

# Pose-Guided Human Animation from a Single Image in the Wild

Jae Shin Yoon<sup>†</sup> Lingjie Liu<sup>‡</sup> Vladislav Golyanik<sup>‡</sup> Kripasindhu Sarkar<sup>‡</sup>  
 Hyun Soo Park<sup>†</sup> Christian Theobalt<sup>‡</sup>

<sup>†</sup>University of Minnesota

<sup>‡</sup>Max Planck Institute for Informatics



**Figure 1.** We present a new method to synthesize a sequence of animated human images from a single image. The synthesized images are controlled by the poses as shown in the inset image. It is capable of generating full appearance of the person at diverse poses, reflecting the input foreground and background in the presence of occlusion and 3D shape deformation, *e.g.*, the occluded texture of the back.

## Abstract

We present a new pose transfer method for synthesizing a human animation from a single image of a person controlled by a sequence of body poses. Existing pose transfer methods exhibit significant visual artifacts when applying to a novel scene, resulting in temporal inconsistency and failures in preserving the identity and textures of the person. To address these limitations, we design a compositional neural network that predicts the silhouette, garment labels, and textures. Each modular network is explicitly dedicated to a subtask that can be learned from the synthetic data. At the inference time, we utilize the trained network to produce a unified representation of appearance and its labels in UV coordinates, which remains constant across poses. The unified representation provides an incomplete yet strong guidance to generating the appearance in response to the pose change. We use the trained network to complete the appearance and render it with the background. With these strategies, we are able to synthesize human animations that can preserve the identity and appearance of the person in a temporally coherent way without any fine-tuning of the network on the testing scene. Experiments show that our method outperforms the state-of-the-arts in terms of synthesis quality, temporal coherence, and generalization ability.

## 1. Introduction

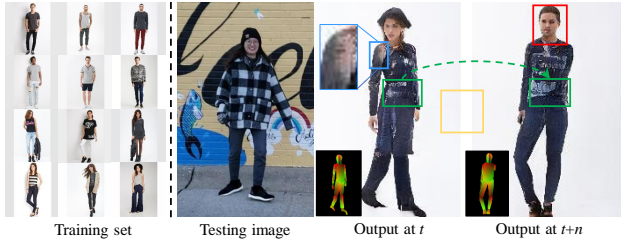
Being able to animate a human in everyday apparel with an arbitrary pose sequence from just a single still image opens the door to many creative applications. For example,

animated photographs can be much more memorable than static images. Furthermore, such techniques not only simplify and democratize computer animation for non-experts, they can also expedite pre-visualization and content creation for more professional animators who may use single image animations as basis for further refinement.

Tackling this problem using classical computer graphics techniques is highly complex and time consuming. A high-quality 3D textured human model needs to be reconstructed from a single image and then sophisticated rigging methods are required to obtain an animatable character. An alternative is to apply 2D character animation methods [17, 19] to animate the person in the image. However, this approach cannot visualize the occluded parts of the character.

In this paper, we approach this problem using a pose transfer algorithm that synthesizes the appearance of a person at arbitrary pose by transforming the appearance from an input image without requiring a 3D animatable textured human model. Existing works on pose transfer have demonstrated promising results only when training and testing take place on the same dataset (*e.g.*, DeepFashion dataset [29]), and some require even more restrictive conditions that testing is performed on the same person in the same environment as training. [8, 26, 27]. However, the domain difference between training and testing data in real applications introduces substantial quality degradation.

A core challenge of pose transfer lies in lack of data that span diverse poses, shapes, appearance, viewpoints, and background. This leads to limited generalizability to a testing scene, resulting in noticeable visual artifacts as shown in



**Figure 2.** The pose transfer results synthesized by a state-of-the-art method [45] on an unconstrained real-world scene, where the network is trained on the Deep Fashion dataset [29]. The target body pose is shown in the inset (black). Each box represents the type of the observed artifacts such as loss of identity (red), misclassified body parts (blue), background mismatch (yellow), and temporal incoherence (green).

Fig. 2. We address this challenge by decomposing the pose transfer task into modular subtasks predicting silhouette, garment labels, and textures where each task can be learned from a large amount of synthetic data. This modularized design makes training tractable and significantly improves result quality. Explicit silhouette prediction further facilitates animation blending with arbitrary static scene backgrounds.

In inference phase, given the trained network from the synthetic data, we introduce an efficient strategy for synthesizing temporally coherent human animations controlled by a sequence of body poses. We first produce a unified representation of appearance and its labels in UV coordinates, which remains constant across different poses. This unified representation provides an incomplete yet strong guidance to generating the appearance in response to the pose change. We use the trained network to complete the appearance and render it with the background. Experiments show that our method significantly outperforms the state-of-the-art methods in terms of synthesis quality, temporal consistency, and generalization ability.

Our technical **contributions** include (1) a novel approach that can generate a realistic animation of a person from a single image controlled by a sequence of poses, which shows higher visual quality and temporal coherence, and generalizes better to new subjects and backgrounds; (2) a new compositional pose transfer framework that produces the silhouette mask, garment labels, and textures, which makes the pose transfer tractable; (3) an effective inference method by generating a unified representation of appearance and its labels for image synthesis, enforcing temporal consistency and preserving identity in the presence of occlusion and shape deformation.

## 2. Related Work

We review the literature for human pose transfer and its application to the pose-guided video generation.

**Human Pose Transfer** Pose transfer refers to the prob-

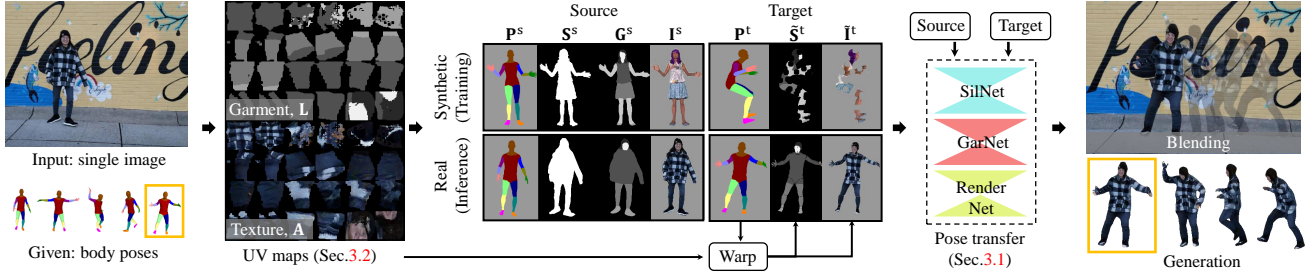
lem of synthesizing human images with a novel user-defined pose. The conditioning pose is often captured by 2D keypoints [31, 58, 63, 11, 42, 51] or parametric mesh [28, 32, 40, 45]. Many recent works also use Densepose [4] which is the projection of SMPL model with UV parameterization in the image coordinates, as conditioning input. This enables direct warping of pixels of the input image to the spatial locations at the output with target pose [32, 40, 45]. While the aforementioned methods produce photo-realistic results within the same dataset, they often exhibit serious artifacts on in-the-wild scenes, such as pixel blending around the boundaries between the different garment types.

To address these limitations, some recent methods use garment segmentation map, *i.e.*, a label image where each pixel belongs to a semantic category such as clothing, face, and arm, as input to a neural network [36, 49, 10, 39]. [9] condition garment type, whereas [5] handles each garment parts in different transformation layers to preserve the clothing style in the generated image. However, these works still do not generalize to new appearances and unseen scenes.

Some new methods explicitly handle appearance in the occluded areas by matching their style to the visible regions. [1] transforms the features of the input image to a target body pose with bidirectional implicit affine transformation. [15, 44] learn pixel-wise appearance flow in an unsupervised way based on the photometric consistency. [15] establishes direct supervision by fitting a body model to the images. However, the predicted warping fields is often unstructured, resulting in artifacts such as shape distortion.

**Pose-Guided Video Generation** Since the methods for pose transfer are designed to output a single image, their application to a sequence of poses to perform pose guided video generation can exhibit temporal inconsistency. To mitigate this problem, many methods enforce explicit temporal constraints in their algorithm. [8] predicts the person image in two consecutive frames. [57] conditions the temporally coherent semantics on a generative adversarial network. Recent video generation approaches have leveraged the optical flow prediction [54], local affine transformation [47], grid-based warping field [47], body parts transformation [62], and future frame prediction [7, 55] to enforce the temporal smoothness. [26] learns to predict a dynamic texture map that allows rendering physical effects, *e.g.*, pose-dependent clothing deformation, to enhance the visual realism on the generated person. Unfortunately, the above methods are either person-specific or requiring the fine-tuning on unseen subjects for the best performance. While few-shot video generation [53] addressed this generalization problem, it still requires fine-tuning on the testing scene to achieve full performance. In contrast, our method works with a single conditioning image in the wild and performs pose guided video synthesis without any fine-tuning.

**Other Related Techniques** In contrast to the data-driven



**Figure 3.** Overview of our approach. Given an image of a person and a sequence of body poses, we aim for generating video-realistic human animation. To this end, we train a compositional pose transfer network that predicts silhouette, garment labels, and textures with synthetic data (Sec. 3.1). In inference phase, we first produce a unified representation of appearance and garment labels in the UV maps, which remains constant across different poses, and these UV maps are conditioned on our pose transfer network to generate person images in a temporally consistent way (Sec. 3.2). The generated images are composited with the inpainted background to produce the animation.

neural rendering methods, few works reconstruct a personalized animatable 3D model from a single image. For example, [3] reconstructs 3D geometry by regressing shape in UV-space. [16] learns an implicit function from a neural network to predict person’s surface and appearance on top of a parametric body model. [56] leverages the graphics knowledge, *e.g.*, skinning and rigging, to enable the character animation from a single image.

### 3. Methodology

Our goal is to synthesize human animations from a single image guided with a sequence of arbitrary body poses. The overview of our pipeline is outlined in Fig. 3. In the training stage, our pose transfer network learns to generate a person’s appearance in different poses using a synthetic dataset which provides full ground truth. At inference time, given a single image of a person and a different body pose, the learned pose transfer network generates the person’s appearance that is conditioned on the partial garment and texture warped from the coherent UV maps (scene-specific priors). The generated foreground is blended with the inpainted background. In Sec. 3.1, we introduce our compositional pose transfer network, and in inference time, we use this network to create coherent UV maps and human animation from a single image in Sec. 3.2.

#### 3.1. Compositional Pose Transfer

The problem of pose transfer takes as input a source image  $I^s$  and a target pose  $P^t$  and generates an image of the person in the target pose  $I^t$ :

$$I^t = f(P^t, I^s). \quad (1)$$

where the superscript  $s$  denotes the source as the domain of the observation from the input image, and  $t$  denotes the target as of the generation from a body pose.

Albeit possible, directly learning the function in Eq. (1) is challenging as requiring large amount of multiview data [31, 36, 45], *i.e.*, it requires to learn the deformation

of the shape and appearance with respect to every possible 3D pose, view, and clothing style. This results in a synthesis of unrealistic human images that are not reflective of the input testing image as shown in Fig. 2. We address this challenge by leveraging synthetic data that allows us to decompose the the function into the modular functions that are responsible to predict silhouette, garment labels, and appearance, respectively. This makes the learning task tractable and adaptable to the input testing image.

#### 3.1.1 Dataset and Notation

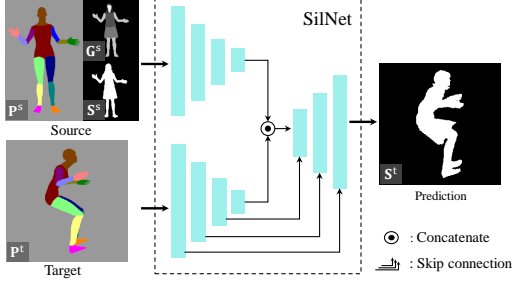
For training, we use 3D people synthetic dataset [43] which contains 80 subjects in diverse clothing styles with 70 actions per subject captured from four different virtual views, where each action is a sequence of 3D poses. For each subject we randomly pick two instances as the source and target with different views and 3D poses. Each instance contains the following associated information:

- Image:  $I \in \mathbb{R}^{W \times H \times 3}$  is the person image where the foreground is masked using  $S$ .
- Pose map:  $P \in \{0, \dots, 14\}^{W \times H}$  is a map of body-part labels of the undressed body (14 body parts and background).
- Silhouette mask:  $S \in \{0, 1\}^{W \times H}$  is a binary map indicating one if it belongs to the person foreground, and zero otherwise.
- Garment labels:  $G \in \{0, \dots, 6\}^{W \times H}$  is a map of garment labels of dressed human body, indicating hair, face, skin, shoes, garment top and bottom, and background.

In inference time, given  $I^s$  and  $P^s$ , we estimate the  $P^s$ ,  $S^s$  and  $G^s$  from  $I^s$  using off-the-shelf methods, and our pose transfer network predicts  $S^t$ ,  $G^t$ , and  $I^t$ .

#### 3.1.2 Silhouette Prediction

We predict the silhouette of the person in the target pose given the input source triplet: source pose map, silhouette,



**Figure 4.** *SilNet* predicts the silhouette mask in the target pose.

and garment label. It is designed to learn the shape deformation as a function of the pose change:

$$\mathbf{S}^t = f^{\text{Sil}}(\mathbf{P}^t | \{\mathbf{P}^s, \mathbf{S}^s, \mathbf{G}^s\}). \quad (2)$$

We use a neural network called *SilNet* to learn this function. It has two encoders and one decoder, as shown in Fig. 4. One encoder encodes the spatial relationship of the body and silhouette from the source triplet, which is used to condition the silhouette generation of the target pose by mixing their latent codes. The garment labels  $\mathbf{G}^s$  provides an additional spatial cue to control the deformation, *i.e.*, pixels that do not belong to garment (*i.e.*, skin) less likely undergo large deformation. The features extracted from the target pose at each level are passed to the counterpart of the decoder through skip connections. We train *SilNet* by minimizing the  $L_1$  distance of the predicted silhouette mask  $\mathbf{S}^t$  and the ground truth  $\mathbf{S}_{\text{gt}}^t$ :

$$L_{\text{Sil}} = \|\mathbf{S}^t - \mathbf{S}_{\text{gt}}^t\|_1. \quad (3)$$

Note that, as  $f^{\text{Sil}}$  does not take as input the source image  $\mathbf{I}^s$ , using synthetic data does not introduce the domain gap.

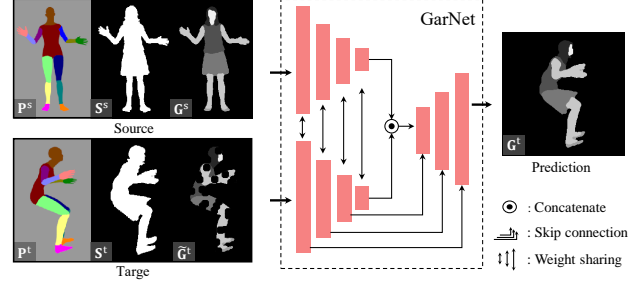
### 3.1.3 Garment Label Prediction

Given the source triplet and the predicted target silhouette, we predict the target garment labels  $\mathbf{G}^t$  that guide the generation of the target appearance. We take two steps.

First, we warp the source garment labels to produce the pseudo target garment labels,  $\tilde{\mathbf{G}}^t$ ,

$$\tilde{\mathbf{G}}^t(\mathbf{x}) = \mathbf{G}^s(\mathcal{W}_s^{-1}(\mathcal{W}_t(\mathbf{x}))), \quad (4)$$

where  $\mathcal{W}_s, \mathcal{W}_t : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  are the warping functions that transform a point in the source and target image  $\mathbf{x}$  to the UV coordinate of the body. The pseudo target garment label is incomplete because the body silhouette is a subset of the dressed body silhouette. Note that this first step, *i.e.*, producing  $\tilde{\mathbf{G}}^t$  by warping, only applies in the inference time, while in training time, we synthetically create the incomplete pseudo garment labels  $\tilde{\mathbf{G}}^t$  by removing the outside region of the body silhouette from the ground truth  $\mathbf{G}_{\text{gt}}^t$  and further removing some parts using random binary patches.



**Figure 5.** *GarNet* predicts the garment labels in the target pose.

Second, given the input triplet and the predicted target silhouette, we complete the full target garment labels  $\mathbf{G}^t$ :

$$\mathbf{G}^t = f^{\text{Gar}}(\tilde{\mathbf{G}}^t | \mathbf{P}^t, \mathbf{S}^t, \{\mathbf{P}^s, \mathbf{S}^s, \mathbf{G}^s\}). \quad (5)$$

We design a neural network called *GarNet* to learn the target garment label completion. It consists of a Siamese encoder and a decoder, as shown in Fig. 5. The Siamese encoder encodes the spatial relationship from both source and target triplets. A decoder completes the garment labels by classifying every pixel in the target silhouette. Similar to *SilNet*, we use skip connections to facilitate the target feature transform. We train *GarNet* by minimizing the following loss:

$$L_{\text{Gar}} = \|\mathbf{G}^t - \mathbf{G}_{\text{gt}}^t\|_1. \quad (6)$$

$f^{\text{Gar}}$  does not take as input the source image  $\mathbf{I}^s$  where using synthetic data does not introduce the domain gap.

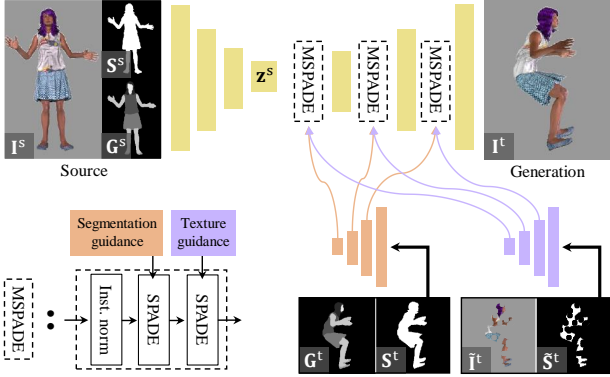
### 3.1.4 Foreground Rendering

We synthesize the foreground person image in a target pose given the predicted target garment label and the source image triplet: source image, silhouette, and garment label. Similar to the garment label completion in Sec. 3.1.3, we generate the pseudo target image  $\tilde{\mathbf{I}}^t$  and its silhouette  $\tilde{\mathbf{S}}^t$  using the UV coordinate transformation of  $\mathcal{W}_s$  and  $\mathcal{W}_t$  in inference time, while synthetically create the incomplete  $\tilde{\mathbf{I}}^t$  and  $\tilde{\mathbf{S}}^t$  from the ground truth  $\mathbf{I}_{\text{gt}}^t$  and  $\mathbf{S}_{\text{gt}}^t$  in training time.

We learn a function that can render the full target foreground image:

$$\mathbf{I}^t = f^{\text{Render}}(\tilde{\mathbf{I}}^t, \tilde{\mathbf{S}}^t | \mathbf{S}^t, \mathbf{G}^t, \{\mathbf{I}^s, \mathbf{S}^s, \mathbf{G}^s\}). \quad (7)$$

We design a neural network called *RenderNet* to learn this function. As shown in Fig. 6, *RenderNet* encodes the spatial relation  $\mathbf{z}^s$  of the source image triplet, and mixes the latent representations from the target. We use two encoders to extract the features of the target garment label  $\mathbf{G}^t$  and pseudo target image  $\tilde{\mathbf{I}}^t$  where  $\mathbf{S}^t$  and  $\tilde{\mathbf{S}}^t$  are combined with them. We condition these features at each level of the decoder using spatially adaptive normalization blocks [41, 34] to guide the network to be aware of the subject’s silhouette, and garment and texture style in the target pose.



**Figure 6.** *RenderNet* synthesizes the image of a person in the target pose.

We train *RenderNet* by minimizing the following loss:

$$L_{Render} = L_{rec} + \lambda_1 L_{VGG} + \lambda_2 L_{CX} + \lambda_3 L_{cAdv} + \lambda_4 L_{KL},$$

where the weight  $\lambda_i$  are empirically chosen that all the losses have comparable scale.

**Reconstruction Loss.**  $L_{rec}$  measures the per-pixel errors between the synthesized image  $\mathbf{I}^t$  and the ground truth  $\mathbf{I}_{gt}^t$ :

$$L_1 = \|\mathbf{I}^t - \mathbf{I}_{gt}^t\|_1.$$

**VGG Loss.** Beyond the low-level constraints in the RGB space,  $L_{VGG}$  measures the image similarity in the VGG feature space [20] which is effective in generating natural and smooth person image proven by existing works [36, 11, 50]:  $L_{VGG} = \sum_{i=1}^4 \|\text{VGG}_i(\mathbf{I}^t) - \text{VGG}_i(\mathbf{I}_{gt}^t)\|_1$ , where  $\text{VGG}_i(\cdot)$  maps an image to the activation of the conv-i-2 layer of VGG-16 network [48].

**Contextual Loss.**  $L_{CX}$  measures the similarity of two set of features considering global image context:  $L_{CX} = -\log(g(\text{VGG}_3(\mathbf{I}^t), \text{VGG}_3(\mathbf{I}_{gt}^t)))$ , where  $g(\cdot, \cdot) \in [0, 1]$  denotes the similarity metric of the matched features based on the normalized cosine distance [35]. Existing work [36] proved that combining  $L_{CX}$  with  $L_{VGG}$  further helps to preserve the style patterns in the generated image in a semantically meaningful way, *i.e.*, less distorted facial structure.

**Adversarial loss.** We employ the conditional adversarial loss  $L_{cAdv}$  [37] with a discriminator conditioned on garment labels to classify the synthesized image into real or fake, *i.e.*,  $\{\mathbf{I}_{gt}^t, \mathbf{G}_{gt}^t\}$  is real and  $\{\mathbf{I}^t, \mathbf{G}_{gt}^t\}$  is fake. Here, we use the PatchGAN discriminator [18].

**KL divergence.**  $L_{KL}$  is to enforce the latent space  $\mathbf{z}^s$  to be close to a standard normal distribution [25, 23].

### 3.2. Consistent Human Animation Creation

With the learned pose transfer network, it is possible to generate the shape and appearance given a target pose map at each time instant. However, it makes independent prediction for each pose, which leads to unrealistic jittery animation. Instead, we build a unified representation of appearance and its labels that provide a consistent guidance across

different poses, which enforces the network to predict temporally coherent appearance and shape.

We construct the garment labels  $\mathbf{L}$  and textures  $\mathbf{A}$  that remain constant in UV coordinates by warping the garment label and appearance of an image, *i.e.*,  $\mathbf{L}(\mathbf{x}) = \mathbf{G}(\mathcal{W}^{-1}(\mathbf{x}))$  and  $\mathbf{A}(\mathbf{x}) = \mathbf{I}(\mathcal{W}^{-1}(\mathbf{x}))$ . These UV representations ( $\mathbf{L}$  and  $\mathbf{A}$ ) cannot be completed from a single view input image because of occlusion. To complete the UV representations, we use the multiview images synthesized from the rendered 3D human model of which texture is predicted by the learned pose transfer network. This set of generated images are used to incrementally complete the UV representations as shown in Fig. 7-(left).

In practice, we generate multiview images by synthesizing the SMPL model at the T pose from six views: front, back, left, right, top and bottom views. We assume that the source image is taken from the frontal view. The back view is generated by applying front-back symmetry assumption [38, 12, 56] as shown in Fig. 7-(right).

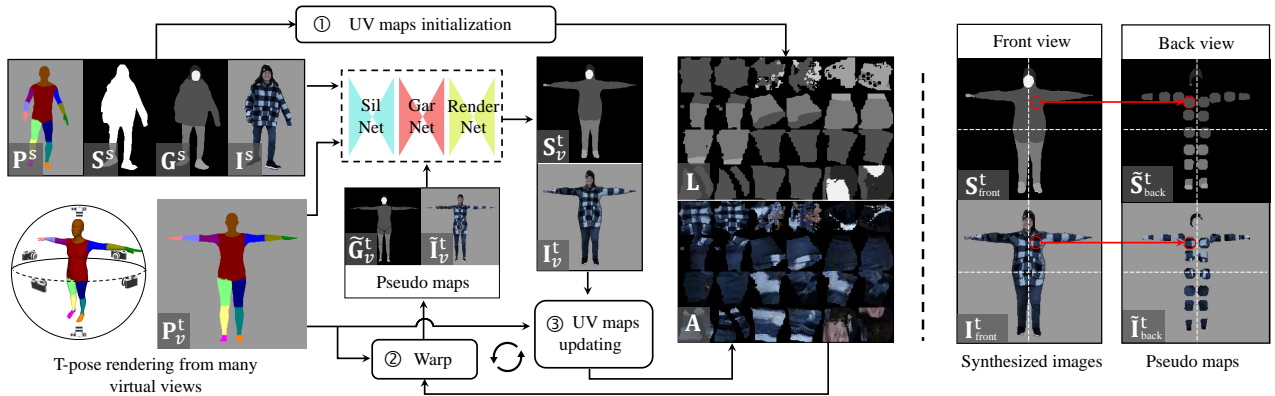
In the inference phase, this unified UV representation allows us to consistently generate the pseudo garment labels  $\tilde{\mathbf{G}}^t(\mathbf{x}) = \mathbf{L}(\mathcal{W}_t(\mathbf{x}))$  and appearance  $\tilde{\mathbf{I}}^t(\mathbf{x}) = \mathbf{A}(\mathcal{W}_t(\mathbf{x}))$  given a target pose by transforming the SMPL T-pose to the target pose. This pseudo representations provide an incomplete yet strong guidance to the pose transfer network to complete the target foreground.

In order to have both foreground and background in the animation, we segment the foreground from the source image using  $\mathbf{S}^s$  and apply an inpainting method [60] to the background. We then composite our synthesized human animation with the background.

## 4. Implementation Details

We train the proposed *SilNet*, *GarNet*, and *RenderNet* separately in a fully supervised way using only 3D people synthetic dataset [43] which is described in Sec 3.1. For training, we set the parameters of  $\lambda_1 = 0.5$ ,  $\lambda_2 = 0.1$ ,  $\lambda_3 = 0.01$ ,  $\lambda_4 = 10$  and use the Adam optimizer [22] ( $lr = 1 \times 10^{-3}$  and  $\beta = 0.5$ ). After training, no further fine-tuning on the testing scene is required. For the pose map  $\mathbf{P}$  and garment label map  $\mathbf{S}$ , we convert them to rgb and gray scale images for the network input.

In inference time, we obtain  $\mathbf{S}^s$  and  $\mathbf{G}^s$  using person segmentation [6] and fashion segmentation [13]. For  $\mathbf{P}^s$ , we fit a 3D body model [30] to an image using recent pose estimator [24] and render the parts label onto the image where we follow the same color coding as synthetic data [43]. We generate a sequence of body poses  $\{\mathbf{P}_i^t\}_{i=1}^N$  by animating the 3D body model using recent motion archive [33], where we represent the  $z$ -directional motion as scale variation [21] with weak-perspective camera projection, and rendering the pose map from each body pose similarly to  $\mathbf{P}^s$ . The image resolution is  $256 \times 256$ , and UV maps are  $512 \times 768$ .



**Figure 7.** We reconstruct the complete UV maps of the garment labels and textures, *i.e.*,  $L$  and  $A$ , in an incremental manner. (Left) We first initialize these maps by warping the pixels in the source image, *i.e.*,  $I^s$  and  $S^s$ , to the UV maps. We further update the UV maps by combining the synthesized images of a person in a T pose captured from six virtual views. For each virtual view  $v$ , we create the pseudo images, *i.e.*,  $\tilde{G}_v^t$  and  $\tilde{I}_v^t$ , from the previously updated UV maps. (Right) Only for the back view, we construct  $\tilde{G}_v^t$  and  $\tilde{I}_v^t$  by sampling the patches from the synthesized images in the frontal view with the front-back symmetry assumption where the face regions are removed.

## 5. Experiments

In order to evaluate our approach, we collect eight sequences of the subjects in various clothing and motions from existing works [46, 59, 28, 2, 14] and capture two more sequences which include a person with more complex clothing style and motion than others. Each sequence contains 50 to 500 frames. We use one frame in the sequence as source image and estimated body poses from the rest of frames using a pose estimator [24] as a target pose sequence.

**Baseline.** We compare our method with related works including *PG* [31], *SGAN* [52], *PPA* [63], *GFLA* [44], *NHRR* [45], *LWG* [28]. Note that all these methods except *LWG* are not designed to handle background. We compare all the methods on foreground synthesis and conduct an additional comparison with *LWG* on the full image synthesis including both foreground and background. For a fair comparison, we train all the methods except *LWG* on 3D people dataset [43]. For training, *LWG* requires a SMPL model which is not provided by the 3D people dataset. Since registering a SMPL model to each 3D model in the 3D people dataset may introduce fitting error, we use the pre-trained model provided by the authors, which are trained on the iPER dataset [28]. We also evaluate the methods with the pretrained models provided by the authors, which were trained on the Deep Fashion dataset [29] (see the supplementary material). In addition, we provide a qualitative comparison with *Photo Wake-Up* [56] which reconstructs a textured animatable 3D model from a single image.

### 5.1. Comparisons

**Qualitative Comparisons.** We show the qualitative comparison with the baselines on the foreground synthesis in

Fig. 8. Note that for the results in Fig. 8, all methods are trained on 3D people datasets (see the results of the baselines trained on the Deep Fashion dataset in the supplementary video). Our method significantly outperforms other baselines as preserving the facial identity, body shape and texture patterns of clothes over all the subjects with various challenging poses. Furthermore, compared to the baseline methods, our method generalizes better on the real data and achieves more realistic results that are close to the ground truth, although only synthetic data is used for training. We conduct a comparison with *LWG* on the full image synthesis, where our method can synthesize higher quality foreground as well as background, as shown in Fig. 9. Compared to *Photo Wake-Up* in Fig. 10, we can render the better textures on the right and back side of the person.

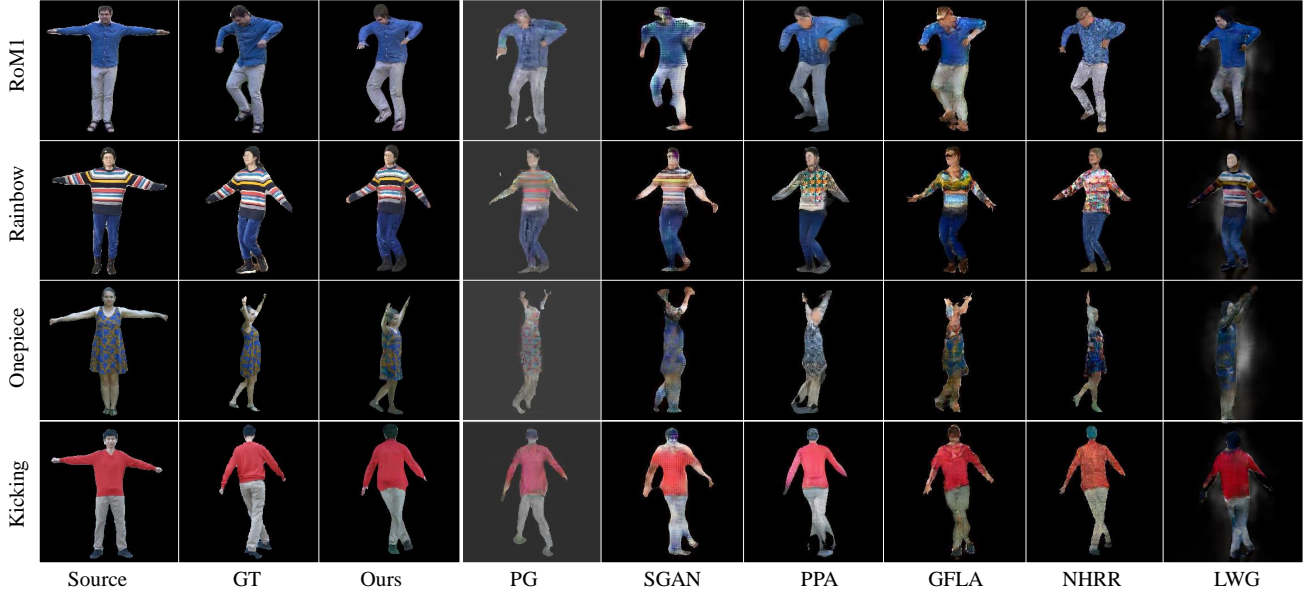
**Quantitative Comparisons.** We measure the quality on testing results with two metrics: LPIPS [61] and CS [35] where both metrics measure the similarity of the generated image with ground truth based on the deep features, and CS can handle the non-aligned two images. As shown in Table 3, our method outperforms all baseline methods over almost all the sequences in LPIPS and CS. In Kicking, our method performs the second best in LPIPS metric mainly due to the misalignment with the ground truth originated from the pose estimation error. In Fig. 11, we measure temporal stability of the synthesized animations with the standard deviation of the LPIPS scores with respect to all the frames, where our results show the best temporal stability.

### 5.2. Ablation Study

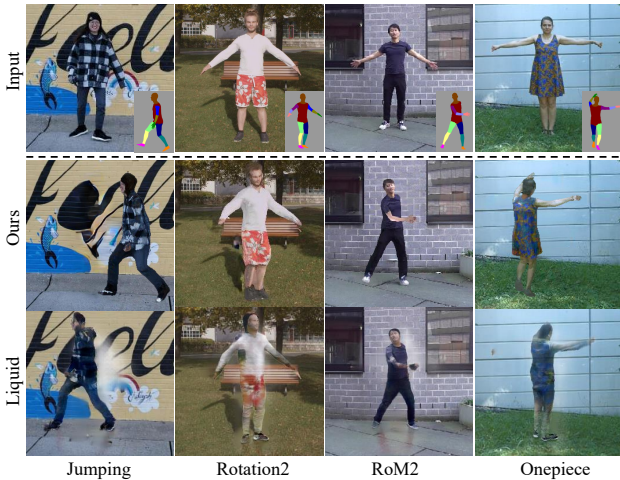
We study the importance of each module in our pose transfer pipeline where we term “S”, “G”, and “R” as *SilNet*, *GarNet*, and *RenderNet*, and our full model as *SGR*.

	Maskman	Rainbow	RoM1	RoM2	Jumping	Kicking	Onepiece	Checker	Rotation1	Rotation2	Average
PG (3P)	3.45 / 4.46	3.28 / 4.58	3.38 / 4.37	3.30 / 4.57	3.87 / 4.68	2.98 / 4.36	3.26 / 4.63	3.03 / 4.45	3.67 / 4.41	2.95 / 4.06	2.93 / 4.45
SGAN (3P)	1.93 / 2.97	1.61 / 3.04	1.62 / 2.94	1.60 / 3.02	2.27 / 3.12	1.56 / 2.98	1.82 / 3.06	1.53 / 3.03	1.56 / 3.01	1.65 / 2.84	1.71 / 3.00
PPA (3P)	1.88 / 2.89	1.62 / 2.95	1.40 / 2.82	1.66 / 3.00	2.43 / 3.03	1.33 / 2.88	1.86 / 2.95	1.49 / 2.88	1.38 / 2.84	1.40 / 2.69	1.64 / 2.89
GFLA (3P)	1.90 / 2.92	1.59 / 3.05	1.53 / 2.91	1.71 / 2.95	2.11 / 3.06	1.42 / 2.97	1.60 / 3.03	1.64 / 2.97	1.64 / 2.93	1.65 / 2.80	1.68 / 2.96
NHHR (3P)	1.65 / 2.81	1.61 / 2.94	1.49 / 2.80	1.41 / 2.88	1.99 / 3.01	<b>1.00</b> / 2.85	1.78 / 2.98	1.65 / 2.94	1.39 / 2.88	1.67 / 2.75	1.56 / 2.88
LWG	2.66 / 3.54	2.16 / 3.57	2.17 / 3.49	2.24 / 3.73	4.11 / 3.49	2.42 / 3.57	2.31 / 3.65	2.40 / 3.57	2.14 / 3.64	2.20 / 3.48	2.48 / 3.57
Ours	<b>1.54 / 2.27</b>	<b>1.24 / 2.38</b>	<b>1.25 / 2.24</b>	<b>1.38 / 2.36</b>	<b>1.87 / 2.53</b>	<b>1.08 / 2.19</b>	<b>1.23 / 2.32</b>	<b>1.09 / 2.24</b>	<b>1.00 / 2.19</b>	<b>1.12 / 2.16</b>	<b>1.28 / 2.29</b>

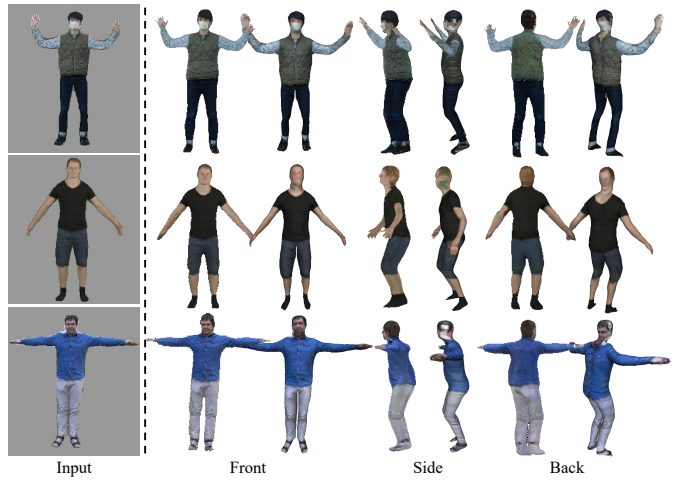
**Table 1.** Quantitative results with LPIPS (left, scale:  $\times 10^{-1}$ ) and CS (right) where the lower is the better.



**Figure 8.** Qualitative comparisons of our approach with other baseline methods. See our supplementary video for more results.



**Figure 9.** Qualitative comparison with LWG on the input images with background. The target pose is shown as inset. See our supplementary video for more results.



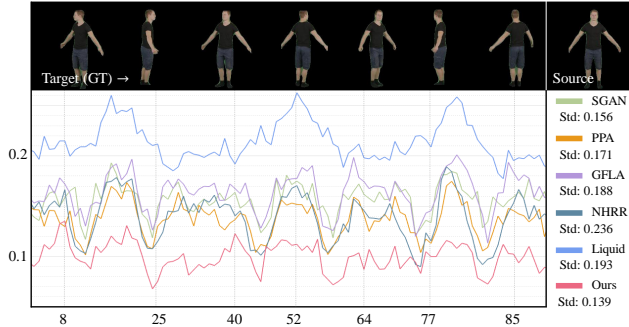
**Figure 10.** Qualitative comparison of ours (left) with Photo Wake-Up (right).

- 1) We analyze the effectiveness of our modular network by removing each from SGR where the intermediate results are also removed from the entire pipeline:  $R$ ,  $SR$ , and  $GR$ .
- 2) We evaluate the impact of using silhouette mask and garment label from the source by removing each of them from

- the entire pipeline:  $SGR-S^s$  and  $SGR-M^s$ .
- 3) We investigate the improvement factor on the *RenderNet*:  $SGR-z^s$ ,  $SGR-\tilde{I}^t$ , and  $SGR-L_{KL}$ . For  $SGR-L_{KL}$ , we represent the latent space with fully connected layers. On top of that, we investigate the impact of reconstructing a complete

	R	GR	SR	SGR-M <sup>s</sup>	SGR-S <sup>s</sup>	SGR-z <sup>s</sup>	SGR-L <sub>KL</sub>	SGR-I <sup>t</sup>	SGR-A	SGR+2view	SGR+4view	SGR (full)
LPIPS	0.148	0.147	0.133	0.135	0.141	0.132	1.304	0.162	0.136	0.125	<b>0.124</b>	0.128
CS	2.354	2.352	2.311	2.322	2.342	2.308	2.301	2.335	2.310	2.281	<b>2.274</b>	2.292

**Table 2.** Quantitative results of our ablation study. We denote our complete model with a single image as input as SGR(full).



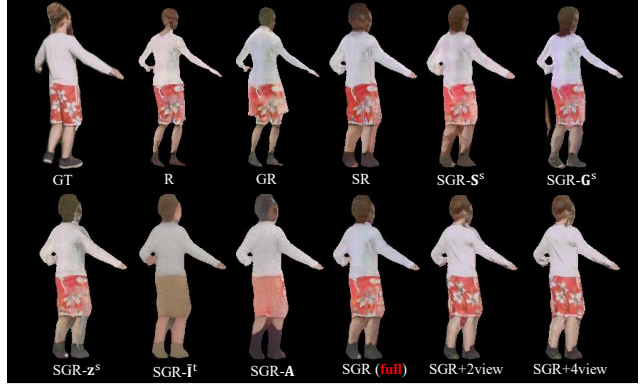
**Figure 11.** The accuracy graph for the entire frames of a video.  $x$ -axis and  $y$ -axis represent time instance and LPIPS, respectively.

UV map:  $SGR-A$ . In this case, we create the pseudo target image  $I^t$  by directly warping the source image to the target. 4) Finally, we show that our method is readily extendable to the multiview setting by unifying all the pixels from multiple images in the coherent UV maps. For this, we choose two or four frames from the testing videos that include salient body sides, *e.g.*, front, back, right, and left:  $SGR+2view$  and  $SGR+4view$ .

We summarize the results of our ablation study in Table 2 and the qualitative results are shown in Fig. 12. Separating the silhouette prediction module from rendering network brings out notable improvement, and the predicted garment labels  $G^t$  further improve the results, *e.g.*, clear boundary between different classes. Without the garment labels from the source  $G^s$  the performance is largely degraded due to the misclassified body parts. Conditioning the style code  $z^s$  from the source improves the generation quality, *e.g.*, seamless inpainting. Conditioning the pseudo images  $I^t$  warped from the coherent UV maps  $A$  plays the key role to preserve the subject’s appearance in the generated image. Leveraging multiview images better can preserve the clothing texture, *e.g.*, the flower patterns in the subject’s half pants.

### 5.3. User Study

We evaluate the qualitative impact of our method by a user study with 25 videos where each video shows a source image and animated results. Four videos compare our method to LWG on the scenes with a background. 21 videos are without background (15 of them compare our method to randomly-chosen four baselines, excluding ground truth, and 6 videos include ground truth). 47 people participated in total. In 84.3% and 93% of the cases, our method was found to produce the most realistic animations in the settings with and without ground truth, respectively. Moreover, these numbers strongly correlate with the identity-



**Figure 12.** Qualitative results of our ablation study. See our supplementary video for more results.

preserving properties of our method. Finally, our technique preserves the background better compared to LGW, in the opinion of respondents (96.8% of the answers). The user study shows that our method significantly outperforms the state of the arts in terms of synthesis quality, temporal consistency and generalizability. Also, our results were often ranked as more realistic than the ground truth videos. The full results will be shown in the supplementary material.

### 5.4. Limitations

Our method has several limitations. Although the unified representation of appearance and its labels allow us to synthesize temporally consistent results, it prevents from generating realistic physical effects such as pose-dependent clothing secondary motion, wrinkles, shading, and view-dependent lighting. Because of non-end-to-end nature of our method, the errors from the pre-processing step, *e.g.*, person and garment segmentation, and pose estimation, cannot be corrected by our pose transfer network.

## 6. Conclusion

We introduce a new pose transfer framework to animate humans from a single image. We addressed the core domain gap challenge for the testing data in the wild by designing a new compositional pose transfer network that predicts silhouette, garment labels, and textures in series, which are learned from synthetic data. In inference time, we reconstruct coherent UV maps by unifying the source and synthesized images, and utilize these UV maps to guide the network to create coherent human animation. The evaluation on diverse subjects demonstrates that our framework works well on the unseen data without any fine-tuning and

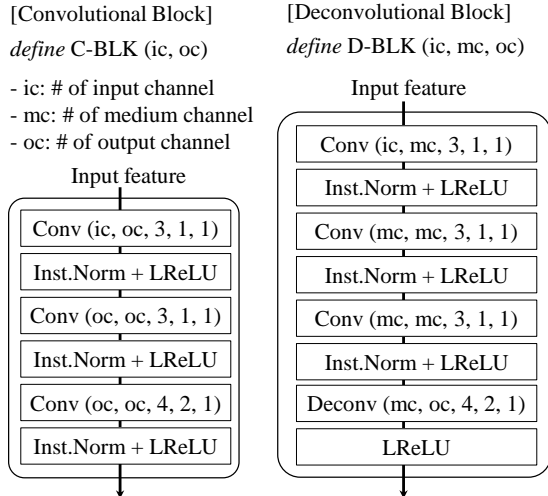


preserves the identity and texture of the subject as well as background in a temporally coherent way, showing a significant improvement over the state-of-the-arts.

## References

- [1] Badour AlBahar and Jia-Bin Huang. Guided image-to-image translation with bi-directional feature transformation. In *ICCV*, 2019. 2
- [2] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *CVPR*, 2018. 6, 11
- [3] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *ICCV*, 2019. 3
- [4] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018. 2
- [5] Guha Balakrishnan, Amy Zhao, Adrian V Dalca, Fredo Durand, and John Guttag. Synthesizing images of humans in unseen poses. In *CVPR*, 2018. 2
- [6] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019. 5
- [7] Wonmin Byeon, Qin Wang, Rupesh Kumar Srivastava, and Petros Koumoutsakos. Contextvp: Fully context-aware video prediction. In *ECCV*, 2018. 2
- [8] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *ICCV*, pages 5933–5942, 2019. 1, 2
- [9] Xu Chen, Jie Song, and Otmar Hilliges. Unpaired pose guided human image generation. In *CVPRW*, 2019. 2
- [10] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bochao Wang, Hanjiang Lai, Jia Zhu, Zhiting Hu, and Jian Yin. Towards multi-pose guided virtual try-on network. In *ICCV*, 2019. 2
- [11] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *CVPR*, 2018. 2, 5
- [12] Valentin Gabeur, Jean-Sébastien Franco, Xavier Martin, Cordelia Schmid, and Gregory Rogez. Moulding humans: Non-parametric 3d human shape estimation from single images. In *ICCV*, 2019. 5
- [13] Ke Gong, Xiaodan Liang, Dongyu Zhang, Xiaohui Shen, and Liang Lin. Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing. In *CVPR*, pages 932–940, 2017. 5
- [14] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *CVPR*, 2020. 6, 11
- [15] Xintong Han, Xiaojun Hu, Weilin Huang, and Matthew R Scott. Clothflow: A flow-based model for clothed person generation. In *ICCV*, 2019. 2
- [16] Zeng Huang, Yuanlu Xu, Christoph Lassner, Hao Li, and Tony Tung. Arch: Animatable reconstruction of clothed humans. In *CVPR*, 2020. 3
- [17] Takeo Igarashi, Tomer Moscovich, and John F Hughes. As-rigid-as-possible shape manipulation. *SIGGRAPH*. 1
- [18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 5
- [19] Alec Jacobson, Ilya Baran, Jovan Popovic, and Olga Sorkine. Bounded biharmonic weights for real-time deformation. *ACM Trans. Graph.*, 2011. 1
- [20] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 5
- [21] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 5
- [22] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, 2014. 5
- [23] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2013. 5
- [24] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *ICCV*, 2019. 5, 6
- [25] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 1951. 5
- [26] Lingjie Liu, Weipeng Xu, Marc Habermann, Michael Zollhofer, Florian Bernard, Hyeonwoo Kim, Wenping Wang, and Christian Theobalt. Neural human video rendering by learning dynamic textures and rendering-to-video translation. *IEEE Transactions on Visualization and Computer Graphics*, 2020. 1, 2
- [27] Lingjie Liu, Weipeng Xu, Michael Zollhofer, Hyeonwoo Kim, Florian Bernard, Marc Habermann, Wenping Wang, and Christian Theobalt. Neural rendering and reenactment of human actor videos. *SIGGRAPH*, 2019. 1
- [28] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In *ICCV*, 2019. 2, 6, 11
- [29] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016. 1, 2, 6
- [30] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-person linear model. *SIGGRAPH*, 2015. 5
- [31] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *NeurIPS*, 2017. 2, 3, 6
- [32] Liqian Ma, Zhe Lin, Connelly Barnes, Alexei A Efros, and Jingwan Lu. Unselfie: Translating selfies to neutral-pose portraits in the wild. *ECCV*, 2020. 2
- [33] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *ICCV*, 2019. 5
- [34] Arun Mallya, Ting-Chun Wang, Karan Sapra, and Ming-Yu Liu. World-consistent video-to-video synthesis. *ECCV*, 2020. 4, 13
- [35] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. In *ECCV*, 2018. 5, 6

- [36] Yifang Men, Yiming Mao, Yuning Jiang, Wei-Ying Ma, and Zhouhui Lian. Controllable person image synthesis with attribute-decomposed gan. In *CVPR*, 2020. 2, 3, 5
- [37] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 5
- [38] Ryota Natsume, Shunsuke Saito, Zeng Huang, Weikai Chen, Chongyang Ma, Hao Li, and Shigeo Morishima. Siclope: Silhouette-based clothed people. In *CVPR*, 2019. 5
- [39] Assaf Neuberger, Eran Borenstein, Bar Hilleli, Eduard Oks, and Sharon Alpert. Image based virtual try-on network from unpaired data. In *CVPR*, 2020. 2
- [40] Natalia Neverova, Riza Alp Guler, and Iasonas Kokkinos. Dense pose transfer. In *Proceedings of the European conference on computer vision (ECCV)*, pages 123–138, 2018. 2
- [41] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 4, 13
- [42] Albert Pumarola, Antonio Agudo, Alberto Sanfeliu, and Francesc Moreno-Noguer. Unsupervised person image synthesis in arbitrary poses. In *CVPR*, 2018. 2
- [43] Albert Pumarola, Jordi Sanchez, Gary Choi, Alberto Sanfeliu, and Francesc Moreno-Noguer. 3DPeople: Modeling the Geometry of Dressed Humans. In *ICCV*, 2019. 3, 5, 6
- [44] Yurui Ren, Xiaoming Yu, Junming Chen, Thomas H Li, and Ge Li. Deep image spatial transformation for person image generation. In *CVPR*, 2020. 2, 6
- [45] Kripasindhu Sarkar, Dushyant Mehta, Weipeng Xu, Vladislav Golyanik, and Christian Theobalt. Neural re-rendering of humans from a single image. 2, 3, 6
- [46] Soshi Shimada, Vladislav Golyanik, Weipeng Xu, and Christian Theobalt. Physcap: Physically plausible monocular 3d motion capture in real time. *SIGGRAPH Asia*, 2020. 6, 11
- [47] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *NeurIPS*, 2019. 2
- [48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICML*, 2014. 5
- [49] Sijie Song, Wei Zhang, Jiaying Liu, and Tao Mei. Unsupervised person image generation with semantic parsing transformation. In *CVPR*, pages 2357–2366, 2019. 2
- [50] Hao Tang, Song Bai, Li Zhang, Philip HS Torr, and Nicu Sebe. Xinggan for person image generation. *ECCV*, 2020. 5
- [51] Hao Tang, Dan Xu, Gaowen Liu, Wei Wang, Nicu Sebe, and Yan Yan. Cycle in cycle generative adversarial networks for keypoint-guided image generation. In *ICMM*, 2019. 2
- [52] Hao Tang, Dan Xu, Yan Yan, Jason J Corso, Philip HS Torr, and Nicu Sebe. Multi-channel attention selection gans for guided image-to-image translation. *CVPR*, 2020. 6
- [53] Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. Few-shot video-to-video synthesis. *NeurIPS*, 2019. 2
- [54] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *NeurIPS*, 2018. 2
- [55] Tsun-Hsuan Wang, Yen-Chi Cheng, Chieh Hubert Lin, Hwann-Tzong Chen, and Min Sun. Point-to-point video generation. In *ICCV*, 2019. 2
- [56] Chung-Yi Weng, Brian Curless, and Ira Kemelmacher-Shlizerman. Photo wake-up: 3d character animation from a single photo. In *CVPR*, 2019. 3, 5, 6
- [57] Ceyuan Yang, Zhe Wang, Xinge Zhu, Chen Huang, Jianping Shi, and Dahua Lin. Pose guided human video generation. In *ECCV*, 2018. 2
- [58] Lingbo Yang, Pan Wang, Chang Liu, Zhanning Gao, Peiran Ren, Xinfeng Zhang, Shanshe Wang, Siwei Ma, Xiansheng Hua, and Wen Gao. Towards fine-grained human pose transfer with detail replenishing network. *TIP*, 2020. 2
- [59] Jae Shin Yoon, Kihwan Kim, Orazio Gallo, Hyun Soo Park, and Jan Kautz. Novel view synthesis of dynamic scenes with globally coherent depths from a monocular camera. 2020. 6, 11
- [60] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *ICCV*, 2019. 5
- [61] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6
- [62] Yipin Zhou, Zhaowen Wang, Chen Fang, Trung Bui, and Tamara Berg. Dance dance generation: Motion transfer for internet videos. In *ICCVW*, 2019. 2
- [63] Zhen Zhu, Tengting Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *CVPR*, 2019. 2, 6



**Figure 13.** Description of our convolutional and deconvolutional blocks. The convolutional (Conv) and deconvolutional layers (Deconv) take parameters including the number of input channels, the number of output channels, filter size, stride, and the size of zero padding. We use 0.2 for the LeakyReLU (LReLU) coefficient.

This supplementary material provides additional implementation details of our compositional pose transfer network (Sec. A) and more results (Sec. B). In the supplementary video, we included the full results of the qualitative comparison, ablation study, more results, and the description of our overall pipeline.

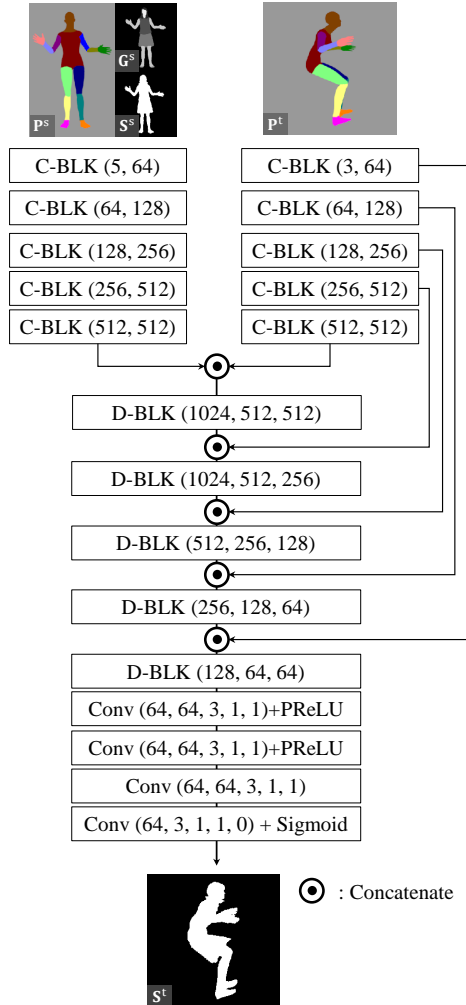
## A. Additional Implementation Details

We provide the implementation details of each modular function in our compositional pose transfer network. Fig. 14 describes the *SilNet* architecture which takes as input source triplet of the pose map, garment labels, and silhouette, and target pose map, and predicts the silhouette mask in the target pose. Fig. 15 describes the architecture of our *GarNet* that takes as input source triplet of the pose map, silhouette, and garment labels, and target triplet of the pose map, predicted silhouette, and pseudo garment labels, and predicts the complete garment labels. In Fig. 19, we show the details of our *RenderNet* which takes as input source triplet of image, silhouette mask, and garment labels, target silhouette and garment labels, and target pseudo image and its mask, and generates the person image.

## B. More Results

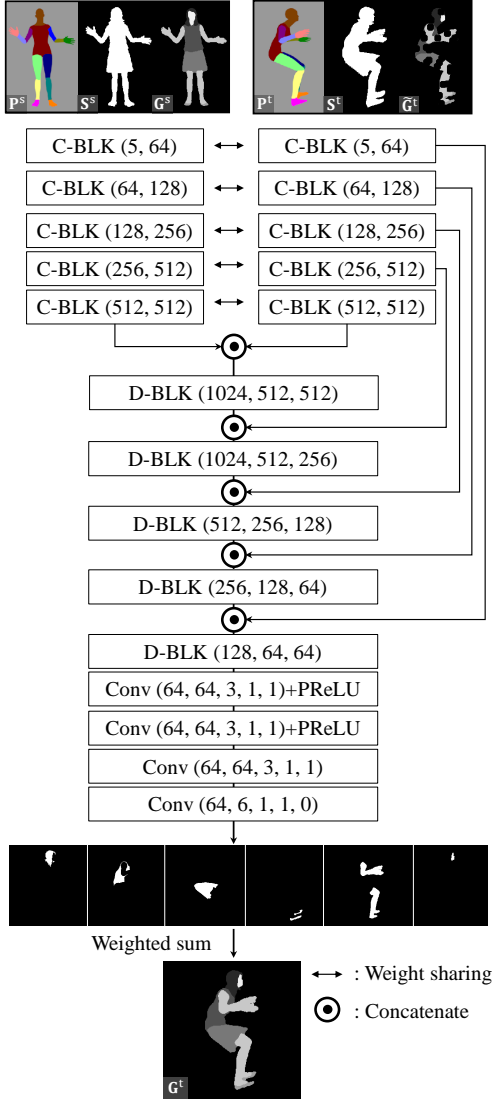
### B.1. Additional Dataset Description

We provide more details of the videos used for the evaluation. In order to evaluate our approach, we use eight sequences of the subjects in various clothing and motions from existing works [46, 59, 28, 2, 14] and capture two



**Figure 14.** The details of our *SilNet* implementation where C-BLK and D-BLK are described in Fig. 13. Conv and Deconv take as input parameters of (the number of input channels, the number of output channels, filter size, stride, the size of zero padding). We use 0.2 for the LeakyReLU (LReLU) coefficient.

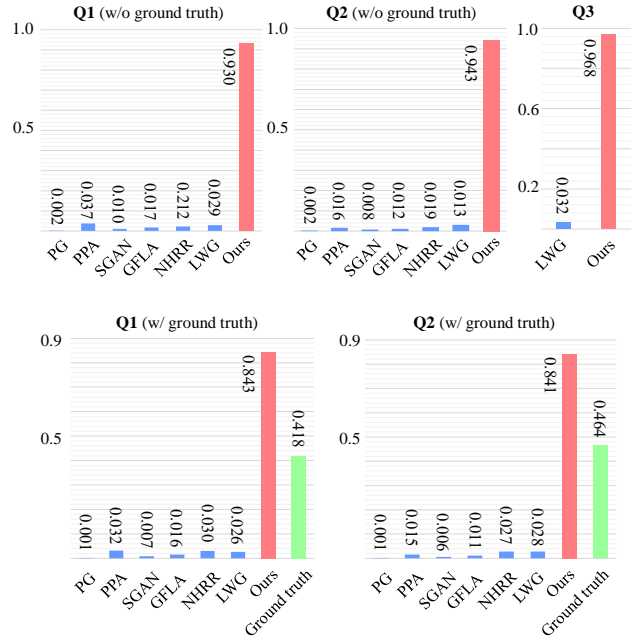
more sequences which include a person with more complex clothing style and movement than others. *RoM1* and *RoM2*: Two men show their range of motion with various poses [46]. *Jumping* [59]: A woman in a black and white coat jump from one side to another. *Kicking* and *One-piece* [14]: A man and woman take the motion of kicking and dancing where the woman is wearing a unique one-piece. *Checker* [28]: A man in shirts with checkered pattern swings his hands. *Rotation1* and *Rotation2* [2]: Two A-posed men rotate their body. *Maskman*: A man wearing a facial mask shows his various motion. *Rainbow*: A woman in a sweater with rainbow pattern turns her body with dancing motion.



**Figure 15.** The details of our *GarNet* implementation where C-BLK and D-BLK are described in Fig. 13. Conv and Deconv take as input parameters of (the number of input channels, the number of output channels, filter size, stride, the size of zero padding). We use 0.2 for the LeakyReLU (LReLU) coefficient.

## B.2. User Study Results

In our user study, three questions are asked: Q1: Which video looks most realistic including temporal coherence? Q2: Which video preserves the identity best including facial details, shape, and overall appearance? Q3: In which video, the background is preserved better across the frames (only for the case of scenes with background)? For each method, we measure the performance based on the number of entire votes divided by the number of participants and the number of occurrence in the questionnaires. The full results are shown in Fig. 16. The first question was answered in



**Figure 16.** The full results of the user study where  $x$ -axis represents the number of votes for the associated method which is normalized by the number of participants and the number of occurrence in the questionnaires. Q1, Q2, and Q3 represent the question type. Our results were often ranked as more realistic than the real videos because they involve a significant boundary noise from the person segmentation error while our method produces the human animation with clean boundary.

84.3% and 93.0% of the cases in favour of our method with and without the ground truth sequence, respectively, and the second question 84.1% and 94.2%. In the third question, the background is preserved better in our method than LWG in 96.8 % of the answers. The results show that our method outperforms other state of the art, and our animations are in many cases qualitatively comparable to real videos of the subjects. The choice between a real video and our animation did not fall easy because the ground-truth video often contains noisy boundary originated from the person segmentation error while the generated person images from our method shows the clear boundary.

## B.3. Additional Quantitative Results

We include the quantitative results which do not appear in the main paper. In Table 3, the performance of the baseline models that are pretrained from the DeepFashion (DF) dataset by the authors is summarized in the first chunk (from 2 to 6 row), ablation study in the second chunk (from 7 to 16), and application to the multiview data in the third chunk (from 17 to 18).

	Maskman	Rainbow	RoM1	RoM2	Jumping	Kicking	Onepiece	Checker	Rotation1	Rotation2	Average
PG (DF)	2.01 / 4.24	2.14 / 4.41	2.22 / 4.48	1.81 / 4.31	2.33 / 4.32	2.15 / 4.49	2.43 / 4.66	2.07 / 4.25	1.74 / 4.18	2.58 / 4.47	2.15 / 4.38
SGAN (DF)	2.33 / 3.96	2.39 / 4.22	2.50 / 4.16	2.12 / 4.22	2.63 / 4.09	2.49 / 4.29	2.67 / 4.25	2.34 / 3.99	1.89 / 3.93	2.74 / 4.22	2.43 / 4.13
PPA (DF)	2.84 / 3.76	2.70 / 3.80	2.78 / 3.91	2.65 / 3.97	2.89 / 3.87	2.88 / 3.94	3.21 / 4.05	2.26 / 3.76	2.26 / 3.75	3.01 / 3.77	2.74 / 3.86
GFLA (DF)	1.96 / 3.86	1.64 / 3.93	2.19 / 3.89	1.50 / 3.99	2.01 / 3.85	2.05 / 3.96	2.23 / 3.94	1.74 / 3.84	1.60 / 3.88	1.92 / 3.89	1.88 / 3.90
NHRH (DF)	1.71 / 2.96	1.89 / 3.06	1.82 / 3.07	1.56 / 3.03	2.06 / 3.03	1.68 / 3.11	2.16 / 3.16	1.48 / 2.94	1.80 / 3.02	2.77 / 3.11	1.89 / 3.05
R	1.64 / 2.31	1.48 / 2.43	1.41 / 2.30	1.53 / 2.44	2.00 / 2.54	1.16 / 2.18	1.36 / 2.34	1.41 / 2.31	1.22 / 2.31	1.62 / 2.33	1.48 / 2.35
GR	1.64 / 2.30	1.45 / 2.42	1.51 / 2.30	1.44 / 2.42	1.91 / 2.53	1.40 / 2.24	1.24 / 2.35	1.39 / 2.29	1.21 / 2.30	1.60 / 2.32	1.47 / 2.35
SR	1.57 / 2.26	1.30 / 2.42	1.31 / 2.24	1.41 / 2.37	1.89 / 2.54	1.17 / 2.20	1.24 / 2.33	1.11 / 2.24	1.05 / 2.23	1.25 / 2.22	1.33 / 2.31
SGR-S <sup>s</sup>	1.58 / 2.29	1.33 / 2.41	1.26 / 2.26	1.43 / 2.39	1.99 / 2.54	1.18 / 2.23	1.29 / 2.35	1.10 / 2.36	1.05 / 2.23	1.24 / 2.20	1.35 / 2.32
SGR-G <sup>s</sup>	1.66 / 2.30	1.38 / 2.39	1.31 / 2.32	1.48 / 2.35	1.89 / 2.51	1.18 / 2.23	1.31 / 2.40	1.31 / 2.31	1.19 / 2.28	1.42 / 2.30	1.41 / 2.34
SGR-I <sup>s</sup>	1.79 / 2.28	1.97 / 2.49	1.55 / 2.30	1.52 / 2.38	2.13 / 2.50	1.31 / 2.23	1.79 / 2.39	1.49 / 2.31	1.15 / 2.22	1.50 / 2.21	1.62 / 2.33
SGR-z <sup>s</sup>	1.57 / 2.27	1.31 / 2.40	1.25 / 2.26	1.42 / 2.38	1.90 / 2.52	1.15 / 2.19	1.29 / 2.31	1.11 / 2.22	1.05 / 2.19	1.24 / 2.23	1.32 / 2.30
SGR-L <sub>KL</sub>	1.54 / 2.27	1.25 / 2.38	1.27 / 2.25	1.40 / 2.38	1.88 / 2.55	1.13 / 2.19	1.25 / 2.32	1.09 / 2.24	1.04 / 2.20	1.15 / 2.19	1.30 / 2.30
SGR-A	1.59 / 2.28	1.28 / 2.40	1.31 / 2.26	1.40 / 2.38	1.86 / 2.51	1.23 / 2.21	1.32 / 2.33	1.14 / 2.25	1.15 / 2.23	1.28 / 2.20	1.36 / 2.31
SGR (full)	<b>1.54 / 2.27</b>	<b>1.24 / 2.38</b>	<b>1.25 / 2.24</b>	<b>1.38 / 2.36</b>	<b>1.87 / 2.53</b>	<b>1.08 / 2.19</b>	<b>1.23 / 2.32</b>	<b>1.09 / 2.24</b>	<b>1.00 / 2.19</b>	<b>1.12 / 2.16</b>	<b>1.28 / 2.29</b>
SGR+2view	1.50 / 2.25	1.22 / 2.38	1.21 / 2.23	1.33 / 2.36	1.80 / 2.51	1.15 / 2.17	1.20 / 2.31	1.07 / 2.23	0.97 / 2.16	1.06 / 2.14	1.25 / 2.28
SGR+4view	1.49 / 2.25	1.21 / 2.38	1.21 / 2.23	1.33 / 2.35	1.80 / 2.51	1.12 / 2.17	1.20 / 2.31	1.07 / 2.23	0.98 / 2.16	1.07 / 2.14	1.24 / 2.27

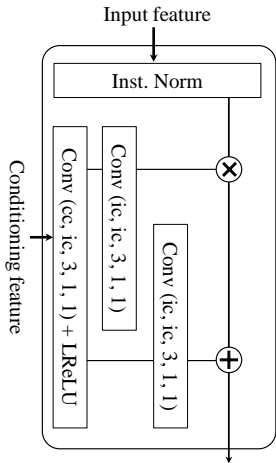
**Table 3.** Quantitative results with LPIPS (left, scale:  $10^{-1}$ ) and CS where the lower is the better. We denote the full model used for the comparison with other baseline methods as SGR (full).

[SPADE Block]

define S-BLK (ic, fc)

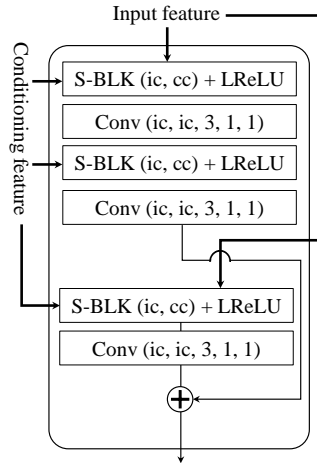
- ic: # of input feature channels

- cc: # of conditioning feature channels



[SPADE Residual Block]

define S-ResBLK (ic, cc)



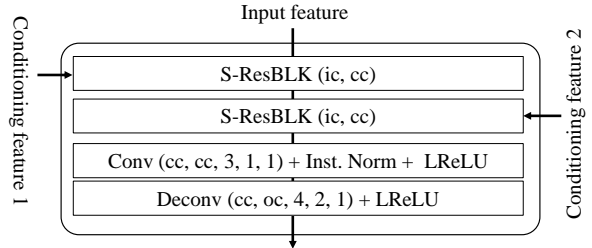
[Multi-SPADE Residual Block with Deconvolution]

define MS-ResBLK-D (ic, cc, oc)

ic: input feature channel

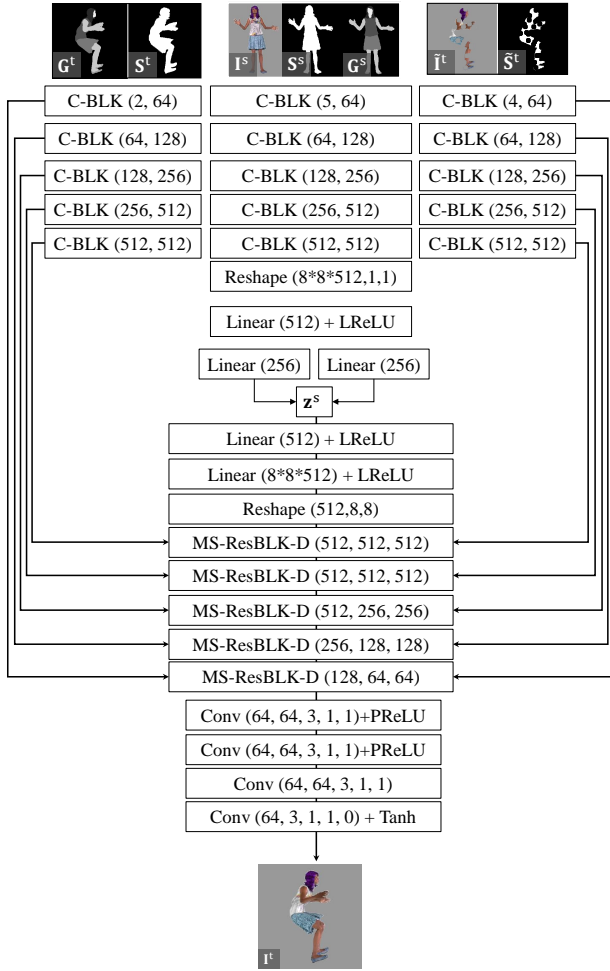
cc: conditioning feature channel

oc: output feature channel



**Figure 18.** The description of Multi-Spade blocks similar to [34] where the details of S-ResBLK is described in Fig. 17. Conv and Deconv take as input parameters of (the number of input channels, the number of output channels, filter size, stride, the size of zero padding). We use 0.2 for the LeakyReLU (LReLU) coefficient.

**Figure 17.** The description of SPADE and SPADE Residual blocks similar to [41]. Conv take as input parameters of (the number of input channels, the number of output channels, filter size, stride, the size of zero padding). We use 0.2 for the LeakyReLU (LReLU) coefficient.



**Figure 19.** The details of our *RenderNet* where C-BLK and D-BLK are described in Fig 13, and MS-ResBLK-D is in Fig. 18. Conv takes as input parameters of (the number of input channels, the number of output channels, filter size, stride, the size of zero padding). We use 0.2 for the LeakyReLU (LReLU) coefficient.