

UVA: Towards Unified Volumetric Avatar for View Synthesis, Pose rendering, Geometry and Texture Editing

Jinlong Fan
The University of Sydney
jfan0939@uni.sydney.edu.au

Jing Zhang
The University of Sydney
chaimi.ustc@gmail.com

Dacheng Tao
The University of Sydney
dacheng.tao@gmail.com

Abstract

*Neural radiance field (NeRF) has become a popular 3D representation method for human avatar reconstruction due to its high-quality rendering capabilities, e.g., regarding novel views and poses. However, previous methods for editing the geometry and appearance of the avatar only allow for global editing through body shape parameters and 2D texture maps. In this paper, we propose a new approach named **Unified Volumetric Avatar (UVA)** that enables local and independent editing of both geometry and texture, while retaining the ability to render novel views and poses. UVA transforms each observation point to a canonical space using a skinning motion field and represents geometry and texture in separate neural fields. Each field is composed of a set of structured latent codes that are attached to anchor nodes on a deformable mesh in canonical space and diffused into the entire space via interpolation, allowing for local editing. To address spatial ambiguity in code interpolation, we use a local signed height indicator. We also replace the view-dependent radiance color with a pose-dependent shading factor to better represent surface illumination in different poses. Experiments on multiple human avatars demonstrate that our UVA achieves competitive results in novel view synthesis and novel pose rendering while enabling local and independent editing of geometry and appearance. The source code will be released.*

1. Introduction

Neural implicit fields are widely recognized for their remarkable success in 3D representation, owing to their high-quality rendering and flexible representation of complex shapes and scenes without the need for explicit surface meshes [9], voxels [16, 19, 2], or point clouds [37], especially after the emergence of NeRF [26]. Despite the numerous efforts made to adapt neural fields for dynamic objects [33, 27, 28] in the wild scenes [24, 42, 25] and large scenes [45, 43], manipulating the reconstructed neural fields

Methods	NV	NP	Local Tex.	Local Geo.
NeRF [26]	✓	✗	✗	✗
Neumesh [51]	✓	✗	✓	✓
Ani-NeRF [31]	✓	✓	✗	✗
UV-Volumes [6]	✓	✓	✓	✗
Ours (UVA)	✓	✓	✓	✓

Table 1: Comparison of the rendering and editing abilities of different methods. Our UVA enables local and independent editing of both **Geometry** and **Texture** while retaining the ability to render Novel Views and Novel Poses.

remains challenging due to the black-box nature of the implicit representation. Previous methods have attempted to disentangle lighting, materials, geometry, and/or texture as separate latent variables [21], neural texture maps [48], or controllable sub-neural fields [12] to facilitate the editing of the scene [34, 17, 56]. However, these approaches are limited to static objects or are only capable of global manipulation.

Editing the dynamic field becomes particularly challenging when dealing with articulated objects, e.g. human avatars. Earlier works on reconstructing human avatars using neural radiance fields have mainly focused on novel view synthesis and novel pose rendering [32, 47]. With the given camera and human pose parameters, the reconstructed avatar can be rendered from different views and in various poses. Besides, by bounding the avatar with a 3D bounding box, it can be relocated to any position in the 3D world [53]. However, these approaches still lack the ability to edit the geometry and texture of the avatar.

The other line of work stores appearance information as separate neural features, which can be decoded into textures via deferred rendering [48, 39, 58]. The correspondence between the 3D query points and the 2D neural texture map is determined using pre-defined UV mapping [35, 8, 36]. However, UV mapping is only defined in the 3D-to-2D direction, and mapping a specific pixel on the texture map to the 3D world is not directly available, which makes local texture editing difficult. Additionally, these methods can only



Input Reconstruction Novel view Novel pose Geometry editing Texture swapping Texture painting

Figure 1: We reconstruct a unified volumetric avatar from a sparse multi-view video. With camera and pose parameters, we can render new views and poses of the performer. Our approach uses disentangled neural fields based on structured latent codes to capture the 3D geometry and appearance, allowing independent and local geometry and texture editing. To see the animation, please check out the documents with a compatible software like *Adobe Acrobat* or *KDE Okular*. We also provide the animation as a separate file in the supplementary material.

control the shape of the human body globally via shape parameters in statistical human models, *e.g.* SMPL [22], which are unable to edit the geometry of the human body locally.

In general, editing the geometry of objects is more challenging than editing their texture, as changes to geometry can affect appearance, *e.g.*, the length of limbs, whereas appearance can be updated while maintaining the same geometry. Since explicit mesh representations can be precisely controlled and edited by altering vertices and faces, some approaches transfer a pre-trained neural radiance field to a polygon mesh by baking the geometry and appearance information from coordinate-based networks [51, 50]. However, utilizing two-stage distilling methods poses a challenging task, as it requires manual and extensive parameter tuning while being only suitable for static objects. How to edit the geometry of dynamic avatars remains an open question.

In this paper, we present a novel approach that enables both geometry deformation and local texture manipulation, while retaining the ability to render novel views and poses. The core idea is to encode geometry and appearance in separate fields and anchor the neural field on a deformable mesh in canonical space. Technically, we first transform the query points, which are sampled from variant poses in each frame, to the canonical space via the skinning motion field. The geometry and appearance fields are then represented by two sets of structured latent codes independently, which are attached to anchor nodes on the canonical mesh. Since the latent codes are tightly aligned with the mesh nodes, the deformation of the mesh can directly guide the deformation of the neural field. Meanwhile, the texture can also be edited locally, as each of the latent codes can be tuned individually. Although the discrete latent codes defined on mesh nodes can be diffused to the entire space using nearest-neighbor (NN) interpolation, it results in spatial query indistinguishability. To address this, we propose using a local signed height indicator comprising UV coordinates and point-to-surface distance to mark query points. The local signed height, relative to the anchored mesh, can track mesh deformation.

Thanks to the mesh-guided geometry and appearance, as

well as mesh-tracked signed height, our proposed model, named **Unified Volumetric Avatar (UVA)**, offers an all-in-one solution for novel view synthesis, novel pose rendering, local geometry deformation, and local texture manipulation. The comparison against existing methods regarding the rendering and editing abilities is shown in Table 1. In summary, the main contribution of this paper is threefold:

- We propose a novel baseline model named Unified Volumetric Avatar that enables local and independent editing of both geometry and texture while retaining the ability to render novel views and poses.
- We propose to use a skinning field to represent the motion while employing structured latent codes and the signed height indicator to characterize the neural field. Both are defined relative to the anchored mesh, facilitating the tracking of mesh deformation.
- The experimental results demonstrate the efficacy and versatility of our UVA, establishing it as a baseline model capable of rendering and manipulation.

2. Related Work

Neural human. NeRF has emerged as an effective approach for volume rendering, enabling high-quality novel view synthesis and serving as a compact 3D representation in various applications [26]. While neural rendering originally focused on static scenes or objects, recent works have introduced a deformation field to adapt the approach to dynamic scenes or objects, particularly for human reconstruction. Typically, a motion field such as a displacement field [27, 28] or a skinning field based on human prior [31] is used to align the dynamic human with a template in canonical space, where the canonical radiance field captures the human in coordinate-based networks. However, implicit representations of reconstructed humans do not allow for easy manipulation. Our method employs structured latent codes and a signed height indicator to describe the canonical space

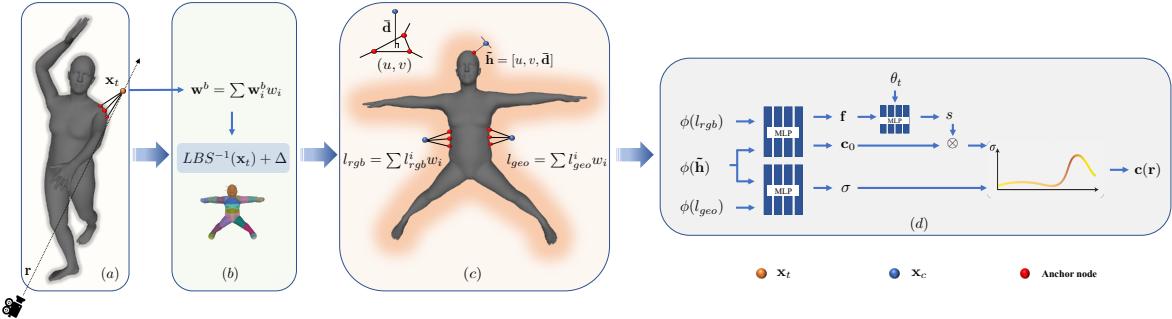


Figure 2: Overview of the proposed method. Given a sampled point \mathbf{x}_t in the target space (a), we first determine its LBS (Linear Blending Skinning) weights \mathbf{w}^b by interpolating the weights of its Top-K nearest neighbors (NNs). Combining a neural deformation field with the NN motion field (b), the target point \mathbf{x}_t is transformed into the canonical space (c). Both the structured geometry latent codes l_{geo} and appearance latent codes l_{rgb} are anchored to nodes on the deformable canonical mesh \mathcal{M} . A latent code for an arbitrary point \mathbf{x}_c is then calculated through NN interpolation. Given the signed height indicator $\tilde{\mathbf{h}}$, we use an MLP to predict the volume density σ and color \mathbf{c}_0 . After being modulated by the shading factor s , the colors are integrated along the target ray to produce the final color $\mathbf{c}(\mathbf{r})$.

based on mesh-based representation, enabling simple and intuitive manipulation of the reconstructed human.

Editable neural field. To make the neural field editable, several previous works have proposed various approaches. Liu et al. [21] introduce learnable variables to control the attributes of the object, while Xiang et al. [48] disentangle textures from the geometry as separated neural texture maps for texture editing. Yuan et al. [52] build a continuous deformation field between the edited mesh and the reconstructed mesh and use it to render the edited object. On the other hand, Xu et al. [50] and Yang et al. [51] distill the neural field to a controllable polygon mesh for subsequent editing. However, the two-stage distilling methods are difficult to use as it requires extensive parameter tuning. In contrast, our method is end-to-end trainable, and the canonical neural field is jointly optimized with the skinning field from scratch, requiring no manual effort.

Editable Human Avatar. Editing a human avatar is a more challenging task than editing the neural field of a static object because the avatar must maintain its drivability and natural appearance after modifications. Conventional techniques typically encode the avatar’s texture as a feature map and points on the surface of the 3D object are projected onto the map using precomputed 3D-to-2D UV mapping to extract its color feature. [54, 36, 1, 39, 58]. However, direct editing of neural features can be challenging. To address this issue, [23] and [6] combine explicit RGB texture stacks with implicit neural texture maps, allowing direct editing of RGB textures. Nevertheless, these methods lack the capacity for geometry editing. In contrast, our proposed method allows for the local and independent manipulation of both the texture and geometry of the reconstructed human avatar.

3. Method

Figure 2 provides an overview of our proposed method, which comprises two key components: a backward skinning motion field (Sec. 3.1) and a canonical radiance field (Sec. 3.2). Given a query point \mathbf{x}_t in the observation space, we first apply the skinning motion to deform it into the canonical space. Here, we combine the skeleton motion with a non-rigid deformation field as the skinning field. In the canonical space, we represent the geometry field and texture field using two sets of structured latent codes that are anchored on a deformable mesh. To capture pose-dependent surface illumination variance, we modulate the radiance color with a pose-conditioned shading factor. A key advantage of our approach is that the geometry and texture are stored in two separate and structured fields, which enables independent and local editing of both geometry (Sec. 3.3) and texture (Sec. 3.4).

3.1. Skinning Motion

Parametric human body models, such as SMPL [22] or SMPLx [29] model, typically use linear blending skinning (LBS) to deform points from the canonical pose to the target pose based on rigid bone transformations and skinning weights [13, 10]. However, these skinning weights are only defined on the template mesh points. Recent methods have used neural networks to predict the spatial skinning weights for arbitrary 3D points in space [18, 46]. While implicit LBS weights can provide detailed deformation, joint training of the radiance field and skinning field can result in a local optimum [31]. In this paper, we employ the nearest neighbor (NN) interpolation method to diffuse the skinning weights into the space. The NN motion field is a simple yet effective technique for human deformation, and it helps to stabilize the training of the separated radiance fields. Moreover, the NN

skinning field is naturally a mesh-based representation, making it possible to track vertex movements (*i.e.*, deformation of the mesh) and thus desired for geometry editing.

Here, we take SMPL as the parametric model. For a specific point \mathbf{x}_t in target pose θ_t , we first determine its Top-K nearest SMPL mesh vertexes $\{\mathbf{v}_i\}_{i=1}^K$. Given the LBS weights \mathbf{w}_i^b of each vertex \mathbf{v}_i , we interpolate the LBS weights using the inverse distance $w_i = 1 / \|\mathbf{x}_t - \mathbf{v}_i\|_2$:

$$\mathbf{w}^b = \sum_{i=1}^K \mathbf{w}_i^b w_i, w_i = \frac{w_i}{\sum w_i}. \quad (1)$$

Using this NN skinning motion field, we can deform \mathbf{x}_t as:

$$LBS^{-1}(\mathbf{x}_t) = \left(\sum_{b=1}^{N+1} \mathbf{w}^b \mathbf{B}^b \right)^{-1} \mathbf{x}_t, \quad (2)$$

where \mathbf{B}^b denotes the rigid transformation of bone b , and N represents the total number of bones. In order to more effectively describe the motion of off-the-body points, an additional transformation is used for the background points [47]. Specifically, if the distance between a given point and its nearest SMPL vertex exceeds a certain threshold, we classify the point as a static background point, assign it a bone weight of zero, and set the background weight to one.

LBS only accounts for rigid transformations. To accommodate non-rigid deformation, we make use of a neural network, denoted by \mathcal{F}_Δ , to predict per-point displacement Δ in canonical space. As point movements are inherently tied to pose, we condition the deformation on the target pose θ_t via the following equation:

$$\Delta = \mathcal{F}_\Delta(\phi(\tilde{\mathbf{h}}), \theta_t), \quad (3)$$

where $\phi(\cdot)$ represents the position encoding function [44], and $\tilde{\mathbf{h}}$ is the signed height indicator, which will be introduced in Sec. 3.2. Combining this neural deformation field with the NN skinning field, we can obtain canonical points \mathbf{x}_c as:

$$\mathbf{x}_c = LBS^{-1}(\mathbf{x}_t) + \Delta. \quad (4)$$

3.2. Canonical Space

We represent the volumetric avatar’s geometry and appearance using two separate neural radiance fields in canonical space. Each field is composed of a set of structured latent codes l which are anchored to a deformable mesh [32, 57, 15, 3]. When the mesh is deformed, the position of the structured latent codes changes correspondingly, resulting in deformed neural fields. Local appearance latent codes in the region of interest can be adjusted with new patterns, allowing targeted manipulation of appearance while leaving other areas unaffected. In this way, it enables independent and local editing of either the geometry or appearance of the reconstructed volumetric avatar.

Anchor mesh. In order to edit the geometry and appearance locally, we employ a deformable mesh $\mathcal{M} = \{\mathbf{v}_i\}$ as an anchor. In principle, the anchored mesh can take any form, such as a parametric model or one obtained from off-the-shelf methods [7, 49, 11]. Here, we use the SMPL mesh in *rest* pose as the anchored mesh for our baseline. The structured latent codes l_{geo} and l_{rgb} are stored at the anchor nodes on the mesh, where the mesh vertices serve as the anchor nodes for simplicity, though other sampled nodes may also be used [51, 57]. For a 3D point \mathbf{x} in canonical space, we use NN interpolation to get its latent code, similar to the way we obtain the LBS weights:

$$l_{geo/rgb} = \sum_{i=1}^K l_{geo/rgb}^i w_i, \quad (5)$$

where w_i is the inverse distance weighting defined in Eq. 1. However, such interpolation alone can lead to ambiguity for points along the direction perpendicular to the surface, where multiple points can have the same latent codes but in different positions. To address this, we introduce a local signed height $\tilde{\mathbf{h}}$ as an indicator [30, 20, 54]. The signed height plays a similar role as the signed distance but includes the UV coordinate of \mathbf{x} . As the body surface is intrinsically a 2D manifold, and the anchored mesh can be projected onto a pre-defined 2D UV map, the signed height indicator can locate any arbitrary point in 3D to a 2.5D volume by equipping the UV coordinate with a signed distance.

$$\tilde{\mathbf{h}} = [u, v, \bar{d}], \quad (6)$$

where $[u, v]$ is the 2D coordinate of \mathbf{x} on the UV map, \bar{d} is the signed distance. To calculate \bar{d} , we first project \mathbf{x} onto the mesh surface at \mathbf{x}_0 and obtain the direction of point $\mathbf{r} = \mathbf{x} - \mathbf{x}_0$. Given the surface normal $\tilde{\mathbf{n}}$ at \mathbf{x}_0 , the signed distance can be calculated as $\bar{d} = sign(\tilde{\mathbf{r}} \cdot \tilde{\mathbf{n}}) \|\tilde{\mathbf{r}}\|_2$.

Mesh-based geometry. In previous methods such as [50] and [51], the polygon mesh is baked from a pre-trained neural field and then taken as a proxy for user editing. However, the two-stage distilling methods are difficult to use as it requires manual and extensive parameter tuning. In contrast, our method jointly optimizes the geometry field with other components from scratch without any extra effort. Given the interpolated geometry latent code l_{geo} and the signed height $\tilde{\mathbf{h}}$, we use an MLP \mathcal{F}_σ to predict the density σ as follows:

$$\sigma = \mathcal{F}_\sigma(\phi(\tilde{\mathbf{h}}), l_{geo}), \quad (7)$$

where $\phi(\cdot)$ is the same position encoding function as in Eq. 3. As the structured latent codes and the signed height indicator are both tightly attached to the canonical mesh, the density field has the ability to track the deformation of the anchored mesh, which is crucial for mesh-guided geometry editing.

Mesh-based appearance. In NeRF, the radiance field is conditioned on ray directions. However, when dealing with posed humans, the target rays are bent in canonical space, and the sampled points along a ray no longer share the same view directions. Additionally, self-occlusion can result in points along the same ray having different illumination. To address these issues, we replace the ray direction-dependent radiance color with a pose-related shading factor. Specifically, we decompose the radiance color in canonical space into two factors. The first factor comprises pose-independent RGB colors that are shared across all frames. We utilize a neural network \mathcal{F}_c that takes the signed height and appearance latent code as input to predict the shared RGB colors:

$$\mathbf{c}_0, \mathbf{f} = \mathcal{F}_c(\phi(\tilde{\mathbf{h}}), l_{rgb}), \quad (8)$$

where \mathbf{c}_0 represents the RGB color in canonical pose and \mathbf{f} is the output feature vector. The second factor is a pose-dependent shading factor that differs from frame to frame. Conditioned on the target pose θ_t , we use a shallow MLP \mathcal{F}_s to estimate the per-point shading factor s for each frame:

$$s = \mathcal{F}_s(\mathbf{f}, \theta_t). \quad (9)$$

For points in the target pose, we modulate \mathbf{c}_0 with the shading factor to obtain the final color $\mathbf{c} = \mathbf{c}_0 \odot s$.

3.3. Mesh-guided Geometry Editing

Traditional parametric body models, such as SMPL, offer global control over the shape of the body through shape parameters. However, these parameters affect the entire mesh, and any changes to them will alter the whole body. Our proposed method, UVA, offers local geometry editing in fine detail, allowing for precise modifications such as bulges on specific parts of the body, by attaching structured latent codes to anchor nodes on the mesh and defining signed height indicators relative to the mesh surface. This ensures that the density field can track the movement of anchored nodes and the deformation of the mesh, enabling mesh-guided geometry editing. Users can move anchor nodes easily and freely in 3D modeling software (*e.g.*, Blender) or with out-of-the-box mesh deforming methods (*e.g.*, as-rigid-as-possible (ARAP) [40]). Additionally, the edited avatar can be rendered in novel views and poses, as our skinning motion field is also mesh-based. By copying the movements of corresponding vertices from the canonical mesh to the target mesh, the skinning field can be deformed in sync, enabling animation.

3.4. Texture Editing

Texture editing for human avatars is still a challenging task in the field. In previous methods, attempts have been made to disentangle the geometry and appearance into separate neural implicit fields, and update the entire texture with new ones by means of neural texture map replacement [48],

or by painting on the 2D neural texture stack [6, 23]. In contrast, our proposed method represents the appearance field with anchored structured latent code, which allows for free 3D editing without requiring expert knowledge to catch the 2D-to-3D correspondence.

Geometry editing focuses on manipulating the positions of anchor nodes, while texture editing involves modifying the underlying latent code itself. It is possible to replace or fine-tune the latent codes with the desired texture, and local edits to the latent codes associated with interested anchor nodes will not affect the other regions or the geometry field, *i.e.*, allowing users to edit the texture locally and independently. In principle, there are no limitations to the types of texture editing that can be performed. Here we take texture swapping and texture painting as two examples.

Texture swapping. Given two trained UVAs, we can swap their textures by selecting anchor nodes of interest in 3D or by projecting 2D mask rays into 3D space to identify the closest nodes to be considered. Similar to [51], non-rigid 3D alignment is performed to align the source and target anchor nodes. Intuitively, there are two ways to infer the swapping texture. One way is to assign the NN interpolated source codes to the corresponding target ones and decode the pre-blended codes with the source decoder. The other way is to infer the source texture in the target pose and blend the pre-inferred texture in the swapping area. We chose the pre-blended method due to its better 3D consistency.

Texture painting. Given an arbitrary painting and a binary mask, we identify the affected anchor nodes using the point-to-ray distance. To obtain a smooth texture on the boundary, we dilate the mask slightly and use it to sample the training rays, following [51]. These locally selected latent codes are fine-tuned to match the painting, which transfers the 2D painting to 3D structured latent codes and the edited texture can be rendered in novel views and poses. However, fine-tuning the texture decoder locally is non-trivial. In UVA, a pixel value is uniquely determined by the interpolated latent code, the local signed height indicator, and the shading factor, among which only the signed height is not trainable. We can fix the shading network and the uninvolved latent codes, but the texture decoder may overfit to the signed height during single-image fine-tuning and generate artifacts at other positions. During training, to prevent the signed height from dominating the training and to make the network depend only on spatial position, we use a higher learning rate for the latent code (*e.g.*, 10 times higher in our experiments). In texture painting, we further increase the learning rate gap to reduce the impact of the signed height indicator. With spatial-aware fine-tuning, the texture can be edited locally and independently.



Figure 3: Visual results of geometry editing. The first column shows the edited geometry in canonical pose, while the remaining columns show the rendering results from different poses. The images in the first row are rendered in reconstructed geometry. In the second row, the geometry is manipulated globally by adjusting the SMPL shape parameters to increase its size. Local geometry editing is shown in the last two rows, where the size of limbs is partly increased in the third row, and the face, ears, and arms are stretched out slightly in the last row.

4. Experiments

4.1. Experiment Settings

Dataset and metrics. To demonstrate the effectiveness of our proposed method, we conduct experiments on nine performers from the ZJU-MoCap dataset [32]. We choose ZJU-MoCap as our testbed due to the sprawling pose and relatively slow motion of the performers in this dataset. Additionally, considering the simplicity of our NN skinning field, we believe that evaluating UVA on ZJU-MoCap is reasonable. The ZJU-MoCap setup includes 23 cameras, and following [46], we select four equally spaced views and choose frames ranging from 60 to 300 for training, while the remaining frames are used for evaluation. In novel pose rendering, unseen poses in the test set are set as the target poses and images are rendered from test views. For geometry and appearance editing, we render the avatar in both novel views and novel poses. To compare our method with existing methods, we use standard metrics such as LPIPS [55], SSIM, and PSNR. The results of more performers can be found in supplementary materials.

Implementation details. The network is optimized us-

ing Adam [14] with a learning rate decay from $5e^{-4}$ to $5e^{-6}$, and a $10\times$ learning rate for the structured latent codes. The dimensions of geometry latent codes and appearance codes are both 32. The appearance network consists of an 8-layer MLP with a skip connection at the 4th layer. The density network is a 4-layer MLP with softplus activation, while the shading network is a 3-layer MLP. The motion field and canonical field are jointly trained for 300K iterations with a batch size of 1024. The training process takes about 10 hours using two NVIDIA Tesla V100 GPUs.

4.2. Novel view and Novel pose synthesis

For novel view synthesis and novel pose rendering, we compare our method against five existing approaches: 1) Neural Body (NB) [32] diffuses per-SMPL-vertex latent codes in observation space to condition the NeRF model and achieves high-quality novel view synthesis results on training poses; 2) Ani-NeRF [31] learns a backward LBS weight field and a canonical NeRF to reconstruct the human avatar; 3) A-NeRF [41] employs skeleton-relative embedding to predict the radiance field; 4) we implement TAVA-NN based on TAVA [18], but replace the learnable deformation field

Methods	Novel View			Novel Pose		
	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow
NB [32]	28.3	0.946	0.095	23.8	0.897	0.142
Ani-NeRF [31]	26.1	0.921	0.139	23.3	0.891	0.159
A-NeRF [41]	27.4	0.937	0.101	22.4	0.862	0.199
ARAH [46]	28.5	0.948	0.081	24.6	0.911	0.107
TAVA-NN [18]	26.3	0.924	0.112	23.7	0.892	0.139
Ours	26.7	0.924	0.110	23.9	0.892	0.137

Table 2: Quantitative results for novel view synthesis and novel pose rendering. The metrics have been averaged over nine performers from the ZJU-MoCap dataset.

with the NN skinning field to verify the effectiveness of our canonical field representation; 5) ARAH [46] adopts a joint root-finding module to establish the correspondence between the observation space and the canonical space and stores the geometry in SDF. It performs very well on out-of-distribution poses for novel view and novel pose rendering.

Quantitative results are presented in Table 2. Our method exhibits competitive performance on novel view synthesis and comparable results on novel pose synthesis. Notably, UVA outperforms A-NeRF on novel pose rendering. While A-NeRF utilizes over-parameterized bone-relative embeddings to locate 3D query points, our method leverages mesh-relative signed height indicators, demonstrating its superior ability. Compared with Ani-NeRF, our method performs marginally better, showing that the NN skinning field has a comparable ability to the neural deformation field. Although TAVA-NN employs the same motion field as our method, our approach yields slightly better results, verifying the effectiveness of our canonical field.

Qualitative results are shown in Figure 4 and Figure 5. As can be seen, despite using a simple and intuitive approach (NN interpolation) to diffuse the LBS weights and latent codes into the continuous space, UVA achieves comparable performance as Ani-NeRF that uses the learnable deformation field and NB that employs the 3D sparse convolution diffusion. Moreover, our method shows good generalization to unseen poses, while A-NeRF without surface priors struggles in this regard. We attribute this to the mesh-guided representations and surface-relative signed height indicator.

4.3. Geometry Editing

The proposed mesh-guided geometry neural field allows us to track the deformation of the anchored mesh and animate it with novel poses. In Figure 3, we show the rendered results of edited geometry in several poses. The first row shows the rendered results of the reconstructed avatar. In the second row, the geometry is globally edited by adjusting all vertex positions on the mesh via the SMPL shape parameters, where the structured latent codes move accordingly, and the geometry field deforms as expected. In the third and last row, we show UVA’s ability to deform the field locally. We import the anchored mesh into a 3D graphics tool, *e.g.*, Blender, and edit the mesh freely. In the third row, we edit the geom-

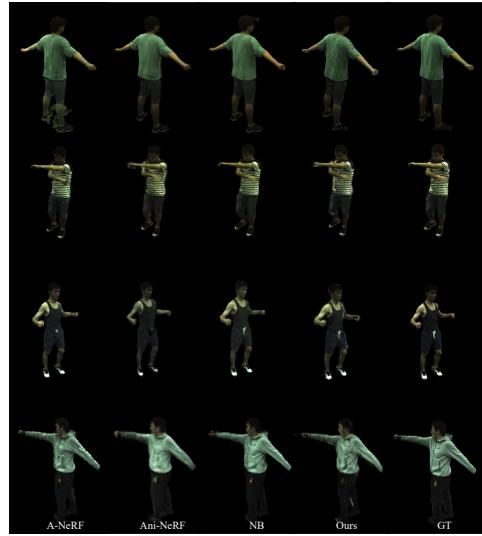


Figure 4: Qualitative results for novel view synthesis.

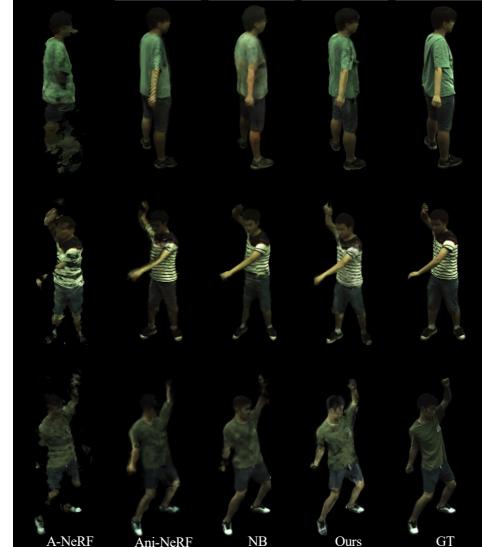


Figure 5: Qualitative results for novel pose synthesis.

etry by changing the size of the limbs, and in the last row, we stretch parts of the face, ears, and arms (Although these manipulations may appear simplistic, they serve to demonstrate UVA’s capability for local editing). UVA leverages a mesh-guided geometry field and mesh-based signed height indicator to yield reasonable global manipulation and local deformation outcomes. The user-friendly editing process provides a “What You See Is What You Get” experience.

4.4. Texture Editing

We demonstrate the ability of our method to perform texture editing through texture swapping and texture painting

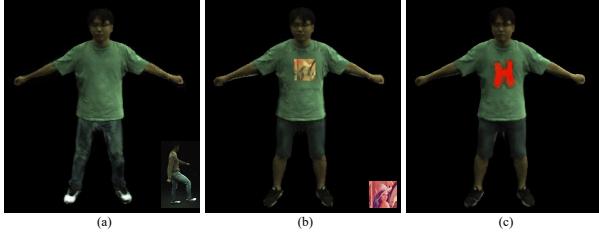


Figure 6: Visual results of texture editing. From left to right, the results are texture swapping (a), texture painting using a second image (b), and a pattern drawing by hand (c).

in Figure 6. In texture swapping, we swap the latent codes and texture decoder in the target area to generate the corresponding color. To smooth the texture near the boundary after editing, we blend the latent codes across target areas with their neighboring codes. Here, we swap the lower half of the body of two subjects, 313 and 394, with manually masked-out target areas. In texture painting, we update the local latent codes of interest through per-vertex fine-tuning to match the target texture, while the rest of the latent codes remain fixed. In Figure 6(b), we fill the area of interest with a reference image (*i.e.*, the Lena image), and in Figure 6(c), we draw a pattern “H” on the T-shirt. The visual results demonstrate that through the spatial-aware and per-latent code fine-tuning, the texture is edited locally without affecting the non-interested areas and is independent of the manipulation of geometry.

4.5. Ablation Study

Table 3 shows the ablation study results on the key components of UVA using the sequence of subject 313 for novel view synthesis and novel pose rendering.

Displacement field. For novel view synthesis, the target pose is included in the training poses, and therefore, the displacement Δ has a marginal effect on point alignment. However, for novel pose rendering, unseen poses lead to frame-by-frame deformation that cannot be captured by skeleton motion alone. In such scenarios, the displacement field Δ matters for achieving better rendering results.

Shading factor. Due to self-occlusion, relative position to the light, and camera exposure time, the same point on a surface may have varying levels of illumination when a performer is continuously acting. Without the pose-related shading factor, the color decoder cannot adequately represent the differences between frames, resulting in a slight performance drop for both novel view and novel pose synthesis.

Signed height indicator. To investigate the influence of the signed height indicator, we set them to constant zeros during training. Without the aid of indicators, the network struggled to differentiate between diffused spatial features that could be the same at distinct positions. This resulted

	Novel View		Novel Pose	
	PSNR \uparrow	SSIM \uparrow	PSNR \uparrow	SSIM \uparrow
w/o Δ	30.2	0.958	23.9	0.903
w/o shading	29.0	0.953	24.1	0.902
w/o indicator	29.3	0.945	24.0	0.894
Att. fusion	30.4	0.957	23.9	0.902
Ours(UVA)	30.5	0.958	24.5	0.905

Table 3: Ablation study of design choices in our method.

in artifacts along lines perpendicular to the surface and a significant drop in SSIM.

NN interpolation. As a viable alternative, we employ a self-attention block to integrate the nearest latent codes instead of relying on NN interpolation. These learnable weights possess more flexibility than inverse distance weights and can achieve comparable performance in novel view synthesis. Nevertheless, consistency cannot be guaranteed when the positions of anchor nodes shift. Consequently, the integrated latent code of a given point may not accurately reflect the motion of the anchor nodes in a predictive manner, which is crucial for geometry editing.

4.6. Limitation and Discussion

Our method struggles with complex body motions due to the limited capacity of the NN skinning field. Recent works have employed root-finding-based deformation fields to align the target space with the canonical space and enhance the generalization ability of the reconstructed avatar significantly [5, 4]. Therefore, it is worth trying to replace the NN skinning with a root-finding module to further enhance the capacity of UVA. Another limitation is the fixed number of anchor nodes, which may cause artifacts when the mesh vertexes are pushed away or pulled together during editing. As shown in the second row of Figure 3, the anchor nodes become sparse in the belly region, resulting in black stripes. This issue can be mitigated by dynamically adjusting the nodes [38], *e.g.*, adding more in sparse regions and removing redundant ones. We leave it as the future work.

5. Conclusion

We present a novel approach for volumetric avatar reconstruction that enables local geometry and texture editing, high-quality novel view synthesis, and novel pose rendering. Our method incorporates structured latent codes that are attached to a deformable mesh to track the deformation of the anchored mesh to guide the geometry editing. It also enables local texture editing by spatial-aware latent code fine-tuning. Utilizing volumetric rendering, it can produce high-quality images in novel views, and the reconstructed volumetric avatar can be reposed via the skinning motion field. We believe that our method sets a baseline for all-in-one avatar models that can be rendered, reposed, and edited.

References

- [1] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2293–2303, 2019.
- [2] John Ashburner and Karl J. Friston. Voxel-based morphometry—the methods. *Neuroimage*, 11(6):805–821, 2000.
- [3] Mingfei Chen, Jianfeng Zhang, Xiangyu Xu, Lijuan Liu, Yujun Cai, Jiashi Feng, and Shuicheng Yan. Geometry-guided progressive nerf for generalizable and efficient neural human rendering. In *European Conference on Computer Vision*, pages 222–239, 2022.
- [4] Xu Chen, Tianjian Jiang, Jie Song, Max Rietmann, Andreas Geiger, Michael J. Black, and Otmar Hilliges. Fast-SNARF: A Fast Deformer for Articulated Neural Fields, Dec. 2022.
- [5] Xu Chen, Yufeng Zheng, Michael J. Black, Otmar Hilliges, and Andreas Geiger. SNARF: Differentiable forward skinning for animating non-rigid neural implicit shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11594–11604, 2021.
- [6] Yue Chen, Xuan Wang, Qi Zhang, Xiaoyu Li, Xingyu Chen, Yu Guo, Jue Wang, and Fei Wang. UV Volumes for Real-time Rendering of Editable Free-view Human Performance. 2023.
- [7] Zijian Dong, Chen Guo, Jie Song, Xu Chen, Andreas Geiger, and Otmar Hilliges. PINA: Learning a personalized implicit neural avatar from a single RGB-D video sequence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20470–20480, 2022.
- [8] Artur Grigorev, Karim Iskakov, Anastasia Ianina, Renat Bashirov, Ilya Zakharkin, Alexander Vakhitov, and Victor Lempitsky. Stylepeople: A generative model of fullbody human avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5151–5160, 2021.
- [9] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Mesh optimization. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, pages 19–26, 1993.
- [10] Doug L. James and Christopher D. Twigg. Skinning mesh animations. *ACM Transactions on Graphics (TOG)*, 24(3):399–407, 2005.
- [11] Boyi Jiang, Yang Hong, Hujun Bao, and Juyong Zhang. Self-Recon: Self Reconstruction Your Digital Avatar from Monocular Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5605–5615, 2022.
- [12] Yoni Kasten, Dolev Ofri, Oliver Wang, and Tali Dekel. Layered neural atlases for consistent video editing. *ACM Transactions on Graphics (TOG)*, 40(6):1–12, 2021.
- [13] Ladislav Kavan, Steven Collins, Jiří Žára, and Carol O’Sullivan. Skinning with dual quaternions. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, pages 39–46, 2007.
- [14] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [15] Youngjoong Kwon, Dahun Kim, Duygu Ceylan, and Henry Fuchs. Neural Human Performer: Learning Generalizable Radiance Fields for Human Performance Rendering. In *Advances in Neural Information Processing Systems*, volume 34, pages 24741–24752. Curran Associates, Inc., 2021.
- [16] Samuli Laine and Tero Karras. Efficient sparse voxel octrees. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 55–63, 2010.
- [17] Verica Lazova, Vladimir Guzov, Kyle Olszewski, Sergey Tulyakov, and Gerard Pons-Moll. Control-nerf: Editable feature volumes for scene rendering and manipulation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4340–4350, 2023.
- [18] Ruilong Li, Julian Tanke, Minh Vo, Michael Zollhöfer, Jürgen Gall, Angjoo Kanazawa, and Christoph Lassner. Tava: Template-free animatable volumetric actors. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*, pages 419–436. Springer, 2022.
- [19] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020.
- [20] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural actor: Neural free-view synthesis of human actors with pose control. *ACM Transactions on Graphics (TOG)*, 40(6):1–16, 2021.
- [21] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5773–5783, 2021.
- [22] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):1–16, Nov. 2015.
- [23] Li Ma, Xiaoyu Li, Jing Liao, Xuan Wang, Qi Zhang, Jue Wang, and Pedro V. Sander. Neural parameterization for dynamic human head editing. *ACM Transactions on Graphics (TOG)*, 41(6):1–15, 2022.
- [24] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021.
- [25] Moustafa Meshry, Dan B. Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. Neural rerendering in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6878–6887, 2019.
- [26] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and

- Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, Lecture Notes in Computer Science, pages 405–421, Cham, 2020. Springer International Publishing.
- [27] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021.
 - [28] Keunhong Park, Utkarsh Sinha, Peter Hedman, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Ricardo Martin-Brualla, and Steven M. Seitz. HyperNeRF: A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields. *ACM Trans. Graph.*, 40(6), Dec. 2021.
 - [29] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10975–10985, 2019.
 - [30] Bo Peng, Jun Hu, Jingtao Zhou, and Juyong Zhang. SelfNeRF: Fast Training NeRF for Human from Monocular Self-rotating Video, Oct. 2022.
 - [31] Sida Peng, Junting Dong, Qianqian Wang, Shangzhan Zhang, Qing Shuai, Xiaowei Zhou, and Hujun Bao. Animatable neural radiance fields for modeling dynamic human bodies. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14314–14323, 2021.
 - [32] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9054–9063, 2021.
 - [33] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021.
 - [34] Yi-Ling Qiao, Alexander Gao, and Ming Lin. NeuPhysics: Editable Neural Geometry and Physics from Monocular Videos. In *Advances in Neural Information Processing Systems*, 2022.
 - [35] Amit Raj, Julian Tanke, James Hays, Minh Vo, Carsten Stoll, and Christoph Lassner. Anr: Articulated neural rendering for virtual avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3722–3731, 2021.
 - [36] Edoardo Remelli, Timur Bagautdinov, Shunsuke Saito, Chen-gelei Wu, Tomas Simon, Shih-En Wei, Kaiwen Guo, Zhe Cao, Fabian Prada, and Jason Saragih. Drivable Volumetric Avatars using Texel-Aligned Features. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022.
 - [37] Fabio Remondino. From point cloud to surface: The modeling and visualization problem. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34, 2003.
 - [38] Darius Rückert, Linus Franke, and Marc Stamminger. Adop: Approximate differentiable one-pixel point rendering. *ACM Transactions on Graphics (TOG)*, 41(4):1–14, 2022.
 - [39] Aliaksandra Shysheya, Egor Zakharov, Kara-Ali Aliev, Renat Bashirov, Egor Burkov, Karim Iskakov, Aleksei Ivakhnenko, Yury Malkov, Igor Pasechnik, and Dmitry Ulyanov. Textured neural avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2387–2397, 2019.
 - [40] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry Processing*, volume 4, pages 109–116, 2007.
 - [41] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-nerf: Articulated neural radiance fields for learning human shape, appearance, and pose. *Advances in Neural Information Processing Systems*, 34:12278–12291, 2021.
 - [42] Jiaming Sun, Xi Chen, Qianqian Wang, Zhengqi Li, Hadar Averbuch-Elor, Xiaowei Zhou, and Noah Snavely. Neural 3D reconstruction in the wild. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022.
 - [43] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P. Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022.
 - [44] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 33:7537–7547, 2020.
 - [45] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. Mega-NeRF: Scalable Construction of Large-Scale NeRFs for Virtual Fly-Throughs. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12912–12921, June 2022.
 - [46] Shaofei Wang, Katja Schwarz, Andreas Geiger, and Siyu Tang. Arah: Animatable volume rendering of articulated human sdbs. In *European Conference on Computer Vision*, volume 4, 2022.
 - [47] Chung-Yi Weng, Brian Curless, Pratul P. Srinivasan, Jonathan T. Barron, and Ira Kemelmacher-Shlizerman. Humanerf: Free-viewpoint rendering of moving people from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16210–16220, 2022.
 - [48] Fanbo Xiang, Zexiang Xu, Milos Hasan, Yannick Hold-Geoffroy, Kalyan Sunkavalli, and Hao Su. Neutex: Neural texture mapping for volumetric neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7119–7128, 2021.
 - [49] Yuliang Xiu, Jinlong Yang, Dimitrios Tzionas, and Michael J. Black. ICON: Implicit Clothed humans Obtained from Normals. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13286–13296, June 2022.

- [50] Tianhan Xu and Tatsuya Harada. Deforming Radiance Fields with Cages. In *European Conference on Computer Vision*, pages 159–175, 2022.
- [51] Bangbang Yang, Chong Bao, Junyi Zeng, Hujun Bao, Yinda Zhang, Zhaopeng Cui, and Guofeng Zhang. NeuMesh: Learning Disentangled Neural Mesh-Based Implicit Field for Geometry and Texture Editing. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, Lecture Notes in Computer Science, pages 597–614, Cham, 2022. Springer Nature Switzerland.
- [52] Yu-Jie Yuan, Yang-Tian Sun, Yu-Kun Lai, Yuwen Ma, Rongfei Jia, and Lin Gao. NeRF-editing: Geometry editing of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18353–18364, 2022.
- [53] Jiakai Zhang, Liao Wang, Xinhang Liu, Fuqiang Zhao, Minzhang Li, Haizhao Dai, Boyuan Zhang, Wei Yang, Lan Xu, and Jingyi Yu. NeuVV: Neural Volumetric Videos with Immersive Rendering and Editing. *arXiv:2202.06088 [cs]*, Feb. 2022.
- [54] Ruiqi Zhang and Jie Chen. NDF: Neural Deformable Fields for Dynamic Human Modelling. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, Lecture Notes in Computer Science, pages 37–52, Cham, 2022. Springer Nature Switzerland.
- [55] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.
- [56] Chengwei Zheng, Wenbin Lin, and Feng Xu. EditableNeRF: Editing Topologically Varying Neural Radiance Fields by Key Points. *arXiv preprint arXiv:2212.04247*, 2022.
- [57] Zerong Zheng, Han Huang, Tao Yu, Hongwen Zhang, Yandong Guo, and Yebin Liu. Structured local radiance fields for human avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15893–15903, 2022.
- [58] Tiancheng Zhi, Christoph Lassner, Tony Tung, Carsten Stoll, Srinivasa G. Narasimhan, and Minh Vo. TexMesh: Reconstructing Detailed Human Texture and Geometry from RGB-D Video. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, volume 12355, pages 492–509. Springer International Publishing, Cham, 2020.