SAGA: Surface-Aligned Gaussian Avatar

Ronghan Chen, Yang Cong, *Senior Member, IEEE*, Jiayue Liu

Abstract—This paper presents a Surface-Aligned Gaussian representation for creating animatable human avatars from monocular videos, aiming at improving the novel view and pose synthesis performance while ensuring fast training and real-time rendering. Recently, 3D Gaussian Splatting (3DGS) has emerged as a more efficient and expressive alternative to neural radiance fields (NeRF), and has been used for creating dynamic human avatars. However, when applied to the severely ill-posed task of monocular dynamic reconstruction, the Gaussians tend to overfit the constantly changing regions such as clothes wrinkles or shadows since these regions cannot provide consistent supervision, resulting in *noisy geometry* and *abrupt deformation* that typically fail to generalize under novel views and poses. To address these limitations, we present SAGA, *i.e.*, Surface-Aligned Gaussian Avatar, which aligns the Gaussians with a mesh to enforce *well-defined geometry* and *consistent deformation*, thereby improving generalization under novel views and poses. Unlike existing strict alignment methods that suffer from limited expressive power and low realism, SAGA employs a two-stage alignment strategy where the Gaussians are first *adhered on* while then *detached from* the mesh, thus facilitating both good geometry and high expressivity. In the first *Adhered Stage*, we improve the flexibility of Adhered-on-Mesh Gaussians by allowing them to flow on the mesh, in contrast to existing methods that rigidly bind Gaussians to fixed location. In the second *Detached Stage*, we introduce a Gaussian-Mesh Alignment regularization, which allows us to unleash the expressivity by detaching the Gaussians but maintain the geometric alignment by minimizing their location and orientation offsets from the bound triangles. Finally, since the Gaussians may drift outside the bound triangles during optimization, an efficient Walking-on-Mesh strategy is proposed to dynamically update the bound triangles, ensuring accurate regularization even as the geometry evolves. Experiments on challenging datasets demonstrate that SAGA outperforms both NeRF and Gaussian-based methods on novel view and pose synthesis tasks, with fast training time of **12** minutes, and real-time rendering efficiency at **60+** FPS. Additionally, we showcase that SAGA enables direct high-quality mesh extraction from Gaussians, marking the first attempt at deformable Gaussians learned from monocular human videos.

Index Terms—Neural Rendering, 3D Gaussian Splatting, Human Synthesis, Monocular Reconstruction

✦

1 INTRODUCTION

Free-view rendering of animatable humans is a challenging task with broad applications in Augmented Reality (AR), telepresence, movies and video production. Traditional methods [1]–[4] reconstruct high-resolution textured mesh and estimate material properties [3] to achieve photorealistic rendering, which often require meticulously arranged lab setups and costly equipments, such as dense RGB and IR camera arrays [3], [4] or 3D scanners [1], [2], making these methods impractical for general consumers.

The advent of neural rendering [5], particularly Neural Radiance Fields (NeRF) [6] has revolutionized novel view synthesis, enabling photorealistic human rendering [7], [8] and animation [7], [9]–[11] from sparse-view images. NeRF-based methods ease the traditional modeling setup and reach new levels of realism by learning a neural representation that can be optimized to minimize the render error. But they model such representation with large multi-layer perceptron (MLP), which requires extended training time ($>$ 10 hours), and cannot be rendered in real-time. While efficient neural implicit representations [12]–[17] have emerged, the efficiency of neural human reconstruction and rendering has yet not been satisfying [18], [19], due to the high memory and computational complexity of volumetric representation. Moreover, they still struggle to fit highly dynamic motions from monocular video, leading to blurry synthesized results.

Recently, 3D Gaussian Splatting [20] (3DGS), as a point cloud-like representation, has significantly surpassed NeRF-based methods in both quality and rendering speed. Leveraging 3DGS, some recent methods [21]–[27] have demonstrated its potential for dynamic human reconstruction and rendering. However, 3D Gaussians with ultimate expressive power tend to overfit the region with inconsistent view-dependent appearance rather than form a good geometry [28], leading to artifacts in distinct views [29], [30]. This problem, initially identified in static scenes, is exacerbated in monocularly captured dynamic scenes, where much severer inconsistency emerges from transient regions such as clothes wrinkles or shadows. Changing constantly with rapid human motion, these regions cannot provide consistent supervision for the Gaussians, resulting in ***noisy geometry*** and ***abrupt deformation*** that typically fail to generalize to novel views and poses.

On the other side of the spectrum, meshes have long been explored as a representation for human rendering and animation, offering consistent multi-view performance and well-defined geometry [31]–[34]. They are also easier to manipulate and generalize to new poses compared to Gaussians. To this end, some methods anchor the Gaussians on the mesh to regularize them with well-defined mesh geometry [28], [35]. However, due to great geometry and topology discrepancy between the SMPL [36] mesh and the real human surface, such rigid binding significantly limits the expressive power of 3DGS, making it extremely difficult to capture the highly non-rigid deformations caused by complex human motion, resulting in severe loss of realism.

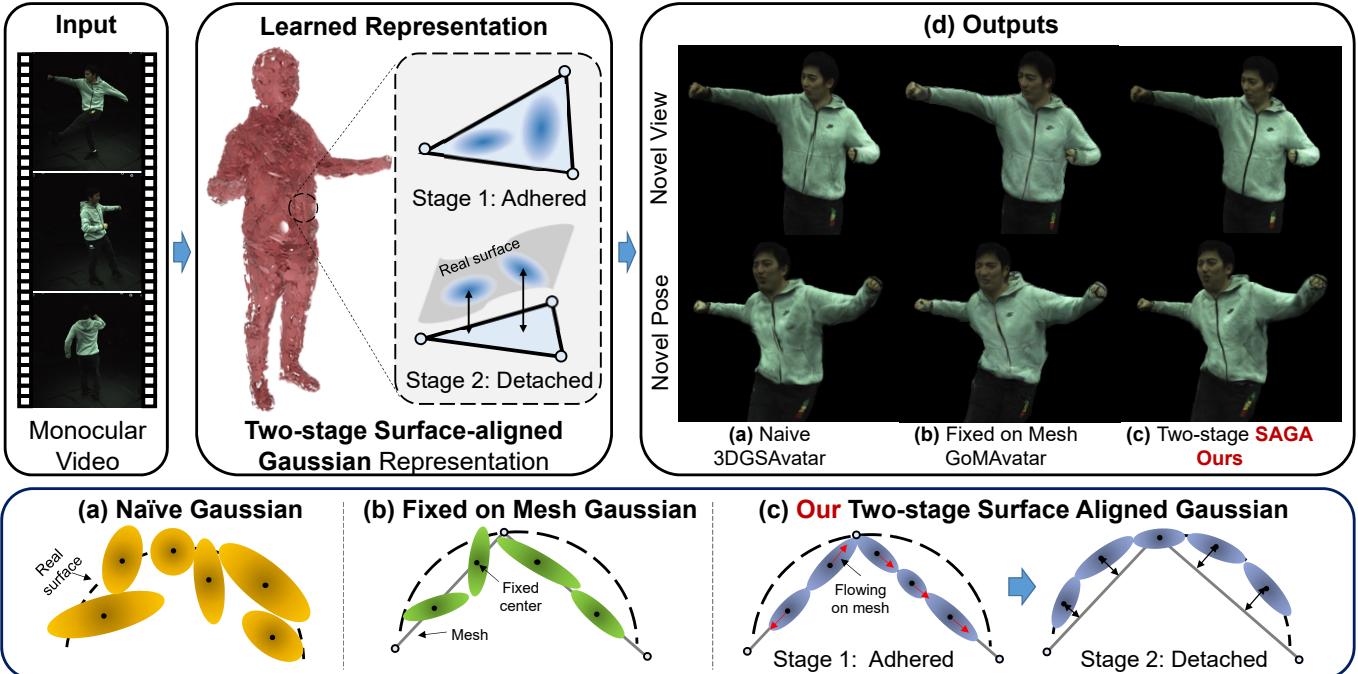To address these challenges, we propose a new Two-Stage

Fig. 1: **UPPER:** Illustration of SAGA, *i.e.* Surface-aligned Gaussian Avatar for monocular drivable avatar reconstruction and animation. **LOWER:** Since monocular dynamic reconstruction is severely ill-posed, state-of-the-art methods either **(a)** overfit the scene with naive Gaussians or **(b)** overconstrain the Gaussians by fixing them on the mesh. In contrast, **(c)** SAGA applies a first-adhered-then-detached manner to effectively regularize Gaussians without sacrificing the expressivity, **(d)** leading to more photorealistic rendering results.

Surface-Aligned Gaussian representation, which aligns the Gaussians with a coarse human mesh [36] to enforce the Gaussians to form **well-defined geometry** and learn **consistent deformation**, thereby improving novel view and pose generalization ability. Meanwhile, we aim at maximally maintaining the expressivity of the Gaussians to ensure the rendering realism by designing a two-stage alignment strategy. We name the method SAGA, i.e., Surface-Aligned Gaussian Avatar.

Specifically, in the two stages, the Gaussians are first **adhered** on the mesh to guide them to form a well-defined geometry, and then **detached** from the mesh to unleash the expressivity of 3DGS for fine structures. In the first stage, unlike previous methods that rigidly adhere the Gaussians on the mesh [28], [35], we allow them to flow freely to improve the flexibility. Specifically, we learn Gaussian centers by simultaneously optimizing barycentric coordinates and the corresponding triangle vertices, which not only ensures them to be strictly on the mesh but also allows local adjustment to better fit the scene. We also align their orientation by flattening the Gaussians and setting their normals to match the triangle normals. This design ensures that the mesh regulates the Gaussians to form good geometry while the Gaussians in turn drive the SMPL mesh to quickly align with the real surface.

In the second stage, the Gaussians are detached from the mesh to fit finer structures. To maintain the geometry quality of the detached Gaussians, we propose a *Gaussian-Mesh Alignment Regularization* to regularize them by minimizing the center and orientation offset from the mesh. This regularization can also serve as a deformation regularizer, since it constrains the deformed Gaussians to be bound to the same triangles across all the training frames. Finally, during optimization, the Gaussians may drift out of the bound triangles, resulting in incorrect mesh-based regularization. Thus, we propose an efficient *Walking-on-Mesh*

strategy to accurately update the corresponding triangles with minimal computational overhead.

Experiments on challenging datasets show that our method produces more photorealistic results in novel view synthesis, and generalizes better in novel pose synthesis. Additionally, while existing Gaussian-based methods achieve satisfactory rendering results, they cannot extract high-quality meshes from Gaussians. In contrast, SAGA significantly improves the geometry quality, marking the first successful attempt at direct high-quality mesh extraction from deformable Gaussians reconstructed from monocular human videos.

In summary, our main technical contributions include:

- We propose to leverage mesh as a geometric regularizer for Gaussians based monocular avatar reconstruction with a new two-stage surface-aligned representation, where the Gaussians are first adhered on the mesh to enforce well-defined geometry thereby preventing overfitting, and then detached to fully exploit the expressivity to fit finer details.
- An Adhered-on-Mesh representation that, contrary to existing methods that rigidly bind Gaussians to fixed locations, allows them to flow freely on the mesh with higher flexibility.
- A Gaussian-Mesh Alignment regularization that enforces the detached Gaussians to form well-defined geometry and learn more consistent deformation.
- Since the Gaussians can drift outside the bound triangle, we propose an efficient Walking-on-Mesh strategy to update the triangles, ensuring correct mesh-based regularization.
- Our method achieves SOTA novel view and pose synthesis results, and, to our knowledge, for the first time enables direct mesh extraction from deformable human Gaussians reconstructed from monocular videos.

2 RELATED WORK

We divide previous methods into NeRF-based human avatars, methods that leverage efficient NeRFs [12] to accelerate training and rendering, more recent Gaussian-based human avatar and finally methods that integrating Gaussians with meshes.

Neural Radiance Fields based Human Avatar. Given the unprecedented success of Neural Radiance Fields [6], many methods have applied NeRF to reconstruct and render humans from videos [7], [8], [10], [11], [37]–[40]. Since the original NeRF cannot model dynamic human motions, these methods leverage deformation priors such as skeletons [8], [37]–[39] or SMPL [11], [36] model [7], [10] to warp the neural fields. A line of works [8], [10], [11], [37], [40] build a canonical neural radiance field, and warp the points in each frame to the canonical space via inverse Linear Blend Skinning (LBS). AnimatableNeRF [37] learns a neural blend weight field. [10], [11] further introduce a non-rigid module to compensate detailed deformation. Other methods develop learnable [7] or hand-crafted embeddings [39] to encode the sampled points. NeuralBody [7] anchors latent codes to the vertices of a SMPL model, and diffuses to the whole observation space via sparse convolution. A-NeRF [39] handcrafts a skeleton-relative encoding, thus avoiding ill-posed inverse transformation.

Despite the impressive rendering quality, these methods generally take a long training time (>10h) to reconstruct only a *single* person, and cannot render in real-time. Though some generalizable methods achieve fast finetuning on new persons, they are limited to well-calibrated multi-view setting [41]–[43].

Efficient Neural Human Avatar. To enable efficient training and rendering of dynamic human video, recent methods [18], [19] have applied InstantNGP [12] as the human representation. InstantAvatar [19] further improves the efficiency by introducing an occupancy field to prune points from the empty space. Instant-NVR [18] designs specific hash embedders for each human part to adjust representational power based on part complexity thus accelerating the convergence. Though reducing training time to minutes, they still cannot achieve real-time rendering at ≥ 24 FPS. Another line of work builds efficient representation based on shape primitives, such as meshes [44], voxels [45] or patches [46] to effectively reduce the sampled points for acceleration. However, they generally require to reconstruct a more accurate template or bake a texture map from multi-view inputs, which cannot be applied to monocular videos and require days of training.

Gaussian based Human Avatar. Since 3D Gaussian Splatting [20] achieves significant breakthrough in novel view synthesis on static scenes in terms of rendering quality and efficiency, recent trend has shifted to transferring 3DGS from static scene to dynamic scene reconstruction [47]–[50], especially dynamic humans [21]–[27], [51]–[54].

Similar to NeRF-based methods, these methods typically model Gaussians in the canonical space, and apply LBS or DQB model to warp the Gaussians to different poses. For methods taking multi-view videos as inputs [26], [27], [54], [55], an image-to-image translator [56] is typically used to generate 2D texture map to model high-fidelity motion-dependent textures. Moreover, Animatable Gaussian [27] leverages a more accurate template and directly translates it to Gaussian parameters. GPS-Gaussian [26] proposes a generalizable NVS method, which first predicts depth map with stereo-based methods, and then regresses Gaussian parameters. Generally, the calibrated multi-view setup of these methods relatively limits their usage. Moreover, the image translator requires long training time of more than 10 hours.

For monocular based methods, 3DGS-Avatar [21] applies pose-dependent color and non-rigid MLP to predict color and deformation for each Gaussian in the canonical space. An as-isometric-as-possible regularization [57] is further introduced to regularize the deformation. A similar framework is also applied in [22], [24], [25], [53]. These methods typically require minutes of training, and render at > 50 FPS. However, due to the ill-posedness of dynamic monocular reconstruction and ultimate expressivity of Gaussians, they still suffer from overfitting, leading to undesirable artifacts under novel views and poses.

Integrating Gaussians with Mesh. Meshes are naturally multi-view consistent with well-defined geometry and easier to manipulate, offering better generalization ability. Some methods propose to integrate Gaussians with meshes to combine their advantages [28], [35], [58], [59]. SuGaR [28] extracts a mesh from Gaussians and re-anchors them onto the mesh for further refinement. However, this approach is not suitable for dynamic scenes, where it is more difficult to reconstruct an accurate enough mesh to anchor on. In the context of dynamic humans, most methods employ the parametric human model, SMPL [36]. For example, GoMAvatar [35] anchors Gaussians at the centers of mesh triangles. However, such constraints can be overly rigid and prevent the Gaussians from fitting the scene accurately, leading to loss of rendering realism. Moreover, they ignore the alignment of orientation. SplattingAvatar [58] defines Gaussian center as optimizable uv coordinates and distance above the triangle and does not optimize the SMPL mesh. It primarily focuses on leveraging the mesh to manipulate the Gaussians without providing adequate geometric regularization. Moreover, while lifted optimization techniques [60]–[62] are introduced to update the bound triangles, they are inefficient and prone to inaccurate triangles, thus, as we will show (Fig. 8, Tab. 6) being $200\times$ slower with artifacts. In contrast, we develop a flexible Surface-aligned Gaussian representation that effectively regularizes them to form a well-defined geometry without sacrificing the fitting ability, and a Walking-on-Mesh strategy that tracks precise triangles efficiently.

3 PRELIMINARY

3D Gaussians. As an explicit representation, 3DGS models a static scene as a set of 3D Gaussians [20] $\{\mathcal{G}_i\}_{i=1}^{N}$, where each Gaussian $\mathcal{G}_i$ is defined by mean $\mathbf{x}_i \in \mathbb{R}^3$ and covariance matrix $\mathbf{\Sigma}_i \in \mathbb{R}^{3\times3}$, and additionally assigned with opacity $\sigma_i \in \mathbb{R}^1$ and color $\mathbf{c}_i \in \mathbb{R}^3$ values:

$$\mathcal{G}_i = (\mathbf{x}_i, \mathbf{\Sigma}_i, \sigma_i, \mathbf{c}_i). \tag{1}$$

Intuitively, a 3D Gaussian is analogous to an ellipsoid. So $\mathbf{x}$ describes the center location, and $\mathbf{\Sigma}$ describes the scaling and orientation via the decomposition [20]:

$$\mathbf{\Sigma} = \mathbf{R}\mathbf{S}\mathbf{S}^T\mathbf{R}^T, \tag{2}$$

where $\mathbf{R} \in \mathbb{R}^{3\times3}$ is the rotation matrix determining the orientation, and $\mathbf{S} \in \mathbb{R}^{3\times3}$ is the diagonal scaling matrix defined as:

$$\mathbf{S} = \mathrm{diag}(\mathbf{s}), \tag{3}$$

where its diagonal entries are parameterized by the scaling vector $\mathbf{s} = [s_0, s_1, s_2]^T$ determining the scales along each axis.

Rendering 3D Gaussians. In contrast to sampling along rays in NeRF-based methods, 3D Gaussians [63] are rendered
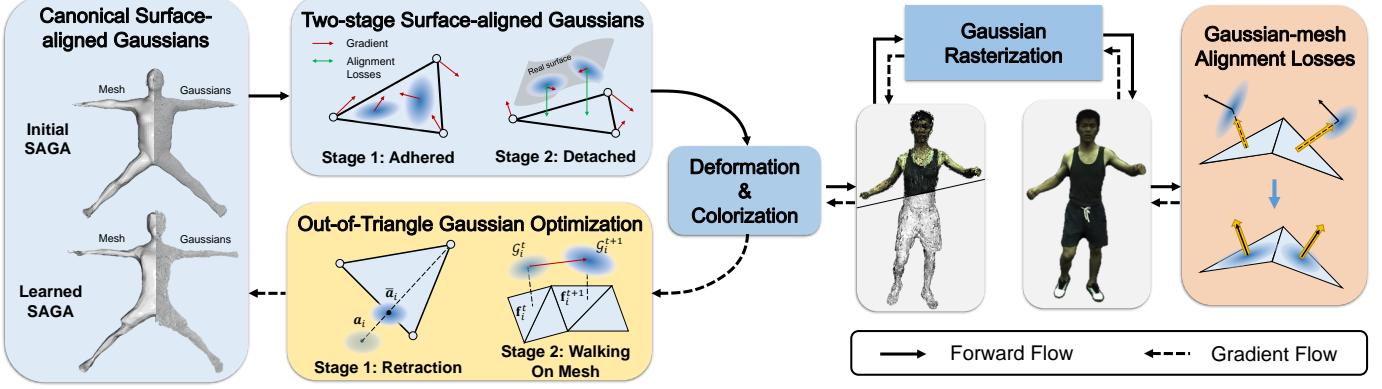
Fig. 2: **The framework of Surface-aligned Gaussian Avatar (SAGA).** We model the human with a *two-stage Surface-Aligned Gaussian* representation in the canonical space, where the Gaussians are first strictly adhered on the SMPL mesh (Stage 1, Sec. 4.1), and then detached from the mesh to fit finer details (Stage 2, Sec. 4.2). The canonical Gaussians are sent into the *Deformation & Colorization Module* to transform them to the observation space, predict the non-rigid deformation, and compensate the color changes caused by motion (Sec. 4.4). Finally, the Gaussians are rasterized to render the image. For backpropagation, we compute the *Gaussian-Mesh Alignment Losses* to regularize the deformed Gaussians to align with the mesh in the Detached Stage (Sec. 4.2.2). To prevent the incorrect regularization when a Gaussian moves outside the triangle, we use the proposed *retraction* and *Walking-on-Mesh* strategies to retract the Gaussian back within the triangle or update the new bound triangle in the first and second stages, respectively (Sec. 4.3).

more efficiently by projecting them onto 2D image: $\mathcal{G}_i^{2D} = (\mathbf{x}_i^{2D}, \mathbf{\Sigma}_i^{2D}, \sigma_i, \mathbf{c}_i)$, where $\mathbf{x}_i^{2D}$ is the projection of the 3D Gaussian mean $\mathbf{x}_i$ in the image, and the covariance $\mathbf{\Sigma}_i^{2D}$ is given by:

$$\mathbf{\Sigma}_i^{2D} = \mathbf{J}\mathbf{V}\mathbf{\Sigma}_i\mathbf{V}^T\mathbf{J}^T, \qquad (4)$$

where $\mathbf{V}$ is the viewpoint matrix, and $\mathbf{J}$ is the Jacobian of the projection matrix. Finally, the color of each pixel p is obtained by α-blending the 2D Gaussians that overlap it:

$$C(p) = \sum_i^N \alpha_i \prod_{j<i}(1-\alpha_j)\mathbf{c}_i, \qquad (5)$$

where α_i is given by the opacity σ_i multiplied by the probability of the pixel in the projected 2D Gaussian distribution.

Animatable 3D Gaussians. To reconstruct animatable 3D Gaussians, most methods decompose the dynamic human into canonical human Gaussians, and a deformation function to warp the Gaussians into different human poses. The deformation function is typically composed of articulated deformation that blends the rigid transformations of each human bone, and non-rigid deformation that models fine-grained deformation, such as clothes wrinkles or human expressions. Details of these modules are introduced in Sec. 4.4. We further propose a mesh-based regularization to regularize the non-rigid deformation which we introduce in Sec. 4.2.2.

4 METHOD

Given a monocular video of a human performer with estimated human poses and masks, our goal is to learn a Gaussian representation to rerender the video from free views, and animate the human to perform novel actions. Moreover, by exploiting the efficiency of 3DGS, we expect our method to inherit its fast training time and real-time rendering efficiency.

For monocular human reconstruction, naive Gaussians often suffer from overfitting and struggle to synthesize plausible results under novel views and poses. To address these challenges, we present SAGA, Surface-Aligned Gaussian Avatar, which leverages a template human mesh as a proxy regularizer to constrain the Gaussians to form well-defined surface that enhances generalization in novel view and pose rendering. Meanwhile, SAGA maximally maintains the expressivity of 3DGS with a flexible two-stage representation. As shown in Fig. 2, in stage 1, *i.e.*, the **Adhered Stage**, we adhere the Gaussians on the mesh to guide them to form a well-defined geometry (Sec. 4.1). Then in stage 2, *i.e.*, the **Detached Stage**, we detach the Gaussians from the mesh to unleash the expressivity for fine details (Sec. 4.2). To prevent detached Gaussians' geometry from being corrupted and constrain the non-rigid deformation, we introduce the *Gaussian-Mesh Alignment Regularization* in the detached stage (Sec. 4.2.2). Additionally, since the Gaussians may move outside their bound triangles during optimization, we develop a *retraction strategy* for the Adhered Stage and a *Walking-on-Mesh strategy* for the Detached Stage to accurately update the corresponding triangles (Sec. 4.3). Finally, we describe how the canonical Gaussians are colorized and transformed to the observation space in Sec. 4.4.

4.1 Stage 1: Adhering Gaussians on the surface

As pointed out in [20], one limitation of original 3DGS is its tendency to produce artifacts in regions that have view-dependent appearance due to inconsistent supervision. Originally found on static scenes, such limitation only becomes even severer on monocularly captured dynamic scene, where there are much more dynamic regions that change constantly following human motions.

We believe that well-defined geometry is a prior that helps to prevent such overfitting. Thus, we introduce the SMPL mesh as a proxy regularizer to enforce the Gaussians to form a well-defined surface. The key idea is to align the Gaussian position and orientation with the mesh. Former methods achieve this by binding one or several orderly arranged Gaussians rigidly at fixed location in each triangle [28], [35], [64]. Unfortunately, this ties Gaussians' density with mesh topology, and limits the expressivity. To solve this problem, we allow the Gaussians to flow freely on the mesh during optimization.
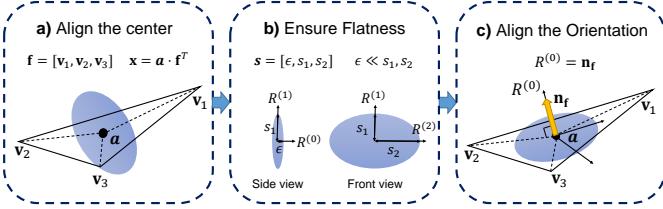
Fig. 3: **Illustration of the Adhered-on-Mesh Gaussian.** We first **a)** align the Gaussian center on the triangle by defining it based on barycentric coordinates. Then in **b)**, we make the Gaussian flat, and **c)** fix the direction of the smallest scale $\mathbf{R}^{(0)}$ as the triangle normal $\mathbf{n_f}$ to align the Gaussian orientation with the surface. Different from former fixed-on-mesh representation [28], [35], we simultaneously optimize the barycentric coordinates $\boldsymbol{a}$ and the mesh vertices $\mathbf{v}$, which allows Gaussians to flow on the mesh for higher flexibility while driving the mesh to fit the scene.

Specifically, we represent Gaussian centers as barycentric coordinates in the bound mesh triangles, and jointly optimize these coordinates and triangle vertices to fit the scene. We use the SMPL model [36] as a coarse human mesh in the canonical space, which is denoted as $\mathcal{M} = (V, F)$ composed of the vertex set $V = \{\mathbf{v}_i\}$ and the face set $F = \{\mathbf{f}_i\}$, where $\mathbf{v}_i \in \mathbb{R}^3$ is the 3D vertex coordinates, and $\mathbf{f}_i = [\mathbf{v}_{i,1}, \mathbf{v}_{i,2}, \mathbf{v}_{i,3}] \in \mathbb{R}^{3\times3}$ is a tuple of vertices that comprise the triangle face.

Here we take one Gaussian as an example, and omit the subscript of Gaussian index for clarity. As illustrated in Fig. 3 **a)**, for a Gaussian $\mathcal{G}$ bound to the triangle $\mathbf{f}$, we define its barycentric coordinates as $\boldsymbol{a} = [a_1, a_2, a_3] \in \mathbb{R}^{1\times3}$. Then, the Gaussian center $\mathbf{x}$ is computed by interpolating the face vertices $\mathbf{f}$ with the barycentric coordinates $\boldsymbol{a}$:

$$\mathbf{x} = \boldsymbol{a} \cdot \mathbf{f}^T = a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + a_3\mathbf{v}_3. \tag{6}$$

With the centers on the mesh, existing dynamic Gaussian representations [35], [64] often ignore the alignment of orientation. As a result, the Gaussians may stick out of the surface, causing artifacts. Thus, as illustrated in Fig. 3 **b)**, we further flatten the Gaussians and, in Fig. 3 **c)**, align the orientation with the mesh. Specifically, to ensure flatness, we set the first Gaussian scale $s_0 = \epsilon$ as a small constant value ϵ with $\epsilon \ll s_1, s_2$:

$$\mathbf{s} = [\epsilon, s_1, s_2] \tag{7}$$

In this way, each flat Gaussian can be regarded as a surfel, and its normal $\mathbf{n}_{\mathcal{G}}$ is the axis with the smallest scale, represented as $R^{(0)}$. Here the index 0 means the *first* column of the rotation matrix R, which corresponds to the index of the axis with the smallest scale. Then we align the Gaussians' orientation with the mesh by setting the Gaussian normal as the normal of its bound triangle $\mathbf{n_f}$:

$$R^{(0)} = \mathbf{n_f} = (\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_1). \tag{8}$$

To this end, the original 3 degree-of-freedom (DoF) rotation is reduced to only one DoF in-plane rotation denoted as an angle β.

In summary, the SAGA in the adhered stage can be fully parameterized by the following learnable parameters:

$$\mathcal{G} = (\boldsymbol{a}, \mathbf{f}, s_1, s_2, \beta). \tag{9}$$

4.2 Stage 2: Detaching for refinement

While the Adhered Stage ensures the Gaussians are well-aligned with the surface, the constraint could be overly restrictive and hinder the Gaussians to fit the scene accurately, due to the great discrepancy between the coarse SMPL mesh and real surface. To solve this problem, we introduce the second Detached Stage, where the strictly adhered-on-mesh constraint is relaxed, allowing Gaussians to detach from the mesh to fit finer details.

4.2.1 Detaching and rebinding

The *detached* Gaussian representation is defined as:

$$\mathcal{G} = (\mathbf{x}, \mathbf{S}, \mathbf{R}, \boldsymbol{a}, \mathbf{f}), \tag{10}$$

where we remove the strict center and orientation alignment constraints by resetting the Gaussian parameters as the original form (first three terms in Eq. 10).

We then *rebind* the Gaussians to the triangles to maintain a loose connection for further mesh-based regularizations. We achieve this by keeping the last two terms in Eq. 10 from the adhered Gaussians, *i.e.*, the barycentric coordinates $\boldsymbol{a}$ and the bound triangle $\mathbf{f}$. Different from the first stage, we no longer use these terms to compute the Gaussian center but directly optimize the center $\mathbf{x}$ through gradient descent. Moreover, we do not optimize the barycentric coordinates $\boldsymbol{a}$, but directly obtain it analytically by projecting the Gaussian center $\mathbf{x}$ on the corresponding triangle $\mathbf{f}$:

$$\boldsymbol{a} = \mathrm{Proj}(\mathbf{x}, \mathbf{f}), \tag{11}$$

We provide the detailed derivation of this equation in the supplementary material.

After rebinding, we develop several regularizations between each Gaussian and the bound triangle to constrain the detached Gaussians, which are introduced as follows.

4.2.2 Gaussian-Mesh alignment regularization

We align the detached Gaussians' geometry with the mesh through position and orientation alignment losses, which also enforce smoother and more consistent non-rigid deformation by keeping the deformed Gaussians being bound to the same triangle across all training frames. This is illustrated in Fig. 4.

Fig. 4: **Illustration of the Gaussian-mesh alignment losses**, which consists of a position and a normal alignment loss.

Position Alignment Loss. For a Gaussian $\mathcal{G}$, the position-alignment loss is defined as the minimum distance between the Gaussian center $\mathbf{x}$ and the corresponding triangle $\mathbf{f}$:

$$L_{\mathrm{pa}} = \sum_{i=0}^{N} \min_{\mathbf{p} \text{ in } \mathbf{f}} \|\mathbf{x} - \mathbf{p}\|^2, \tag{12}$$

where $\mathbf{p}$ is a point within the triangle $\mathbf{f}$. Note that comparing to a naive point-to-plane distance, this loss additionally confines the Gaussians within the triangle from the tangential direction.

Orientation Alignment Loss. We align the orientation of the Gaussians with the mesh by penalizing the difference between their normals. Specifically, we use the cosine distance:

$$L_{\mathrm{na}} = 1 - | < \mathbf{n}_{\mathcal{G}}, \mathbf{n}_{\mathbf{f}} > |, \tag{13}$$

$$\mathbf{n}_{\mathcal{G}} = \mathbf{R}^{(j)} \quad j = \underset{j}{\operatorname{argmin}} \{s_j\}_{j=1}^3, \tag{14}$$

where $| < \cdot, \cdot > |$ denotes the absolute value of the cosine similarity, and $\mathbf{n}_{\mathcal{G}}$ denotes the Gaussian normal, which is the axis with the smallest scale in the rotation matrix $\mathbf{R}$.

Mesh-based Regularization. With the above alignment losses, we can use the mesh as a proxy to transfer desired property, e.g., smoothness, from the mesh to Gaussians with sophisticated mesh-based regularizations. Here, we introduce a Laplacian-based and a normal-based mesh smoothness regularization:

$$L_s = \lambda_{\mathrm{lap}} L_{\mathrm{lap}} + \lambda_{\mathrm{normal}} L_{\mathrm{normal}}, \tag{15}$$

where the Laplacian loss L_{lap} minimizes differences of adjacent vertices weighted by cotangent weights, and the normal loss L_{normal} minimizes the cosine distance of adjacent face normals.

4.3 Optimizing the out-of-triangle Gaussians

During optimization, the Gaussians may drift outside their bound triangles. This causes the Gaussians to be aligned with wrong triangles, thus resulting in incorrect regularization that corrupts the geometry.

To address this, we propose two strategies: **1) Retraction** that moves the out-of-triangle Gaussian back within the same triangle, which is simple but somewhat limits the flexibility, and **2) Walking-on-Mesh** that updates the bound triangle as the Gaussian moves, which is more flexible but could be expensive.

We apply the **retraction** in the Adhered Stage for simplicity, and design an efficient **Walking-on-Mesh** strategy in the Detached Stage to maintain the expressivity for fitting finer details. Details of these two strategies are introduced as follows.

4.3.1 Retraction

(a) Within Triangle (b) Out of Triangle (c) Retraction

$\forall k \in \{1,2,3\}, a_k \geq 0$ $\exists k \in \{1,2,3\}, a_k < 0$ $\bar{a}_k = \begin{cases} 0 & a_k < 0 \\ a_k & 0 \leq a_k < 1 \\ 1 & a_k \geq 1 \end{cases}$
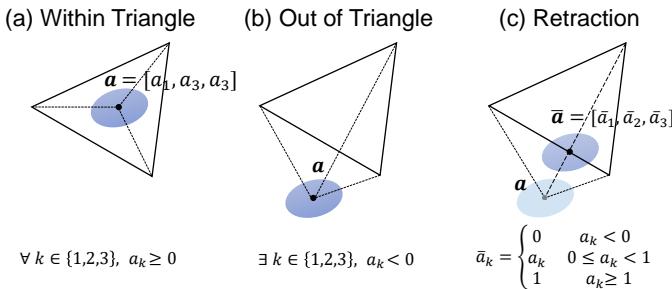
Fig. 5: **Illustration of the retraction strategy** for out-of-triangle Gaussians optimization in the Adhered Stage.

As shown in Fig. 5(a)(b), we first determine if a Gaussian $\mathcal{G}$ is out of the triangle based on the barycentric coordinates $\boldsymbol{a} = (a_1, a_2, a_3)$:

$$\mathrm{OutOfTriangle}(\boldsymbol{a}) = \begin{cases} 0 & \forall k \in \{1,2,3\}, \frac{a_k}{a_1+a_2+a_3} \geq 0 \\ 1 & \exists k \in \{1,2,3\}, \frac{a_k}{a_1+a_2+a_3} < 0, \end{cases} \tag{16}$$

which means that a Gaussian is inside the triangle if all the normalized barycentric coordinates a_k are ≥ 0, and is outside

the triangle if one of the normalized barycentric coordinates a_k is < 0. Then, we retract the out-of-triangle Gaussian back on the closest edge in Fig. 5(c), via:

$$\bar{a}_k = \begin{cases} 0 & a_k \leq 0 \\ a_k & 0 < a_k \leq 1 \\ 1 & a_k > 1 \end{cases}, \quad k \in \{1,2,3\} \tag{17}$$

where $\bar{a}_k$ denotes the retracted barycentric coordinates.

4.3.2 Walking-on-Mesh strategy

(a) Gaussian Update   (b) Check if outside by computing Bary. Coord.   (c) Find the nearest triangle $\mathbf{f}^{t+1}$ (d) Adjacent set $\mathbf{f}_j \in \mathcal{A}(\mathbf{f}^t)$
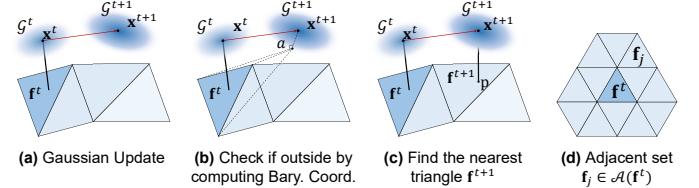
Fig. 6: **Illustration of the Walking-on-Mesh strategy** applied in the Detached Stage. **(a)** An optimization step updates Gaussian $\mathcal{G}^t$ bound with triangle $\mathbf{f}^t$ to $\mathcal{G}^{t+1}$. **(b)** We first check whether the updated Gaussian $\mathcal{G}^{t+1}$ is out of the current triangle $\mathbf{f}^t$ based on projected bary. coord. $\boldsymbol{a}$ (Eq. 16). **(c)** If it is, we update its bound triangle as the closest one in the adjacent set $\mathcal{A}(\mathbf{f}^t)$ shown in (d).

In the Detached Stage, we update the bound triangle as a Gaussian drifts outside the current one by proposing a Walking-on-Mesh algorithm.

As illustrated in Fig. 6, given a Gaussian $\mathcal{G}^t$ centered at $\mathbf{x}^t$ and bound to triangle $\mathbf{f}^t$, an optimization step updates the Gaussian position to $\mathbf{x}^{t+1}$. The Walking-on-Mesh algorithm finds the new triangle $\mathbf{f}^{t+1}$ that the Gaussian should be bound to. We define the new triangle $\mathbf{f}^{t+1}$ as the nearest triangle to the new center $\mathbf{x}^{t+1}$ based on Euclidean distance. However, naively finding the closest triangle for tens of thousands of Gaussians on the whole human mesh with $> 130,000$ triangles is expensive and significantly decreases the training speed.

Thus, we **first** reduce the number of walking Gaussians by only considering the out-of-triangle Gaussians. As shown in Fig 6(b), we project the new center $\mathbf{x}^{t+1}$ back to the current face f^t and obtain the barycentric coordinates $\boldsymbol{a}$ (Eq. 11). Then we determine whether it is outside via Eq 16.

Secondly, we reduce the search scope to a local triangle set $\mathcal{A}(\mathbf{f}^t)$ that is adjacent to the current triangle $\mathbf{f}^t$:

$$\mathcal{A}(\mathbf{f}^t) = \{\mathbf{f}_j | \mathbf{f}_j \cap \mathbf{f}^t \neq \phi, \mathbf{f}_j \in F\}, \tag{18}$$

where each adjacent triangle $\mathbf{f}_j$ should share at least one vertex with the current bound triangle $\mathbf{f}^t$. This significantly reduces the search scope from hundreds of thousands triangles to ~ 12.

Finally, the updated triangle $\mathbf{f}^{t+1}$ is defined as the adjacent triangle that is nearest to the Gaussian center $\mathbf{x}^{t+1}$:

$$\mathbf{f}^{t+1} = \underset{\mathbf{f}_j}{\operatorname{argmin}} \min_{\mathbf{p} \in \mathbf{f}_j} \|\mathbf{x}^{t+1} - \mathbf{p}\|^2, \quad \mathbf{f}_j \in \mathcal{A}(\mathbf{f}^t). \tag{19}$$

The whole process is summarized in Algorithm 1.

4.4 Pose-driven Gaussian Deformation & Colorization

We use the pose-dependent deformation and colorization modules to warp the canonical Gaussians to the observation space and compensate the appearance changes. It includes an articulated

Algorithm 1 Walking on mesh

1: **Input:** $\mathbf{x}^t, \mathbf{f}^t, \mathbf{x}^{t+1}, F$
2: **Output:** $\mathbf{f}^{t+1}$
3: $\quad \mathbf{a} \leftarrow ComputeBaryCoordinates(\mathbf{x}^{t+1}, \mathbf{f}^t)$ ▷ Eq. 11
4: **if** OutOfTrianlge($\mathbf{a}$) **then**   ▷ Eq. 16
5: $\quad\quad \mathcal{A} \leftarrow AdjacentFaceSet(\mathbf{f}^t, F)$ ▷ Eq. 18
6: $\quad\quad$ **for** Faces $\mathbf{f}_j \in \mathcal{A}$ **do**
7: $\quad\quad\quad D[j] \leftarrow PointMeshDistance(\mathbf{x}^{t+1}, \mathbf{f}_j)$
8: $\quad\quad$ **end for**
9: $\quad\quad j \leftarrow \arg\min_j D$
10: $\quad\quad$ **return** $\mathbf{f}_j$
11: **else**
12: $\quad\quad$ **return** $\mathbf{f}^t$
13: **end if**

---

deformation module, a non-rigid deformation module, and a pose-dependent colorization module.

Articulated deformation module. To render Gaussians under arbitrary human poses $\boldsymbol{\theta}$, we use a forward articulated deformation function $\mathcal{W}$ to warp the canonical Gaussians $\mathcal{G}_c$ to the posed Gaussians $\mathcal{G}_o$ in the observation space:

$$\mathcal{G}_o = \mathcal{W}(\mathcal{G}_c, \boldsymbol{\theta}), \tag{20}$$

Specifically, we apply Linear Blend Skinning (LBS) as the warp function $\mathcal{W}$, which defines a local transformation $[\mathbf{A}, \mathbf{b}]$ for each Gaussian based on the input human pose $\boldsymbol{\theta}$:

$$[\mathbf{A}, \mathbf{b}] = \sum_{b=1}^{B} w_b(\mathbf{x}_c)[\mathbf{R}_b(\boldsymbol{\theta}), \mathbf{t}_b(\boldsymbol{\theta})], \tag{21}$$

where $\mathbf{x_c} \in \mathbb{R}^3$ is the Gaussian center in the canonical space, $b \in 1, ..., B$ denotes B human bones, $\mathbf{R}_b(\boldsymbol{\theta}) \in \mathrm{SO}(3)$ and $\mathbf{t}_b(\boldsymbol{\theta}) \in \mathbb{R}^3$ is the rotation and translation of bone b depending on the human pose $\boldsymbol{\theta}$, $[\cdot, \cdot]$ denotes the concatenation, and w_b is the blending weight of the point $\mathbf{x}_c$. Finally, the outputs are the blended rotation $\mathbf{A} \in \mathbb{R}^{3 \times 3}$, and translation $\mathbf{b} \in \mathbb{R}^3$.

Then, the canonical Gaussians are transformed to the observation space by:

$$\mathbf{x}_o = \mathbf{A}\mathbf{x}_c + \mathbf{b}, \quad \boldsymbol{\Sigma}_o = \mathbf{A}\boldsymbol{\Sigma}_c\mathbf{A}^T. \tag{22}$$

Non-rigid Gaussian deformation. The above articulated deformation alone cannot capture fine-grained clothes deformation. Thus, we further introduce a non-rigid deformation module [8], [9], [11], [21], [65].

To fit the high-frequency non-rigid deformation efficiently, we adopt the multi-resolution hash encoder MHE [12] as the spatial encoder, and decode the deformation via a shallow MLP:

$$\Delta\mathbf{x}, \Delta\mathbf{s}, \Delta\mathbf{q} = \mathrm{MLP}_{\mathrm{NR}}(\mathrm{MHE}(\mathbf{x}), f(\boldsymbol{\theta})), \tag{23}$$

where the input is the spatial embedding $\mathrm{MHE}(\mathbf{x})$ and a latent code $f(\boldsymbol{\theta})$, which encodes human pose $\boldsymbol{\theta}$ via a human pose encoder f [66], and the outputs are the center offset $\Delta\mathbf{x}$, rotation offset $\Delta\mathbf{q}$ represented as quaternions and the scale changes $\Delta\mathbf{s}$.

Since the non-rigidly deformed Gaussians will deviate from the mesh, we only apply the module in the second Detached Stage.

Pose-driven colorization. Human motion causes shadows on the surface. This leads to inconsistent textures for the same Gaussians rendered in different frames. Due to the strong expressive ability

of 3D Gaussians, directly learning appearance by optimizing per-Gaussian color leads to overfitting. Thus, we use an MLP to learn pose-dependent Gaussian color $\mathbf{c}$. Specifically, we condition the color with a per-frame latent vector $\boldsymbol{\psi} \in \mathbb{R}^{16}$ for global lighting change due to self-rotating, and a pose-aware latent vector $h(\boldsymbol{\theta})$ for local shading caused by wrinkles:

$$\mathbf{c} = \mathrm{MLP}_{\mathrm{RGB}}(\mathbf{x}, \boldsymbol{\psi}, h(\boldsymbol{\theta})), \tag{24}$$

where we obtain $h(\boldsymbol{\theta})$ by extracting the latent feature from the intermediate layer of the non-rigid network $\mathrm{MLP}_{\mathrm{NR}}$.

5 OPTIMIZATION

5.1 Training objective

The final training objective function is composed of three main terms:

$$L = L_{\mathrm{app}} + L_{\mathrm{geo}} + L_{\mathrm{smooth}}, \tag{25}$$

where L_{geo} is the Gaussian-Mesh alignment term introduced in Sec. 4.2.2, and L_{smooth} is the smooth regularization defined in Eq. 15.

For L_{app}, we minimize the difference between the rendered and ground-truth images, defined as:

$$L_{\mathrm{app}} = L_1 + \lambda_{\mathrm{mask}} L_{\mathrm{mask}} + \lambda_{\mathrm{LPIPS}} L_{\mathrm{LPIPS}}, \tag{26}$$

where the L_1 and L_{mask} minimize the L1 norm of the RGB difference and opacity difference between the rendered and ground-truth images, respectively. L_{LPIPS} is the perceptual loss [67], which improves image sharpness when small misalignment exists.

5.2 Implementation details

SAGA is trained for 15k iterations, with 3k iterations for the first Adhered Stage and 12k iterations for the Detached Stage on one NVIDIA RTX3090 GPU, taking 12 minutes on average. We simultaneously optimize the Gaussian parameters, non-rigid deformation module and the colorization module with the Adam optimizer [68]. The Gaussian parameters are optimized with the same learning rate and scheduler as used in 3DGS [20]. We mute the non-rigid deformation until after 3k iterations. The initial learning rate of the non-rigid and colorization networks is set as 1×10^{-3} and gradually decayed by 0.1 with the exponential scheduler. During inference, SAGA renders a 512×512 image in real-time at 60 FPS on one NVIDIA RTX3090 GPU. Please refer to the supplementary material for more details.

6 EXPERIMENT

We evaluate SAGA against state-of-the-art methods for novel view and pose synthesis, as well as geometry reconstruction, using monocular videos from challenging datasets [7], [9], [69].

6.1 Datasets

ZJU-MoCap [7] dataset contains 23-view videos of 9 human subjects performing complex dynamic motions, such as kicking and swirling. We conduct novel view synthesis on 6 commonly used subjects [8]. For training, we select $\sim$500 frames from *camera 4* for training, and use the rest 22 views for evaluation.
MonoCap dataset [9] consists of multi-view videos from Deep-Cap dataset [31] and DynaCap dataset [33] selected by [9]. We use 500 frames from one view for training and evaluate on 10 other novel views that distribute uniformly, following [18], [22].

Ground Truth	Ours	3DGS-Avatar	GoMAvatar	GauHuman	Instant-NVR	HumanNeRF	AnimNeRF
Training Time:	12 min	30 min	30 hours	2 min	5 min	72 hours	10 hours
Render Speed:	60 FPS	50 FPS	40 FPS	150 FPS	5 FPS	0.4 FPS	4 FPS

Fig. 7: **Comparison results of novel view synthesis on ZJU-MoCap dataset [7] and MonoCap dataset [9].** For Gaussian based methods, 3DGS-Avatar [21] suffers from artifacts, GoMAvatar [35] struggles to fit details in the faces and hands, and GauHuman [22] synthesizes oversmoothed results, while other NeRF-based methods generally suffer from blurred [18], [37] or distorted results [8]. In contrast, our method achieves more photorealistic rendering results, and is also more efficient than most methods except for GauHuman and Instant-NVR, which, however, cannot fit high-frequency details within such short training time.

PeopleSnapshot [69] dataset contains monocular videos of humans that self-rotate in a fixed A-pose. We conduct experiments on 4 subjects, and follow the protocol of Anim-NeRF [70] to train all comparison methods on their optimized pose parameters. We evaluate on images rendered under a white background following [19], [70].

6.2 Baselines

We compare our proposed SAGA with **NeRF-based methods** [7], [8], [18], [19], [37], and more recent **3DGS-based methods** [21], [22], [35], [58]. For NeRF-based methods: NeuralBody [7] anchors latent codes on the SMPL mesh and diffuse in 3D space with 3D convolution networks for neural volume rendering; AnimatableNeRF [37] represents the scene with a large MLP, and learns a forward and backward blending weight MLP for animation; HumanNeRF [8] further incorporates a non-rigid deformation network and achieves SOTA performance; InstantAvatar [19] applies the efficient Instant-NGP [12] as the canonical representation; InstantNVR [18] designs multi-part hash encoder based on [12]. For Gaussian based methods, GauHuman [22] applies naive Gaussians in the canonical space, and optimizes a blending weight offset network; 3DGSAvatar [21] further learns the non-rigid deformation and the pose-dependent colors; GoMAvatar [35] binds the Gaussians rigidly on the mesh with fixed location, and does not align the orientation; SpattingAvatar [58] parameterizes

			377			386			387			392			393			394	
Subjects																				
Metrics	Train↓	FPS↑	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeuralBody [7]	10h	4	29.11	0.9674	40.95	30.54	0.9678	46.43	27.00	0.9518	59.47	30.10	0.9642	53.27	28.61	0.9590	59.05	29.10	0.9593	54.55
AnimNeRF [37]	10h	4	29.12	0.9727	26.58	32.94	0.9695	36.04	27.93	0.9601	41.76	29.50	0.9635	39.45	27.64	0.9566	43.17	29.15	0.9595	38.08
HumanNerf [8]	72h	0.4	31.12	0.9774	22.80	33.31	0.9726	33.48	**28.27**	0.9617	38.89	31.34	0.9712	33.57	29.19	0.9644	34.67	30.74	0.9662	34.67
InstantNVR [18]	0.1h	5	31.36	0.979	26.03	33.53	0.977	33.02	28.11	0.963	46.96	32.03	0.973	39.30	29.55	0.964	46.29	31.39	0.969	40.00
GoMAvatar [35]	30h	40	31.11	0.9787	21.04	33.26	0.9764	26.63	27.87	0.9616	37.03	31.28	0.9721	30.75	29.06	0.9640	33.97	30.46	0.9655	31.12
GauHuman [22]	**2min**	**150**	32.04	0.9751	19.13	33.77	0.9681	28.62	28.26	0.9548	38.72	32.17	0.9648	30.02	29.75	0.9565	35.27	31.46	0.9589	30.82
3DGSAvatar [21]	0.5h	50	30.88	0.9785	19.09	33.38	0.9772	25.65	27.75	0.9630	34.71	31.88	0.9742	29.30	29.28	0.9659	32.46	30.68	0.9679	28.93
Ours	0.2h	60	**31.64**	**0.9806**	**18.61**	**33.80**	**0.9788**	**25.54**	28.06	**0.9648**	**34.20**	**32.30**	**0.9752**	**28.83**	**29.77**	**0.9680**	**32.15**	31.29	**0.9699**	**28.48**

TABLE 1: **Qualitative results of novel view synthesis on ZJU-MoCap [7].** Performance is evaluated with PSNR, SSIM and LPIPS metrics. Our method outperforms all comparison methods on SSIM and LPIPS, while being more efficient than most of them in training and rendering. Though InstantNVR and GauHuman are faster, our method can surpass them on challenging subjects by 37% and 13%.

			Lan			Marc			Olek			Vlad		
Subjects														
Metrics	Train↓		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
AnimNeRF [9]	10 hours		31.40	0.9863	0.0183	30.81	0.9834	0.0242	34.18	0.9880	0.0155	27.90	0.9810	0.0200
HumanNeRF [8]	72 hours		33.50	0.9895	0.0134	34.66	0.9904	0.0164	34.08	0.9895	0.0143	28.49	0.9814	0.0178
InstantNVR [18]	10 min		32.78	0.9871	0.0171	33.84	0.9894	0.0169	34.52	0.9892	0.0139	28.70	0.9830	0.0197
GauHuman [22]	**2 min**		33.53	0.9852	0.0108	34.68	0.9855	0.0159	34.77	0.9850	0.0134	28.46	0.9796	0.0182
GoMAvatar [35]	30 hours		33.37	0.9887	0.0120	34.17	0.9881	0.0128	34.41	0.9866	0.0134	27.55	0.9782	0.0202
3DGSAvatar [21]	30 min		34.27	0.9900	0.0099	35.40	0.9906	0.0134	34.69	0.9893	**0.0115**	**29.06**	0.9837	**0.0158**
Ours	12 min		**34.37**	**0.9901**	**0.0098**	**35.59**	**0.9908**	**0.0128**	**34.86**	**0.9895**	0.0116	28.92	**0.9837**	0.0165

TABLE 2: **Quantitative results of novel view synthesis on MonoCap dataset [9].** Our method outperforms the comparison methods on most subjects with fast training speed.

Gaussian location with uv coordinates and offset d along the mesh normal, but fixes the mesh vertices.

6.3 Comparison Results on Novel View Synthesis

6.3.1 Results on ZJU-MoCap and Monocap dataset

We evaluate all comparison methods by running their official code. Differences from the results reported in the original papers are due to different settings including the training view, frame number, or implementation of evaluation metrics, which are all unified in our experiments for a fair comparison.

We present quantitative results in Tab. 1. Our proposed SAGA consistently outperforms the state-of-the-art Gaussian and NeRF-based methods on the SSIM and LPIPS metrics. Moreover, SAGA also achieves *third* highest training efficiency of ∼12 minutes and *second* highest real-time rendering speed at 60 FPS, which is comparable to the most efficient methods InstantNVR [18] and GauHuman [22]. Although they can be trained more efficiently, they struggle at fitting fine deformation within such short training time. As a result, our method can notably surpass them on challenging subject by 37% and 13% respectively. Notably, compared to the only method that aligns Gaussian on mesh, i.e., GoMAvatar [35], our method is 150× faster while achieving higher rendering quality. We attribute this to the higher flexibility of our two-stage Gaussian alignment strategy over their rigidly binding on mesh strategy. Since their overconstrained Gaussians struggle to move to the real surface, it requires more steps of careful optimization to converge.

For the qualitative results shown in Fig. 7, our method can synthesize higher quality results with more realistic details, such as clothes wrinkles, fists and human faces, while former NeRF-based method typically generates oversmoothed results. Human-NeRF [8] performs well but requires three days of training and still produces artifacts in thin structures like zippers. For Gaussian based methods, 3DGS-Avatar [21] suffers from noisy textures in novel views (Row 1), and fails to synthesize consistent results

at details, such as the zipper (Row 1) and the lower rim of the shirt (Row 2). GoMAvatar [35] struggles with facial realism and detailed features like fists and drawstrings due to its fixed-on-mesh representation. In contrast, our method can synthesize more plausible and photorealistic results with well-preserved details, demonstrating the superiority of the proposed surface-aligned representation in regulating Gaussians to improve multi-view consistency without compromising the rendering realism.

6.3.2 Results on PeopleSnapshot dataset

As shown in Tab. 3, our method surpasses the comparison methods on all subjects in terms of LPIPS metric. Though SplattingA-vatar [58] achieves higher PSNR and SSIM metrics on the *Female-3-causal* subject, we believe it is because these metrics favor the smoothed blurry results over sharper but slightly misaligned grid-like texture of the sweater (Fig. 8 Col. 4). Contrastively, our method preserves the details of the sweater, and surpasses SplattingAvatar by 42% in terms of LPIPS, which we believe to be more persuasive under this scene.

We show qualitative results in Fig. 8. Our method synthesizes more photorealistic results, especially for finer structures such as hands, faces, and wrinkles. NeRF-based method [19] suffers from salt noises from the close-up images due to the ray sampling in the volume rendering process. For Gaussian-based methods, without proper regularization, 3DGS-Avatar and SplattingAvatar tend to generate artifacts at the armpit and shoulder, while mesh-based GoMAvatar struggles to synthesize realistic human faces (Row 2).

6.4 Comparison results on novel pose synthesis

To evaluate novel pose synthesis performance, we animate the reconstructed Gaussians with out-of-distribution pose sequences from AIST++ [71] and AMASS [72] datasets, containing complex dancing motions.

Ground Truth   Ours   3DGS-Avatar   SplattingAvatar   GoMAvatar   Instant-Avatar
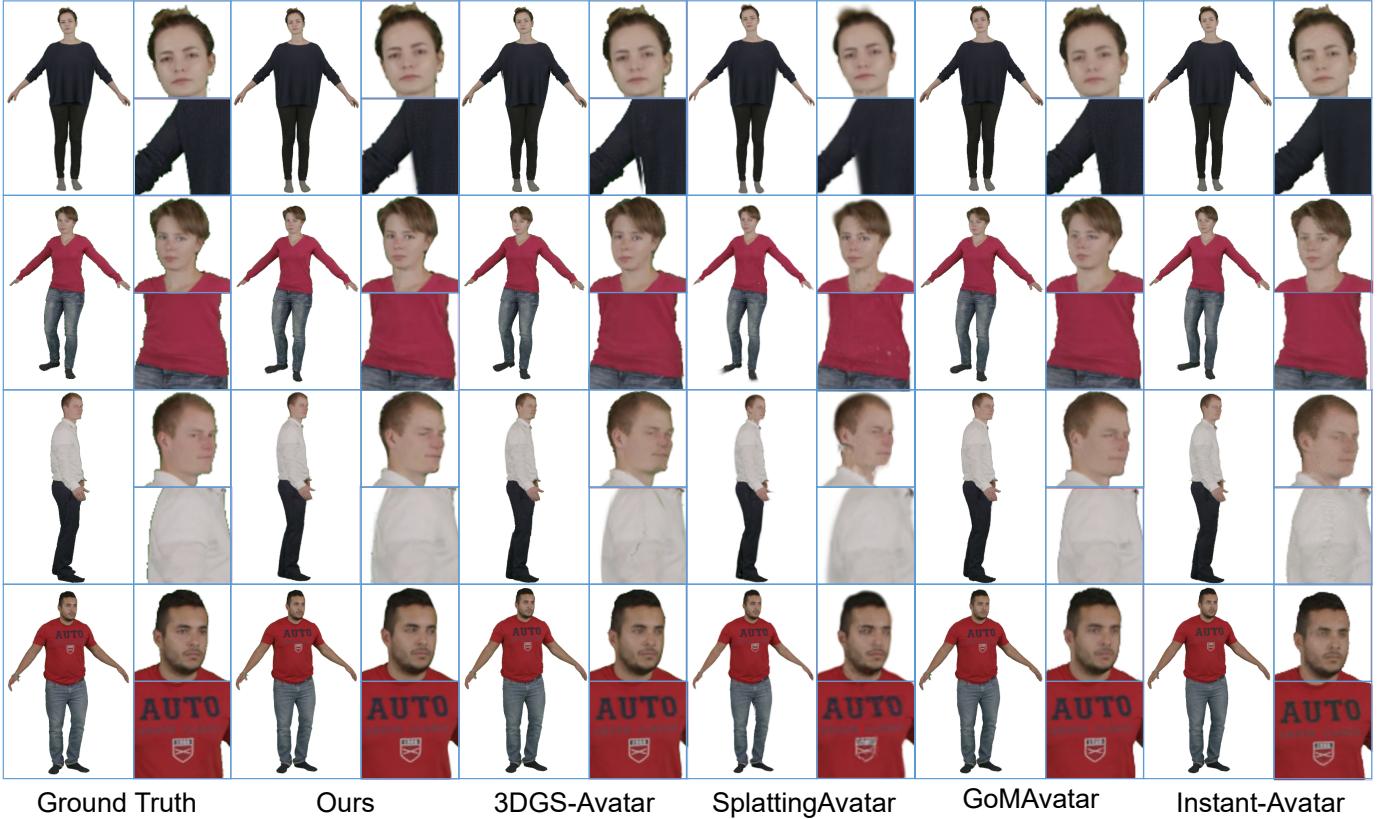
Fig. 8: **Comparison results of novel view synthesis with State-of-the-art methods on PeopleSnapshot Dataset [69].** All methods are Gaussian-based, except the last column that is NeRF-based. Our method can synthesize more photorealistic images, which are free of the artifacts (Row 1 Col. 3, Row 2 Col. 4, Row 3 Col. 4) and distortions (Row 2 Col. 3) in 3DGS-Avatar [21] and SplattingAvatar [58], while being shaper than GoMAvatar [35] and InstantAvatar [19] with higher fidelity.

	Subjects		male-3-casual			male-4-casual			female-3-casual			female-4-casual		
	Metrics	Train↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF	Neural Body [7]	∼ 14h	24.94	0.9428	0.0326	24.71	0.9469	0.0423	23.87	0.9504	0.0346	24.37	0.9451	0.0382
	Anim-NeRF [70]	∼ 13h	29.37	0.9703	0.0168	28.37	0.9605	0.0268	28.91	0.9743	0.0215	28.90	0.9678	0.0174
	InstantAvatar [19]	**1 min**	29.65	0.9730	0.0192	27.97	0.9649	0.0346	27.90	0.9722	0.0249	28.92	0.9692	0.0180
Gaussian	3DGSAvatar [21]	45 min	31.82	0.9800	0.0196	29.67	0.9755	0.0302	29.49	0.9736	0.0214	30.28	0.9769	0.0188
	SplattingAvatar [58]	18 min	32.47	0.9784	0.0243	30.74	0.9764	0.0347	**30.65**	**0.9784**	0.0343	31.19	0.9763	0.0287
	GoMAvatar [35]	20 h	31.74	0.9793	0.0187	29.78	0.9738	0.0282	29.83	0.9758	0.0209	31.38	0.9780	0.0174
	Ours	12 min	**33.15**	**0.9828**	**0.0135**	**30.95**	**0.9776**	**0.0248**	29.85	0.9760	**0.0196**	**31.92**	**0.9784**	**0.0145**

TABLE 3: **Quantitative results on the PeopleSnapshot dataset [69].** Our method outperforms all the Gaussian and NeRF-based methods on LPIPS metric while being the second fastest method in terms of training time, and notably surpasses the fastest method InstantAvatar [19] in terms of rendering quality.

6.4.1 Novel pose synthesis results on ZJU-MoCap dataset

As shown in Fig. 9, our method produces consistently high-quality results on out-of-distribution poses with well-preserved details at the zipper, hands and even tiny buttons. In contrast, the comparison methods generally suffer from artifacts. For instance, 3DGS-Avatar [21] synthesizes blurred zipper, and fractures at the armpit. GoMAvatar [35] and GauHuman [22] are limited by lower-quality reconstructions during training. Specifically, GoMAvatar synthesizes distorted faces and hands due to the constrained fitting ability of its fixed-on-mesh representation. GauHuman produces oversmoothed results. For NeRF-based methods, Instant-NVR [18] suffers from severe artifacts due to the poor generalization ability of their multi-part hash encoder when applied to novel poses. HumanNeRF [8] produces unnatural deformations leading to distorted faces and bodies.

6.4.2 Novel pose results on PeopleSnapshot dataset

Animating the subjects from the PeopleSnapshot dataset with novel poses can be more challenging because the pose variety in this dataset is more limited, containing only self-rotating A-pose.

As shown in Fig. 10, our synthesized images maintain the high quality as on the seen poses. Contrastively, 3DGS-Avatar [21] suffers from needle-like artifacts at the joints, due to the inaccurate LBS weights learned during training. Our surface-aligned representation allows the Gaussians to inherent more natural LBS weight from the SMPL mesh, effectively avoiding such issues. Moreover, GoMAvatar [35] produces artifacts stemming from oversized Gaussians. This is likely because GoMAvatar only aligns Gaussian center on the mesh, but ignores the orientation,
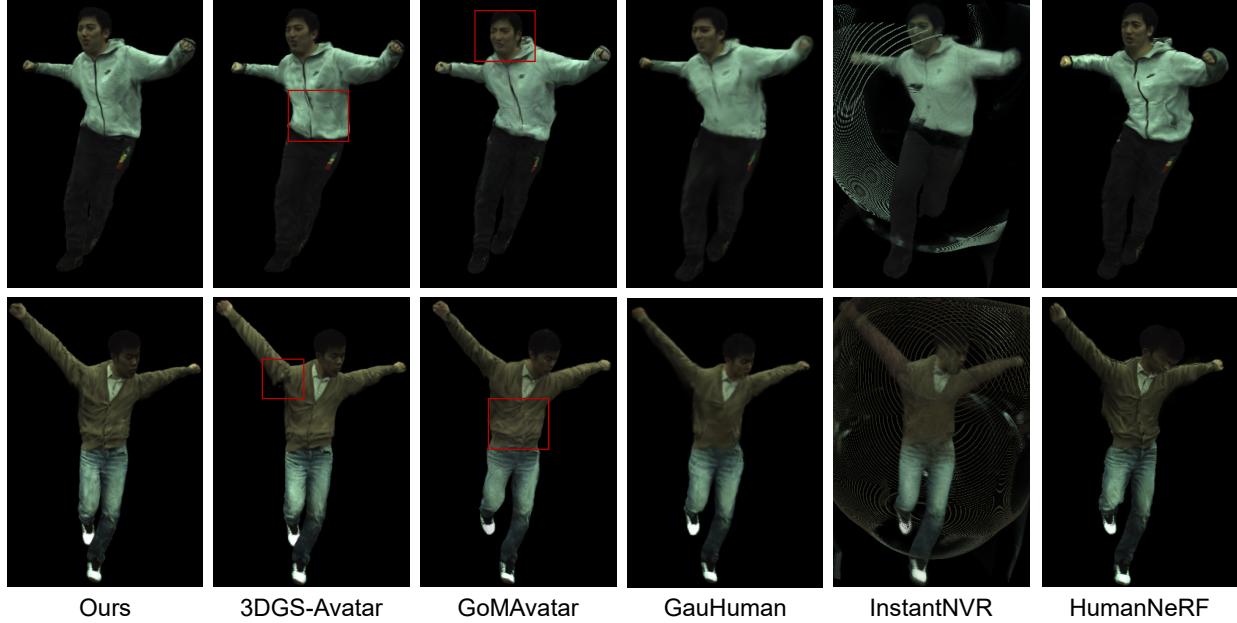
Fig. 9: **Comparison results of animating ZJU-MoCap subjects with out-of-distribution poses from AIST++ dataset [71].** Our method can synthesize more natural results without undesired artifacts.

leading to overfitting to the low-variety A-pose. Consequently, these oversized Gaussians produce artifacts when animated with novel poses. In contrast, the proposed SAGA alleviates this problem by aligning both position and orientation, demonstrating the superior generalization ability in out-of-distribution poses.

6.5 Ablation Study

In this section, we discuss the effect of various strategies proposed in SAGA including the *Two-stage Surface-Aligned Gaussian representation*, the *Gaussian-Mesh Alignment losses*, and the *Walking-on-Mesh strategy*.

6.5.1 Effect of the two-stage Surface-Aligned Gaussians

We evaluate the individual effects of the first *Adhered Stage* and the second *Detached Stage* by removing each of them respectively. The mean results across 6 instances on ZJU-MoCap dataset are reported in Tab. 4. Our full two-stage SAGA representation achieves the best performance. We show the qualitative results in Fig. 11, and analyze them as follows.

Effect of the Adhered Stage. The Adhered Stage constrains the Gaussians to stay strictly on the surface, which we expect to guide the Gaussians to learn well-defined geometry in the early training stage. As shown in Fig. 11(d), without this stage, artifacts such as diverged zippers emerge. This shows that directly optimizing Gaussians cannot ensure the multi-view consistency of such thin structure. For **w/o adhered stage, w/ detached stage** shown in Fig. 11(b), though the Gaussian-Mesh alignment regularization in the Detached Stage can mitigate the issue of diverged zipper, it is not powerful enough to regularize the Gaussians from scratch, compared to the strict alignment of the Adhered Stage .

Effect of the Detached Stage. The Detached Stage further unleashes the fitting potential of the Gaussians from the Adhered Stage by allowing them to move freely near the surface during optimization. Fig. 11 (c) shows that omitting this stage results in blurrier details at the fists and the unnatural face expressions.

Ultimately, our full two-stage SAGA renders sharper images without any artifacts.

Effect on novel pose synthesis. As shown in Fig. 12, our full two-stage representation generates more plausible results, indicating that our SAGA learns canonical Gaussians with improved geometry, leading to better generalization on novel poses.

Stage 1 Adhered	Stage 2 Detached	PSNR↑	SSIM↑	LPIPS↓
×	✓	31.05	0.9725	0.0281
✓	×	31.06	0.9726	0.0298
×	×	30.64	0.9702	0.0290
✓	✓	**31.11**	**0.9728**	**0.0279**

TABLE 4: **Ablation study on the two-stage representaion.** Mean results over all subjects from ZJU-MoCap is reported.

6.5.2 Effect of the Gaussian-Mesh alignment loss

The alignment loss regularizes both canonical and deformed Gaussians to align with the underlining surface, and enforces more consistent non-rigid deformation. As shown in Fig. 13, on the left are the training images from the seen view and the novel view ground-truth images. And on the right are the rendering results. Without the alignment loss, the non-rigid module cannot learn plausible deformation in the area unseen during training, resulting in fractured artifacts (row 1 red box), and misalignment of the buttons (row 2 red box). On the contrary, the proposed alignment loss eliminates the above artifacts, indicating it can effectively regularize the detached Gaussians to learn more natural deformation in the unseen regions, which is also proved by the quantitative results in Tab. 5.

6.5.3 Effect of the Walking-on-Mesh strategy

The Walking-on-Mesh Strategy tracks the nearest triangles for the moving Gaussians to ensure them to be aligned with correct

Fig. 10: **Comparison results of animating PeopleSnapshot subjects with out-of-distribution poses from AIST++ dataset [71].** Our method can synthesize more photorealistic and plausible results, while 3DGS-Avatar [21] and GoMAvatar [35] suffer from artifacts.
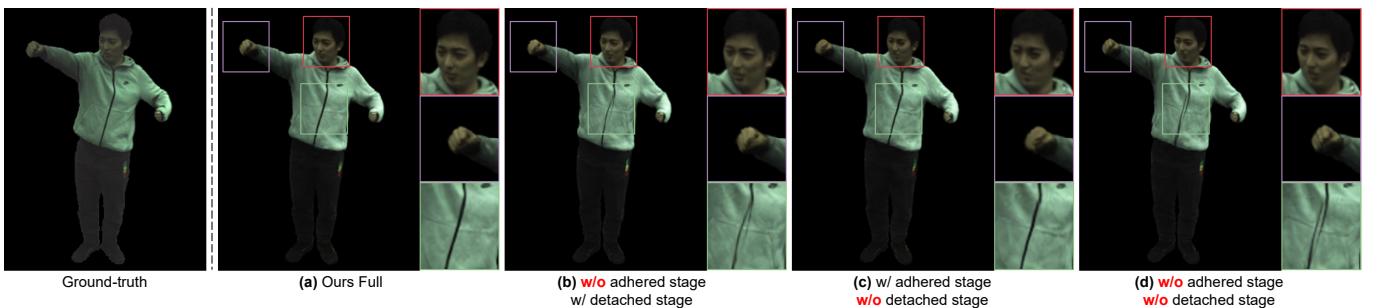

Fig. 11: **Novel view synthesis results of ablation study on the training stages**. *Ours Full* with both the *Adhered* and the *Detached Stage* achieves the most photorealistic results, which avoid artifacts while having sharper details and more natural human expressions.

Fig. 12: **Novel pose synthesis results of ablation study on the training stages**, ours full synthesizes more photorealistic results, while the others suffer from blurry details at the zipper.

Fig. 13: **Qualitative results of ablation study on the proposed Gaussian-mesh Alignment Regularization.** On the left are the ground-truth images from the training view and novel view, respectively. On the right are the rendering results. Ours with the alignment loss produces more plausible results even in the region barely visible during training.

triangles. We compare our strategy with the **UV-based walk** used in SplattingAvatar [58], which is originally developed by lifted optimization techniques [60]–[62]. It walks in the unfolded triangle space based on the UV updates. As shown in Tab. 6, our method achieves the best performance. We think the reason is that our reprojection based method finds more accurate closest triangles in the Euclidean space. In contrast, UV-based methods are proceeded in the unfolded space, which is agnostic to the Euclidean distance. As shown in Fig. 14, this results in artifacts. In contrast, our method can eliminate the artifacts by providing more accurate regularization from more accurate triangles.

Moreover, our carefully optimized implementation (Algorithm 1) runs at a neglectable cost of **1.7** ms over **200x** faster than the UV-based strategy implemented by SplattingAvatar [58].

	PSNR↑	SSIM↑	LPIPS↓
w/o Alignment	31.08	0.9726	**0.0279**
Ours w/ Alignment	**31.11**	**0.9728**	**0.0279**

TABLE 5: **Ablation study on the Alignment Loss.** Mean results over all subjects from ZJU-MoCap is reported.
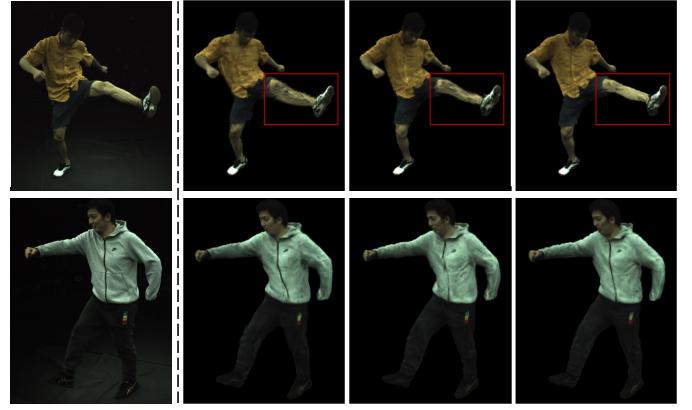

Fig. 14: **Ablation study on the Walking-on-Mesh strategy.** Our method learns more consistent texture at the legs under highly dynamic motion because our Gaussian walking strategy updates the triangles more accurately.

	Time↓	PSNR↑	SSIM↑	LPIPS↓
w/o walk	-	31.07	0.9726	0.0282
uv walk	208ms	31.01	0.9724	0.0292
Ours	**1.7ms**	**31.11**	**0.9728**	**0.0279**

TABLE 6: **Ablation study on the Walking-on-Mesh strategy.** Mean results over all subjects from ZJU-MoCap is reported.

6.5.4 Design of the Adhered-on-Mesh representation

We conduct an ablation study on the design of the Adhered-on-mesh representation applied in the Adhered Stage by comparing it with the **fixed-on-mesh** representation from [35]. It rigidly binds the Gaussian to fixed location on the mesh, and only optimizes the mesh vertices. We evaluate them on Peoplesnapshot dataset.

Our approach outperforms the counterparts on all metrics in Tab. 7. Additionally, as shown in Fig. 15, **ours** can already synthesize photorealistic results after 5k training iterations. In contrast, the **fixed-on-mesh** representation still struggles with distorted eyes and characters on the chest even after 10k iterations. This demonstrates the flexibility and faster convergence of our adhered-on-mesh representation, yielding more realistic results.

	PSNR	SSIM	LPIPS
Fixed-on-Mesh	32.02	0.9787	0.0122
Ours	**32.19**	**0.9789**	**0.0119**

TABLE 7: **Ablation study on the design of the Adhered-on-mesh representation.** Performance is evaluated on the PeopleSnapshot dataset.

6.6 Applications in geometric reconstruction

High-quality geometry reconstruction from 3DGS is still under-explored. Existing works typically focus on the static scenes [28], leaving dynamic human reconstruction from monocular video an unsolved problem. Here, we provide a viable solution, and show-case that existing Gaussian-based human avatar methods [21], [22], though achieving comparable rendering quality, cannot reconstruct high-quality geometry. On the contrary, leveraging the
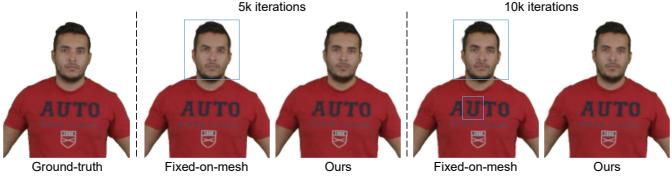
Fig. 15: **Qualitative results of the ablation study on the design of the Adhered-on-mesh representation.** Our flexible representation achieves higher quality in less training time.

mesh as a geometry regularizer, our surface-aligned representation significantly improves the geometric quality.

Visualization of rendered depth. As shown in Fig. 16, naive Gaussian representations [21], [22] cannot learn faithful geometry, especially in the textureless area, *e.g.*, the pants. This validates our assumption that the monocular input cannot guide 3DGS to form well-defined geometry, leading to severe overfitting. Our method, by contrast, produces more plausible depth images.
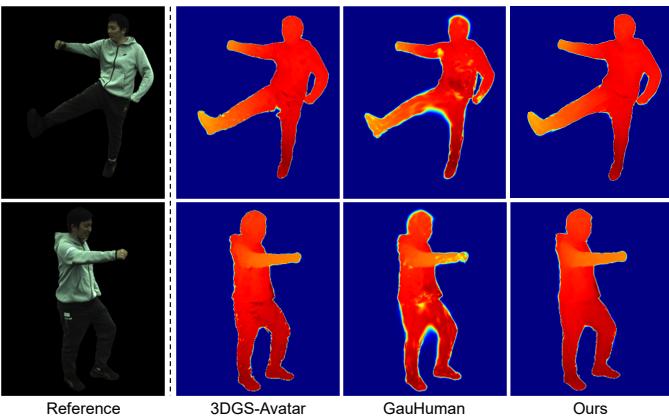

Fig. 16: **Qualitative comparison of depth rendered by naive Gaussian-based methods [21], [22].** While other results suffer from discontinuity and holes, our results are smooth and complete.

Surface reconstruction results. We conduct surface reconstruction on subjects from ZJUMocap and Peoplesnapshot datasets by rendering multi-view depth images and fuse them via TSDF fusion [73]. We sample depth cameras uniformly from a sphere.

As shown in Fig. 17, naive Gaussian based method 3DGS-Avatar [21] reconstructs extremely noisy surface, while our method achieves smoother and more accurate results. This validates our surface-aligned representation's ability to generate high-quality geometry and improve generalization under novel views and poses. Furthermore, it demonstrates the potential of our proposed SAGA in Gaussian-based dynamic surface reconstruction.

7 CONCLUSION

This paper presents a two-stage Surface-aligned Gaussian representation for monocular human reconstruction and rendering. In the first stage, the on-mesh Gaussians are allowed to flow, with their centers and normals aligned with the bound triangles. In the second stage, we unleash the fitting ability of Gaussians by detaching them from the mesh, while maintaining the geometry quality by introducing the Gaussian-Mesh Alignment regularization. Additionally, we propose a Walking-on-Mesh algorithm to
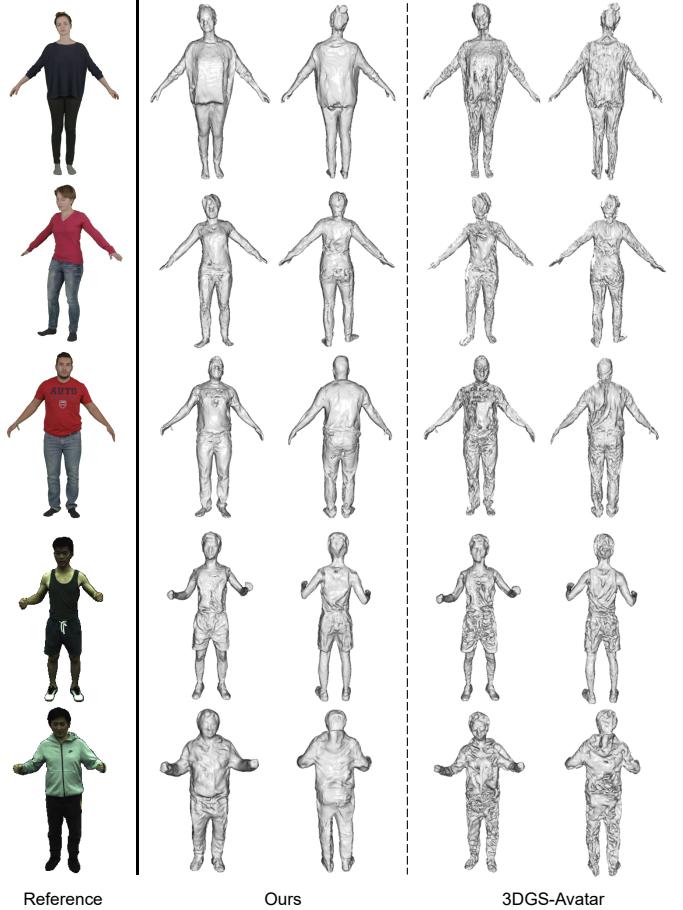

Fig. 17: **Surface reconstruction results.** Our proposed SAGA significantly improves the geometric quality comparing to the SOTA naive Gaussian based method 3DGS-Avatar [21].

accurately track the bound triangle as the Gaussians move during optimization. Our method efficiently fits the scene in ∼12 minutes and renders in real-time at over 60 FPS. Extensive experiments on challenging datasets demonstrate that the proposed surface-aligned Gaussian representation effectively regularizes the Gaussians to generate superior geometry without compromising the fitting capabilities of the original 3DGS. This leads to significantly better performance in novel view and novel pose synthesis compared to the state-of-the-art Gaussian-based methods, while marking the first successful attempt at deformable Gaussian-based mesh extraction from monocular videos.

REFERENCES

[1] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor *et al.*, "Fusion4d: Real-time performance capture of challenging scenes," *ACM Transactions on Graphics (ToG)*, vol. 35, no. 4, pp. 1–13, 2016.

[2] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou *et al.*, "Holoportation: Virtual 3d teleportation in real-time," in *Proceedings of the 29th annual symposium on user interface software and technology*, 2016, pp. 741–754.

[3] K. Guo, P. Lincoln, P. Davidson, J. Busch, X. Yu, M. Whalen, G. Harvey, S. Orts-Escolano, R. Pandey, J. Dourgarian *et al.*, "The relightables: Volumetric performance capture of humans with realistic relighting," *ACM Transactions on Graphics (ToG)*, vol. 38, no. 6, pp. 1–19, 2019.

[4] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan, "High-quality streamable free-viewpoint video," *ACM Transactions on Graphics (ToG)*, vol. 34, no. 4, pp. 1–13, 2015.

[5] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner *et al.*, "State of the art on neural rendering," in *Computer Graphics Forum*, vol. 39, no. 2. Wiley Online Library, 2020, pp. 701–727.

[6] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *European Conference on Computer Vision*, 2020, pp. 405–421.

[7] S. Peng, Y. Zhang, Y. Xu, Q. Wang, Q. Shuai, H. Bao, and X. Zhou, "Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9054–9063.

[8] C.-Y. Weng, B. Curless, P. P. Srinivasan, J. T. Barron, and I. Kemelmacher-Shlizerman, "HumanNeRF: Free-viewpoint rendering of moving people from monocular video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 16 210–16 220.

[9] S. Peng, Z. Xu, J. Dong, Q. Wang, S. Zhang, Q. Shuai, H. Bao, and X. Zhou, "Animatable implicit neural representations for creating realistic avatars from videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[10] L. Liu, M. Habermann, V. Rudnev, K. Sarkar, J. Gu, and C. Theobalt, "Neural actor: Neural free-view synthesis of human actors with pose control," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 6, pp. 1–16, 2021.

[11] W. Jiang, K. M. Yi, G. Samei, O. Tuzel, and A. Ranjan, "Neuman: Neural human radiance field from a single video," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII.* Springer, 2022, pp. 402–418.

[12] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530127

[13] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *European Conference on Computer Vision (ECCV)*, 2022.

[14] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5459–5469.

[15] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5501–5510.

[16] P. Hedman, P. P. Srinivasan, B. Mildenhall, J. T. Barron, and P. Debevec, "Baking neural radiance fields for real-time view synthesis," *ICCV*, 2021.

[17] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "PlenOctrees for real-time rendering of neural radiance fields," in *ICCV*, 2021.

[18] C. Geng, S. Peng, Z. Xu, H. Bao, and X. Zhou, "Learning neural volumetric representations of dynamic humans in minutes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8759–8770.

[19] T. Jiang, X. Chen, J. Song, and O. Hilliges, "Instantavatar: Learning avatars from monocular video in 60 seconds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 922–16 932.

[20] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics (ToG)*, vol. 42, no. 4, pp. 1–14, 2023.

[21] Z. Qian, S. Wang, M. Mihajlovic, A. Geiger, and S. Tang, "3dgs-avatar: Animatable avatars via deformable 3d gaussian splatting," *arXiv preprint arXiv:2312.09228*, 2023.

[22] S. Hu, T. Hu, and Z. Liu, "Gauhuman: Articulated gaussian splatting from monocular human videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 418–20 431.

[23] W. Zielonka, T. Bagautdinov, S. Saito, M. Zollhöfer, J. Thies, and J. Romero, "Drivable 3d gaussian avatars," 2023.

[24] M. Kocabas, R. Chang, J. Gabriel, O. Tuzel, and A. Ranjan, "Hugs: Human gaussian splats," 2023. [Online]. Available: https://arxiv.org/abs/2311.17910

[25] J. Lei, Y. Wang, G. Pavlakos, L. Liu, and K. Daniilidis, "Gart: Gaussian articulated template models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 876–19 887.

[26] S. Zheng, B. Zhou, R. Shao, B. Liu, S. Zhang, L. Nie, and Y. Liu, "Gps-gaussian: Generalizable pixel-wise 3d gaussian splatting for real-time human novel view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 680–19 690.

[27] Z. Li, Z. Zheng, L. Wang, and Y. Liu, "Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 711–19 722.

[28] A. Guédon and V. Lepetit, "Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering," *arXiv preprint arXiv:2311.12775*, 2023.

[29] T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, and B. Dai, "Scaffold-gs: Structured 3d gaussians for view-adaptive rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 20 654–20 664.

[30] K. Cheng, X. Long, K. Yang, Y. Yao, W. Yin, Y. Ma, W. Wang, and X. Chen, "Gaussianpro: 3d gaussian splatting with progressive propagation," in *Forty-first International Conference on Machine Learning*, 2024.

[31] M. Habermann, W. Xu, M. Zollhofer, G. Pons-Moll, and C. Theobalt, "Deepcap: Monocular human performance capture using weak supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5052–5063.

[32] S. Saito, J. Yang, Q. Ma, and M. J. Black, "Scanimate: Weakly supervised learning of skinned clothed avatar networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2886–2897.

[33] M. Habermann, L. Liu, W. Xu, M. Zollhoefer, G. Pons-Moll, and C. Theobalt, "Real-time deep dynamic characters," *ACM Transactions on Graphics (ToG)*, vol. 40, no. 4, pp. 1–16, 2021.

[34] Z. Huang, Y. Xu, C. Lassner, H. Li, and T. Tung, "Arch: Animatable reconstruction of clothed humans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[35] J. Wen, X. Zhao, Z. Ren, A. G. Schwing, and S. Wang, "Gomavatar: Efficient animatable human modeling from monocular video using gaussians-on-mesh," *arXiv preprint arXiv:2404.07991*, 2024.

[36] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *ACM transactions on graphics (TOG)*, vol. 34, no. 6, pp. 1–16, 2015.

[37] S. Peng, J. Dong, Q. Wang, S. Zhang, Q. Shuai, X. Zhou, and H. Bao, "Animatable neural radiance fields for modeling dynamic human bodies," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 314–14 323.

[38] A. Noguchi, X. Sun, S. Lin, and T. Harada, "Neural articulated radiance field," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5762–5772.

[39] S.-Y. Su, F. Yu, M. Zollhöfer, and H. Rhodin, "A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 278–12 291, 2021.

[40] T. Xu, Y. Fujita, and E. Matsumoto, "Surface-aligned neural radiance fields for controllable 3d human synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 883–15 892.

[41] Y. Kwon, D. Kim, D. Ceylan, and H. Fuchs, "Neural human performer: Learning generalizable radiance fields for human performance rendering," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[42] X. Gao, J. Yang, J. Kim, S. Peng, Z. Liu, and X. Tong, "Mps-nerf: Generalizable 3d human rendering from multiview images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–12, 2022.

[43] R. Shao, H. Zhang, H. Zhang, M. Chen, Y.-P. Cao, T. Yu, and Y. Liu, "Doublefield: Bridging the neural surface and radiance fields for high-fidelity human reconstruction and rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 15 872–15 882.

[44] Y. Kwon, L. Liu, H. Fuchs, M. Habermann, and C. Theobalt, "Deliffas: Deformable light fields for fast avatar synthesis," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[45] E. Remelli, T. Bagautdinov, S. Saito, C. Wu, T. Simon, S.-E. Wei, K. Guo, Z. Cao, F. Prada, J. Saragih *et al.*, "Drivable volumetric avatars using texel-aligned features," in *ACM SIGGRAPH 2022 Conference Proceedings*, 2022, pp. 1–9.

[46] Z. Zheng, X. Zhao, H. Zhang, B. Liu, and Y. Liu, "Avatarrex: Real-time expressive full-body avatars," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, pp. 1–19, 2023.

[47] J. Luiten, G. Kopanas, B. Leibe, and D. Ramanan, "Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis," *arXiv preprint arXiv:2308.09713*, 2023.

[48] Y. Duan, F. Wei, Q. Dai, Y. He, W. Chen, and B. Chen, "4d gaussian splatting: Towards efficient novel view synthesis for dynamic scenes," *arXiv preprint arXiv:2402.03307*, 2024.

[49] D. Das, C. Wewer, R. Yunus, E. Ilg, and J. E. Lenssen, "Neural parametric gaussians for monocular non-rigid object reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 715–10 725.

[50] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin, "Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 331–20 341.

[51] Y. Jiang, Z. Shen, P. Wang, Z. Su, Y. Hong, Y. Zhang, J. Yu, and L. Xu, "Hifi4g: High-fidelity human performance rendering via compact gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 734–19 745.

[52] L. Hu, H. Zhang, Y. Zhang, B. Zhou, B. Liu, S. Zhang, and L. Nie, "Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 634–644.

[53] R. Jena, G. S. Iyer, S. Choudhary, B. Smith, P. Chaudhari, and J. Gee, "Splatarmor: Articulated gaussian splatting for animatable humans from monocular rgb videos," *arXiv preprint arXiv:2311.10812*, 2023.

[54] Y. Jiang, Q. Liao, X. Li, L. Ma, Q. Zhang, C. Zhang, Z. Lu, and Y. Shan, "Uv gaussians: Joint learning of mesh deformation and gaussian textures for human avatar modeling," *arXiv preprint arXiv:2403.11589*, 2024.

[55] H. Pang, H. Zhu, A. Kortylewski, C. Theobalt, and M. Habermann, "Ash: Animatable gaussian splats for efficient and photoreal human rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 1165–1175.

[56] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[57] S. Prokudin, Q. Ma, M. Raafat, J. Valentin, and S. Tang, "Dynamic point fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2023, pp. 7964–7976.

[58] Z. Shao, Z. Wang, Z. Li, D. Wang, X. Lin, Y. Zhang, M. Fan, and Z. Wang, "Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 1606–1616.

[59] D. Svitov, P. Morerio, L. Agapito, and A. Del Bue, "Haha: Highly articulated gaussian human avatars with textured mesh prior," *arXiv preprint arXiv:2404.01053*, 2024.

[60] J. Shen, T. J. Cashman, Q. Ye, T. Hutton, T. Sharp, F. Bogo, A. Fitzgibbon, and J. Shotton, "The phong surface: Efficient 3d model fitting using lifted optimization," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 687–703.

[61] J. Taylor, R. Stebbing, V. Ramakrishna, C. Keskin, J. Shotton, S. Izadi, A. Hertzmann, and A. Fitzgibbon, "User-specific hand modeling from monocular depth sequences," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 644–651.

[62] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff *et al.*, "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," *ACM Transactions on Graphics (ToG)*, vol. 35, no. 4, pp. 1–12, 2016.

[63] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross, "Ewa splatting," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 3, pp. 223–238, 2002.

[64] L. Gao, J. Yang, B.-T. Zhang, J.-M. Sun, Y.-J. Yuan, H. Fu, and Y.-K. Lai, "Mesh-based gaussian splatting for real-time large-scale deformation," *arXiv preprint arXiv:2402.04796*, 2024.

[65] Z. Yang, X. Gao, W. Zhou, S. Jiao, Y. Zhang, and X. Jin, "Deformable 3d gaussians for high-fidelity monocular dynamic scene reconstruction," *arXiv preprint arXiv:2309.13101*, 2023.

[66] M. Mihajlovic, Y. Zhang, M. J. Black, and S. Tang, "Leap: Learning articulated occupancy of people," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 461–10 471.

[67] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.

[68] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[69] T. Alldieck, M. Magnor, W. Xu, C. Theobalt, and G. Pons-Moll, "Video based reconstruction of 3d people models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2018, pp. 8387–8397, CVPR Spotlight Paper.

[70] J. Chen, Y. Zhang, D. Kang, X. Zhe, L. Bao, X. Jia, and H. Lu, "Animatable neural radiance fields from monocular rgb videos," *arXiv preprint arXiv:2106.13629*, 2021.

[71] R. Li, S. Yang, D. A. Ross, and A. Kanazawa, "Ai choreographer: Music conditioned 3d dance generation with aist++," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 401–13 412.

[72] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black, "Amass: Archive of motion capture as surface shapes," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5442–5451.

[73] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 303–312.

8 BIOGRAPHY SECTION

Ronghan Chen Ronghan Chen is currently a Ph.D candidate in State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences. His current research interests include 3D reconstruction and neural rendering.

Yang Cong (S'09-M'11-SM'15) is a full professor of the College of Automation Science and Engineering, South China University of Technology. He received the he B.Sc. de. degree from Northeast University in 2004, and the Ph.D. degree from State Key Laboratory of Robotics, Chinese Academy of Sciences in 2009. He was a Research Fellow of National University of Singapore (NUS) and Nanyang Technological University (NTU) from 2009 to 2011, respectively; and a visiting scholar of University of Rochester. He has served on the editorial board of the Journal of Multimedia. His current research interests include image processing, compute vision, machine learning, multimedia, medical imaging, data mining and robot navigation. He has authored over 70 technical papers. He is also a senior member of IEEE.

Jiayue Liu Jiayue Liu is a Ph.D candidate in the School of Automation Science and Engineering, South China University of Technology. Her current research interests include 3D reconstruction.