AvatarReX: Real-time Expressive Full-body Avatars

ZERONG ZHENG, Department of Automation, Tsinghua University, China
XIAOCHEN ZHAO, Department of Automation, Tsinghua University and NNKosmos Technology, China
HONGWEN ZHANG, Department of Automation, Tsinghua University, China
BONING LIU, Department of Automation, Tsinghua University, China
YEBIN LIU, Department of Automation, Tsinghua University, China

Fig. 1. **Example animation results produced by our method.** Our method can learn photorealistic full-body avatars that provides full controllability over the body pose, hand genstrue and the face expression all together. Moreover, it can be rendered in real time without compromising image quality.

We present AvatarReX, a new method for learning NeRF-based full-body avatars from video data. The learnt avatar not only provides expressive control of the body, hands and the face together, but also supports real-time animation and rendering. To this end, we propose a compositional avatar representation, where the body, hands and the face are separately modeled in a way that the structural prior from parametric mesh templates is properly utilized without compromising representation flexibility. Furthermore, we disentangle the geometry and appearance for each part. With these technical designs, we propose a dedicated deferred rendering pipeline, which can be executed at a real-time framerate to synthesize high-quality free-view images. The disentanglement of geometry and appearance also allows us to design a two-pass training strategy that combines volume rendering and surface rendering for network training. In this way, patch-level supervision can be applied to force the network to learn sharp appearance details on the basis of geometry estimation. Overall, our method enables automatic construction of expressive full-body avatars with real-time rendering capability, and can generate photo-realistic images with dynamic details for novel body motions and facial expressions.

CCS Concepts: • **Computing methodologies** → **Computer vision**; **Rendering**.

Additional Key Words and Phrases: full-body avatars, real-time rendering, neural radiance fields

ACM Reference Format:
Zerong Zheng, Xiaochen Zhao, Hongwen Zhang, Boning Liu, and Yebin Liu. 2023. AvatarReX: Real-time Expressive Full-body Avatars. *ACM Trans. Graph.* 42, 4 (August 2023), 19 pages. https://doi.org/10.1145/3592101

1 INTRODUCTION

Animatable human avatar modeling, as an important topic in special effects industry, can be applied in many applications such as content creation and immersive entertainment. Virtual characters are believed to have the potential to open up a new way for people to interact with others or intelligent machines in AR/VR settings [Chu et al. 2020]. Unfortunately, the traditional pipeline for creating 3D human avatars involves tedious procedures, including scanning, meshing, rigging and many more. Furthermore, such a pipeline requires expert knowledge and sophisticated capture systems, limiting its access and increasing its cost [Alexander et al. 2010].

In order to lower the entry barrier for novices and automate the workflow of experienced artists, researchers have devoted great efforts in learning 3D human body avatars from images or videos of real humans. Building upon explicit meshes [Bagautdinov et al.

Authors' addresses: Zerong Zheng, Department of Automation, Tsinghua University, Beijing, China, zrzheng1995@foxmail.com; Xiaochen Zhao, Department of Automation, Tsinghua University and NNKosmos Technology, Beijing, China, zhaoxc19@mails.tsinghua.edu.cn; Hongwen Zhang, Department of Automation, Tsinghua University, Beijing, China, zhanghongwen@mail.tsinghua.edu.cn; Boning Liu, Department of Automation, Tsinghua University, Beijing, China, liuboning@mail.tsinghua.edu.cn; Yebin Liu, Department of Automation, Tsinghua University, Beijing, China, liuyebin@mail.tsinghua.edu.cn.

0730-0301/2023/8-ART $15.00
https://doi.org/10.1145/3592101

2021; Habermann et al. 2021; Xiang et al. 2021], implicit radiance fields [Peng et al. 2021a; Su et al. 2021; Wang et al. 2022b; Weng et al. 2022; Zheng et al. 2022b] or both [Liu et al. 2021; Lombardi et al. 2021; Remelli et al. 2022], current approaches are able to synthesize realistic human motions under free-view points. Despite the plethora of these systems, most of them only model the torsos and the limbs, without combining other fine-grained body parts like faces and hands. However, an expressive human avatar demands for full controllability of the body, hands and the face together, as the essential nuance of human behaviors is conveyed through a concert of body movements, hand gestures and facial expressions. Up to now, only research works from industrial labs are able to achieve this goal [Bagautdinov et al. 2021; Remelli et al. 2022; Xiang et al. 2022, 2021]. Unfortunately, their methods rely on video data captured from dense-view camera rigs, which are not accessible for most individuals and academic organizations.

Apart from expressiveness, another unsolved challenge lies in the rendering speed. In many applications, the avatars are expected to interact with users as real human-beings, which emphasizes the need for real-time animation and rendering. However, the most recent approaches [Liu et al. 2021; Peng et al. 2021a; Su et al. 2021; Wang et al. 2022b; Zheng et al. 2022b] in this field are typically built upon neural radiance fields (NeRF) [Mildenhall et al. 2020], which densely samples the 3D space and queries the network millions of times for volume rendering. Consequently, they are difficult to render a dynamic avatar at a real-time framerate, preventing them from being adopted in interactive scenarios.

In this work, we present **Re**al-time e**X**pressive **Avatar** (AvatarReX), a novel system that simultaneously achieves expressive control and real-time animation of full-body human avatars. To this end, we propose a compositional representation that models the face, hands and the body with independent implicit fields (Section 3). Such a compositional design allows us to adopt the most suitable technique for each part according to its characteristics in shape and texture variations. In our method, all part representations rely on the corresponding parametric templates, *i.e.*, SMPL-X [Pavlakos et al. 2019] for the body, MANO [Romero et al. 2017] for the hands and Faceverse [Wang et al. 2022a] for the face, but the prior in these templates is leveraged in totally different manners. For example, we employ structured local radiance fields [Zheng et al. 2022b] for clothed body representation in order to eliminate the reliance on the SMPL-X topology, and, in contrast, directly build the the radiance fields of hands on top of the MANO geometry model. This is because clothes vary in topology, while the shape variation of hands is rather limited. To enhance the expressiveness of our avatar, we introduce several novel techniques for the body and face parts. Specifically, to capture the rich dynamic details like cloth wrinkles exhibited in the body part, we propose explicit feature patches to facilitate the learning of these appearance details. Meanwhile, as the face part contains many subtle yet important details relating to different expressions, we utilize convolution networks to extract distinctive features from the expression space of Faceverse, which is a better condition for high-fidelity facial rendering. Overall, each part representation in our avatar is carefully designed, ensuring that the 3D prior of the parametric templates is properly utilized without sacrificing representation power and model flexibility.

With these building blocks at hand, we can now derive the final expressive avatar by assembling the face, hands and the body into one holistic model. However, it remains a challenge to train an avatar with sharp appearance details and render it efficiently at test time. To resolve these problems, we further improve our avatar representation by disentangling the geometry and appearance for each part and using the signed distance function (SDF) as the common geometric representation. These technical designs enable us to develop a real-time rendering pipeline based on deferred rendering (Section 4). The core of our pipeline is to take advantage of the implicit surface definition in SDF for surface rendering. Compared to volume rendering in most NeRF-based methods [Mildenhall et al. 2020], surface rendering avoids the need for expensive sampling along camera rays, leading to a significant speedup. We further accelerate this process with a dedicated deferred shading scheme, where the geometry model is firstly reconstructed in the form of an explicit SDF volume, enabling fast surface location for the later color evaluation step. As a result, our method successfully accelerates the avatar rendering process by two orders of magnitude without compromising the quality.

In addition to testing acceleration, the disentanglement of geometry and appearance can also benefit network training. To this end, we combine deferred surface rendering with volume rendering to form a two-pass training strategy (Section 5). On one hand, volume rendering allows the geometry networks to learn various cloth shape from scratch with sparse pixel supervision. On the other, when the surface reconstruction is available, surface rendering minimizes network queries, making it possible to apply perceptual supervision on image patches, which is essential to force the color network to learn high-quality appearance. With our disentangled design, we can combine the merits from both worlds into two training passes, and consequently obtain an avatar with sharp appearance details.

In summary, our system is able to create expressive full-body avatars with high-quality appearance details, and the avatars can be animated and rendered in real time. The training data for our avatar is multi-view videos of approximately 2000 frames in length, captured from 22 cameras (16 for the body and hands and 6 for the face). After data collection, our method can automatically learn the avatar representation without the need for pre-scanning efforts or other manual intervention. Moreover, the learned avatar can be driven in real-time given new body poses, hand gestures and facial expressions. Experimental results clearly demonstrate the potential of applying our method in interactive applications.

2 RELATED WORK

Body Avatars. In the last decade, many efforts have been made to achieve animatable human avatars. Pioneer works in this field resort to statistical mesh templates [Joo et al. 2018; Loper et al. 2015; Osman et al. 2020; Pavlakos et al. 2019] to model minimally clothed bodies. To handle the varying shape of clothing, recent methods explore more flexible representations such as implicit fields to model the shapes of clothed humans [Chen et al. 2021; Li et al. 2022b; Lin et al. 2022; Mihajlovic et al. 2021; Saito et al. 2021; Tiwari et al. 2021; Zhang et al. 2023a]. For instance, Neural-GIF [Tiwari

et al. 2021] factorizes human motion into articulation and non-rigid deformation, and learns to map every point in the space to a canonical pose using backward skinning. LEAP [Mihajlovic et al. 2021] and SCANimate [Saito et al. 2021] learn the forward and backward skinning fields with neural networks and regularize the cycle consistency between them. For better generalization to unseen poses, SNARF [Chen et al. 2021] proposes a differentiable forward skinning model based on iterative root finding, which finds the canonical correspondences of any query point in the posed space.

To acquire animatable characters with color, traditional pipelines typically reconstruct a subject-specific textured mesh in advance, and then generate its motions using physics simulation [Guan et al. 2012; Stoll et al. 2010], database retrieval and blending [Xu et al. 2011], or deformation space modeling [Habermann et al. 2021; Joo et al. 2018]. With accurate tracking of the underlying geometry, Bagautdinov et al. [2021] model high-fidelity avatars by decoding dynamic geometry and appearance from disentangled driving signals. This approach is further extended in [Xiang et al. 2022, 2021] by representing the clothing as a separate layer in order to recover sharper garment boundaries. Their reliance on pre-scanning subject-specific templates can be eliminated via deforming a general body template. For instance, several works proposed to directly learn this deformation from geometric data [Ma et al. 2020, 2021b; Pons-Moll et al. 2017] or RGB videos [Alldieck et al. 2018a,b; Burov et al. 2021]. The texture map and the rasterization step in these methods are later replaced with neural texture maps and image decoders [Hu et al. 2022; Liu et al. 2020, 2019b; Prokudin et al. 2021; Raj et al. 2021; Shysheya et al. 2019] to achieve more photo-realistic rendering.

In the past three years, neural volumetric representations have demonstrated impressive results on novel view synthesis of static scenes [Mildenhall et al. 2020] or dynamic scenes [Lombardi et al. 2019]. Since then, great efforts have been made to extend these neural representations to human avatars. For example, Neural Body [Peng et al. 2021b] uses SMPL [Loper et al. 2015] to establish correspondences across different frames and trains a sparse convolution network to convert the SMPL vertices into a radiance volume. It supports high-quality view synthesis, but can only playback the training sequences. To achieve better pose generalization, recent works adopt a disentangled representation, which maps the points in different poses to a canonical radiance field using inverse skinning [Li et al. 2022a, 2023; Peng et al. 2021a; Su et al. 2021; Wang et al. 2022b; Weng et al. 2022]. For modeling the dynamic appearance details, Neural Actors [Liu et al. 2021] introduces a texture map as an additional condition. In contrast to them, Zheng et al. [2022b] present a structured local representation, where the radiance field of dynamic characters is assembled by a set of local ones, similarly to MVP [Lombardi et al. 2021]. DANBO [Su et al. [n. d.]] employs a part-based volumetric representation defined by the skeleton structure using graph neural networks. Recently, Remelli et al. [2022] combine localized volumetric primitives with the dense signal from image observations, allowing faithful synthesis of appearance details like cloth wrinkles. However, their avatars can only be driven by the same person in the same attire due to its requirement of driving views. In contrast, our method takes as input solely the pose parameters and the expression coefficients no matter where they come from, thus our avatar can be driven by another person or other signal sources. Concurrent to us, TotalSelfScan [Dong et al. 2022b] reconstructs a full-body model from self-portrait videos of faces, hands, and bodies, but it can only produce articulated body motions and fails to synthesize photo-realistic dynamic appearance.

Face Avatars. Similar to human bodies, face avatar techniques have also undergone significant advancements. The seminal work dating back to 1999 built the first 3D morphable model, enabling representation of facial shapes by embedding different identities and expressions into low-dimensional PCA spaces [Blanz and Vetter 1999]. To model complex deformations and textures, researchers have exploited more advanced modeling tools, such as multi-linear models [Cao et al. 2014; Vlasic et al. 2006], nonlinear models [Guo et al. 2021a; Tran and Liu 2018] and the articulated control of expression [Li et al. 2017]. Recent methods further recover high-frequency deformations of expressions by learning additional displacement maps on top of the base mesh model [Danecek et al. 2022; Feng et al. 2021; Grassal et al. 2022]. Moreover, some researchers propose reconstructing facial avatars with remarkable quality for immersive telepresence based on dense multiview capture systems [Chu et al. 2020; Lombardi et al. 2018; Ma et al. 2021a; Wang et al. 2023].

Since the debut of implicit representations like DeepSDF [Park et al. 2019] or NeRF [Mildenhall et al. 2020], it becomes an growing trend to model 3D faces or heads in an implicit fashion. Building upon neural implicit functions, Yenamandra et al. [2021] developed the first deep implicit 3D morphable model of full heads including faces and hairs. Similarly, Hong et al. [2022] adopt NeRF to create a parametric head model that supports high-fidelity head image rendering in real-time. However, these models focus on learning generic head models using data from multiple subjects, often lacking personalized appearance details. To address this issue, NeRFace [Gafni et al. 2021] proposes a personalized head avatar by taking expression coefficients as the additional inputs for the head NeRF, and demonstrates state-of-the-art reenactment and rendering results. IMAvatar [Zheng et al. 2022a] incorporates skinning fields with an implicit morphing-based model, which allows better geometry reconstruction and stronger generalization capability for novel expressions. Gao et al. [2022] bridge traditional mesh blendshapes with voxel-based implicit fields, enabling fast construction of personalized head NeRF models from monocular videos. These methods typically require dense images or videos as input, and efforts have been made to alleviate the reliance on large amounts of data [2022; 2023a; 2023b; 2022]. Some recent research works also propose to replace the expression coefficients with other driving signals, such as audio [Guo et al. 2021b] and gaze [Richard et al. 2021].

NeRF Acceleration. Numerous works have emerged with the purpose of speeding up static NeRF using explicit data structures including feature maps, voxels and tensors. For instance, DVGO [Sun et al. 2022b] achieves fast convergence through an explicit representation of a density voxel grid and a feature voxel grid. Plenoxels [Fridovich-Keil et al. 2022] and PlenOctree [Yu et al. 2021] model a scene through a hierarchical 3D grid with spherical harmonics, which can realize an optimization with two orders of magnitude faster than NeRF. DIVeR [Wu et al. 2022] accelerates volumetric rendering by limiting ray marching to a fixed number of hits on the voxel grid. Hashing encoding [Muller et al. 2022] and tensor decomposition [Chen et al. 2022b] are also used as compact representations

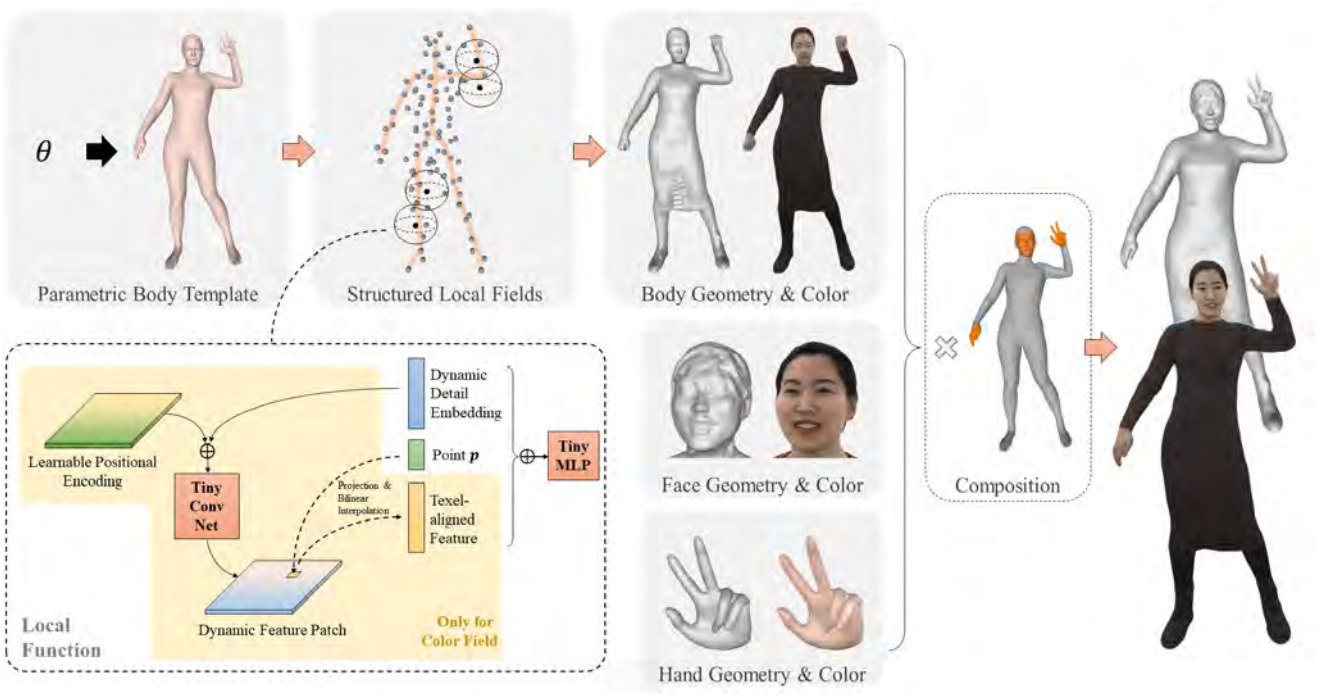


Fig. 2. **Illustration of our compositional avatar representation.** Our expressive avatar is composed of three parts, namely the major body, the hands and the face. For clarity, we only illustrate the body representation here, and leave the other two in Figure 3 and Figure 4. The core of our body representation is a set of structured local implicit fields, and we enhance their detail representation power by introducing an explicit dynamic feture patch for each field.

for NeRF acceleration. Similar techniques have also been applied for dynamic scene rendering. For example, DeVRF [Liu et al. 2022] enables fast non-rigid neural rendering with both 3D volumetric and 4D voxel fields. TiNeuVox [Fang et al. 2022] represents scenes with optimizable explicit data structures and accelerates radiance fields modeling, while Wang et al. [2022d] extended PlenOctree into free-view video rendering. Despite demonstrating significant speedup in NeRF training and testing, most of these works can only render static scenes or playback a dynamic sequence, making them unsuitable for character animation settings.

In this paper, we propose a novel real-time rendering pipeline based on deferred surface rendering. The philosophy behind our design is similar to MobileNeRF [Chen et al. 2022a], which also computes the pixel color in the image space rather than volume rendering. However, MobileNeRF only works for static scenes because it represents the scene as a fixed triangle mesh. In contrast, the topology of our avatar varies from pose to pose, and cannot be modeled with a stationary mesh. Therefore, we extract the pose-dependent geometry model on the fly via taking advantage of the SDF definition and the parallelism in our representation. We further disentangle geometry and appearance to reduce the computational burden in geometry reconstruction. Although disentangling geometry and appearance has been proposed for multi-view geometry reconstruction [Yariv et al. 2020], we are the first to employ it for real-time rendering. This requires us to accelerate the expensive operation of sphere tracing, and we achieve this goal by caching SDF values in an explicit volumetric grid.

3 AVATAR REPRESENTATION

This section discusses how we represent our expressive avatar. The avatar is composed of three parts, namely the body, the hands and the face. Considering their different characteristics in shape and texture variations, we design different neural implicit representations for them. They all rely on the corresponding parametric mesh templates, *i.e.*, SMPL-X for the body [Pavlakos et al. 2019], MANO for the hands [Romero et al. 2017] and Faceverse for the face [Wang et al. 2022a], but these templates is leveraged in totally different manners, as we will discuss in Section 3.1, 3.2 and 3.3. Finally, we introduce the technique for combining all the parts into one final avatar model in Section 3.4.

Notation. In the following text, we denote the part representation with $\mathcal{A}_*(s)$, where s is the driving signal and the subscript "$*$" can be {"B", "H", "F"}, representing "body", "hand" and "face", respectively. Accordingly, the driving signal s comes from body poses $\boldsymbol{\theta}$, hand poses $\boldsymbol{\phi}$ or facial expressions $\boldsymbol{\psi}$. Each part representation consists of two components, namely a geometry field $\mathcal{G}_*$ and a view-dependent color field C_*:

$$\mathcal{A}_*(s) = \{\mathcal{G}_*(\boldsymbol{p}|s), C_*(\boldsymbol{p}, \boldsymbol{v}|s)\}, \tag{1}$$

where $\boldsymbol{p}$ and $\boldsymbol{v}$ are the 3D point position and the viewing direction, respectively. For ease of notation, we drop the dependency on view directions when discussing the color fields in the upcoming sections.

In our method, we represent the geometry field as a signed distance function (SDF), where the true surface is embedded as its zero-level set $\{\boldsymbol{p} \in \mathbb{R}^3 | \mathcal{G}_*(\boldsymbol{p}|s) = 0\}$. Note that we do not follow the

vanilla NeRF [Mildenhall et al. 2020] and most NeRF-based avatar works [Peng et al. 2021a; Zheng et al. 2022b] that use one network to simultaneously model the geometry (density) field and the color field. Instead, we model them in a disentangled fashion and use smaller network size for the geometry fields. Such a design is for the purpose of real-time implementation as well as two-pass training, as we will discussed in Section 4 & 5 .

3.1 Body Representation

We adopt structured local radiance fields [Zheng et al. 2022b] as the representation of our body geometry field. Here we briefly review its construction for completeness. Specifically, we pre-define a set of nodes $\{\bar{\boldsymbol{n}}_i\}_{i=1}^{N}$ on the SMPL-X model [Pavlakos et al. 2019] via farthest point sampling. Since the nodes are sampled from the SMPL-X model, each of them has associated skinning weights, thus can be driven by the skeleton. Furthermore, we allow the nodes to have their own residual movements $\Delta\boldsymbol{n}_i$ to represent the non-rigid deformation of garments.

Around each node, we construct a local implicit field centered at it. Take the geometry field $\mathcal{G}_\mathrm{B}$ as an example. For each node, we define a function $\mathcal{G}_i$ in a local space around it, and use a tiny MLP to represent this function. This MLP takes as input a coordinate in the local space of node i and outputs a high-dimensional feature vector, which will be blended with the outputs from other local MLPs and finally decoded into an SDF value. Formally, given any point $\boldsymbol{p} \in \mathbb{R}^3$ in the posed space of pose $\boldsymbol{\theta}$, we first calculate its coordinate in the local space of node i as:

$$\boldsymbol{p}_i = \mathbf{T}^{-1}\boldsymbol{p} - (\bar{\boldsymbol{n}}_i + \Delta\boldsymbol{n}_i), \tag{2}$$

where $\mathbf{T}$ is the skinning matrix computed from the pose parameter $\boldsymbol{\theta}$ using linear blending skinning and $\boldsymbol{n}_i$ is the position of node i in the posed space. After that, we feed it into the local network $\mathcal{G}_i$ and blend the feature vectors produced by all the local MLPs:

$$\boldsymbol{f} = \frac{\sum w_i \mathcal{G}_i(\boldsymbol{p}_i, \boldsymbol{e}_i)}{\sum w_i}, \tag{3}$$

where $\boldsymbol{e}_i$ is a dynamic detail embedding predicted from the pose parameters and models the fine-grain deformations that cannot be represented by node movements. The blending weight w_i is calculated as:

$$w_i = \max\left\{\exp\left(\frac{-\|\boldsymbol{p} - \boldsymbol{n}_i\|_2^2}{2\sigma^2}\right) - \epsilon, 0\right\}, \tag{4}$$

where σ and ϵ are hyperparameters controlling the influence radius of the local networks. The blended feature $\boldsymbol{f}$ is fed into an additional MLP $\mathcal{G}_\mathrm{blending}$ to compute the SDF value of $\boldsymbol{p}_i$:

$$\mathcal{G}_\mathrm{B}(\boldsymbol{p}|\boldsymbol{\theta}) = \mathcal{G}_\mathrm{blending}(\boldsymbol{f}). \tag{5}$$

Dynamic Feature Patch. The color field of our body representation can be modeled in a similar way to the geometry field, *i.e.*, with a set of local MLPs $\{C_i\}$ and a blending MLP C_blending. However, due to the low-frequency bias [Tancik et al. 2020], the local MLPs $\{C_i\}$ are not powerful enough to represent the high-frequency details like the cloth wrinkles and texture patterns. To address this limitation, we introduce a dynamic feature patch for each local function of the body color field. Compared to purely implicit representation, an explicit feature patch contains more spatial information and provides stronger capability to store the information about high-frequency local details [Liu et al. 2021]. The feature patch, denoted as $\mathbf{F}_i$, is regressed by a tiny convolution network from the dynamic detail embedding $\boldsymbol{e}_i$ and a 2D learnable positional encoding, as illustrated in Figure 2. Given the local point position $\boldsymbol{p}_i$, we project it onto the feature patch:

$$\boldsymbol{u}_i = \Pi_i \boldsymbol{p}_i, \tag{6}$$

where $\Pi_i \in \mathbb{R}^{2\times 3}$ is the projection matrix used for projecting 3D points along a specific direction. Here we pre-compute the projecting direction by averaging the normal orientations of the SMPL-X vertices that locate nearby node i. After that, we query for the feature vector at $\boldsymbol{u}_i$ through bilinear interpolation:

$$\boldsymbol{f}_i = \mathrm{Bilinear}(\mathbf{F}_i, \boldsymbol{u}_i). \tag{7}$$

The interpolated feature vector is finally taken as an additional input by the local function C_i to produce the final color value:

$$\begin{aligned} \boldsymbol{f}' &= \frac{\sum w_i C_i(\boldsymbol{p}_i, \boldsymbol{e}_i, \boldsymbol{f}_i)}{\sum w_i}, \\ C_\mathrm{B}(\boldsymbol{p}|\boldsymbol{\theta}) &= C_\mathrm{blending}(\boldsymbol{f}'). \end{aligned} \tag{8}$$

As shown in Section 6.3, the proposed dynamic feature patches allow our color networks to learn more appearance details for the major body.

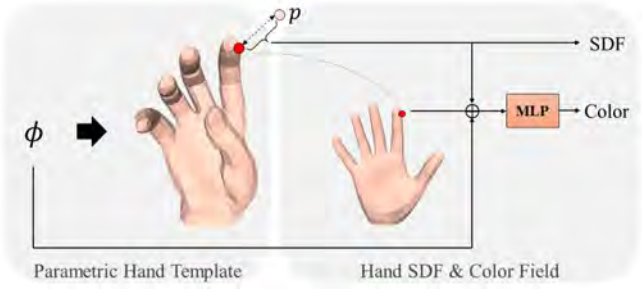


Fig. 3. **Hand representation.** We directly use the SDF from the parametric template as the geometry field of our hand representation, and learn an MLP to model the color field in the canonical space.

3.2 Hand Representation

Unlike the major body with clothing, hands show limited variations in shape and topology. Therefore, we directly use the surface of a parametric hand template to construct the geometry field of the hand, thus alleviating the need for learning complex articulated motion of fingers. Here we use MANO [Romero et al. 2017] as the base hand representation. For any spatial point $\boldsymbol{p} \in \mathbb{R}^3$ in hand pose $\boldsymbol{\phi}$, we project it onto the MANO mesh by first calculating its barycentric projection on each triangle face of the MANO surface and then finding the nearest one. Mathematically, this procedure can be formulated as:

$$\begin{aligned} (u^*, v^*, w^*, \mathbf{F}^*) = \arg\min_{u,v,w,\mathbf{F}} &\|\boldsymbol{p} - \mathrm{Barycentric}(\mathbf{F}, u, v, w)\|_2^2, \\ s.t., \quad &0 \le u, v, w \le 1, \\ &u + v + w = 1 \end{aligned} \tag{9}$$

where $\mathbf{F}$ is a triangle of the MANO mesh and Barycentric(...) is the barycentric interpolation function. Using the barycentric coordinates (u^*, v^*, w^*) and the interpolation function, we determine the nearest point of $\boldsymbol{p}$ on the MANO mesh, which we denoted as $\boldsymbol{m}_{\text{np}}$. Similarly, we can compute its normal direction $\boldsymbol{n}_{\text{np}}$ through barycentric interpolation. The SDF value of $\boldsymbol{p}$ can be finally calculated as:

$$\mathcal{G}_{\text{H}}(\boldsymbol{p}|\boldsymbol{\phi}) = \begin{cases} ||\boldsymbol{p} - \boldsymbol{m}_{\text{np}}||_2, & \boldsymbol{n}_{\text{np}}^{\top}(\boldsymbol{p} - \boldsymbol{m}_{\text{np}}) \geq 0 \\ -||\boldsymbol{p} - \boldsymbol{m}_{\text{np}}||_2, & \boldsymbol{n}_{\text{np}}^{\top}(\boldsymbol{p} - \boldsymbol{m}_{\text{np}}) < 0 \end{cases}. \tag{10}$$

To model the color field of the hand, we turn to a neural perspective. We first calculate the canonical position of $\boldsymbol{m}_{\text{np}}$ via interpolating the position of $\mathbf{F}$ in the canonical pose according to the barycentric coordinate. The interpolated result $\bar{\boldsymbol{m}}_{\text{np}}$, together with the normal direction in posed space $\boldsymbol{n}_{\text{np}}$, the signed distance $\text{SDF}(\boldsymbol{p})$ and the hand pose $\boldsymbol{\phi}$, is fed into an MLP to produce the color value:

$$C_{\text{H}}(\boldsymbol{p}|\boldsymbol{\phi}) = C_{\text{H}}^{'}\left(\bar{\boldsymbol{m}}_{\text{np}}, \boldsymbol{n}_{\text{np}}, \text{SDF}(\boldsymbol{p}), \boldsymbol{\phi}\right), \tag{11}$$

where $C_{\text{H}}^{'}$ denotes the MLP network. In this way, our method is able to model the pose-dependent appearance of hands without the burden of learning complex hand motions. Figure 3 illustrates our hand representation.

3.3 Face Representation

Compared to the body and the hands, representing the face is more challenging as humans are social animals that rely on facial expressions to read and convey emotions. As a result, we need to achieve both photo-realistic synthesis and accurate expression control in order to overcome the well-known uncanny valley. To this end, we propose to combine the the rendering power of NeRF and the prior knowledge from the facial morphable model [Wang et al. 2022a].

Given the expression coefficients $\boldsymbol{\psi}$, we first compute the corresponding 3D facial model. The model is an extremely coarse approximation of the real face, but it provides structural information about the specific expression. To utilize this structural knowledge in an efficient manner, we take inspiration from EG3D [Chan et al. 2021] and propose to learn a triplane-alike facial avatar representation. Specifically, we render the model from its front view and two side views using orthogonal projection, as illustrated in Figure 4. The rendered images are passed through three convolutional neural encoders, which extract the corresponding feature tri-planes denoted as $\{\mathbf{F}_{\text{front}}, \mathbf{F}_{\text{left}}, \mathbf{F}_{\text{right}}\}$. Given a spatial point $\boldsymbol{p} \in \mathbb{R}^3$, we project it onto the feature tri-planes and retrieve its pixel-aligned feature vectors through bilinear interpolation:

$$\boldsymbol{f}_v = \text{Bilinear}(\mathbf{F}_v, \Pi_v(\boldsymbol{p})), \tag{12}$$

where $\Pi_v \in \mathbb{R}^{2\times 3}$ is the projection matrix and the subscript $v \in \{\text{“front”}, \text{“left”}, \text{“right”}\}$ denotes the projection direction. The feature vectors, together with the point coordinate, are fed into an MLP to produce the sign distance of $\boldsymbol{p}$:

$$\mathcal{G}_{\text{F}}(\boldsymbol{p}|\boldsymbol{\psi}) = \mathcal{G}_{\text{F}}^{'}(\boldsymbol{p}, \boldsymbol{f}_{\text{front}}, \boldsymbol{f}_{\text{left}}, \boldsymbol{f}_{\text{right}}), \tag{13}$$

where $\mathcal{G}_{\text{F}}^{'}$ is an MLP network. The color of $\boldsymbol{p}$ is predicted in a similar way using another set of convolutional encoders and another MLP.

Previous works on NeRF-based facial avatar typically use pure MLPs as the network architecture and take the global expression coefficients as the network condition [Gafni et al. 2021; Zheng et al. 2022a]. Compared to them, the explicit feature tri-planes allow us to keep the MLP decoder as light-weight as possible, thus reducing the computational cost of neural rendering. Furthermore, the feature tri-planes provide spatially varying conditioning for the 3D space, which has stronger power in representing appearance details than a global expression condition.

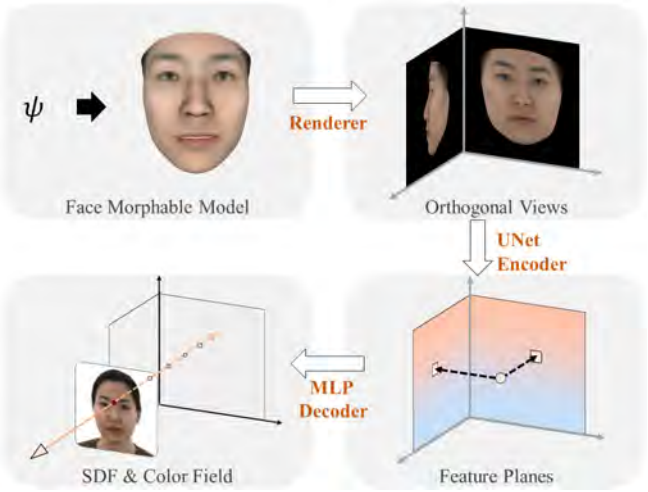


Fig. 4. **Face representation.** We condition NeRF on the orthogonal views of a 3D morphable model, which provides structural prior of the face and controllability over expressions.

3.4 Composition

With all the necessary building blocks at hand, we can now introduce how we combine different body parts into an expressive avatar. There are two key technical designs that assist us towards this goal. One is the usage of SDF, which has a unified, unambiguous definition across different parts. The other one is that our hand representation and face representation are tightly coupled with their underlying parametric templates, which provide correspondences for us to assemble them with the body. Using these correspondences, we pre-compute the transformation between the body space and the hand/face space. We also pre-define the blending weights on the SMPL-X mesh, as shown in Figure 2. Given a point $\boldsymbol{p}$ in the body posed space, we first project it onto the SMPL-X mesh and query the blending weight ω. Without loss of generality, let's say the point falls on the left hand. Then we query its SDF value and color in both the body representation and face representation, which we denote as s_{B}, $\boldsymbol{c}_{\text{B}}$, s_{H} and $\boldsymbol{c}_{\text{H}}$. The final SDF value and color for $\boldsymbol{p}$ is computed as:

$$s = \begin{cases} \omega s_{\text{H}} + (1-\omega)s_{\text{B}}, & \text{if } -0.05 < s_{\text{H}} < 0.025 \\ s_{\text{B}}, & \text{otherwise} \end{cases}. \tag{14}$$

$$\boldsymbol{c} = \begin{cases} \omega \boldsymbol{c}_{\text{H}} + (1-\omega)\boldsymbol{c}_{\text{B}}, & \text{if } -0.05 < s_{\text{H}} < 0.025 \\ \boldsymbol{c}_{\text{B}}, & \text{otherwise} \end{cases}. \tag{15}$$

In this way, we can obtain the final avatar given some specific body poses, hand poses and facial expressions.

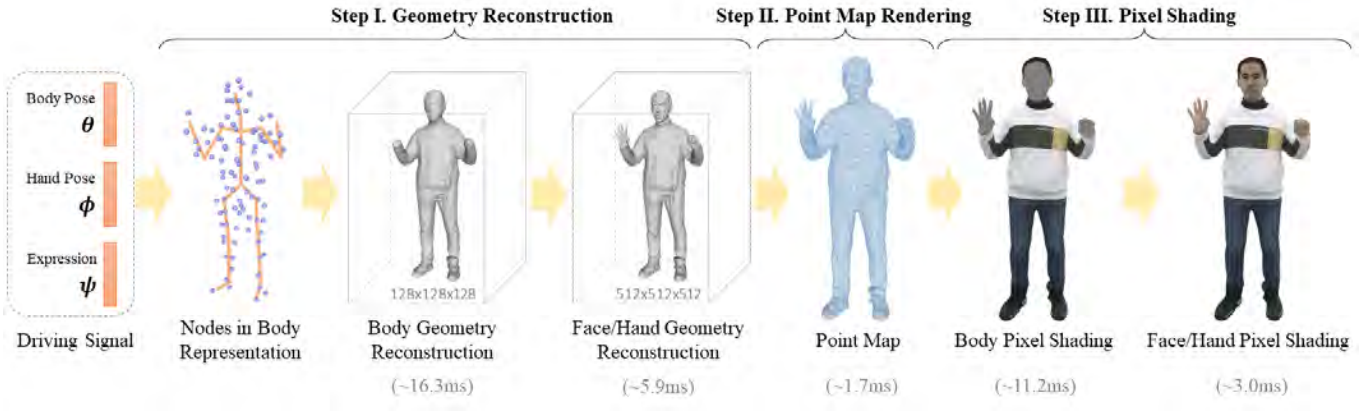


Fig. 5. **Illustration of our real-time rendering pipeline.** Our rendering pipeline firstly reconstructs the geometry model in the form of an SDF volume, from which we render a point map to query the pixel colors given a specific viewpoint.

4 REAL-TIME ANIMATION AND RENDERING

Since our avatar representation is based on NeRF rather than meshes, it cannot be sent to traditional graphics pipelines for efficient rendering. In fact, efficient rendering of NeRF is a difficult problem on its own, and has attracted significant research interest since the debut of NeRF [Chen et al. 2022a; Liu et al. 2022; Muller et al. 2022; Reiser et al. 2021; Yu et al. 2021]. However, current NeRF acceleration techniques mostly focus on static scenes or dynamic sequence playback, and rely on specific data structures that cannot be easily adapted for avatar animation. Therefore, we design a dedicated deferred surface rendering pipeline for our avatar representation. Our deferred surface rendering consists of three steps, namely geometry reconstruction, point map rendering and pixel shading, as illustrated in Figure 5 and described in the following.

Step I. Geometry Reconstruction. In the first step of our deferred rendering pipeline, we reconstruct the complete geometry model in the form of an SDF volume. Note that this step is agnostic to the viewing camera. For the reason of computational efficiency, we do not directly infer a high-resolution SDF volume using the full geometry field in Equantion (14). Instead, we follow a coarse-to-fine scheme: we firstly infer only the body geometry field with a low-resolution volume, and then update the hands and the face in the upsampled one.

Specifically, we first compute the bounding box from the node positions in the body representation. At the center of the bounding box, we create a volume grid with a resolution of 128×128×128, where the side length of each voxel is 2 cm. Such a resolution is not fine enough for representing thin structures like fingers, but sufficient for other major body parts. Recall that in our body representation in Section 3.1, a local network $\mathcal{G}_i$ only influences a small local space around node i. Based on this property, we directly collect the voxels that fall into the influence space of each node, and pass them through the corresponding local networks in parallel. Since the influence radiuses of all local networks are identical, the voxel numbers assigned to the local networks are similar, which is beneficial for maximizing parallelism. The outputs of the local networks are then gathered and blended following Equation (3), and finally decoded to SDF values.

After that, we upsample the SDF volume by a factor of 4, and refine the hands and the face. To this end, we firstly compute the bounding boxes for both hands and the face, and collect the voxels that locate inside the boxes. Then we query the hand geometry field or the face geometry field for these voxels, depending on the bounding boxes they belong to. The final SDF values of these voxels is obtained using the method described in Section 3.4.

Step II. Point Map Rendering. The second step of our rendering pipeline is to raycast the SDF volume to extract views of the implicit surface [Izadi et al. 2011]. Given the camera origin and viewing direction, the raycasting operation traverses the SDF volume along camera rays and extracts the intersection between the rays and the zero-crossing surface for each pixel. The intersection point is then recorded as a 2D point map for next step.

Step III. Pixel Shading. In the final step of our rendering pipeline, we compute the pixel colors of the point map to obtain the final rendering results. Similar to geometry reconstruction, we first query the body color field for all the points in the point map and utilize the "local influence" property of the local MLPs to parallelize network queries. After that, we query the face color field and the hand color field to update the pixels of the hands and the face, as described in Section 3.4. This produces the final image that corresponds to the given body pose, hand pose, facial expression and view point, marking the end of our rendering pipeline.

Most NeRF-based methods synthesize images through *volume rendering*, which samples millions of points along camera rays for network evaluation. In contrast, we design a *deferred surface rendering* pipeline by taking advantage of the implicit surface definition and the disentanglement of geometry and appearance [Sun et al. 2022a]. Our pipeline first stores geometry reconstruction in an explicit grid and then use it to cull unnecessary points for color inference as much as possible. In this way, we significantly speed up the rendering process of our expressive avatar and finally achieve real-time performance using custom CUDA kernels and modern inference engines like NVIDIA TensorRT. Note that this is only

possible when the surface prediction is reliable. In the next section, we will discuss how we train the network from scratch to acquire the surface, and how we use surface rendering to boost network learning.

5 MODEL TRAINING

In this section, we provide technical details about how we train our expressive avatars. We first present our novel two-pass training strategy in Section 5.1. We then shortly describe our capture setup and the data pre-processing pipeline, followed by the full training procedure (Section 5.2)

5.1 Network Training Strategy

Similar to other NeRF-based methods, we can train our networks by sampling image pixels, shooting camera rays, performing volume rendering and penalizing the error between the rendered colors and the ground-truth ones [Mildenhall et al. 2020]. However, we empirically find that the results produced by such a training strategy are not satisfactory enough: the networks trained with this scheme tend to synthesize over-smoothed images and fail to recover dynamic appearance details like the texture patterns on the garments. This is mainly because volume rendering relies on dense network evaluation along camera rays, which limits the number of pixel queries in one forward pass. Consequently, we could only apply pixel-wise supervision on sparse pixels during network training. In addition, this training strategy is based on volume rendering, rather than surface rendering that we use for real-time animation. The gap between training and testing further deteriorates the rendering quality of our real-time system. To address these issues, we propose a two-pass training strategy. It consists of two stages, namely topology-free training and topology-based finetuning, as illustrated in Figure 6 and elaborated in the following.

Pass I. Topology-free training. In this stage we assume no prior knowledge about the avatar topology and directly train the full network with volume rendering. To this end, we follow the practice of [Yariv et al. 2021] and convert the SDF into a density function using:

$$\sigma(\boldsymbol{p}) = \frac{1}{\gamma}\Phi_\gamma(-\mathcal{G}_*(\boldsymbol{p})), \tag{16}$$

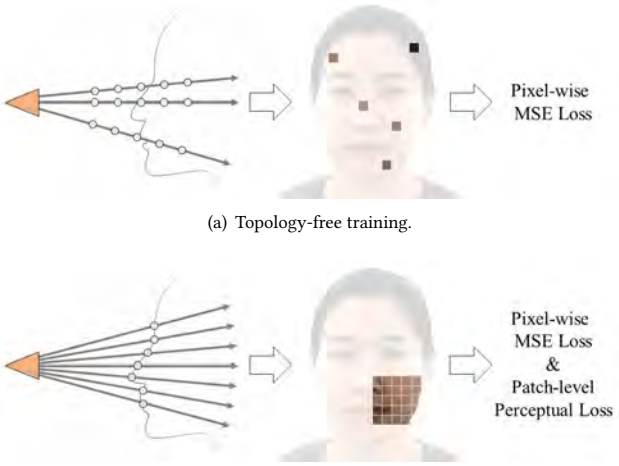
where $\Phi_\gamma(\cdot)$ is the Cumulative Distribution Function (CDF) of the Laplace distribution with zero mean, and its scale γ is a learnable hyperparameter. This transformation links the density in neural radiance fields with our geometry definition, allowing us to optimize the avatar SDF using neural volume rendering. At each iteration of network optimization, we randomly fetch a batch of frames, from which we randomly sample a fixed number of pixels to construct the training loss. The loss is defined as:

$$\begin{aligned}\mathcal{L}_\mathrm{I} =& \mathcal{L}_\mathrm{rgb} + \lambda_\mathrm{mask}\mathcal{L}_\mathrm{mask} + \lambda_\mathrm{Eikonal}\mathcal{L}_\mathrm{Eikonal} + \\ & \lambda_\mathrm{node}\mathcal{L}_\mathrm{node} + \lambda_\mathrm{ebd}\mathcal{L}_\mathrm{ebd} + \lambda_\mathrm{KL}\mathcal{L}_\mathrm{KL},\end{aligned} \tag{17}$$

where $\mathcal{L}_\mathrm{rgb}$ measures the MSE between the rendered and true pixel colors, $\mathcal{L}_\mathrm{mask}$ is an MAE loss supervising the occupancy values of the rendered pixels, $\mathcal{L}_\mathrm{Eikonal}$ is the Eikonal loss encouraging the geometry fields to approximate a true signed distance function [Yariv et al. 2021], $\mathcal{L}_\mathrm{node}$, $\mathcal{L}_\mathrm{ebd}$ and $\mathcal{L}_\mathrm{KL}$ are the regularization losses inherited from [Zheng et al. 2022b]. To stabilize network training and obtain consistent performance for different subjects, we do not optimize γ in Equation (16) during the training process. Instead, we manually set its value as:

$$\gamma(n) = \gamma_0 + \gamma_1 \cdot \max\{0, (1 - n/N)\}, \tag{18}$$

where n is the iteration steps, while $\gamma_0 = 5 \times 10^{-4}$, $\gamma_0 = 0.02$ and $N = 100000$ are hyperparameters controlling how γ varies as training iterates.



(a) Topology-free training.

(b) Topology-based finetuning.

Fig. 6. **Illustration of the two-pass training strategy.** In the topology-free training stage, the networks learn the avatar geometry from scratch with volume rendering on sparse pixels. Then the topology-based finetuning stage utilizes the learned geometry to increase the number of pixel evaluations in one forward pass, allowing us to apply structural supervision with a patch-level perceptual loss.

Pass II. Topology-based finetuning. After the first stage of training, we observe the estimate of the avatar shape is sufficiently reliable although the texture is not clear enough. Based on this observation, we design a topology-based finetuning strategy to finetune the color fields in our avatar while keeping the geometry branches fixed. Specifically, we sample a set of patches with the size of $H \times H$ on a training frame, and compute the corresponding camera locations and ray directions. Then we leverage ray marching to locate the nearest surface intersection between the rays and the underlying implicit surface, similar to the first two steps in our real-time rendering pipeline. The intersection points are sent to the color networks to query their RGB values, which we take as the pixel color prediction to construct the training loss. Different from the first stage, the training loss in this stage is defined as:

$$\mathcal{L}_\mathrm{II} = \mathcal{L}_\mathrm{MSE} + \lambda_\mathrm{LPIPS}\mathcal{L}_\mathrm{LPIPS}, \tag{19}$$

where $\mathcal{L}_\mathrm{MSE}$ is a simple L2 loss to match pixel-wise appearance with the ground-truth and $\mathcal{L}_\mathrm{LPIPS}$ is a perceptual loss applied on patch level, which provides stronger supervision on high-frequency appearance details as shown in Figure 17. We choose VGG as the backbone of LPIPS loss.

Note that we only apply the training loss on the ray-surface intersection points; we do not back-propagate the loss gradient through the ray marching process. Compared to the previous stage where dense point samples are required for pixel color computation, the topology-based finetuning stage predicts pixel colors with a much smaller number of points to be evaluated. This advantage increases the maximum number of pixel evaluations in one forward pass, thus enabling perceptual supervision on image patches. Furthermore, this finetuning stage is built upon surface rendering, similar to the real-time animation system in Section 4. As a result, it closes the gap between training and testing.

One may ask why not apply perceptual loss in the first training pass based on volume rendering. In fact, this is possible, as done in HumanNeRF [Weng et al. 2022]. However, due to the complexity of our network and the memory limits of GPUs, we can only render 400 pixels in one forward pass with volume rendering. Consequently, the patch-level supervision can only be applied on a small patch with a resolution around 20×20, which we found too small to contain any structural information. In contrast, topology-based surface rendering allows us to render a larger patch at 128×128 resolution, enabling stronger supervision on structural accuracy.

5.2 Data Capture and Processing

5.2.1 Data Capture. We use two multi-camera capture systems in this work, one for the full body and the other for the face. The body capture system consists of 16 synchronized cameras, each capable of producing 1500×2048 images. The cameras are evenly distributed around the yaw axis in order to cover the 360-degree view point of the body. For the face capture system, we use 6 synchronized cameras that focus on the frontal face, spanning about 130 degrees horizontally. The image resolution of our face capture system is also 1500×2048.

We collect data from 4 subjects, two males and one female in two sets of clothing. For each subject, we use the full-body capture system to collect video data of some casual motions with neutral face expression, which will be used to train the body avatar and the hand avatar. Each captured sequence is about 2000 frames in length. To collect training data for face avatar, we use the face system to capture sequences of 1500-2000 frames in length, which include a range of expressions and natural reading sequences. Note that we do not rely on either pre-scanned templates or non-rigid mesh tracking to learn the avatar geometry, which is a departure from existing work [Bagautdinov et al. 2021; Habermann et al. 2021; Remelli et al. 2022]

5.2.2 Data Pre-processing. To train our networks, we need to obtain SMPL-X fitting for every frame of the full-body sequence and Faceverse fitting for both the full-body sequence and the face sequence. For the former purpose, we directly use off-the-shelf tools [Pavlakos et al. 2019; Zhang et al. 2023b]. However, the fitting results obtained using such a method are not accurate enough due to the sparseness of keypoints and inevitable detection errors. Therefore, we regard the results as the initial fitting and design a refinement step based on inverse rendering. Specifically, we use Background Matting v2 [Lin et al. 2021] to extract the foreground body segmentation and render the silhouette of SMPL-X model with a differentiable rasterizer [Liu et al. 2019a]. Then we penalize the inconsistency between them to optimize both the shape and pose parameters for SMPL-X. As shown in Figure 7, the refinement step leads to better alignment between the SMPL-X model and the image observation. Please see the appendix for more details.

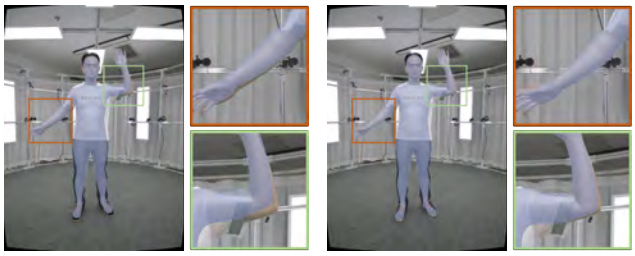


Fig. 7. **Effect of fitting refinement.** Compared to the initial results (left), the refinement step can produce better alignment between the SMPL-X's arms and their image observations (right).

(a) Without tone mapping. (b) With tone mapping.

Fig. 8. **Effects of the tone mapping network.** Our tone mapping networks addresses the skin tone difference between body data and face data, and successfully harmonizes the composition results.

To register Faceverse, we directly utilize the open-sourced tool provided by Wang et al. [2022a]. We also apply Background Matting V2 [Lin et al. 2021] for foreground segmentation.

5.2.3 Full training procedure. After collecting the training data for a specific person, we first use the face data to train the networks for face representation with our two-pass training strategy. Then we fix the face networks, and jointly train the body and the hand networks using the body data. This training step also follows the two-pass strategy. To address the color tone difference between the body capture and the facial one, we additionally introduce an tone mapping network to adjust the output of the face color field, as shown in Figure 8. It is a tiny MLP that takes as input the viewing directions in both the body capture system and the face system, and outputs a mapping matrix $\mathbf{M} \in \mathbb{R}^{3\times3}$ for tone mapping. The full training pipeline takes about 3 days on one NVIDIA GeForce RTX 3090 GPU with 24 GB GPU Memory, and we report the detailed training statistics in Table 1. Note that the training cost of our method is much lower than existing works that spends two weeks on hundreds of GPUs to build a full-body avatar [Bagautdinov et al. 2021]. Please refer to the appendix for more training details.

Fig. 9. **Qualitative results on novel pose synthesis.** We train our network for four identities and show the novel pose synthesis results, where two different subjects perform the same motions and expressions.

Table 1. **Training statistics of our method.** We report the number of training iterations, the elapsed training time and the number of network parameters that will be updated for each training step.

Part	Training Pass	#Iter.	Time	#Param. to Update
Face	Pass I	200k	~14 h	1.76M
	Pass II	100k	~3 h	1.10M
Body+Hands	Pass I	300k	~48 h	18.55M
	Pass II	100k	~3 h	17.57M

Fig. 10. **Real-time avatar manipulation.** We present an application of interactive avatar manipulation to demonstrate the real-time rendering capability of our avatar.

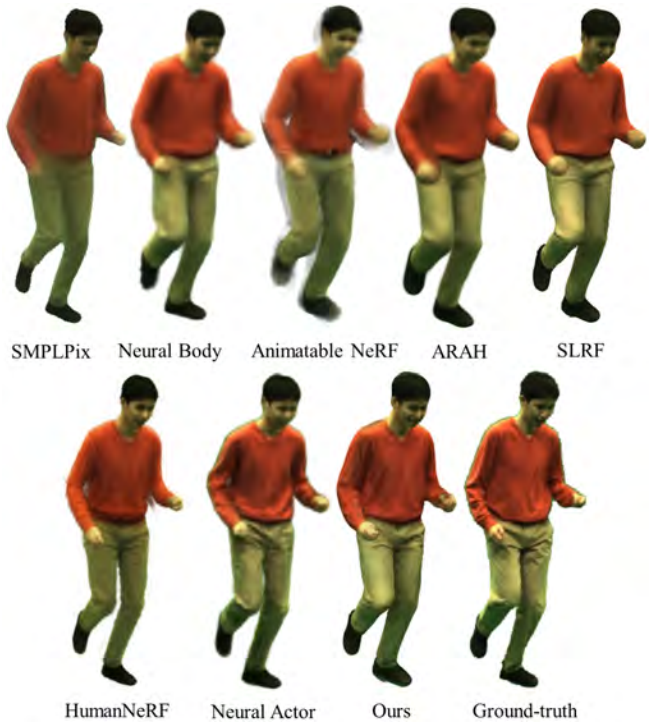


Fig. 11. **Qualitative comparison against state-of-the-art body avatars.** Given an unseen body pose, our method is able to generate high-quality appearance details. The synthesis quality is comparable with Neural Actor, while outperforming other baselines.

6 EXPERIMENTS

This section provides an experimental evaluation of our system for creating expressive full-body avatars that can be rendered at a real-time framerate. We first present qualitative results on different identities in Section 6.1, followed by a comparison against state-of-the-art methods in Section 6.2. Finally, we study the components of our method in Section 6.3.

6.1 Results

As formulated in Section 3, the input to our avatar is a set of driving signals consisting of body poses, hand poses and facial expressions. Therefore, we can directly manipulate the pose parameters and expression coefficients to animate our avatar, or in other word, to synthesize novel poses and expressions. In Figure 1 and Figure 9, we present some animation examples, where two or more identities follow the same driving signals. The results cover various garment types, cloth materials, body motions and expressions. As we can see from these results, our method supports simultaneous control of the body, the face and the hands together. The results of the male in white long sleeves show that our method can generate photo-realistic dynamic appearance details for different body poses, while the results of another male in short pants show that our method can gracefully handle the relative motion between the legs and the shorts. In addition, we demonstrate the results of a female in different clothing, and prove that our method is more flexible in modeling different cloth shapes and able to handle the garment type that is not topologically similar to the naked body. Furthermore, our avatars can be animated and rendered in real time on a modern high-end graphics card, *e.g.*, 25 FPS at a resolution of 1024×1024 on an NVIDIA GeForce RTX 3090 GPU. To confirm its real-time animation capability, we develop a simple application which allows users to manipulate the body motion, the hand gesture or the facial expression, and the manipulated results can be visualized from any viewpoints in real time, as presented in Figure 10. We encourage readers to see our supplemental video for better visualization.

6.2 Comparison

To validate our method, we compare with recent state-of-the-arts on novel pose synthesis. Unfortunately, existing baselines only model the clothed body while neglecting other fine-grained parts like the hands and the face. For a fair comparison, we also remove the hands and the face in our representation, leaving the body only for comparison. We mainly compared with the following methods:

- **Neural Body** [Peng et al. 2021b]. Neural Body attaches learnable latent codes to the vertices of SMPL model, and employs

Table 2. **Quantitative comparison with state-of-the-art body avatars.** To ease reading, we highlight the best scores with orange shading, and the second best with light orange.

Methods	PSNR ↑	LPIPS ↓	FID ↓	Framerate ↑
SMPLPix	23.199	0.050	42.832	36 fps
Neural Body	23.946	0.096	81.527	0.32fps
Animatable NeRF	22.453	0.097	76.258	0.54fps
ARAH	22.070	0.092	104.330	0.07fps
SLRF	23.363	0.051	58.038	0.16fps
HumanNeRF	23.395	0.054	39.014	0.14fps
Neural Actor	23.531	0.066	19.714	0.25fps
Ours	23.709	0.044	30.860	25 fps

sparse 3D convolutions to diffuse the latent codes into a radiance field in the 3D space.

- **Animatable NeRF** [Peng et al. 2021a]. Animatable NeRF factorizes a deforming human body into a canonical radiance field and a deformation field that establishes correspondences between the observations and the canonical space. The deformation field is generated from the backward skinning motion of the underlying SMPL model.
- **HumanNeRF** [Weng et al. 2022]. HumanNeRF follows a similar scheme to Animatable NeRF, and introduces an additional non-rigid motion field to better handle large motions like dancing. Additionally, HumanNeRF also introduce perceptual supervision on image patches to facilitate appearance learning.
- **ARAH** [Wang et al. 2022b]. ARAH also models a dynamic humans with a canonical field and a motion field, but it uses forward LBS root finding to model the motion field.
- **SLRF** [Zheng et al. 2022b]. SLRF models the clothed body with a set of structured local radiance fields, which are loosely attached to the SMPL model. It is the most related work to ours; in fact, our body representation is built upon it. Compared to SLRF, our method not only introduces dynamic feature patches to local fields, but also disentangles geometry and appearance. Both modification are of significant importance for learning high-frequency appearance details.
- **Neural Actor** [Liu et al. 2021]. Similar to Animatable NeRF and ARAH, Neural Actor also use a canonical radiance field, but it encodes appearance features on the 2D texture maps of the SMPL model to better capture dynamic details.
- **SMPLpix** [Prokudin et al. 2021]. Unlike the baselines mentioned above, SMPLpix is a 2D technique that uses neural rendering in the image space. It works by rendering the SMPL vertices onto the image plane and subsequently converting the rendered image into a final RGB image using a neural network based on 2D UNet. Despite its simplicity and speed, such a scheme suffers from view inconsistencies and temporal jittering.

The dataset we use for comparison is the "Lan" sequence from DeepCap [Habermann et al. 2020]. It has 33,605 training frames and 23,062 testing frames captured from 11 cameras, covering a large variety of body motions. The resolution of its image frames is 1024×1024. We use the original training/testing split in the dataset for evaluation, and follow the same protocol in [Liu et al. 2021]. The results of Neural Actors are borrowed from [Liu et al. 2021], while the others are evaluated by ourselves using the original training settings in their implementation.

To measure the quality of novel pose synthesis, we adopt three widely used metrics, namely peak-to-signal ratio (PSNR), learned perceptual image patch similarity (LPIPS) [Zhang et al. 2018] and the

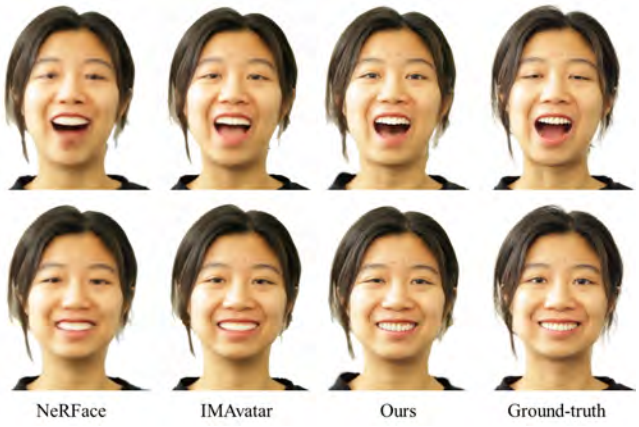


Fig. 12. **Qualitative comparison against state-of-the-art facial avatars.** Compared to existing methods, ours can generate sharper appearance details like the teeth.

Table 3. **Quantitative comparison with state-of-the-art facial avatars.** To ease reading, we highlight the best scores with orange shading, and the second best with light orange.

Methods	PSNR ↑	LPIPS ↓	FID ↓
NeRFace	19.816	0.133	51.182
IMAvatar	21.220	0.092	46.015
Ours	19.608	0.089	33.685

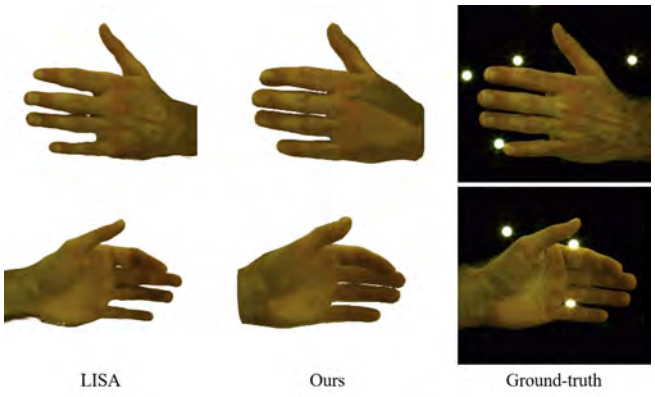


Fig. 13. **Qualitative comparison against LISA [Corona et al. 2022].** As shown in the figure, LISA performs better than our method in terms of recovering appearance details like the veins.

(a) Without feature patches or finetuning. (b) With feature patches, without finetuning. (c) Full model.

Fig. 14. **Effects of dynamic feature patches and topology-based finetuning on the body model.** Compared to the baseline in (a), assigning a dynamic feature patch leads to stronger capability of detail representation (b), which is further improved by our two-passed training strategy (c).

Frechet inception distance (FID) [Heusel et al. 2017]. PSNR simply measures pixel-level accuracy, while LPIPS and FID are more similar to human perception. The numeric results are reported in Table 2. It is easy to see that our method can synthesize image with quality on par to state-of-the-art baselines. In Figure 11, we show that our results are comparable with Neural Actor while outperforming other methods in terms of capturing appearance details. This may be attributed to the fact that both Neural Actor and our approach utilize an explicit feature grid to predict high-frequency details, while the remaining methods (Neural Body, Animatable NeRF, ARAH, SLRF, and HumanNeRF) solely rely on MLPs to learn the appearance. Although the image produced by HumanNeRF shows that perceptual supervision could alleviate this issue, the outcome remains unclear. It is worth noting that Neural Actor performs well only for tight garments due to its reliance on SMPL topology, whereas our local feature patches are more flexible, as we have demonstrated in Section 6.1. Furthermore, our method renders images faster than other NeRF-based methods by two orders of magnitude.

Apart from the above comparisons, we also conduct evaluation on our face and hand representations for completeness. Specifically, to validate our face representation, we compare with NeRFace [Gafni et al. 2021] and IMAvatar [Zheng et al. 2022a], two state-of-the-art baselines on facial avatars. Both baselines and our method are based on implicit representations. Among them, NeRFace [Gafni et al. 2021] extends the vanilla NeRF by directly taking the expression coefficients as the additional inputs, while IMAvatar [Zheng et al. 2022a] incorporates skinning fields with an implicit representation, leading to better geometry reconstruction and stronger generalization capability. The data we use for this experiment is a monocular video sequence released by Zheng et al. [2022a]. It contains 2904 training frames and 1825 testing frames, with a resolution of 512×512. We follow the same evaluation protocal as IMAvatar, and use FLAME [Li et al. 2017] as the base 3DMM model for both baselines and our method.

The qualitative comparison is presented in Figure 12. Although all methods perform well at synthesizing different expressions, our method generates more appearance details such as teeth compared to the baselines. This is due to the feature tri-planes in our method, which provide stronger power for encoding spatially-varying signals. Additionally, our two-pass training strategy encourages the network to fully utilize this advantage. The numeric evaluation reported in Table 3 confirms that our method is superior to the baselines. The results also show that collecting multi-view video is not a necessity for our facial representation; given monocular input, our method can still learn a photo-realistic facial avatar.

To evaluate our hand representation, we qualitatively compare with LISA [Corona et al. 2022] on the dataset from InterHand2.6M [Moon et al. 2020]. LISA is a state-of-the-art method for hand avatars that models implicit shape and appearance of hands through a collection of rigid parts defined by the hand bones. As demonstrated in Figure 13, LISA outperforms our current approach by recovering more appearance details on the hand skin. However, this comes at

Table 4. **Quantitative evaluation of the dynamic feature patches and topology-based finetuning on the body model.** The quantitative results are in line with the findings in Figure 14.

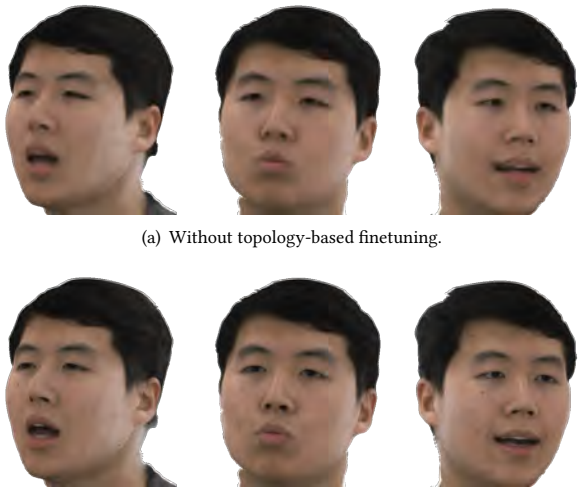
	PSNR ↑	LPIPS ↓
Without feature patches or finetuning	25.225	0.079
With feature patches, without finetuning	25.604	0.068
Full model	25.731	0.056

Fig. 15. **Visualization of the effects of dynamic feature patches.** Top: Avatar animation and rendering results. Bottom: avatar rendering with dynamic feature patches under the canonical A-pose.

the expense of a higher computational load. In the future, we can integrate LISA into our framework to enhance the quality of our hand representation.

6.3 Ablation Study

Here we conduct evaluation on the effects of our novel technical components. First we analyze the effects of the dynamic feature patches in Section 2. In our method, we introduce a dynamic feature patch for each local color field C_i, and extract feature vectors for the points in the local space of this field. We compare with a baseline that uses purely MLPs to model the local color field without any explicit feature grids, and present the results in Figure 14 (a, b) and Table 4. We can see that the explicit feature patches allows the network to learn more high-frequency details compared to the baseline. To better understand what the dynamic feature patches encode, we conduct an additional visualization experiment in Figure 15, where the avatar are rendered with different feature patches under the same A-pose. As we expect, the feature patches successfully learn the dynamic wrinkles of the garment.



(a) Without topology-based finetuning.

(b) With topology-based finetuning.

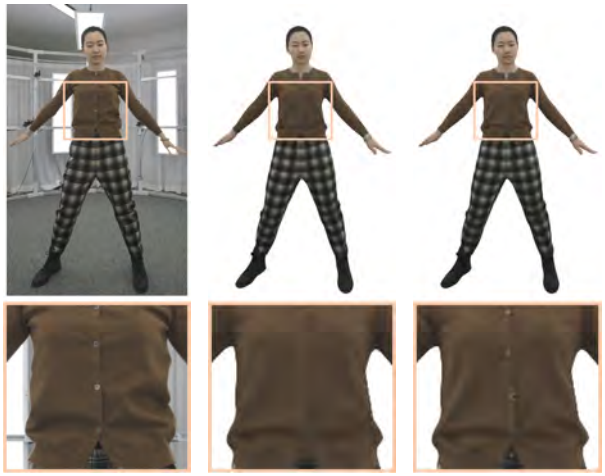
Fig. 16. **Effects of topology-based finetuning on the facial area.** The two-pass training strategy force the face network to learn more photo-realistic facial appearance.

Table 5. **Quantitative evaluation of topology-based finetuning on the facial area.** The quantitative results further confirm that topology-based finetuning leads to perceptually better rendering.

	PSNR ↑	LPIPS ↓
Before finetuning	21.109	0.202
After finetuning	21.197	0.137

Next we validate the effectiveness of our two-pass training strategy. In this training strategy, the first pass is the vanilla training process in most NeRF-based methods, while the second pass, *i.e.*, topology-based finetuning, is our contribution. To evaluate its effect, we compared with the results with one-pass training only. As shown in Figure 14 (b, c), the topology-based finetuning step force the network to learn more photo-realistic appearance details. This is because the reconstruction loss in the first pass, either in the form of $\ell 1$ or $\ell 2$, only penalizes pixel-wise error and ignores structural quality. In contrast, the patch-level perceptual loss used in the finetuning step is more sensitive to structure errors and matches human perception better. We conduct similar experiments on the facial area in Figure 16 and Table 5, which further proves the effectiveness of our two-pass training strategy.

Finally we evaluate the role of perceptual loss in the topology-based finetuning pass. We conduct another experiment, where only a simple MSE loss is applied for topology-based finetuning. As presented in Figure 17, this experiment confirms again that patch-level supervision is necessary for learning sharp details. Without it, the model only learns blurry appearance.



(a) Ground-truth. (b) Without $\mathcal{L}_{\text{LPIPS}}$. (c) With $\mathcal{L}_{\text{LPIPS}}$.

Fig. 17. **Effects of the perceptual loss $\mathcal{L}_{\text{LPIPS}}$ in topology-based fine-tuning.** Patch-level supervision with the perceptual loss leads to better recovery of high-frequency details like the wrinkles and the buttons.

7 LIMITATION AND FUTURE WORK

Although the results produced by our method is mostly photo-realistic, artifacts occurs occasionally. There are two main factors behind the visual artifacts in our results. Firstly, our method reconstructs a coarse SDF volume as the scaffold for rendering. Its oversmooth nature and inevitable errors result in stitching texture and the artifacts around boundaries when performing ray-casting in the volume. Secondly, our method models the articulated motion around joints through learning the assembling of local fields, instead of relying on full surface of SMPL-X. This may lead to the occasional artifacts around body joints.

Currently our method uses one single model to represent the whole clothed human body, no matter how many layers of garments the actor actually wears. Although this is a convenient choice and has been widely adopted in previous work [Bagautdinov et al. 2021; Liu et al. 2021; Peng et al. 2021a; Wang et al. 2022b; Weng et al. 2022; Zheng et al. 2022b], it may lead to ghosting effects along the boundary between clothing and skin or between upper and lower garments. For our method, these artifacts is mainly caused by the fact that some local fields happen to fall near the garment boundaries and consequently model two cloth layers all together. It will result in more noticeable artifacts when the cloth is loose and contains more dynamic deformations. For future work, we could replace our unified body representation with a multi-layer one and model different cloth layers separately as in [Xiang et al. 2021].

Another limitation of our method is about lighting and self-shadowing effects. In particular, when limbs interact with each other or occlude the torso, they cast shadows. However, we do not explicitly address this issue and fully rely on the network to learn the self-shadowing effects as pose-dependent appearance, which adversely impacts generalization. In fact, modeling the interactions between lighting and objects in neural radiance fields is still an area of active research [Chen and Liu 2022; Srinivasan et al. 2021; Zhang et al. 2021], and we could employ similar techniques from these works and enable avatar relighting under new illumination. Moreover, the SDF fields in our avatar representation also provide a coarse estimation of the underlying avatar geometry, which we could also utilize to approximate the self-shadowing effects as done in [Bagautdinov et al. 2021].

Creating full-body avatars involves the modeling of many components, including the body, the face, hands, garments, hairs, eyes, teeth, soft tissues and accessories. This work only considers three dominant parts (i.e., the major body, hands and the face) and is unable to model the components all at once. In addition, the complex dynamics of loose garments, hairs or soft tissues require more sophisticated modeling technology that is beyond the scope of this paper. Therefore, we leave them as future work.

Potential Social Impact. Our method enables automatic creation of a digital "twin" of any individual. This poses a risk of misusing the technology to re-target individuals with poses or actions they never actually perform. To prevent such misuse, it is essential to exercise caution before deploying the technology. Several techniques could be adopted to mitigate this risk. For example, active watermarking of generated content can be employed to detect unauthorized use [Luo et al. 2020]. Moreover, forgery detection technology can be utilized to identify manipulated or synthetic imagery in fake videos [Dong et al. 2022a; Wang et al. 2022c]. By taking these measures, we hope that our technology is used responsibly.

8 CONCLUSION

We have presented AvatarReX, a new method for learning full-body avatars from multi-view video data. Compared to existing works, our avatar has two advantages: for one thing, our avatar supports expressive control of the body, hands and the face together; for another, our avatar can be animated and rendered at a real-time framerate with our dedicated rendering pipeline. This is achieved by a compositional representation with the disentanglement of geometry and appearance. Moreover, we introduce a two-pass training strategy that incorporates surface rendering and patch-level perceptual supervision to further improve the appearance quality. In our experiments, we showcase the capabilities of our method by demonstrating its synthesis results given novel poses and expressions, showing its great potential in many interactive applications.

ACKNOWLEDGMENTS

This paper is supported by National Key R&D Program of China (2022YFF0902200), the NSFC project No.62125107 and No.61827805

REFERENCES

Oleg Alexander, Mike Rogers, William Lambeth, Jen-Yuan Chiang, Wan-Chun Ma, Chuan-Chang Wang, and Paul E. Debevec. 2010. The Digital Emily Project: Achieving a Photorealistic Digital Actor. *IEEE Computer Graphics and Applications* 30, 4 (2010), 20–31. https://doi.org/10.1109/MCG.2010.65

Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. 2018a. Detailed Human Avatars from Monocular Video. In *3DV*.

Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. 2018b. Video Based Reconstruction of 3D People Models. In *CVPR*.

Timur Bagautdinov, Chenglei Wu, Tomas Simon, Fabián Prada, Takaaki Shiratori, Shih-En Wei, Weipeng Xu, Yaser Sheikh, and Jason Saragih. 2021. Driving-Signal Aware Full-Body Avatars. 40, 4 (2021).

Volker Blanz and Thomas Vetter. 1999. A Morphable Model for the Synthesis of 3D Faces. In *SIGGRAPH*, Warren N. Waggenspack (Ed.). ACM, 187–194.

Andrei Burov, Matthias Nießner, and Justus Thies. 2021. Dynamic Surface Function Networks for Clothed Human Bodies. (2021).

Chen Cao, Tomas Simon, Jin Kyu Kim, Gabe Schwartz, Michael Zollhoefer, Shun-Suke Saito, Stephen Lombardi, Shih-En Wei, Danielle Belko, Shoou-I Yu, Yaser Sheikh, and Jason Saragih. 2022. Authentic Volumetric Avatars from a Phone Scan. *ACM Trans. Graph.* 41, 4, Article 163 (jul 2022), 19 pages.

Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. 2014. FaceWarehouse: A 3D Facial Expression Database for Visual Computing. *IEEE TVCG* 20, 3 (2014), 413–425.

Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. 2021. Efficient Geometry-aware 3D Generative Adversarial Networks. In *arXiv*.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. 2022b. Tensorf: Tensorial radiance fields. In *ECCV*.

Xu Chen, Yufeng Zheng, Michael J Black, Otmar Hilliges, and Andreas Geiger. 2021. SNARF: Differentiable Forward Skinning for Animating Non-Rigid Neural Implicit Shapes. In *ICCV*.

Zhiqin Chen, Thomas A. Funkhouser, Peter Hedman, and Andrea Tagliasacchi. 2022a. MobileNeRF: Exploiting the Polygon Rasterization Pipeline for Efficient Neural Field Rendering on Mobile Architectures. *CoRR* abs/2208.00277 (2022). https://doi.org/10.48550/arXiv.2208.00277

Zhaoxi Chen and Ziwei Liu. 2022. Relighting4D: Neural Relightable Human from Videos. In *ECCV*.

Hang Chu, Shugao Ma, Fernando De la Torre, Sanja Fidler, and Yaser Sheikh. 2020. Expressive Telepresence via Modular Codec Avatars. In *ECCV (Lecture Notes in Computer Science, Vol. 12357)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 330–345.

Enric Corona, Tomas Hodan, Minh Vo, Francesc Moreno-Noguer, Chris Sweeney, Richard Newcombe, and Lingni Ma. 2022. LISA: Learning Implicit Shape and Appearance of Hands. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 20533–20543.

Radek Danecek, Michael J. Black, and Timo Bolkart. 2022. EMOCA: Emotion Driven Monocular Face Capture and Animation. In *CVPR*. IEEE, 20279–20290.

Junting Dong, Qi Fang, Yudong Guo, Sida Peng, Qing Shuai, Xiaowei Zhou, and Hujun Bao. 2022b. TotalSelfScan: Learning Full-body Avatars from Self-Portrait Videos of Faces, Hands, and Bodies. In *NeurIPS*.

Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Ting Zhang, Weiming Zhang, Nenghai Yu, Dong Chen, Fang Wen, and Baining Guo. 2022a. Protecting Celebrities from DeepFake with Identity Consistency Transformer. In *CVPR*. IEEE, 9458–9468.

Jiemin Fang, Taoran Yi, Xinggang Wang, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Matthias Nießner, and Qi Tian. 2022. Fast Dynamic Radiance Fields with Time-Aware Neural Voxels. In *arXiv preprint arXiv:2205.15285*.

Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. 2021. Learning an animatable detailed 3D face model from in-the-wild images. *TOG* 40, 4 (2021), 88:1–88:13.

Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. 2022. Plenoxels: Radiance fields without neural networks. In *CVPR*. 5491–5500.

Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2021. Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction. (2021).

Xuan Gao, Chenglai Zhong, Jun Xiang, Yang Hong, Yudong Guo, and Juyong Zhang. 2022. Reconstructing Personalized Semantic Facial NeRF Models From Monocular Video. *SIGGRAPH Asia* 41, 6 (2022).

Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. 2022. Neural head avatars from monocular RGB videos. In *CVPR*. 18653–18664.

P. Guan, L. Reiss, D. Hirshberg, A. Weiss, and M. J. Black. 2012. DRAPE: DRessing Any PErson. *ACM TOG (Proc. SIGGRAPH)* 31, 4 (July 2012), 35:1–35:10.

Yudong Guo, Lin Cai, and Juyong Zhang. 2021a. 3D Face From X: Learning Face Shape From Diverse Sources. *IEEE TIP* 30 (2021), 3815–3827.

Yudong Guo, Keyu Chen, Sen Liang, Yong-Jin Liu, Hujun Bao, and Juyong Zhang. 2021b. AD-NeRF: Audio Driven Neural Radiance Fields for Talking Head Synthesis. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 5764–5774.

Marc Habermann, Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. 2021. Real-time Deep Dynamic Characters. *ACM TOG* 40, 4, Article 94 (aug 2021).

Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. 2020. Deepcap: Monocular human performance capture using weak supervision. In *CVPR*.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *NeurIPS*. 6626–6637.

Yang Hong, Bo Peng, Haiyao Xiao, Ligang Liu, and Juyong Zhang. 2022. HeadNeRF: A Real-time NeRF-based Parametric Head Model. In *CVPR*.

Tao Hu, Tao Yu, Zerong Zheng, He Zhang, Yebin Liu, and Matthias Zwicker. 2022. HVTR: Hybrid Volumetric-Textural Rendering for Human Avatars. In *2022 International Conference on 3D Vision (3DV)*.

Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. 2011. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *24th annual ACM symposium on User interface software and technology*.

Hanbyul Joo, Tomas Simon, and Yaser Sheikh. 2018. Total Capture: A 3D Deformation Model for Tracking Faces, Hands, and Bodies. In *CVPR*.

Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhöfer, Jürgen Gall, Angjoo Kanazawa, and Christoph Lassner. 2022a. TAVA: Template-free Animatable Volumetric Actors. In *ECCV*, Vol. 13692. 419–436.

Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. 2017. Learning a model of facial shape and expression from 4D scans. *ACM TOG (SIGGRAPH Asia)* 36, 6 (2017), 194:1–194:17.

Zhe Li, Zerong Zheng, Yuxiao Liu, Boyao Zhou, and Yebin Liu. 2023. PoseVocab: Learning Joint-structured Pose Embeddings for Human Avatar Modeling. In *ACM SIGGRAPH 2023 Conference Proceedings*.

Zhe Li, Zerong Zheng, Hongwen Zhang, Chaonan Ji, and Yebin Liu. 2022b. AvatarCap: Animatable Avatar Conditioned Monocular Human Volumetric Capture. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*. Springer, 322–341.

Shanchuan Lin, Andrey Ryabtsev, Soumyadip Sengupta, Brian L. Curless, Steven M. Seitz, and Ira Kemelmacher-Shlizerman. 2021. Real-Time High-Resolution Background Matting. In *CVPR*.

Siyou Lin, Hongwen Zhang, Zerong Zheng, Ruizhi Shao, and Yebin Liu. 2022. Learning Implicit Templates for Point-Based Clothed Human Modeling. In *ECCV*.

Jia-Wei Liu, Yan-Pei Cao, Weijia Mao, Wenqiao Zhang, David Junhao Zhang, Jussi Keppo, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. 2022. Devrf: Fast deformable voxel radiance fields for dynamic scenes. In *arXiv preprint, arXiv:2205.15723*.

Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. 2021. Neural Actor: Neural Free-view Synthesis of Human Actors with Pose Control. *ACM TOG (ACM SIGGRAPH Asia)* (2021).

Lingjie Liu, Weipeng Xu, Marc Habermann, Michael Zollhöfer, Florian Bernard, Hyeongwoo Kim, Wenping Wang, and Christian Theobalt. 2020. Neural Human Video Rendering by Learning Dynamic Textures and Rendering-to-Video Translation. *IEEE TVCG* PP (05 2020), 1–1.

Lingjie Liu, Weipeng Xu, Michael Zollhöfer, Hyeongwoo Kim, Florian Bernard, Marc Habermann, Wenping Wang, and Christian Theobalt. 2019b. Neural Rendering and Reenactment of Human Actor Videos. *ACM TOG* 38, 5 (2019), 139:1–139:14.

Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. 2019a. Soft Rasterizer: A Differentiable Renderer for Image-based 3D Reasoning. In *ICCV*.

Stephen Lombardi, Jason M. Saragih, Tomas Simon, and Yaser Sheikh. 2018. Deep appearance models for face rendering. *TOG* 37, 4 (2018), 68.

Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. 2019. Neural Volumes: Learning Dynamic Renderable Volumes from Images. *ACM TOG* 38, 4, Article 65 (July 2019), 14 pages.

Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhöfer, Yaser Sheikh, and Jason M. Saragih. 2021. Mixture of volumetric primitives for efficient neural rendering. *ACM TOG* 40, 4 (2021), 59:1–59:13.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. 2015. SMPL: A skinned multi-person linear model. *ACM TOG* 34, 6 (2015), 1–16.

Xiyang Luo, Ruohan Zhan, Huiwen Chang, Feng Yang, and Peyman Milanfar. 2020. Distortion Agnostic Deep Watermarking. In *CVPR*. Computer Vision Foundation / IEEE, 13545–13554.

Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J. Black. 2020. Learning to Dress 3D People in Generative Clothing. In *CVPR*.

Qianli Ma, Jinlong Yang, Siyu Tang, and Michael J. Black. 2021b. The Power of Points for Modeling Humans in Clothing. In *ICCV*.

Shugao Ma, Tomas Simon, Jason M. Saragih, Dawei Wang, Yuecheng Li, Fernando De la Torre, and Yaser Sheikh. 2021a. Pixel Codec Avatars. In *CVPR*. Computer Vision Foundation / IEEE, 64–73.

Marko Mihajlovic, Yan Zhang, Michael J Black, and Siyu Tang. 2021. LEAP: Learning Articulated Occupancy of People. In *CVPR*.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. 2020. InterHand2.6M: A Dataset and Baseline for 3D Interacting Hand Pose Estimation from a Single RGB Image. In *European Conference on Computer Vision (ECCV)*.

Thomas Muller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG* 41, 4, Article 102 (jul 2022).

Ahmed A A Osman, Timo Bolkart, and Michael J. Black. 2020. STAR: A Sparse Trained Articulated Human Body Regressor. In *ECCV*.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *CVPR*.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. 2019. Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In *CVPR*.

Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. 2021a. Animatable Neural Radiance Fields for Modeling Dynamic Human Bodies. In *ICCV*.

Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. 2021b. Neural Body: Implicit Neural Representations with Structured Latent Codes for Novel View Synthesis of Dynamic Humans. In *CVPR*.

Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J Black. 2017. ClothCap: Seamless 4D clothing capture and retargeting. *ACM TOG* 36, 4 (2017), 1–15.

Sergey Prokudin, Michael J. Black, and Javier Romero. 2021. SMPLpix: Neural Avatars from 3D Human Models. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*. IEEE, 1809–1818. https://doi.org/10.1109/WACV48630.2021.00185

Amit Raj, Julian Tanke, James Hays, Minh Vo, Carsten Stoll, and Christoph Lassner. 2021. ANR: Articulated Neural Rendering for Virtual Avatars. In *CVPR*.

Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. 2021. KiloNeRF: Speeding up Neural Radiance Fields with Thousands of Tiny MLPs. In *ICCV*.

Edoardo Remelli, Timur Bagautdinov, Shunsuke Saito, Chenglei Wu, Tomas Simon, Shih-En Wei, Kaiwen Guo, Zhe Cao, Fabian Prada, Jason Saragih, and Yaser Sheikh. 2022. Drivable Volumetric Avatars Using Texel-Aligned Features. Association for Computing Machinery, New York, NY, USA.

Alexander Richard, Colin Lea, Shugao Ma, Juergen Gall, Fernando De la Torre, and Yaser Sheikh. 2021. Audio- and Gaze-driven Facial Animation of Codec Avatars. In *WACV*. IEEE, 41–50.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. 36, 6 (2017).

Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J. Black. 2021. SCANimate: Weakly Supervised Learning of Skinned Clothed Avatar Networks. In *CVPR*.

Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, Dmitry Ulyanov, Alexander Vakhitov, and Victor Lempitsky. 2019. Textured Neural Avatars. In *CVPR*.

Pratul P. Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T. Barron. 2021. NeRV: Neural Reflectance and Visibility Fields for Relighting and View Synthesis. In *CVPR*. 7495–7504.

Carsten Stoll, Juergen Gall, Edilson de Aguiar, Sebastian Thrun, and Christian Theobalt. 2010. Video-Based Reconstruction of Animatable Human Characters. 29, 6 (2010).

Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. 2021. A-NeRF: Articulated Neural Radiance Fields for Learning Human Shape, Appearance, and Pose. In *NeurIPS*. 12278–12291.

Shih-Yang Su, Timur Bagautdinov, and Helge Rhodin. [n. d.]. DANBO: Disentangled Articulated Neural Body Representations via Graph Neural Networks. In *European Conference on Computer Vision*.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. 2022b. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*. 5459–5469.

Jiaming Sun, Xi Chen, Qianqian Wang, Zhengqi Li, Hadar Averbuch-Elor, Xiaowei Zhou, and Noah Snavely. 2022a. Neural 3D Reconstruction in the Wild. In *SIGGRAPH Conference Proceedings*.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. 2020. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *NeurIPS*.

Garvita Tiwari, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. 2021. Neural-GIF: Neural Generalized Implicit Functions for Animating People in Clothing. In *ICCV*.

Luan Tran and Xiaoming Liu. 2018. Nonlinear 3D Face Morphable Model. In *CVPR*. Computer Vision Foundation / IEEE Computer Society, 7346–7355.

Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popovic. 2006. Face transfer with multilinear models. In *SIGGRAPH Courses*, John W. Finnegan and Dave Shreiner (Eds.). ACM, 24.

Lizhen Wang, Zhiyua Chen, Tao Yu, Chenguang Ma, Liang Li, and Yebin Liu. 2022a. FaceVerse: a Fine-grained and Detail-controllable 3D Face Morphable Model from a Hybrid Dataset. In *CVPR*.

Liao Wang, Jiakai Zhang, Xinhang Liu, Fuqiang Zhao, Yanshun Zhang, Yingliang Zhang, Minye Wu, Jingyi Yu, and Lan Xu. 2022d. Fourier plenoctrees for dynamic radiance field rendering in real-time. In *CVPR*. 13524–13534.

Lizhen Wang, Xiaochen Zhao, Jingxiang Sun, Yuxiang Zhang, Hongwen Zhang, Tao Yu, and Yebin Liu. 2023. StyleAvatar: Real-time Photo-realistic Portrait Avatar from a Single Video. In *ACM SIGGRAPH 2023 Conference Proceedings*.

Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. 2022b. ARAH: Animatable Volume Rendering of Articulated Human SDFs. In *ECCV*, Vol. 13692. 1–19.

Zhuo Wang, Zezheng Wang, Zitong Yu, Weihong Deng, Jiahong Li, Tingting Gao, and Zhongyuan Wang. 2022c. Domain Generalization via Shuffled Style Assembly for Face Anti-Spoofing. In *CVPR*. IEEE, 4113–4123.

Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. 2022. HumanNeRF: Free-Viewpoint Rendering of Moving People From Monocular Video. In *CVPR*. 16210–16220.

Liwen Wu, Jae Yong Lee, Yu-Xiong Wang Anand Bhattad, and David Forsyth. 2022. DIVeR: Real-time and Accurate Neural Radiance Fields with Deterministic Integration for Volume Rendering. In *CVPR*. 16200–16209.

Donglai Xiang, Timur M. Bagautdinov, Tuur Stuyck, Fabian Prada, Javier Romero, Weipeng Xu, Shunsuke Saito, Jingfan Guo, Breannan Smith, Takaaki Shiratori, Yaser Sheikh, Jessica K. Hodgins, and Chenglei Wu. 2022. Dressing Avatars: Deep Photorealistic Appearance for Physically Simulated Clothing. *CoRR* abs/2206.15470 (2022).

Donglai Xiang, Fabian Prada, Timur Bagautdinov, Weipeng Xu, Yuan Dong, He Wen, Jessica Hodgins, and Chenglei Wu. 2021. Modeling Clothing as a Separate Layer for an Animatable Human Avatar. *ACM TOG* 40, 6 (2021).

Feng Xu, Yebin Liu, Carsten Stoll, James Tompkin, Gaurav Bharaj, Qionghai Dai, Hans-Peter Seidel, Jan Kautz, and Christian Theobalt. 2011. Video-Based Characters: Creating New Human Performances from a Multi-View Video Database. 30, 4 (2011).

Yuelang Xu, Lizhen Wang, Xiaochen Zhao, Hongwen Zhang, and Yebin Liu. 2023a. AvatarMAV: Fast 3D Head Avatar Reconstruction Using Motion-Aware Neural Voxels. In *ACM SIGGRAPH 2023 Conference Proceedings*.

Yuelang Xu, Hongwen Zhang, Lizhen Wang, Xiaochen Zhao, Han Huang, Guojun Qi, and Yebin Liu. 2023b. LatentAvatar: Learning Latent Expression Code for Expressive Neural Head Avatar. In *ACM SIGGRAPH 2023 Conference Proceedings*.

Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. 2021. Volume rendering of neural implicit surfaces. In *NeurIPS*.

Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. 2020. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. *NeurIPS* 33 (2020).

Tarun Yenamandra, Ayush Tewari, Florian Bernard, Hans-Peter Seidel, Mohamed Elgharib, Daniel Cremers, and Christian Theobalt. 2021. i3DMM: Deep Implicit 3D Morphable Model of Human Heads. In *CVPR*. 12803–12813.

Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. 2021. Plenoctrees for real-time rendering of neural radiance fields. In *ICCV*. 5752–5761.

Hongwen Zhang, Siyou Lin, Ruizhi Shao, Yuxiang Zhang, Zerong Zheng, Han Huang, Yandong Guo, and Yebin Liu. 2023a. CloSET: Modeling Clothed Humans on Continuous Surface with Explicit Template Decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Hongwen Zhang, Yating Tian, Yuxiang Zhang, Mengcheng Li, Liang An, Zhenan Sun, and Yebin Liu. 2023b. PyMAF-X: Towards Well-aligned Full-body Model Regression from Monocular Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

Jingbo Zhang, Xiaoyu Li, Ziyu Wan, Can Wang, and Jing Liao. 2022. FDNeRF: Few-Shot Dynamic Neural Radiance Fields for Face Reconstruction and Expression Editing. In *SIGGRAPH Asia 2022 Conference Papers*. Association for Computing Machinery, New York, NY, USA, Article 12, 9 pages. https://doi.org/10.1145/3550469.3555404

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*. 586–595.

Xiuming Zhang, Pratul P. Srinivasan, Boyang Deng, Paul E. Debevec, William T. Freeman, and Jonathan T. Barron. 2021. NeRFactor: neural factorization of shape and reflectance under an unknown illumination. *ACM Trans. Graph.* 40, 6 (2021), 237:1–237:18.

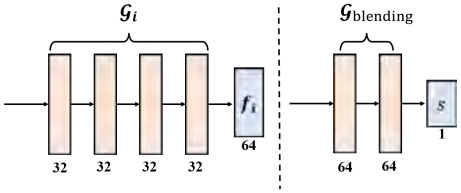
Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C. Bühler, Xu Chen, Michael J. Black, and Otmar Hilliges. 2022a. I M Avatar: Implicit Morphable Head Avatars from Videos. In *CVPR*. IEEE, 13535–13545.

Zerong Zheng, Han Huang, Tao Yu, Hongwen Zhang, Yandong Guo, and Yebin Liu. 2022b. Structured Local Radiance Fields for Human Avatar Modeling. In *CVPR*.

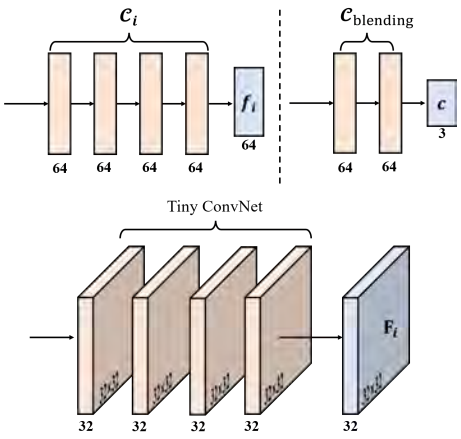
A NETWORK ARCHITECTURE

As described in Section 3.1, the geometry field of our body representation consists of N local MLPs and a blending MLP with N being the node number, while the color field consists of N local MLPs, a blending MLP as well as N tiny convolutional networks that extract dynamic feature patches from the learnable positional encoding

and dynamic detail embeddings. Throughout the paper we sample $N = 128$ nodes to construct the structured local representation .We illustrate the body network architecture in Figure 18.



(a) Body geometry networks.



(b) Body color networks.

Fig. 18. **Network architecture of our body representation** with the numbers of output channels labeled underneath. All networks are implemented as MLPs except the tiny convolutional network in the bottom of (b). Each fully connected layer in the MLP is followed by ELU activation for the geometry network and ReLU for the color network. For all the layers in the tiny convolutional network, the resolution of the output feature maps is 32×32 and the convolutional kernel size is 3×3.

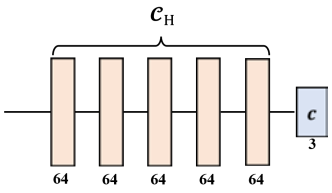


Fig. 19. **Network architecture of our hand color field** with the numbers of output channels labeled underneath. Each fully-connected layer is followed by ReLU activation.

The hand geometry field in our avatar is derived analytically using Equation 9 , so it does not contain any neural networks. The color field of hands is modeled with a simple MLP, which we illustrate in Figure 19.

As shown in Figure 4, the face network consists of 2 main components, namely a set of UNet for extracting feature triplanes and an MLP to regress the SDF/color value. We use separate sets of networks to model the geometry field and the color field. The detailed network architectures are illustrate in Figure 20. The resolution of the orthogonal rendering is 256×256.

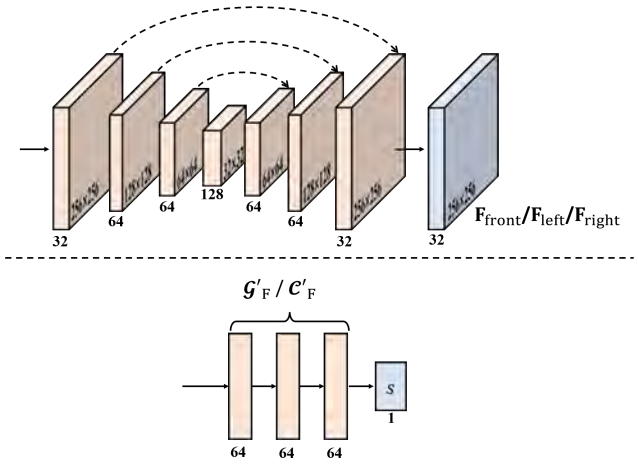


Fig. 20. **Network architecture of our face network** with the numbers of output channels labeled underneath. Top: the UNet architecture for extracting feature triplanes from orthogonal rendering of Faceverse. Bottom: the MLP architecture for predicting the SDF or color values in the face representation.

Note that before feeding the coordinates and view directions into the networks, we augment them using sinusoidal positional encoding [Mildenhall et al. 2020], which is defined as:

$$\gamma(\mathbf{x}) = \left(\mathbf{x}, \sin(\mathbf{x}), \cos(\mathbf{x}), ..., \sin(2^{m-1}\mathbf{x}), \cos(2^{m-1}\mathbf{x})\right).$$

The value of m is 6 for coordinates and 4 for view directions.

B ADDITIONAL IMPLANTATION DETAILS

We use PyTorch to implement our networks and use the Adam optimizer to train the networks. The hyperparameters needed for network implementation and training are reported in Table 6, while the number of iterations is set according to Table 1. During network training, the learning rate decays exponentially every 20k iterations. Note that the vanilla NeRF adopts a hierarchical sampling strategy, while we only train one network with uniform sampling for volume rendering in the first training pass. For baseline methods, we use the author-provided code and run all the experiments using their default training settings.

In order to achieve real-time testing performance, we implement our rendering pipeline fully on GPU using CUDA/C++. The network inference is performed using NVIDIA TensorRT, while the other parts of our rendering pipeline is implemented with custom CUDA kernels. The running time of each step in reported in Figure 5.

Table 6. Hyperparameters for network training and evaluation.

Parameter Name	Value
σ (In Equantion 4)	0.05
ϵ (In Equantion 4)	0.001
λ_{mask} (In Equation 17)	1.0
λ_{Eikonal} (In Equation 17)	2.0
λ_{node} (In Equation 17)	0.04
λ_{ebd} (In Equation 17)	0.01
λ_{KL} (In Equation 17)	1×10^{-6}
λ_{LPIPS} (In Equation 18)	1.0
Number of Ray Samples Per Batch (Training Pass I)	400
Number of Point Samples Per Ray (Training Pass I)	64
Patch Resolution (Training Pass II)	128×128
Batch Size	4
Learning Rate (Training Pass I)	5×10^{-4}
Learning Rate (Training Pass II)	1×10^{-4}

C ADDITIONAL DETAILS OF DATA

In our experiments, we use the capture system in Section 5.2.1 and collect multi-view video data for 4 subjects, two males and one female in two sets of clothing. Figure 21 present some example video frames.

Fig. 21. Subjects in our experiments.

To obtain the initial SMPL-X fitting for training data, we follow a classical optimization-based method [Pavlakos et al. 2019]. Then we refine the initial fitting results using differentiable rendering. Specifically, we use Background Matting v2 [Lin et al. 2021] to extract the foreground body segmentation M_t^* for frame t. Then we render the silhouette of SMPL-X model with a differentiable rasterizer [Liu et al. 2019a]. The rendered silhouette is denoted as $M(\theta_t, \beta)$, where β are shape coefficients of SMPL-X model and θ_t is the pose parameters at frame t. We optimize θ_t and β through the following energy function:

$$\mathcal{E} = \mathcal{E}_{sil} + \lambda_1 \mathcal{E}_{kpt} + \lambda_2 \mathcal{E}_{reg}, \tag{20}$$

where $\mathcal{E}_{sil}$ measures the MSE between M_t^* and $M(\theta_t, \beta)$, $\mathcal{E}_{kpt}$ penalizes keypoint reprojection errors with an L2 loss, and $\mathcal{E}_{reg}$ serves as a regularization term to prevent the parameters from deviating from the initial values. The regularization term $\mathcal{E}_{reg}$ is defined as:

$$\mathcal{E}_{reg} = ||\boldsymbol{\theta} - \boldsymbol{\theta}_{init}||_2^2 + ||\boldsymbol{\beta} - \boldsymbol{\beta}_{init}||_2^2, \tag{21}$$

Table 7. **Quantitative comparison on ZJU-Mocap [Peng et al. 2021b].** We report PSNR and LPIPS on synthesized images under unseen poses from the testset of the ZJU-MoCap. To ease reading, we highlight the best scores with orange shading, and the second best with light orange.

Methods	PSNR ↑	LPIPS ↓
Neural Body	22.7	0.135
Animatable NeRF	23.1	0.145
ARAH	24.2	0.099
SLRF	23.6	0.109
Ours	23.6	0.104

where $\boldsymbol{\theta}_{init}$ and $\boldsymbol{\beta}_{init}$ are the initial value of pose parameters and shape coefficients, respectively. For all our experiments, we set $\lambda_1 = 1 \times 10^{-6}$ and $\lambda_2 = 1 \times 10^{-3}$.

For computational efficiency, we first optimize θ_t and β for 20 frames that are uniformly sampled from the sequence. After that, we fix the shape coefficients and optimize the pose parameters for each individual frame.

D RESULTS ON ZJU-MOCAP

This paper focuses on developing full-body avatars that can be driven with everyday actions such as walking, talking, sports and so on. Therefore, we collect training videos of about 2000 frames in length, capturing common body movements and usual facial expressions. While we recommend users to use our method in this setting, it is worth noting that our method still works with shorter videos and fewer cameras, such as those in the ZJU-Mocap system (which uses only four cameras and has a length of 300 frames). In Table 7, we report the numeric results of our method's performance with regards to pose generalization on Sequence "387" from ZJU-MoCap. The results show that our approach can achieve comparable performance with state-of-the-art methods under such extremely sparse inputs.