Modeling Clothing as a Separate Layer for an Animatable Human Avatar

DONGLAI XIANG, Carnegie Mellon University, USA and Facebook Reality Labs Research, USA
FABIAN PRADA, Facebook Reality Labs Research, USA
TIMUR BAGAUTDINOV, Facebook Reality Labs Research, USA
WEIPENG XU, Facebook Reality Labs Research, USA
YUAN DONG, Facebook Reality Labs Research, USA
HE WEN, Facebook Reality Labs Research, USA
JESSICA HODGINS, Carnegie Mellon University, USA and Facebook AI Research, USA
CHENGLEI WU, Facebook Reality Labs Research, USA

We have recently seen great progress in building photorealistic animatable full-body codec avatars, but generating high-fidelity animation of clothing is still difficult. To address these difficulties, we propose a method to build an animatable clothed body avatar with an explicit representation of the clothing on the upper body from multi-view captured videos. We use a two-layer mesh representation to register each 3D scan separately with the body and clothing templates. In order to improve the photometric correspondence across different frames, texture alignment is then performed through inverse rendering of the clothing geometry and texture predicted by a variational autoencoder. We then train a new two-layer codec avatar with separate modeling of the upper clothing and the inner body layer. To learn the interaction between the body dynamics and clothing states, we use a temporal convolution network to predict the clothing latent code based on a sequence of input skeletal poses. We show photorealistic animation output for three different actors, and demonstrate the advantage of our clothed-body avatars over the single-layer avatars used in previous work. We also show the benefit of an explicit clothing model that allows the clothing texture to be edited in the animation output.

CCS Concepts:
• Computing methodologies → Image-based rendering
• Animation

Additional Key Words and Phrases: clothing animation, codec avatar

ACM Reference Format:
Donglai Xiang, Fabian Prada, Timur Bagautdinov, Weipeng Xu, Yuan Dong, He Wen, Jessica Hodgins, and Chenglei Wu. 2021. Modeling Clothing as a

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
© 2021 Copyright held by the owner/author(s).
0730-0301/2021/12-ART1
https://doi.org/10.1145/3478513.3480545

Separate Layer for an Animatable Human Avatar. ACM Trans. Graph. 40, 6, Article 1 (December 2021), 15 pages. https://doi.org/10.1145/3478513.3480545

1 INTRODUCTION

Animatable photorealistic digital humans are a key capability for enabling social telepresence, and have the potential to open up a new way for people to remain connected without geographic constraints. Early work on human body modeling built low-dimensional geometric representations of the body surface with minimal clothing [Loper et al. 2015; Osman et al. 2020; Romero et al. 2017]. As a separate field of work, cloth simulation has been studied and used to create clothing deformation that does not conform tightly to the human body [Baraff and Witkin 1998; Buffet et al. 2019; Kavan et al. 2011; Narain et al. 2012]. However, both these lines of work focus on modeling only the geometry, and cannot directly produce photorealistic rendering output. Even with the recent data-driven methods using neural networks (for example [Lahner et al. 2018]), animating a photorealistic clothed human is still far from a solved problem.

In this work, we seek to build photorealistic full-body clothed avatars that can be animated with driving signals that can be easily accessed, for example, 3D body pose and facial keypoints. Simultaneously modeling both geometry and texture with a deep generative model, like Variational Autoencoders (VAE), has been demonstrated to be an effective way to create photorealistic face avatars [Lombardi et al. 2018]. Recently, Bagautdinov and colleagues [Bagautdinov et al. 2021] extend this approach to model full-body avatars with VAE, conditioned on body pose and facial keypoints. Because these conditional signals cannot uniquely describe the states for the clothing, hair and gaze, the VAE latent code is used to distinguish between these different states. In addition, it is essential to disentangle the effects of driving signals and the latent code, in order to reduce the spurious correlations between them.

Despite the progress in previous work [Bagautdinov et al. 2021], challenges still remain in building high-fidelity animatable full-body avatars, and we identify the modeling of clothing as one major difficulty. Artifacts include the imperfect correlation between body pose and clothing state, ghosting effects along the boundary between clothing and skin, as well as loss of wrinkle details and dynamics in the clothing. These artifacts become more noticeable when the captured clothing is loose and the performer moves more dynamically. On the one hand, due to registration error, the network may underfit the data, making it unable to reproduce high-frequency clothing detail; on the other hand, in spite of the disentanglement, the network may still overfit, capturing unwanted chance correlation between the driving signal and the clothing state.

In this work, we explicitly represent the body and clothing as separate layers of meshes in a codec avatar. The separation leads to several benefits. First, it allows us to accurately register both body and clothing, especially with our newly developed photometric tracking approach that uses inverse rendering to align clothing texture to a reference. Second, modeling the body and clothing in separate layers alleviates the aforementioned problem of chance correlation for a single-layer avatar, as the separate layers are naturally disentangled from each other. With our two-layer VAE, a single frame of joint angles can well describe the body state, while the clothing dynamics can be inferred from the sequences of poses with a Temporal Convolutional Network (TCN), which evolves the clothing state in a way that is consistent with the body motion. Third, thanks to the explicit modeling of clothing, the animation output can be further edited by changing the clothing texture.

To summarize, our contributions are as follows:

- We present an animatable two-layer codec avatar model for photorealistic full-body telepresence; our proposed avatar can produce more temporally coherent animation with sharper boundaries and fewer ghosting artifacts compared to a single-layer avatar;
- Inverse rendering with our proposed two-layer codec avatar allows a photometric tracking algorithm that aligns the salient clothing texture, significantly improving correspondence in the registered clothing meshes;
- We demonstrate an application of our two-layer codec avatar for editing of the clothing texture that is hard to achieve with the single-layer model used in previous work.

We evaluate the proposed pipeline on the captured sequences of three different actors. We demonstrate the effectiveness of our proposed method against alternative approaches. We show that our model, with only a sequence of poses and facial keypoints as input, achieves high-quality body animation and rendering with photorealistic clothing that can be viewed from arbitrary viewpoints.

2 RELATED WORK

Our goal in this paper is to build a realistic virtual avatar of a human that can be animated by driving signals of skeletal poses and facial keypoints to create a telepresence experience. The classical pipeline for modeling such an animatable avatar typically relies on building a textured template mesh from a 3D scan and rigging the template mesh to a parameterized skeleton model such that the deformation of the template mesh is associated with the skeletal pose according to the skinning weights. The most commonly used skinning method is the Linear BlendSkinning (LBS), which we also use to model the skeletal motion. In the literature, many methods have been developed in order to reduce the unnatural skinning artifacts that occur with LBS, e.g., [Kavan et al. 2008; Kavan and Zara 2005]. However, a fundamental disadvantage of these approaches is that high-frequency deformations of skin and clothing, such as muscle bulging, folds, and wrinkles, cannot be precisely modeled. In order to solve this problem, pose dependent blend shapes [Lewis et al. 2000] have been proposed to reduce skinning artifacts. These blend shapes are corrective shapes that can be interpolated with respect to the pose and added to the skinned mesh. Although blend shapes work well for skin and tight clothing, the non-rigid deformation of soft tissue and loose clothing is not modeled well by this approach.

Physical simulation provides an automatic way to create secondary motion of virtual characters, such as muscle bulging and cloth deformation. Cloth simulation is typically not real-time due to the computational complexity and therefore many of the earlier methods focus on efficiency [Gillette et al. 2015; Goldenthal et al. 2007; Kavan et al. 2011; Kim et al. 2013; Wang et al. 2010]. More recent research tackles efficiency by learning the mapping from body pose and shape to the clothing deformation produced by
physical simulations [Bertiche et al. 2020a,b; Chentanez et al. 2020; Gundogdu et al. 2019; Jin et al. 2020; Patel et al. 2020; Santesteban et al. 2019; Vidaurre et al. 2020; Wang et al. 2019; Zhang et al. 2021]. Among those methods, one notable concurrent work [Santesteban et al. 2021] adopts a similar strategy to model clothing with a VAE and animates clothing with a temporal model. Compared with our work, this approach focuses on avoiding collision in the clothing output, but does not model clothing from real-world captured data, or produce a photo-realistic rendering of the clothing. Cloth simulation has been leveraged in human performance capture to produce more realistic dynamic deformation of the clothing. Stoll and colleagues reconstruct a time-varying surface geometry of the clothing from multiview video recordings and then estimate the parameters of a physical simulation model of the clothing [Stoll et al. 2010]. SimuCap contributes a monocular human performance capture system that not only captures the skeleton motion but also simulates cloth dynamics and cloth-body interactions [Yu et al. 2019].

**Data-driven human modeling** has been leveraged very effectively in recent years. The seminal work, SCAPE [Anguelov et al. 2005], learns a parametrized human body shape model from a large-scale dataset of 3D scans. A variation of SCAPE that integrates the learned pose dependent blend shapes, SMPL [Loper et al. 2015], has been widely used for human modeling and pose estimation. However, these models can only model a human body dressed in skin tight clothing. In order to synthesize the deformation of clothing, apart from the aforementioned simulation-based learning approaches, many methods resort to learning the deformation from real 4D capture data. DeepWrinkle [Lahner et al. 2018] consists of two modules that learn the global cloth deformation in a PCA subspace as well as high frequency details, such as finer wrinkles, on a normal texture. Similarly, Ma and colleagues learn a pose-dependent clothing shape from 4D scans with different geometric representation, including mesh-based graph convolution [Ma et al. 2020], surface elements [Ma et al. 2021] and implicit functions [Saito et al. 2021]. Compared with our work, these methods mostly focus on modeling the clothing geometry, with less effort on creating photo-realistic rendering of clothing appearance.

Another family of generative human modeling methods does not focus on the 3D geometry, but aims to synthesize photo-realistic human images. These neural rendering approaches typically formulate the task as an image translation problem, and learn the mapping from joint heatmaps [Aberman et al. 2019], rendered skeleton [Chan et al. 2019; Esser et al. 2018; Pumarola et al. 2018; Si et al. 2018], or rendered meshes [Liu et al. 2019c,b; Prokudin et al. 2021; Raj et al. 2021; Sarkar et al. 2020; Wang et al. 2018], to real images. In contrast to these approaches, Deep Appearance Models [Lombardi et al. 2018] explicitly handle both facial geometry and appearance in the form of view-dependent texture, and is capable of producing view-dependent effects and correcting geometric artifacts. In recent work, Bagautdinov and colleagues extend deep appearance models to full bodies [Bagautdinov et al. 2021]. However, as this method does not explicitly model clothing, it may struggle in settings where clothing is loose or exhibits significant dynamics. Most related to our paper is the concurrent work of Habermann and colleagues [Habermann et al. 2021]. This work addresses a similar problem of creating a dynamic free-view point rendering of a specific subject given skeleton motion as input. It uses a neural network to regress the clothed body shape represented by an embedded graph plus additional deformation and a dynamic texture. Compared with this work, our method uses a two-layer formulation for both registration and modeling that enables high-quality animation output.

**Dynamic scene capture** is an alternative yet less practical approach for telepresence, because it does not compress the dynamic information of the scene as a latent code like our approach, and therefore requires a much higher communication bandwidth. That said, our method is still highly related to these methods, as we rely on dynamic scene capture to obtain training data. Most of the existing approaches rely on multi-camera systems to recover detailed geometry using silhouettes or photometric stereo. They reconstruct either the shapes of each individual time step [Matusik et al. 2000; Starck and Hilton 2007; Waschkus et al. 2005], or a temporally coherent shape by deforming a template to match the multi-view constraints [Carranza et al. 2003; de Aguiar et al. 2008]. While some of the methods work for general scenes, many of them are dedicated to human bodies [Bray et al. 2006; Brox et al. 2010; Gall et al. 2009; Liu et al. 2011; Mustafa et al. 2015; Vlasic et al. 2008; Wu et al. 2013, 2012]. In recent years, many attempts have been made to alleviate the requirement of multi-camera systems by using depth sensors [Bogo et al. 2015; Guo et al. 2015; Helten et al. 2013; Li et al. 2009; Zhang et al. 2014] or even a monocular RGB camera [Habermann et al. 2019, 2020; Huang et al. 2017; Xu et al. 2018]. Although compelling results have been demonstrated, these approaches are fundamentally ill-posed and suffer from occlusion and depth ambiguities. Furthermore, in contrast to our method, they typically treat the character as a topologically connected template, and therefore are not able to handle movement of the clothing, such as sliding of the sleeves on the arms. Another line of work specifically focuses on capturing clothing deformations [Bradley et al. 2008; Chen et al. 2015; Pons-Moll et al. 2017; Xiang et al. 2020; Zhou et al. 2013]. For instance, ClothCap [Pons-Moll et al. 2017] automatically segments the different pieces of clothing and tracks the deformation of the clothing over time from 4D scans. Zhang and colleagues recover the detailed body shape under the clothing [Zhang et al. 2017]. Our approach relies on these two methods for the generation of training data. More recently, multiple approaches have been proposed to capture human appearance by modeling the radiance field with a deep neural network [Park et al. 2020; Peng et al. 2021; Pumarola et al. 2021; Wang et al. 2021]. These methods can synthesize photo-realistic novel views of the captured scene or human subject, but unlike our work, cannot be used as animatable virtual avatars.

3 METHOD OVERVIEW

Our goal in this paper is to build full-body clothed digital avatars that enable photorealistic rendering from any viewpoint. To make the avatars useful, they should be animatable given some driving...
signals that can be obtained at modest cost. We choose 3D skeletal joint angles and facial keypoints as the input conditioning, similar to previous work [Bagautdinov et al. 2021]. For example, these driving signals can be obtained by multi-view triangulation and inverse kinematics from a sparse set of cameras.

The central idea of our method is to explicitly represent body and clothing as two separate layers. We take this approach for three reasons. First, we notice that the deformation of the body and the clothing follow different movement patterns because of their different dynamics. A single frame of joint angles in the driving signal can largely determine the body state through Linear Blending Skinning (LBS) and pose-dependent deformation. In contrast, the dynamics of clothing can vary too much to be described only by current body pose without considering temporal information. Thus the body and clothing layers need to be controlled by different input conditioning. Second, in the single-layer registration of the body with the clothing, a specific vertex along the clothing boundary can belong to either the body region or the clothing region in different frames due to the sliding motion of the clothing relative to the body, which violates the single layer assumption. A codec avatar trained with such data often has a color between the clothing and skin colors in such a region, leading to ghosting effects around the sleeves and neck of the garment. Although disentanglement could alleviate this kind of artifact, it cannot eliminate it due to limited training data capturing the complex interaction between clothing and the body. In our work, with the registration of body and clothing in separate layers, such artifacts can be avoided because each vertex is either part of body or the clothing across all frames. Third, separate layers for body and clothing open up opportunities for further changing the appearance of the avatar, such as temporally consistent editing of the clothing texture without interfering with the body appearance. This capability might also make it possible to alter the clothing style through physical simulation, which we leave for future work.

In this work, we assume that the subject to be modeled wears a T-shirt and pants. We only model the T-shirt in the second, outer layer because it exhibits most of the dynamics and variations in geometry and texture. In the inner layer, we model the body region covered by the outer layer (torso and upper arms) and the rest of human surface, including the head, arms, pants and shoes.

In Section 4, we briefly describe our two-layer geometry-based surface registration method to generate the necessary training data for the codec avatars. In Section 5, we present our two-layer codec avatars. We describe the architecture of the body branch in Section 5.1 and clothing branch in Section 5.2, as well as the joint training of both branches through inverse rendering in Section 5.3. In Section 5.4, we propose a method for texture alignment to improve the photometric correspondences between registered clothing meshes across different frames. In Section 6, we present the temporal model used to animate our clothed avatars using a sequence of joint angles as the driving signal. A visualization of the method is shown in Fig. 2.

4 CLOTHED BODY REGISTRATION

The pipeline to generate the data for training our two-layer codec avatars is illustrated in Fig. 3. Our goal is to register the body and clothing geometry in two separate layers. A more detailed description of this pipeline can be found in the supplementary document.

Data preprocessing. The input to our pipeline is a sequence of RGB images of the subject captured by a synchronized multi-camera system. The raw RGB images are used to create a dense 3D reconstruction of the human surface with a multi-view Patchmatch reconstruction algorithm [Galliani et al. 2015]. An example of the reconstructed mesh can be seen in Fig. 3. In addition, we obtain a

---

1 The pants of the captured subjects in this work are tight and thus not worth the effort of modeling as a separate layer. We demonstrate in the results that the advantage of clothing modeling as a separate layer is obvious when the garment is loose.
We now present our two-layer codec avatars with explicit clothing of the inner layer in this work. Unlike [Zhang et al. 2017], we only need to estimate the underlying direct obtained by matching with the single-layer tracking results. In this step, we segment the single-layer tracked meshes into separate body and clothing parts. We unproject the image segmentation labels onto the mesh and for each vertex take the majority of votes across different views. Similar to [Pons-Moll et al. 2017], we also run the Markov Random Field (MRF) to remove noisy segmentation labels.

Clothing Registration. Our clothing registration step is similar to [Pons-Moll et al. 2017]. We manually create a template clothing mesh and use it to register the clothing region of the single-layer tracked mesh for each frame. Essentially we run a non-rigid ICP algorithm that aligns the template and target clothing region. To provide good initialization for the optimization, we find it useful to apply Biharmonic Fecomation Fields [Jacobson et al. 2010] which provide good initialization. As shown on the left of Fig. 4, our body network is similar to the decoder structure in [Bagautdinov et al. 2021], without the encoder. Once the clothing is decoupled from the body, the skeletal pose and facial keypoints contain sufficient information to describe the body state (including pants that are relatively tight). We do not use a latent code as conditioning for the body network to avoid the difficult problem of disentanglement between the latent space and the driving signal, as described in [Bagautdinov et al. 2021]. Our body decoder takes in the skeletal pose, facial keypoints and view-conditioning as input, produces unposed geometry in a UV positional map and view-conditioned texture for the body as output. A LBS transformation is then applied to the unposed mesh restored from the UV map to produce the final output mesh.

The loss function to train the body network is defined as:

\[ \mathcal{L}_{train} = \lambda_0 \| \mathbf{V}_B^p - \mathbf{V}_B^c \|^2 + \lambda_{lap} \| \mathcal{L}(\mathbf{V}_B^c) - \mathcal{L}(\mathbf{V}_B^p) \|^2 \\
+ \lambda_f \| (\mathbf{T}_B^c - \mathbf{T}_B^p) \odot M_B^p \|_2^2, \tag{1} \]

where \( \mathbf{V}_B^p \) is the vertex position interpolated from the predicted position map in UV, and \( \mathbf{V}_B^c \) is the vertex from inner layer registration from Sec. 4. \( \mathcal{L}(\cdot) \) is the Laplacian operator, \( \mathbf{T}_B^c \) is the predicted

5 CLOTHED BODY MODELING
We now present our two-layer codec avatars with explicit clothing modeling. Similar to [Lombardi et al. 2018] and [Bagautdinov et al. 2021], we employ a Variational Autoencoder (VAE) as our generative model. In our two-layer formation, we train a separate network to learn the deformation space for body and clothing, while the correlation between body and clothing can be learned afterwards with a temporal model for animation. To this end, we train a body decoder which takes the skeletal pose as input, and predicts geometry and view-conditioned texture for the inner body layer, as shown on the right of Fig. 4. Similarly, we train a clothing decoder with a VAE, as shown on the right of Fig. 4. Similar to existing approaches to body modeling [Loper et al. 2015; Osman et al. 2020], we only learn the geometry in the canonical pose space for both the body layer and the clothing layer by applying an inverse LBS transform. This technique reduces the deformation space that needs to be learned. In the following sections, we introduce the detailed structure for the body and clothing networks, and explain how we train them. Implementation details including loss weights and network architecture can be found in the supplementary document.

5.1 Body Decoder
As shown on the left of Fig. 4, our body network is similar to the decoder structure in [Bagautdinov et al. 2021], without the encoder. The inner-layer geometry consists of two parts: the invisible body region covered by the clothing in the upper body, which we estimate using the method in [Zhang et al. 2017], and the visible region of the human surface, which can be directly obtained by matching with the single-layer tracking results. Unlike [Zhang et al. 2017], we only need to estimate the underlying body shape of the upper body, because the pants are treated as part of the inner layer in this work.

![Network architecture of our two-layer full-body codec avatar. We show the body network on the left and the clothing network on the right, including the input and output of each network.](image)
texture, $T_B^i$ is the reconstructed texture per-view, and $M_B^V$ is the mask indicating the valid UV region.

5.2 Clothing Network
As shown on the right of Fig. 4, we model the clothing appearance with a Conditional Variational Autoencoder (cVAE). The encoder takes as input the unposed clothing geometry and mean-view texture, and produces parameters of a Gaussian distribution, from which a latent code $z$ is sampled. Besides the latent code, the decoder also takes spatial-varying view conditioning as input, and predicts geometry and texture for the clothing. Then, the training loss is described as:

$$E_{\text{train}}^C = \lambda_g \|V_C^r - V_C^i\|^2 + \lambda_{\text{lap}} \|L(V_C^r) - L(V_C^i)\|^2 + \lambda_t \|T_C^r - T_C^i\|_2^2 + \lambda_{E_k}\|\mathcal{C}_V - \mathcal{M}_C^r\|_2^2 + \lambda_{E_k}E_{\text{kl}},$$

where $V_C^r$, $V_C^i$, $T_C^r$, $T_C^i$, and $M_C^r$ are all defined similarly to the parameters in the body decoder but with respect to clothing, $E_{\text{kl}}$ is a conventional KL divergence loss.

5.3 Inverse Rendering with Two-layer Representation
The ICP-based clothing registration algorithm in Section 4 and previous work [Pons-Moll et al. 2017] aims to align the boundary of the clothing template with the target area, while there is no explicit constraint for the interior correspondences except for the mesh regularization. Therefore, the registered meshes from Sec. 4 may suffer from correspondence errors in the interior (see the first column of Fig. 8), which significantly influences the decoder quality, especially for dynamic clothing. In order to correct the correspondences in the training stage, we need to link the predicted geometry and texture to the input multi-view images in a differentiable way. To this end, after the body and clothing networks are separately trained as described in Sec. 5.1 and 5.2, we jointly train the body and clothing networks by rendering the output with a differentiable renderer. We use the following loss functions:

$$E_{\text{inv}}^C = \lambda_g \|F_C^r - F_C^i\|_2^2 + \lambda_m \|M_C^r - M_C^i\|_2^2 + \lambda_{E_{\text{softvisi}}} + \lambda_{E_{\text{lap}}},$$

where $F_C^r$ and $F_C^i$ are the rendered image and the captured image, $M_C^r$ and $M_C^i$ are the rendered foreground mask and the captured foreground mask, and $E_{\text{lap}}$ is the Laplacian geometry loss similar to that defined in Eqn. 1 and 2. $E_{\text{softvisi}}$ is a soft visibility loss, similar to [Liu et al. 2019a], that is specifically designed to handle the depth reasoning between the body and clothing so that the gradient can be back-propagated through if the depth order is wrong. In detail, we define the soft visibility for a specific pixel as:

$$S = \sigma\left(\frac{D_C - D_B}{c}\right),$$

where $\sigma(\cdot)$ is the sigmoid function, $D_C$ and $D_B$ are the depth rendered from the current viewpoint for the clothing and body layer, and $c$ is a scaling constant. Then the soft visibility loss is defined as:

$$E_{\text{softvisi}} = S^2,$$

when $S > 0.5$ and the current pixel is assigned to be clothing according to the 2D cloth segmentation. Otherwise, $E_{\text{softvisi}}$ is set to 0. If the pixel is labeled as clothing but the body layer is on top of the clothing layer from this viewpoint, the soft visibility loss will back-propagate the information to update the surfaces until the correct depth order is achieved.

Following [Bagautdinov et al. 2021] in this inverse rendering stage, we also use a shadow network that computes quasi-shadow maps for body and clothing given the ambient occlusion maps. In contrast to the approach of [Bagautdinov et al. 2021] where the ambient occlusion is approximated with the body template after the LBS transformation, we compute the exact ambient occlusion using the output geometry from the body and clothing decoders because we aim to model a more detailed clothing deformation than can be produced by the LBS transformation. The quasi-shadow map is then multiplied with the view-dependent texture before applying the differentiable renderer.

5.4 Texture Alignment with Inverse Rendering
The inverse rendering method mentioned in Sec. 5.3 already has the capability to improve photometric correspondences to some extent, because the network tends to predict texture with less variance across frames, along with deformed geometry to align the rendering output with the ground truth images. Ideally, we only need to train the two decoders simultaneously with the inverse rendering loss to correct the correspondences while creating the generative model for driving the animation. However, we find that this alone would not correct all the correspondence errors. The model might not find a good minimum for two reasons. First, the variation in photometric correspondences in our initial registration may be too large for the network to fix. Secondly, our VAE model with view conditioning may allow the decoder to cheat with the view-dependent texture rather than moving the geometry.
These problems motivate us to propose a new way to use inverse rendering for correspondence improvement. First, we separate the registered meshes into chunks of 50 neighboring frames. Then, we select the first chunk as the anchor frames, and train an anchor network for this chunk using the inverse rendering model described in Sec. 5.3. After convergence, we use the trained network parameters to initialize the training of other chunks. To make sure that the alignment of the other chunks does not drift from the anchor frames, we set a small learning rate (1e-4 for the AdamW optimizer), and mix the anchor frames with each other chunk during training. We remove the view conditioning from the texture branch of our decoder in Sec. 5.3, and use a single texture prediction for inverse rendering in all the camera views. The output geometry predicted by the network of each chunk after training has more consistent correspondences across frames compared with the input, which is manifested by the consistent projected texture pattern in the UV space shown in Fig. 8. A visual illustration of this process is provided in Fig. 5. This method has a similar spirit to previous UV-template-based texture alignment approaches [Bogo et al. 2017; Garrido et al. 2013], but naturally extends the idea to a neural-network formulation under the framework of codec avatars.

The method described here is applied after the two-layer registration is obtained in Section 4, as shown in Fig. 2. For each frame, we use the output geometry predicted by the network as a new registered mesh with the improved correspondences. We use these data to train the body and the clothing networks, as described in Section 5.1-5.3.

### 6 TEMPORAL MODELING FOR POSE-DRIVEN CLOTHING ANIMATION

In our two-layer codec avatars, the body output is conditioned on a single frame of skeletal pose and facial keypoints, while the clothing state is determined by the latent code. In order to animate the clothing from the driving signal, we use a Temporal Convolution Network (TCN) to learn the correlation between body dynamics and clothing deformation. Our TCN takes in the sequence of previous and current skeletal pose and infers the latent clothing state.

An illustration of our animation pipeline is shown in Fig. 6. The temporal convolution network takes as input the joint angles in a window of $L$ frames up to the target frame, and passes through several 1D temporal convolution layers to predict the clothing latent code for the current frame $z$. To train the TCN, we minimize the following loss function:

$$E_{\text{train}} = ||z - z'|^2,$$

where $z'$ is the ground truth latent code obtained from the trained clothing VAE.

An alternative formulation would be to condition the prediction on not just previous body states, but also previous clothing states. This formulation is inspired by cloth simulation, where the clothing vertex position and velocity in the previous frame are needed to compute the current clothing state. However, in our data-driven setting, we find that such an auto-regressive model that takes in previous clothing states is hard to train and does not outperform the non-auto-regressive model given the limited amount of data (25 min). Therefore, the input to our TCN is a temporal window of skeletal poses, not including the previous clothing states.

**Resolving Intersection.** One solution is to add a training loss for TCN to make sure that the predicted clothing does not intersect with the body. However, even without a loss to penalize intersection, the clothing states predicted by our TCN model already match the body shape well, resulting in only minimal intersection. Thus we only need to resolve intersection as a post processing step. We project the intersecting clothing back onto the body surface with an additional margin in the body normal direction. This operation will solve most intersections and make sure that the clothing and body are in the right depth order for rendering. An example of these results can be seen in Fig. 7.

### 7 RESULTS

In this section, we first introduce our capture system and captured data. Then we show the results of our photometric texture alignment method to demonstrate its effectiveness in achieving better photometric correspondence in the UV space. After that, we show...
the animation output of our two-layer codec avatars with explicit clothing modeling. In particular, we demonstrate the advantage of our two-layer formulation against the single-layer model in previous work. We close by demonstrating clothing texture editing for animation.

### 7.1 Data Capture

The training data for our codec avatars are captured by a multi-view capture system consisting of around 140 cameras that are distributed uniformly on a half dome above the ground. All the cameras run with hardware synchronization, capturing at the resolution of $4096 \times 2668$ and 30 fps. Three identities, one female (Subject 1) and two males (Subject 2 and Subject 3), are captured with a pre-defined acting script. The script is designed to capture peak poses with the activation going through all body joints, followed by a 10-minute conversation to capture social behavior. For each subject, we collect sequences of 40k-50k frames in total and intentionally leave out approximately 4-5k contiguous frames for testing.

### 7.2 Texture Alignment with Inverse Rendering

In this section, we show the results of texture alignment based on inverse rendering (Section 5.4) on the sequence of Subject 2. Textures are projected from the raw captured images to the registered meshes before and after the texture alignment procedure, and then unwrapped into the UV space for comparison. Example results for several frames are shown in the first and third column of Fig. 8. To assess the quality of alignment, we compare the mean UV texture of the anchor frames with the unwrapped texture of each individual.

---

Table: Texture Alignment Results

<table>
<thead>
<tr>
<th>Before texture alignment</th>
<th>Error before texture alignment</th>
<th>After texture alignment</th>
<th>Error after texture alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 8. Inverse-rendering-based texture alignment results. From left to right, we show (1) projected texture on the clothing mesh before texture alignment, (2) error map between the first column and the mean texture of anchor frames, (3) projected clothing texture after texture alignment, and (4) the difference between the third column and the mean texture of anchor frames. The error maps are visualized with the Jet colormap; lighter color represents larger error. We also show a zoomed-in version of the text region to highlight the difference.
Modeling Clothing as a Separate Layer for an Animatable Human Avatar

Before texture alignment

After texture alignment

Mean

Standard deviation

Fig. 9. Mean (top row) and standard deviation (bottom row, converted to jet colormap) of unwrapped texture before (left column) and after (right column) texture alignment on the sequence of Subject 2.

The error map is then visualized by the Jet colormap, shown in the second and fourth column of Fig. 8 respectively.

The visible pattern in the heatmap before texture alignment (the second column) verifies the lack of accurate interior correspondences in the registered clothing meshes from the ICP algorithm (Section 4). After the texture alignment (the fourth column), the error between the UV texture of those frames and the mean of anchor frames is significantly reduced. This result suggests that the correspondences in the mesh interior are improved in the inverse rendering process, and demonstrates the effectiveness of our texture alignment method.

To statistically evaluate the quality of photometric correspondence in the UV space, we compute the mean and standard deviation of the unwrapped texture across different frames, as visualized in Fig. 9. Comparing the mean texture images, we observe a much sharper text pattern after texture alignment than before. Similarly, the standard deviation after texture alignment becomes smaller and more concentrated in the spatial domain. This result also verifies the improvement of photometric correspondence thanks to our texture alignment approach.

7.3 Pose-Driven Animation

In this section, we present animation results produced by our two-layer codec avatars driven by the 3D skeletal pose and facial keypoints. In our animation pipeline, the body decoder is directly driven by skeletal pose and facial keypoints of the current frame; on the other hand, the clothing decoder is driven by latent clothing code generated by the temporal clothing model in Section 6, which takes a temporal window of history and current poses as input. We compare the quality of our animation with previous work [Bagautdinov et al. 2021] that uses a single-layer codec avatar. We follow the method described in [Bagautdinov et al. 2021] to animate the single-layer codec avatar: we randomly sample the unit Gaussian distribution, and use the resulting noise values for imputation of the latent code. The sampled latent code, the skeletal pose and facial keypoints are fed together into decoder network. We present qualitative animation...
results on all three testing sequences, shown in Fig. 10, 11, and 12. Our animation results are better seen in the supplementary video. Our two-layer formulation helps remove the severe artifacts in the clothing regions in the animation output of [Bagautdinov et al. 2021], especially around the clothing boundary of Fig. 10, and 12. Indeed, as the body and clothing are modeled together, the single-layer avatars rely on the latent code to describe the many possible clothing states corresponding to the same body pose. During animation, however, the absence of a ground truth latent code leads to degradation of the output, despite the efforts in [Bagautdinov et al. 2021] to disentangle the latent space from the driving signal. In contrast, our animation model achieves better animation quality by separating body and clothing into different modules: the body decoder can determine the body states given the driving signal of the current frame; the temporal model learns to infer the most plausible
Modeling Clothing as a Separate Layer for an Animatable Human Avatar

We also quantitatively compare the animation output of our two-layer codec avatars with the baseline method [Bagautdinov et al. 2021] by evaluating the output images against the captured ground truth images. We report the evaluation metrics of Mean Square Error (MSE) and Structural Similarity Index Measure (SSIM) over the foreground pixels. The results are shown in Tab. 1. Our method consistently outperforms [Bagautdinov et al. 2021] on all three sequences and both evaluation metrics. In particular, it is worth noting that our advantage on MSE is most obvious on the sequence of Subject 3, who is wearing a loose T-shirt that is hard to model with the single-layer avatar. This result agrees with our qualitative observation of the images as well.

7.4 Ablation Analysis

In this section, we present an ablation analysis on several different components in the design choice of our system. The results are shown in Fig. 13.

First, we analyze our design of VAE (Sec. 5.2) + temporal modeling (Sec. 6) for clothing animation. One alternative for this design is to combine the functionality of these two networks into one: to train a decoder that takes a sequence of skeleton poses as input and predicts clothing geometry and texture as output. The result of this comparison is shown on the left of Fig. 13. Here, the baseline model produces blurry output around the logo on the T-shirt. Even a sequence of skeleton poses does not contain enough information to fully determine the clothing state. Therefore, similar to the analysis in [Bagautdinov et al. 2021], directly training a regressor from the information-deficient input to final clothing output leads to under-fitting to the data by the model. In contrast, in our proposed system, the VAE network can model different clothing states in detail with a generative latent space, while the temporal modeling network infers the most probable clothing state. In this way, our method can produce high-quality animation output with sharp detail.
Next, we demonstrate the influence of photometric texture alignment (Sec. 5.4) on the final animation output. We compare the results generated by our full model, which is trained on registered body and clothing data with texture alignment, against a baseline model trained on data without texture alignment (output of Sec. 4). The result is shown in the middle of Fig. 13. We see that photometric texture alignment also helps to produce sharper detail in the animation output, as the better texture alignment makes the data easier for the network to model.

In addition, we also validate the ability of our network to generate view-dependent effects. We compare our full model with a baseline model where the body and clothing networks do not take view conditioning as input. The results are shown on the right of Fig. 13.
Our output with view-dependent effects is visually more similar to the ground truth than the baseline model without view conditioning. The most obvious difference is observed near the silhouette of the subject, where the view-dependent output is brighter due to Fresnel reflectance when the incidence angle gets close to 90° \cite{Lafortune et al. 1997}, an important factor that makes the view-dependent output more photo-realistic.

In the supplementary material, we also include a video comparison of animation results with different lengths $L$ of the temporal window as input to our TCN (Sec. 6). With a small temporal window (for example $L = 1, 3, 8$), the temporal model tends to produce output with jittering. We find $L = 15$ or $30$ achieves a good tradeoff between visual temporal consistency and model efficiency. For a more detailed analysis of this issue, please refer to the supplementary document.

### 7.5 Application: Clothing Texture Editing

In this section, we demonstrate editing for the clothing texture. On top of our photorealistic animation output, we further edit the clothing pattern in four different styles. First, we multiply the RGB channels of the clothing UV texture with different scaling factors to modify the color of the clothing. Second, we apply a checkerboard pattern on our clothing layer. Third, we ask an artist to create a stylistic pattern and then apply it to our clothing animation output. Fourth, we add the ACM SIGGRAPH Logo and text to the front side of the clothing. The results are shown in Fig. 14. Once the desired pattern is determined, our model can produce animation with the edited texture for any motion sequence similar to those shown in Sec. 7.3.

Compared with the single-layer model, our two-layer structure naturally allows us to easily manipulate the clothing texture in the UV space without interfering with the inner layer in a temporally coherent manner. For comparison, we apply the same blue color transformation to the single-layer output. For this purpose, we manually segment out the clothing region for the first frame in the sequence in the UV space, and apply the color transformation in the segmented region to all the following frames. This approach produces reasonable results for the first frame (shown on the first column of Fig. 15); for the following frames, however, applying the color transformation in the same UV region will suffer from misalignment of the edited area and actual clothing region, as shown in the right two columns of Fig. 15. The visual artifact caused by this misalignment is highlighted in the zoomed-in boxes in the figure.

### 8 DISCUSSION

We have proposed a two-layer mesh representation for building an animatable avatar for clothed body. Results have demonstrated that the explicit clothing modeling not only improves the rendered clothing quality in animation, but also enables the editability of the clothing texture, opening up new possibilities for codec avatars. The two-layer avatar models cannot be obtained without the success of two-layer registration of the clothed body. We thus have presented a new clothed body registration method along with a texture alignment method to improve the photometric correspondences using inverse rendering.

Our clothed body model is trained for each individual subject and also can only be animated for that individual. All the driving signals have been captured from the same subject performing social interactions. The animatable model may not be able to generalize to poses deviating significantly from the training pose distribution. Artifacts may appear if our model is used for arbitrary motion retargeting.

In this work, we are only focusing on T-shirts. To extend the work to lower body clothing, like short pants with the boundary shifting on the legs, we need to extend the current two-layer work to handle multiple layers, potentially with occlusion between layers, which poses additional challenges to both registration and modeling. Another common piece of clothing is a skirt, which could be even more difficult due to its large motion and deformation. We cannot handle topology-changing clothing, like opening a zipped jacket.

Even with the current two-layer framework, our clothing registration method would fail if the hands and clothing interact significantly, for example, hands dragging the clothing or hands put under the clothing. The current non-physical interaction modeling between clothing and body may not easily extend to handle these challenges. One possibility is to integrate more physical constraints into registration and learning for animation.

### REFERENCES


Timur Bagautdinov, Chenglei Wu, Tomas Simon, Fabian Prada, Takaaki Shiratori, Shih-En Wei, Weipeng Xu, Yaser Sheikh, and Jason Saragih. 2021. Driving-signal aware...


