# GSTAR: Gaussian Surface Tracking and Reconstruction

Chengwei Zheng[1]  Lixin Xue[1]  Juan Zarate[1]  Jie Song[1,2,3]

[1]ETH Zürich  [2]HKUST(GZ)  [3]HKUST

Figure 1. We propose GSTAR, a novel method that (a) enables photo-realistic rendering, surface reconstruction, and 3D tracking for dynamic scenes while handling topology changes. (b) GSTAR adapts to topology changes through two mechanisms: consistent tracking for stable surfaces (red circles) and dynamic surface generation for newly appearing geometry (orange circles).

## Abstract

*3D Gaussian Splatting techniques have enabled efficient photo-realistic rendering of static scenes. Recent works have extended these approaches to support surface reconstruction and tracking. However, tracking dynamic surfaces with 3D Gaussians remains challenging due to complex topology changes, such as surfaces appearing, disappearing, or splitting. To address these challenges, we propose GSTAR, a novel method that achieves photo-realistic rendering, accurate surface reconstruction, and reliable 3D tracking for general dynamic scenes with changing topology. Given multi-view captures as input, GSTAR binds Gaussians to mesh faces to represent dynamic objects. For surfaces with consistent topology, GSTAR maintains the mesh topology and tracks the meshes using Gaussians. In regions where topology changes, GSTAR adaptively unbinds Gaussians from the mesh, enabling accurate registration and the generation of new surfaces based on these optimized Gaussians. Additionally, we introduce a surface-based scene flow method that provides robust initialization for tracking between frames. Experiments demonstrate that our method effectively tracks and reconstructs dynamic surfaces, enabling a range of applications. Our project page with the code release is available at https://chengwei-zheng.github.io/GSTAR/.*

## 1. Introduction

In the realm of dynamic scene representations, we seek methods capable of delivering photorealistic renderings

1

from arbitrary viewpoints, as well as surface reconstructions that adapt to changing topologies. Scenarios involving human or robotic interactions with objects require dynamic adaptation to surfaces that split, merge, or deform. Furthermore, downstream applications such as visual effects and markerless motion capture benefit significantly from the ability to track persistent regions over time without relying on templates. Therefore, methods must efficiently handle these topology changes to ensure high-quality renderings and accurate reconstruction while also maintaining consistent tracking of existing surfaces.

Classical methods primarily rely on meshes and texture maps, which provide reasonable appearances but heavily depend on mesh resolution. They often fail to render fine details and view-dependent effects accurately. Although these mesh representations allow for some level of tracking, they struggle to handle significant topology changes, necessitating new keyframes to accommodate major transformations. The advent of Neural Radiance Fields (NeRF) [28] brought significant improvements in appearance and novel view synthesis for both static [1, 46] and dynamic scenes [17, 30]. While surfaces can be derived from implicit Signed Distance Functions (SDF) using Marching Cubes [37, 44], they lack consistent tracking unless underlying templates are used. Recently, 3D Gaussian Splatting (3DGS) [20] has emerged with explicit texture representation, rivaling NeRF in appearance while achieving more efficient renderings. Its explicit representation facilitates tracking, and several techniques have been developed for this purpose [26, 50]. However, accurate dynamic surface reconstruction remains a challenge, and balancing the tracking of existing surfaces with the introduction of new ones proves difficult.

To address these challenges, we propose GSTAR, a method capable of reconstructing photorealistic appearances and accurate surface geometries with consistent tracking as topology changes. GSTAR leverages multi-view capture and combines meshes with bound Gaussians to create Gaussian Surfaces. These Gaussians move along with the mesh faces to represent objects that move and deform. When new surfaces become visible, new Gaussians are generated, and the mesh topology updates. The adaptable mesh provides a time-consistent and accurate geometry, while the Gaussians bring a photorealistic appearance.

This problem is difficult because there is always a trade-off. Methods that allow easier tracking via fixed topologies or templates [24, 50] tend to degrade the quality of the appearance and geometry under new poses or deformations. Conversely, methods that overfit static scenes [8, 14, 16] lack temporal consistency or miss new frame details. GSTAR addresses this trade-off by tracking as many surfaces over time as possible while remaining flexible to enable new faces and Gaussians to appear where the topology

changes. We adapt the preceding frame by deforming the mesh and optimizing Gaussian parameters. For topology-changing surfaces such as newly emerging ones, GSTAR first unbinds the Gaussians in these regions, allowing them to move beyond the mesh faces. New Gaussian Surfaces are then generated based on the unbound Gaussians, enabling accurate reconstruction of the new surfaces. Additionally, we propose a surface-based scene flow method that leverages 2D optical flow back-projected into 3D space using depth images. It provides an initialization for frame-by-frame tracking to robustly manage large 3D or fast motions.

Our contributions are as follows.

- A new framework for tracking and reconstructing dynamic scenes combining 3D Gaussians and meshes, effectively managing changes in topology.
- A method for Gaussian unbinding and surface re-meshing that allows the generation of new surfaces as topologies evolve.
- A method for handling large or fast deformation of surfaces between frames via scene flow warping.

As demonstrated in our experiments, GSTAR matches or surpasses SOTA methods in appearance metrics, thanks to the performance of Gaussian Surfaces. This makes it a strong representation for high-quality 3D rendering applications such as VR/XR and telepresence. Simultaneously, GSTAR provides a high-resolution, explicit 3D representation with robust tracking capabilities, as illustrated by our AprilTag-based experiments. We expect these tracked meshes will facilitate numerous tasks beyond rendering, benefiting fields such as computer vision, computer graphics, robotics, biomechanics, spatial audio, and more.

## 2. Related Work

### 2.1. 3D Neural Representations

The reconstruction of general dynamic scenes has been a long-standing problem in computer vision. Traditional methods relied on triangle meshes with texture maps [7, 34]. While these explicit representations enabled efficient rendering and intuitive geometry editing, they struggled with view-dependent effects and fine surface details. Recent neural representations such as NeRF [28] and 3DGS [20] have significantly advanced the static reconstruction field with coordinate-based networks and explicit 3D Gaussians. Both NeRF and 3DGS can achieve photo-realistic rendering, while the surfaces cannot be extracted accurately. NeRF-based methods address this by introducing SDF fields [37, 41, 44] and using Marching Cubes [25] to generate surfaces. Gaussian-based methods propose to use surface-alignment regularization terms [5, 8, 11, 14] during training, followed by mesh extraction. As 3D Gaussians are discrete presentations rather than continuous representations like NeRF, Poisson reconstruction [8, 14] and SDF fusion [4, 16, 42]

are also employed for mesh extraction, along with Marching Cubes [22, 23, 45]. To further improve the quality, Gaussian Surfels [8] propose a novel point-based representation and a self-supervised normal-depth consistency regularizer. 2D Gaussian Splatting [16] uses 2D Gaussians that are tightly aligned to surfaces. NeuSG [3] jointly optimizes implicit surface reconstruction with 3D Gaussian Splatting. While these methods perform well for static scenes, applying them to achieve consistent dynamic reconstruction is far from straightforward.

## 2.2. 4D Dynamic Representations

To handle dynamic scenes, several NeRF-based methods [12, 31, 41, 48] incorporate time-dependent variables into the reconstruction model to represent movements and deformations. For instance, HumanRF [17] reconstructs radiance fields using 4D feature grids over temporal segments. While these approaches achieve photo-realistic rendering of general dynamic scenes, they lack consistent tracking capabilities, limiting their practical applications.

For 3DGS-based methods, deformation modules [2, 24, 47] are also introduced to deform Gaussians and reconstruct meshes from monocular inputs. MaGS [27] constrains 3D Gaussians to hover on the mesh surface, creating a mutual-adsorbed mesh-Gaussian representation. Spacetime 2D Gaussian Splatting [40] leverages 2D Gaussians to reconstruct dynamic scenes and extract surfaces from them. However, these methods extract meshes independently for each frame, limiting their ability to generate face correspondences across frames.

## 2.3. Tracking

Tracking methods aim to estimate the motion trajectories of surface points. 2D tracking methods [9, 15, 19] take video inputs and track pixels across frames. OmniMotion [38] improves pixel-wise tracking by introducing a 3D canonical volume and a set of bijections. Shape of Motion [39] represents scene motion using a set of motion bases, providing a globally consistent representation of dynamic scenes. With the use of 3D Gaussians as an explicit representation, new possibilities emerge for more efficient 3D tracking. Dynamic 3D Gaussians [26] tracks Gaussians by directly optimizing their positions with multi-view inputs. PhysAvatar [50] tracks time-consistent meshes and models human clothes via physics-based simulation and rendering. To improve performance, traditional non-rigid fusion techniques [18], optical flow methods [6, 13, 51], and multi-head deformation decoders [43] are also employed. Additionally, some Gaussian-based reconstruction methods [10, 21] deform sets of Gaussians to represent dynamic scenes; however, their use of temporary Gaussians restricts them to short-range tracking. While these methods demonstrate the ability to track Gaussians or meshes with fixed topology, tracking surfaces as topologies evolve remains an open challenge.

## 3. Method

Our system takes multi-view RGB-D videos as input. We aim to achieve consistent reconstruction and tracking even when surfaces undergo topology changes. To represent dynamic objects, we introduce Gaussian Surfaces - meshes with Gaussians attached to their faces - which enable both accurate geometry reconstruction and photo-realistic rendering (Sec. 3.1 and Fig. 2 (c)). For each frame, we first initialize surface positions through scene flow warping from the previous frame (Sec. 3.2 and Fig. 2 (b)). We then optimize the Gaussian Surfaces with the topology from the previous frame using multi-view constraints (Sec. 3.3 and Fig. 2 (c)). For regions experiencing topology changes, which are detected through our novel Gaussian unbinding weights, we allow Gaussians to detach from the original mesh faces and optimize their positions independently (Sec. 3.4 and Fig. 2 (d)). Lastly, we perform re-meshing to update the topology-changing geometry, and ensure our representation remains consistent in other regions (Sec. 3.5 and Fig. 2 (e)).

## 3.1. Gaussian Surface Representation

Our Gaussian Surface representation, shown in (c) of Fig. 2, augments traditional meshes with $N$ Gaussians per triangular face [14, 50]. Following the formulation from 3DGS [20],

$$G(\mathbf{x}) = \sigma(\alpha) \cdot \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{p})^\top \Sigma^{-1}(\mathbf{x} - \mathbf{p})\right), \quad (1)$$

$$\Sigma = \mathbf{R}\mathbf{S}\mathbf{S}^\top\mathbf{R}^\top, \quad (2)$$

where each Gaussian is defined by its opacity $\alpha$, center position $\mathbf{p}$, scales $\mathbf{S}$, rotation $\mathbf{R}$, and appearance color (represented by spherical harmonics). Here, $\sigma()$ is the standard sigmoid function and $\Sigma$ represents the covariance matrix.

To construct Gaussian Surfaces, we uniformly distribute $N$ Gaussians on each triangular face. Each Gaussian center $\mathbf{p}$ is computed from the face vertices $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ using its pre-defined barycentric coordinate $(b_1, b_2, b_3)$.

$$\mathbf{p} = b_1\mathbf{v}_1 + b_2\mathbf{v}_2 + b_3\mathbf{v}_3. \quad (3)$$

To ensure Gaussians remain aligned with the mesh surface, we constrain their orientation and thickness: the z-axis $\mathbf{R}_z$ is set to the face normal $\mathbf{n}(f)$, while the z-scale $\mathbf{S}_z$ is set to a small value $\delta$. The remaining Gaussian parameters are jointly optimized following the volumetric rendering approach of 3DGS [20].
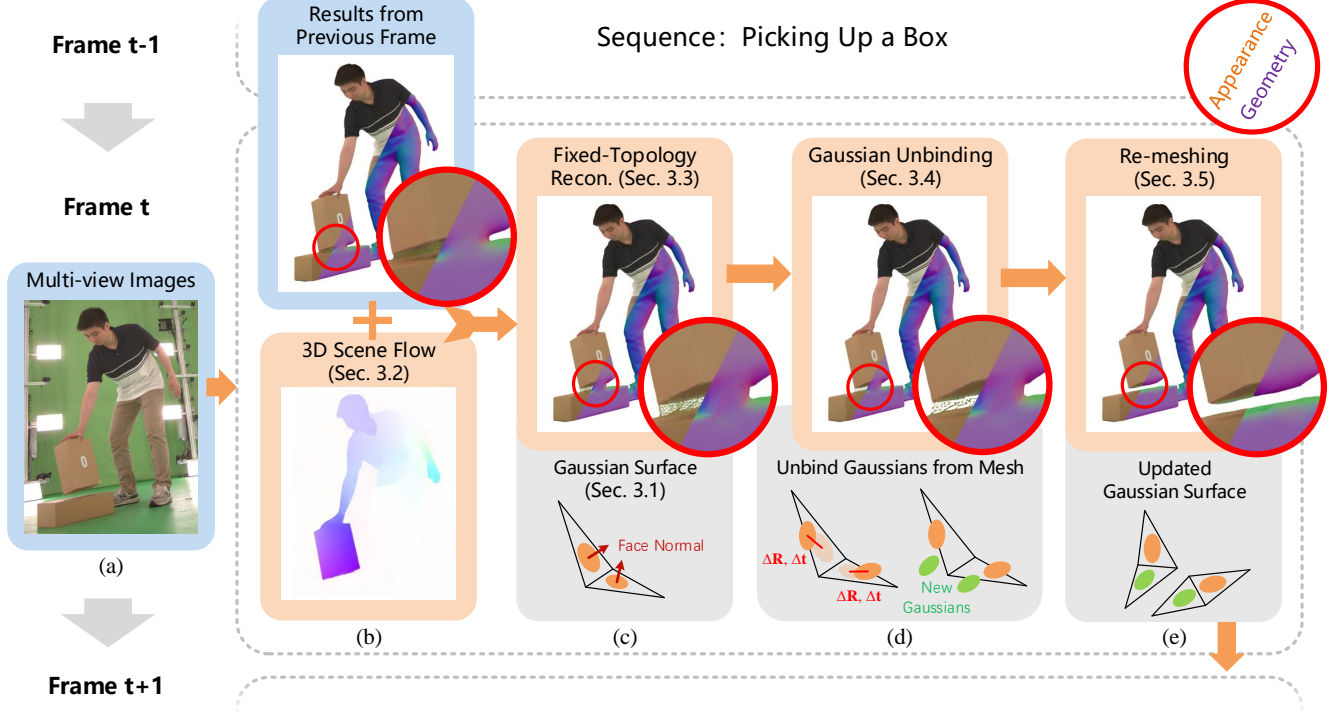
Figure 2. Taking multi-view captures as input, GSTAR tracks and reconstructs dynamic objects frame by frame. For each frame, GSTAR first warps the previous frame's result using scene flow (Sec. 3.2). It then reconstructs Gaussian Surfaces (Gaussian-attached mesh, Sec. 3.1) by fixed-topology reconstruction (Sec. 3.3). To handle topology-changing surfaces, GSTAR detects topology changes, unbinds Gaussians on these surfaces, and adds new Gaussians as needed (Sec. 3.4). Finally, the Gaussian Surfaces are updated through re-meshing (Sec. 3.5).

## 3.2. Scene Flow Warping

Dynamic scenes often exhibit large or rapid deformations between frames, making tracking and optimization challenging. To address this, we estimate 3D scene flow by re-projecting 2D optical flow using depth information, providing robust initialization for each frame's reconstruction.

Given adjacent frames $t$ and $t + 1$, we compute scene flow in four steps. First, we project each vertex from frame $t$ into all visible input views. Second, we compute the corresponding positions in frame $t + 1$ using optical flow [36]. Third, we re-project these 2D positions in $t + 1$ back to 3D using the respective depth images. Finally, we aggregate the 3D movements between frames across all views to obtain the scene flow for each vertex.

To improve robustness, we filter out unreliable flows using both bi-directional optical flow consistency checks between frames $t$ and $t + 1$ and depth discontinuity detection. We further refine the scene flow through surface-aware smoothing:

$$\mathcal{F}'(v) = \frac{1}{|\mathbf{N}(v)|} \sum_{u \in \mathbf{N}(v)} w(u,v)\mathcal{F}(u), \qquad (4)$$

where $\mathcal{F}(v)$ and $\mathcal{F}'(v)$ are the scene flow before and after

smoothing, $\mathbf{N}(v)$ represents mesh-connected neighbors of vertex $v$, and $w(u,v)$ weights neighbors by their distance. The final vertex positions are updated as $v + \mathcal{F}'(v)$.

## 3.3. Fixed-Topology Surface Reconstruction

We first reconstruct Gaussian Surfaces assuming fixed topology, establishing a baseline reconstruction that will later be refined to handle topology changes. Given multi-view RGB-D inputs and a mesh initialized through scene flow warping, we optimize vertex positions and Gaussian parameters using RGB, depth, and mask supervision:

$$\mathcal{L}_{\text{rgb}} = \left\| \hat{C}_i - C_i \right\|_1 + \lambda_{\text{SSIM}}\mathcal{L}_{\text{SSIM}}(\hat{C}_i, C_i), \qquad (5)$$

$$\mathcal{L}_{\text{depth}} = \left\| \hat{D}_i - D_i \right\|_1, \ \ \mathcal{L}_{\text{mask}} = \left\| \hat{M}_i - M_i \right\|_1. \qquad (6)$$

Here $\hat{C}_i, \hat{D}_i, \hat{M}_i$ represent predicted color, depth, and mask images from Gaussian rendering, while $C_i, D_i, M_i$ are the input images. As Gaussian positions are determined by vertex positions in Eq. (3), optimizing these losses effectively updates the underlying mesh.

We further introduce three regularization terms: normal smoothing to ensure surface continuity, area preservation to

4

maintain local geometry, and color consistency to promote temporal coherence:

$$\mathcal{L}_{\text{smooth}} = \frac{1}{|\mathbf{F}|} \sum_{f_i \in \mathbf{F}} \sum_{f_j \in \mathbf{N}(f_i)} (1 - \mathbf{n}(f_i) \cdot \mathbf{n}(f_j)), \quad (7)$$

$$\mathcal{L}_{\text{area}} = \frac{1}{|\mathbf{F}|} \sum_{f \in \mathbf{F}} \|\mathbf{Area}(f;t) - \mathbf{Area}(f;0)\|_1, \quad (8)$$

$$\mathcal{L}_{\text{SH}} = \frac{1}{|\mathbf{G}|} \sum_{g \in \mathbf{G}} \|\mathbf{SH}(g;t) - \mathbf{SH}(g;t-1)\|_2^2, \quad (9)$$

where $\mathbf{F}$ is the set of triangular faces, $\mathbf{N}(f_i)$ denotes neighboring faces of $f_i$, and $\mathbf{n}(f)$ computes the face normal. $\mathbf{Area}(f;t)$ measures face area at frame $t$, penalizing deviation from initial areas $\mathbf{Area}(f;0)$. $\mathbf{SH}(g;t)$ represents the spherical harmonics parameters of Gaussian $g$ at frame $t$, where $\mathbf{G}$ denotes the set of Gaussians. Additionally, we constrain each Gaussian's scale by its face's edge length and enforce a minimum opacity to ensure surface opacity.

## 3.4. Adaptive Gaussian Unbinding

While fixed-topology surface reconstruction refines meshes based on multi-view input, it cannot handle emerging surfaces or topology changes. We address this problem by allowing Gaussians to detach from the mesh faces in regions where topology changes are detected.

As illustrated in Fig. 2 (d), we introduce additional transformation parameters for each Gaussian: a rotation $\Delta\mathbf{R}$ applied to the Gaussian orientation $\mathbf{R}$, and a translation $\Delta\mathbf{t}$ added to its center position $\mathbf{p}$. To optimize these transformations, we extend the fixed-topology reconstruction process to jointly optimize both the original parameters and these additional transformations. This allows Gaussians to move independently from their underlying mesh faces when necessary.

To identify which Gaussians should be unbound from the mesh, we develop a weighting scheme based on geometric and photometric cues. Inspired by the adaptive density control in 3DGS [20], we observe that topology changes typically manifest as large positional gradients and high reconstruction errors, as shown in the top part of Fig. 2 (c). We define an *unbinding weight* for each face $f$:

$$\mathcal{W}(f) = \mathcal{G}_{\text{pos}}(f) + \lambda_{\text{rgb}}\mathcal{L}_{\text{rgb}}(f) + \lambda_{\text{depth}}\mathcal{L}_{\text{depth}}(f). \quad (10)$$

Here unbinding weight $\mathcal{W}$ measures the likelihood of topology changes for each face $f$. It combines positional gradients $\mathcal{G}_{\text{pos}}$ [20] and reconstruction errors from fixed-topology optimization. We cap $\mathcal{W}$ at 1 and visualize an example in Fig. 3 (a).

Based on these unbinding weights, we introduce a regularization term to control the extent of transformations for
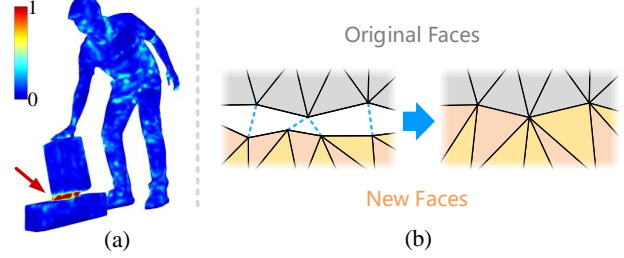


Figure 3. Details of the mesh update process. (a) Visualization of unbinding weights defined in Eq. (10), where red indicates high weights in topology-changing regions. (b) Mesh connection process between original and new surfaces, with blue dotted lines showing vertex correspondences.

Gaussians:

$$\mathcal{L}_{\text{unb}}(g) = (1-\mathcal{W}(f_g))\left(\|\Delta\mathbf{R}(g)-\mathbf{I}\|_1 + \lambda_{\text{t}}\|\Delta\mathbf{t}(g)\|_1\right). \quad (11)$$

This loss regulates the transformation of each Gaussian $g$ on face $f_g$. When unbinding weight $\mathcal{W}$ is high, indicating likely topology changes, the loss term weight becomes small, allowing larger transformations and effectively unbinding the Gaussian from its face.

In regions experiencing significant topology changes, we not only allow existing Gaussians to detach but also introduce new ones. As shown in Fig. 2 (d), for faces where unbinding weights exceed a threshold, we duplicate their associated Gaussians. These newly introduced Gaussians are also unbound and governed by the unbinding regularization loss in Eq. (11), allowing them to move independently from the original mesh.

## 3.5. Surface Re-meshing

After Gaussian unbinding, we face two key technical challenges: identifying which mesh regions to update and ensuring smooth transitions at region boundaries. We address these by first localizing topology changes through unbinding weights, then selectively reconstructing and integrating new surfaces while preserving the original mesh structure elsewhere.

To generate new surfaces from unbound Gaussians, we first render depth images from multiple viewpoints, including both capture views and uniformly sampled views on a surrounding sphere. We then employ a TSDF volume fusion method [29] to reconstruct new meshes from these depth images.

After obtaining new meshes, we identify regions on the original mesh for replacement where unbinding weights exceed a threshold. Since the unbound Gaussians from these regions have moved to new surface locations during optimization, we construct a voxel volume to locate these Gaussian positions. And within this volume, we identify corre-
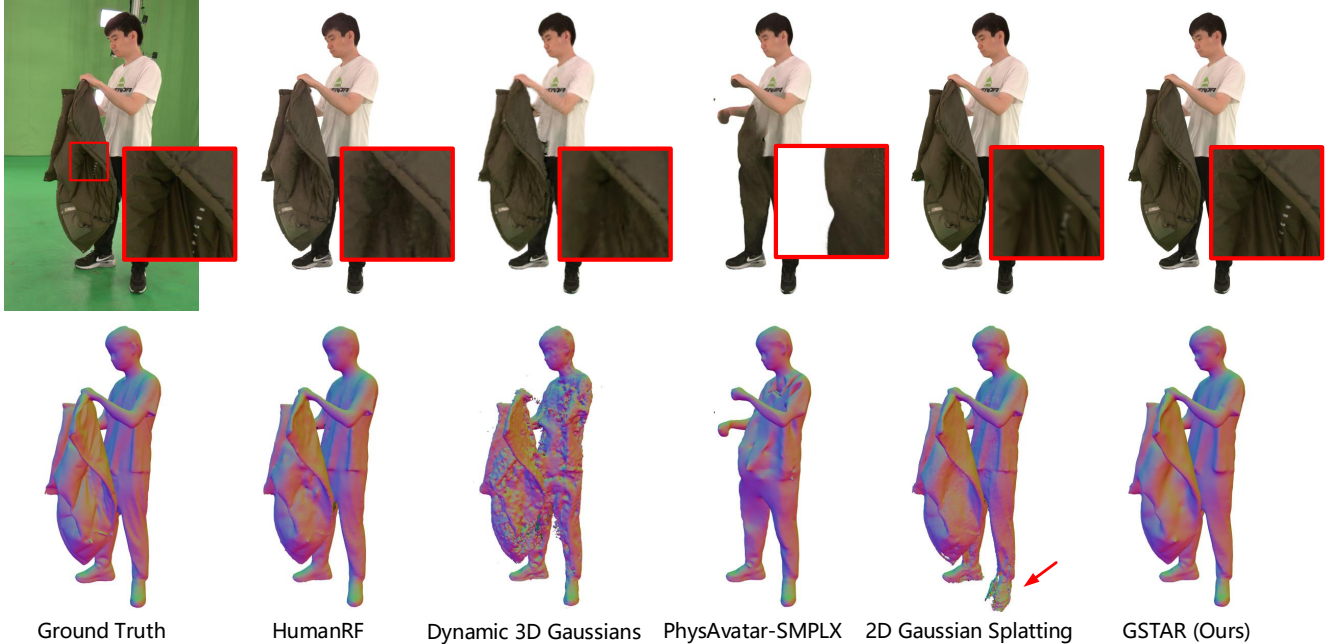
Figure 4. Comparisons of appearance and geometry reconstruction. Dynamic 3D Gaussians [26] and PhysAvatar [50] yield suboptimal reconstruction results. HumanRF [17] and 2DGS [16], lacking tracking capabilities, struggle under heavy occlusion. In contrast, GSTAR provides high-quality reconstruction while supporting tracking. Additional comparisons are provided in our supplementary materials.

sponding faces from the new meshes to replace the original faces, connecting them with the remaining original mesh.

To connect old and new meshes, we establish vertex correspondences along their boundaries, as shown in Fig. 3 (b). Each boundary vertex from one mesh is matched to its closest vertices on the other mesh boundary, allowing for one-to-many or many-to-one correspondences. We merge matched vertices, and then perform post-processing steps, including edge flipping and hole filling [33], to refine the boundary region.

At the boundary of topology-changing surfaces, the unbinding weights gradually change from 1 (fully unbound) to 0 (fully bound), as illustrated in Fig. 3 (a). Accordingly, the constraints on Gaussian transformations $\Delta\mathbf{R}, \Delta\mathbf{t}$ in Eq. (11) vary smoothly, ensuring continuity between original and new surfaces at their connection. After re-meshing, we perform an additional round of fixed-topology reconstruction described in Sec. 3.3 to fine-tune the updated Gaussian Surfaces.

## 4. Experiments

### 4.1. Implement Details

We use a capture studio with 52 RGB cameras and 52 IR cameras for capturing. Sequences are captured at a resolution of $3004 \times 4092$ and 30 fps. Unstructured IR laser lights are used, allowing us to generate depth images from the IR captures. Specifically, we generate raw point clouds

from the IR images, refine them using the method in [7], and project them onto the camera views. We attach $N = 6$ Gaussians per face. The initial mesh for the first frame can be obtained using any multi-view reconstruction method and we use [7]. Additional implementation details are provided in our supplementary materials.

### 4.2. Comparisons

We compare our method with SOTA multi-view reconstruction methods HumanRF [17], Dynamic 3D Gaussianss [26], PhysAvatar [50], and 2D Gaussian Splatting (2DGS) [16], as shown in Tab. 1 and Fig. 4. 1) HumanRF is a NeRF-based method for dynamic scene reconstruction. Its implicit representations do not provide correspondence or tracking information between frames. Since its mesh extraction code is not publicly available, we implemented it based on the paper. 2) Dynamic 3D Gaussians extend 3DGS [20] to handle dynamic scenes, enabling Gaussian tracking but without producing a surface mesh. Extracting meshes from its Gaussians results in noisy outputs. 3) PhysAvatar tracks an initial mesh using Gaussians in its first stage, with subsequent stages focusing on clothing reconstruction through simulation. We compare two versions of its first stage: one for general objects and another tailored for human subjects using SMPL-X [32]. PhysAvatar is limited in handling topology changes and struggles with large deformations. 4) 2DGS uses flat Gaussians for surface reconstruction, which we apply independently to each frame in our experiments.

| Method | Appearance | | | Geometry | | Tracking | |
|---|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | CD ↓ | F-Score ↑ | 3D ATE ↓ | 2D ATE ↓ |
| HumanRF [17] | 30.59 | 0.947 | 0.128 | 0.284 | 0.968 | - | - |
| Dynamic 3D Gaussians [26] | 27.61 | 0.905 | 0.214 | 1.113 † | 0.733 † | 3.15 | 13.84 |
| PhysAvatar-general [50] | 22.69 | 0.893 | 0.216 | 1.372 | 0.793 | 12.94 | 56.95 |
| PhysAvatar-SMPLX [50] | 24.50 | 0.908 | 0.193 | 0.625 | 0.837 | 8.98 | 39.61 |
| 2D Gaussian Splatting [16] | 30.17 | 0.938 | 0.155 | 0.699 | 0.946 | - | - |
| GSTAR w/o IR input | 30.05 | 0.946 | 0.110 | 0.335 | 0.960 | 0.671 | 3.02 |
| GSTAR (Ours) | **31.87** | **0.952** | **0.102** | **0.237** | **0.980** | **0.452** | **2.03** |

Table 1. Quantitative comparisons on appearance, geometry, and tracking. The best , second-best , and third-best results are highlighted. Our method achieves the best performance on reconstruction and tracking. CD and 3D ATE are reported in cm. †: Dynamic 3D Gaussians [26] doesn't provide surface reconstruction and we extract per-frame meshes using TSDF fusion [16].

| Method | Appearance | | | Geometry | | Tracking | |
|---|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | CD ↓ | F-Score ↑ | 3D ATE ↓ | 2D ATE ↓ |
| GSTAR w/o unbinding | 29.30 | 0.940 | 0.132 | 0.411 | 0.938 | 2.85 | 12.78 |
| GSTAR w/o re-meshing | 29.77 | 0.943 | 0.129 | 0.418 | 0.936 | 2.08 | 9.16 |
| GSTAR w/o scene flow | 29.92 | 0.943 | 0.127 | 0.433 | 0.916 | 6.56 | 29.96 |
| GSTAR (Ours final) | **31.87** | **0.952** | **0.102** | **0.237** | **0.980** | **0.45** | **2.03** |

Table 2. Ablations. Quantitative evaluation of each key component: Gaussian unbinding, re-meshing, and scene flow warping. Results demonstrate that each component is integral to the final reconstruction and tracking quality.

Without incorporating temporal information, this method converges to artifacts in several frames.

To ensure a fair comparison, we implemented a new version of our method without IR depth inputs, using rendered depth images from HumanRF [17] as the depth inputs. Additional experiment details and results are provided in our supplementary materials.

**Appearance.** We evaluate the quality of appearance reconstruction through novel view synthesis. The experiment includes 4 sequences (totaling 850 frames) and 5 test views. We report PSNR, SSIM, and LPIPS metrics in Tab. 1 and present qualitative comparisons in Fig. 4. Our method achieves superior appearance quality, particularly on surfaces with significant occlusion, benefiting from the ability to track prior surface appearance.

**Geometry.** For geometry comparison, we use an RGB-D multi-view reconstruction method [7] to obtain the ground truth meshes. Chamfer Distance (CD, in cm) and F-Score [35] are reported in Tab. 1, with qualitative comparisons shown in Fig. 4. Since Dynamic 3D Gaussians [26] does not provide surface reconstruction, we extract per-frame meshes using TSDF fusion [16], similar to our approach for new face generation. Our method yields high-quality geometry while maintaining surface tracking.

**Tracking.** We capture two sequences with AprilTags attached to the human body to evaluate tracking performance. Six AprilTags are placed on the front, back, arms, and legs,
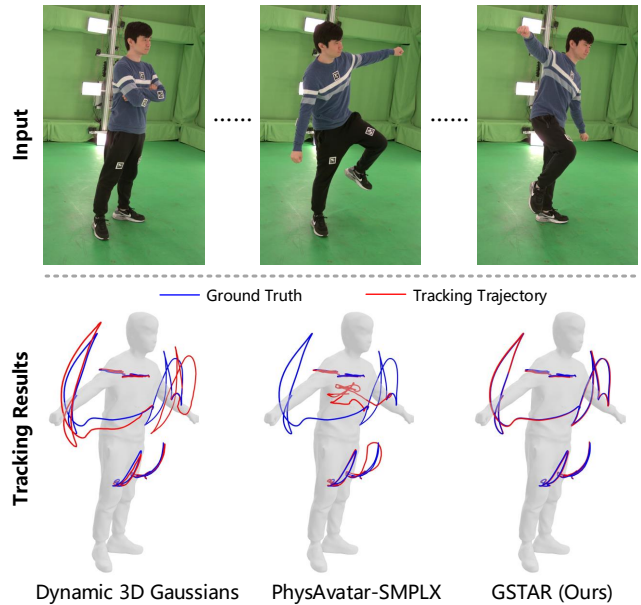


Figure 5. Tracking comparisons using AprilTags. GSTAR achieves more accurate tracking results, with predicted (red) and ground truth (blue) trajectories of tag centers shown.

as illustrated in Fig. 5. For each tag, we track 5 key points: the center and 4 corner points. The 3D and 2D ground truth are provided by AprilTag detection. We report 3D
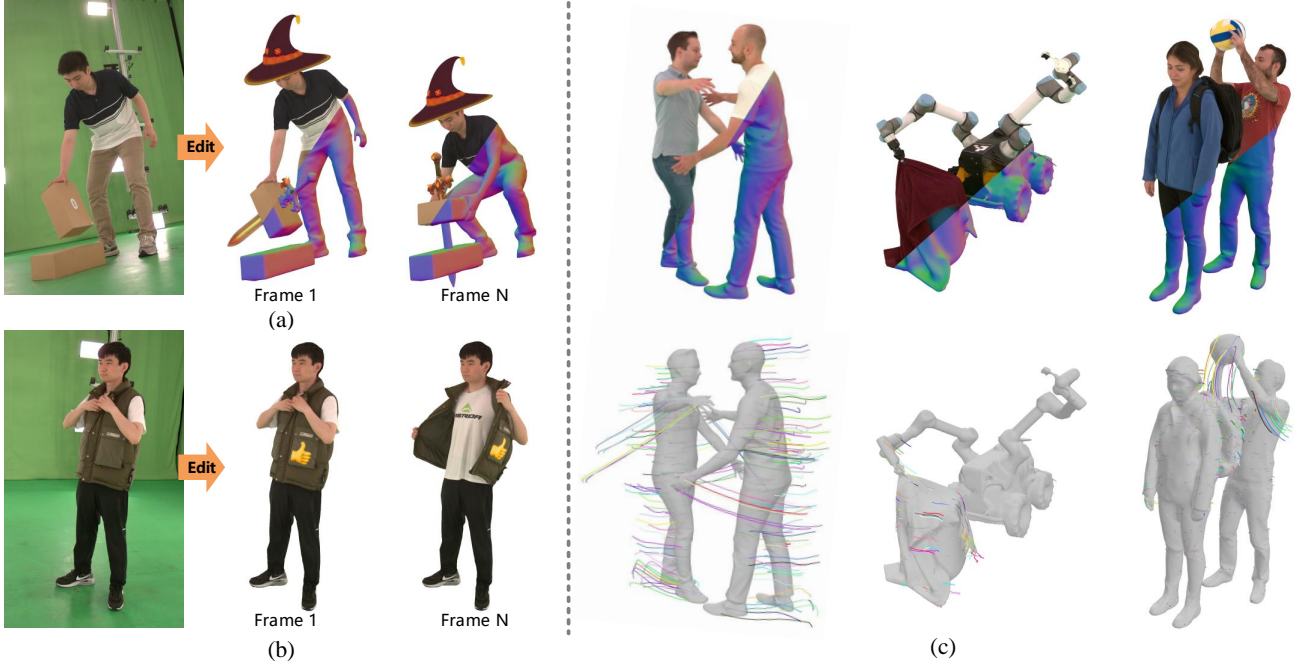
Figure 6. Applications. (a) Object editing: virtual objects sync with dynamic surfaces. (b) Appearance editing: texture changes propagate across frames. (c) General applicability to diverse scenes including multiple people, human-object interactions, and robotic motion.

and 2D Average Trajectory Error (ATE) [15, 49] in Tab. 1, with visualizations of the tracking trajectories presented in Fig. 5. Our method achieves accurate tracking results, even on topology-changing surfaces (e.g., arms uncrossing).

## 4.3. Ablations

We evaluate the key contributions of our method, including Gaussian unbinding, surface re-meshing, and scene flow warping. For ablation, each component is disabled individually, with the results shown in Tab. 2 and in our supplementary materials.

**Gaussian unbinding**. Since newly emerging surfaces cannot be modeled by fixed-topology meshes, our method unbinds Gaussians on topology-changing surfaces to ensure accurate reconstruction. Without unbinding, Gaussians remain attached to outdated faces, preventing updates to the surface topology and failing to address topology changes.

**Surface re-meshing**. Following Gaussian unbinding, our method updates the underlying mesh to reconstruct new surfaces with the correct topology. Without this update, the geometry topology remains consistent with the initial input mesh, leading to suboptimal quality.

**Scene flow warping**. For each frame initialization, we construct scene flow to warp the Gaussian Surfaces from the previous frame, effectively handling large deformations. This approach significantly enhances tracking quality and prevents the optimization process from getting trapped in local minima.

## 5. Discussion

**Applications.** Our method enables continuous tracking throughout entire sequences. We present results for object editing and appearance editing in (a) and (b) of Fig. 6 and in the supplementary video. For object editing, new objects can be inserted into dynamic scenes and move in sync with other surfaces through surface tracking. For appearance editing, modifications made in a single frame are propagated across frames via surface tracking. Our approach does not require a template and is applicable to various general dynamic scenes, including robots, multiple people, and interacting objects, as shown in Fig. 6 (c).

**Limitations.** GSTAR may face challenges with complex or sudden topology changes, such as when a new person suddenly enters the scene. Its dependence on multi-view video data restricts its applicability in general public scenarios. Like other Gaussian-based methods, it requires substantial resources for Gaussian optimization and mesh optimization, though it benefits from real-time rendering.

**Conclusions.** We propose GSTAR, a unified method for high-quality appearance reconstruction, surface reconstruction, and 3D tracking. Our approach represents dynamic surfaces by binding Gaussians to mesh faces. For surfaces with changing topology, new surfaces are reconstructed by unbinding Gaussians. GSTAR effectively handles a wide range of dynamic scenes, paving the way for new applications of Gaussian-based representations.

# References

[1] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 2

[2] 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. 3

[3] Hanlin Chen, Chen Li, and Gim Hee Lee. Neusg: Neural implicit surface reconstruction with 3d gaussian splatting guidance. *arXiv preprint arXiv:2312.00846*, 2023. 3

[4] Hanlin Chen, Fangyin Wei, Chen Li, Tianxin Huang, Yunsong Wang, and Gim Hee Lee. Vcr-gaus: View consistent depth-normal regularizer for gaussian surface reconstruction. *arXiv preprint arXiv:2406.05774*, 2024. 2

[5] Jaehoon Choi, Yonghan Lee, Hyungtae Lee, Heesung Kwon, and Dinesh Manocha. Meshgs: Adaptive mesh-aligned gaussian splatting for high-quality rendering. In *Proceedings of the Asian Conference on Computer Vision*, pages 3310–3326, 2024. 2

[6] Wen-Hsuan Chu, Lei Ke, and Katerina Fragkiadaki. Dreamscene4d: Dynamic multi-object scene generation from monocular videos. *arXiv preprint arXiv:2405.02280*, 2024. 3

[7] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (ToG)*, 34(4):1–13, 2015. 2, 6, 7

[8] Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. High-quality surface reconstruction using gaussian surfels. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2, 3

[9] Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10061–10072, 2023. 3

[10] Yuanxing Duan, Fangyin Wei, Qiyu Dai, Yuhang He, Wenzheng Chen, and Baoquan Chen. 4d gaussian splatting: Towards efficient novel view synthesis for dynamic scenes. *arXiv preprint arXiv:2402.03307*, 2024. 3

[11] Lue Fan, Yuxue Yang, Minxing Li, Hongsheng Li, and Zhaoxiang Zhang. Trim 3d gaussian splatting for accurate geometry representation. *arXiv preprint arXiv:2406.07499*, 2024. 2

[12] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-planes: Explicit radiance fields in space, time, and appearance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12479–12488, 2023. 3

[13] Quankai Gao, Qiangeng Xu, Zhe Cao, Ben Mildenhall, Wenchao Ma, Le Chen, Danhang Tang, and Ulrich Neumann. Gaussianflow: Splatting gaussian dynamics for 4d content creation. *arXiv preprint arXiv:2403.12365*, 2024. 3

[14] 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. 2, 3

[15] Adam W Harley, Zhaoyuan Fang, and Katerina Fragkiadaki. Particle video revisited: Tracking through occlusions using point trajectories. In *European Conference on Computer Vision*, pages 59–75. Springer, 2022. 3, 8

[16] 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, 3, 6, 7

[17] 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, 3, 6, 7

[18] Yuheng Jiang, Zhehao Shen, Penghao Wang, Zhuo Su, Yu Hong, Yingliang Zhang, Jingyi Yu, and Lan Xu. Hifi4g: High-fidelity human performance rendering via compact gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19734–19745, 2024. 3

[19] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. *arXiv preprint arXiv:2307.07635*, 2023. 3

[20] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4):1–14, 2023. 2, 3, 5, 6

[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. 3

[22] Zhuoxiao Li, Shanliang Yao, Yijie Chu, Angel F Garcia-Fernandez, Yong Yue, Eng Gee Lim, and Xiaohui Zhu. Mvg-splatting: Multi-view guided gaussian splatting with adaptive quantile-based geometric consistency densification. *arXiv preprint arXiv:2407.11840*, 2024. 3

[23] Ancheng Lin and Jun Li. Direct learning of mesh and appearance via 3d gaussian splatting. *arXiv preprint arXiv:2405.06945*, 2024. 3

[24] Isabella Liu, Hao Su, and Xiaolong Wang. Dynamic gaussians mesh: Consistent mesh reconstruction from monocular videos. *arXiv preprint arXiv:2404.12379*, 2024. 2, 3

[25] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987. 2

[26] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, pages 800–809. IEEE, 2024. 2, 3, 6, 7

[27] Shaojie Ma, Yawei Luo, and Yi Yang. Reconstructing and simulating dynamic 3d objects with mesh-adsorbed gaussian splatting. *arXiv preprint arXiv:2406.01593*, 2024. 3

[28] B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, 2020. 2

[29] 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. 5

[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 Transactions on Graphics (TOG)*, 40 (6):1–12, 2021. 3

[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. 6

[33] Nicholas Sharp and Keenan Crane. You can find geodesic paths in triangle meshes by just flipping edges. *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020. 6

[34] Jonathan Starck and Adrian Hilton. Surface capture for performance-based animation. *IEEE computer graphics and applications*, 27(3):21–31, 2007. 2

[35] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3405–3414, 2019. 7

[36] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 4

[37] 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. *arXiv preprint arXiv:2106.10689*, 2021. 2

[38] Qianqian Wang, Yen-Yu Chang, Ruojin Cai, Zhengqi Li, Bharath Hariharan, Aleksander Holynski, and Noah Snavely. Tracking everything everywhere all at once. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19795–19806, 2023. 3

[39] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of motion: 4d reconstruction from a single video. *arXiv preprint arXiv:2407.13764*, 2024. 3

[40] 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. 3

[41] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3295–3306, 2023. 2, 3

[42] Yaniv Wolf, Amit Bracha, and Ron Kimmel. Gs2mesh: Surface reconstruction from gaussian splatting via novel stereo views. In *ECCV 2024 Workshop on Wild 3D: 3D Modeling, Reconstruction, and Generation in the Wild*, 2024. 2

[43] 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*, pages 20310–20320, 2024. 3

[44] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. *Advances in neural information processing systems*, 35:25018–25032, 2022. 2

[45] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient adaptive surface reconstruction in unbounded scenes. *ACM Transactions on Graphics*, 2024. 3

[46] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[47] Shuai Zhang, Guanjun Wu, Xinggang Wang, Bin Feng, and Wenyu Liu. Dynamic 2d gaussians: Geometrically accurate radiance fields for dynamic objects. *arXiv preprint arXiv:2409.14072*, 2024. 3

[48] Chengwei Zheng, Wenbin Lin, and Feng Xu. Editablenerf: Editing topologically varying neural radiance fields by key points. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8317–8327, 2023. 3

[49] Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19855–19865, 2023. 8

[50] Yang Zheng, Qingqing Zhao, Guandao Yang, Wang Yifan, Donglai Xiang, Florian Dubost, Dmitry Lagun, Thabo Beeler, Federico Tombari, Leonidas Guibas, et al. Physavatar: Learning the physics of dressed 3d avatars from visual observations. *arXiv preprint arXiv:2404.04421*, 2024. 2, 3, 6, 7

[51] Ruijie Zhu, Yanzhe Liang, Hanzhi Chang, Jiacheng Deng, Jiahao Lu, Wenfei Yang, Tianzhu Zhang, and Yongdong Zhang. Motiongs: Exploring explicit motion guidance for deformable 3d gaussian splatting. *arXiv preprint arXiv:2410.07707*, 2024. 3