ClothFlow: A Flow-Based Model for Clothed Person Generation

Xintong Han Xiaojun Hu Weilin Huang* Matthew R. Scott
Malong Technologies, Shenzhen, China
Shenzhen Malong Artificial Intelligence Research Center, Shenzhen, China
{xinhan,xiahu,whuang,mscott}@malong.com

Abstract

We present ClothFlow, an appearance-flow-based generative model to synthesize clothed persons for posed-guided person image generation and virtual try-on. By estimating a dense flow between source and target clothing regions, ClothFlow effectively models the geometric changes and naturally transfers the appearance to synthesize novel images as shown in Figure 1. We achieve this with a three-stage framework: 1) Conditioned on a target pose, we first estimate a person semantic layout to provide richer guidance to the generation process. 2) Built on two feature pyramid networks, a cascaded flow estimation network then accurately estimates the appearance matching between corresponding clothing regions. The resulting dense flow warps the source image to flexibly account for deformations. 3) Finally, a generative network takes the warped clothing regions as inputs and renders the target view. We conduct extensive experiments on the DeepFashion dataset for pose-guided person image generation and on the VITON dataset for the virtual try-on task. Strong qualitative and quantitative results validate the effectiveness of our method.

1. Introduction

Pose-guided person generation [28] is of great importance in a plethora of real-world applications, especially for fashion industry where customers or stylists wish to transfer clothing from one person to another. Recent advances in generative networks for image-to-image translation inspired researchers to tackle this problem by feeding a source image and a target pose as input, and then synthesizing the target image [28, 31, 29]. Yet, the non-rigid nature of clothes might cause drastic deformations and severe occlusions which cannot be properly handled [18], thus limiting their performance on rendering clothing details (*e.g.*, patterns, graphics, logos) in the target view.

To overcome this issue, methods of two different paradigms are used to take the geometric deformation into consideration for better appearance transfer, namely deformation-based methods and DensePose-based methods. Deformation-based methods [14, 39, 36, 4] estimate a transformation, either affine or thin plate spine (TPS), to deform the source image pixels or CNN feature maps to deal with the misalignment introduced by pose differences. However, despite great improvements have been achieved by these two geometric modeling techniques, they only have limited degrees of freedom (e.g., 6 for affine and $2 \times 5 \times 5$ for TPS as in [39]), which leads to inaccurate and unnatural transformation estimations when large geometric changes occur.

Recently, a few approaches [30, 12, 42] take DensePose [1] descriptor as inputs instead of tradition 2D keypoints for pose-guided person generation. DensePose is able to map human pixels of a 2D image to the 3D human body surface, allowing it to convey 3D geometry information of the body. This makes it much easier to obtain the texture correspondence between source and target images even with large spatial deformation. However, warping 2D image textures to the predefined surface-based coordinate system further introduces artifacts. For example, holes can be produced in positions that are invisible in the source image, which need to be addressed by complicated texture inpainting algorithms. At the same time, as estimating DensePose is highly challenging, the synthesized results are usually affected by the performance of a DensePose estimator. Thus the DensePose transferred results might look less photorealistic than the deformation-based methods [30] .

To address the problems in existing methods, we propose ClothFlow, a flow-based generative model to accurately estimate the clothing deformation between source and target images for better synthesizing clothed person. Specifically, ClothFlow consists of 3 stages as shown in Figure 2:

(1) A **conditional layout generator** first predicts the target human body segmentation layout conditioned on the target pose. This disentangles the generation of shape and appearance, allowing ClothFlow to generate more spatially coherent results.

*Weilin Huang is the corresponding author.

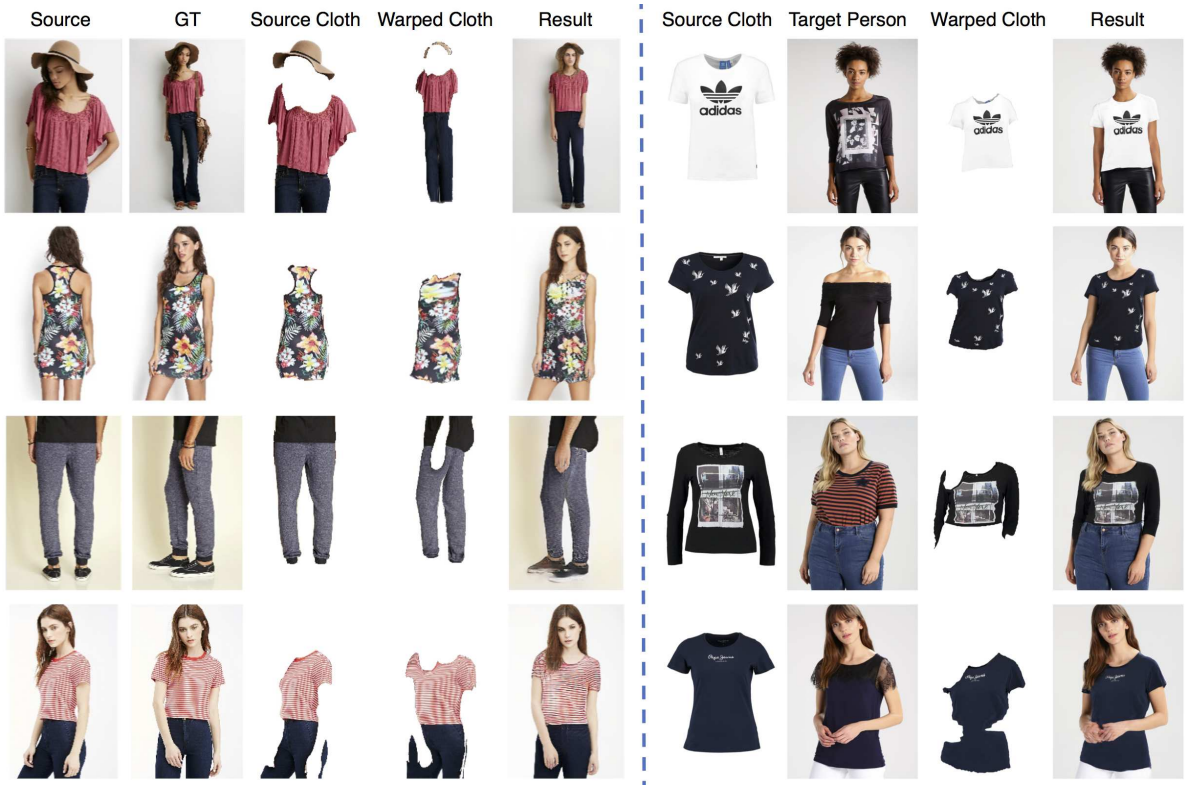

Figure 1: **Results of ClothFlow. Left: pose-guided person image generation. Right: virtual try-on.** ClothFlow warps the source clothing regions conditioned on the desired pose. The warped clothing accurately accounts for the geometric changes between the source and target image, and also addresses the occlusion (*e.g.*, when arm and hair occlude the clothes) and partial observability (*e.g.*, stretching pants region in the first example and dress in the second example). As a result, photorealistic images with detailed clothing patterns can be generated with the warped clothes as input.

(2) The generated layout serves as an input to our **clothing flow estimation stage**, which predicts the appearance flow [45] (*i.e.*, 2D coordinate vectors indicating which pixels in the source image can be used to synthesize the target) from the source clothes to those of the target. Source clothing regions are then warped according to this estimated flow to account for geometric deformation. The predicted appearance flow offers accurate estimation of the visual correspondences and helps seamlessly transfer the source clothing regions to synthesize the target image.

(3) Finally, a **clothing preserving rendering stage** synthesizes the target image with a generative network [34] while trying to preserve details from the warped source clothing regions.

Our method can be regarded as a deformation-based method. However, in contrast to most deformation-based methods utilizing a geometric transformation with few degrees of freedom, ClothFlow estimates a dense flow field (*e.g.*, $2 \times 256 \times 256$) allowing for high flexibility and accuracy when capturing the spatial deformations. Differing from DensePose-based methods that explicitly utilize 3D body surface to transfer textures, we implicitly capture the geometric transformation through approximating the target clothing regions by warping that of the source image.

ClothFlow makes the following main contributions:

- We precisely predict an appearance flow field that aligns the source and target clothing regions in a cascaded manner. In each cascaded stage, a feature warping module progressively improves the estimation from previous stages and better approximates the desired spatial deformation.
- Evaluated on DeepFashion [27] dataset, ClothFlow synthesizes more realistic pose-guided images by better preserving detailed clothing textures compared to state-of-the-art methods. We further demonstrate the effectiveness of ClothFlow with promising results achieved on VITON dataset [14] for virtual try-on task.

2. Related Work

Warping-based Image Matching and Synthesis. Spatial transformer networks [19], allowing CNNs to predict a spatial transformation, have inspired many recent works to semantically warp one object to another [23, 33] or warp an

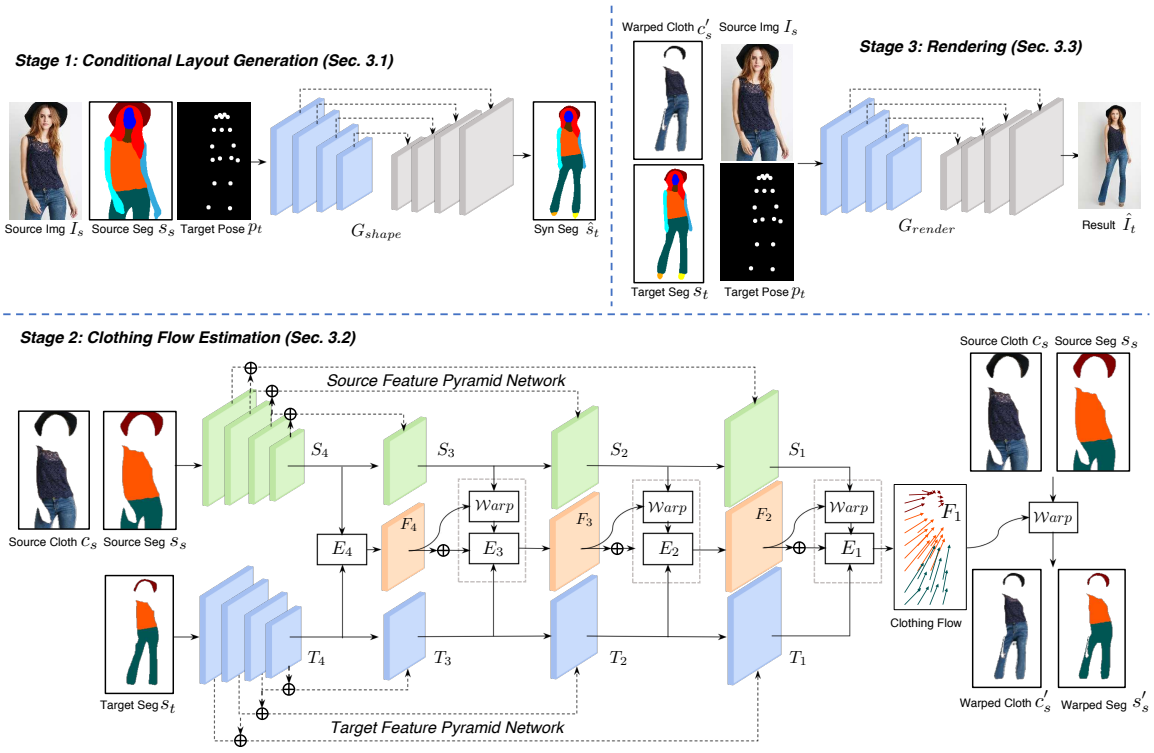

Figure 2: **Framework of ClothFlow for pose-guided person image generation.** ClothFlow has three stages: **Stage 1:** A conditional layout generator hallucinates the target segmentation map. **Stage 2:** [1] Two feature pyramid networks encode geometric information and progressively refine the estimation of clothing flow (the color of arrowed lines indicates its clothing category) between the source and the target in a cascaded warping manner. The resulting flow is then used to warp the source clothes to eliminate the misalignment. **Stage 3:** The warped source clothing together with other guidances synthesize the final result. Note that in Stage 2 and 3 we use the target segmentation map s_t for training, while during inference, we use the synthesized target segmentation $\hat{s}_t$ since s_t is not available.

image to synthesize novel views [45, 46]. In this paper, we also aim to learn an appearance flow [45] to warp a source clothing to the corresponding regions in a target view. It is worth noting that existing methods [21, 46, 45] are usually applied to rigid objects where dense correspondences and visibilities are easy to estimate, while the problem we are addressing is more challenging: clothing regions are often highly non-rigid, and there is no clear correspondence between source and target.

Pose-guided Person Generation. Originally introduced by [28], pose-guided person image generation has spurred a growing interest. Ma *et al*. utilized a two-stage image-to-image translation network [18] to generate target image with the guidance of the source appearance and the target pose in a coarse-to-fine fashion. The authors further improved the results by separating the process of generating pose, foreground and background [29]. Recently, Esser *et al*. [8] disentangled the pose and appearance generation with a variational U-NET. However, these methods are unaware of the spatial deformation between the source and the target, and thus fail to generate perceptually convincing results with large pose discrepancies.

Recent work [36, 4, 2, 12, 44] transformed the pixels or feature maps of the source image to align with the target view, which, to some extent, eliminates the appearance discrepancy and enhances the synthesizing quality. Similarly, our method also warps the source clothing pixels, but by predicting a highly refined appearance flow for clothing regions and thus is capable of accurately modeling large geometric changes as well as occlusions. On the other hand, DensePose [1] descriptor contains much richer information than 2D keypoints, and can mitigate the challenge of modeling such spatial deformation [30, 12, 42]. In the experiments, we show that our method achieves better qualitative transfer results, which can be further improved when combined with DensePose.

Virtual Try-On. Utilizing deep generative models for virtual try-on or garment transfer [14, 20, 39, 31, 5] also requires modeling the deformation between the clothing region in the source image (*e.g*., a product image) and the corresponding region in the target person. However, most existing methods like [14, 39] assume this deformation can be modeled by a thin plate spine (TPS) transformation. But TPS transformation can only models limited geometric

changes and might unnaturally deform the source clothing; instead, we show that our method can be easily modified to take the clothing product image as input and warp it more naturally and seamlessly onto the target person.

Optical Flow Estimation. Our clothing flow estimation is also related to optical flow estimation for videos [17, 16, 32, 38, 7], where they usually learn a Siamese network, by taking two consecutive video frames as inputs, and then warp the raw pixels or features of the first frame to the second one. We draw inspirations from optical flow estimation to predict the clothing flow between the source and target images. There are two main differences between ClothFlow and optical flow estimation. First, we use two separate feature pyramid networks—one for modeling the source clothing appearance and the other one to capture the target information that guides the prediction of appearance flow. Further, we propose a different loss that uses the conditional parsing generated in the first stage to ensure structural and visual coherence of the synthesized image.

3. ClothFlow

Given a source person image I_s and a target pose p_t, the goal of ClothFlow is to synthesize an image of target person I_t, whose pose is p_t with the same appearance as I_s. One key desideratum of a pose-guided image synthesis system is to preserve the texture details in the source image. A straightforward approach is to feed I_s and p_t in an image-to-image translation network like pix2pix [18] or PG^2 [28] to reconstruct I_t. Such approach, although can synthesize realistic skin regions, fails to preserve clothing details due to the lack of considering the spatial deformation induced by pose changes. Driven by this observation, we aim to explicitly estimate an appearance flow [45] for all clothing regions, which we term as *clothing flow*. The clothing flow is a 2D dense flow field specifying which pixels in the source image could be redirected to reconstruct the target image.

Figure 2 summarizes our framework. First, in the conditional layout generation stage (Sec 3.1), we generate the target semantic layout conditioned on the source image, the source semantic map and the target pose. The target semantic layout is then used to guide the warping of the source clothing region in a cascaded network for clothing flow estimation (Sec 3.2). Finally, in the rendering stage, we use the warped source clothes together with other conditions (*i.e.*, source image, target poses and target semantic layout) to synthesize the image of target person (Sec 3.3).

3.1. Conditional Layout Generation

To synthesize a person image, a good practice is to first predict a semantic layout that poses structural constraints for the generation of appearance [4, 47, 13, 25]. We follow this line of work to synthesize the target layout, followed by the generation of clothing details.

To obtain the person pose and layout representations used for training our target layout generator, we utilize an off-the-shelf pose estimator [3] and human parser [11]. More specifically, the pose estimator predicts a set of 2D coordinates of person keypoints, which are then converted to a heatmap where each channel captures the spatial information of a keypoint [29]. For the person layout representations, we follow [47, 6, 13] and encode the layout information as a multi-channel binary map, such that each channel indicates a semantic segmentation of a specific human part. We denote the segmentation map of the source image and the target image as s_s and s_t, respectively.

As shown in Figure 2, the target layout generation network G_{layout} has an encoder-decoder architecture, which takes the source image I_s, the source semantic layout s_s, and the target pose p_t as input, and estimates the target semantic layout $\hat{s}_t = G_{layout}(I_s, s_s, p_t)$ by minimizing the pixel-wise cross entropy loss between s_t and $\hat{s}_t$. Note that directly generating texture details, which often requires one to explicitly model geometric transformation, is more challenging than estimation of the target layout. We solve the problem of person image generation in a coarse-to-fine manner by first predicting the target layout, which severs as an intermediate guidance that is helpful in obtaining a more accurate appearance flow, and also enforces structural constraints during the generation of clothing textures.

3.2. Cascaded Clothing Flow Estimation

The predicted target layout provides important clues on understanding the spatial transform of each clothing item in the target domain. Thus we explore this information to estimate a dense appearance flow (with size of $2 \times H \times W$ where H and W denote the image size) between the clothing regions in the source and target images. Directly estimating this appearance flow is difficult due to the fact that clothes are highly deformable with large misalignment existing between the source and target images. Inspired by recent approaches that employ pyramidal architectures for gradually refining the estimation of optical flow for videos [16, 17, 32], we propose a cascaded warping network whose framework is summarized at the bottom of Figure 2. Note that in this work, we only focus on modeling the flow of clothing regions (*e.g.*, hats, pants, tops, dresses), while modeling the skin regions is more straightforward and has already been handled by a vanilla generative model.

Dual Feature Pyramid Networks. The cascaded clothing flow estimation model contains two feature pyramid networks (FPN) [26]—a source FPN and a target FPN. More specifically, taking the source clothing regions c_s and the source clothing segmentation map s_s [1] as inputs, the source FPN consists of N encoding layers where each layer has

[1] In stage 2, we redefine the notation s_s and s_t to represent the clothing semantic map instead of the whole image.

a downsample convolution with a stride of 2 followed by one residual block [15]. The features output by these encoding layers are used to build the source feature pyramid in the same fashion as in [26], resulting a set of features $\{S_1, S_2, ..., S_N\}$. We set $N = 5$ in our experiments but illustrate the case when $N = 4$ in Figure 2 for simplicity. Similarly, the target FPN has the same network architecture except that the input is the target semantic layout s_t, and yields pyramidal features $\{T_1, T_2, ..., T_N\}$. Note that two FPNs do not share weights because they encode features from different modalities, which is different from the way to estimate optical flow [38] or object matching [23]. Then the extracted pyramidal features will be used to estimate the clothing flow from source clothes c_s to target clothes c_t in a cascaded manner.

Clothing Flow Estimation. The estimation of clothing flow starts from the pyramidal features with the lowest resolution. We feed the concatenated S_N and T_N into a convolutional layer (denoted as E_N), to produce the initial clothing flow F_N. Then, for the features in a higher-level pyramid, we warp the source features conditioned on F_N and refine F_N by predicting the residual flow with a subsequent convolutional layer E_{N-1}. Formally,

$$F_N = E_N([S_N, T_N]), \tag{1}$$

$$F_{n-1} = \mathcal{U}(F_n) + E_{n-1}([\mathcal{W}(S_{n-1}, \mathcal{U}(F_n)), T_{n-1}]), \tag{2}$$

where $n = N, N-1, ..., 2$. $\mathcal{U}(\cdot)$ is a $\times 2$ nearest-neighbor upsampling and $\mathcal{W}(S, F)$ denotes warping feature map S according to flow F using bilinear interpolation, which enables optimization with back-propagation during training [19]. Finally, the last clothing flow F_1 is used to generate a warped source clothing image $c'_s = \mathcal{W}(c_s, \mathcal{U}(F_1))$. Intuitively, the network first learns a rough clothing flow between the source and target with high-level CNN features. Then, warping the source features at each pyramid level eases the process of directly modeling large misalignment and significant deformation that usually occur in clothing transfer. The warped source features are used to estimate a residual flow for refinement of the rough flow in the previous level. This process continues until the network generates the finest flow that helps align small displacements (*e.g.*, logos or graphics) between source and target clothes.

Since we encourage the visual appearance of warped clothing c'_s to be the same as the target one c_t, a perceptual loss [22] between c'_s and c_t can be minimized as:

$$L_{perc}(c'_s, c_t) = \sum_{l=0}^{5} \lambda_l ||\phi_l(c'_s) - \phi_l(c_t)||_1, \tag{3}$$

where $\phi_l(I)$ is the l-th feature map of image I in a VGG-19 [37] network pre-trained on ImageNet with $\phi_0(x) = x$ denoting pixel L_1 loss.

However, only minimizing $L_{perc}(c'_s, c_t)$ may produce inaccurate warping when different clothing items have similar visual patterns, making it hard for the network to determine their boundaries and introducing undesired misalignment. To address this issue, we further design a structure loss to enforce structural constraints of the warped clothing regions. More specifically, the source semantic segmentation map is also warped according to the estimated clothing flow: $s'_s = \mathcal{W}(s_s, \mathcal{U}(F_1))$ and we minimize:

$$L_{struct}(s'_s, s_t) = \sum_i \mathbb{1}(s_{s,i})\mathbb{1}(s_{t,i})||s'_{s,i} - s_{t,i}||_1, \tag{4}$$

where the subscript i denotes the channel index of a segmentation map (*i.e.*, a specific clothing category). $\mathbb{1}$ is an indicator function, and $\mathbb{1}(s_{s,i})\mathbb{1}(s_{t,i})$ specifies if a clothing category i exists both in the source and target images. We modify the perceptual loss to be aware of each clothing region-of-interest (ROI):

$$L_{roi\_perc}(c'_s, c_t, s'_s, s_t) = \sum_{l=0}^{5} \lambda_l \sum_i \mathbb{1}(s_{s,i})\mathbb{1}(s_{t,i})||\phi_l(s'_{s,i} \odot c'_s) - \phi_l(s_{t,i} \odot c_t)||_1, \tag{5}$$

which guides our model to focus on warping the texture specific for each ROI. Consequently, each warped clothing will not be affected by other regions or background, yielding a more coherent warping result.

Flow Regularization. Since appearance flow is dense and has a high dimension of freedom, our clothing flow estimation network allows pixel-to-pixel matching between the source and target clothing regions, leading to a better estimation of geometric changes, which is the key to generate photorealistic results. However, using dense flows usually presents unappealing artifacts without proper regularization, thus we further introduce a total variation loss that regularizes the estimated flow field to enforce smoothness:

$$L_{smt} = \sum_{n=1}^{N} ||\nabla F_n||_1, \tag{6}$$

which is similar in spirit to the regularization term in TV-L1 method [43, 9] for estimating optical flow. Finally, the whole objective function of our cascaded warping network is presented as:

$$L_{flow} = L_{roi\_perc} + \lambda_{struct} L_{struct} + \lambda_{smt} L_{smt} \tag{7}$$

with λ balancing different losses. On the left of Figure 1, we illustrate some examples of the warped clothing regions. The results demonstrate strong robustness to occlusions, partial observability and large deformations.

Discussion. Our flow estimation network predicts more accurate deformations than existing methods [39, 4, 33] which compute a thin plane spline (TPS) transformation. On one hand, the TPS transformation has much fewer transformation parameters and fails to model highly non-rigid transformations. On the other hand, they estimate the transformation in a late stage of a feature encoder (*e.g.*, 4×4 feature map in the $6^t h$ conv. layer), and hence discard the low-level information in the early stages which is essential to align clothing fine details. Besides, several recent works estimate the transformations in a non-learnable fashion [2, 14, 36]—they use keypoints or clothing masks to compute the parameters of a transformation, which is unaware of the appearance correspondence and can only roughly align textures.

Moreover, unlike most optical flow estimation methods [17, 16, 7, 38] that need to search matching features in a local range, ClothFlow estimates the clothing flow with the feature extracted on the whole image and does not struggle to model long-range correspondence or partial observability. Also, they usually require to obtain a computationally expensive cost volume, but ClothFlow achieves satisfactory performance with one conv layer E_i to predict the flow at each pyramid level.

3.3. Clothing Preserving Rendering

At the final stage of ClothFlow, we simply take the warped clothes c'_s with other guidances, including source image I_s, target semantic layout s_t, and target pose p_t, to produce our final result $\hat{I}_t$ using an encoder-decoder generative network [34], as shown in Figure 2. Two standard losses are combined for generating a high-quality result $\hat{I}_t$:

$$L_{render} = L_{perc} + L_{style}. \tag{8}$$

where L_{perc} is the perceptual loss between I_t and $\hat{I}_t$ as defined in Eqn. 3. $L_{style} = \sum_{l=1}^{5} \gamma_l ||\mathcal{G}_l(\hat{I}_t) - \mathcal{G}_l(I_t)||_1$, is the style loss [10, 22] widely used in style transfer tasks to match the style information between two images. $\mathcal{G}_l$ is the Gram matrix [10] for the l-th layer in the pre-trained VGG network. We do not use an adversarial loss because we found that adding the style loss guides the rendering network to directly learn texture details in the warped clothes, which is good enough to generate reasonable results.

Note that during training, we use the ground truth target semantic map s_t to train the second and third stages of our ClothFlow (*i.e.*, input to target feature pyramid network and rendering network). At test time, as s_t is not accessible, the synthesized conditional semantic map $\hat{s}_t$ will serve as input to the two networks.

3.4. Virtual Try-on

Given a product clothing image and a person wearing different garments, the goal of virtual try-on [14] is to synthesize the product item onto the target person with his/her pose and identity preserved as shown on the right of Figure 1. Note that the virtual try-on [14] task is essentially very similar to pose-guide person image generation—they both target to synthesize a novel image that has the appearance of a source image with the pose of a target person. Therefore, we demonstrate that ClothFlow, with a small modification, can tackle this problem. In specific, we treat the product image as the source image and the person's pose as the target pose in ClothFlow, and obtain the clothing flow between the product image and the corresponding region on that person. Following CP-VTON [39], a composition mask is applied with an L_1 loss encouraging the synthesized virtual try-on image to preserve clothing details in the warped product image, and the style loss in ClothFlow is removed for fair comparison. We keep the original pants regions as suggested in the supplementary material of VITON [14]. Experiments illustrate that ClothFlow warps the product image more seamlessly on the target person, and renders try-on results with fewer artifacts.

4. Experiments

4.1. Data and Experiment Setup

Datasets. We evaluate ClothFlow on DeepFashion dataset [27] for pose-guided person image generation and the VITON [14] dataset for virtual try-on task. The DeepFashion In-shop Clothes Retrieval Benchmark contains 52,712 fashion images of resolution 256×256. Image pairs containing the same person in the same outfit with different poses are used for pose-guided person image generation. Following the original protocol in [28, 36], we use 89,262 pairs for training and 12,000 pairs for testing, making sure there is no overlap between two sets. For VITON, we follow [40] to remove duplicates and clean the train/test splits in the original dataset. As in real-world scenarios, each testing pair contains a product image and a person wearing a clothing item different from the one in the product image.

Implementation Details. We use Adam [24] as optimizer with $\beta_1 = 0.5$, $\beta_2 = 0.999$ and a fixed learning rate of 0.0002. We train our 3 stages for 50K, 200K, 200K iterations, all with a batch size of 12. The conditional layout generator and the rendering generator are U-Net [34] type networks with skip connections. The backbones of two FPNs have similar structure to the encoder of a U-Net. We set the dimension of the pyramidal features to 256 as in [26]. The detailed network structures can be found in the supplementary material. We set $\lambda_{struct} = 10$ and $\lambda_{smooth} = 2$, such that losses in Eqn. 7 are in similar scales.

Evaluation Metrics. We adopt Structural Similarity (SSIM) [41] to measure the patch-wise synthesis accuracy, and Inception Score (IS) [35] to measure the realism. Further, we perform a user study to evaluate the perceptual quality of generated images. Resulted Images of the com-

Methods	Deform	Dense	SSIM	IS
PG^2 [28]	✗	✗	0.762	3.09
DSC [36]	✓	✗	0.761	3.35
VUNET [8]	✗	✗	0.786	3.09
BodyROI7 [29]	✗	✗	0.614	3.23
DPT [30]	✗	✓	0.785	3.61
Soft-Gated [4]	✓	✗	0.793	3.31
CBI [12]	✓	✓	**0.835**	2.92
w/o Layout	✓	✗	0.758	3.63
w/o Flow	✗	✗	0.757	3.71
w/o Flow + TPS	✓	✗	0.758	3.74
w/o Cascade	✓	✗	0.759	3.74
w/o Style	✓	✗	0.756	3.56
ClothFlow	✓	✗	0.760	3.75
ClothFlowDense	✓	✓	0.771	**3.88**

Table 1: **Comparisons in terms of SSIM and IS on DeepFashion.** Deform and Dense denote if the method models deformation and uses DensePose descriptor, respectively.

DSC [36]	VUNET [8]	DPT [30]	CBI [12]	Soft-Gated [4]
80.9%	63.4%	90.2%	69.7%	57.9%

Table 2: **Human preference of ClothFlow against other methods on DeepFashion.** Chance is 50%, higher is better.

pared methods are kindly provided by the original authors or generated using publicly available codes.

4.2. Pose-guided Image Generation

Quantitative Results. We compare ClothFlow with state-of-the-art pose-guided image generators by reporting SSIM and IS in Table 1. We show results of ClothFlow by leveraging a DensePose descriptor [1] (ClothFlowDense). We replace the 2D keypoint heatmap p_t with the DensePose descriptor encoded as a heatmap (UV + one-hot I [30]) in our conditional layout generator and rendering network. We find that ClothFlow achieves comparable quantitative performance to other methods, while ClothFlowDense can further improve the performance by injecting more accurate clues about the target person.

User Study. Since SSIM and IS may not correlate well with image visual quality, we conduct a perceptual study by following [12, 1]. Given two generated images, human raters are asked to choose the one transferring source appearance more realistically. Table 2 shows the percentage of trials where our method is preferred over the others. The results verify the ability of ClothFlow to generate realistic images.

Qualitative Results. Figure 3 demonstrates that ClothFlow generates more perceptually convincing results compared to state-of-the-art methods. In particular, we precisely model the clothing deformation between different views,

Figure 3: **Visual comparisons on pose-guided person image generation.** Please zoom in for details.

Figure 4: **Visual results when removing different component in ClothFlow.** Please zoom in for details.

and transfers clothing details more naturally. ClothFlow is free of the adversarial artifacts appeared in most of adversarial methods [1, 12, 4, 36].

Ablation Study. To evaluate the contributions of key components, we compare ClothFlow with the following ablations: *w/o Layout*: removing the conditional layout generator and replacing target semantic layout in the input of the other stages with the target keypoint heatmap; *w/o Flow*: removing stage 2 in ClothFlow and the warped clothing region in the rendering stage; *w/o Flow + TPS*: predicting a TPS transformation instead of clothing flow in stage 2 as [4, 39, 6]; *w/o Cascade*: estimating clothing flow only with T_1 and S_1 without cascades; *w/o Style*: removing style loss in stage 3. From Table 1, we can find that, firstly, all these baselines have similar SSIM which is also observed in related papers [4, 36, 12] as SSIM only roughly measures local similarities. Secondly, adding layout as an intermediate guidance and style loss slightly improve the realism.

In Figure 4, we further illustrate the importance of these components with a visual example with a large non-rigid deformation. We observe that (1) compared to *w/o Layout*, our method generates structurally realistic warping and rendering results by predicting a conditional semantic layout; (2) without our cascaded clothing flow estimation, predict-

Methods	Warp-SSIM	Mask-SSIM	Human
VITON [14]	0.779	0.786	87.3%
CP-VTON [39]	0.806	0.792	81.2%
w/o Cascade	0.833	0.802	69.6%
ClothFlow	**0.841**	**0.803**	-

Table 3: **Quantitative comparisons** in terms of Warp-SSIM, Mask-SSIM and how often ClothFlow is preferred in the user study (Human) on the VITON dataset.

ing a TPS transformation and directly estimating a dense flow field both fail to warp textures on two legs; (3) detailed textures are missing if we do not warp the clothing region (*w/o Flow*) or remove the style loss (*w/o Style*), which is consistent with their quantitative performance.

4.3. Virtual Try-on

We further evaluate ClothFlow on virtual try-on task. In contrast to pose-guided image generation that mainly focuses on transferring textures, finer details like logos, graphics are desired in virtual try-on applications, making it more important to obtain an accurate estimation of spatial deformations. We present experimental results in Table 3 and Figure 5 by comparing with state-of-the-art virtual try-on networks [14, 39], as well as our *w/o Cascaded* baseline. The other baselines are not considered because 1) *Layout* and *Flow* have been well studied in [14, 39] and are necessary recipes to realistic synthesis; 2) *w/o Flow + TPS* is very similar to CP-VTON and has almost identical performance; 3) we do not use style loss for virtual try-on.

As test image pairs are shuffled to ensure that the target person wears a different item from the source product image, we do not have the ground truth to conduct quantitative comparisons. Instead, we take matched pairs (*i.e.*, a product image and a person wearing the product) to obtain evaluation metrics as in [6]. Moreover, inspired by [29, 36] that only compute SSIM for human pixels to isolate the influence of generating various backgrounds, we compute the SSIM between the warped product image and the ground truth clothing region (Warp-SSIM), as well as the SSIM between clothing regions in the real and generated images (Mask-SSIM), rather than computing SSIM for the whole image since we focus on clothing regions. As a result, Warp-SSIM measures how accurate the warping is, while Mask-SSIM measures how well a method reconstructs the clothing. However, for visual comparisons and the user study, we stick to the original evaluation protocol.

Results presented in Table 3 indicate that ClothFlow (1) significantly improves the warping accuracy, with 0.03 higher Warp-SSIM score than CP-VTON, and (2) is better at synthesizing the desired clothing with the highest Mask-SSIM score. Compared to its own variant, ClothFlow obtains higher performance by gradually warping the pyramidal features and predicting the residual flow in a cascaded fashion. Figure 5 qualitatively compares these methods, from which we can see that our method naturally deforms the clothing image conditioned on the target person and renders try-on results with clothing details preserved.

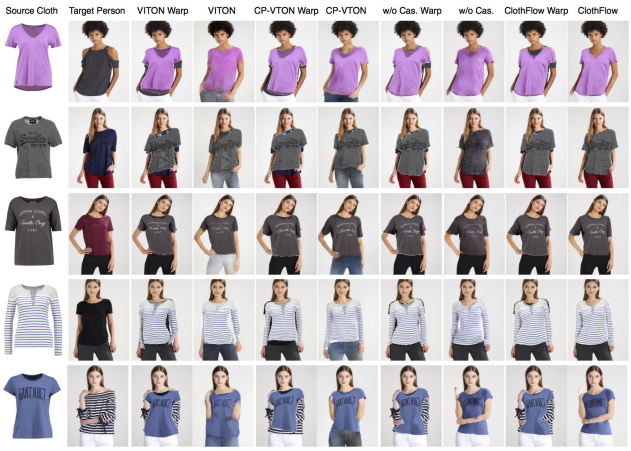

Figure 5: **Virtual try-on comparisons.** We compare ClothFlow to VITON [14] and CP-VTON [39] by visualizing the warping (overlaid on the target person) and try-on results. ClothFlow estimates a more natural and accurate clothing deformation. Interestingly, ClothFlow warps the occluded regions (*e.g.*, unwanted inner collars, regions behind hair and arms) to avoid artifacts that usually occur in other methods. Please zoom in for details.

5. Conclusion

We introduce ClothFlow to model the appearance flow between source and target clothing regions for pose-guided person image generation and virtual try-on. At the core of ClothFlow is a cascaded appearance flow estimation network with a two-stream architecture to progressively warp the source image features and refine the flow prediction. The estimated flow properly handles the geometric deformation as well as occlusions/invisibility between the source and target image, making ClothFlow favorable to other state-of-the-art methods on two standard image synthesizing tasks. We believe the encouraging qualitative results of ClothFlow will inspire computer vision researchers to explore more effective means for capturing the geometric changes in generative models.

Acknowledgement The authors would like to thank Miao Kang for helping create Figure 2, many Malongers for their support and valuable discussion, Haoye Dong and Artur Grigorev for kindly providing the visual results of other compared methods.

References

[1] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018. 1, 3, 7

[2] Guha Balakrishnan, Amy Zhao, Adrian V Dalca, Fredo Durand, and John Guttag. Synthesizing images of humans in unseen poses. In *CVPR*, 2018. 3, 6

[3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 4

[4] Haoye Dong, Xiaodan Liang, Ke Gong, Hanjiang Lai, Jia Zhu, and Jian Yin. Soft-gated warping-gan for pose-guided person image synthesis. In *NeurIPS*, 2018. 1, 3, 4, 6, 7

[5] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bowen Wu, Bing-Cheng Chen, and Jian Yin. Flow-navigated warping gan for video virtual try-on. In *ICCV*, 2019. 3

[6] Haoye Dong, Xiaodan Liang, Bochao Wang, Hanjiang Lai, Jia Zhu, and Jian Yin. Towards multi-pose guided virtual try-on network. In *ICCV*, 2019. 4, 7, 8

[7] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *CVPR*, 2015. 4, 6

[8] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *CVPR*, 2018. 3, 7

[9] Lijie Fan, Wenbing Huang, Chuang Gan, Stefano Ermon, Boqing Gong, and Junzhou Huang. End-to-end learning of motion representation for video understanding. In *CVPR*, 2018. 5

[10] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 6

[11] Ke Gong, Xiaodan Liang, Yicheng Li, Yimin Chen, Ming Yang, and Liang Lin. Instance-level human parsing via part grouping network. In *ECCV*, 2018. 4

[12] Artur Grigorev, Artem Sevastopolsky, Alexander Vakhitov, and Lempitsky Victor. Coordinate-based texture inpainting for pose-guided image generation. In *CVPR*, 2019. 1, 3, 7

[13] Xintong Han, Zuxuan Wu, Weilin Huang, Matthew R Scott, and Larry S Davis. Compatible and diverse fashion image inpainting. In *ICCV*, 2019. 4

[14] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S Davis. Viton: An image-based virtual try-on network. In *CVPR*, 2018. 1, 2, 3, 6, 8

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[16] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *CVPR*, 2018. 4, 6

[17] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 4, 6

[18] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 1, 3, 4

[19] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *NeurIPS*, 2015. 2, 5

[20] Nikolay Jetchev and Urs Bergmann. The conditional analogy gan: Swapping fashion articles on people images. In *ICCVW*, 2017. 3

[21] Dinghuang Ji, Junghyun Kwon, Max McFarland, and Silvio Savarese. Deep view morphing. In *CVPR*, 2017. 3

[22] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 5, 6

[23] Angjoo Kanazawa, David W Jacobs, and Manmohan Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. In *CVPR*, 2016. 2, 5

[24] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[25] Christoph Lassner, Gerard Pons-Moll, and Peter V Gehler. A generative model of people in clothing. In *ICCV*, 2017. 4

[26] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 4, 5, 6

[27] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, 2016. 2, 6

[28] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *NeurIPS*, 2017. 1, 3, 4, 6, 7

[29] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. Disentangled person image generation. In *CVPR*, 2018. 1, 3, 4, 7, 8

[30] Natalia Neverova, Riza Alp Guler, and Iasonas Kokkinos. Dense pose transfer. In *ECCV*, 2018. 1, 3, 7

[31] Amit Raj, Patsorn Sangkloy, Huiwen Chang, Jingwan Lu, Duygu Ceylan, and James Hays. Swapnet: Garment transfer in single view images. In *ECCV*, 2018. 1, 3

[32] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, 2017. 4

[33] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *CVPR*, 2017. 2, 6

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2, 6

[35] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016. 6

[36] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuilière, and Nicu Sebe. Deformable gans for pose-based human image generation. In *CVPR*, 2018. 1, 3, 6, 7, 8

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 5

[38] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018. 4, 5, 6

[39] Bochao Wang, Huabin Zheng, Xiaodan Liang, Yimin Chen, Liang Lin, and Meng Yang. Toward characteristic-preserving image-based virtual try-on network. In *ECCV*, 2018. 1, 3, 6, 7, 8

[40] Kaili Wang, Liqian Ma, Jose Oramas M., Luc Van Gool, and Tinne Tuytelaars. Integrated unpaired appearance-preserving shape translation across domains. *arXiv preprint arXiv:1812.02134*, 2018. 6

[41] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004. 6

[42] Zhonghua Wu, Guosheng Lin, Qingyi Tao, and Jianfei Cai. M2e-try on net: Fashion from model to everyone. *arXiv preprint arXiv:1811.08599*, 2018. 1, 3

[43] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint pattern recognition symposium*, 2007. 5

[44] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris Metaxas. Learning to forecast and refine residual motion for image-to-video generation. In *ECCV*, 2018. 3

[45] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *ECCV*, 2016. 2, 3, 4

[46] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Josh Tenenbaum, and Bill Freeman. Visual object networks: image generation with disentangled 3d representations. In *NeurIPS*, 2018. 3

[47] Shizhan Zhu, Sanja Fidler, Raquel Urtasun, Dahua Lin, and Change Loy Chen. Be your own prada: Fashion synthesis with structural coherence. In *ICCV*, 2017. 4