A vital task of the wider digital human effort is the creation of realistic garments on digital avatars, both in the form of characteristic fold patterns and wrinkles in static frames as well as richness of garment dynamics under avatars’ motion. Existing workflow of modeling, simulation, and rendering closely replicates the physics behind real garments, but is tedious and requires repeating most of the workflow under changes to characters’ motion, camera angle, or garment resizing. Although data-driven solutions exist, they either focus on static scenarios or only handle dynamics of tight garments. We present a solution that, at test time, takes in body joint motion to directly produce realistic dynamic garment image sequences. Specifically, given the target joint motion sequence of an avatar, we propose dynamic neural garments to jointly simulate and render plausible dynamic garment appearance from an unseen viewpoint. Technically, our solution generates a coarse garment proxy sequence, learns deep dynamic features attached to this template, and neurally renders the features to produce appearance changes such as folds, wrinkles, and silhouettes. We demonstrate generalization behavior to both unseen motion and unseen camera views. Further, our network can be fine-tuned to adopt to new body shape and/or background images. We also provide comparisons against existing neural rendering and image sequence translation approaches, and report clear quantitative improvements.

1 INTRODUCTION

A great dress can make you remember what is beautiful about life.

Rachel Roy

In the real world, we observe a wide range of garments on humans. Such garments display characteristic details (e.g., crease and folds) arising from a variety of factors, including their stitching layout, underlying materials, or printed patterns on the base fabrics. Naturally, there is a strong motivation, both for games and VR applications, to similarly ‘clad’ virtual avatars in realistic garments.

A popular workflow, enabled by commercial software (e.g., Marvelous Designer), is to mimic the real world in every stage: geometrically model a garment mesh using real-world stitching patterns for guidance, anticipate mesh dynamics using a physically-based simulator under the action of the body movements, and finally render the resultant dynamic mesh using available texture information. Achieving realistic results using this workflow is tedious and expensive due to the high level of precision required across the modeling, simulation, and rendering stages. Further, much of the process has to be repeated when any of the garment details, body motion, or viewing camera location need to be updated. A modified workflow encourages using a coarse garment model that is physically-simulated and high-resolution details added in a postprocessing stage using baked-in texture maps. While the approach is efficient and robust,
the skinned garments appear stiff as the baked-in details do not move realistically on loose garments.

Recently, data-driven approaches have been proposed to replace parts of the above process. For example, hallucinating geometric details on simulation of coarse garment templates [28, 78], extending parametric human body models with per-vertex displacements to capture the deformation of tight garments [3], or directly translating image-space body joint locations to final rendered images to recreate person-specific dance sequences [7]. Another intriguing alternative to produce realistic renderings is to train a network to produce detailed high-quality (rendered) images that can be directly supervised with multi-view image information [42, 64]. Such neural rendering approaches [47, 64], however, are mostly restricted to static objects and cannot handle intricate dynamics like the movement of garments under the action of the underlying body movements. An exception and particularly relevant to ours is the work of Chen et al. [7] that handles dynamic sequences with tight garments on actors. The method, however, does not generalize to produce dynamics of loose fabrics, as in our focus (see Section 5.3 for comparison).

In this work, we consider the problem of estimating detailed garment dynamics and generating realistic image sequences with plausible details driven by the given 3D (human) body motion information. A good solution should (i) produce realistic image-level garment details that are consistent across frames and camera views; (ii) produce plausible garment motion in response to any input body motion; (iii) be flexible enough to be trained using multiview target appearance supervision; (iv) independent of physically-based simulation at test time; and (v) generalize across motion and view variations.

To this end, starting from only a sequence of body joint information, we predict a coarse proxy template sequence over frames and learn dynamic deep features attached to this coarse template. The learned features capture both the geometric differences between the coarse and target garment (e.g., lace structure in Figure 2) and the dynamic appearance changes (e.g., appearance/disappearance of folds under body motion). The features are dynamic in two ways: first, the overall movement is captured by their geometric displacement across frames (i.e., coarse template motion), and second, the learned neural features are concatenated with body joint information before being interpolated and rendered. By projecting and rendering these features from input camera views, we train the whole process using image information while using intermediate coarse simulation as a proxy loss (i.e., our current realization is not end-to-end trained). Finally, in order to ensure temporal smoothness, we propose an encoder structure that explicitly models the correlation between the temporal latent codes as well as a temporal discriminator to ensure synthesized consecutive frames are temporally coherent.

We evaluate our algorithm on a range of examples with varying body motion and target garment styles. We demonstrate that our network generalizes over a range of viewpoints and body motion. Further, the approach handles changes to body sizes and illumination conditions (provided via a background image) with fine-tuning. We compare with strong baselines including image translation networks [7, 24] and deferred neural rendering [64] to demonstrate clear improvements. Compared to the strongest baseline, our method achieves an improvement of 70% and 46% in the image based [20] and video based [68] FID scores, respectively. Our is efficient running at 16 fps, while traditional physically-based simulation approaches can only achieve 0.5 fps on average for similar visual quality.

In summary, our key contributions are:

- **motion driven rendering** where the input 3D body motion sequence is used to generate realistic pixel-level frame renderings of complex target garments;
- **dynamic deferred neural rendering** based on an intermediate coarse proxy to synthesize varying motion appearance with the target colors and textures;
- **generalization** across views and motions to effectively avoid explicitly modeling, rendering, and simulating garment sequences on digital avatars for every new motion sequence.

2 RELATED WORK

Garment authoring. Starting from an initial, possibly coarse garment, several methods have been proposed for intuitive editing of the shape and appearance of the garment. Umetani et al. [66] propose an interactive system that enables to edit both 2D sewing patterns and the 3D draped garments. In a similar fashion, Bartle et al. [4] present a method to map 3D edits to plausible 2D garment patterns. Several methods focus on specific type of edits such as resizing [41] or stylizing tight-fitting garments [27]. The SecondSkin system proposed by De Paoli et al [11] focuses on modeling layered accessories and garments directly on a 3D character. More recently, Li et al. [29] has presented an intuitive system to add folds and pleats to a garment. All of these approaches work on a static garment, while we focus on generating dynamics of a given garment. Hence our work is complimentary, our system can be used to learn the dynamic appearance of a garment authored by the previous methods.

Modeling garment dynamics. Physically based simulation provides an accurate way to model garment dynamics [9, 31, 45, 46, 62, 76]. However, as the complexity of the garments increases, computational cost and stability issues become major sources of concern. Hence, several methods have been proposed to approximate the costly physically based simulation process. One line of work utilizes constraint-based optimization methods to compute a high resolution mesh from a coarse one [15, 44, 54]. Other popular approaches include utilizing data-driven methods to generate high resolution garment meshes by learning a mapping from a coarse garment mesh to fine-scale displacements [13, 81], utilizing learned upsampling operators [26], interpolating and blending example samples in the database [67, 73], or learning a subspace model of garment deformations [16, 19].

In recent years, researchers have explored the use of deep learning methods to learn how garments deform under body motion. In case of tight clothing (i.e., uv coordinates on body closely match uv coordinates of garments), representing garments as per-vertex displacements with respect to the naked body is a common and efficient practice [3, 5, 25, 37, 51]. However, this approach falls short in modeling loose garments. One approach to address this limitation is to utilize recent implicit based representations to recover animatable 3D characters given input images [23, 75]. However, such approaches do not model the garments separately and are limited in
terms of modeling the complex garment dynamics. Hence, several works assume the template garment to be known at rest pose and learn to construct the deformed garment shape under different body poses. While some methods utilize subspace techniques to operate on a reduced deformation space [21, 70, 74], others directly learn to predict the deformed garment shape [18, 50, 55]. Finally, some recent work augment a low-resolution of a garment normal map with plausible wrinkles [28, 78].

In our method, we use a similar learning based approach to predict the deformed shape of a coarse garment under body motion [71]. Unlike previous methods, however, our method adopts a neural rendering approach to synthesize garment deformations with rich details under different styles and viewpoints. The coarse template required by our method is not only a downsampling version of the desired garment but also is free of certain geometric details such as laces or multi-layer components which are hard to model and simulate. We show that starting from the same coarse garment deformation, we can plausibly synthesize different high resolution garment styles under various body motion (Section 5).

Neural rendering. We are recently witnessing an exciting breakthrough in the field of neural rendering where deep neural features are learned for controllable image synthesis including viewpoint change and modeling deformations (see [63] for a survey). Neural features have been learned on various representations including voxel grids [33, 42, 61], 3D meshes [64], point clouds [2], and multi-layer images [36]. Most of these approaches, however, focus on modeling viewpoint changes or illumination changes for static scenes or objects. Most recently, these methods have been extended to handle non-rigidly deforming [48, 65] and dynamic scenes [30, 52].

In the context of humans, neural rendering has been utilized to synthesize faces under different head pose and expression [35, 72], render humans under different viewpoint and illumination [40], and for reposing [56, 58]. Most of these approaches assume humans wear tight clothing and do not explicitly focus on modeling the dynamics of loose garments. Our work aims to fill this void by proposing a novel neural rendering pipeline to render plausible dynamics for loose garments that generalizes across motion changes and view variations.

Image-to-image translation. Image-to-image translation methods have shown incredible success recently in terms of translating images from a particular domain to another [14, 22, 24, 32]. These methods have also been extended to the temporal domain to synthesize videos from a guiding signal such as segmentation masks [39, 68]. Inspired by the success of such methods, several works have explored these translation methods to animate various objects.

Many video-based translation methods utilize an intermediate representation based on keypoints [43, 59, 60] and transfer the motion of a source video to a different object. In the context of animating humans and virtual try-on, keypoint representations have been replaced with 2D pose [1, 7, 12], human body part segmentation [79], and dense correspondences obtained from a 3D body mesh [34, 77]. The work of Chan and colleagues [7] is particularly relevant for our focus and we provide an explicit comparison in Section 5. Most of these methods model humans in tight clothing, however, and do not focus on garment dynamics. In contrast, our method learns dynamic deep garment features together with a corresponding neural renderer to synthesize high quality deformations for loose garments.

3 OVERVIEW

Given the rendering of a character with a target garment \( V \) under certain motion sequences at training time, our method learns to synthesize the image space appearance of the garment over the character given a new (unseen) body motion sequence \( M_i \) (where \( t \) refers to the frame number in the sequence) and user-specified camera parameters \( p \).

Directly generating highly dynamic garment appearance, especially loose garments, only from the body joint motion is challenging due to the highly nonlinear nature of garment dynamics. Hence, we first learn 3D dynamics of a coarse garment proxy, \( V^c \), as an intermediate representation. Then, at the core of our approach, we propose...
a neural rendering technique that learns dynamic neural features
over the coarse garment template along with a neural renderer that
interprets these features to synthesize the final appearance of
the target garment from the desired viewpoint. As shown in Fig. 2, given
the input character motion \( M_t \) represented as the trajectory of body
joints, we first train a network to drive the pre-defined coarse gar-
ment in 3D resulting in \( V^c_t \) (Section 4.1). The coarse template is not
only a downsampled version of the target garment but also lacks
detailed geometric features such as laces or multi-layer components.
Hence, the same coarse template can be shared by different target
garments of the same type (e.g., a coarse garment in the form of a
plain skirt can drive both a pleated and a laced skirt). In other words,
the dynamics of the coarse garment can be shared across different
garment styles.

Inspired by the recent work of Thies et al. [64], we learn a d-
dimensional neural feature \( F \) for the coarse garment and generate a
neural feature image \( F^p_t \) from the desired viewpoint by sampling
the texture based on the predefined UV coordinates of \( V^c \). In order
to capture the dynamics of the garment, we also define a motion
descriptor and generate another feature image \( S^c_t \) that encodes the
body motion. Our dynamic neural rendering network \( G \) synthesizes
the final garment appearance conditioned on the two neural feature
images and a sequence of rendered background images \( \{ B^p_c \} \) which
provide the appearance of the character body without the garment
(Section 4.3).

In order to ensure the generated garment appearance blends
well with varying illumination and character body rendering styles,
\( G \) uses a multi-layer perceptron (MLP) based decoder to further
harmonize the synthesized garment with the background images.
Our neural rendering component is trained end-to-end to learn both
the neural texture \( F \) and the parameters of the rendering network
\( G \) based on the objective function:

\[
F^*, G^* := \arg \min_{F, G} \sum_{t,p} \mathcal{L}(G(p^p_t, S^c_t, B^p_c), i^p_t)
\]

where \( \mathcal{L} \) is a suitable training loss computed on the generated
image and the ground truth image \( i^p_t \). We also adopt an adversarial
setup by applying a spatial-and-temporal discriminator to improve
the accuracy and consistency of the synthesised output. Next, we
describe the individual steps in detail.

4 ALGORITHM

4.1 Dynamic Coarse Garment

Given a desired body motion sequence \( M_t \), our method first synthe-
sizes the dynamics of a coarse garment proxy \( V^c \). Inspired by [21, 71],
we adopt a Joint2Coarse network, which generates the coarse gar-
ment geometry from the motion of the 3D joints as shown in Fig. 3.
We represent the body pose at each frame as the global positions
of \( J \) selected body joints (see Fig. 3). Since the dynamics of
the garment depends on the body motion in the past several frames,
we combine the pose at the current frame and the past \( K \) frames as
\( \tilde{M}_t = M_{(t-K:1)} \in \mathbb{R}^{(K+1) \times J \times 3} \) to define a motion descriptor
for frame \( t \). Note that in \( \tilde{M}_t \), all the joint positions are represented
relative to the root position at frame \( t \).

We first adopt an auto-encoder (cf., [71]) \( \xi_D(\xi_E(\cdot)) \) to learn a
compact representation of the coarse template \( V^c \) as it deforms
under varying body motions by optimizing for \( \xi_D(\xi_E(V^c)) \approx V^c \).
We also learn a motion encoder network, \( \xi_M(\cdot) \), that maps a given
motion descriptor \( \tilde{M}_t \) to the corresponding latent representation of
the coarse template at frame \( t \), i.e., \( \xi_M(\tilde{M}_t) = \xi_E(V^c_t) \). At test time,
given a new motion descriptor, we can generate the coarse garment
geometry by applying \( V^c_t = \xi_D(\xi_M(\tilde{M}_t)) \). While this approach
synthesizes plausible dynamics of the coarse garment, it does not
explicitly handle potential penetrations between the garment and
the body. We find that utilizing additional post-processing steps to
resolve such collisions as in previous work [16, 71] is not necessary
in our setting. The generated coarse template is used as a guide to
drive the dynamic neural rendering which is prone to such collisions
as shown in Fig. 4.

4.2 Dynamic Neural Garment

Given the generated coarse garment sequence, \( V^c_t \), we present a
dynamic neural rendering approach to synthesize the final target
garment appearance from a desired viewpoint. We explore the fact
that the coarse template provides spatial-temporal correspondences
across the motion sequence and learn neural features directly on the
coarse garment to encode the style-specific appearance changes be-
tween the coarse and the target garment. The global neural features
learned on the coarse garment are dynamically displaced across
frames as the garment deforms. Importantly, to encode pose-specific
dynamic appearance changes, we also condition the neural renderer
on motion features extracted from the underlying body motion.
We next describe the details of the neural and motion features, the
architecture of the neural rendering network, and the loss function
used to train it.

Learnable neural features. In order to encode the style specific
appearance relationship between the coarse and target garments,
we learn a d-dimensional neural texture \( F \) for the coarse garment
\( V^c \). In order to avoid overfitting to a specific texture resolution, our
neural texture is represented in a multi-scale manner [64]. In our

Fig. 3. Given 3D body joint positions of a motion sequence, we train an
encoder-decoder type of network to map the motion information to the
learned latent space of a coarse garment template and decode its 3D de-
formed shape.
We concatenate the pose feature images, \( \hat{\mathbf{x}}_{ij} \), where for each pixel \( v \) which we call \( \mathbf{Z}_i \) the first three channels of the neural texture to be similar to the RGB color of the corresponding pixel in the target image since our coarse template differs from the target garment both from a geometric and appearance perspective. To achieve this, for each pixel in the target rendering at frame \( t \) to different camera parameters. To achieve this, for each pixel in the target rendering at frame \( t \) from a specific viewpoint \( p \) to generate a neural feature image \( \mathbf{F}_i \).

**Motion features.** While the neural features learned over the coarse garment are geometrically displaced as the garment deforms, they are global and do not encode pose-specific appearance changes of the target garment. Hence, we condition our neural renderer on additionally defined motion features. Since our neural renderer is not restricted to a specific view, our motion features should be invariant to different camera parameters. To achieve this, for each pixel in the target rendering at frame \( t \) from a specific viewpoint \( p \), we first compute the 3D position of the corresponding location on the surface of the deformed coarse garment via a barycentric coordinate, which we call \( \mathbf{v}_i \). We form a \( J \) dimensional pose feature image \( \mathbf{S}_i^p \) where for each pixel \( i \) the \( j^{th} \) channel encodes the distance of the corresponding vertex to the \( j^{th} \) joint:

\[
\mathbf{S}_i^p(j) = \exp\left( -\| \mathbf{v}_i^j - \mathbf{M}_i^j \|^2 / \sigma \right).
\]

In order to incorporate the motion information in the past \( L \) frames, we concatenate the pose feature images, \( \mathbf{S}_i^p(t-L+1:t) \), along the channel dimension to construct a motion feature image \( \mathbf{S}_i^p \) with \( J \times (L+1) \) channels.

**Neural descriptor map.** We concatenate the motion feature image \( \mathbf{S}_i^p \) and the neural feature image \( \mathbf{F}_i^p \) to form a final neural descriptor map \( \mathbf{Q}_i^p = [\mathbf{F}_i^p, \mathbf{S}_i^p] \) for each frame \( t \). Our renderer network takes as input the features \( \mathbf{Q}_i^p \) along with the corresponding background image \( \mathbf{B}_i^p \) and synthesizes the final rendering of the target garment, as we will describe next.

### 4.3 Rendering Network

Our rendering network \( G \) is mainly composed of a temporally coherent encoder and a layered based decoder. Given the neural descriptor maps of two consecutive frames \( \mathbf{Q}_t^p \), \( \mathbf{Q}_{t-1}^p \), we first encode them into the latent space resulting in latent codes \( \mathbf{Z}_t^p \) and \( \mathbf{Z}_{t-1}^p \), respectively. We utilize a spatially adaptive normalization layer (SPADE) \([49]\) to normalize \( \mathbf{Z}_t^p \) into \( \tilde{\mathbf{Z}}_t^p \) conditioned by \( \mathbf{Z}_{t-1}^p \). This helps to implicitly achieve temporal consistency between the latent codes of consecutive frames. The layered based decoder takes as input \( \tilde{\mathbf{Z}}_t^p \) along with the background image \( \mathbf{B}_t^p \) and synthesizes the appearance of the target garment on the character. In order to ensure further temporal consistency between consecutive frames, we introduce a patch-based temporal-and-spatial discriminator to train our rendering network. We show the architecture of our network in Fig. 4.

**Temporally coherent encoder.** As shown in Fig. 4, the encoder component of our network consists of a set of convolutional layers that encode the input neural descriptor maps \( \mathbf{Q}_t^p \) into latent codes \( \mathbf{Z}_t^p \). We define a SPADE block which is used to normalize the signal conditioned with the concatenation of \( \mathbf{Z}_t^p \) and \( \mathbf{Z}_{t-1}^p \). This normalization with respect to the previous frame enables to extract the temporal dependency between the consecutive frames at every spatial location. Our SPADE block first normalizes the input feature map using scale \( \gamma \) and bias \( \beta \) parameters learned from \( [\mathbf{Z}_t^p, \mathbf{Z}_{t-1}^p] \) with respect to both each channel \( c \) and spatial location \( (x, y) \) as described in \([49]\). Specifically, for an input signal \( \mathbf{w} \), we obtain the

![Diagram of the rendering process](image-url)
normalized \( \hat{w} \) as:
\[
\hat{w}[c, x, y] := \text{SPADE}(w) = \frac{w[c, x, y] - \mu_c}{\sigma_c} + \beta_{c, x, y},
\]
where \( \mu_c \) and \( \sigma_c \) are respectively the mean and standard deviation of \( w \) in channel \( c \). We apply our SPADE blocks to \( Z_t^p \) in a residual style. Specifically we obtain the SPADE-normalized latent code \( Z_t^p \) as:
\[
\hat{Z}_t^p = \text{SPADE}(I(\text{SPADE}(I(2Z_t^p)))) + \text{SPADE}(I(Z_t^p)),
\]
where \( I(\cdot) \) refers to an instance normalization layer.

**Layered based decoder.** The architecture of our decoder consists of a set of ConvTranspose2D layers symmetric to the encoder. Given \( \hat{Z}_t^p \) these layers first generate a feature image \( U_t^p \) with the same width and height as the target rendering. The provided latent code and hence \( U_t^p \) captures the appearance and dynamics of the target garment only. In order to composite the garment with the character body and the background, we introduce a layer blending module to blend \( U_t^p \) with \( B_t^p \), the features extracted from the provided background image \( B_t^p \). Specifically, we generate a mask \( A_t^p \) and apply the learned mask to blend the foreground and the background by \((1 - A_t^p) \cdot B_t^p + A_t^p \cdot U_t^p \). We finally apply two residual convolution blocks to refine the result and generate the final rendering \( R_t^p \).

### 4.4 Loss Function

We train our dynamic neural rendering component to learn the parameters of the network \( G \) and the neural feature \( F \) jointly in an end-to-end manner. Given the final rendering \( R_t^p \) synthesized by the network and the corresponding ground truth image \( I_t^p \), we first consider an \( L_1 \) loss with respect to colours as well as multi-layer features of the pretrained network VGG:
\[
L_{\text{percept}} := \sum_i \| \text{VGG}^i[R_t^p] - \text{VGG}^i[I_t^p] \|_1 + \| R_t^p - I_t^p \|_1.
\]

**Temporal-and-spatial discriminator.** Furthermore, we adapt an adversarial loss to ensure the spatio-temporal plausibility of the renderings. Specifically, we train a discriminator network \( D \) that takes consecutive frames as input and classifies them as real or fake. We provide two types of input to \( D \). First, we provide \([R_t^p, I_{t-1}^p] \) as input while the samples \([I_t^p, I_{t-1}^p] \) provide real examples. We also provide \([I_t^p, R_{t+1}^p] \) as input where samples \([I_{t+1}^p, I_t^p] \) provide the real examples. These samples ensure that the generated rendering at frame \( t \) is temporally consistent with the ground truth previous and next frames. We adopt a patch-based discriminator \( D \) (cf., [24]) to model high frequency details by restricting the attention to local patch structures. To further ensure temporal consistency, given two renderings \([I_t^p, I_{t-1}^p] \) of the neighboring frames, \( D \) tries to identify if each \( N \times N \) patch is real or fake both in the spatial domain as well as the temporal domain along the channel direction when they are concatenated.

The overall loss function for \( D \) to minimize is:
\[
L_D = -\log(D[I_t^p, I_{t-1}^p]) - \log(1 - D[R_t^p, I_{t-1}^p]),
\]
\[
L_D = -\log(D[I_{t+1}^p, I_t^p]) - \log(1 - D[I_{t+1}^p, R_t^p]),
\]
\[
D := \arg\min_D [L_D + L_D].
\]

For the rendering network, the adversarial loss is defined as:
\[
L_{\text{GAN}} = -\log(D[R_t^p, I_{t-1}^p]) - \log(1 - D[R_t^p, I_{t+1}^p]).
\]

We also use feature matching in multiple discriminator layers \( D^i[\cdot] \) to enforce similarity across different scales of features:
\[
L_{\text{feat}} = \sum_i \| D^i[R_t^p, I_{t-1}^p] - D^i[R_t^p, I_{t+1}^p] \|_1
\]
\[
+ \sum_i \| D^i[I_{t+1}^p, R_t^p] - D^i[I_{t+1}^p, I_t^p] \|_1.
\]

Given the different losses, we learn the weights of network \( G \) as well as the neural features \( F \) that minimize:
\[
G, F := \arg\min_{G, F} [\lambda_1 L_{\text{feat}} + \lambda_2 L_{\text{percept}} + \lambda_3 L_{\text{GAN}}].
\]

Following SPADE [49] and SEAN [80], we set \( \lambda_1 = 5, \lambda_2 = 10, \lambda_3 = 0.5 \) in our experiments.

### 4.5 Post-processing

While \( G \) consists of a layering module to composite the synthesized garment with the target body, occasionally we observe that the depth ordering between the arms which are highly dynamic and garment types such as skirts is not correct. To resolve these, we adopt a simple heuristic through an image layer re-composition approach. We first identify the arm regions in the character body renderings and the garment region in the final renderings. If the arm is closer to the camera (i.e., close or in front of the hip) but is occluded by the garment in the final rendering, we bring it (i.e., layer it) to the front.

## 5 RESULTS AND EXPERIMENTS

In this section, we evaluate our approach with various scenarios and show that our method outperforms existing character appearance synthesis approaches qualitatively and quantitatively. We show that our method generalizes across unseen motion, novel camera view, and adapts to novel body shape, background and environment illumination with an efficient fine-tuning strategy.

### 5.1 Data generation

To train our method, we establish a synthetic dataset by first creating virtual avatars using Adobe Fuse. We rig and animate each avatar via Mixamo (https://www.mixamo.com/) to generate a training motion sequence of 850 frames. Next, we run a physically-based simulation using Marvellous Designer [10] to generate the ground truth mesh sequence for both the coarse template and the target garments. Specifically, we use a particle distance of 30mm when simulating the coarse template and a distance of 10mm when simulating the target garments to capture high-frequency details. Since we know the ground truth body joint positions, we can also compute the motion descriptor used by both the coarse garment synthesis (Section 4.1) as well as the motion features used by the dynamic neural rendering (Section 4.2) of our method. The coarse templates are used to train our Joint2Coarse network to provide the neural feature proxies. To ensure generalisation of the rendering network across camera views, at each frame, we randomly position 10 cameras on a circle around the avatar. Under each view \( \gamma \), we generate the required ground
truth input and output maps for 3 consecutive frames \(t-1, t, t+1\) as required in the temporal adversarial loss function (Section 4.4). We set up the rendering scene in Blender with an HDRI environment map. In order to keep the illumination consistent across different views, we rotate the HDRI map to keep a fixed relative position with respect to the camera. In total, our training set for the rendering network includes 8500 samples.

Our dataset consists of three coarse garment templates: a long skirt (2287 vertices), a short skirt (1756 vertices), and a full-body dress (5087 vertices). We also define 5 different target garments: (i) a multi-lace skirt with 48389 vertices; (ii) a double-layer skirt with 33196 vertices; (iii) a tango skirt with 26718 vertices; (iv) a short hem skirt with 56657 vertices; and (v) a full-body combination composed of a delicate dress skirt, blouse, ropes, and strings with a total of 51941 vertices. For the short hem skirt, we use a particle distance of 5mm when training our method to produce rich folds, while the others use 10mm spacing. For the first three types of skirts, we use the same coarse long skirt template. For each coarse template and target garment combination, we train a specific neural rendering network along with a new set of neural textures.

5.2 Implementation details

Our Joint2Coarse network consists of a shape encoder \(\zeta_E(\cdot)\), a shape decoder \(\zeta_O(\cdot)\), and a motion encoder \(\gamma(\cdot)\). The shape encoder \(\zeta_E(\cdot)\) takes the \(v \times 3\)-dimensional vector as input \((v\) is the number of vertices in the template) and maps it into a latent space through 6 fully connected layers with output dimensions gradually decreasing to 2048, 1024, 512, 256, 128, and 64. The shape decoder reverses this process symmetrically. We build a motion descriptor consisting of \(J = 19\) body joint positions of the current as well as the past 34 frames (a total of \(K = 35\) frames), resulting in \(M_t \in \mathbb{R}^{35 \times 19 \times 3}\). We flatten the motion descriptor and obtain a 969-dimensional vector as the input of \(\gamma(\cdot)\). The motion descriptors are then mapped into the 64-dimensional latent space through 4 fully connected layers with output dimensions gradually decreasing to 512, 256, 128, and 64. We adopt ReLU activations for all the layers and dropout connections with a rate of 0.05. The encoders and the decoder are trained by the RMSprop algorithm with a learning rate of \(10^{-3}\).

As shown in Fig. 4, the input of the rendering network is dynamic pairs of neural descriptor maps \(Q^t_E\) and \(Q^t_O\). Each map consists of per-pixel neural features of dimension 64 sampled from a neural texture hierarchy composed of 4 levels, each with 16 channels. In addition, the maps also include per-pixel motion features with dimension \(J \times L\), where \(J = 19\) (the 19 body joints), and \(L = 5\) (the 5 sampled previous frames with an interval of 2 to cover a period of 10 frames. The encoder first down-samples the resolution of the input maps from \(512 \times 512\) to \(16 \times 16\), and increases the feature dimension to 512 by 2D convolution layers each with a Leaky ReLU activation. In the SPADE module, we use a SPADE residual block [49], keeping the feature size to be \(16 \times 16 \times 512\). Then we use 2D convTranspose layers to increase the output resolution to \(512 \times 512\) but still maintain high dimensional features of size 32. After blending the background image features and the garment rendering features, we apply 2 residual convolution blocks to output the final rendering images with a size of \(512 \times 512\) in RGB color space. We adopt a patch size of \(70 \times 70\) [24] in our temporal-and-spatial discriminator. In the training stage, we use the Adam optimiser, with the default parameters \(\beta_1 = 0.9, \beta_2 = 0.999\), and a learning rate of \(l_{r_1} = 1.e - 4\) for the rendering network including the neural texture optimization, and \(l_{r_2} = 3.e - 4\) for the temporal-and-spatial discriminator. It took about 35 epochs to converge when training the rendering network with a batch size of 2.

Fig. 5. Our method can synthesize the target garment appearance from different viewpoints not seen during training.

Fig. 6. We test our method on the training motion sequence on unseen viewpoints. For each example, our result and the ground truth. Notice how our results are closer to the ground truth in the highlighted regions compared to the nearest views in the training data shown in green.
5.3 Results and evaluation

We now evaluate the generalization ability of our method. While showing illustrative figures, we also provide a supplementary video to better evaluate the visual quality of the generated garments under motion.

(i) Unseen views. For the motion sequence seen during training, we test how well our method works with different, unseen viewpoints as shown in Fig. 5 and 6. In Fig. 6, for each test view, we show the training samples with the nearest viewpoint. As can be seen in the supplemental video, compared to the nearest training samples, our method generates temporally coherent sequences for the unseen views.

(ii) Unseen motions. Next, we evaluate the generalization ability of our method across unseen motion sequences. In Fig. 7, we first visualize the distribution of the body poses observed both in training and testing motion sequences via t-SNE [38] over the 3D body joint positions with respect to the root joint (i.e., in local coordinates). We use the liexa break dancing motion sequence to train our network and test on the pagode variation, gandy variation, and circle walk sequences. Our method generalizes well to motion sequences covered by the distribution of the poses seen during training. As with many deep learning approaches, as the distribution of testing motions (e.g., circle walk) becomes significantly different than the training data, we observe a drop in performance (e.g., flickering across frames).

(iii) Different body shapes. We also evaluate our method with varying body shapes, as shown in Fig. 8. Given neural textures and the renderer learned for a particular body shape, e.g., a thin body, we fine-tune the network for a new body shape. We use the same coarse template of the thin body. We find that fine-tuning the network reduces the number of training iterations to 20 – 30% compared to training from scratch.

(iv) Different background. Finally, we extend our method to synthesize images with various background images demonstrating different illumination conditions by fine-tuning the last two layers of our layered based decoder jointly with the discriminator as shown in Fig. 11. Our fine-tuning strategy allows us to achieve plausible results by training with two example images rendered with novel background and environment illumination, which takes 100 training iterations to converge with a batch size of 2.

Real capture. Our method can be combined with 3D body motion estimation methods to drive a 3D character wearing a desired garment and synthesize the corresponding dynamic appearance. In Fig. 12, we provide an example where we have used the method of Rempe et al. [53] to extract the 3D human body motion from the source video.

Computational performance. Our method learns to simulate and render a garment efficiently given a body motion sequence. Once trained, our method takes 40 ms per frame to generate the coarse template mesh and 20ms per frame to render the final image. We run our experiments on a PC with Intel Xeon CPU E5-1650, 64GB of memory, and an NVIDIA GeForce GTX 1080Ti graphics card. In comparison, in a professional garment simulation tool such as Marvellous Designer, each frame of the simulations takes 1.58 seconds for the multi-lace skirt, 1.05 seconds for the tango skirt, 1.10 seconds for the double-layer skirt, 2.03 seconds for the short hem skirt, and 2.31 seconds for the full-body dress.

5.4 Baseline comparisons

We compare our method against image and video translation methods, (a) Pix2Pix [24] and (b) Vid2Vid [69], (c) a human body reposing method Everybody Dance Now [8] (EBDN), and (d) a closely related neural rendering method, Deferred Neural Rendering (DNR) [64].

For Pix2Pix [24], we experimented with two variants: (i) learning the mapping between the joints and the final appearance; (ii) learning the mapping between the rendering of the coarse template and the final appearance. The first version provided unsatisfactory results. Hence, we only present the second version in our comparisons.

For Vid2Vid [69], we adopt Openpose [6] to extract 2D skeletons and Densepose [17] to extract dense UV maps for the character body as input. The extracted 2D skeleton is also used for EBDN to drive the motion synthesis. For DNR [64], we use our generated coarse template as the underlying geometry for each frame and jointly learn a corresponding neural texture and a neural renderer.
Fig. 9. We evaluate our method on different garment types with unseen motion sequences. We note that we learn a garment specific neural texture and renderer.
Fig. 10. We compare our method to image and video translation methods, Pix2Pix [24] and Vid2Vid [68] as well as DNR [64], a deferred neural rendering approach, and EBDN [8] that focuses on pose editing of humans. Our approach outperforms these baseline methods.

Fig. 11. Our dynamic neural renderer can easily be fine-tuned with a few examples to generalize to different backgrounds showing various illumination conditions.

Fig. 12. Given a source input video, we use the estimated motion of the actor to drive a 3D character while synthesizing plausible garment appearances as described in the original paper with the L1 loss. We also train the same network with adversarial loss as in our method (Section 4.4).

All the methods are trained with the same training set as ours and tested with the same motion under different viewpoints as shown in Fig. 10. In Table 1, we provide a quantitative comparison and show that our method outperforms the baselines with seen motion. With the strongest baseline, EBDN, we also show a comparison with unseen motion in the supplementary video. Our method performs significantly better than the baseline in this case.

Table 1. For quantitative comparisons, we show the mean ($\mu_{mse}$) and standard deviation ($\sigma_{mse}$) of the per-frame mean-square-error between different methods and the ground truth image; the Fréchet Inception Distance (FID) [20]; and the video based Fréchet Inception Distance (V-FID) [68] for the whole sequence generated by each method. The scale of the pixel value is $[0, 1]$ for each channel.

|                | Pix2Pix | Vid2Vid | DNR-L1  | DNR-GAN | EBDN   | Ours  |
|----------------|---------|---------|---------|---------|--------|-------|
| $\mu_{mse}$    | 0.591   | 0.734   | 0.204   | 0.254   | 0.204  | 0.116 |
| $\sigma_{mse}$ | 0.120   | 0.188   | 0.081   | 0.082   | 0.064  | 0.045 |
| FID            | 75.77   | 96.21   | 71.12   | 36.41   | 38.31  | 11.39 |
| V-FID          | 2.18    | 3.31    | 2.08    | 1.68    | 0.74   | 0.40  |

5.5 Ablation Study

(i) Effect of the coarse template. When choosing the coarse templates for a particular garment, we tend to choose a garment of a neutral type without geometric details such as laces or layering. While a coarse template with more geometric details similar to a specific target garment can potentially yield better results, it is hard to generalize across different target garments with the same template. Hence, we aim to achieve a balance. As seen in Fig. 9, we use the same plain long skirt garment for three different skirt types, including a multi-lace skirt, a double-layer skirt, and a tango style skirt. For another short hem skirt, we experimented with using both the aforementioned long skirt template and a short skirt template. While the use of the short skirt template improves the quality of the results, the long skirt template still yields reasonable results as shown in Fig. 13 and Table 2.

(ii) Effect of motion features. To demonstrate the importance of using motion features ($S_P^t$) for dynamic neural rendering, we train the renderer with and without motion features. Our renderer is able to model the rough motion of the garment even without motion features due to the adoption of the coarse template. However, the
Dynamic Neural Garments

•

long template short templateground truth

Fig. 13. For a target short pleated skirt, we show the results of using a long and a short skirt template.

5 random views 10 random views ground truth
no motion features with motion features ground truth

Fig. 14. We perform an ablation study to demonstrate the importance of the number of views used during training as well as the use motion features representing the underlying body motion.

results improve significantly and high-frequency details are better preserved when motion features are added as shown in Fig. 14 and Table 2, increasing the number of views improves the visual quality of the results.

Table 2. We show the Fréchet Inception Distance (FID) [20] for the experiments in our ablation study. We show that our approach achieves better performance compare to other alternatives.

|                      | 5 random views | 10 random views (ours) |
|----------------------|---------------|------------------------|
| FID score            | 24.00         | 19.58                  |
| multi-lace skirt in Fig 14 | no motion features | with motion features (ours) |
| FID score            | 13.46         | 11.34                  |
| green skirt in Fig 13 | long template | short template         |
| FID score            | 8.34          | 7.57                   |

6 CONCLUSION AND FUTURE WORK

In this paper, we presented a novel neural rendering pipeline to jointly simulate and render dynamic garments. Our two-stage solution first generates intermediate 3D deformations of a coarse template, and then, by learning deep neural features attached to the coarse template, we synthesize the final appearance of the target garment. In order to capture dynamic appearance changes, we augment the learned neural features with motion features that encode the underlying body motion. Finally, our temporal-and-spatial discriminator ensures to produce plausible details in a temporally coherent manner. We evaluate our method thoroughly across unseen views, character body motion, environment illumination for various garment types and demonstrate the state of the art performance. We also test our framework on real video sequences to drive both the motion of the avatar and the draped garment dynamics.

Limitations and Future Work. Our method has limitations, which we plan to address in future work. First, our learned network is character- and garment-specific. While we show how our network can be fine-tuned to handle different body shapes, a promising future direction is to employ few-shot learning techniques or investigate meta-learning to improve generalization.

Second, we encode the garment appearance with the learned neural features. Hence, we need to learn a new set of neural features for different garment texture patterns. A possible solution is to train our rendering network to synthesize the UV coordinates and a shading layer instead. Such UV coordinates can then be used to sample different texture patterns.

Third, powered by a CNN based network architecture and the patch-based generative method, our network can be trained with a small motion dataset with only 850 frames. The garment appearance generated for each frame is plausible, but some artifacts (e.g., temporal consistency and flickering) can be observed, especially when testing on a new motion that is not covered well by the distribution of training poses. Such artifacts can be largely resolved by enriching our training dataset with more motion clips and sampled views.

Fourth, in our current setup, we do not model the garment intersection with body skeleton, or the interaction between the foreground character and the background scene. Therefore, our method
accurate 3d cloth draping. In Proceedings of the IEEE International Conference on Computer Vision. 8739–8748.
[19] Fabian Hahn, Bernhard Thomaszewski, Stefan Coros, Robert W Sumner, Forrester Cole, Mark Meyer, Tony DeRose, and Markus Gross. 2014. Subspace clothing simulation using adaptive bases. ACM Transactions on Graphics (TOG) 33, 4 (2014), 1–9.
[20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Advances in neural information processing systems. 6626–6637.
[21] Daniel Holden, Bang Chi Duong, Sayantan Datta, and Derek Nowrouzezahrai. 2016. Sparse neural textures: fast data-driven interactive simulation. In Proceedings of the 18th annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation. 1–12.
[22] Xun Huang and Serge Belongie. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In Proceedings of the IEEE International Conference on Computer Vision. 1501–1510.
[23] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. 2020. ARCH: Animatable Reconstruction of Clothed Humans. In Proceedings of the CVPR 2020. IEEE, Seattle, Washington. https://doi.org/10.1109/CVPR42600.2020.00316
[24] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition. 1125–1134.
[25] Ning Jin, Yilin Zhu, Zhenglan Geng, and Ronald Fedkiw. 2018. A pixel-based framework for data-driven clothing. arXiv preprint arXiv:1812.01677 (2018).
[26] Laszlo Kavan, Dan Gerszewski, Adam W. Bargteil, and Peter-Pike Sloan. 2011. Physics-inspired Upsampling for Cloth Simulation in Games. ACM Trans. Graph. 30, 4, Article 93 (July 2011), 10 pages.
[27] Tsz-Ho Kwok, Yan-Qin Zhang, Charlie CL Wang, Yong-Jin Liu, and Kai Tang. 2016. Styling evolution for tight-fitting garments. IEEE transactions on visualization and computer graphics 22, 5 (2016), 1580–1591.
[28] Zorah Lahner, Daniel Cremers, and Tony Tung. 2018. Deepwrinkles: Accurate and realistic clothing modeling. In Proceedings of the European Conference on Computer Vision (ECCV). 667–684.
[29] Minchen Li, Alla Sheffer, Eitan Grinspun, and Nicholas Vining. 2018. FoldSketch: Enriching Garments with Physically Reproducible Folds. ACM Transaction on Graphics 37, 4 (2018). https://doi.org/10.1145/3197517.3201310
[30] Zhengli Li, Simon Niklaus, Noah Snavely, and Oliver Wang. 2020. Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. arXiv preprint arXiv:2011.13084 (2020).
[31] Jinhong Liang, Ming Lin, and Vladlen Koltun. 2019. Differentiable Cloth Simulation for Inverse Problems. In Advances in Neural Information Processing Systems. 771–780.
[32] Jing Luo, Yuan Yao, Lu Yuan, Gang Huang, and Sing Bing Kang. 2017. Visual attribute transfer through deep image algebra. arXiv preprint arXiv:1705.05088 (2017).
[33] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. 2019. Neural Volumes: Learning Dynamic Rendering of the 18th annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation. 9 pages. https://www.marvelousdesigner.com/
[34] Chris De Paoli and Karon Singh. 2015. SecondSkin: Sketch-Based Construction of Layered 3D Models. ACM Trans. Graph. 34, 4, Article 126 (July 2015), 10 pages. https://doi.org/10.1145/2766948
[35] Haoye Dong, Xiaodan Liang, Xiaochen Shen, Bowen Wu, Bing-Cheng Chen, and Jian Yin. 2019. FW-GAN: Flow-Navigated Warping GAN for Video Virtual Try-On. In Proceedings of the IEEE-CVF International Conference on Computer Vision (ICCV).
[36] Wei-Wen Feng, Yizhou Yu, and Byung-Uck Kim. 2010. A Deformation Transformer for Real-Time Cloth Animation. ACM Trans. Graph. 29, 4, Article 108 (July 2010), 9 pages.
[37] Jakub Fiser, Ondrej Jamirova, Michal Lukac, Eli Shechtman, Paul Asente, Jingwan Lu, and Daniel Sylkora. 2016. StyLit: illumination-guided example-based stylization of 3D renderings. ACM Transactions on Graphics (TOG) 35, 4 (2016), 1–11.
[38] Russell Gillette, Craig Peters, Nicholas Vining, Essex Edwards, and Alla Sheffer. 2015. Real-Time Dynamic Wrinkling of Coarse Animated Cloth. In SCA (SCA ’15).
[39] Peng Guan, Loretta Reiss, David A Hirshberg, Alexander Weiss, and Michael J Black. 2012. Draping: Dressing any person. ACM Transactions on Graphics (TOG) 31, 4 (2012), 1–10.
[40] Riza Alp Guler, Natalia Neverova, and Iasonas Kokkinos. 2018. Densenpose: Dense human pose estimation in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 7297–7306.
[41] Erhan Gundogdu, Vietor Constantin, Amrollah Sefidokini, Minh Dang, Mathieu Salzmann, and Pascal Fua. 2019. Garnet: A two-stream network for fast and
