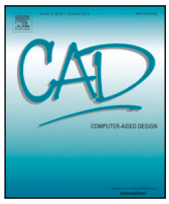
Computer-Aided Design 190 (2026) 103973

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad



LoGAvatar: Local Gaussian Splatting for human avatar modeling from monocular video☆

Jinsong Zhang [a,b], Xiongzheng Li [a], Hailong Jia [a], Jin Li [a], Zhuo Su [c], Guidong Wang [c], Kun Li [a,*]

[a] College of Intelligence and Computing, Tianjin University, Tianjin, China
[b] College of Computer and Data Science, Fuzhou University, Fuzhou, China
[c] ByteDance, Shanghai, China

ARTICLE INFO

Keywords:
Human avatar reconstruction
Gaussian splatting
Avatar editing
Monocular video

ABSTRACT

Avatar reconstruction from monocular videos plays a pivotal role in various virtual and augmented reality applications. Recent methods have utilized 3D Gaussian Splatting (GS) to model human avatars, achieving fast rendering speeds with high visual quality. However, due to the independent nature of GS primitives, existing approaches often struggle to capture high-fidelity details and lack the ability to edit the reconstructed avatars effectively. To address these limitations, we propose Local Gaussian Splatting Avatar (LoGAvatar), a novel framework designed to enhance both geometry and texture modeling of human avatars. Specifically, we introduce a hierarchical Gaussian splatting framework, where local GS primitives are predicted based on sampled points from a human template model, such as SMPL. For texture modeling, we design a convolution-based texture atlas that preserves spatial continuity and enriches fine details. By aggregating local information for both geometry and texture, our approach reconstructs high-fidelity avatars while maintaining real-time rendering efficiency. Experimental results on two public datasets demonstrate the superior performance of our method in terms of avatar fidelity and rendering quality. Moreover, based on our LoGAvatar, we can edit the shape and texture of the reconstructed avatar, which inspires more customized avatar applications. The code is available at http://cic.tju.edu.cn/faculty/likun/projects/LoGAvatar.

1. Introduction

Human avatar reconstruction has emerged as a foundational technology in immersive telepresence [1–3], virtual try-on [4–7], and entertainment [8–10]. While existing methods achieve high-quality reconstructions using multi-view camera systems [11–13], their high cost and complexity make them impractical for consumer-level applications. In this work, we address a more accessible yet challenging task: *photorealistic human avatar reconstruction from monocular videos*.

Neural Radiance Fields (NeRFs) [14] have demonstrated remarkable results in novel view synthesis and pose-driven avatar generation from monocular videos. However, NeRF-based methods require querying densities and colors at numerous spatial locations, resulting in slow volume rendering [15–17]. Despite acceleration techniques such as plenoptic voxels [18] and hash encoding [19], the computational overhead remains a significant barrier to real-time deployment.

The advent of 3D Gaussian Splatting (3DGS) [20] has enabled high-quality, real-time rendering, sparking interest in its adoption for human avatar reconstruction from monocular videos. Existing methods [21–25] leverage 3DGS as a canonical representation while incorporating parametric human models, such as SMPL [26] or SMPL-X [27], for deformation priors through linear blend skinning. Although these approaches achieve impressive real-time rendering, they optimize individual Gaussian primitives independently, neglecting local correspondences in the 3D human body. This limitation leads to blurry textures and the loss of fine-grained details in reconstructed avatars. Besides, it is also difficult to edit the shape and texture of the reconstructed avatar.

Prior works have struggled to model high-fidelity avatars from monocular videos and to edit reconstructed avatars due to the lack of local information inherent in the discrete properties of Gaussian avatars. To address these challenges, we introduce LoGAvatar, a novel framework that integrates local geometric and textural information into 3DGS for high-fidelity human avatar modeling. Our key insights are twofold: (1) Human deformations exhibit local coherence governed by musculoskeletal structures, suggesting that each Gaussian should be influenced by its neighboring Gaussians. (2) Texture appearance follows spatial continuity, which should be preserved across

☆ This article is part of a Special issue entitled: 'SPM-25' published in Computer-Aided Design.

* Corresponding author.
E-mail addresses: jinszhang@tju.edu.cn (J. Zhang), lik@tju.edu.cn (K. Li).

https://doi.org/10.1016/j.cad.2025.103973
Received 24 June 2025; Received in revised form 9 August 2025; Accepted 13 September 2025
Available online 29 September 2025

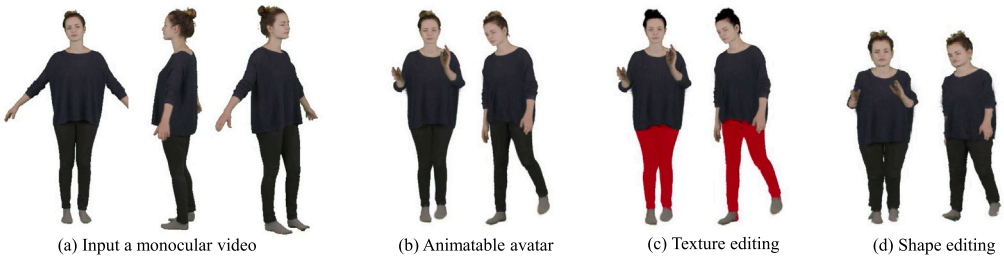


Fig. 1. Given a monocular video (a), our LoGAvatar can reconstruct an animatable avatar (b), which enables texture editing (c) and shape editing (d) applications.

adjacent Gaussian primitives. However, aggregating local information for point-wise GS representation is non-trivial. To enforce these principles, we propose a hierarchical Gaussian prediction scheme that anchors Gaussians to sampled SMPL surface points, ensuring deformation consistency through learnable local coordinate transformations. Additionally, we introduce a convolutional texture atlas that encodes appearance features in a coordinate-aligned canonical space, facilitating detail-preserving texture transfer. This explicit design also enables texture editing of reconstructed avatars, a feature highly beneficial for many avatar applications.

Extensive experiments on the widely used People-Snapshot [28] and ZJU-MoCap datasets [29] demonstrate that our method achieves state-of-the-art performance, surpassing previous GS-based approaches while maintaining real-time rendering speeds. Ablation studies confirm that our local Gaussian binding reduces positional drift in extreme poses, resulting in more stable and visually consistent avatars. Fig. 1 shows some results and applications of LoGAvatar.

Our main contributions can be summarized as follows:

- We propose a local Gaussian splatting framework that enhances high-fidelity avatar reconstruction and editing from monocular video.
- We develop a local geometry prediction module that derives geometric attributes for each Gaussian from the vertices of a human parametric model, which supports shape editing and improves avatar reconstruction quality.
- We introduce a convolution-based texture atlas that preserves spatial consistency and enhances appearance modeling, yielding detailed and editable avatars.
- We conduct extensive experiments demonstrating superior rendering quality and editing applications of the proposed method.

The structure of this paper is organized as follows: In Section 2, we review related works on human avatar reconstruction, covering various representations such as meshes, neural radiance fields, and point-based methods. Section 3 provides preliminaries on the human parametric model used in our work and introduces the fundamentals of 3D Gaussian splatting. In Section 4, we present the proposed LoGAvatar framework in detail, including its loss functions and training strategy. Section 5 describes the experimental setup, including datasets, baseline methods, and evaluation metrics. We then analyze the comparative results to demonstrate the effectiveness of our approach and showcase several applications, such as avatar animation, texture editing, and shape editing. Finally, we conclude our work and discuss future directions in Section 6.

2. Related work

Human avatar reconstruction has been extensively studied using various 3D representations. In this section, we categorize existing approaches into three main groups: mesh-based methods, neural radiance field-based methods, and 3D Gaussian splatting-based methods.

2.1. Meshes

Mesh-based methods [28,30] utilize parametric human models such as Skinned Multi-Person Linear (SMPL) [26] and Skinned Multi-Person Linear eXpressive (SMPL-X) [27] to estimate body shape and pose from images or videos. These models provide explicit surface representations, enabling precise geometry reconstruction. Traditional approaches rely on optimization techniques [31,32] to fit the parametric model to image observations, while learning-based methods [33–36] improve reconstruction accuracy by leveraging deep neural networks. However, these methods focus solely on body pose estimation while ignoring texture information, making them unsuitable for rendering realistic human images. To address this limitation, some methods incorporate SMPL-based texture maps to jointly model geometry and appearance for avatar reconstruction.

Alldieck et al. [28] proposed an optimization framework using a visual hull approach to refine SMPL geometry from monocular videos, enabling the creation of personalized blend shape models. Zhao et al. [37] introduced a dynamic surface network to reconstruct pose-dependent geometry and coarse textures, which were subsequently refined using a reference-based neural rendering network for enhanced details. To reduce the number of required input images, Alldieck et al. [38] estimated geometry with vertex displacements directly from monocular images and applied a graph-cut optimization technique over eight frames to construct texture maps. While mesh-based methods provide structured surface representations, they often struggle to capture high-frequency details, leading to artifacts and blurry results.

2.2. Neural Radiance Fields

Neural Radiance Fields (NeRF) [14] have revolutionized novel view synthesis by learning a continuous volumetric representation from images, effectively capturing fine details and complex lighting effects. Due to their ability to model high-frequency textures and global scene properties, NeRF-based approaches have been widely adopted for human avatar reconstruction, particularly for photorealistic novel view generation and pose-dependent appearance synthesis.

Early NeRF-based human modeling primarily focused on static representations. However, modeling dynamic humans presents challenges related to temporal consistency and pose generalization. To address these issues, Neural Body [29] introduced an explicit deformation field that aligns NeRF with the SMPL body model, improving temporal coherence in motion sequences [17]. HumanNeRF [39] conditioned NeRF on articulated human poses, enabling more realistic pose-dependent rendering. Similarly, AnimatableNeRF [40] incorporated a deformation network to model pose-dependent shape and appearance variations. To further improve pose generalization, AniNeRF [41] employed skeleton-driven deformation fields to better capture human motion dynamics. ARAH [42] introduced an articulated neural field that disentangles shape and pose information, facilitating high-quality synthesis of unseen poses. Liu et al. [43] adopt local coordinate for each query point based on SMPL body model to learn better dynamic geometry from multi-camera inputs. Despite these advances, Gao et al. [44] propose

to learn a generalizable avatar representation from multi-view inputs to predict the results of unseen identities. However, some of these works [43,44] rely on multi-camera inputs, which is different from our monocular video setup. Besides, NeRF-based approaches suffer from significant computational bottlenecks. The need to query densities and colors at numerous spatial locations along each ray results in slow inference speeds, making real-time applications impractical. To mitigate this, various acceleration techniques have been proposed. Plenoptic voxels [18] and sparse voxel grids [45] reduce the number of sampled points, while hash encoding [19] drastically improves efficiency by compactly storing NeRF features in a learnable hash table. Although these techniques enhance rendering speeds, they still fall short of real-time performance, particularly for applications requiring interactive avatar control.

In summary, NeRF-based approaches deliver exceptional rendering quality but remain constrained by slow inference speeds and high computational costs. These limitations have driven the exploration of alternative representations, such as 3D Gaussian Splatting, which achieves high-quality rendering with significantly improved efficiency.

2.3. 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) [20] has recently emerged as an efficient alternative to traditional volumetric representations, enabling real-time rendering with high visual fidelity. Unlike Neural Radiance Fields (NeRF) [14], which rely on volumetric ray sampling and require costly Monte Carlo integration, 3DGS represents a scene as a set of anisotropic 3D Gaussians that are directly projected and rasterized into 2D space. This formulation facilitates efficient forward rendering, significantly accelerating novel view synthesis.

Several methods integrate 3DGS with human parametric models to achieve efficient and high-quality rendering. For instance, MonoGaussianAvatar [46] and SplattingAvatar [47] leverage SMPL priors to guide Gaussian placements and transformations, improving temporal stability. However, these approaches lack local information for each Gaussian, resulting in discontinuities in the rendered outputs. Further advancements in 3DGS explore hierarchical structures and hybrid representations to enhance rendering quality. For example, Dongye et al. [48] introduced a hierarchical Gaussian structure to enable adaptive levels of detail, though the primary focus is efficiency rather than rendering fidelity. Hybrid methods such as GoMAvatar [24] propose a Gaussian-on-Mesh representation, binding each Gaussian to the faces of a mesh to improve animation robustness. Similarly, iHuman [49] and ExAvatar [50] attach Gaussians to the faces of SMPL/SMPL-X body meshes, yielding promising results. However, as each Gaussian's properties are learned independently without considering local spatial correlations, these methods still suffer from suboptimal reconstruction quality.

Despite these advancements, existing 3DGS-based methods treat each Gaussian independently, neglecting spatial correlations that could improve geometric coherence and texture fidelity. Consequently, artifacts such as blurry textures, ghosting effects, and loss of fine details remain significant challenges. To address this, Animatable Gaussian [12], LayGA [51], and PhysAvatar [13] employ convolutional neural networks to aggregate local information, enhancing rendering fidelity. However, these methods require explicit geometry preprocessing to extract query Gaussians from Gaussian maps, introducing additional computational overhead. Besides, SC-GS [52] proposes a hierarchical approach to obtain a compact transformation by interpolating attributes from some anchor Gaussians, which can introduce local information. However, it fails to achieve shape editing or texture editing.

In this paper, we build upon the advantages of 3DGS and propose an improved method that enhances both the texture and geometry representation of human avatars, which can be used to edit the reconstructed avatars.

3. Preliminaries

In this section, we first introduce the human parametric model, *i.e.*, the Skinned Multi-Person Linear (SMPL) model (3.1), and then provide a brief introduction to 3D Gaussian Splatting (GS) (3.2).

3.1. Skinned Multi-Person Linear model

The Skinned Multi-Person Linear (SMPL) model [26] is a skinned, vertex-based human body model initially designed to accurately represent a wide range of body shapes and natural human poses. Based on a mesh representation, SMPL consists of $N = 6890$ vertices and $K = 23$ joints. The model can be formulated as a function that maps a shape parameter $\vec{\beta}$ and a pose parameter $\vec{\theta}$ to a mesh with N vertices.

Starting from a template shape $\overline{\mathbf{T}} \in \mathbb{R}^{3N}$, which represents a canonical human body in a predefined rest pose $\vec{\theta}^*$, SMPL deforms the shape using both shape-dependent and pose-dependent blend shapes.

Shape blend shapes. The shape-dependent deformation is represented by a linear blend shape function B_S, which models variations in body shape using a set of principal components:

$$B_S(\vec{\beta}) = \sum_{n=1}^{10} \beta_n \mathbf{S}_n, \tag{1}$$

where $\vec{\beta} = [\beta_1, \ldots, \beta_{10}]^T$ is a 10-dimensional shape coefficient vector, and $\mathbf{S}_n \in \mathbb{R}^{3N}$ are orthonormal principal components of shape displacements.

Pose blend shapes. To account for pose-dependent deformations, SMPL employs a pose blend shape function B_P:

$$B_P(\vec{\theta}) = \sum_{n=1}^{K} (R_n(\vec{\theta}) - R_n(\vec{\theta}^*))\mathbf{P}_n, \tag{2}$$

where $R(\vec{\theta})$ maps the pose vector $\vec{\theta}$ to a vector of concatenated part-relative rotation matrices, and $\mathbf{P}n \in \mathbb{R}^{3N}$ represents vertex displacements induced by pose changes.

Using the template shape $\overline{\mathbf{T}}$ and the two deformation functions, the final posed body shape T_P can be expressed as:

$$T_P(\vec{\beta}, \vec{\theta}) = \overline{\mathbf{T}} + B_S(\vec{\beta}) + B_P(\vec{\theta}). \tag{3}$$

Joint regression. The body joints are obtained as a function of the shape parameters $\vec{\beta}$:

$$J(\vec{\beta}) = \mathcal{J}(\overline{\mathbf{T}} + B_S(\vec{\beta})), \tag{4}$$

where $\mathcal{J}$ is a precomputed regression matrix that maps the rest-pose vertices to corresponding joint locations.

Linear blend skinning (LBS). The final deformed mesh $M(\vec{\beta}, \vec{\theta})$ is obtained using a linear blend skinning function W:

$$M(\vec{\beta}, \vec{\theta}) = W\left(T_P(\vec{\beta}, \vec{\theta}), J(\vec{\beta}), \vec{\theta}, \mathcal{W}\right), \tag{5}$$

where $\mathcal{W} \in \mathbb{R}^{N \times K}$ represents the skinning weights that define how each vertex is influenced by the skeletal joints.

3.2. 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) [20] represents a 3D scene as an explicit radiance field composed of a set of learnable 3D Gaussians. It combines the advantages of neural implicit fields and point-based rendering methods, achieving the high-fidelity rendering quality of the former while maintaining the real-time rendering capability of the latter.

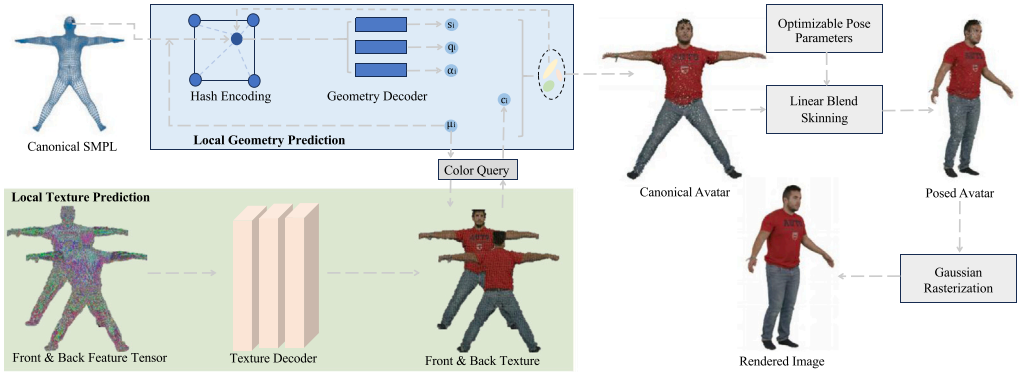


Fig. 2. The overall framework of our LoGAvatar. We predict the geometric attributes from the surface points of the canonical SMPL model to obtain local geometric information. Then, we query the color from the predicted front and back texture Gaussian map predicted from learnable tensors using convolutional neural network. Afterwards, we animate the canonical avatar using pose parameters and render the final image.

Gaussian representation. In 3DGS, a 3D point is represented as an anisotropic 3D Gaussian ellipsoid, which is defined as:

$$G(x) = \exp\left(-\frac{1}{2}x^T \Sigma^{-1} x\right), \tag{6}$$

where Σ is the 3D covariance matrix defined in world space, and x represents the position relative to the Gaussian mean μ.

To ensure that Σ remains a positive semi-definite matrix with physical meaning, it is reparameterized using a rotation matrix R and a scaling matrix S:

$$\Sigma = RSS^T R^T. \tag{7}$$

Here, S is a diagonal scaling matrix that can be parameterized by a 3D vector s, while R is derived from a learnable quaternion q, ensuring a valid rotation.

In addition to geometric parameters, each Gaussian is associated with an opacity value σ and a set of learnable Spherical Harmonics (SH) coefficients sh to model view-dependent appearance. Consequently, a scene is parameterized as a set of Gaussians:

$$\mathcal{G} = \{G_i : \mu_i, s_i, q_i, \sigma_i, sh_i\}_{i=1}^{\mathcal{N}}. \tag{8}$$

Rendering process. To render an image from a given viewpoint, the covariance matrix Σ' in the camera coordinate system is obtained by projecting Gaussians from 3D space onto the 2D image plane:

$$\Sigma' = JW\Sigma W^T J^T, \tag{9}$$

where W represents the viewing transformation, and J is the Jacobian matrix of the affine approximation of the projective transformation.

Finally, the pixel color C for a given image location is computed by alpha compositing over the set of ordered Gaussians that overlap the pixel:

$$C = \sum_{i \in \mathcal{N}} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j), \tag{10}$$

where c_i denotes the color of the ith Gaussian, and α_i is computed by multiplying the projected covariance Σ' with the corresponding learned per-point opacity σ_i.

4. Methods

Given a monocular video capturing a clothed human in motion, our objective is to reconstruct a high-fidelity, animatable human avatar. Recent approaches leveraging 3D Gaussian Splatting (3DGS) have demonstrated promising results in monocular avatar reconstruction. However, these methods often neglect the intrinsic local connectivity in human geometry and appearance, resulting in suboptimal detail preservation. To address this limitation, we propose LoGAvatar (Local Gaussian Avatar), a novel framework that enhances local detail reconstruction by aggregating information from neighboring Gaussians. Our approach introduces local geometry and texture prediction mechanisms to improve spatial coherence and appearance fidelity.

In the following sections, we first define the representation of LoGAvatar in canonical space (4.1), then describe its rendering process (4.2). Finally, we detail the loss functions (4.3) and training procedures (4.4) of our model.

4.1. Canonical representation

To represent human avatars using 3D Gaussian Splatting (3DGS), we construct a canonical avatar as a set of Gaussians $G_i = \{\mu_i, s_i, q_i, \alpha_i, c_i\}_{i=1}^{N}$, where $\mu_i \in \mathbb{R}^3$ represents the position, $s_i \in \mathbb{R}^3$ denotes the scale, $q_i \in \mathbb{S}^3$ is the quaternion rotation, $\alpha_i \in [0, 1]$ corresponds to opacity, and $c_i \in R^3$ encodes the color parameters, *i.e.*, Spherical Harmonics (SH) coefficients, of each Gaussian. As illustrated in Fig. 2, the proposed LoGAvatar framework incorporates two key components: local geometry prediction and local texture prediction, which together ensure both spatial coherence and high-fidelity appearance modeling.

4.1.1. Local geometry prediction

To better capture local geometric correlations, a hierarchical prediction module is designed to estimate the geometric attributes of Gaussian primitives. The process begins by upsampling the canonical SMPL model [26] and sampling L anchor vertices from the mesh surface, forming a set of geometrically meaningful initialization points denoted as $V_a = \{v_a^i\}_{i=1}^{L}$. The anchor vertices are initialized through a two-step process. First, we subdivide the SMPL body mesh to create a dense surface representation. Then, we apply farthest point sampling (FPS) algorithm to select 10,000 geometrically meaningful anchor points that optimally cover the mesh surface while maintaining uniform spatial distribution. This number was determined empirically to provide sufficient surface coverage while remaining computationally efficient. The FPS algorithm ensures that the selected anchors capture key geometric features of the body shape. For each anchor v_a^i, a trainable displacement tensor $D_i \in \mathbb{R}^{M \times 3}$ is introduced to learn position offsets, where the final Gaussian positions are computed as

$$\mu_i = v_a^i + d_i^m, \tag{11}$$

with $d_i^m \in D_i$ being the mth displacement corresponding to the ith anchor, which is used only in prediction pipeline, and not used in rendering. Rather than predicting individual Gaussian positions independently, the model learns M position offsets for each anchor, thereby generating a group of locally coherent Gaussians.

Beyond position estimation, the model predicts other geometric attributes by leveraging a multiresolution hash encoding [53], which extracts geometric features f_i from the base Gaussian point G_i. The input of the multiresolution hash encoding is the anchor point, and the channel dimension of f_i is 16. These features are then fed into a geometry decoder that infers scale, rotation, and opacity. The decoding process employs three parallel branches, each transforming the encoded geometric feature through a sequence of linear layers with ReLU activations. Taking the scale prediction as an example, the output is computed as

$$s_i = W_3\sigma(W_2\sigma(W_1 f_i + \mathbf{b}_1) + \mathbf{b}_2) + \mathbf{b}_3, \tag{12}$$

where W_k and $\mathbf{b}_k$ denote the weights and biases of the kth linear layer, while $\sigma(\cdot)$ represents the ReLU activation function. By employing independent branches for different geometric attributes, the model ensures that feature interference is minimized, allowing each attribute to be learned in a manner that respects its inherent constraints.

The local geometry prediction process is hierarchically structured, meaning that position estimates generated at an earlier stage serve as inputs for subsequent stages. At each stage, we refine the model's predictions, improving the accuracy of the geometric attributes. In particular, the position refinement at stage t can be expressed as

$$\mu_i^{t+1} = \mu_i^t + \delta_i^t, \tag{13}$$

where μ_i^t is the position of the ith Gaussian at stage t, and δ_i^t represents the predicted offset at stage t. In the next stage, we take the position of generated Gaussian μ_i^t as the input of the local geometry prediction module to predict anther group of Gauusians. Note that all the predicted Gaussians are used for rendering process.

Rather than treating Gaussian estimation as an isolated process, the proposed approach enforces local geometric consistency by associating Gaussians originating from the same anchor point while maintaining flexibility through per-Gaussian displacements. Each anchor generates multiple Gaussians (*e.g.*, $M = 4$) via parameter expansion, with the decoder producing M parallel predictions for each geometric attribute. Besides, the hierarchical nature of this prediction process allows for iterative refinement, where positions estimated at an earlier stage serve as inputs for subsequent levels. This progressive modeling strategy enables a coarse-to-fine reconstruction of the avatar, capturing intricate geometric details through multiple stages of prediction.

4.1.2. Local texture prediction

Building upon the local geometry prediction module, the geometric attributes of the Gaussians can effectively incorporate local information. However, directly applying a similar approach to texture estimation presents challenges, as texture attributes are not inherently linked to the anchor points in the SMPL model. While geometric attributes can be easily modified through the SMPL shape parameters, texture attributes require a more flexible representation. To address this, we introduce a dedicated local texture prediction module that aggregates local texture information and improves the flexibility of texture editing.

In this approach, a learnable neural texture T_n is defined with a resolution of 512×512, which is optimized end-to-end during training. To extract and encode texture information, we design a convolution-based texture encoder that estimates the texture atlas T_a. The texture encoder follows a sequential structure of convolutional layers interleaved with Rectified Linear Unit (ReLU) activation functions [54]. Specifically, the encoder begins with an initial convolutional layer to extract fundamental spatial features from T_n, followed by a ReLU activation that introduces non-linearity. As the data progresses through subsequent convolutional layers, the encoder progressively refines the extracted features, effectively aggregating local texture information. The output of the texture encoder is a texture atlas, which represents the spherical harmonics coefficients corresponding to different spatial locations. These coefficients are encoded as spherical harmonics, and the first three harmonics are used to generate RGB colors. This enables texture editing by directly modifying the texture atlas.

To accurately map the texture atlas to each Gaussian in the human avatar, we introduce a color query operation. This operation retrieves spherical harmonics coefficients based on the position of each Gaussian. Given the set of Gaussian positions $\{\mu_i\}_{i=1}^{L}$, we first normalize the positions to the range $[-1, 1]$. Then, we compute the mean value of the z-coordinate to distinguish between front-facing and back-facing points. Subsequently, the x- and y-coordinates are used as UV coordinates for bilinear sampling to obtain the corresponding spherical harmonics coefficients.

The overall texture mapping process can be expressed as:

$$T_a(\mu_i) = \text{BilinearSample}\left(\left(\frac{x_i}{W}, \frac{y_i}{H}\right), T_a\right), \tag{14}$$

where x_i and y_i are the normalized coordinates of the ith Gaussian, and W and H represent the width and height of the texture atlas. The bilinear sampling operation interpolates between the texture atlas values based on these UV coordinates.

Once the spherical harmonics coefficients are retrieved for each Gaussian, we combine them with the outputs from the local geometry prediction module to construct the final canonical Gaussian avatar G_c. By first predicting a UV color map and then querying colors for each Gaussian, we achieve two key benefits: (1) maintaining global color consistency through the unified texture representation, and (2) enabling intuitive texture editing capabilities that would be impossible with anchor-predicted colors alone. This avatar enables both shape and texture modifications, allowing for high-fidelity appearance and flexible editing of the human avatar.

Discussions. Compared with the hierarchical approach introduced by SC-GS [52], our local geometry and texture prediction has three main differences.

First, in terms of motivation, our anchor-based approach uniquely binds to SMPL model surfaces to provide comprehensive local geometric information (including transformations, positions, and opacity) while enabling avatar shape editing. However, SC-GS focus on compact transformation bases other than local geometric information aggregation or avatar editing.

Second, our implementation differs fundamentally through a hierarchical prediction framework that iteratively generates new Gaussians, as opposed to SC-GS's KNN-based interpolation of existing Gaussian properties. This direct prediction approach better preserves geometric details during editing operations.

Most importantly, our method achieves superior avatar reconstruction quality while enabling both shape and texture editing. To the best of our knowledge, these capabilities not demonstrated by previous GS-based approaches on avatar reconstruction from monocular video.

4.2. Deformation

Given the pose parameters θ of the SMPL model, our goal is to deform the canonical avatar into the observation space, i.e., the posed avatar. To capture the dynamic details of the human body, we adopt a learnable forward skinning approach, where we learn a skinning correction that adjusts the canonical model based on the human parametric model.

For each Gaussian G_i, the skinning weight $\mathcal{W}(\mu_i)$ is adjusted by adding a learnable correction term w_i, which yields the corrected skinning weight $\hat{\mathcal{W}}(\mu_i)$:

$$\hat{\mathcal{W}}(\mu_i) = \mathcal{W}(\mu_i) + w_i, \tag{15}$$

where $\mathcal{W}(\mu_i)$ represents the skinning weights queried at position μ_i in canonical space, obtained by diffusing the mesh skinning weights, and w_i represents the learnable skinning correction for the ith Gaussian G_i. This correction term allows for more accurate deformation, ensuring that the transformation is dynamically adjusted based on the pose.

After obtaining the corrected skinning weights $\hat{\mathcal{W}}(\mu_i)$, we replace the skinning weight in Eq. (5) with the estimated $\hat{\mathcal{W}}$. This substitution enables the deformation of the canonical avatar to generate the posed avatar.

Once the posed avatar is obtained, the rasterization process follows the standard procedure of 3D Gaussian Splatting (3DGS), as described in Section 3.2. This process involves projecting the posed avatar onto the image plane and performing point-based rendering to produce the final output.

4.3. Optimization

To optimize the entire model along with the learnable pose parameters, we employ multiple loss functions to minimize the discrepancy between the rendered and ground-truth images. Our loss formulation includes the standard RGB loss $\mathcal{L}_1$, the SSIM loss $\mathcal{L}_{\text{ssim}}$, and the LPIPS loss $\mathcal{L}_{\text{lpips}}$, each serving a specific role. The RGB loss enforces pixel-wise consistency, SSIM loss preserves structural similarity, and LPIPS loss ensures perceptual quality by aligning high-level feature representations. Additionally, to enhance fine details in the reconstructed images, we introduce a sharpening loss inspired by [37]. This loss helps capture high-frequency information, improving the clarity of rendered textures and edges.

Given a rendered image $\hat{I}$ and a ground-truth image I_r, the overall image loss is defined as:

$$\mathcal{L}_{\text{img}} = \alpha\mathcal{L}_1(I_r, \hat{I}) + (1-\alpha)\mathcal{L}_{\text{ssim}}(I_r, \hat{I}) + \beta\mathcal{L}_{\text{lpips}} + \gamma\mathcal{L}_{\text{sh}}, \tag{16}$$

where α, β, γ are weighting factors that balance the contributions of different losses.

To compute the sharpening loss, we employ the unsharp masking (USM) method [55], which derives a sharpening kernel by subtracting a Gaussian filter kernel from the identity kernel. This allows us to effectively extract high-frequency details, further enhancing the visual fidelity of the reconstructed images. Denote the sharpening kernel is f_s, the sharpening loss between rendered image $\hat{I}$ and a ground-truth image I_r is computed by:

$$\mathcal{L}_{\text{sh}} = |f_s(\hat{I}) - f_s(I_r)|_1. \tag{17}$$

We adopt MLP to predict geometric attributes, which can prove the smoothness similar with the NeRF-based methods when 2D observations are sparse. To further enhance the animation robustness, inspired by previous works [21,23], we adopt a regularization term, which constrain the local smooth for the geometric attributes of Gaussians. Specifically, we constrain the standard deviation of attributes in local Gaussians that computed by KNN neighborhood of current Gaussian, *i.e.*, G_i, which can be written as:

$$\mathcal{L}_{\text{reg}} = \sum_{att\in\{s_i,q_i,o_i,c_i\}_{i=1}^{L}} \text{STD}(att_k), \tag{18}$$

where c_i is the spherical harmonics of G_i, and STD is the standard deviation, and att_k is the attribute of kth Gaussian, and $k \in KNN(G_i)$ denotes the neighbor of G_i.

Therefore, the full loss functions can be written as:

$$\mathcal{L} = \mathcal{L}_{\text{img}} + \mathcal{L}_{\text{reg}}. \tag{19}$$

4.4. Implemented details

We implemented the model using PyTorch on a desktop with a NVIDIA 2080Ti with 12G GPU memory. The weights $[\alpha, \beta, \gamma]$ are set to $[0.8, 0.2, 1]$. For training, we employ the Adam optimizer to optimize all model parameters. During inference, all the attributes predicted by the geometry and texture decoders are explicitly stored for each Gaussian. As a result, during inference, we no longer require the neural network to predict these attributes. Instead, we simply deform and render the canonical avatar using the proposed pose parameters, enabling fast inference speeds. Specifically, the model achieves a frame rate of 110 FPS. Following GART [21], we adopt pose optimization during training to correct the inaccurate pose parameters. Besides, we use order-2 spherical harmonics (SH) coefficients for color representation. The RGB values are converted to SH coefficients as the firs-order SH coefficients through an SH-to-RGB transformation during training. We first optimize the first-order SH coefficients to obtain a stable texture map, then progressively increase the SH orders in later iterations.

5. Experimental results

In this section, we first introduce the experimental settings, including the evaluation datasets and metrics. Then, we present both quantitative and qualitative results, comparing our method with several state-of-the-art approaches. Finally, we provide ablation studies to validate the effectiveness of our proposed contributions.

5.1. Experimental settings

For monocular avatar reconstruction, our model design follows the dominant paradigm in the field—unlike generalized models capable of inferring unseen subjects directly, our approach adopts an identity-specific optimization framework. Specifically, the model is trained using image data of a single identity, and the reconstructed avatar corresponds exclusively to the target subject in the training set. This avatar supports subsequent tasks such as novel view synthesis, novel pose generation, and avatar editing for the specific subject. Therefore, to validate the effectiveness of the proposed method, we conduct experiments on two widely-used public datasets: the ZJU-MoCap dataset [29] and the People-Snapshot dataset [28]. We evaluate our results using three different metrics.

ZJU-MoCap dataset. The ZJU-MoCap dataset consists of multi-view videos of humans performing diverse poses from multiple camera angles. The pose parameters of this dataset are obtained from EasyMoCap [61]. Following the protocols established in previous works [21,60], we evaluate six distinct identities, using the same data splits for consistency in benchmarking. For each identity, one camera view is designated as the training data, and the remaining views are reserved for testing. This setup ensures a fair comparison with existing methods. To evaluate the efficacy of our approach, we compare it against three state-of-the-art methods: MonoHuman [59], Instant-NVR [60], GART [21], and GauHuman [22]. We reproduce their results using publicly available implementations and the same training/testing data splits to ensure a consistent comparison. We adopt black background for both quantitative results and qualitative results following GART.

People-Snapshot dataset. The People-Snapshot dataset contains videos of humans performing rotations in an A-pose in front of a static camera. The pose parameters of People-Snapshot are obtained from AnimNeRF [62]. Following prior works [19,21], we evaluate our method on four selected identities and use the same training-testing data splits to ensure fair comparisons. As noted, many recent avatar reconstruction approaches (including Instant-NVR and GauHuman) are not evaluated on People-Snapshot in their original publications, nor do their release implementations support this dataset. To ensure fair and meaningful comparisons, we follow the established practice in GART of organizing results by dataset availability. We benchmark our approach against InstantAvatar [19], 3DGS-Avatar [23], and GART [21]. To ensure reproducibility, we utilize the publicly available codebases and default hyperparameters provided by these methods during training. Notably, for 3DGS-Avatar, we incorporate pose refinement during testing, which enhances the reconstruction quality beyond the results reported in the original paper. Note that we adopt white background for both quantitative results and qualitative results following GART, which is different with 3DGS-Avatar.

Table 1
Comparison on ZJU-MoCap dataset [29]. Note that all metrics are computed with images in black background following GART [21].

	377			386			387		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
NB [29]	29.50	0.970	28.3	31.05	0.970	39.9	27.14	0.955	47.6
AN [56]	29.91	0.971	32.9	31.69	0.970	44.5	27.03	0.957	54.0
NHP [57]	27.67	0.957	60.1	30.62	0.965	55.8	26.23	0.952	69.7
InstantAvatar [58]	29.65	0.973	19.2	28.0	0.965	34.6	27.90	0.972	24.9
MonoHuman [59]	29.12	0.973	26.6	32.94	0.970	36.0	27.93	0.960	41.8
Instant-nvr [60]	31.36	0.979	26.0	33.53	0.977	33.0	28.11	0.963	47.0
GART [21]	31.90	0.975	18.8	33.50	0.967	29.9	27.74	0.952	40.3
GauHuman [22]	32.24	0.976	18.9	33.72	0.969	29.0	28.19	0.956	39.3
Ours	**34.30**	**0.989**	**12.5**	**34.73**	**0.982**	**24.2**	**29.53**	**0.972**	**29.2**
	392			393			394		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
NB [29]	28.38	0.965	44.4	28.38	0.958	49.6	29.37	0.963	45.1
AN [56]	31.18	0.968	47.7	28.55	0.959	52.9	30.28	0.964	49.5
NHP [57]	29.30	0.956	66.6	27.13	0.949	70.5	28.53	0.951	65.7
InstantAvatar [58]	29.65	0.973	19.2	27.97	0.965	34.6	27.90	0.972	24.9
MonoHuman [59]	29.50	0.964	39.5	27.64	0.957	43.2	29.15	0.960	38.1
Instant-nvr [60]	32.03	0.973	39.3	29.55	0.964	46.3	31.46	0.969	39.1
GART [21]	31.92	0.964	32.6	29.34	0.954	37.9	31.08	0.958	31.5
GauHuman [22]	32.27	0.967	30.2	30.24	0.958	35.2	31.42	0.961	30.6
Ours	**33.16**	**0.980**	**26.0**	**30.32**	**0.971**	**30.6**	**32.72**	**0.977**	**25.1**

Table 2
Comparison on People-Snapshot [28]. Note that all metrics are computed with images in white background following GART [21].

	Male-3-casual			Male-4-casual			Female-3-casual			Female-4-casual		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
NB [29]	24.94	0.9428	32.6	24.71	0.9469	42.3	23.87	0.9504	34.6	24.37	0.9451	38.2
InstantAvatar [58]	29.65	0.9730	19.2	27.97	0.9649	34.6	**27.90**	**0.9722**	24.9	28.92	0.9692	18.0
3DGS-Avatar [23]	28.01	0.9657	33.5	27.25	0.9614	47.4	27.09	0.9679	29.0	27.62	0.9631	31.3
GART [21]	30.40	0.9769	37.7	27.57	0.9657	60.7	26.26	0.9656	49.8	29.23	0.9720	37.8
Ours	**30.56**	**0.9791**	**16.3**	**28.31**	**0.9701**	**29.4**	27.60	0.9700	**22.3**	**29.99**	**0.9763**	**16.4**

Metrics. To evaluate reconstruction quality, we employ three metrics: PSNR, SSIM, and LPIPS. PSNR measures pixel-wise fidelity by computing the mean squared error, with higher values indicating better quality. SSIM assesses structural similarity by considering luminance, contrast, and texture, providing a more perceptual measure of image similarity [63,64]. LPIPS captures perceptual differences by utilizing deep feature representations, with lower values signifying higher perceptual fidelity [65]. We report the LPIPS scores scaled by a factor of 10^3. These three metrics together offer a comprehensive evaluation of both reconstruction accuracy and visual quality.

5.2. Comparison results

Quantitative results. Quantitative comparisons on the ZJU-MoCap and People-Snapshot datasets are presented in Tables 1 and 2, respectively. As shown in Table 1, our method consistently outperforms existing approaches across PSNR, SSIM, and LPIPS metrics. The highest PSNR values indicate superior pixel-wise reconstruction accuracy, while the highest SSIM scores demonstrate better structural preservation. Additionally, our approach achieves the lowest LPIPS values, reflecting improved perceptual quality. These results confirm the effectiveness of our method in generating high-fidelity reconstructions. A similar trend is observed in Table 2. Although our method does not achieve the highest PSNR and SSIM scores for "female-3-causal", it achieves the best LPIPS, indicating more perceptually realistic renderings. Moreover, our method consistently outperforms competing approaches on the remaining subjects, demonstrating its ability to reconstruct high-quality avatars with enhanced visual fidelity. For detailed comparison with 3DGS-Avatar can be found in the supplemental document.

Qualitative results. Figs. 3 and 4 present qualitative comparisons on the ZJU-MoCap and People-Snapshot datasets, respectively. Compared to other methods, our approach generates more detailed renderings, particularly in the intricate regions of clothing and hands. Notably, fine patterns in the fourth and fifth rows are better preserved, demonstrating the efficacy of our method in capturing subtle textures. Both GART and GauHuman, which utilize 3D Gaussian Splatting (3DGS) as their representation, exhibit enhanced visual quality over NeRF-based methods such as Instant-NVR, particularly in terms of edge sharpness and detail preservation. Our method further improves these results through the proposed local geometry and texture prediction mechanisms, which facilitate more accurate surface reconstruction. By enabling local Gaussians to interact and influence each other during optimization, our approach effectively preserves fine-grained details and maintains sharp boundaries. Consequently, our model achieves high-quality avatar reconstruction with superior rendering fidelity, setting a new benchmark in the field.

5.3. Ablation study

To systematically evaluate the effectiveness of key components in LoGAvatar, we conduct comprehensive ablation studies by comparing our full implementation against the following model variants:

Model without local geometry prediction (w/o log). This variant removes the local geometry prediction module to assess its contribution. Unlike our approach, which predicts geometric attributes from anchor points, this baseline treats each Gaussian as an independent entity with directly optimizable geometric parameters, following the conventional 3D Gaussian Splatting (3DGS) training paradigm. All other components remain unchanged.

Fig. 3. Qualitative results on ZJU-MoCap dataset compared with Instant-nvr [58], GauHuman [23], and GART [21]. Please zoom in for details.

Model without local texture prediction (w/o lot). To evaluate the impact of the local texture prediction module, this variant replaces it with standard optimizable color tensors per Gaussian. While geometric learning remains intact, the appearance modeling follows conventional per-Gaussian optimization.

Model without sharpening loss (w/o sha). This experiment removes the sharpening loss to assess its role in preserving high-frequency details and refining geometry. The absence of this regularization allows us to analyze its effect on detail sharpness and artifact suppression.

The ablation studies validate the necessity of each proposed component. Removing local geometry prediction leads to the most significant performance degradation, as directly optimizing isolated Gaussians fails to maintain structural coherence. Our anchor-based formulation addresses this through geometric attribute learning, enforcing local similarity while permitting flexible Gaussians. The absence of local texture prediction notably impacts perceptual quality, particularly in high-frequency detail preservation. This confirms the advantage of the design of the local texture prediction module, which successfully captures fine details. While removing the sharpening loss maintains comparable performance in some metrics, the increased variance across test cases reveals its role as a crucial regularization. Our full model achieves superior performance in almost all metrics, which aligns with our theoretical analysis in Section 4, confirming the effectiveness of the proposed design.

Fig. 5 illustrates the ablation study results on the People-Snapshot dataset, highlighting the contributions of key components in our framework. The model without local geometry prediction (w/o log) introduces noticeable artifacts, particularly in challenging poses, such as bent limbs or complex body rotations, whereas our full model produces clear and coherent reconstructions with precise anatomical details. Similarly, the model without local texture prediction (w/o lot) struggles to capture intricate textures, such as fabric patterns or skin pores, resulting in less realistic appearances and blurred surfaces. While the

Fig. 4. Qualitative results on People-Snapshot dataset compared with InstantAvatar [58], 3DGS-Avatar [23], and GART [21]. Please zoom in for details.

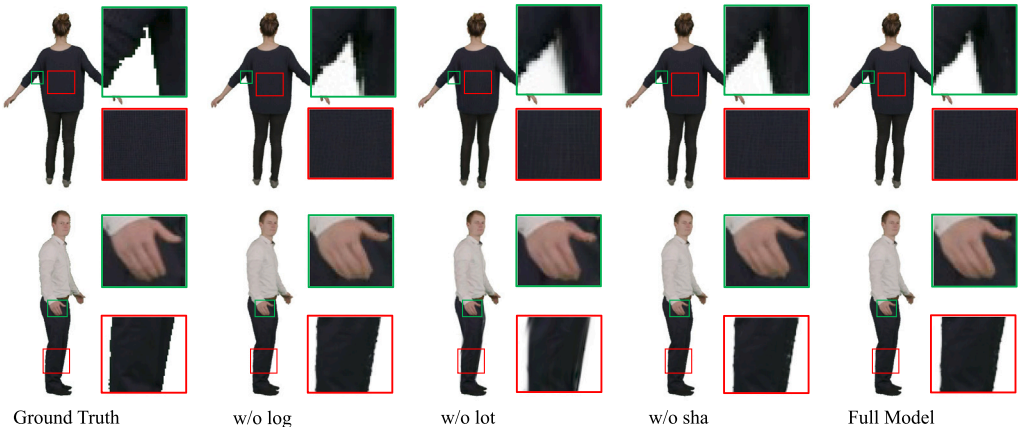


Fig. 5. Ablation results on People-Snapshot dataset. Please zoom in for details.

Table 3
Ablation study on People-Snapshot [28].

	Male-3-casual			Male-4-casual			Female-3-casual			Female-4-casual		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
w/o log	30.54	**0.9791**	17.2	28.23	0.9694	30.3	27.43	0.9697	23.0	29.97	0.9759	**16.4**
w/o lot	30.37	0.9770	23.6	28.02	0.9666	45.5	27.31	0.9663	38.6	29.89	0.9752	21.8
w/o sha	30.52	0.9789	16.6	28.13	0.9697	**28.8**	27.48	0.9695	22.4	29.79	0.9755	16.5
Ours	**30.56**	**0.9791**	**16.3**	**28.31**	**0.9701**	29.4	**27.60**	**0.9700**	**22.3**	**29.99**	**0.9763**	**16.4**

variant without sharpening loss (w/o sha) still generates reasonable outputs, it fails to preserve fine details (*e.g.*, the stripes in the first row) and introduces artifacts in specific regions (*e.g.*, the legs in the second row), compromising the overall visual fidelity. In contrast, our full model consistently delivers high-quality avatars with sharper and more accurate textures, demonstrating the effectiveness of our integrated approach in achieving realistic and detailed avatar reconstruction. Table 3 also suggests similar conclusions.

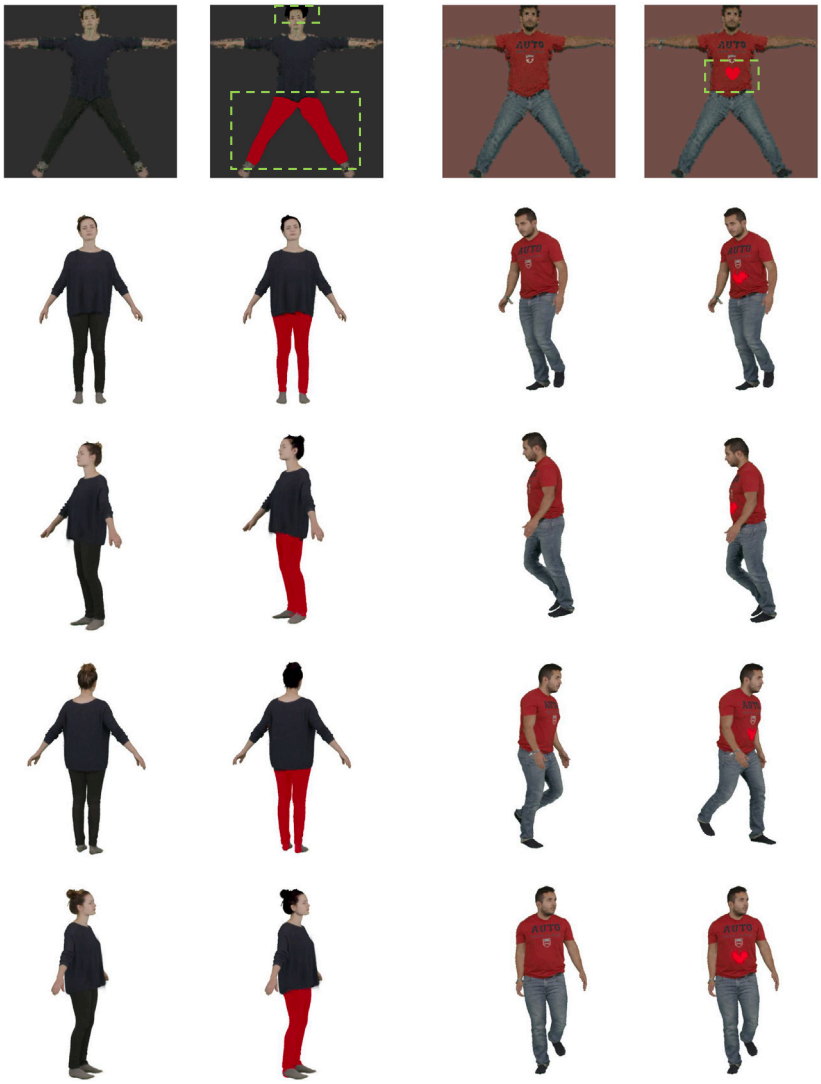


Fig. 6. Texture editing results. The first row shows the learned front texture and the modified front texture after editing.

Different training time. Compared with GART, our model incorporates neural networks to achieve local geometry and local texture prediction, which can decrease the training and rendering speed. To investigate this, we conduct experiments by comparing with GART in different training times. The quantitative results can be found in Table 4. which compares performance across different training durations (Time), including corresponding rendering speeds (Frame-Per-Second, FPS) and iteration counts (Iter.) achieved within each time interval. Note that the unit of time is minutes.

Our experiments demonstrate that while the introduced neural components do incur some computational overhead, the impact on overall training speed is mitigated through careful architectural design. Specifically, on the People-Snapshot dataset, our method achieves rendering speeds exceeding 110 fps while maintaining superior reconstruction quality compared to existing approaches. Importantly, when comparing methods trained for identical wall-clock time, our approach consistently outperforms GART despite requiring fewer iterations, highlighting its efficiency. Furthermore, we observe that unlike GART's performance which tends to plateau with extended training, our method continues to show quality improvements with additional training time. This suggests that our architecture can effectively utilize increased computational resources when available.

The number of predicted Gaussians. In Section 4.1.1, we adopt $N = 4$, which means one anchor point can predict 4 Gaussians. To thoroughly investigate value of N, we conducted comprehensive experiments testing various configurations from 1 to 20 across multiple subjects. As shown in Table 5, the results demonstrate consistent performance across different balues, with the $N = 4$ configuration achieving optimal balance between quality and efficiency. For instance, the Female-4-casual case shows PSNR of 29.99 at $N = 4$ compared to 29.65 at $N = 1$ and 29.72 at $N = 10$. Higher ratios (1:10 and 1:20) exhibit marginally reduced geometric detail as evidenced by slightly elevated LPIPS values, while lower ratios incur unnecessary computational overhead without commensurate quality improvements.

5.4. Applications

Texture editing. Leveraging the local texture prediction module, our model generates front and back Gaussian maps, which enable intuitive texture editing by modifying these maps directly.

Fig. 6 illustrates two texture editing cases. The first two columns of the first row display the original Gaussian map and its edited version, with the modified regions highlighted by green bounding boxes. The remaining rows present the corresponding rendered images using different Gaussian maps. In the first case, the same regions of the back

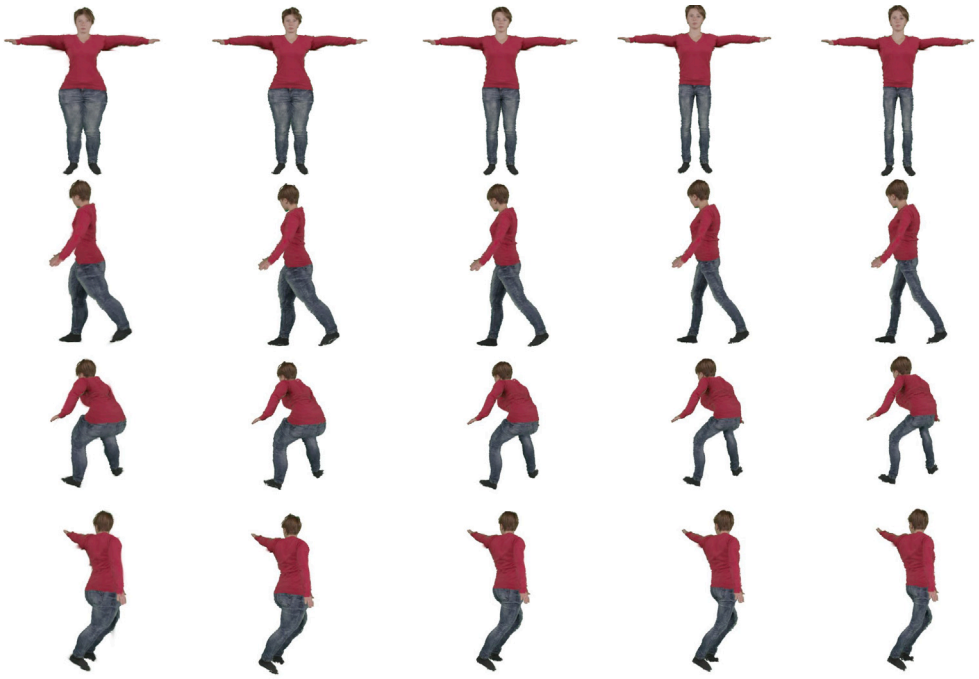


Fig. 7. Shape editing results. The middle row shows the reconstructed avatar and the other rows present modified results.

Table 4
Ablation study on People-Snapshot [28].

	Time	FPS	Iter.	Male-3-casual			Male-4-casual			Female-3-casual			Female-4-casual		
				PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
GART	1	150	2000	26.25	0.9674	44.1	**29.52**	**0.9742**	32.1	**30.72**	**0.9797**	33.3	27.55	0.9686	54.5
Ours	1	110	500	**26.58**	**0.9676**	**25.3**	29.28	0.9735	**17.8**	30.01	0.9763	**18.4**	**27.57**	**0.9669**	**34.7**
GART	2	145	3000	26.11	0.9670	46.5	29.15	0.9740	35.0	**30.68**	**0.9801**	36.1	27.50	**0.9689**	59.1
Ours	2	110	1000	**26.79**	**0.9684**	**25.5**	**29.57**	**0.9741**	**19.6**	30.27	0.9772	**19.7**	**27.74**	0.9674	**35.8**
GART	30	140	50 000	25.88	0.9663	36.9	28.70	0.9738	27.9	**30.67**	**0.9808**	29.2	27.19	0.9688	46.1
Ours	30	110	10 000	**27.60**	**0.9700**	**22.3**	**29.99**	**0.9763**	**16.4**	30.56	0.9791	**16.3**	**28.31**	**0.9701**	**29.4**

Table 5
Ablation study on People-Snapshot [28].

N	Male-3-casual			Male-4-casual			Female-3-casual			Female-4-casual		
	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS	PSNR	SSIM	LPIPS
1	330.36	0.9772	14.9	27.82	0.9670	25.6	27.36	0.9688	19.1	29.65	0.9743	15.0
2	30.48	0.9780	13.7	27.94	0.9679	24.2	27.37	0.9694	**18.4**	29.69	0.9746	**14.0**
4	**30.56**	0.9791	16.3	**28.31**	0.9701	29.4	**27.60**	0.9700	22.3	**29.99**	**0.9763**	16.4
10	30.52	0.9786	**13.1**	27.98	0.9680	**23.7**	27.36	0.9695	18.5	29.72	0.9752	14.4
20	30.53	**0.9797**	16.5	28.20	**0.9705**	30.4	27.55	**0.9710**	27.7	29.88	0.9762	16.7

Gaussian map are also edited to maintain consistency. As observed, our approach allows direct modifications to the Gaussian maps, enabling efficient texture editing of reconstructed avatars. The resulting renderings maintain consistency across different poses, demonstrating the robustness of our method for avatar customization.

Shape editing. The results of local geometry prediction are derived from the fixed surface points of the SMPL model. This indicates that modifying the shape parameters of the SMPL model allows us to edit the final avatar's shape. Specifically, by adjusting these shape parameters, we retain the same surface points with identical indices as the originally sampled points. Consequently, the position of the anchor point is altered. Utilizing the other estimated geometric and texture attributes, we can generate an animatable avatar with the new shape.

Fig. 7 showcases visual examples of leg and stomach editing. It is evident that our method effectively modifies the shape of the reconstructed avatar while preserving high-quality rendering results across various poses.

5.5. Limitation

To achieve an editable texture Gaussian map, we disregard texture variations across different poses, which limits the photorealistic quality of the results in diverse poses. In future work, we plan to retain the basic Gaussian map and introduce a position-aware texture residual to enhance the realism of the outcomes. Besides, our current work does not account for complex scenes with occlusions in the input images. We recognize this as a critical direction and plan to address it in our future research. Additionally, we aim to incorporate physics-based priors to model more dynamic and natural details.

6. Conclusion

This paper presents LoGAvatar, a novel approach for high-fidelity 3D human avatar reconstruction based on local geometry and texture prediction. By introducing an anchor-based local geometry prediction

module, our method effectively captures fine-grained shape details while maintaining structural consistency. Additionally, the proposed local texture prediction module enables enhanced texture representation, leading to improved perceptual quality. Through extensive experiments on the ZJU-MoCap and People-Snapshot datasets, LoGAvatar consistently outperforms state-of-the-art methods in both quantitative and qualitative evaluations. Ablation studies further validate the effectiveness of each proposed component. Additionally, we demonstrate the flexibility of our method for texture and shape editing, showcasing its potential for avatar customization and interactive applications.

CRediT authorship contribution statement

Jinsong Zhang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Xiongzheng Li:** Writing – original draft, Methodology, Formal analysis. **Hailong Jia:** Writing – original draft, Visualization. **Jin Li:** Visualization. **Zhuo Su:** Supervision, Funding acquisition. **Guidong Wang:** Supervision. **Kun Li:** Writing – review & editing, Writing – original draft, Supervision, Resources, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by National Key R&D Program of China (2023YFC3082100), National Natural Science Foundation of China (62171317), and Science Fund for Distinguished Young Scholars of Tianjin (No. 22JCJQJC00040).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cad.2025.103973.

Data availability

Data will be made available on request.

References

[1] Yu K, Gorbachev G, Eck U, Pankratz F, Navab N, Roth D. Avatars for teleconsultation: Effects of avatar embodiment techniques on user perception in 3d asymmetric telepresence. IEEE Trans Vis Comput Graphics 2021;27(11):4129–39.
[2] Schwarz M, Lenz C, Memmesheimer R, Pätzold B, Rochow A, Schreiber M, Behnke S. Robust immersive telepresence and mobile telemanipulation: Nimbro wins ana avatar xprize finals. In: 2023 IEEE-RAS 22nd international conference on humanoid robots (humanoids). IEEE; 2023, p. 1–8.
[3] Bourdot P, Convard T, Picon F, Ammi M, Touraine D, Vézien J-M. VR–CAD integration: Multimodal immersive interaction and advanced haptic paradigms for implicit edition of CAD models. Computer-Aided Des 2010;42(5):445–61.
[4] Yuan M, Khan IR, Farbiz F, Yao S, Niswar A, Foo M-H. A mixed reality virtual clothes try-on system. IEEE Trans Multimed 2013;15(8):1958–68.
[5] Santesteban I, Otaduy MA, Casas D. Learning-based animation of clothing for virtual try-on. In: Comput. graph. forum, vol. 38, (2):Wiley Online Library; 2019, p. 355–66.
[6] Fang N, Qiu L, Zhang S, Wang Z, Wang Y, Gu Y, Tan J. A modeling method for the human body model with facial morphology. Computer-Aided Des 2021;141:103106.
[7] Jiang L, Ye J, Sun L, Li J. Transferring and fitting fixed-sized garments onto bodies of various dimensions and postures. Computer-Aided Des 2019;106:30–42.
[8] Lee S, El Ali A, Wijntjes M, Cesar P. Understanding and designing avatar biosignal visualizations for social virtual reality entertainment. In: Proceedings of the 2022 CHI conference on human factors in computing systems. 2022, p. 1–15.
[9] Miao F, Kozlenkova IV, Wang H, Xie T, Palmatier RW. An emerging theory of avatar marketing. J Mark 2022;86(1):67–90.
[10] Kosmadoudi Z, Lim T, Ritchie J, Louchart S, Liu Y, Sung R. Engineering design using game-enhanced CAD: The potential to augment the user experience with game elements. Computer-Aided Des 2013;45(3):777–95.
[11] Collet A, Chuang M, Sweeney P, Gillett D, Evseev D, Calabrese D, Hoppe H, Kirk A, Sullivan S. High-quality streamable free-viewpoint video. ACM Trans Graph 2015;34(4):1–13.
[12] Li Z, Zheng Z, Wang L, Liu Y. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human avatar modeling. In: IEEE conf. comput. vis. pattern recog.. 2024, p. 19711–22.
[13] Zheng Y, Zhao Q, Yang G, Yifan W, Xiang D, Dubost F, Lagun D, Beeler T, Tombari F, Guibas L, et al. Physavatar: Learning the physics of dressed 3d avatars from visual observations. In: Eur. conf. comput. vis.. Springer; 2024, p. 262–84.
[14] Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R. Nerf: Representing scenes as neural radiance fields for view synthesis. Commun ACM 2021;65(1):99–106.
[15] Hu T, Liu S, Chen Y, Shen T, Jia J. Efficientnerf efficient neural radiance fields. In: IEEE conf. comput. vis. pattern recog.. 2022, p. 12902–11.
[16] Li R, Gao H, Tancik M, Kanazawa A. Nerfacc: Efficient sampling accelerates nerfs. In: IEEE conf. comput. vis. pattern recog.. 2023, p. 18537–46.
[17] Li X, Zhang J, Lai Y-K, Yang J, Li K. High-quality animatable dynamic garment reconstruction from monocular videos. IEEE Trans Circuits Syst Video Technol 2023;34(6):4243–56.
[18] Fridovich-Keil S, Yu A, Tancik M, Chen Q, Recht B, Kanazawa A. Plenoxels: Radiance fields without neural networks. In: IEEE conf. comput. vis. pattern recog.. 2022, p. 5501–10.
[19] Jiang T, Chen X, Song J, Hilliges O. Instantavatar: Learning avatars from monocular video in 60 seconds. In: IEEE conf. comput. vis. pattern recog.. 2023, p. 16922–32.
[20] Kerbl B, Kopanas G, Leimkühler T, Drettakis G. 3D gaussian splatting for real-time radiance field rendering. ACM Trans Graph 2023;42(4):1–14.
[21] Lei J, Wang Y, Pavlakos G, Liu L, Daniilidis K. Gart: Gaussian articulated template models. In: IEEE conf. comput. vis. pattern recog.. 2024, p. 19876–87.
[22] Hu S, Hu T, Liu Z. Gauhuman: Articulated gaussian splatting from monocular human videos. In: IEEE conf. comput. vis. pattern recog.. 2024, p. 20418–31.
[23] Qian Z, Wang S, Mihajlovic M, Geiger A, Tang S. 3Dgs-avatar: Animatable avatars via deformable 3d gaussian splatting. In: IEEE conf. comput. vis. pattern recog.. 2024, p. 5020–30.
[24] Wen J, Zhao X, Ren Z, Schwing AG, Wang S. Gomavatar: Efficient animatable human modeling from monocular video using gaussians-on-mesh. In: IEEE conf. comput. vis. pattern recog.. 2024, p. 2059–69.
[25] Zhang J, Shen I-C, Sakamiya J, Lai Y-K, Igarashi T, Li K. DualAvatar: Robust Gaussian splatting avatar with dual representation. In: SIGGRAPH Asia 2024 posters. 2024, p. 1–3.
[26] Loper M, Mahmood N, Romero J, Pons-Moll G, Black MJ. SMPL: A skinned multi-person linear model. In: Seminal graphics papers: pushing the boundaries. vol. 2, 2023, p. 851–66.
[27] Pavlakos G, Choutas V, Ghorbani N, Bolkart T, Osman AA, Tzionas D, Black MJ. Expressive body capture: 3d hands, face, and body from a single image. In: IEEE conf. comput. vis. pattern recog.. 2019, p. 10975–85.
[28] Alldieck T, Magnor M, Xu W, Theobalt C, Pons-Moll G. Video based reconstruction of 3d people models. In: IEEE conf. comput. vis. pattern recog.. 2018, p. 8387–97.
[29] Peng S, Zhang Y, Xu Y, Wang Q, Shuai Q, Bao H, Zhou X. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In: IEEE conf. comput. vis. pattern recog.. 2021, p. 9054–63.
[30] Chen Y, Zheng Z, Li Z, Xu C, Liu Y. Meshavatar: Learning high-quality triangular human avatars from multi-view videos. In: Eur. conf. comput. vis.. Springer; 2024, p. 250–69.
[31] Bogo F, Kanazawa A, Lassner C, Gehler P, Romero J, Black MJ. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In: Eur. conf. comput. vis.. Springer; 2016, p. 561–78.
[32] Huang Y, Bogo F, Lassner C, Kanazawa A, Gehler PV, Romero J, Akhter I, Black MJ. Towards accurate marker-less human shape and pose estimation over time. In: Int. conf. on. 3D vision.. IEEE; 2017, p. 421–30.
[33] Kanazawa A, Black MJ, Jacobs DW, Malik J. End-to-end recovery of human shape and pose. In: IEEE conf. comput. vis. pattern recog.. 2018, p. 7122–31.
[34] Pavlakos G, Choutas V, Ghorbani N, Bolkart T, Osman AA, Tzionas D, Black MJ. Expressive body capture: 3d hands, face, and body from a single image. In: IEEE conf. comput. vis. pattern recog.. 2019, p. 10975–85.
[35] Kolotouros N, Pavlakos G, Daniilidis K. Convolutional mesh regression for single-image human shape reconstruction. In: IEEE conf. comput. vis. pattern recog.. 2019, p. 4501–10.
[36] Li X, Huang J, Zhang J, Sun X, Xuan H, Lai Y-K, Xie Y, Yang J, Li K. Learning to infer inner-body under clothing from monocular video. IEEE Trans Vis Comput Graphics 2022;29(12):5083–96.

[37] Zhao H, Zhang J, Lai Y-K, Zheng Z, Xie Y, Liu Y, Li K. High-fidelity human avatars from a single rgb camera. In: IEEE conf. comput. vis. pattern recog.. 2022, p. 15904–13.
[38] Alldieck T, Magnor M, Bhatnagar BL, Theobalt C, Pons-Moll G. Learning to reconstruct people in clothing from a single RGB camera. In: IEEE conf. comput. vis. pattern recog.. 2019, p. 1175–86.
[39] Weng C-Y, Curless B, Srinivasan PP, Barron JT, Kemelmacher-Shlizerman I. Humannerf: Free-viewpoint rendering of moving people from monocular video. In: IEEE conf. comput. vis. pattern recog.. 2022, p. 16210–20.
[40] Peng S, Dong J, Wang Q, Zhang S, Shuai Q, Zhou X, Bao H. Animatable neural radiance fields for modeling dynamic human bodies. In: Int. conf. comput. vis.. 2021, p. 14314–23.
[41] Su S-Y, Yu F, Zollhöfer M, Rhodin H. A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. In: Adv. neural inform. process. syst., 2021, p. 12278–91.
[42] Wang S, Schwarz K, Geiger A, Tang S. Arah: Animatable volume rendering of articulated human sdfs. In: Eur. conf. comput. vis.. Springer; 2022, p. 1–19.
[43] Liu L, Habermann M, Rudnev V, Sarkar K, Gu J, Theobalt C. Neural actor: Neural free-view synthesis of human actors with pose control. ACM Trans Graph 2021;40(6):1–16.
[44] Gao Q, Wang Y, Liu L, Liu L, Theobalt C, Chen B. Neural novel actor: Learning a generalized animatable neural representation for human actors. IEEE Trans Vis Comput Graphics 2023;30(8):5719–32.
[45] Hedman P, Srinivasan PP, Mildenhall B, Barron JT, Debevec P. Baking neural radiance fields for real-time view synthesis. In: Int. conf. comput. vis.. 2021, p. 5875–84.
[46] Chen Y, Wang L, Li Q, Xiao H, Zhang S, Yao H, Liu Y. Monogaussianavatar: Monocular gaussian point-based head avatar. In: ACM SIGGRAPH 2024 conference papers. 2024, p. 1–9.
[47] Shao Z, Wang Z, Li Z, Wang D, Lin X, Zhang Y, Fan M, Wang Z. Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting. In: IEEE conf. comput. vis. pattern recog.. 2024, p. 1606–16.
[48] Dongye X, Guo H, Jiang H, Weng D. Adaptive levels of detail for human Gaussian splats with hierarchical embedding. In: 2024 IEEE international symposium on mixed and augmented reality adjunct. IEEE; 2024, p. 361–2.
[49] Paudel P, Khanal A, Paudel DP, Tandukar J, Chhatkuli A. Ihuman: Instant animatable digital humans from monocular videos. In: Eur. conf. comput. vis.. Springer; 2024, p. 304–23.
[50] Moon G, Shiratori T, Saito S. Expressive whole-body 3D gaussian avatar. In: Eur. conf. comput. vis.. Springer; 2024, p. 19–35.
[51] Lin S, Li Z, Su Z, Zheng Z, Zhang H, Liu Y. Layga: Layered Gaussian avatars for animatable clothing transfer. In: SIGGRAPH conference papers. 2024.
[52] Huang Y-H, Sun Y-T, Yang Z, Lyu X, Cao Y-P, Qi X. SC-GS: Sparse-controlled gaussian splatting for editable dynamic scenes. In: IEEE conf. comput. vis. pattern recog.. 2024, p. 4220–30.
[53] Müller T, Evans A, Schied C, Keller A. Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans Graph 2022;41(4):1–15.
[54] Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. In: International conference on machine learning. 2010, p. 807–14.
[55] Neycenssac F. Contrast enhancement using the Laplacian-of-a-Gaussian filter. CVGIP, Graph Models Image Process 1993;55(6):447–63.
[56] Zhou X, Peng S, Xu Z, Dong J, Wang Q, Zhang S, Shuai Q, Bao H. Animatable implicit neural representations for creating realistic avatars from videos. IEEE Trans Pattern Anal Mach Intell 2024;46(6):4147–59.
[57] Kwon Y, Kim D, Ceylan D, Fuchs H. Neural human performer: Learning generalizable radiance fields for human performance rendering. In: Adv. neural inform. process. syst, vol. 34, 2021, p. 24741–52.
[58] Jiang T, Chen X, Song J, Hilliges O. Instantavatar: Learning avatars from monocular video in 60 seconds. In: IEEE conf. comput. vis. pattern recog.. 2023, p. 16922–32.
[59] Yu Z, Cheng W, Liu X, Wu W, Lin K-Y. Monohuman: Animatable human neural field from monocular video. In: IEEE conf. comput. vis. pattern recog.. 2023, p. 16943–53.
[60] Geng C, Peng S, Xu Z, Bao H, Zhou X. Learning neural volumetric representations of dynamic humans in minutes. In: IEEE conf. comput. vis. pattern recog.. 2023, p. 8759–70.
[61] Shuai Q, Fang Q, Dong J, Peng S, Huang D, et al. Easymocap-make human motion capture easier. Github 2021;1(3):6.
[62] Chen J, Zhang Y, Kang D, Zhe X, Bao L, Jia X, Lu H. Animatable neural radiance fields from monocular rgb videos. 2021, arXiv preprint arXiv:2106.13629.
[63] Li K, Zhang J, Liu Y, Lai Y-k, Dai Q. PoNA: Pose-guided non-local attention for human pose transfer. IEEE Trans Image Process 2020;29:9584–99.
[64] Zhang J, Li K, Yu-Kun L, Jingyu Y. PISE: Person image synthesis and editing with decoupled GAN. In: IEEE conf. comput. vis. pattern recog.. 2021.
[65] Zhang J, Liu X, Li K. Human pose transfer by adaptive hierarchical deformation. Comput Graph Forum 2020;39(7):325–37.