Pose Flow Learning From Person Images for Pose Guided Synthesis

Haitian Zheng, *Student Member, IEEE*, Lele Chen, *Student Member, IEEE*, Chenliang Xu, *Member, IEEE*, and Jiebo Luo, *Fellow, IEEE*

***Abstract*— Pose guided synthesis aims to generate a new image in an arbitrary target pose while preserving the appearance details from the source image. Existing approaches rely on either hard-coded spatial transformations or 3D body modeling. They often overlook complex non-rigid pose deformation or unmatched occluded regions, thus fail to effectively preserve appearance information. In this article, we propose a pose flow learning scheme that learns to transfer the appearance details from the source image without resorting to annotated correspondences. Based on such learned pose flow, we proposed GarmentNet and SynthesisNet, both of which use multi-scale feature-domain alignment for coarse-to-fine synthesis. Experiments on the DeepFashion, MVC dataset and additional real-world datasets demonstrate that our approach compares favorably with the state-of-the-art methods and generalizes to unseen poses and clothing styles.**

***Index Terms*— Pose guided synthesis, pose correspondence, optical flow learning.**

I. Introduction

POSE guided synthesis aims to generate a realistic person image that preserves the appearance details of the source image given an arbitrary target pose. As a central task in virtual reality [46], online garment retail [10], and game character rendering, realistic pose guided synthesis will have a crucial impact on numerous applications.

Despite the recent successes of conditional image synthesis [11], [41], pose guided synthesis still faces many unsolved challenges. Among them, the main challenge is the complex, part-independent pose deformation, with garment, from the source pose to an arbitrary target pose. As a result, models [4], [10], [22], [30] built on the plain U-Net [33] network structure often fail to generate precise details or textures due to the lack of a robust spatial alignment component.

Recently, several approaches [3], [29], [36], [45] have been proposed to address spatial alignment. Specifically, Siarohin *et al.* [36] apply deformable skip connections for spatial alignment. However, the oversimplified affine transformation on the predefined rectangles does not necessarily capture the non-rigid deformation. Different from Neverova *et al.* [29] and Wu *et al.* [45] resort to a pre-trained pose estimator, DensePose [1], to perform non-rigid alignment on 3D-model. Since such model-level alignment is not capable of handling occluded regions caused by drastic pose changes, inpainting is then applied to fill the occluded region. Nonetheless, the results are usually blurry in occluded regions.

A later work [3] relies on the combination of affine transformation and thin-plate splines (TPS) transformation to perform spatial alignment. However, the TPS transformation is inflexible to model the highly non-rigid human pose deformation. In addition, their matching module is trained on simplified synthetic transformations [32]. Therefore, the human pose deformation is not properly handled. Most recently, Li *et al.* [18] use the 3D human model [21] to generate human pose flow ground-truth for training a flow estimator. However, similar to other 3D-modeling approaches [29], [45], the issue of large occluded regions is not well addressed due to the lack of correspondence. Moreover, the 3D human modeling is computationally expensive, and it is not always precise on loose clothes, as 3D human modeling focus on body reconstruction rather than the clothes surface reconstruction. Recently, Siarohin *et al.* [35] propose a general image animation model that learns optical flow without correspondence

Fig. 1. Images generated by different methods. The first column contains source images, while the second column contains ground truth images with target poses. We compare our results (last column) with the state-of-the-art methods (rows 3-7). The odd rows display the entire images, and the even rows display the corresponding texture details. In comparison, our method clearly produces the most visually plausible and pleasing effects.

Manuscript received September 30, 2019; revised April 8, 2020, May 30, 2020, and July 18, 2020; accepted August 28, 2020. Date of publication October 20, 2020; date of current version January 20, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Chia-Kai Liang. *(Corresponding author: Haitian Zheng.)*

Haitian Zheng, Lele Chen, and Jiebo Luo are with the Department of Computer Science, University of Rochester, Rochester, NY 14627 USA (e-mail: hzheng15@ur.rochester.edu).

Chenliang Xu is with the Department of EECS, University of Michigan at Ann Arbor, Ann Arbor, MI 48109 USA, and also with the Department of Computer Science, University of Rochester, Rochester, NY 14627 USA.

This article has supplementary downloadable material available at https://doi.org/10.1109/TIP.2020.3031108, provided by the authors.

Digital Object Identifier 10.1109/TIP.2020.3031108

annotation. However, the issue of texture preservation under large occlusion for pose-guided synthesis is not sufficiently addressed by [35].

In this article, we present i) a novel pose flow learning scheme (Stage-I) that does not require correspondence ground-truth to tackle the pose guided transfer task. Next, we propose ii) a coarse-to-fine garment-to-image synthesis pipeline (Stage-II) using feature domain alignment based on the learned flow. Without using affine or TPS transformation [3], [36] or resorting to explicit 3D human modeling [18], [29], [45] to extract correspondence, our method utilizes learned pose flow to capture the complex pose deformation. Our pose flow learning scheme effectively addresses the issue of the occlusion caused by drastic pose changes as our scheme can learn to transfer appearance to occluded regions. In contrast to [18], our approach avoids the computationally inefficient flow ground-truth generation step.

To enable our pose flow learning scheme, we propose in Stage-I a novel texture preserving objective to improve the quality of the learned flow, which is shown to be crucial for the pose-guided synthesis task. We also propose augmentation-based self-supervision to stabilize the flow training. Based on the trained pose flow predictor, we proposed in Stage-II a coarse-to-fine garment-to-image synthesis pipeline using our proposed GarmentNet and SynthesisNet. GarmentNet and SynthesisNet share a unified network structure that utilizes the learned pose flow for multi-scale feature domain warping. Furthermore, we propose a novel gated multiplicative attention module for misalignment-aware synthesis.

Finally, to synthesize more realistic images, we design masking layers in GarmentNet and SynthesisNet to better preserve the image background and person identity. Finally, we use DensePose parsing [1] instead of person keypoints as pose inputs. DensePose parsing contains body segmentation and mesh coordinates, which provide richer information for realistic pose-guided synthesis.

Our main contributions as follows:

- A pose flow learning scheme that learns to transfer the appearance from target images without correspondence annotations. To enable such a learning scheme, a novel texture preserving objective and an augmentation-based self-supervision strategy are proposed, which improve the quality of the transferred appearance.
- A coarse-to-fine synthesis pipeline that consists of GarmentNet and SynthesisNet. GarmentNet and SynthesisNet utilize the trained pose flow predictor for multi-scale feature domain alignment. Furthermore, a novel gated multiplicative attention module is proposed to address the misalignment issue.
- Several design improvements to facilitate more realistic pose-guided synthesis. Specifically, we design masking layers that better preserve person identities and background information. Furthermore, we use DensePose parsing as the pose representation to provide richer pose details for pose-guided synthesis.

The remainder of the paper is organized as follows. Sec. II introduces related work on (pose guided) image synthesis and optical flow learning. The proposed approach is detailed in Sec. III. Experiments are described in Sec. IV. Sec. V concludes the paper.

II. Related Work

A. Image Synthesis

Generative Adversarial Network (GAN) [8] has been widely used for image synthesis tasks. Conditional GAN [26] aims to synthesize an image from a given conditional input content. Based on conditional GAN, Isola *et al.* propose Pix2Pix [11] for image style transfer tasks. Later on, many techniques have been proposed to improve both the synthesis quality and resolution of the generated images. Specifically, Johnson *et al.* [14] use distance on feature vectors yield by layers of VGG network [37] to measure the perceptual similarities. The Gram matrix loss [6] is proposed by Gatys *et al.* for texture synthesis. To improve the image synthesis resolution, Zhang *et al.* [47] propose a two-stage network for generating images from coarse to fine scales. Likewise, Sun *et al.* [39] propose a multiple-stage synthesis model that generates face landmarks for person head inpainting. PatchGAN discriminator [17] is used by Li *et al.* to penalize unrealistic patches. Wang *et al.* [41] and Chen and Koltun [2] propose new generator structures for realistic image synthesis. In addition, techniques such as Wasserstein distance [9] and Spectral Normalization [27] are proposed to stabilize GAN training. Those approaches have improved the synthesized image quality. However, these approaches are limited to spatial deformation as their networks are built on local convolution. In this work, we present a flow-based approach to address the spatial alignment problem in pose-guided synthesis.

B. Pose Guide Synthesis

Ma *et al.* [22] use the source image and target pose landmarks as the conditional input and the UNet [33] structure for pose guided synthesis. Later, Siarohin *et al.* [36] utilize skip connections with hard-coded part-level affine transformation to transform feature maps for new pose image synthesis. Dong *et al.* [3] use the thin-plate spline (TPS) transform trained on synthetic transformations [32] to warp the source domain content. Additionally, Han *et al.* [10] and Wang *et al.* [40] use the TPS transformer for virtual try-on. To handle pose deformations, Neverova *et al.* [29] use DensePose [1] to transfer appearance patterns and utilize inpainting to fill occluded regions. In addition, pose guided synthesis is formulated as a pose-appearance disentanglement problem. Specifically, Esser *et al.* [4] use variational autoencoder [16] to capture the latent space of pose and appearance for appearance manipulation under given poses. Ma *et al.* [23] learn disentangled pose-appearance representation using a multi-branch encoding and decoding scheme. However, the plain UNet structure [4], [22], predefined transformation [29], [36] or TPS transformer [3], [40] are insufficient for handling the complex human pose deformation and occlusion caused by drastic pose changes. Recently, Li *et al.* [18] uses 3D human model [21] to correspondence annotation, then fit a flow estimator to speed up inference. However, generating the correspondence supervision is computationally exhausted. Furthermore, the ground-truth correspondence cannot effectively transfer appearance to

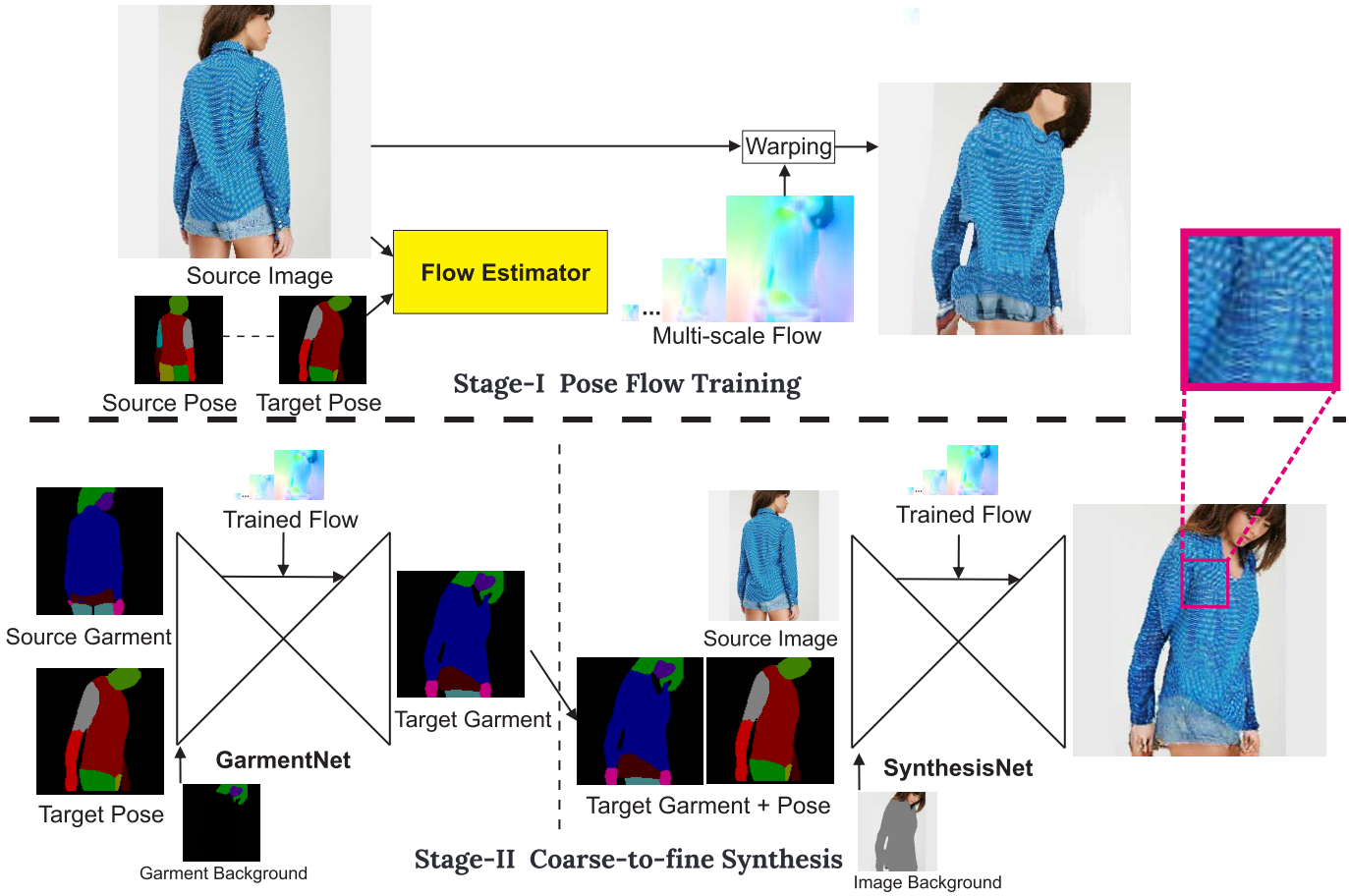


Fig. 2. Our two-stage framework for pose-guided person image synthesis. In stage-I, a flow estimator is trained using our proposed texture-preserving objective. In stage-II, GarmentNet and SynthesisNet use the trained flow estimator to sequentially estimate garment parsing and image output, following a coarse-to-fine pipeline.

occluded regions. In contrast, our flow-training scheme learns to transfer appearance under complex pose deformation and occlusion without using explicit correspondence annotation.

C. Unsupervised Optical Flow Learning

Recently, several approaches have been proposed to learn optical flow in the absence of the ground-truth annotation. Specifically, Jason *et al.* [13] optimize a predictive model using a combination of photometric and smoothness objectives to predict flow. Meister *et al.* [25] utilize left-right consistency to filter out occluded regions. Wang *et al.* [42] further propose an occlusion-aware objective function for unsupervised flow learning. Different from these works, we focus on learning a flow that better preserves the appearance information. Furthermore, our optical flow is estimated using only the source image and pose information. Recently, Siarohin *et al.* [35] apply optical flow learning in an unsupervised fashion for deformed image synthesis. Different from [35], we address the issue of preserving complex garment patterns under large deformation and occlusion.

III. Approach

In this section, we present a flow-based approach to the pose-guided synthesis task that does not require additional correspondence annotations. To this end, we adopt a two-stage pipeline, as illustrated in Fig. 2. In Stage-I, a flow estimator is trained using our proposed texture-preserving objective. In Stage-II, we present GarmentNet and SynthesisNet to sequentially generate garment parsing and image output, using the flow obtained from the previous stage.

In Sec. III-A, we first define the notations that are required by our model. In Sec. III-B, we propose our texture-preserving objective and other details for training a flow estimator for pose-guided alignment. In Sec. III-C, we propose GarmentNet and SynthesisNet to respectively estimate garment parsing and image output.

A. Notations

Given a pair of images I_s and I_t from the source and target domains respectively, pose-guided synthesis aims to generate a image $\hat{I}_t$ that preserves the appearance of I_s and the pose of I_t. To this end, we respectively generate *pose representation* P_s, P_t and *garment parsing* G_s, G_t from I_s and I_t, to capture useful information from the source and target domains. In addition, we extract image residue I_t^r from I_t and garment residues G_t^r from garment G_t, in the hope to capture target identity (i.e., face, hair, and background regions). Fig. 3 illustrates (P_s, P_t), (G_s, G_t), (I_s, I_t) and residues (I_t^r, G_t^r). In fact, P_t, G_t and I_t form a hierarchical structure that gradually provide richer information of the target person. We leverage this hierarchical structure in Sec. III-C to design our coarse-to-fine synthesis pipeline. We note that

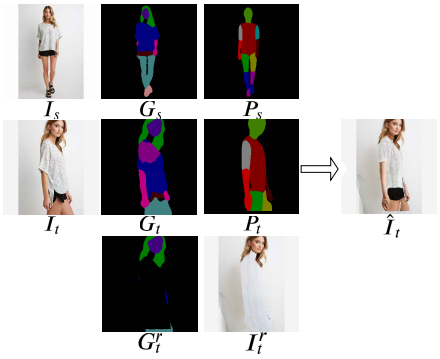


Fig. 3. Notation illustrations for the required data for training and testing. We use subscripts s and t to represent source and target domains, respectively. The notions of I, G and P represent images, garment parsing and pose representation, respectively. (I_t^r, G_t^r) denote image residue and garment residue from the target person. The output of our approach is denoted by $\hat{I}_t$. Please refer to Sec. III-A for more details.

during training, I_s and I_t are from the same outfit of the same person. In testing phase, however, I_s and I_t can be arbitrary person with arbitrary outfits.

To be more specific, the pose representations P_s and P_t are the concatenation of the one-hot pose parsing and the mesh coordinate map from Densepose [1]. Likewise, the garment representations G_s and G_t are the one-hot garment parsing generated using the method by Gong *et al.* [7]. The image residue r_t^i are generated by first removing person region from I_t then perform inpainting [28]. Then, hair and face regions are appended on the inpainted results.[1] Finally, garment residue r_t^g are generated by setting values of one-hot parsing G_t to 0 for background, face and hair channels.

Although our approach can adapt key-point heat maps as an alternative human pose representation, we argue that sparse key-points do not provide sufficient pose information for accurate person image generation. By contrast, DensePose parsing and mesh coordinates provide dense, pseudo-3D information, which is informative to represent pose detail.

B. Stage-I: Texture Preserving Pose Flow Training

With the extracted pose representations P_s and P_t, we present a flow training scheme to generate adaptive, texture-preserving alignment without resorting to pseudo flow ground-truth that is generated by the computationally inefficient SMPL model [21] or oversimplified affine [36] or TPS transformation [3], [40].

As shown in Fig. 2, our flow estimator takes the source image, pose and target pose as inputs to generate multi-scale flow-fields to indicate the pose deformation. Formally, let $\text{Flow}(\cdot,\cdot)$ denote our flow estimator, which takes $[I_s; P_s]$ and P_t from source and target domains as inputs and outputs flow fields at multiple scales:

$$\{\mathbf{w}_{t\to s}^{(0)}, \mathbf{w}_{t\to s}^{(1)}, \cdots, \mathbf{w}_{t\to s}^{(5)}\} = \text{Flow}([I_s; P_s], P_t). \tag{1}$$

where notation $\mathbf{w}_{t\to s}^{(l)}$ denotes the backward flow field from the target image to the source images at scale $l \in \{0, \cdots, 5\}$.

We employ FlowNetS [5] as the baseline structure to implement $\text{Flow}([I_s; P_s], P_t)$. Note that, unlike a normal flow estimator, $\text{Flow}(\cdot,\cdot)$ leverages pose information for flow estimation. Meanwhile, we have also modified FlowNetS to improve the flow-field definition and to reduce memory usage. Please refer to Appendix A for more details.

Unsupervised flow training on natural images has been explored in several recent works. These approaches mainly rely on the photometric loss [13]

$$\mathcal{L}_p(I_s, I_t, \mathbf{w}_{t\to s}^{(0)}) = \left|\left|\rho\left(I_t - \text{warp}(I_s; \mathbf{w}_{t\to s}^{(0)})\right)\right|\right|_1 \tag{2}$$

to measure the difference between the target image and the backward-warped source image using the predicted flow. Here, $\text{warp}(\cdot;\cdot)$ denotes the image domain backward warping operation implemented by a bilinear sampler [12] and $\rho(x) = (x^2 + \epsilon^2)^\alpha$ is a robust loss function [38]. Furthermore, total variation-based (TV) spatial smoothness loss is also utilized to regularize the flow prediction [31]:

$$\mathcal{L}_{TV}(\mathbf{w}_{t\to s}^{(l)}) = \left|\left|\frac{\partial}{\partial x}\mathbf{w}_{t\to s}^{(l)}\right|\right|_1 + \left|\left|\frac{\partial}{\partial y}\mathbf{w}_{t\to s}^{(l)}\right|\right|_1. \tag{3}$$

Due to the complexity of person images and the large displacement from source pose to target pose, the warping-based photometric term is highly non-convex. As as result, the gradient descendent training with the naive photometric loss and spatial smoothness loss will lead to difficulty in convergence. To solve this issue, we use multi-scale strategy, where photometric losses and spatial smoothness losses summed at multiple scales $l \in \{0, \cdots, 5\}$.

In our experiment, we found that the multi-scale training will still suffer from damaged local textures for the warped images $\text{warp}(I_s; \mathbf{w}_{t\to s}^{(0)})$, and the learned flow fails to transfer realistic details from source images (see Fig. 8 for details). We attribute this deficiency to the poor ability of $\mathcal{L}_p$ and $\mathcal{L}_{TV}$ in preserving the high-frequency texture. In order to preserve realistic details and textures for better pose-guided synthesis, we propose a texture-preserving objective $\mathcal{L}_{texture}^{(l)}$ that enforces texture similarity between the I_t and $\text{warp}(I_s; \mathbf{w}_{t\to s}^{(0)})$ at scale l:

$$\mathcal{L}_{texture}^{(l)}(I_t, I_s, \mathbf{w}_{t\to s}^{(0)}) = \left|\left|\mathbf{G}\left(\mathbf{f}_{vgg}^{(l)}(I_t)\right) - \mathbf{G}\left(\mathbf{f}_{vgg}^{(l)}(\text{warp}(I_s; \mathbf{w}_{t\to s}^{(0)}))\right)\right|\right|_1, \tag{4}$$

where $\mathbf{f}_{vgg}^{(l)}(\cdot)$ represents the l'th VGG [37] feature map from layer {`relu1_2`, `relu2_2`, `relu3_2`, `relu4_2`, `relu4_3`} of the given input image, and $\mathbf{G}(\cdot)$ denotes the Gram matrix [6] to capture the second-order statistic of the given feature map. Although the objective $\mathcal{L}_{texture}^{(l)}$ is widely used in style transfer tasks, we are the first to show that the texture loss is crucial for learning a reasonable flow estimator for pose-guided synthesis tasks (see Fig. 8 for details).

Finally, we use a multi-scale version of the three losses, which are then weighted summed to compute the final loss. Let $I_s^{(l)}$ and $I_t^{(l)}$ denote the resized images of I_s and I_t at scale $l \in \{0, \cdots, 5\}$, the overall objective is given by:

$$\begin{aligned}\mathcal{L}_{StageI} = \sum_{l=0}^{5} s_l(&\mathcal{L}_p(I_s^{(l)}, I_t^{(l)}, \mathbf{w}_{t\to s}^{(l)}) \\ &+ \beta_l \mathcal{L}_{texture}^{(l)}(I_t, I_s, \mathbf{w}_{t\to s}^{(0)}) \\ &+ \gamma_l \mathcal{L}_{TV}(\mathbf{w}_{t\to s}^{(l)})).\end{aligned} \tag{5}$$

[1] We use the garment parsing G_t to generate the regions of human body, hair and face.

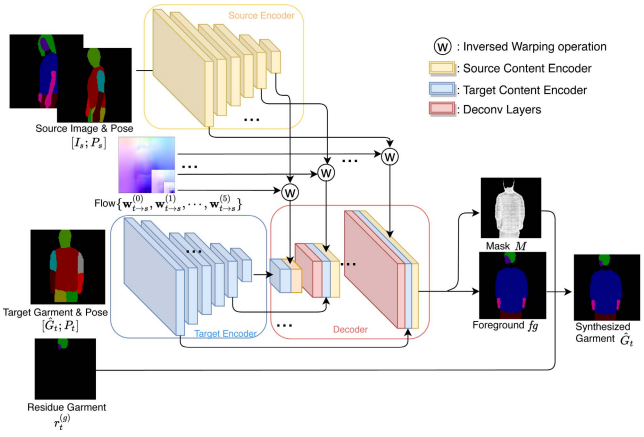


Fig. 4. The network structure of GarmentNet. Given the generated flow from Stage-I, GarmentNet encodes information from the source and target domains using a Source Domain Encoder (yellow) and a Target Domain Encoder (blue), respectively. After warping-based alignment, the source domain features are aggregated with the target domain features at multiple scales by our Decoder (red). Finally, the generated foreground is alpha-blended with the residue garment to synthesize garment parsing. In the testing stage, the source and target image are from different persons.

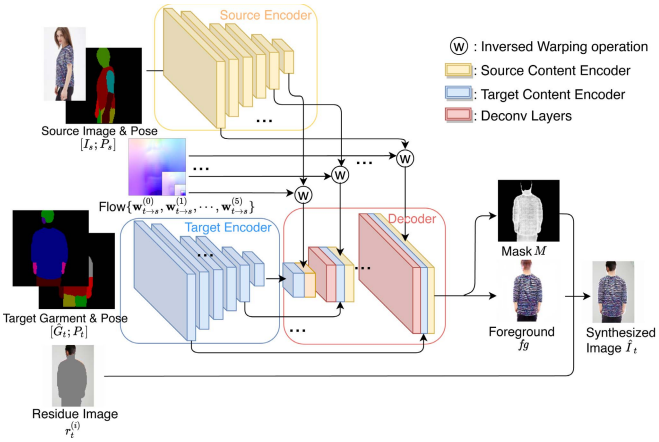


Fig. 5. The network structure of SynthesisNet. Given the generated flow from Stage-I and the synthesized garment parsing, GarmentNet encodes information from the source and target domains using a Source Domain Encoder (yellow) and a Target Domain Encoder (blue), respectively. After warping-based alignment, the source domain features are aggregated with the target domain features at multiple scales by the Decoder (red). Finally, the generated foreground is alpha-blended with the residue image to synthesize image output. In testing stage, the source and target image are from different persons.

To further stabilize training, an augmentation-based self-supervision strategy is employed to regularize the learned flow. Specifically, we apply random augmentation on r ratio of source inputs to generate the target pose and pseudo ground-truth. Formally, we use the following update rules to transform the original data before an iteration of flow estimator training:

$$\begin{aligned} a &\sim \text{Bern}(r), \\ P_t &\leftarrow \text{Aug}(aP_s + (1-a)P_t, \theta) \\ I_t &\leftarrow \text{Aug}(aI_s + (1-a)I_t, \theta), \end{aligned} \tag{6}$$

where $\text{Aug}(\cdot, \theta)$ denotes an augmentation transformation based on cropping, affine transformation and flipping with a random control parameter θ, and a denotes a binary random variable generated by a Bernoulli distribution $\text{Bern}(r)$. We set the ratio of the Bernoulli distribution r to a small value such that a small proportion of training samples are generated from random synthetic transformation. This procedure can help stabilize the flow model training as the flow estimator can learn from simple affine transformations in the initial stage of training before learning the complex pose deformation.

C. Stage-II: Coarse-to-Fine Synthesis

Based on the learned flow estimator in Stage-I, we propose GarmentNet and SynthesisNet to sequentially synthesize garment parsing and image output following a coarse-to-fine pipeline (Fig. 2 bottom). As illustrated in Fig. 4 and Fig. 5, GarmentNet and SynthesisNet share a unified network structure, which utilize the learned flow in stage-I for feature alignment. Afterwards, U-Net decoder serves to fuse information from both the source and target domains. On top of the decoder, an alpha blending layer is applied to preserve background information and to generate final outputs.

Formally, GarmentNet utilizes $[G_s, P_s]$ to encode source domain information, P_t to encode target domain information, $\{\mathbf{w}_{t\to s}^{(0)}, \mathbf{w}_{t\to s}^{(1)}, \cdots, \mathbf{w}_{t\to s}^{(5)}\}$ from stage-I for alignment, and G_t^r to keep the shape of target hair and face. The notation $[\cdot, \cdot]$ denotes channel-wise concatenation. The output target garment of GarmentNet is denoted by $\hat{G}_t$:

$$\begin{aligned} \hat{G}_t = \text{GarmentNet}([G_s, P_s], P_t, \\ \{\mathbf{w}_{t\to s}^{(0)}, \mathbf{w}_{t\to s}^{(1)}, \cdots, \mathbf{w}_{t\to s}^{(5)}\}, \quad I_t^r). \end{aligned} \tag{7}$$

Similarly, SynthesisNet (see Eq. 8) utilizes $[I_s, P_s]$ to encode source domain information, $[\hat{G}_t, P_t]$ to encode target domain information, $\{\mathbf{w}_{t\to s}^{(0)}, \mathbf{w}_{t\to s}^{(1)}, \cdots, \mathbf{w}_{t\to s}^{(5)}\}$ from stage-I for alignment, and I_t^r to keep the background, hair and face of target image. The output of SynthesisNet is the synthesized image $\hat{I}_t$:

$$\begin{aligned} \hat{I}_t = \text{SynthesisNet}([I_s, P_s], [\hat{G}_t, P_t], \\ \{\mathbf{w}_{t\to s}^{(0)}, \mathbf{w}_{t\to s}^{(1)}, \cdots, \mathbf{w}_{t\to s}^{(5)}\}, \quad I_t^r). \end{aligned} \tag{8}$$

Since the two networks share the similar inputs format and network structure, we elaborate on the shared network structure in the next section.

Network Structure: As shown in Fig. 4 and 5, our model relies on a source encoder $\text{Enc}_s(\cdot)$ and a target encoder $\text{Enc}_t(\cdot)$ to respectively generate multi-scale feature maps from source and target domains inputs IN_s, IN_t:

$$\begin{aligned} \{\mathbf{f}_s^{(0)}, \cdots, \mathbf{f}_s^{(5)}\} &= \text{Enc}_s(IN_s), \\ \{\mathbf{f}_t^{(0)}, \cdots, \mathbf{f}_t^{(5)}\} &= \text{Enc}_t(IN_t). \end{aligned} \tag{9}$$

For GarmentNet, inputs are set to $IN_s = [G_s, P_s], IN_t = P_t$. For SynthesisNet, inputs are set to $IN_s = [I_s, P_s], IN_t = [\hat{G}_t, P_t]$.

We use six stacked strided convolutional layers to implement $\text{Enc}_t(\cdot)$ and six stacked strided convolutional layers following seven residue blocks to implement $\text{Enc}_s(\cdot)$. The additional residue blocks serve to increase feature representation capacity.

To perform spatial alignment, the source domain features $\mathbf{f}^{(l)}$ at all scales $l \in \{0, \cdots, 5\}$ are warped to the target domain by a bilinear sampler [12] according to the flow fields $\mathbf{w}_{t\to s}^{(l)}$

for layers $l \in \{1, \cdots, 5\}$, formally:

$$\mathbf{f}_{s\to t}^{(l)} = \text{warp}(\mathbf{f}_s^{(l)}; \mathbf{w}_{t\to s}^{(l)}). \tag{10}$$

After spatial alignment, a U-Net fusion decoder is used for feature aggregation. However, instead of directly concatenating feature maps for aggregation, we propose a gated multiplicative attention module to filter the misaligned source domain features. Specifically, the gated multiplicative attention filtering at scale l is defined as:

$$\mathbf{f}_{s\to t}^{(l)\prime} = \mathbf{f}_{s\to t}^{(l)} \odot \sigma(\mathbf{f}_{s\to t}^{(l)\top} \mathbf{W}^{(l)} \mathbf{f}_t^{(l)}), \tag{11}$$

where $\sigma(\cdot)$ represents the sigmoid function, $\odot$ represents element-wise multiplication and $\mathbf{W}^{(l)}$ is a learnable matrix that measures dot product similarities between $\mathbf{f}_s^{(l)}$ and $\mathbf{f}_t^{(l)}$ on to-be-learned linear space. The gated multiplicative attention filtering can be efficiently implemented on the 2D feature maps using 1×1 convolution, element-wise multiplication and summation. Please refer to Appendix B for details. Building on top of the gated multiplicative attention filtering operation, our decoder uses the following equations to generate the aggregated feature maps $\mathbf{f}_{dec}^{(l)}$:

$$\begin{aligned} \mathbf{f}_{dec}^{(0)} &= \text{Deconv}([\mathbf{f}_{s\to t}^{(0)\prime}; \mathbf{f}_t^{(0)}]), \\ \mathbf{f}_{dec}^{(l)} &= \text{Deconv}([\mathbf{f}_{dec}^{(l-1)}; \mathbf{f}_{s\to t}^{(l)\prime}; \mathbf{f}_t^{(l)}]), \quad l \in \{1, \cdots, 5\}. \end{aligned} \tag{12}$$

Afterwards, our network simultaneously generates foreground content *fg* along with a mask M that ranges from 0 to 1 to avoid changing the residue content of the target r_t. Specifically,$\mathbf{f}_{dec}^{(5)}$ is passed to two independent convolutional layers to respectively generate foreground content *fg* and a corresponding foreground mask M:

$$\begin{aligned} fg &= \text{Conv}(\mathbf{f}_{dec}^{(5)}), \\ M &= \text{Conv}(\mathbf{f}_{dec}^{(5)}). \end{aligned} \tag{13}$$

Finally, the output content *out* is generated by alpha-blending the foreground content *fg* with the residue content r:

$$out = M \odot fg + (1 - M) \odot r. \tag{14}$$

For GarmentNet, softmax function is applied after *out* to generate the garment parsing, i.e. $\hat{G}_s = \text{softmax}(out)$. For SynthesisNet, tanh function is applied after *out* to generate the normalized image, i.e. $\hat{I}_s = \tanh(out)$.

Training Objective: For GarmentNet training, we use the cross entropy loss between the target garment G_t and prediction $\hat{G}_t$:

$$\mathcal{L}_{\text{GarmentNet}} = -\sum_{i,j}\sum_{n} (G_t)_{i,j,n} \log((\hat{G}_t)_{i,j,n}), \tag{15}$$

where i, j enumerate pixel positions and n enumerates channals of garment parsing.

For SynthesisNet training, we use a combination of ℓ_1 pixel domain loss, VGG feature loss, texture loss, and GAN loss. The training objective is represented as:

$$\mathcal{L}_{\text{SynthesisNet}} = \lambda_1\mathcal{L}_1 + \lambda_2\mathcal{L}_{\text{VGG}} + \lambda_3\mathcal{L}_{\text{texture}} + \lambda_4\mathcal{L}_{\text{GAN}}, \tag{16}$$

where $\mathcal{L}_1 = \left|\left|\hat{I}_t - I_t\right|\right|_1$ computes the ℓ_1 differences between the synthesized image and the ground-truth, $\mathcal{L}_{\text{VGG}} = \left|\left|\mathbf{f}_{\text{VGG}}(\hat{I}_t) - \mathbf{f}_{\text{VGG}}(I_t)\right|\right|_1$ computes feature map differences on the `relu4_2` layer of the VGG network of the two image. Similar to Eq. 4, $\mathcal{L}_{\text{texture}} = \left|\left|\mathbf{G}\left(\mathbf{f}_{\text{VGG}}(\hat{I}_t)\right) - \mathbf{G}\left(\mathbf{f}_{\text{VGG}}(I_t)\right)\right|\right|_1$ (Eq. 4) computes the texture-level differences of the two images, and $\mathcal{L}_{\text{GAN}} = (D(I_t) - 1)^2 + D(\hat{I}_t)^2$ measures how well the synthetic image can fool a trained discriminator $D(\cdot)$. Similar to CycleGAN [49], we use least-square distance [24] rather than negative log likelihood to compute the $\mathcal{L}_{\text{GAN}}$, whereas the discriminator is implemented using the PatchGAN architecture [11] with spectrum normalization [27]. The hyper-parameters $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are set to $\lambda_1 = 1.0$, $\lambda_2 = 0.1, \lambda_3 = 0.002, \lambda_4 = 0.5$ respectively in our experiments.

Additionally, we use a similar augmentation-based self-supervision strategy as described in Sec. III-B to regularize SynthesisNet. During training, 25% percent of the source domain samples come from the augmented target domain samples to help SynthesisNet to learn from simple tasks first.

IV. Experiments

A. Dataset

We train and evaluate our method on the DeepFashion [20] dataset, which contains 52,712 person images of sizes 256×256. Images that only contain trousers are removed using DensePose [1], resulting in 40,906 valid images. We randomly divide the dataset into 68,944 training pairs and 1,000 testing pairs. Additionally, we evaluate our DeepFashion trained model on other datasets to understand how well our model can generalize to unseen poses, clothing styles or background.

As detailed in Section III-A, pose representation is generated using DensePose, while garment representation is generated using the method of [7]. Finally, we additionally use keypoint heatmap [22] as pose representation to test our algorithm.

B. Implementation Details

In Stage-I and Stage-II, we set the learning rate to 0.0001 for the flow estimator and the generator. Following [27], the learning rate for the discriminator is 0.0004. We adopt Adam [15] optimizer ($\beta_1 = 0.9$ and $\beta_2 = 0.999$) in all experiments. Random cropping, affine transformation and flipping are used to augment data. The flow estimator, GarmentNet and SynthesisNet are trained for 20, 20 and 40 epochs, respectively. In Stage-I, we set the ratio from Eq. 6 to $r = 0.25$ and the parameters from Eq. 5 to $(s_0, s_1, s_2, s_3, s_4) = (1, 1, 0.5, 0.25, 0.125)$, $(\beta_0, \beta_1, \beta_2, \beta_3, \beta_4) = (0.002, 0.002, 0.002, 0.002, 0)$, $(\gamma_0, \gamma_1, \gamma_2, \gamma_3, \gamma_4) = (0.1, 0.1, 0.1, 0.1, 0)$.

Since our approach can adopt keypoint heatmap [22] as pose representation by simply altering P_s, P_t, we additionally train our model using the key point representation while maintaining other inputs unchanged.

C. Quantitative Evaluation

To quantitatively evaluate the synthesis results, low-level metrics like Structural Similarity (SSIM) [43], Multi-scale Structural Similarity (MS-SSIM) [44] and perceptual-level

TABLE I

QUANTITATIVE COMPARISON OF DIFFERENT METHODS IN TERMS OF BOTH THE MASKED SSIM/MSSSIM/INCEPTION SCORE (IS) AND THE LEARNED PERCEPTUAL IMAGE PATCH SIMILARITY (LPIPS) AT 256×256 AND 128×128 RESOLUTION. HIGHER SCORES ARE BETTER FOR METRICS WITH UPARROW (↑), AND VICE VERSA

Methods	SSIM-128↑	msSSIM-128↑	SSIM↑	msSSIM↑	IS-128↑	IS↑	LPIPS↓	LPIPS-128↓
PG2 [22]	0.864	0.911	0.857	0.891	3.455 ± 0.226	4.266 ± 0.371	0.192	0.190
BodyROI7 [23]	0.842	0.882	0.837	0.865	3.282 ± 0.173	3.855 ± 0.158	0.193	0.201
DSCF [36]	0.856	0.902	0.851	0.884	3.458 ± 0.198	4.226 ± 0.326	0.159	0.157
Vunet [4]	0.822	0.830	0.827	0.827	3.424 ± 0.143	4.176 ± 0.320	0.226	0.258
Soft-gate [3]	0.860	0.908	0.853	0.888	3.270 ± 0.219	3.868 ± 0.387	0.140	0.135
IF [18]	**0.877**	**0.926**	**0.865**	**0.906**	3.262 ± 0.293	3.809 ± 0.360	0.128	0.128
full model	0.854	0.905	0.848	0.884	3.540 ± 0.294	4.197 ± 0.291	**0.124**	**0.124**
Ours-kp	0.831	0.870	0.831	0.852	**3.646 ± 0.285**	**4.295 ± 0.296**	0.163	0.169

Fig. 6. Comparison with the state-of-the-art approaches. The last four columns depict the warped source image, foreground prediction in stage-II, mask prediction in stage-II, and our final output. In comparison, our method clearly produces the most visually plausible and pleasing effects.

metrics like Inception Score (IS) [34] and the Perceptual Image Patch Similarity Distance (LPIPS) [48] are measured on different approaches, including PG2 [22], BodyROI [23], Vunet [4], DSCF [36], Soft-gated GAN (Soft-gate) [3] and Intrinsic Flow (IF) [18]. For LPIPS, we use the linearly calibrated Alex model, please refer to [48] for details. Since our approach relies on the background information, we report the masked version of all the metrics for fair comparisons. The masks are generated by running [7] to exclude background, hair, and face region. We additionally test all the metrics at resolution 128×128 to measure similarities at a global scale.

From Table I, our methods (*ours*) substantially outperforms the remaining methods in IS-based measurements and LPIPS distances, as our texture-preserving flow is able to preserve texture patterns form source images. In terms of the low-level SSIM-based measurements, our method achieves competitive performance compared to the other approaches. When trained using keypoint heatmap (*ours-kp*), we observe similar high IS scores for both models and better LPIPS scores for our model. It suggests both models (*ours* and *ours-kp*) preserve realistic texture. However, with the help of the DensePose pose representation, our model (*ours*) generates better global shape.

D. Qualitative Evaluation

We conduct a subjective assessment to evaluate our method qualitatively. Specifically, we ask 15 subjects to rank image qualities among the 6 algorithms ([3], [4], [22], [23], [36] and ours). The subjects are instructed to rank the six images, based on the realism of the generated garments as well as global garment structures. The subjects are then asked to provide a score from 1 to 6 for each image, representing the best quality to the worst quality, respectively. We plot the ranking

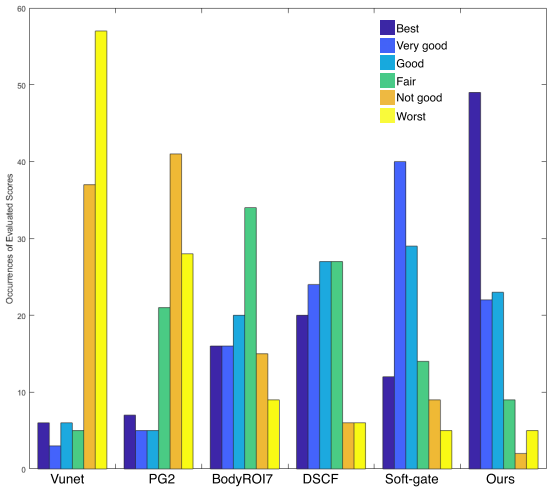


Fig. 7. Subjective quality assessment of different algorithms. For each algorithm, the bar depicts the number of occurrences of scores, while blue to yellow colors represent the scores from the best to the worst.

TABLE II

QUANTITATIVE COMPARISON OF DIFFERENT FLOW TRAINING SCHEMES AND SYNTHESISNET TRAINING SCHEMES IN TERMS OF BOTH THE MASKED SSIM/MSSSIM/INCEPTION SCORE (IS) AND THE LEARNED PERCEPTUAL IMAGE PATCH SIMILARITY (LPIPS) AT 256 × 256 AND 128 × 128 RESOLUTION. HIGHER SCORES ARE BETTER FOR METRICS WITH UP ARROWS (↑), AND VICE VERSA

Flow training schemes	SSIM-128↑	msSSIM-128↑	SSIM↑	msSSIM↑	IS-128↑	IS↑	LPIPS↓	LPIPS-128↓
w/o multi-scale	0.822	0.853	0.825	0.839	**4.115 ± 0.211**	**4.689 ± 0.327**	0.240	0.240
w/o texture	**0.837**	0.880	**0.837**	0.861	3.843 ± 0.246	4.204 ± 0.245	0.217	0.217
w/o semi	0.835	0.880	0.834	0.861	3.978 ± 0.348	4.412 ± 0.223	0.196	0.196
full training scheme	0.836	**0.882**	0.835	**0.863**	3.934 ± 0.274	4.404 ± 0.331	**0.193**	**0.193**
SynthesisNet training schemes	**SSIM-128↑**	**msSSIM-128↑**	**SSIM↑**	**msSSIM↑**	**IS-128↑**	**IS↑**	**LPIPS↓**	**LPIPS-128↓**
w/o flow	0.849	0.898	0.844	0.877	3.421 ± 0.177	3.952 ± 0.291	0.141	0.141
w/o att	0.853	0.904	0.848	0.883	3.391 ± 0.161	3.946 ± 0.374	0.128	0.128
w/o semi	0.851	0.903	0.846	0.882	3.480 ± 0.273	3.995 ± 0.333	0.128	0.128
full model	0.854	0.905	0.848	0.884	3.540 ± 0.294	**4.197 ± 0.291**	0.124	0.124
full model w/ joint	**0.859**	**0.910**	**0.849**	**0.888**	**3.618 ± 0.233**	4.002 ± 0.458	**0.123**	**0.123**

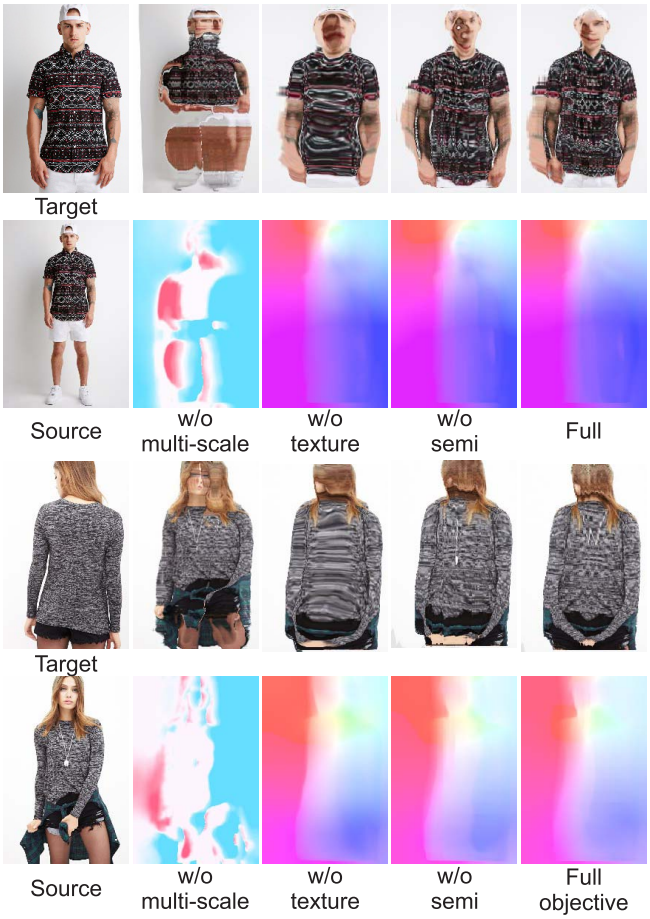


Fig. 8. Comparisons of different pose flow training schemes. Our full flow training objective (Eq. 5) generates more visually plausible and pleasing textures and more consistent flow.

histogram of different algorithms in Fig. 7. From the figure, our method is most frequently chosen as the best due to structurally consistent texture. DSCF [36] achieves the second place due to its ability to maintain texture structure from the source image using rigid transformations. The qualitative results of different approaches, the warped source image and foreground/mask prediction from stage-II are shown in Fig. 6. It can be noticed that the existed approaches generate blurry results or incorrect textures. By contrast, our method can preserve texture details from source images. Notably, our approach generates better warping results in comparison with IF, especially under large pose changes.

Fig. 9. Visual comparisons of different SynthesisNet training schemes. Our full model generates more visually plausible and pleasing texture details with more coherent global structures.

E. Ablation Study

Pose Flow Training: To evaluate the effectiveness of each component in the flow training scheme, we separately train three variants of the proposed flow estimators: i) *w/o multi-scale*, only computing loss at the finest scale, ii) *w/o texture*, removing texture loss $\mathcal{L}_{texture}$, and iii) *w/o semi*, removing the augmentation-based self supervision. Table II compares the three models with our full model by computing the SSIM, IS, and LPIPS-based scores of the inversely warped images using the trained flow at the finest scale. The inversely warped images are also visualized in Fig.8. It is observed that our full model outperforms *w/o semi* and *w/o multi-scale* in terms of LPIPS scores. It is consistent with the visualization from Fig. 8, showing that our full model can generate flow with more visually plausible and pleasing details. The *w/o multi-scale* performs well in IS scores, and it is possibly because *w/o multi-scale* tends to retain the realistic original source image. However, *w/o multi-scale* does not preserve the semantics of the target pose. In terms of SSIM-based measurement, the full flow training scheme achieves the best ms-SSIM scores, suggesting that the full model is better at preserving global structures.

SynthesisNet Design: To evaluate the effectiveness of each component in training SynthesisNet, ablation studies are

TABLE III

QUANTITATIVE COMPARISON OF VARIOUS APPROACHES ON THE MVC DATASET USING THE MODELS TRAINED ON THE DEEPFASHION DATASET. PERFORMANCES ARE MEASURED IN TERMS OF THE MASKED SSIM/MSSSIM/IS SCORES AT 256×256 RESOLUTION AND 128×128 RESOLUTION. HIGHER SCORES ARE BETTER FOR METRICS WITH UP ARROWS (↑), AND VICE VERSA. TOP TWO SCORES ARE IN BOLD

Methods	SSIM↑	SSIM-128↑	msSSIM↑	msSSIM-128↑	IS↑	IS-128↑
PG2 [22]	0.817	0.806	0.851	0.840	3.401 ± 0.269	3.662 ± 0.361
BodyROI7 [23]	0.798	0.792	0.828	0.823	3.043 ± 0.250	3.039 ± 0.152
DSCF [36]	0.816	0.810	0.846	0.841	3.358 ± 0.229	3.151 ± 0.229
Vunet [4]	0.806	0.794	0.840	0.833	3.294 ± 0.190	2.871 ± 0.222
Ours	**0.836**	**0.839**	**0.857**	**0.853**	**3.603 ± 0.300**	**3.451 ± 0.426**
Ours-MVC-finetuned	**0.839**	**0.840**	**0.863**	**0.859**	**3.737 ± 0.415**	**3.365 ± 0.273**

Fig. 10. Comparison with the state-of-the-art approaches on the MVC dataset. Patches are zoomed in to visualize detailed textures. The last two columns depict our DeepFashion trained model and our MVC finetuned model.

performed in the following ways: i) we remove the flow estimator for alignment, resulting in *w/o flow*, a UNet-like structure that does not perform feature alignment, ii) we replace the gated multiplicative attentive fusion modules with concatenation operations, which is called *w/o att*, iii) we replace the semi-supervised data generation scheme with only the supervised data, which is called *w/o semi*. Table II compares the qualitative scores in terms of SSIM, ms-SSIM, IS and their masked versions. From the table, we observe that the SSIM-based performances substantially deteriorate without the flow-based alignment module. Meanwhile, the gated multiplicative attentive fusion helps to improve the inception scores of the generated images. Also, semi-supervised training improves performance marginally. Visualization is also shown in Fig. 9. From the figure, we observe that our full model is able to retain the global structure due to flow-based alignment. Comparing *w/o att* and *full*, we see that with the gated multiplicative attention module, our model generates globally consistent texture details. In addition to the following ablation models, we perform a joint fine-tuning on the trained GarmentNet and SynthesisNet, which is called *full model w/ joint*. From Table II, joint fine-tuning can further improve the synthesis performance.

F. Generalization

To understand the generalization ability of our trained model and how well our model can perform on real-world datasets, we evaluate our trained model on three additional datasets:

Multi-View Clothing Dataset: The Multi-view Clothing dataset (MVC) [19] contains 161,260 person images and 645,040 pairs in total. We report the results on the MVC dataset using various models that are trained on the DeepFashion dataset. We also report the performance of our fine-tuned model using 120,000 pairs selected from the MVC training set. Table III shows the evaluation of our approach in comparison to other approaches. The generated new-person images are visualized in Fig. 10.

Amazon Fashion Video Data: We evaluate our approach on a set of online video data. Specifically, we crawl clothing item demo videos from the Amazon Fashion website. The initial frame from various source video is used as the source images to synthesize each frame from the target video. The synthesized videos are shown in the supplementary materials. In Fig. 11, the top row shows the target video, while the resting rows show the synthesized video with different clothing styles from source images. As demonstrated in Fig. 11, our approach generates temporal-consistent frames with distinctive texture details, suggesting that our method can effectively generalize to unseen poses and clothing styles.

Garment Transfer to Real Person: To examine the applicability of our approach in real-world scenes, we collect videos of people in real scenes with various poses using a typical smartphone. Fig. 12 visualizes consecutive frames of our

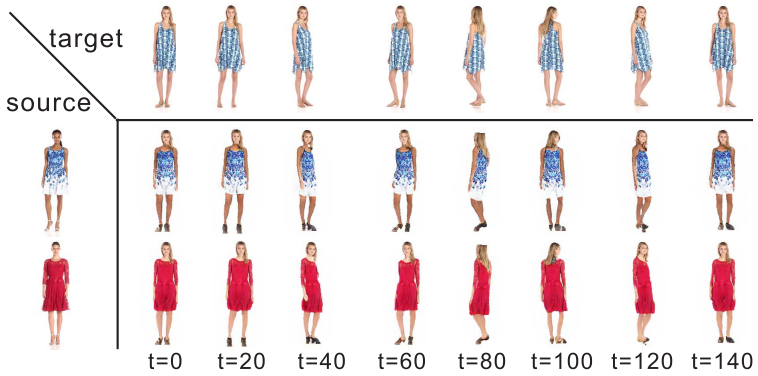


Fig. 11. Garment transfer on the Amazon Fashion videos. The top row shows the target frames, while the resting rows show the synthesized frames. The horizontal axis represents the time step. Our approach can generate temporally consistent frames with distinctive texture details.

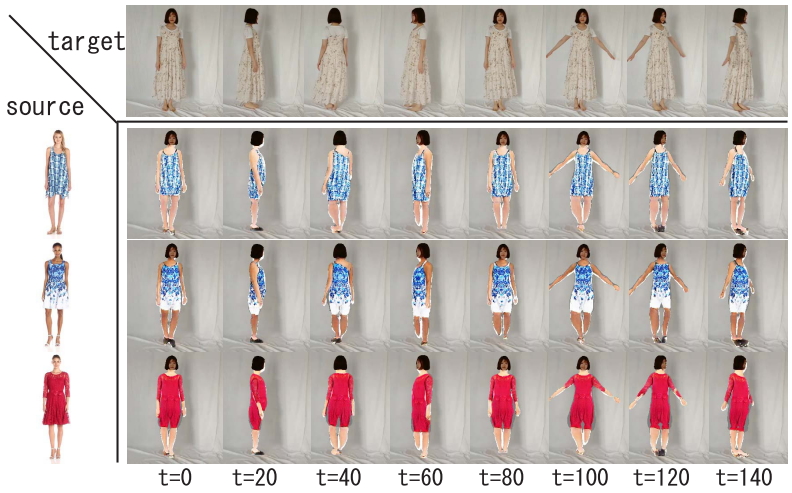


Fig. 12. Garment transfer on our self-collected real-world videos. The top row shows the target frames, while the remaining rows show the synthesized frames. The horizontal axis represents the time step. Our approach can generate temporally consistent frames with distinctive texture details.

captured video and our transferred video, showing that our approach can generate visually plausible and pleasing new clothing styles under challenging real-world environments.

V. Conclusion

To better model person appearance transformation for pose-guided synthesis, we propose a novel pose flow learning scheme that learns to transfer appearance from target images without using generated pseudo correspondence ground-truth. Furthermore, we propose a texture preserving objective and an augmentation-based self-supervision scheme, which are shown to be effective for learning appearance-preserving pose flow. Based on the learned pose flow, we propose a coarse-to-fine synthesis pipeline using a carefully designed network structure for multi-scale feature domain alignment. To address the misalignment issue, we propose a gated multiplicative attention module. In addition, masking layers are proposed to preserve target identities and background information. Experiments on the DeepFasion, MVC, and other real-world datasets have validated the effectiveness and robustness of our approach.

Appendix A
Adaptation of FlowNetS

To implement the flow estimator function Flow() from Eq. 1, we use the FlowNetS network structure.

Algorithm 1 Gated Multiplicative Attention Filtering

Input: $\mathbf{f}_{s\to t}^{(l)\prime}, \mathbf{f}_t^{(l)}$
Output: $\mathbf{f}_{s\to t}^{(l)\prime}$
compute filter $\sigma(\mathbf{f}_{s\to t}^{(l)\top}\mathbf{W}^{(l)}\mathbf{f}_t^{(l)})$:
`att = torch.sum(conv_W(`$\mathbf{f}_{s\to t}^{(l)}$`) *`$\mathbf{f}_t^{(l)}$`, 1)`
`att = torch.sigmoid(att)`
perform filtering :
$\mathbf{f}_{s\to t}^{(l)\prime}$ `= torch.mul(`$\mathbf{f}_{s\to t}^{(l)}$`, att)`
return $\mathbf{f}_{s\to t}^{(l)\prime}$

However, several adaptations are made. First, we reduce the channel of each convolution/deconvolution layer to 64 for memory efficiency. Second, to improve the flow definition at scale 0, the $\times 4$ bilinear upsampling layer at the end of the original FlowNetS is replaced by two $\times 2$ U-Net upsampling modules.

Appendix B
Code for Gated Multiplicative Attention Filtering

We show that the gated multiplicative attention filtering

$$\mathbf{f}_{s\to t}^{(l)\prime} = \mathbf{f}_{s\to t}^{(l)} \odot \sigma(\mathbf{f}_{s\to t}^{(l)\top}\mathbf{W}^{(l)}\mathbf{f}_t^{(l)}),$$

from Eq. 11 can be implemented using 3 lines of code in PyTorch in Algorithm 1, where function `conv_W()` defines a 1×1 convolutional operation with its trainable parameters $\mathbf{W}^{(l)}$.

REFERENCES

[1] R. A. Güler, N. Neverova, and I. Kokkinos, "DensePose: Dense human pose estimation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7297–7306.

[2] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1511–1520.

[3] H. Dong, X. Liang, K. Gong, H. Lai, J. Zhu, and J. Yin, "Soft-gated warping-GAN for pose-guided person image synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 472–482.

[4] P. Esser, E. Sutter, and B. Ommer, "A variational u-net for conditional appearance and shape generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8857–8866.

[5] P. Fischer *et al.*, "FlowNet: Learning optical flow with convolutional networks," 2015, *arXiv:1504.06852*. [Online]. Available: http://arxiv.org/abs/1504.06852

[6] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 262–270.

[7] K. Gong, X. Liang, D. Zhang, X. Shen, and L. Lin, "Look into person: Self-supervised structure-sensitive learning and a new benchmark for human parsing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 932–940.

[8] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[9] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.

[10] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis, "VITON: An image-based virtual try-on network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7543–7552.

[11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1125–1134.

[12] M. Jaderberg *et al.*, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

[13] J. Y. Jason, A. W. Harley, and K. G. Derpanis, "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 3–10.

[14] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. ECCV*. New York, NY, USA: Springer, 2016, pp. 694–711.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[16] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*. [Online]. Available: http://arxiv.org/abs/1312.6114

[17] C. Li and M. Wand, "Precomputed real-time texture synthesis with Markovian generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 702–716.

[18] Y. Li, C. Huang, and C. Change Loy, "Dense intrinsic appearance flow for human pose transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3688–3697.

[19] K.-H. Liu, T.-Y. Chen, and C.-S. Chen, "MVC: A dataset for view-invariant clothing retrieval and attribute prediction," in *Proc. ACM Int. Conf. Multimedia Retr.*, 2016, pp. 313–316.

[20] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "DeepFashion: Powering robust clothes recognition and retrieval with rich annotations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1096–1104.

[21] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 248:1–248:16, Oct. 2015.

[22] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, "Pose guided person image generation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 406–416.

[23] L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele, and M. Fritz, "Disentangled person image generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 99–108.

[24] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2794–2802.

[25] S. Meister, J. Hur, and S. Roth, "Unflow: Unsupervised learning of optical flow with a bidirectional census loss," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 7251–7259.

[26] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: http://arxiv.org/abs/1411.1784

[27] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 2018, *arXiv:1802.05957*. [Online]. Available: http://arxiv.org/abs/1802.05957

[28] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi, "EdgeConnect: Generative image inpainting with adversarial edge learning," 2019, *arXiv:1901.00212*. [Online]. Available: http://arxiv.org/abs/1901.00212

[29] N. Neverova, R. Alp Guler, and I. Kokkinos, "Dense pose transfer," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 123–138.

[30] A. Raj, P. Sangkloy, H. Chang, J. Hays, D. Ceylan, and J. Lu, "SwapNet: Image based garment transfer," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2018, pp. 679–695.

[31] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha, "Unsupervised deep learning for optical flow estimation," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1495–1501.

[32] I. Rocco, R. Arandjelovic, and J. Sivic, "Convolutional neural network architecture for geometric matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6148–6157.

[33] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* New York, NY, USA: Springer, 2015, pp. 234–241.

[34] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[35] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "Animating arbitrary objects via deep motion transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 2377–2386.

[36] A. Siarohin, E. Sangineto, S. Lathuilière, and N. Sebe, "Deformable GANs for pose-based human image generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3408–3416.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[38] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2432–2439.

[39] Q. Sun, L. Ma, S. Joon Oh, L. V. Gool, B. Schiele, and M. Fritz, "Natural and effective obfuscation by head inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5050–5059.

[40] B. Wang, H. Zheng, X. Liang, Y. Chen, L. Lin, and M. Yang, "Toward characteristic-preserving image-based virtual try-on network," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 589–604.

[41] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8798–8807.

[42] Y. Wang, Y. Yang, Z. Yang, L. Zhao, P. Wang, and W. Xu, "Occlusion aware unsupervised learning of optical flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4884–4893.

[43] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[44] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, vol. 2, 2003, pp. 1398–1402.

[45] Z. Wu, G. Lin, Q. Tao, and J. Cai, "M2E-try on net: Fashion from model to everyone," 2018, *arXiv:1811.08599*. [Online]. Available: http://arxiv.org/abs/1811.08599

[46] L. Xu *et al.*, "FlyCap: Markerless motion capture using multiple autonomous flying cameras," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 8, pp. 2284–2297, Aug. 2018.

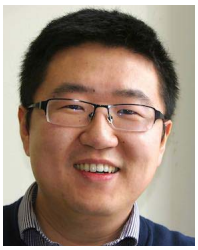
[47] H. Zhang *et al.*, "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 5907–5915.

[48] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[49] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2223–2232.

Haitian Zheng (Student Member, IEEE) received the B.Sc. and M.Sc. degrees in electronics engineering and informatics science from the University of Science and Technology of China, under the supervision of Prof. Lu Fang, in 2012 and 2016, respectively. He is currently pursuing the Ph.D. degree with the Computer Science Department, University of Rochester, under the supervision of Prof. Jiebo Luo. His research interests include computer vision and machine learning.

Lele Chen (Student Member, IEEE) received the B.S. degree in computer science from Donghua University in 2016 and the M.S. degree in computer science from the University of Rochester in 2018. He is currently pursuing the Ph.D. degree under the supervision of Prof. Chenliang Xu in URCS. His research interests include multimodal modeling and video object detection/segmentation.



Chenliang Xu (Member, IEEE) received the B.S. degree in information and computing science from the Nanjing University of Aeronautics and Astronautics in 2010, the M.S. degree in computer science from SUNY Buffalo in 2012, and the Ph.D. degree in computer science from the University of Michigan in 2016. He is currently an Assistant Professor with the Department of Computer Science, University of Rochester. He has authored more than 30 peer-reviewed articles in venues, such as IJCV, CVPR, ICCV, ECCV, IJCAI, and AAAI on topics of his research interest including computer vision and its relations to natural language, robotics, and data science. He was a recipient of multiple NSF awards, including the BIGDATA 2017, CDS&E 2018, and IIS Core 2018, the University of Rochester AR/VR Pilot Award in 2017, the Tencent Rhino-Bird Award in 2018, the Best Paper Award at Sound and Music Computing in 2017, and the Open Source Code Award in CVPR 2012. He co-organized the CVPR 2017 Workshop on Video Understanding and has served as a PC member and a regular reviewer for various international conferences and journals.

Jiebo Luo (Fellow, IEEE) has been a Professor of Computer Science with the University of Rochester since 2011 after a prolific career of over 15 years with Kodak Research. He has authored over 400 technical articles and holds over 90 U.S. patents. His research interests include computer vision, NLP, machine learning, data mining, computational social science, and digital health. He is a fellow of ACM, AAAI, SPIE, and IAPR. He has served on the Editorial Boards of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON BIG DATA, *Pattern Recognition*, *Machine Vision and Applications*, and *ACM Transactions on Intelligent Systems and Technology*. He has served as the Program Co-Chair of the ACM Multimedia 2010, IEEE CVPR 2012, ACM ICMR 2016, and IEEE ICIP 2017, and the General Co-Chair of ACM Multimedia 2018. He is the Editor-in-Chief of the IEEE TRANSACTIONS ON MULTIMEDIA for the period of 2020–2022.