GaussianHand: Real-time 3D Gaussian Rendering for Hand Avatar Animation

Lizhi Zhao, Xuequan Lu, Runze Fan, Sio Kei Im, Lili Wang

***Abstract*—Rendering animatable and realistic hand avatars is pivotal for enhancing user experiences in human-centered AR/VR applications. While recent initiatives have utilized neural radiance fields to forge hand avatars with lifelike appearances, these methods are often hindered by high computational demands and the necessity for extensive training views. In this paper, we introduce GaussianHand, the first Gaussian-based real-time 3D rendering approach that enables efficient free-view and free-pose hand avatar animation from sparse view images. Our approach encompasses two key innovations. We first propose Hand Gaussian Blend Shapes that effectively models hand surface geometry while ensuring consistent appearance across various poses. Secondly, we introduce the Neural Residual Skeleton, equipped with Residual Skinning Weights, designed to rectify inaccuracies involved in Linear Blend Skinning deformations due to geometry offsets. Experiments demonstrate that our method not only achieves far more realistic rendering quality with as few as 5 or 20 training views, compared to the 139 views required by existing methods, but also excels in efficiency, achieving up to 125 frames per second for real-time rendering and remarkably surpassing recent methods.**

***Index Terms*—Virtual Reality, Real-time Rendering, 3D Gaussian Splatting, Hand Avatar Animation**

I. Introduction

ADVANCEMENTS in AR/VR technologies are mixing the humans' physical and computers' virtual worlds [2]–[6]. Hands are crucial for creating immersive human-computer interaction experiences [7], [8]. Thus, rendering and animating realistic hand avatars from motion capture images is essential for human-centered AR/VR applications [9]–[13].

Recent methods for constructing animatable hand avatars with realistic appearance fall into two categories: mesh-based and neural radiance field (NeRF)-based. Mesh-based approaches reconstruct mesh geometry and physical-aware lighting maps for photorealistic rendering [10], [11]. In contrast, NeRF-based hand avatars use implicit radiance field representations [14] to create articulated, animatable radiance fields for free-view rendering [1], [15]–[17].

In addition to these two types of methods, the recent breakthrough in rendering achieved by Gaussian Splatting [18], [19] provides researchers with a new representation. Hu *et al.* presents GaussianAvatar [20] to create animatable human avatars represented by 3D Gaussians [18]. The method initializes 3D Gaussians on the canonical SMPL surface [21] and assigns each Gaussian a static skinning weight by interpolating predefined SMPL skinning weights for pose deformation. The authors then use a dynamic appearance network in the 2D UV space of the posed SMPL to extract pose-dependent appearance features. They use multi-layer perceptrons (MLP) to predict the properties of each canonical 3D Gaussian, including position offset, scale, and color. The position offsets optimize Gaussians from the initial SMPL surface to the clothed surface. These offset canonical 3D Gaussians are then deformed to the posed space by linear blend skinning (LBS) [21] for free-view rendering.

Modeling human hands presents distinct challenges. Though GaussianAvatar achieves promising results for human avatar rendering, directly applying it to hand avatar faces two major problems. 1) GaussianAvatar does not enforce geometric consistency between the same body part across different poses. Hand geometry shows both pose-dependent variations and pose-independent consistency. The pose-dependent geometric features like hand wrinkle depressions and blood vessel bulges dynamically deform with various poses, leading to unsmooth surfaces and thus affecting hand appearance. On the other hand, certain aspects of hand geometry exhibit a pose-independent consistency, such as the positions of wrinkles and vessels, the shapes of fingers and nails, and so on. Unfortunately, though GaussianAvatar achieves pose-dependent geometric modeling, it does not account for pose-independent consistency, which is crucial for achieving realistic and stable hand rendering. 2) GaussianAvatar lacks accuracy for pose deformation. GaussianAvatar uses LBS for deformation. The LBS skinning weights assume that the unposed canonical Gaussians lie on the SMPL surface. Nevertheless, during optimization, the unposed Gaussians offset from the SMPL surface. This offset results in a misalignment between Gaussians and skinning weights, leading to less accurate pose deformation and reducing the fidelity of posed rendering results.

In this paper, we propose GaussianHand, the first real-time 3D Gaussian rendering approach for both efficient free-view and free-pose hand avatar animation from sparse view images, achieving realistic and consistent hand appearance rendering and accurate hand pose deformation. Our method involves two key components. First, we introduce the novel Hand Gaussian Blend Shapes (HGBS) to attain position offsets that not only accommodate pose-dependent geometric variations but also ensure geometry consistency of hand across poses. Our HGBS consists of pose-independent Gaussian shape basis and

Lizhi Zhao, Runze Fan, Lili Wang are with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with the Peng Cheng Laboratory, Shenzhen, Guangdong 518000, China.
E-mail: {lizhizhao, by2106131, wanglily}@buaa.edu.cn.

Xuequan Lu is with the Department of Computer Science and IT, La Trobe University, Australia. E-mail: b.lu@latrobe.edu.au.

Sio Kei Im is with Macao Polytechnic University, Macao, China, 999078.

Lili Wang is the corresponding author.

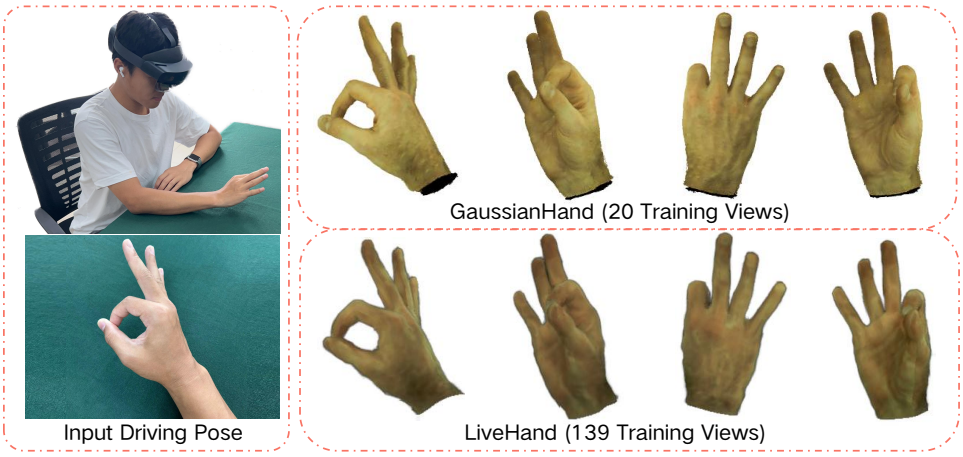

Fig. 1. Left: Given an RGB image of the user input casual driving pose out of the training set, our GaussianHand can animate the trained hand avatar with the input pose, while keeping the hand avatar's original realistic appearance details. Our GaussianHand can be rendered in 125 FPS with free-view and free-pose capabilities. Right: Comparison of our rendering results under 20 training views with LiveHand [1] under 139 training views. Our method achieves more realistic and natural rendering quality with significantly fewer training views, especially for the clear appearance of wrinkles, nails, and blood vessels.

pose-dependent blend coefficients. Linearly blending the two yields the linearly related position offset across poses. Second, we propose the novel Neural Residual Skeleton (NRS) and Residual Skinning Weights (RSW) to rectify the inaccurate pose deformation. We devise a skeleton regressor to regress the NRS from the HGBS position offset. The NRS, along with the RSW, acts as a residual term for LBS, rectifying the predefined skinning weights to the offset Gaussians surface.

We compare our GaussianHand with state-of-the-art (SOTA) methods HumanNeRF [22], HandAvatar [15], LiveHand [1] on the InterHand2.6M dataset [23], and HARP [10] on the Hand Appearance Dataset. The rendering results of our method are closer to the ground truth (GT) in both datasets. In the case of multi-view training, compared to the LiveHand of the highest rendering quality (which employs 139 training views) for the average results of 3 evaluation sequences, our method reduces 9.6% in Learned Perceptual Image Patch Similarity (LPIPS) [24] and improves 2.7%, 6.0% in Peak Signal-to-Noise Ratio (PSNR) [25] and Structural Similarity Index Measure (SSIM) [26], respectively, with only 20 training views. When reducing the training views of our method to 5, our PSNR and SSIM are still 1.8%, 5.7% higher than the LiveHand using 139 training views. Figure 1 shows a comparison of the rendering results for a user input driving pose out of the training set between our method and the recent SOTA LiveHand method. This improvement is especially notable in the detailed and clear appearance of wrinkles, nails, and blood vessels. In monocular view training, our GaussianHand outperforms HARP [10] by 78.1%, $29,4\%$, and 5.5% for LPIPS, PSNR, and SSIM, respectively. In terms of efficiency, the rendering speed of our GaussianHand is about 125 frames per second (FPS), which is 25 FPS faster than the LiveHand.

To summarize, our contributions are as follows:

- We introduce GaussianHand, a new real-time 3D Gaussian rendering technique for both free-pose and free-view hand avatar animation from sparse view images. It achieves realistic hand detail rendering and accurate hand pose deformation, especially for nails, veins, and wrinkles.
- We design the novel Hand Gaussian Blend Shapes, including the pose-independent Gaussian shape basis and pose-dependent blend coefficients, to capture the hand geometric features and ensure consistency across different hand poses.
- We propose the novel Neural Residual Skeleton with Residual Skinning Weights to rectify inaccurate LBS pose deformation due to Gaussian position offset.

II. Related Work

In this section, we introduce hand mesh reconstruction, animatable hand avatar animation, and Gaussian-based human avatars related to our approach. For a comprehensive understanding of neural radiance field and Gaussian splatting, we direct readers to recent surveys [27]–[29].

A. 3D Hand Mesh Reconstruction

3D hand mesh reconstruction task aims to represent the hand using a mesh model and reconstruct it from RGB images. Earlier parametric-based approaches predominantly utilize the MANO model [30], regressing its parameters to fit the input images [31]–[33], [33], [34]. Alternatively, non-parametric methods aim to directly regress the 3D hand vertices from input images without any parametric model [12], [35]–[39]. Lin *et al.* employ transformers to model interactions between hand vertices and joints, facilitating the regression of 3D joints and mesh vertices [35], [37]. Chen *et al.* introduce a lightweight framework with an efficient 2D encoding and 3D decoding structure, achieving up to 83 FPS [38]. Additionally, Xu *et al.* present H2ONet [40], designed to reconstruct hand meshes in the presence of occlusions. Previous methods primarily focus on geometry, often overlooking the preservation of realistic textures. Our method not only reconstructs the hand's geometry but also its photorealistic textured appearance.

B. Animatable Hand Avatar

The neural radiance field [14], [41]–[44] has significantly advanced novel view synthesis and 3D scene reconstruction by modeling the geometry and color properties of any 3D query point from a tracing ray. NeRF representation has facilitated the creation of animatable hand avatars with realistic textures [1], [15]–[17]. Chen *et al.* [15] introduce the HandAvatar framework, a pioneering NeRF-based neural hand rendering method. This framework utilizes a high-resolution MANO model and a local-pair occupancy field to capture personalized hand geometry and integrates a self-occluded illumination field to simulate shadows. Despite its high rendering quality, HandAvatar demands substantial computational resources. Mundra *et al.* [1] develop LiveHand, an efficient neural hand rendering method. Their approach involves a novel mesh-guided sampling strategy for efficient query point sampling near the approximate hand surface and a super-resolution module to minimize the number of rays queried. Consequently, LiveHand achieves a real-time rendering speed of 45 FPS with impressive quality. However, this super-resolution strategy, while enhancing speed, reduces rendering quality, demonstrating the typical trade-off between speed and quality.

While NeRF-based methods provide high-quality hand rendering, they encounter two main challenges: the heavy computational load from implicit representation and ray tracing that hinder rendering speed, and the inverse skinning paradigm for animation. This paradigm deforms query points from posed space to canonical space, causing ambiguous correspondences [45]. Our GaussianHand incorporates Gaussian splatting with explicit representation and forward skinning deformation for animatable hand avatar rendering, which is more efficient and capable of achieving approximately 125 FPS for real-time rendering with enhanced quality.

Furthermore, several studies [10], [11], [46]–[49] concentrate on creating relightable hand avatars that offer controllable lighting effects by embedding illumination information. HARP [10] and UHM [47] reconstruct hand meshes and UV textures, rendering relightable hands with the Phong reflection model. XHand [49] predicts pose-dependent mesh displacements to refine the template hand and renders using the Lambertian reflectance model. However, it overlooks pose-independent geometric consistency. BiTT [48] reconstructs relightable interacting hands from a single RGB image by leveraging the symmetry information of two hands.

C. Gaussian-based Human Avatar

NeRF-based methods are hindered by implicit representation and slow rendering speeds. To overcome this, researchers have adopted explicit 3D Gaussian representations for clothed human avatars [18], [20], [50]–[56]. Lei *et al.* [53] introduce GART, an explicit representation model for non-rigid articulated avatars. GART employs a forward skinning model to capture human and cloth deformations effectively. Li *et al.* [51] develop Animatable Gaussians, which derive 3D Gaussian properties from a 2D StyleGAN-based generator for clothed human avatars. They also propose parameterizing these 3D Gaussians on the avatars' front and back, enhancing multi-view rendering capabilities. Hu *et al.* [20] present GaussianAvatar, which initializes Gaussians on the SMPL surface and introduces a dynamic appearance network to estimate pose-dependent Gaussian properties. They further implement a joint optimization of motion and appearance in avatar modeling to ensure precise motion portrayal. Shao *et al.* [55] propose SplattingAvatar, which disentangles the motion and appearance of a virtual human with explicit mesh geometry and implicit appearance modeling. By embedding the Gaussians on a triangle mesh, SplattingAvatar achieves a free-pose animation effect.

Previous Gaussian-based avatar methods primarily focus on full-body human avatars, without specifically considering the unique features of hands. Our GaussianHand is the first study that introduces 3D Gaussian splatting for effectively rendering hand avatars with realistic detailed nails, veins, and wrinkles appearance while achieving fast real-time rendering speed, marking a leap in the field of animatable hand avatar rendering.

III. Method

In this paper, we introduce the GaussianHand, a new real-time 3D Gaussian rendering pipeline for hand avatar animation. Figure 2 shows the overview of our method.

Given a sequence of sparse view images of identity's moving hand $\{\mathbf{I}_f^v \mid v = 1 \dots V, f = 1 \dots F\}$ captured from F frames and V viewpoints, along with the corresponding coarse estimated parametric hand meshes $\{M(\theta_f, \beta)\}_{f=1 \dots F}$, where M denotes the parametric hand model [30], β denotes the global shape parameters and θ_f is the f-frame pose parameters, our goal is to create a realistic animatable hand avatar capable of free-view and free-pose real-time rendering.

First, we represent the hand avatar with 3D Gaussians and estimate the animatable hand Gaussians properties through the network (Sec. III-A). Second, we present the Hand Gaussian Blend Shapes to offset the initialized Gaussians' position for modeling the unsmoothed hand surface, thus reserving realistic hand appearance details (e.g., nails and wrinkles), while also ensuring pose-independent consistent hand geometric features (Sec. III-B). Third, we propose the Neural Residual Skeleton and Residual Skinning Weights to rectify LBS deformation for posing the canonical Gaussians (Sec. III-C). Finally, we train our network with the rendered hand Gaussians supervised by the input ground truth images (Sec. III-D).

We formulate our GaussianHand as:

$$\hat{\mathbf{I}}(\beta, \theta, \mathcal{G}_c^{bs}, A) = Splat(\widehat{LBS}(\mathcal{G}_c^{bs}, \mathcal{W}, \mathcal{B}(J, \theta, \beta), \hat{\mathcal{W}}, \mathcal{B}(\hat{J}, \theta)), A), \tag{1}$$

where $\hat{\mathbf{I}}(\cdot)$ denotes the rendered posed hand image, and β, θ denote the shape and pose parameters. We denote the initialized canonical hand Gaussians as $\mathcal{G}_c$. Our HGBS optimizes the position of $\mathcal{G}_c$ to achieve $\mathcal{G}_c^{bs}$ for reversing detailed appearance. A denotes the color and scale properties of the Gaussians. $\mathcal{G}_c^{bs}$ are deformed to the posed space through our proposed $\widehat{LBS}$ with correction term, which takes the predefined MANO skeleton J with skinning weights $\mathcal{W}$, as well as our NSR $\hat{J}$ with RSW $\hat{\mathcal{W}}$. The rigid kinematic chain transformation [30]

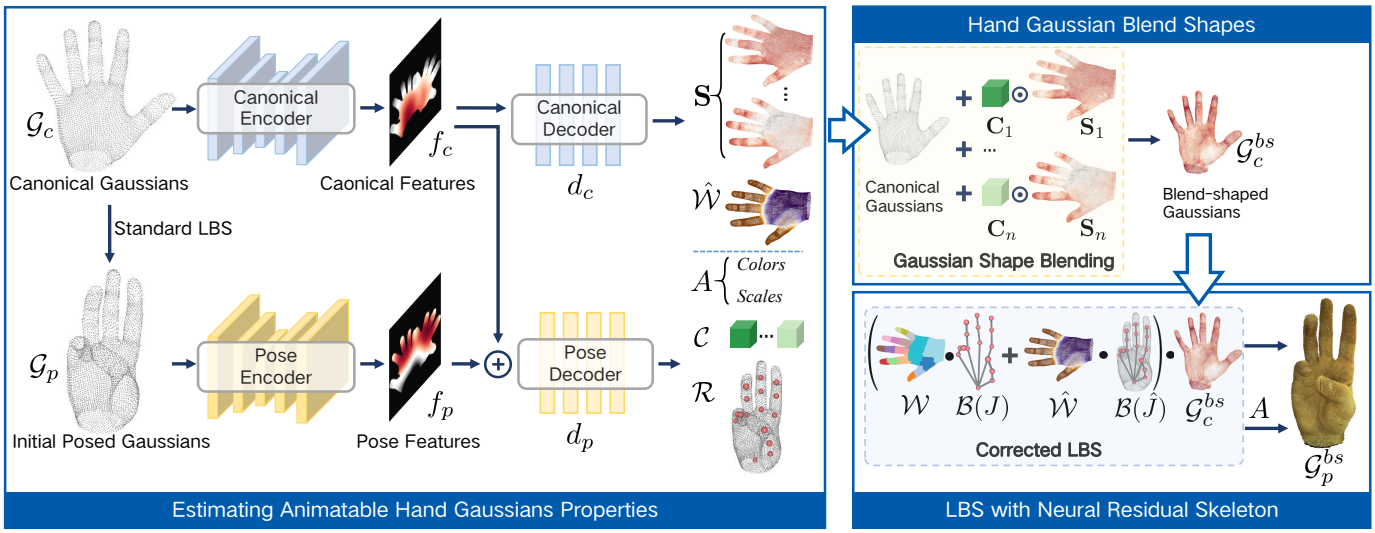

Fig. 2. Overview of our GaussianHand. Left: We initialize unposed canonical Gaussians $\mathcal{G}_c$ on the MANO surface. Through the standard LBS, we can achieve the initial posed Gaussians $\mathcal{G}_p$. The Gaussians color, scale properties, as well as other necessary properties of our HGBS and the NRS and RSW, are estimated by the network. Upper Right: We propose HGBS to model the non-smoothed hand surface. The HGBS optimizes the geometry of $\mathcal{G}_c$ to achieve $\mathcal{G}_c^{bs}$ by applying the position offsets, which are composed by the learned Gaussian shape basis $\mathcal{S}$ and the blend coefficients $\mathcal{C}$. Lower Right: The estimated NRS $\hat{J}$ (calculated from the learned skeleton regressor $\mathcal{R}$) and Residual Skinning Weights $\hat{\mathcal{W}}$ act as a residual term to rectify the LBS, which deforms $\mathcal{G}_c^{bs}$ to attain the final posed blend-shaped Gaussians $\mathcal{G}_p^{bs}$ for splatting rendering.

$\mathcal{B}$ outputs a set of bone transformations. The 3D Gaussian splatting $Splat(\cdot)$ renders posed Gaussians as an output image.

A. Estimating Animatable Hand Gaussians Properties

3D Gaussian Splatting [18] is a scene representation that allows real-time free-view photo-realistic rendering. Each 3D Gaussian is defined by the 3D position $\mathbf{x} \in \mathbb{R}^3$, quaternion rotation $\mathbf{q} \in \mathbb{R}^4$, 3D scaling $\sigma \in \mathbb{R}^3$, opacity $\alpha \in \mathbb{R}$, and color $\tau \in \mathbb{R}^3$ properties. The 3D Gaussians are initialized on SfM [57] points and optimized to fit the input images through the differentiable splatting rendering process. Readers are referred to [18], [27], [29] for more details.

We first initialize canonical Gaussians $\mathcal{G}_c$ on the registered canonical MANO [30] surface and map it as a UV position map $I_c \in \mathbb{R}^{H\times W\times 3}$ on the 2D hand manifold $\mathcal{S}^2$ for extracting canonical features $f_c \in \mathbb{R}^C$ through a 2D UNet encoder [58] $E_c : \mathcal{S}^2 \in \mathbb{R}^3 \rightarrow \mathbb{R}^C$.

Given input shape and pose parameters β, θ, we attain the initial posed Gaussians $\mathcal{G}_p$ by the standard LBS:

$$\mathcal{G}_p = LBS(\mathcal{G}_c, \mathcal{B}(J, \beta, \theta), \mathcal{W}). \tag{2}$$

Specifically, the predefined skeleton J with n_b joints returns n_b bone transformations through rigid kinematic chain transformation $\mathcal{B}$ as:

$$\mathcal{B}(J, \beta, \theta) = [B_1, B_2, \ldots, B_{n_b}], \tag{3}$$

where $B_i \in SE(3)$ denotes transforming the coordinate frame of i-th canonical joint to the posed coordinate frame [53]. With these transformations, $\mathcal{G}_c$ are deformed to the posed space by:

$$x_p^i = \left(\sum_{k=1}^{n_b} \mathcal{W}_k^i B_k\right) x_c^i, \tag{4}$$

where x_c^i, x_p^i denotes the position property of the i-th Gaussian of $\mathcal{G}_c$ and $\mathcal{G}_p$ respectively, and $\mathcal{W}_k^i \in \mathbb{R}$ denotes the skinning weights of the k-th joint at position x_c^i. $\mathcal{W}$ is initialized by interpolating MANO's predefined skinning weights. The pose features $f_p \in \mathbb{R}^C$ of $\mathcal{G}_p$ are extracted through the pose encoder E_p, which shares the same structure with E_c.

We apply two MLP decoders to estimate the Gaussian properties. The *pose-independent* canonical decoder d_c takes input f_c and predicts the Gaussian shape basis $\mathcal{S}$ and the Residual Skinning Weights $\hat{\mathcal{W}}$ as:

$$\mathcal{S}, \hat{\mathcal{W}} = d_c(f_c). \tag{5}$$

The *pose-dependent* decoder d_p takes concatenated $[f_c, f_p]$ as input and predicts the blend coefficients $\mathcal{C}$, the residual skeleton regressor $\mathcal{R}$, and colors, scales properties A as:

$$A, \mathcal{C}, \mathcal{R} = d_p(cat[f_c, f_p]), \tag{6}$$

where $[\cdot, \cdot]$ denotes the concatenation operation.

Our HGBS (Sec. III-B) optimizes the position offset of $\mathcal{G}_c$ to achieve $\mathcal{G}_c^{bs}$, and our NRS (Sec. III-C) rectifies the LBS deformation to attain the final posed blend-shaped Gaussians $\mathcal{G}_p^{bs}$ for splatting rendering [18].

B. Hand Gaussian Blend Shapes

Our Hand Gaussian Blend Shapes aims to offset the initialized canonical Gaussians $\mathcal{G}_c$ on the smooth MANO surface to the realistic hand surface that reverses geometric details, such as the palm wrinkles, the protruding veins, and the nails. The HGBS should also maintain shape consistency across various hand poses, such as the finger length and thickness, nail shape, the position of veins, etc.

To achieve the above goals, we propose a two-factor representation, including the pose-independent Gaussian shape

basis and pose-dependent blend coefficients. The Gaussian shape basis is a set of point offsets that are learned to model the common geometric features of hand appearance that are shared across different poses. The blend coefficients are learned to express pose-dependent features, such as the depth of the wrinkles that varies with poses. These position offsets for the Gaussians are derived by linearly blending the Gaussian shape basis with the blend coefficients. This linear combination provides an efficient and consistent method for depicting the hand's appearance in terms of its geometric structure.

Pose-independent Gaussian Shape Basis. Given the canonical features f_c extracted from the initialized canonical Gaussians $\mathcal{G}_c$, we predict the pose-independent Gaussian shape basis $\mathcal{S}$ through the canonical decoder d_c, where $\mathcal{S} = [\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_n]$ denotes the n Gaussian shape basis, and each shape basis $\mathbf{S}_i \in \mathbb{R}^{N\times 3}$ represents the position offsets of N Gaussians.

Pose-dependent Blend Coefficients. We learn the pose-dependent blend coefficients $\mathcal{C}$ through the pose decoder d_p from $[f_c, f_p]$, where $\mathcal{C} = [\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_n]$ represents the weights of n Gaussian shape basis. $\mathbf{C}_i \in \mathbb{R}^{N\times 3}$ denotes the point-wise blend coefficients.

Canonical Gaussian Position Offset. The Gaussian position offsets Δ_{bs} are obtained by linearly blending the Gaussian shape basis $\mathcal{S}$ with the blend coefficients $\mathcal{C}$ as:

$$\Delta_{bs} = \sum_{i=1}^{n} \mathbf{C}_i \odot \mathbf{S}_i, \tag{7}$$

where $\odot$ denotes the Hadamard element-wise product. By applying the Gaussian position offsets Δ_{bs} to the canonical Gaussians $\mathcal{G}_c$ as:

$$\hat{x}_c = x_c + \Delta_{bs}, \tag{8}$$

where $\hat{x}_c$ denotes the position of $\mathcal{G}_c^{bs}$, we make Gaussians shift from MANO surface for reversing the realistic hand geometric surface.

C. LBS with Neural Residual Skeleton

Given the blend-shaped Gaussians $\mathcal{G}_c^{bs}$ in the canonical space, we aim to deform them to the posed space corresponding to the input pose parameter θ through LBS for free-pose rendering. Typically, MANO predefines $\mathcal{W}$ for LBS, assuming that vertices are located on the original MANO surface. In contrast, our $\mathcal{G}_c^{bs}$ deviates from this surface to preserve realistic geometric features by HGBS. Consequently, simply applying standard LBS to $\mathcal{G}_c^{bs}$ results in discrepancies. We aim to compensate for the deformation discrepancy by introducing the Neural Residual Skeleton and Residual Skinning Weights as a residual term to the standard LBS, ensuring faithful rendering in various poses.

Skeleton Regressor. Since the LBS discrepancies are caused by Δ_{bs}, we propose to learn a skeleton regressor $\mathcal{R} \in \mathbb{R}^{n_b\times N}$ to regress the NRS $\hat{J} \in \mathbb{R}^{n_b\times 3}$ from Δ_{bs} as:

$$\hat{J} = \mathcal{R} \cdot \Delta_{bs}. \tag{9}$$

We define NRS consists of n_b joints and share the same articulation structure with MANO. Given the pose parameter θ, the rigid kinematic chain transformation $\mathbf{B}$ outputs n_b bone transformations of $\hat{J}$:

$$\mathcal{B}(\hat{J}, \theta) = [\hat{B}_1, \hat{B}_2, \ldots, \hat{B}_{n_b}] \tag{10}$$

Corrected LBS. We define the $\widehat{LBS}$ to achieve the posed hand Gaussian $\mathcal{G}_p^{bs}$ from $\mathcal{G}_c^{bs}$. Specifically, we deform $\hat{x}_c^i$, the position property of the i-th Gaussian of $\mathcal{G}_c^{bs}$, as follows:

$$\hat{x}_p^i = \left(\sum_{k=1}^{n_b} \left(\mathcal{W}_k^i B_k + \hat{\mathcal{W}}_k^i \hat{B}_k \right) \right) \hat{x}_c^i, \tag{11}$$

where $\hat{\mathcal{W}}$ denotes the RSW learned from the canonical decoder d_c, and $\hat{\mathcal{W}}_k^i \in \mathbb{R}$ denotes the skinning weight of the k-th neural residual joint at position $\hat{x}_c^i$. $\hat{B}_k$ represents the k-th residual bone transformation.

D. Training Strategy

We employ a two-stage optimization process to train our network. Given that the input ground truth pose parameters θ are initially estimated coarsely and typically inaccurate, following GaussianAvatar, in the first stage we optimize θ through gradient descent with the network. In the second stage, θ is fixed. We adopt the following loss functions to supervise the training process following [20]:

$$\begin{aligned} \mathcal{L} = \lambda_{rbg}\mathcal{L}_{rbg} + \lambda_{ssim}\mathcal{L}_{ssim} + \lambda_{lpips}\mathcal{L}_{lpips} \\ + \lambda_{offset}\mathcal{L}_{offset} + \lambda_{scale}\mathcal{L}_{scale}, \end{aligned} \tag{12}$$

where $\mathcal{L}_{rbg}$, $\mathcal{L}_{ssim}$, $\mathcal{L}_{lpips}$ are the L_1 loss, SSIM loss, and LPIPS loss between the Gaussian splatting rendered image and the ground truth image, respectively. $\mathcal{L}_{offset}$ and $\mathcal{L}_{scale}$ denote the L_2 regularity term of the predicted position offset Δ_{bs} and scale property of the Gaussians. We set $\lambda_{scale} = 0$ in the second stage.

IV. Experiments

A. Implementation Details

Following Hu et al. [20], we employ the U-Net architecture [58] as both the canonical and pose encoder. The pose decoder comprises 4-layer MLPs with 256 hidden units, followed by four 3-layer MLPs dedicated to predicting blend coefficients, the skeleton regressor, and the properties of Gaussian scales and colors. The canonical decoder is constructed using two 4-layer MLPs, each with 256 hidden units. We maintain fixed Gaussian rotations and opacity settings. We initialize $N = 87,779$ Gaussians on the MANO surface and set the number of Gaussian shape basis as $n = 8$. Our NRS $\hat{J}$ consists of $n_b = 16$ joints.

The loss function weights are set as follows: $\lambda_{rgb} = 0.8, \lambda_{ssim} = 0.2, \lambda_{lpips} = 0.2, \lambda_{scale} = 1.0, \lambda_{offset} = 10$. Additionally, we incorporate an optimizable feature tensor to capture a coarse appearance of the model, consistent with [20]. Experiments are trained on a single NVIDIA RTX 4090 GPU. The first stage of our training takes approximately 1.5 hours for $41,600$ iterations, and the second stage requires 2.5 hours for $83,200$ iterations. In comparison, HandAvatar requires about 8 hours, and LiveHand requires about 7 hours.

Following HandAvatar [15], we evaluate the rendering quality using several numerical metrics, including PSNR, LPIPS, and SSIM. We also report the rendering time in terms of FPS on the NVIDIA RTX 4090 GPU.

B. Datasets

InterHand2.6M. InterHand2.6M [23] is a large-scale RGB-based 3D hand pose dataset comprising videos of hands in various poses captured from multi-view cameras. The dataset provides ground truth hand images with foreground masks, camera extrinsic and intrinsic, and coarse MANO parameters for each frame. Consistent with HandAvatar [15], we selected three sequences from InterHand2.6M: *test/Capture0*, *test/Capture1*, and *val/Capture0*. Each sequence contains a training split (named *ROM04_RT_Occlusion*) and a test split (named *ROM03_RT_No_Occlusion*) with different hand poses from 139 camera viewpoints. While HandAvatar utilized all 139 viewpoints in both splits, we adopted a more challenging setting by selecting only 5 and 20 viewpoints from the training split, while maintaining the 139 viewpoints in the validation split. This approach heightened the challenge as our test set encompasses not only novel poses but also novel viewpoints. The comparison baselines, which utilize the entire training set, are evaluated solely on novel poses.

Hand Appearance. Karunratanakul *et al.* [10] publish the Hand Appearance dataset containing a right hand in normal office lighting captured by a monocular phone camera to simulate end-users casual habits. The dataset provides hand RGB images with ground truth masks, coarse-fitted MANO parameters, and camera settings. Following UHM [47], among 9 sub-sequences of *subject_1*, 1 to 5 are used for the training, and 6 to 9 are used for the testing.

C. Comparison Study on InterHand2.6M

We compare our GaussianHand with recent state-of-the-art animatable hand avatar methods with publicly available code, including HandAvatar [15] and LiveHand [1].

Training Viewpoints. For a fair comparison, we conduct experiments on the InterHand2.6M dataset with the same training and testing split as HandAvatar [15]. We re-train LiveHand [1] on the same setting based on their public code. Note that experiments conducted by HandAvatar and LiveHand use 139 viewpoints for both training and testing, focusing on a novel-pose evaluation. In contrast, our GaussianHand uses only 5 and 20 viewpoints for training and 139 viewpoints for testing, addressing both novel-pose and novel-view evaluations, which is more challenging.

Quantitative Results. The quantitative rendering quality comparison between our GaussianHand and prior methods on the InterHand2.6M is shown in Table I. Our GaussianHand demonstrates the best rendering quality on all three sequences with only 20 training views, with the PSNR reaching 32.31, 30.79, and 32.13, respectively. These scores surpass those of the previous state-of-the-art method, LiveHand, by margins of 0.51, 0.74, and 1.29, and exceed the scores of HandAvatar by 4.08, 4.23, and 4.08, respectively. Even under the more extreme condition of only 5 sparse training views, our GaussianHand maintains competitive results compared to other methods, demonstrating its consistency and robustness to novel views. Additionally, the proposed GaussianHand achieves the highest rendering speed of 125 FPS, which is 25 FPS higher than the real-time method LiveHand.

LiveHand uses a super-resolution module to enhance rendering speed and achieves promising results of 100 FPS. However, this super-resolution strategy trades off speed for quality and does not improve neural rendering quality. In contrast, our GaussianHand introduces a distinctly different Gaussian splatting rendering pipeline with explicit representation and forward skinning deformation. Our HGBS effectively reverses the hand appearance details in geometry, and our NRS allows accurate pose deformation. Together, they significantly elevate the rendering quality, demonstrating the superiority in both efficiency and quality.

Visualization Results. The numerical improvements in the metrics are also reflected by the visualization results, as shown in Fig. 3. We visualize our GaussianHand's rendering results under 5 and 20 training views and compare them with HandAvatar and LiveHand, alongside the ground truth images for reference. We summarize the qualitative improvements of our GaussianHand as follows. 1) Nails: As shown in the first three rows, our GaussianHand can render high-fidelity fingernails, while HandAvatar's nails lack realistic colors and shapes, and LiveHand's nails appear blurry. Our GaussianHand under 5 views can also render nails effectively. 2) Blood Vessels: The fourth row shows that GaussianHand can render the blood vessels bulging on the back of the hand. In contrast, LiveHand tends to blur the thread-like vessel shapes, and HandAvatar only renders the obvious veins in the lower right, ignoring the more subtle ones in the upper left. 3) Shape consistency. The fifth and sixth rows show that GaussianHand can maintain the consistency of finger shapes across different poses. HandAvatar tends to render the fingers to be overly rounded, and LiveHand's fingers tend to have unnatural bumps at the knuckles. The first three rows also demonstrate the nail shapes are consistent across different poses. 4) Wrinkles: The last row and the second row show that our GaussianHand can render realistic wrinkles on the palm. HandAvatar also outputs clear wrinkles, but it lacks a realistic shadow effect. LiveHand outputs blurry wrinkles.

These improvements can be explained as follows: 1) Our HGBS can capture the geometric features shared across different poses from the input RGB images, providing accurate canonical position offsets. Our NRS ensures the offsets are correctly deformed to the posed space. Accurate reconstruction of geometry for nails, veins, and wrinkles, is crucial for clear and detailed splatting rendering. 2) The consistency of shapes across different poses, is maintained by the linearly correlated geometric offsets. While the hand geometry can deform with different poses, all deformations in the full pose space adhere to the same pose-independent Gaussian shape basis and thus restrict outliers of shape deformed.

We also retrain HandAvatar and LiveHand using the same 20 training views of our GaussianHand. The visualization results, presented in Figure 4, illustrate that LiveHand struggles with sparse viewpoints. Its rendering results appear more

TABLE I
RENDERING QUALITY COMPARISON AMONG OUR GAUSSIANHAND AND PRIOR METHODS ON THE INTERHAND2.6M DATASET. NOTE THAT PRIOR METHODS ARE TRAINED FROM 139 VIEWPOINTS, WHILE OUR GAUSSIANHAND IS TRAINED FROM ONLY 5 AND 20 VIEWPOINTS.

Method	*test/Capture0*			*test/Capture1*			*val/Capture0*			FPS
	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	
HumanNeRF [22]	0.115	27.64	0.884	0.118	26.31	0.880	0.119	27.80	0.882	-
HandAvatar [15]	0.104	28.23	0.894	0.108	26.56	0.890	0.106	28.04	0.890	0.38
LiveHand [1]	0.030	31.80	0.907	0.033	30.05	0.899	0.031	30.83	0.923	100
GaussianHand (5 views)	0.031	32.07	0.964	0.033	30.64	0.960	0.032	31.65	0.961	125
GaussianHand (20 views)	**0.026**	**32.31**	**0.967**	**0.030**	**30.79**	**0.965**	**0.029**	**32.12**	**0.962**	**125**

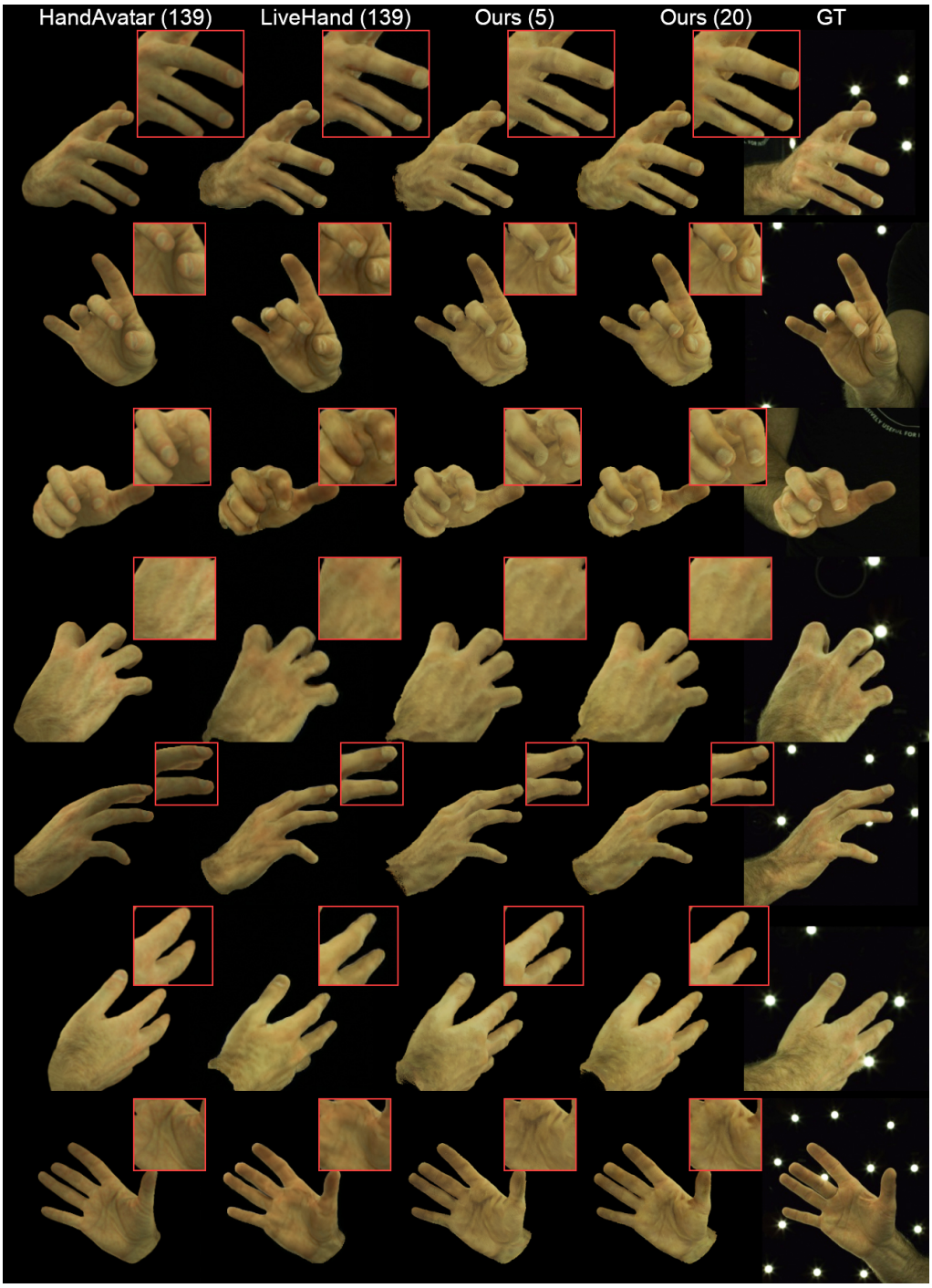

Fig. 3. Visualization results of HandAvatar [15], LiveHand [1], and our GaussianHand under 5 and 20 training views for novel-pose animation on the InterHand2.6M dataset. We label the number of training views in brackets. The exposure of the entire figure is increased for clearer visualization.

blurry around the edges and lack clear texture details. This issue likely stems from its reliance on an implicit representation combined with a super-resolution module, which complicates the training convergence. HandAvatar's wrinkles and veins tend to be overly smooth, lacking realistic bumps or depression effect. This is a consequence of its texture map-based approach, which does not incorporate geometric optimization for modeling the unsmoothed hand surface. Our GaussianHand, on the contrary, consistently maintains high-quality, detailed textures and achieves the best rendering results over the compared methods.

We report the numerical results for the 20 training views experiment. The LPIPS, PSNR, and SSIM of HandAvatar are 0.069, 27.84, and 0.9, respectively. LiveHand achieves 0.031, 30.49, and 0.87, respectively. Our method demonstrates superior rendering quality with 0.026, 32.31, and 0.967.

We argue that two principal reasons contribute to the superior rendering results under the sparse view conditions of our method. Firstly, our GaussianHand incorporates *view-independent* and explicit geometry, and the introduction of HGBS and NRS further improves the geometric accuracy. Secondly, our network directly predicts the Gaussian color property instead of relying on view-dependent spherical harmonic bases for querying color. In contrast, NeRF-based methods employ implicit representations, and the sample points and harmonic bases are heavily dependent on *view-dependent* ray tracing. Consequently, NeRF-based methods typically exhibit slower convergence and require more views to achieve multi-view consistency. To demonstrate the reliance of NeRF-based methods on view-dependence, we disable the view-dependence strategy of LiveHand and train it on the same 20 training views. Experiment shows that LiveHand without view-dependence fails to converge effectively, leading to enormous decreases in LPIPS, PSNR, and SSIM by 0.171, 8.67, and 0.514, compared to LiveHand with view-dependence.

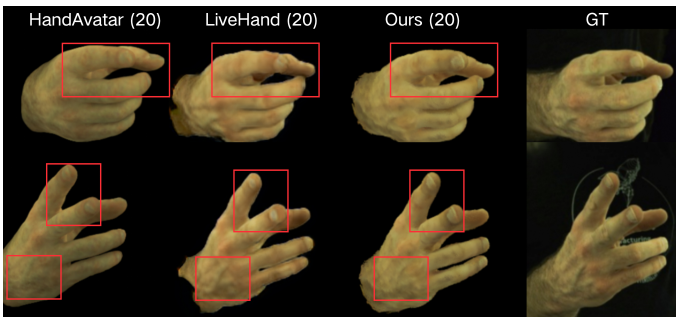

Fig. 4. Visualization results of 20 views training for HandAvatar [15], LiveHand [1] and our GaussianHand. The number of training views is labeled in brackets. The figure exposure is increased.

D. Comparison Study on Hand Appearance

We compared our GaussianHand model with HARP [10] and UHM [47] on the Hand Appearance dataset. Both HARP and UHM fine-tune their models for hand poses, lighting, and shadows on the test set. For a fair comparison, we fine-tune hand poses and the pose encoder E_p on the test set while keeping all other network parameters freezed. The quantitative results, presented in Table II, indicate that the proposed GaussianHand outperforms HARP by margins of 0.036, 5.04, and 0.18 for LPIPS, PSNR, and SSIM metrics, respectively. In comparison to UHM, our method exceeds performance by 0.01 and 0.018 in LPIPS and SSIM while achieving comparable results in PSNR.

Qualitative results, as shown in Fig. 5, reveal that our GaussianHand renders hand appearances with greater fidelity than HARP and UHM. Specifically, the first row shows that our method renders the details of tendons and blood vessels on the back of the hand, whereas HARP and UHM produce blurry results. The second row illustrates that GaussianHand effectively captures pose-dependent variations in appearance. Notably, the tendons on the back of the hand become more pronounced in a spread pose compared to the grasp pose shown in the first row. In contrast, the tendons in HARP and UHM remain blurry, similar to those in the first row. In the third row, we observe that our method renders clear and natural wrist wrinkles, while HARP and UHM show overly concave tendons. Finally, the last row confirms that our model produces fingernails with higher fidelity than both HARP and UHM.

TABLE II
QUANTITATIVE EVALUATION OF THE APPEARANCE RENDERING TASK ON THE HAND APPEARANCE DATASET.

Method	LPIPS↓	PSNR↑	SSIM↑
HARP [10]	0.081	27.50	0.947
UHM [47]	0.055	**32.55**	0.957
GaussianHand (ours)	**0.045**	32.54	**0.965**

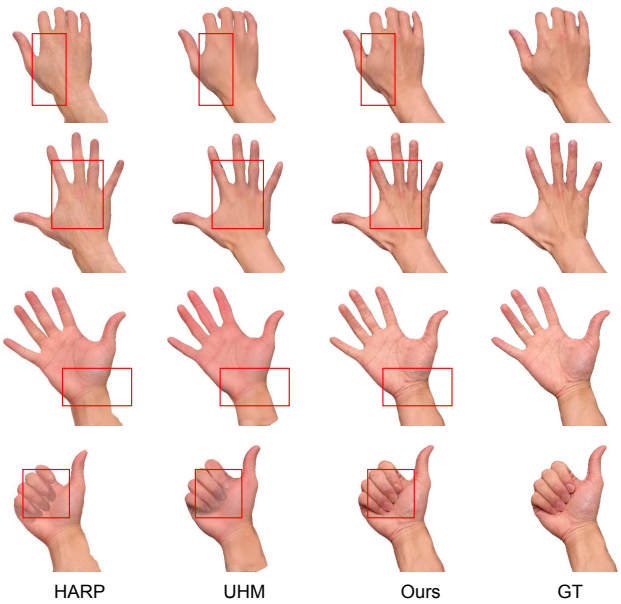

Fig. 5. Visualization results of HARP [10], UHM [47] and our GaussianHand on the Hand Appearance dataset.

E. Ablation Study

We conduct ablation studies using the InterHand2.6M dataset under 5 training views to evaluate the effectiveness

TABLE III
ABLATION STUDY FOR MODEL COMPONENTS ON THE INTERHAND2.6M DATASET UNDER 5 TRAINING VIEWS.

Method	*test/Capture0*			*test/Capture1*			*val/Capture0*		
	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑
Baseline	0.032	31.60	0.962	0.0362	29.63	0.956	0.035	31.08	0.959
Baseline+HGBS	0.031	31.90	0.963	0.033	30.53	**0.961**	0.032	31.61	0.961
Baseline+HGBS+NRS	**0.031**	**32.07**	**0.964**	**0.033**	**30.64**	0.960	**0.032**	**31.65**	**0.961**

of our proposed components, including HGBS and NRS. First, we apply the GaussianAvatar [20] as the baseline by replacing the SMPL model with the MANO model. The baseline predicts each canonical Gaussian's position, scale, and color properties directly through the MLP decoder. Next, we integrate our proposed HGBS module to the baseline (Baseline+HGBS), predicting the Gaussian shape basis and blend coefficients to refine the geometry of canonical Gaussians, instead of directly predicting the position offsets as the baseline. This setting still uses the standard LBS for pose deformation. Finally, we incorporate the proposed NRS to form the complete GaussianHand (Baseline+HGBS+NRS). The NRS specifically rectifies the LBS deformation of position offsets. Numerical results are presented in Table III, and qualitative results are visualized in Fig. 6.

Experimental results of three module combinations are shown in Table III, Specifically, compared to the Baseline, the Baseline+HGBS improves the rendering quality in PSNR by 0.3, 0.9, and 0.53 on the three sequences, respectively, demonstrating the effectiveness of our proposed HGBS in reconstructing pose-consistent geometry. With the addition of the NRS, our complete GaussianHand (Baseline+HGBS+NRS), further improves the rendering quality over Baseline+HGBS in PSNR by 0.17, 0.11, and 0.04, achieving the best performance in terms of LPIPS, PSNR, and SSIM on all three sequences, only except for the SSIM on *test/Capture1*. The improvements of NRS are not as significant as HGBS, since it is a correction residual term to the standard LBS, but the improvements still demonstrate the existence of inaccurate pose deformation and the necessity of NRS.

The visualization results in Fig. 6 further demonstrate the effectiveness of our proposed HGBS and NRS components. The comparison results can be summarized as follows: 1) Nails: The first row shows that the Baseline fails to render the nail shape since it directly predicts the position offsets without considering geometric consistency over poses, leading to varied nail shapes. In contrast, Baseline+HGBS effectively renders the nails, although the nail edges appear slightly sharp. Baseline+HGBS+NRS further smooths the nail edges, rendering more realistic results. The nails in the second row also support this conclusion. 2) Fingers: The first and second rows show that the Baseline method distorts the edge shape of fingers. Our method maintains smooth shape consistency across different poses, demonstrating the effectiveness of HGBS in capturing pose-consistent geometry. 3) Wrinkles: The third row shows rendered palm wrinkles. The Baseline blurs the wrinkles, while the incorporation of HGBS provides clearer wrinkles due to more accurate geometric offsets. NRS further enhances the clarity of both wrinkles and nails. 4) Blood Vessels: The fourth row shows rendered blood vessels. The Baseline renders a shaky vessel shape. The introduction of HGBS produces a straighter and more realistic vessel shape, and the addition of NRS makes the vessels even clearer.

F. Geometry Visulization

To validate that our method accurately models hand surface geometry details (including wrinkles, veins, and nails) while maintaining consistency across various poses, we visualize two Gaussian shape basis, their corresponding relative blend coefficients (calculated by subtracting the coefficients of the canonical pose), and the posed blend-shaped Gaussian positions for three different MANO driving poses in Fig. 7.

Firstly, the visualization results demonstrate that our HGBS effectively formalizes the Gaussian shape basis as pose-independent for preserving geometric consistency, as its values remain stable across different poses. In contrast, the blend coefficients are pose-dependent since the first blend coefficient exhibits significant variability across different poses, while the second exhibits less but still notable variation.

Secondly, the visualization results confirm that our method can identify which hand areas should deform across poses. The first Gaussian shape basis highlights geometric deformation in the back of the hand and the finger roots, as indicated by red coloring (i.e., larger values), indicating that these areas could deform across poses. In contrast, the geometry of the gray areas remains consistent. The second shape basis emphasizes the deformation of the fingertips, particularly capturing the transformation of finger wrinkles.

Finally, we visualize the posed blend-shaped Gaussian positions as point clouds. The results demonstrate that our method accurately models the detailed hand geometry. The first row validates the fingernail silhouettes, the second row highlights the palm wrinkles, and the last row shows that the bulging blood vessels are preserved.

G. User Study

We design a within-subject study to evaluate the perceptual synthesis quality on the InterHand2.6M dataset for our GaussianHand and LiveHand.

Participants and Setup. We recruited 15 participants (8 males and 7 females, aged between 20-28 years), all of whom had normal or corrected-to-normal vision.

Conditions. The conditions included GaussianHand and LiveHand. Both methods are trained on the *test/Capture0* sequence of the InterHand2.6M dataset. Note that GaussianHand is trained on 20 views, while LiveHand is trained on 139 views.

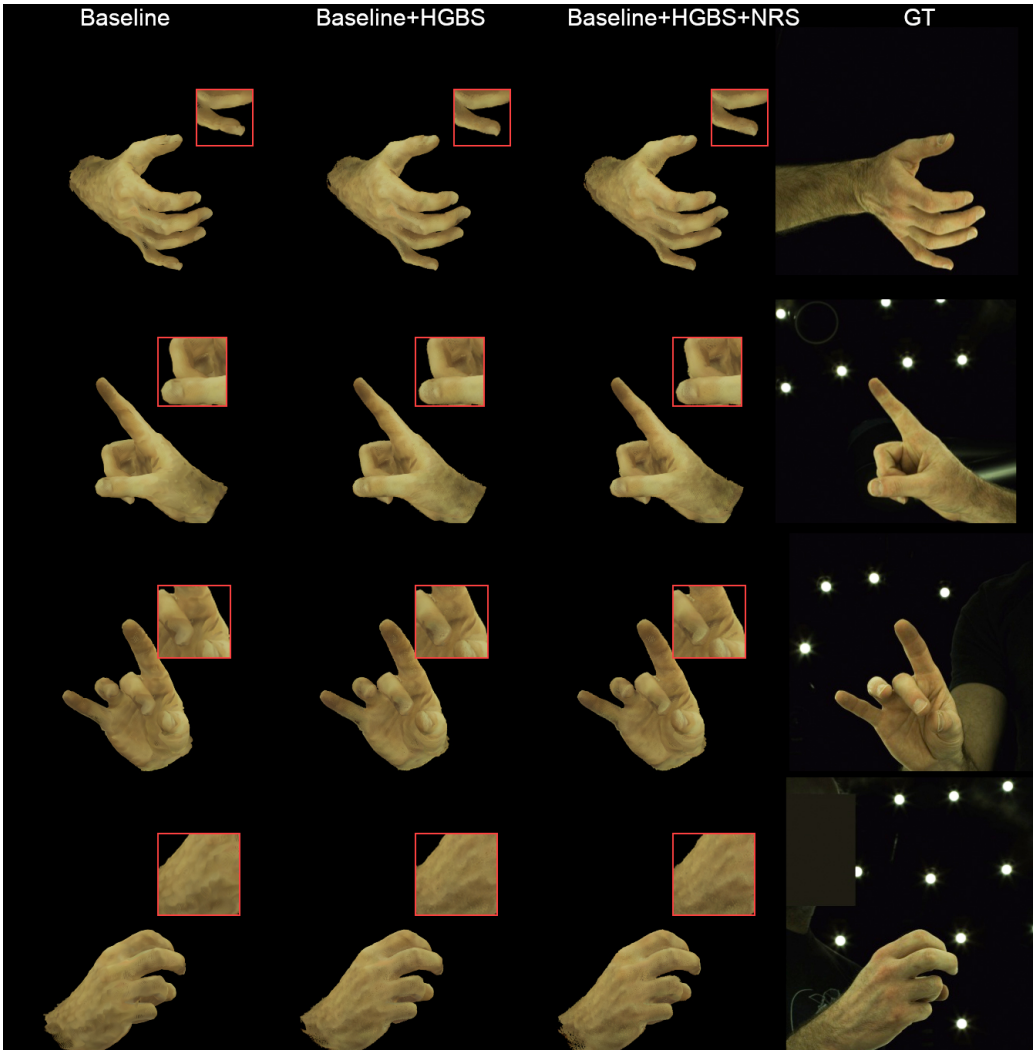

Fig. 6. Visualization results of ablation study for the HGBS and NRS modules on the InterHand2.6M dataset under 5 training views. The figure exposure is increased for clear visualization.

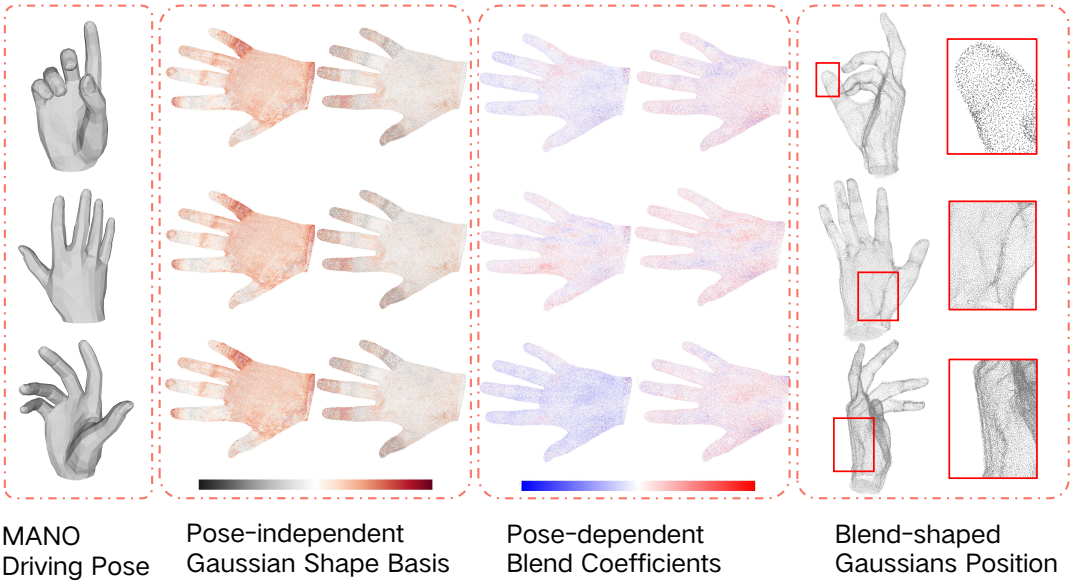

Fig. 7. Visualization of input MANO driving poses, two Gaussian shape basis with their corresponding blend coefficients (relative to the canonical ones), and the positions of the posed blend-shaped Gaussians. The colormaps represent values, with red indicating larger values.

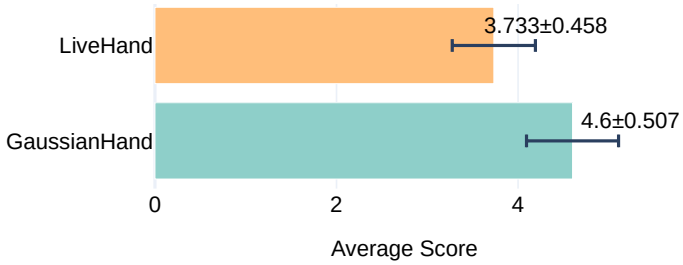

Fig. 8. Statistics results of the subjective user study for scoring the visual perceptual quality of LiveHand and our GaussianHand. Higher scores denote more realistic quality.

Task. We randomly select 15 poses from the validation split. We animate our GaussianHand and LiveHand with these poses to synthesize posed hands and rotate the camera around each hand for rendering free-view videos. Each video lasts for 8 seconds. The selected hand poses and camera trajectories are consistent across the two methods for fair comparison.

We show these videos to the participants and ask for realistic ratings. Specifically, for each pose, we present ground truth hand images of the pose under 3 different views first, then show the videos of the two methods simultaneously on the left and right sides of the screen for the sake of comparison. The orientation of the two videos is randomized. After displaying the videos, we present a questionnaire to the participants to rate the realism of the hands' appearance on a 5-point Likert scale, where 1 indicates "very unrealistic according to the ground truth" and 5 indicates "very realistic according to the ground truth". To mitigate the effects of visual fatigue, after completing the ratings, participants are given a 10-second rest before proceeding to the next pose.

Results. Figure 8 shows the statistical results of the average score across 15 poses for the two methods. The results indicate that our GaussianHand achieves an average score with a mean of 4.6 and a standard deviation of 0.507, while LiveHand achieves an average score with a mean of 3.733 and a standard deviation of 0.458. We apply the p-value and *Cohen's d* to estimate the average score differences. The p-value < 0.001 indicates a *significant* improvement in our GaussianHand, and the *Cohen's* $d = 1.794 > 0.8$, indicating a *huge* effect size. These results demonstrate that the visual perceptual quality of our GaussianHand significantly outperforms LiveHand for free-view rendering, even under the challenging setting of 20-view training. This is because our GaussianHand effectively captures detailed hand geometric features and maintains pose consistency, as demonstrated in the ablation study.

V. Conclusion

We proposed GaussianHand, the first Gaussian-based animatable hand avatar rendering pipeline that achieves high-quality and high efficiency for both free-pose and free-view hand appearance rendering. Our method involves two novel components, HGBS and NRS. The HGBS captures detailed hand geometric features while maintaining consistency across varied poses. Concurrently, the NRS with the learned RSW, serves as a corrective term for addressing inaccuracies in LBS deformations caused by the geometric offsets. Quantitative assessments and visualization experiments conducted on the InterHand2.6M and Hand Appearance datasets indicate that the proposed GaussianHand remarkably surpasses current methods in rendering quality and efficiency, achieving state-of-the-art performance. Our ablation studies further verify the efficacy of the proposed components, and the user study indicates that our rendering results significantly enhance users' subjective perceptual scores compared to the previous method.

Our method has several limitations. First, our network structure has feature coupling, which may hinder fast convergence and reduce interpretability. The coupling arises because all Gaussian properties are estimated from the shared canonical and pose features. Future work will focus on optimizing the network structure to enhance interpretability. Second, our method exclusively models hand geometry, neglecting the variability of pose-dependent self-shadows, the view-dependent appearance, and the relighting conditions. Hence, the hand appearance under our method exhibits static lighting and shadow effects and struggles with dynamic lighting sources. Future work aims to enhance our Gaussian hand by considering dynamic lighting effects. Third, our current model focuses on a single hand, without considering the complexities of two-hand interactions, which can involve intricate poses, occlusions, and potential geometry interpenetration challenges. Future work will consider the mutual information between hands to address these challenges. Fourth, the animation of the trained hand avatar is driven by input pose parameters without accounting for model penetration. As a result, our method renders a molded hand when the input driving poses involve self-penetration.

Hand avatar modeling presents potential societal challenges. The creation of highly realistic hand models raises concerns about consent, particularly when individuals' hands are modeled and driven without their explicit permission. Unauthorized use of avatar models could lead to issues such as identity theft or defamation, posing significant privacy risks. The community needs ethical regulations that address privacy concerns and ensure the fair use of hand modeling technologies.

VI. Acknowledgments

This work is supported by the National Natural Science Foundation of China through Project 61932003, 62372026, by Beijing Science and Technology Plan Project Z221100007722004, and by National Key R&D plan 2019YFC1521102.

References

[1] A. Mundra, J. Wang, M. Habermann, C. Theobalt, M. Elgharib *et al.*, "Livehand: Real-time and photorealistic neural hand rendering," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18 035–18 045.

[2] L. Zhao, X. Lu, Q. Bao, and M. Wang, "In-place gestures classification via long-term memory augmented network," in *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2022, pp. 224–233.

[3] L. Zhao, X. Lu, M. Zhao, and M. Wang, "Classifying in-place gestures with end-to-end point cloud learning," in *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2021, pp. 229–238.

[4] X. Shi, L. Wang, J. Wu, R. Fan, and A. Hao, "Foveated stochastic lightcuts," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 11, pp. 3684–3693, 2022.

[5] B. Chen, Y. Shen, H. Fu, X. Chen, K. Zhou, and Y. Zheng, "Neuralreshaper: single-image human-body retouching with deep neural networks," *Science China Information Sciences*, vol. 66, no. 9, p. 199101, 8 2023. [Online]. Available: https://doi.org/10.1007/s11432-022-3675-1

[6] J. Jiang, L. Zhao, X. Lu, W. Hu, I. Razzak, and M. Wang, "Dhgcn: Dynamic hop graph convolution network for self-supervised point cloud learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 11, 2024, pp. 12 883–12 891.

[7] J. Leng, L. Wang, X. Liu, X. Shi, and M. Wang, "Efficient flower text entry in virtual reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 11, pp. 3662–3672, 2022.

[8] Y. Shi, L. Zhao, X. Lu, T. Hoang, and M. Wang, "Grasping 3d objects with virtual hand in vr environment," in *Proceedings of the 18th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, 2022, pp. 1–8.

[9] D. Han, R. Lee, K. Kim, and H. Kang, "Vr-handnet: A visually and physically plausible hand manipulation system in virtual reality," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–12, 2023.

[10] K. Karunratanakul, S. Prokudin, O. Hilliges, and S. Tang, "Harp: Personalized hand reconstruction from a monocular rgb video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 802–12 813.

[11] Z. Chen, G. Moon, K. Guo, C. Cao, S. Pidhorskyi, T. Simon, R. Joshi, Y. Dong, Y. Xu, B. Pires, H. Wen, L. Evans, B. Peng, J. Buffalini, A. Trimble, K. McPhail, M. Schoeller, S.-I. Yu, J. Romero, M. Zollhöfer, Y. Sheikh, Z. Liu, and S. Saito, "URhand: Universal relightable hands," in *CVPR*, 2024.

[12] X. Tang, T. Wang, and C.-W. Fu, "Towards accurate alignment in real-time 3d hand-mesh reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 698–11 707.

[13] K. Zhou, H. P. H. Shum, F. W. B. Li, and X. Liang, "Multi-task spatial-temporal graph auto-encoder for hand motion denoising," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–17, 2023.

[14] 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," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[15] X. Chen, B. Wang, and H.-Y. Shum, "Hand avatar: Free-pose hand animation and rendering from monocular video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8683–8693.

[16] E. Corona, T. Hodan, M. Vo, F. Moreno-Noguer, C. Sweeney, R. Newcombe, and L. Ma, "Lisa: Learning implicit shape and appearance of hands," in *CVPR*, 2022.

[17] X. Zheng, C. Wen, Z. Su, Z. Xu, Z. Li, Y. Zhao, and Z. Xue, "Ohta: One-shot hand avatar via data-driven implicit priors," *arXiv preprint arXiv:2402.18969*, 2024.

[18] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, July 2023. [Online]. Available: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

[19] G. Wu, T. Yi, J. Fang, L. Xie, X. Zhang, W. Wei, W. Liu, Q. Tian, and X. Wang, "4d gaussian splatting for real-time dynamic scene rendering," *arXiv preprint arXiv:2310.08528*, 2023.

[20] 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," *arXiv preprint arXiv:2312.02134*, 2023.

[21] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, vol. 34, no. 6, pp. 248:1–248:16, Oct. 2015.

[22] 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*, 2022, pp. 16 210–16 220.

[23] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, and K. M. Lee, "Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*. Springer, 2020, pp. 548–564.

[24] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[25] A. Hore and D. Ziou, "Image quality metrics: Psnr vs. ssim," in *2010 20th international conference on pattern recognition*. IEEE, 2010, pp. 2366–2369.

[26] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[27] G. Chen and W. Wang, "A survey on 3d gaussian splatting," *arXiv preprint arXiv:2401.03890*, 2024.

[28] A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, W. Yifan, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi *et al.*, "Advances in neural rendering," in *Computer Graphics Forum*, vol. 41, no. 2. Wiley Online Library, 2022, pp. 703–735.

[29] B. Fei, J. Xu, R. Zhang, Q. Zhou, W. Yang, and Y. He, "3d gaussian splatting as new era: A survey," *IEEE Transactions on Visualization and Computer Graphics*, 2024.

[30] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, vol. 36, no. 6, Nov. 2017.

[31] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid, "Learning joint reconstruction of hands and manipulated objects," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 807–11 816.

[32] X. Zhang, Q. Li, H. Mo, W. Zhang, and W. Zheng, "End-to-end hand mesh recovery from a monocular rgb image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2354–2364.

[33] L. Yang, X. Zhan, K. Li, W. Xu, J. Li, and C. Lu, "Cpf: Learning a contact potential field to model the hand-object interaction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 097–11 106.

[34] B. Zhang, Y. Wang, X. Deng, Y. Zhang, P. Tan, C. Ma, and H. Wang, "Interacting two-hand 3d pose and shape reconstruction from single color image," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 11 354–11 363.

[35] K. Lin, L. Wang, and Z. Liu, "End-to-end human pose and mesh reconstruction with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1954–1963.

[36] X. Chen, Y. Liu, C. Ma, J. Chang, H. Wang, T. Chen, X. Guo, P. Wan, and W. Zheng, "Camera-space hand mesh recovery via semantic aggregation and adaptive 2d-1d registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 274–13 283.

[37] K. Lin, L. Wang, and Z. Liu, "Mesh graphormer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 939–12 948.

[38] X. Chen, Y. Liu, Y. Dong, X. Zhang, C. Ma, Y. Xiong, Y. Zhang, and X. Guo, "Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 544–20 554.

[39] L. Yang, K. Li, X. Zhan, J. Lv, W. Xu, J. Li, and C. Lu, "Artiboost: Boosting articulated 3d hand-object pose estimation via online exploration and synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 2750–2760.

[40] H. Xu, T. Wang, X. Tang, and C.-W. Fu, "H2onet: Hand-occlusion-and-orientation-aware network for real-time 3d hand mesh reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 048–17 058.

[41] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, "pixelnerf: Neural radiance fields from one or few images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4578–4587.

[42] Q. Gao, Y. Wang, L. Liu, L. Liu, C. Theobalt, and B. Chen, "Neural novel actor: Learning a generalized animatable neural representation for human actors," *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[43] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855–5864.

[44] T. Hu, H. Xu, L. Luo, T. Yu, Z. Zheng, H. Zhang, Y. Liu, and M. Zwicker, "Hvtr++: Image and pose driven human avatars using hybrid volumetric-textural rendering," *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[45] X. Chen, Y. Zheng, M. J. Black, O. Hilliges, and A. Geiger, "Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 594–11 604.
[46] S. Iwase, S. Saito, T. Simon, S. Lombardi, T. Bagautdinov, R. Joshi, F. Prada, T. Shiratori, Y. Sheikh, and J. Saragih, "Relightablehands: Efficient neural relighting of articulated hand models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 663–16 673.
[47] G. Moon, W. Xu, R. Joshi, C. Wu, and T. Shiratori, "Authentic hand avatar from a phone scan via universal hand model," in *Computer Vision and Pattern Recognition (CVPR)*, 2024.
[48] M. Kim and T.-K. Kim, "Bitt: Bi-directional texture reconstruction of interacting two hands from a single image," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
[49] Q. Gan, Z. Zhou, and J. Zhu, "Xhand: Real-time expressive hand avatar," 2024. [Online]. Available: https://arxiv.org/abs/2407.21002
[50] Y. Xu, B. Chen, Z. Li, H. Zhang, L. Wang, Z. Zheng, and Y. Liu, "Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.
[51] 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 (CVPR)*, 2024.
[52] J. Wen, X. Zhao, Z. Ren, A. Schwing, and S. Wang, "GoMAvatar: Efficient Animatable Human Modeling from Monocular Video Using Gaussians-on-Mesh," in *CVPR*, 2024.
[53] J. Lei, Y. Wang, G. Pavlakos, L. Liu, and K. Daniilidis, "Gart: Gaussian articulated template models," *arXiv preprint arXiv:2311.16099*, 2023.
[54] S. Qian, T. Kirschstein, L. Schoneveld, D. Davoli, S. Giebenhain, and M. Nießner, "Gaussianavatars: Photorealistic head avatars with rigged 3d gaussians," *arXiv preprint arXiv:2312.02069*, 2023.
[55] 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," *arXiv preprint arXiv:2403.05087*, 2024.
[56] S. Saito, G. Schwartz, T. Simon, J. Li, and G. Nam, "Relightable gaussian codec avatars," *arXiv preprint arXiv:2312.03704*, 2023.
[57] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," in *ACM siggraph 2006 papers*, 2006, pp. 835–846.
[58] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

Lizhi Zhao received his Master's degree from the College of Information Engineering at Northwest A&F University, China, in 2023. He is currently pursuing a Ph.D. at the State Key Laboratory of Virtual Reality Technology and Systems, within the School of Computer Science and Engineering at Beihang University. His research primarily focuses on computer graphics and virtual reality.

Xuequan Lu is a Senior Lecturer at the Department of Computer Science and IT, La Trobe University, Australia. He spent more than two years as a Research Fellow in Singapore. Prior to that, he earned his PhD at Zhejiang University (China) in June 2016. His research interests mainly fall into the category of visual computing, for example, geometry modeling, processing and analysis, animation/simulation, 2D data processing and analysis. More information can be found at http://www.xuequanlu.com.

Runze Fan is a Ph.D student at the State Key Laboratory of Virtual Reality Technology and Systems within the School of Computer Science and Engineering, Beihang University, China. His current research focuses on virtual reality and augmented reality.

Sio Kei Im received his Ph.D. degree in Electronic Engineering from Queen Mary University of London (QMUL), United Kingdom. He is a professor at the Faculty of Applied Sciences, Macau Polytechnic University, and a researcher at the Engineering Research Center of Applied Technology on Machine Translation and Artificial Intelligence, Ministry of Education. His research interests include video coding, image processing, machine learning for NLP and multimedia.

Lili Wang received her Ph.D. degree from the Beihang University, Beijing, China. She is a professor with the School of Computer Science and Engineering of Beihang University, and a researcher with the State Key Laboratory of Virtual Reality Technology and Systems. Her interests include virtual reality, augmented reality, mixed reality, real-time rendering and realistic rendering.