfield: Implicit field representation for multi-person interaction geometry. *Advances in Neural Information Processing Systems*, 37: 38079–38104, 2024. 1, 2

[17] Soohyun Lee, Seoyeon Kim, HeeKyung Lee, Won-Sik Jeong, and Joo Ho Lee. Geoavatar: Geometrically-consistent multi-person avatar reconstruction from sparse multi-view videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21138–21147, 2025. 1, 2

[18] Lingzhi Li, Zhen Shen, Zhongshu Wang, Li Shen, and Ping Tan. Streaming radiance fields for 3d video synthesis. *Advances in Neural Information Processing Systems*, 35: 13485–13498, 2022. 2

[19] Xueting Li, Sifei Liu, Shalini De Mello, Kihwan Kim, Xiaolong Wang, Ming-Hsuan Yang, and Jan Kautz. Online adaptation for consistent mesh reconstruction in the wild. *Advances in Neural Information Processing Systems*, 33: 15009–15019, 2020. 2

[20] Xuesong Li, Jinguang Tong, Jie Hong, Vivien Rolland, and Lars Petersson. Dgns: Deformable gaussian splatting and dynamic neural surface for monocular dynamic 3d reconstruction. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 1812–1821, 2025. 1, 2, 4

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

[22] Haotong Lin, Sida Peng, Zhen Xu, Tao Xie, Xingyi He, Hujun Bao, and Xiaowei Zhou. High-fidelity and real-time novel view synthesis for dynamic scenes. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–9, 2023. 2

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

[24] Isabella Liu, Hao Su, and Xiaolong Wang. Dynamic gaussians mesh: Consistent mesh reconstruction from dynamic

scenes. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 2, 5

[25] Shaojie Ma, Yawei Luo, Wei Yang, and Yi Yang. Mags: Reconstructing and simulating dynamic 3d objects with mesh-adsorbed gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8745–8755, 2025. 1, 2, 4, 5

[26] Wei Mao, Richard Hartley, Mathieu Salzmann, and Miaomiao Liu. Neural SDF flow for 3d reconstruction of dynamic scenes. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 2, 3, 5, 6

[27] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[28] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. Ieee, 2011. 3, 4

[29] Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single rgb images via topology modification networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9964–9973, 2019. 2

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

[31] 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. *ACM Trans. Graph.*, 40(6), 2021. 2

[32] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019. 1, 2

[33] Songyou Peng, Chiyu Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as points: A differentiable poisson solver. *Advances in Neural Information Processing Systems*, 34:13032–13044, 2021. 2

[34] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10318–10327, 2021. 2

[35] Ruizhi Shao, Zerong Zheng, Hanzhang Tu, Boning Liu, Hongwen Zhang, and Yebin Liu. Tensor4d: Efficient neural 4d decomposition for high-fidelity dynamic reconstruction and rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16632–16642, 2023. 2, 5, 6

[36] Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. Nerfplayer: A streamable dynamic scene representation with decomposed neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 29(5):2732–2742, 2023. 2

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

[38] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12959–12970, 2021. 2

[39] Feng Wang, Zilong Chen, Guokang Wang, Yafei Song, and Huaping Liu. Masked space-time hash encoding for efficient dynamic scene reconstruction. *Advances in neural information processing systems*, 36:70497–70510, 2023. 2

[40] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Morpheus: Neural dynamic 360deg surface reconstruction from monocular rgb-d video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20965–20976, 2024. 1, 2

[41] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018. 2

[42] Shuo Wang, Binbin Huang, Ruoyu Wang, and Shenghua Gao. Space-time 2d gaussian splatting for accurate surface reconstruction under complex dynamic scenes. *arXiv preprint arXiv:2409.18852*, 2024. 1, 2, 5, 6

[43] Yifan Wang, Peishan Yang, Zhen Xu, Jiaming Sun, Zhanhua Zhang, Yong Chen, Hujun Bao, Sida Peng, and Xiaowei Zhou. Freetimegs: Free gaussian primitives at anytime anywhere for dynamic scene reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21750–21760, 2025. 1, 2, 5, 6, 7

[44] Zihan Wang, Jeff Tan, Tarasha Khurana, Neehar Peri, and Deva Ramanan. Monofusion: Sparse-view 4d reconstruction via monocular fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8252–8263, 2025. 2, 5

[45] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20310–20320, 2024. 1, 2, 5, 6

[46] Jiang Wu, Rui Li, Yu Zhu, Rong Guo, Jinqiu Sun, and Yanning Zhang. Sparse2dgs: Geometry-prioritized gaussian splatting for surface reconstruction from sparse views. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 11307–11316, 2025. 1, 5, 6, 7, 8

[47] Shangzhe Wu, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. Dove: Learning deformable 3d objects by watching videos. *International Journal of Computer Vision*, 131(10):2623–2634, 2023. 1, 2

[48] Jiawei Xu, Zexin Fan, Jian Yang, and Jin Xie. Grid4d: 4d decomposed hash encoding for high-fidelity dynamic gaussian splatting. *Advances in Neural Information Processing Systems*, 37:123787–123811, 2024. 2

[49] Yuxuan Xue, Bharat Lal Bhatnagar, Riccardo Marin, Nikolaos Sarafianos, Yuanlu Xu, Gerard Pons-Moll, and Tony Tung. Nsf: Neural surface fields for human modeling from monocular depth. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15049–15060, 2023. 2

[50] Jinbo Yan, Rui Peng, Zhiyan Wang, Luyang Tang, Jiayu Yang, Jie Liang, Jiahao Wu, and Ronggang Wang. Instant gaussian stream: Fast and generalizable streaming of dynamic scene reconstruction via gaussian splatting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16520–16531, 2025. 2

[51] Chen Yang, Sikuang Li, Jiemin Fang, Ruofan Liang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Gaussianobject: High-quality 3d object reconstruction from four views with gaussian splatting. *ACM Transactions on Graphics (TOG)*, 43(6):1–13, 2024. 5

[52] Ziyi Yang, Xinyu Gao, Wen Zhou, Shaohui Jiao, Yuqing Zhang, and Xiaogang Jin. Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20331–20341, 2024. 2, 6

[53] Zeyu Yang, Hongye Yang, Zijie Pan, and Li Zhang. Real-time photorealistic dynamic scene representation and rendering with 4d gaussian splatting. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[54] Yifei Yin, Chen Guo, Manuel Kaufmann, Juan Jose Zarate, Jie Song, and Otmar Hilliges. Hi4d: 4d instance segmentation of close human interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17016–17027, 2023. 2, 5, 6, 7, 8

[55] Shuai Zhang, Guanjun Wu, Zhoufeng Xie, Xinggang Wang, Bin Feng, and Wenyu Liu. Dynamic 2d gaussians: Geometrically accurate radiance fields for dynamic objects. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 8144–8153, 2025. 1, 2, 4, 5, 6, 7, 8

[56] Chengwei Zheng, Lixin Xue, Juan Zarate, and Jie Song. Gaustar: Gaussian surface tracking and reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16543–16553, 2025. 1, 2, 5, 6, 8