Learning Locally Editable Virtual Humans

Hsuan-I Ho  Lixin Xue  Jie Song  Otmar Hilliges
Department of Computer Science, ETH Zürich

Abstract

In this paper, we propose a novel hybrid representation and end-to-end trainable network architecture to model fully editable and customizable neural avatars. At the core of our work lies a representation that combines the modeling power of neural fields with the ease of use and inherent 3D consistency of skinned meshes. To this end, we construct a trainable feature codebook to store local geometry and texture features on the vertices of a deformable body model, thus exploiting its consistent topology under articulation. This representation is then employed in a generative auto-decoder architecture that admits fitting to unseen scans and sampling of realistic avatars with varied appearances and geometries. Furthermore, our representation allows local editing by swapping local features between 3D assets. To verify our method for avatar creation and editing, we contribute a new high-quality dataset, dubbed CustomHumans, for training and evaluation. Our experiments quantitatively and qualitatively show that our method generates diverse detailed avatars and achieves better model fitting performance compared to state-of-the-art methods. Our code and dataset are available at https://ait.ethz.ch/custom-humans.

1. Introduction

3D Avatars are an important aspect of many emerging applications such as 3D games or the Metaverse. Allowing for easy personalization of such avatars, holds the promise of increased user engagement. Traditionally, editing 3D assets requires knowledge of computer graphics tools and relies on standardized data formats to represent shapes and appearances. While methods for reconstruction or generative modeling of learned avatars achieve impressive results, it is unknown how such neural avatars can be edited and customized. Thus, the goal of our work is to contribute a simple, yet powerful data-driven method for avatar creation and customization (Fig. 1): our method enables (a) the ability to transfer partial geometric and appearance details between 3D assets, and (b) the ability to author details via 2D-3D transfer. The resulting avatars (c) retain consistent local details when posed.

Existing methods do not allow for such capabilities. While 3D generative models of articulated human bodies [5, 21, 25, 42, 75] leverage differentiable neural rendering to learn from images, they cannot control local details due to highly entangled color and geometry in the 2D supervision signal. Generative models trained on 3D data [9, 13, 36, 44, 45] can produce geometric details for surfaces and clothing. However, the diversity of generated samples is low due to the lack of high-quality 3D human scans and not all methods model appearance.

At the core of the issue lies the question of representation: graphics tools use meshes, UV, and texture maps which provide consistent topologies under deformation. However, human avatar methods that are built on mesh-based representations and linear blend skinning (LBS) are limited in their representational power with respect to challenging geometry (e.g., puffy garments) and flexible topologies (e.g., jackets), even with adaptations of additional displacement parameters [36] and mesh subdivision [66].
Inspired by the recent neural 3D representations [40, 62, 70, 72], we propose a novel hybrid representation for digital humans. Our representation combines the advantages of consistent topologies of LBS models with the representational power of neural fields. The key idea is to decompose the tasks of deformation consistency on one hand and local surface and appearance description on the other. For the former, we leverage existing parametric body models (e.g., SMPL [33] and SMPL-X [48]). For the latter, we leverage the fixed topology of the poseable mesh to store local feature codebooks. A decoder, shared across subjects, is then conditioned on the local features to predict the final signed distance and color values. Since only local information [15] is exposed to the decoder, overfitting and memorization can be mitigated. We experimentally show that this is crucial for 3D avatar fitting and reposing.

Complementing this hybrid representation, we propose a training pipeline in the auto-decoding generative framework [9, 46, 52]. To this end, we jointly optimize multi-subject feature codebooks and the shared decoder weights via 3D reconstruction and 2D adversarial losses. The 3D losses help in disentangling appearance and geometric information from the input scans, while the latter improves the perceptual quality of randomly generated samples. To showcase the hybrid representation and the generative model we implement a prototypical avatar editing workflow shown in Fig. 1.

Furthermore, to enable research on high-quality 3D avatars we contribute training data for generative 3D human models. We record a large-scale dataset (more than 600 scans of 80 subjects in 120 garments) using a volumetric capture stage [11]. Our dataset consists of high-quality 3D meshes alongside accurately registered SMPL-X [48] models and will be made available for research purposes. Finally, we assess our design decisions in detailed evaluations, both on existing and the proposed datasets.

In summary, our contributions are threefold: (a) a novel hybrid representation for 3D virtual humans that allows for local editing across subjects, (b) a generative pipeline of 3D avatars creation that allows for fitting to unseen 3D scans and random sampling, and (c) a new large-scale high-quality dataset of 3D human scans containing diverse subjects, body poses and garments.

2. Related Work

Controllable human representations. Topics of virtual humans have received much attention in the graphics literature, such as skinning and rigging of articulated meshes [33, 43, 48, 55], physical simulation of clothing [19, 22, 41], and deferred rendering [24, 49, 65]. With the advances in neural rendering [63, 64] and the availability of large-scale human datasets [18, 28, 29, 32, 47, 67, 73, 74], numerous approaches have been proposed to reconstruct [4, 56, 57] and explicitly control [10, 30, 50] human avatars in a data-driven manner.

One branch of work focuses on 2D image synthesis via generative adversarial networks (GANs) [20] and techniques of feature manipulation [28, 54, 60]. Typically, a 2D neural renderer creates human images corresponding to pose and appearance latent codes learned from the training data. Related applications such as virtual try-on [16, 17, 71] and video retargeting [7, 31, 68] have shown promising results in light of photo-realistic image synthesis by GANs [18, 26]. However, these methods do not explicitly reason about complex 3D human geometry and can therefore not produce 3D-consistent results.

A newly emerging line of work aims to create controllable avatars with 3D consistency. Some methods extend existing body models with neural networks to predict displacement layers [3, 6, 36] or textures [51]. Other methods learn to model challenging pose-dependent deformations on avatars either by predicting LBS weights [8, 10, 14, 58, 59, 76] or improving the capabilities of body models [23, 30, 34, 35, 37, 50, 53] with the power of implicit neural fields. However, these approaches mainly focus on modeling a single subject in specific clothing and do not scale to create diverse avatars. Our method overcomes this issue by learning a multi-subject generative model which produces realistic virtual humans with disentangled controllability over body poses, clothing geometry, and texture.

Generative 3D human models. Existing generative models of human avatars can be loosely split into two main streams: learning 3D-aware neural rendering from 2D images [5, 21, 25, 42, 75] and learning body shapes from 3D supervision [9, 13, 36, 44, 45]. Powered by recent advances in differentiable neural rendering [64] and neural fields [70], much progress has been made in 3D-aware generative models [61]. However, learning to generate detailed clothed avatars from pure 2D supervision [5, 21, 25, 42, 75] is still challenging due to the complex appearance and articulation of bodies, self-occlusions, and highly entangled colors and geometries in images.

More closely related to our setting are methods that learn to generate detailed body shapes from 3D scans or RGB-D data. For instance, CAPE [36] and SMPLICITY [13] are generative models for clothed humans. The former exploits VAE-GAN to predict additive displacements based on the SMPL vertices while the latter drape an implicitly modeled garment layer onto SMPL. NPMs [44], and SPAM [45] learn pose and shape latent spaces from 3D supervision, which enables latent code inversion using point clouds or depth sequences. gDNA [9] learns to synthesize body shapes in the canonical space and further improves clothing details with adversarial losses. However, none of the above-mentioned works is able to generate human bodies with appearance and neither allows fine-grained editing of the generated avatars. Our method addresses both issues by learning disentangled
local representations for multiple subjects. In addition, we experimentally show that our representation significantly improves the performance of model fitting against state-of-the-art human generative models.

3. Method

Our proposed method is summarized in Fig. 2. We first contribute a novel hybrid human representation that stores local geometric and textural information into two aligned feature spaces (Sec. 3.1 and Fig. 3). To allow fitting to new 3D scans and drawing random samples from the underlying data distribution, we design a training strategy to learn a meaningful latent space under the generative adversarial framework to bring in additional 2D adversarial supervision (Sec. 3.2 & Sec. 3.3). Finally, we demonstrate the utility of our method for creating avatars by enabling local feature editing through existing 3D assets or images (Sec. 3.4).

3.1. Hybrid Representation of Humans

To enable 3D avatars with high-fidelity representational power and local editing capabilities, a suitable representation is needed. To this end, we propose a novel hybrid representation that combines the advantages of neural fields (flexibility and modeling power) with LBS-articulated mesh models (ease of deformation and full explicit control).

An overview of how we leverage the proposed representation is provided in Fig. 2 in the dotted blue box. Given a human scan, we first create a posed, coarse body mesh $\mathcal{M}$ (shown in red) using the registered body parameters $(\theta, \beta)$ of an LBS body model. The mesh consists of $M$ vertices ($\mathcal{V} \in \mathbb{R}^{M \times 3}$) in the posed space and $F$ faces where $\mathcal{F} \in \{1, \ldots, M\}^{M \times 3}$ defines the vertex indices on each face. We then construct a trainable feature codebook $C \in \mathbb{R}^{M \times 2F}$, which stores $F$-dimensional local geometry and texture features respectively for each vertex.

Similar to coordinate-based implicit fields, a 3D coordinate $x_g \in \mathbb{R}^3$ is used to predict its corresponding signed distance $s(x_g)$, and RGB color $c(x_g)$. Instead of using global coordinates directly as inputs, we condition neural field decoders on the local triangle coordinates $x_l \in \mathbb{R}^3$ and the local geometry and texture features $f_l$, $f_s \in \mathbb{R}^F$. We illustrate this conversion from global coordinates to local triangle coordinates in Fig. 3. The global coordinates $x_g$ are first projected onto the mesh by finding the closest point $x^*_g$ (Fig. 3, blue dot):

$$x^*_g = \arg\min_{x_c} \|x_g - x_c\|_2,$$

$$x_c = B_{u,v}(\mathcal{V}[m_0, m_1, m_2]),$$

where $(m_0, m_1, m_2)$ are vertex indices of faces $\mathcal{F}$ and $B_{u,v}(\cdot)$ is the barycentric interpolation function with barycentric coordinates $(u, v, 1 - u - v)$. The closest point $x^*_g$ within the face is used to transform $x_l$ into a local triangle coordinate system. Hence, $x_l$ consists of the barycentric coordinates $(u, v)$, the signed distance $d$ between $x_g$ and $x^*_g$, i.e., $x_l := (u, v, d)$. We also compute a direction vector $\vec{n}$ between $x_g$ and $x^*_g$ as an additional feature to distinguish points near triangle edges. To query local features ($f_l, f_s$), we use the vertex indices on the triangle $(m_0^l, m_1^l, m_2^l)$ to

![Figure 2. Proposed framework. Given a posed scan registered with body pose and shape LBS parameters $(\theta, \beta)$, our proposed human representation stores its local geometry and texture features in a codebook $C$ which is indexed by the vertex indices of $M$-vertex LBS body mesh $\mathcal{M}$ (Sec. 3.1). Given a query point coordinates $x_g$, two separated MLP decoders $\Psi, \Phi$ predict signed distances and colors conditioned on the positional features $(x_l, \vec{n})$ and the local geometry/texture features $(f_l / f_s)$ respectively. We train a generative auto-decoder using $N$ posed scans, whose feature codebooks are stored in the dictionary $D_s, D_c$. We introduce two sampling strategies to sample codebooks (denoted as $C_s, C_c$ respectively) and jointly train our dictionaries and shared MLP decoders with a 3D reconstruction loss and a 2D adversarial loss (Sec. 3.2 & Sec. 3.3).](image-url)
information of the $i$-th subject respectively. The entries $C_i$ can then be learned jointly with the decoder weights via direct 3D supervision using the $i$-th scan (Fig. 2). However, we experimentally show that this is insufficient to learn a well-behaved latent space from which we can draw novel samples (see Fig. 10).

Therefore we introduce a codebook sampling strategy that allows us to draw random samples and update the entries of the dictionaries $D_a, D_c$ via an additional 2D adversarial loss. More specifically, we follow the auto-decoder architecture [52] and perform PCA on the reshaped dictionary to compute eigenvectors $V \in \mathbb{R}^{D \times (MF)}$ and fit a normal distribution to the $D$-dimensional PCA coefficients of $N$ samples. A new random codebook $C_r$ can then be generated by sampling $D$-dimensional PCA parameters and multiplying them with the eigenvectors $V$ (See Supp-B.1 for details). Note that our representation disentangles shapes from appearances with separated geometry and texture branches, which enables independent sampling of geometry and texture features.

### 3.3. Model Training

#### 3D reconstruction loss

To train a codebook $C_i$ with a single scan, we sample points in a thin shell around the scan. For each coordinate we compute its signed distance $s$ to the input scan, closest texture color $c$, and surface normal $n$ on the input scan to attain ground truth values. The codebooks and the decoder weights are then optimized via the following losses:

\[
\mathcal{L}_{sdf} = \|s - s(x_g)\|_1 + \lambda_n \|1 - n \cdot \nabla_{x_g} s(x_g)\|_1, \tag{4}
\]

\[
\mathcal{L}_{rgb} = \|c - c(x_g)\|_1, \tag{5}
\]

\[
\mathcal{L}_{3D} = \lambda_{sdf}\mathcal{L}_{sdf} + \lambda_{rgb}\mathcal{L}_{rgb}. \tag{6}
\]

#### 2D adversarial loss

Adversarial learning does not require exact ground-truth annotations but is trained via a collection of real and fake (rendered) images. Thus, real images are obtained by rasterizing the ground-truth scan to which the coarse body mesh $M$ was fitted. Color and normal images (denoted as “Real Patch” in Fig. 2) are used for learning texture and geometry respectively. Using the same virtual camera parameters and the coarse mesh $M$, we attain rendered patches (denoted as “Rendered Patch” Fig. 2) via implicit surface rendering of a sampled codebook $C_r$. Please refer to Supp-B.2 for more details.

Using these 2D patches, we train dictionaries, decoders, and discriminators jointly with a non-saturating logistic loss $\mathcal{L}_{adv}$ [20], RI regularization $\mathcal{L}_{RI}$ [38], and path length regularization $\mathcal{L}_{path}$ [27]. Note that these losses do not require exact ground-truth replication. Furthermore, we regu-
larize the feature dictionaries to follow a Gaussian distribution [46] with \( \mathcal{L}_{\text{reg}} = \| D \|_F \). In summary, we optimize the discriminator:

\[
\mathcal{L}_{\text{dis}} = \mathcal{L}_{\text{adv}} + \lambda_{R1} \mathcal{L}_{R1},
\]

while updating the remaining components (\( D_c, D_s, \Phi, \Psi \)):

\[
\mathcal{L} = -\mathcal{L}_{\text{adv}} + \mathcal{L}_{3D} + \lambda_{\text{path}} \mathcal{L}_{\text{path}} + \lambda_{\text{reg}} \mathcal{L}_{\text{reg}},
\]

where \( \lambda_{(\cdot)} \) denotes weights to balance the losses.

Since we sample on the fly during training (see Sec. 3.2), the 2D adversarial loss does affect the shared decoders and the whole feature dictionaries (See Supp-Fig.14 for details).

### 3.4. Feature Editing and Avatar Customization

We now describe how we integrate the above-mentioned human representation and the generative architecture into the avatar creation workflow shown in Fig. 1.

**Avatar initialization.** To simplify the avatar creation process, our method allows users to start with a default example \( C_d \), which can be queried from the trained codebook dictionaries directly with index \( i \) (\( C_i \)) or randomly sampled from the learned D-dimensional PCA parameters distribution (\( C_r \) in Sec. 3.2).

**Model fitting.** Being able to extract elements of interest or copying from existing 3D assets is necessary for avatar creation and editing. To this end, we leverage a similar technique to GAN inversion [69], where decoder parameters \( \Phi \) and \( \Psi \) are frozen and we only optimize a new feature codebook \( C_{fit} \) to fit a 3D scan.

Given a 3D target scan and its corresponding body model parameters, we calculate the 3D reconstruction loss in Eq. (6) between the prediction conditioned on \( C_{fit} \) and the ground-truth scan, i.e.,

\[
C_{fit} = \arg \min_{C} (\lambda_{\text{sd}} L_{\text{sd}} + \lambda_{\text{rg}} L_{\text{rg}}).
\]

Note that our generative neural fields (MLP decoders) are conditioned on descriptive local features. We show that it allows us to accurately fits complex clothing geometry and unseen textural patterns in Sec. 4.3.

**Cross-subjects feature editing.** With multiple codebooks each representing different 3D assets, we can easily transfer local geometry and texture from one avatar to another, for instance, changing the top wear from the fitted scan \( C_{fit} \) to the initial \( C_d \). Recalling that all codebooks are indexed by an identical mesh topology, users can easily retrieve the index numbers of \( V_{\text{body}} \subset V \) via standard mesh visualization tools such as Blender [12]. Finally, swapping of the corresponding rows in \( C_{fit} \) and \( C_d \) given the vertex indices also swaps the local appearance.

![Figure 4. Randomly sampled texture and body geometry from the model trained on THuman2.0. Top: Given any poses, our randomly sampled geometries contain realistic details such as wrinkles in garments and facial expressions. Bottom: Given arbitrary poses and body geometries, our model produces reasonable colors on skin, hair, clothes, and pants in each sample (We turn off the shader for visualizing pure texture colors).](image)

4. Experiments

Our goal is locally editable 3D avatar creation. Since we are the first to discuss this problem, we visualize our editing results in Sec. 4.2. Next, we highlight the capability of model fitting by comparing our method with SOTA human generative models in Sec. 4.3. Finally, controlled experiments are presented in Sec. 4.4 and Sec. 4.5 to verify the effectiveness of our design.
4.1. Experiment Settings

**Dataset.** Most generative human works [9, 44, 45] exploit commercial data [1, 2] for training, which is not easily accessible and limits reproducibility. Furthermore, the quality of publicly available 3D human datasets [66, 73] is not satisfactory. Issues such as non-watertight topologies and noise are very common (See Supp-A.2 for examples and comparison). To bridge this gap, we collect a new dataset named CustomHumans for training and evaluation. Here we summarize the datasets used in our experiments.

- **CustomHumans** (Ours) contains more than 600 high-quality scans of 80 participants in 120 garments in varied poses from a volumetric capture stage [11], which is equipped with 106 synchronized cameras (53 RGB and 53 IR cameras). We use our dataset to train models of all quantitative experiments. (Sec. 4.3 ⇐ Sec. 4.5)

- **THuman2.0** [73] is a dataset containing about 500 scans of humans wearing 150 garments in various poses. Since this dataset has more textural diversity, we train our method on it for qualitative random sampling experiments (Sec. 4.2 and Sec. 4.4).

- **SIZER** [66] is a widely used 3D scan dataset containing A-pose human meshes of 97 subjects in 22 garments. These meshes are used as unseen test scans in our fitting experiment (Sec. 4.3).

**Evaluation protocol.** Following the evaluation protocol in OccNet [39], we quantitatively evaluate the model fitting accuracy using three metrics: **Chamfer distance (CD)**, **normal consistency (NC)**, and **f-Score**.
4.4. Ablation Study

Effectiveness of local features and shared decoders. To verify the design choice of using local features, we replace the local features $x_i$ by the global coordinates $x_g$ for conditioning the decoders. Fig. 9 shows that even though the shared decoders are able to achieve similar reconstruction results when training with global information, they do not maintain consistent performance for model fitting and avatar reposing. This is because the shared decoders tend to memorize global coordinates information in a “per-subject” manner, rather than learning shareable information that can be used across vertices and subjects. On the other hand, our representation ensures only local features defined on the triangle coordinates are exposed to the shared decoders. In such cases, the decoders can better handle unseen body poses or out-of-distribution samples for model fitting and avatar editing.

Importance of 2D adversarial loss and 3D disentanglement. As discussed in Sec. 3.3, we introduce feature disentanglement and generative adversarial learning in our training framework. As shown in Fig. 10, sampling within the feature spaces learned without adversarial loss does not yield reasonable body textures. Similarly, training only a single decoder for both geometry and texture does not allow us to maintain desired body geometries when sampling random textures. Our full model can produce disentangled textures, given arbitrary body geometries and poses.
### Table 2. Generalization analysis on CustomHumans. We analyze the model fitting performances with regard to different amounts of training data (100% = 100 training scans). We observe consistent performance gain on all evaluation metrics when using more training subjects to train the shared decoders.

<table>
<thead>
<tr>
<th>Training Data Percentage</th>
<th>10%</th>
<th>25%</th>
<th>50 %</th>
<th>75%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chamfer Distance (mm)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-to-P / P-to-S ↓</td>
<td>1.933/1.798</td>
<td>1.754/1.590</td>
<td>1.543/1.456</td>
<td>1.463/1.385</td>
<td>1.423/1.364</td>
</tr>
<tr>
<td>Normal Consistency↑</td>
<td>0.918</td>
<td>0.931</td>
<td>0.935</td>
<td>0.947</td>
<td>0.949</td>
</tr>
<tr>
<td>f-Score (%) ↑</td>
<td>99.25</td>
<td>99.38</td>
<td>99.65</td>
<td>99.74</td>
<td>99.75</td>
</tr>
</tbody>
</table>

4.5. Generalization Ability Analysis

We are interested in how the amount of training data affects the capacities of the MLP decoders. To analyze this, we design three evaluation protocols: 3D model fitting, avatar reposing, and 2D texture fitting. Tab. 2 summarizes the model fitting performance using different percentages of training data. We observe a 25% accuracy improvement when using the full training set. In Fig. 11 (Top) we show that the reposing artifacts caused by self-contact (e.g., fist and elbow) can be reduced when more training subjects are introduced. Bottom: We visualize the results using different percentages of training data. Using more data results in more robust shared decoders (with a wider color range), which is necessary for the avatar creation task.

5. Conclusion

We propose an end-to-end trainable framework for learning 3D human avatars with high fidelity and full editability. By combining neural fields with explicit skinned meshes, our representation addresses the controllability issue of many previous implicit representations. Moreover, we uniquely integrate the proposed human representation into a generative auto-decoding pipeline that enables local editing across multiple animation-ready avatars. Through our evaluation on the newly contributed CustomHumans dataset, we demonstrate that our approach achieves higher model fitting accuracy and generates diverse detailed avatars. We believe that this work opens up exciting possibilities for accelerating content creation in the Metaverse.

Acknowledgements We express our gratitude to Stefan Walter and Dean Bakker for infrastructure support, Juan Zarate for managing the capture stage, and Deniz Yildiz and Laura Wülffroth for data capture assistance. We thank Andrew Searle for supporting the capturing system and all the dataset participants.
References

[24] Pat Hanrahan and Paul Haeberli. Direct wsiwyg painting and texturing on 3d shapes: (an error occurred during the printing of this article that reversed the print order of pages 118 and 119, while we have corrected the sort order of the 2 pages in the dl, the pdf did not allow us to repaginate the 2 pages.). ACM Transactions on Graphics (TOG), 24(4):215–223, 1990. 2
[26] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönens, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free


ings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 1810–1819, 2021. 2

Supplementary Material for
“Learning Locally Editable Virtual Humans”

Hsuan-I Ho Lixin Xue Jie Song Otmar Hilliges
Department of Computer Science, ETH Zürich

Contents

A CustomHumans Dataset 1
A.1. Dataset description . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
A.2. Comparison with existing datasets . . . . . . . . . . . . . . 1

B Implementation Details 2
B.1. Codebook sampling . . . . . . . . . . . . . . . . . . . . . . . . . 2
B.2. Implicit rendering . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
B.3. Network architecture . . . . . . . . . . . . . . . . . . . . . . . . . . 3
B.4. Training details . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
B.5. Inference speed . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

C More Comparisons and Results 4
C.1. Avatar editing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
C.2. Random sampling . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
C.3. Model fitting . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
C.4. Reposing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

D Discussion 5
D.1. Scalability . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
D.2. Ethical Concerns . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
D.3. Limitations and future works . . . . . . . . . . . . . . . . . . . . . 5

A. CustomHumans Dataset

A.1. Dataset description
In this section, we provide more information about our contributed dataset, CustomHumans. Our dataset is recorded by a multi-view photogrammetry system [4] as shown in Fig. 12, equipped with 53 RGB (12 Megapixels) and 53 (4 Megapixels) IR cameras. The resulting high-quality scan is composed of a 40K-face mesh alongside a 4K-resolution texture map. In addition to the high-quality scans, we provide accurately registered SMPL-X parameters using a customized mesh registration pipeline.

To collect clothed human scans in various poses, we invited 80 participants to our capture studio. We designed several movement instructions for the participants, such as “T-pose”, “Hands Up”, “Squat”, “Turing head”, and “Hand gestures”, to film 5-6 poses in a 10-second long sequence (300 frames). We selected 4-5 best-quality meshes in each sequence as our data samples. In total, our dataset contains more than 600 high-quality scans with 120 different garments. Exemplars of human scans can be found in Fig. 17, where we visualize the textured scans, mesh geometries, and the registered SMPL-X body models.

A.2. Comparison with existing datasets

We summarize the outstanding features of existing 3D clothed human datasets in Tab. 3. Specifically, we are mainly interested in four aspects. Subject Diversity: Does it contain people of diverse genders and races? Garment Diversity: Does it include various clothing and combinations? Pose Variation: Does it consists of subjects in various poses? Quality: Do these scans contain noise near the surfaces? and are they watertight?

Commercial datasets (e.g., RenderPeople [1]) have shown superior quality and have been used in many works of generative modeling. However, they are not easily accessible which limits reproducibility for research purposes. CAPE [8] contains posed sequences of 15 subjects and 8 types of outfits. Since only SMPL+D body meshes are
Dataset | Subject Diversity | Garment Diversity | Pose Variation | Noise-free | Watertight | Registered | Publicly Available
--- | --- | --- | --- | --- | --- | --- | ---
RenderPeople [1] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓
CAPE [8] | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓
SIZER [13] | ✓ | | | ✓ | ✓ | ✓ | ✓
THuman2.0 [15] | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓
Ours | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓

Table 3. **Comparison with existing 3D human scans datasets.** Commercial datasets such as RenderPeople are not easy to obtain whereas either the quality (e.g. SIZER and THuman2.0) or diversity (e.g., CAPE) is not sufficient in the other datasets.

Figure 13. **Examples of noisy data in SIZER [13] (left) and THuman2.0 [15] (right).** The meshes in SIZER and THuman2.0 are generally not watertight and bumpy, which causes issues when learning detailed human body geometry.

used in this dataset, they are clean and watertight. The total number of subjects and garments is limited as the main focus of this dataset is to model pose-dependent deformations of a single subject. SIZER [13] provides 2000 scans of 100 different subjects in a total of 22 types of garments. Each scan is captured in “A-Pose” and registered by SMPL and SMPL+D body models. Nevertheless, as shown in Fig. 13 left, these scans contain non-watertight mesh manifolds and large noise near surfaces. THuman2.0 [15] consists of 525 scans of approximately 150 subjects and garments captured by a dense DLSR rig, which limits the mesh quality due to noise and bumpy surfaces (Fig. 13 right). To foster future research on creating detailed human avatars, we addressed the above-mentioned issues and collected a new high-quality dataset containing 600 posed scans with higher subject diversity and in various clothing.

**B. Implementation Details**

**B.1. Codebook sampling**

Fig. 14 depicts the codebook sampling strategies used for training our model. As mentioned in Sec. 3.2, we store the codebooks $C_i$ in two dictionaries $D_s, D_c \in \mathbb{R}^{N \times (MF)}$ to represent shape and color information of the $i$-th subject. This allows us to learn a unified shared feature space and transfer local feature codebooks across subjects. The entry $C_i$ is queried to be jointly trained with the decoder weights via direct 3D supervision.

To further learn a well-behaved latent space from which we can draw novel samples, we devise an on-the-fly PCA codebook sampling strategy inspired by [12]. We first compute $D$-dimensional eigenvectors $V \in \mathbb{R}^{D \times (MF)}$ by applying PCA to the reshaped dictionary. We then derive the $D$-dimensional PCA coefficients of all $N$ samples using the eigenvectors $V$. The mean and the covariance matrix of these PCA coefficients can be computed and used for fitting a $D$-dimensional normal distribution. We then draw random PCA coefficients $k$ from the normal distribution and compute a new codebook $C_r$ by multiplying $k$ and $V$.

As shown in Fig. 14, the 3D loss is used for updating only the selected codebook $C_i$. On the other hand, the 2D adversarial loss can be backpropagated to the entire dictionary since the on-the-fly PCA operation is differentiable. This PCA sampling strategy allows us to learn a more meaningful latent space for all training samples instead of overfitting each sample independently.

**B.2. Implicit rendering**

As described in Sec. 3.3, we render local color and normal patches for adversarial learning by rasterizing the ground-truth scans. To do so, we place the hip joint of each human scan at the origin and place 4 virtual cameras on $\{0, 90, 180, 270\}^\circ$ of a 2-meter circle. We rasterize images...
of the full body in $1024 \times 1024$ and then crop each image to $25 \times 128 \times 128$ patches based on the body joint positions defined on the SMPL-X model.

We then use the same virtual camera parameters to shoot rays (i.e., pixels on the image patches) from the camera center onto the implicit surface. Sample points on a camera ray can be formulated as $x = r_o + t \times r_d$, where $r_o$ is the ray origin, $r_d$ is the ray direction and $t$ is a scalar for sampling. We determine the intersection by finding the first SDF sign-changing sample along the ray following [2]. These intersection coordinates will be the query points for the feature querying and decoding process to predict corresponding colors and SDF. We compute the finite differences of SDF as an approximation of surface normals. The resulting normal and color maps are served as “fake image patches” for the discriminators.

### B.3. Network architecture

We choose SMPL-X [11] as our LBS body mesh, which consists of $M = 10475$ vertices. We use a feature dimension of $F = 32$ for both texture and geometry features, resulting in a codebook of $10475 \times 64$ for each subject. A positional encoding of 5 frequency bands is applied to the local positional features $x_i$. Our shared decoders consist of 4 layers of 128-dimensional linear layer followed by a ReLU activation. Our discriminator follows similar network architecture with StyleGAN2 [7]. We apply two different discriminators for normal maps and color images.

### B.4. Training details

We use a dictionary size of $N = 150$ for THuman2.0 [15] and $N = 100$ for the CustomHumans dataset. For better facial textures and finger control, we register both datasets with SMPL-X parameters $\theta, \beta$ including facial contours and finger joints.

As mentioned in Sec. 3.3, the discriminators are trained with R1 regularization $\mathcal{L}_{R1}$ [9] with $\lambda_{R1} = 10$. For training the decoders and the feature dictionaries, we set $\lambda_n = 10^{-2}, \lambda_{sdf} = 10^{3}, \lambda_{rgb} = 10^{2}, \lambda_{path} = 2, \lambda_{reg} = 10^{-3}$. We select a PCA dimension of $D = 16$ for geometry features and $D = 8$ for texture features during training.

For each training iteration, we sample 20480 query points near ground-truth mesh surfaces, and a batch size of 8 is used. Our decoders and feature dictionaries are trained end-to-end with Adam Optimizer using a learning rate of 0.001 and first- and second-momentum of 0 and 0.99 respectively. The training takes around two days on an RTX 3090 for 8000 epochs for both datasets.

### B.5. Inference speed

Optimizing the geometry converges in 100 iterations (∼40s), fitting textures to a 2K image takes additional 300 iterations (also ∼40s). Our method is currently meant for offline editing; note that our code is unoptimized and can be further accelerated. The resulting meshes are obtained from the predicted SDF values using marching cubes with a resolution of $300^3$ in ∼10 seconds.
Figure 15. **Avatar customization pipeline.** Two exemplars of avatar creation using our proposed framework. (a) Starting from a random body geometry and texture (Top) or training sample saved in our feature dictionary (Bottom), we keep human identities (i.e., face textures, geometries) unchanged while gradually (b) editing new clothing geometry and (c) textures onto our avatar by fitting to unseen 3D scans and 2D images. (d) Note that our method can handle challenging loose clothing such as jackets and coats and keep their colors and geometries consistent under various poses.

### C. More Comparisons and Results

#### C.1. Avatar editing

We present more details and exemplars of customized avatars using our editing workflows in Fig. 15. Starting from a random sample (i.e., \( C_r \) in Fig. 15(a) Top) or from a subject used in training (i.e., \( C_i \) in Fig. 15(a) Bottom), we gradually add new clothing and textures while keeping the identity (i.e., face colors and geometries) and body pose fixed for both cases. First, we invert the target scans into a feature codebook and store it for later use. Note that the body poses of target scans and initial avatars do not need to match. After fitting target scans into codebooks, we change the clothes on the initial avatars by swapping the local features located in the body regions. As shown in Fig. 15(b) our method is able to transfer challenging garments including jackets and preserves local details under different body poses and shapes. Furthermore, our method allows users to draw personalized logos and letters on garments. We keep the body geometry (Fig. 15(b)) unchanged while transferring only color information from photos of different subjects and clothing. Fig. 15(c) shows that texture features can be applied to avatars with different poses, shapes, and clothing geometry. Finally, the resulting avatars enable pose control by changing the SMPL-X parameters without affecting the fitted texture and geometry (Fig. 15(d)).

#### C.2. Random sampling

Following the same experiment in Sec. 4.2, we visualize more randomly sampled mesh geometries from gDNA [2] and our method in Fig. 18 and Fig. 19, respectively. As mentioned in the main manuscript, gDNA predicts additional normal maps on the surfaces, thus, their generated meshes do not contain real high-frequency details as shown in Fig. 18. Our method, instead, generates more detailed mesh geometries by introducing a 2D adversarial loss during training.

#### C.3. Model fitting

Similar to the experiment in Sec. 4.3, we follow the baselines and the selected test subjects introduced in [2] to compare with more human generative baselines, including SMPLicit [5], NPMs [10], and gDNA [2]. We visualize the fitting results of the selected subjects in Fig. 20. In all cases, our results are qualitatively closer to the ground truth compared to other baselines.

#### C.4. Reposing

In Fig. 21 Top, we repose the created avatars using the motion sequence provided in the AIST [14] dataset. Our representation consistently applies local details to the 3D avatar in different unseen poses.
**D. Discussion**

**D.1. Scalability**

We propose an auto-decoding pipeline that stores learned latent features of $N$ scans in a dictionary. We show that the dictionary size is not a major memory bottleneck since each entry occupies $10475 \times 32 \times 2 \times 4$ bytes = 2.68MB GPU memory. Even with 1000 scans – which exceeds the size of the existing datasets – the memory usage (2.68GB) easily fits onto modern GPUs. In terms of computational time, the PCA operation on a $1000 \times 320K$ dictionary takes 30ms on a single RTX3090. As a reference, one training iteration takes 0.8s. Furthermore, if one wants to apply our method to an even larger dataset (100K), the dictionary and PCA sampling could be replaced by a network that maps a global feature to local feature codebooks as is done in [6].

**D.2. Ethical Concerns**

Data collection and experiments in this work strictly follow the CVPR 2023 Ethics Guidelines. Our data collection procedure has been reviewed and approved by the responsible Institutional Review Board.

Generative modeling and editing of virtual humans are often accompanied by manipulation or fake information. Potential misuse of our creation pipeline to recreate full-body deep fakes for improper applications cannot be fully ruled out even though none of the proposed techniques intend for these purposes. In addition, possible concerns about copyright and privacy might arise since our framework enables the transfer of high-quality local details and facial appearances. Thus, to balance the positive and negative impacts, definite regulations must be established such as creative licenses and personal data protection. By making our code and data publicly available, we hope to raise awareness of future research on detecting fake information on photo-realistic 3D avatars in the Metaverse.

**D.3. Limitations and future works**

**Pose-dependent deformations.** Our method assumes each human scan is a static observation. Therefore, clothing changes caused by motion and poses cannot be explicitly modeled by local features. However, this is a data- not a model limitation. Once sufficiently large datasets become available, pose-dependent terms can be incorporated into our pipeline similar to [3]. Hence, one exciting direction is adding pose-dependent terms to the decoders and introducing more synthetic data under various poses to see if the representation and models can extract useful pose-dependent information.

**Challenging textural details.** As shown in Fig. 16, our method still suffers from very complex logos. Our implementation uses SMPL-X which limits the modeling of very high-frequency details. Note that our method can be used with any mesh template with topological consistency and thus inherently scales to higher resolution representations. Thus, a promising extension of our method would be using a customized human body model that has more vertices on the human body.

**Self-intersection.** Self-intersections are a well-known issue for learning SDFs since they cause wrong surface samples. To stabilize training, we excluded 3D scans with severe self-contacts in the dataset. As shown in Fig. 21 Bottom, self-intersections are also an issue in reposing avatars with contact. Therefore, one potential follow-up is to consider both SDF and occupancy and introduce an additional pose-dependent correction for tackling body poses with self-contacts. We also found that in our generalization analysis, training the decoders with more scans and poses reduced artifacts caused by self-intersections. This motivates us to address this issue in a data-driven manner.

**Automatic and interactive editing.** Our pipeline still requires users to manually select the vertices of interest for avatar editing which potentially limits the capability of editing complicated attributes such as facial expressions and hairstyles. One possible solution might be automating the current pipeline using a semantic human parsing algorithm as guidance. Moreover, another exciting extension is optimizing the fitting and inference speed and combing an interactive user interface for online avatar creation.
Figure 17. **Visualization of data samples in CustomHumans.** In our dataset, we provide high-quality human meshes, high-resolution texture maps, and registered SMPL-X body models. To verify the accuracy of registration, we visualize the overlays of the human scans and the SMPL-X body meshes.
Figure 18. **Random sampled meshes from gDNA** [2]. The meshes obtained from gDNA do not contain wrinkles and other high-frequency details.

Figure 19. **Random sampled meshes from our model trained on THuman2.0**. Our method can randomly sample meshes that contain more stochastic wrinkles on the clothes and more detailed facial geometries.
Figure 20. **Qualitative comparison of model fitting on SIZER.** We visualize the fitting results of baselines and our method. Our results are close to the ground truth.
Figure 21. **Avatar reposing.** *Top:* Given edited avatars, our method consistently applies local details stored in the feature codebooks onto the body surfaces in different poses. *Bottom:* Poses with self-contacts might cause artifacts to both texture and geometry.
<table>
<thead>
<tr>
<th>Input Avatar</th>
<th>Upper Body</th>
<th>Lower Body</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image9.png" alt="Image" /></td>
<td><img src="image10.png" alt="Image" /></td>
<td><img src="image11.png" alt="Image" /></td>
<td><img src="image12.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Figure 22. More cross-subject feature editing results. We partially transfer local clothing details from the unseen scans (upper and lower body) to the input avatars. The results of the edited avatars are shown in the right column.
References

[1] https://renderpeople.com/. 1, 2


