This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Dressing in the Wild by Watching Dance Videos

Xin Dong[1], Fuwei Zhao[2], Zhenyu Xie[2], Xijin Zhang[1]
Daniel K. Du[1], Min Zheng[1], Xiang Long[1], Xiaodan Liang[2*], Jianchao Yang[1]
[1]ByteDance, [2]Shenzhen Campus of Sun Yat-Sen University
{zhaofw@mail2,xiezhy6@mail2,xdliang328@mail}.sysu.edu.cn
{dongxin.1016,zhangxijin,dukang.daniel,zhengmin.666,longxiang.0,yangjianchao}@bytedance.com

Figure 1. The results of the proposed method on in-the-wild images. Our model is capable of transferring arbitrary garments (e.g., shirts, pants, formal dresses, skirts, down jackets) from a source person image I^s onto a challenging posed query person image I^q presented in real-world backgrounds, generating high-fidelity output image O^q with the query's identity now wearing the source's garments.

Abstract

While significant progress has been made in garment transfer, one of the most applicable directions of human-centric image generation, existing works overlook the in-the-wild imagery, presenting severe garment-person misalignment as well as noticeable degradation in fine texture details. This paper, therefore, attends to virtual try-on in real-world scenes and brings essential improvements in authenticity and naturalness especially for loose garment (e.g., skirts, formal dresses), challenging poses (e.g., cross arms, bent legs), and cluttered backgrounds. Specifically, we find that the pixel flow excels at handling loose garments whereas the vertex flow is preferred for hard poses, and by combining their advantages we propose a novel generative network called wFlow that can effectively push up garment transfer to in-the-wild context. Moreover, former approaches require paired images for training. Instead, we cut down the laboriousness by working on a newly constructed large-scale video dataset named Dance50k with self-supervised cross-frame training and an online cycle optimization. The proposed Dance50k can boost real-world virtual dressing by covering a wide variety of garments under dancing poses. Extensive experiments demonstrate the superiority of our wFlow in generating realistic garment transfer results for in-the-wild images without resorting to expensive paired datasets. [1]

[1]Xiaodan Liang is the corresponding author. The project page of wFlow is https://awesome-wflow.github.io.

1. Introduction

Garment transfer, the process of transferring garments onto a query person image without changing the person's identity, is a central problem in human-centric image generation that promises great commercial potential. However, when getting down to in-the-wild scenarios, general solutions are required that can leverage easily accessible training data, handle arbitrary garments, and cope with complex poses presented in real world environments.

Unluckily, most existing works [10, 11, 13, 32, 42, 46, 47, 49, 51] serve to fit an in-shop garment to a target person by utilizing either pixel flow [12] or TPS transformation [2]. Despite their promise, these methods become less effective at exchanging garments directly between two persons, due to the deficiency of the 2D transformations when faced with large pose variations. Also, previous methods require paired data for training, i.e., a person and its associated garment image, which further leads to laborus collection process. These limitations largely hinder their practical use and raise the need for scalable solutions that can be trained on easily accessible data. [27] takes a step forward by replacing the 2D pixel flow with 3D SMPL [29] vertex flow, allowing person-to-person garment transfer and can address complex poses or severe self-occlusion. However, it is error prone to loose garments that can not be modeled as part of the SMPL surface. Albeit subjects to simple poses, the 2D pixel flow can then again predict more faithful pixel mapping for these challenging loose garments.

Therefore, in this paper, we propose *wFlow* that efficiently integrates respective advantages of the 2D pixel flow and the 3D vertex flow. Based on the wFlow, a robust garment transfer network is developed to tackle the essential challenges on in-the-wild imagery. In particular, we design a self-supervised training scheme that works on easily obtainable dance videos by exploiting cross-frame consistency, getting rid of the hard-to-get paired dataset.

Our insight is that a well-designed flow-based model trained on multi-pose images of the same person, which is easily accessible in dance videos, can generalize well at testing to transfer garments between different persons by adding protected body parts that guides the network to focus on the garment regions. Thus, we collect thousands of single-person dance videos with diverse garment types as the training dataset, and sample from it a plethora of multi-pose person frames to train our model in a similar fashion of pose transfer [24, 34], where the designed wFlow that associates different frames allows us to self-supervise the training procedure without ground truth flow supervision.

To fully exploit the wFlow, we first pass the source and query person representation through a conditional segmentation network, producing a person segmentation that complies with both the source garment and the query pose. Given the predicted segmentation, a pixel flow network is employed to estimate the pixel-wise correspondences between the source and the query images. Thereafter, we compute 3D SMPL vertex flow directly from the inputs and project it to image plane where the pixel flow is also injected to form the proposed wFlow. The warped garment can now be obtained by applying the wFlow on the source garment. Finally, a skip-connected inpainting network leverages the warped garment along with the protected person regions and fuses them with the inpainted query background. Additionally, for garments presented scarcely in training data, we further formulate a cyclic online optimization to enhance the quality of their transfer results. Thanks to the contributory video data and the potent texture mapping ability of the wFlow, our model can handle arbitrary garments and seamlessly transfer them onto challenging posed query persons.

Overall, we present three main contributions:

- We are the first to explore the in-the-wild garment transfer problem. By exploiting a self-supervised training scheme that works on easily accessible dance videos, our model generates surprising results with sharp textures and intact garment shape.

- To facilitate arbitrary garment transfer under complex poses in real-world scenario, we introduce a novel **wFlow** (flow in-the-wild) that integrates both 2D and 3D information along with a cyclic online optimization that further enhances the synthesis quality.

- We construct a new large-scale video dataset called *Dance50k*, containing 50k sequences of dancing people wearing a wide variety of garments, which is useful for the development of human-centric image/video processing not limited to virtual try-on.

2. Related Work

2.1. Image-based Garment Transfer

Due to the great application potential, research on 2D garment transfer has been explored intensively [1, 4, 6, 9, 10, 22, 31, 32, 43, 46]. VITON [13] and CP-VTON [42] are the starting point in this convincing field. Both of them utilize a TPS-based deformation module followed by a texture fusion module to warp and fuse a catalog garment to a query person image. VTNFP [49] and ACGPN [47] inherit the same warping scheme but further introduce the human parsing as synthesis guidance, achieving better delineation at cloth-skin boundary. More recently, PF-AFN [11] gets rid of human parsing by exploiting a appearance flow distillation scheme, generating consistently good results via a simpler student model. Despite their promise, all these methods require a training set consisting of paired images. This limits the scale at which training data can be collected since obtaining such paired images is highly laborious. Also, dur-

ing testing only in-shop garments can be transferred to the query persons with simple poses.

To resolve this, [23, 30, 33, 36–38, 45] extend the paired approaches to their unpaired counterparts that realize garment transfer directly between two persons. Most of these works rely on the powerful conditional generator (e.g., StyleGAN2 [21]) to manipulate the latent garment feature embeddings. However, unlike the flow warping that directly transforms textures, these pure generation models inevitably suffer with recovering complex texture patterns from the low-dimensional latent space, even with the state-of-the-art StyleGAN2. Overall, previous garment transfer methods, are either limited by the short-supplied data or the insufficient GAN inversion. Our method, instead, can not only leverage in-the-wild video data but also yield realistic garment and skin textures. In the next subsection, we will give a brief review on the flow estimation that we adapt for modeling the texture mapping mechanism.

2.2. Human-centric Flow Estimation

Optical flow, defined for representing pixel offsets between adjacent frames in videos [8, 15, 19, 40, 41, 52], has also inspired researchers in the field of human-centric generation for tackling problems such as face hallucination [39], pose transfer [24], and virtual try-on [7, 11, 12]. Without loss of meaning, we rename the optical flow as **2D pixel flow** for human generation, which refers to 2D coordinate vectors indicating which pixels in the source can be used to synthesize the target. ClothFlow [12] and FW-GAN [7] are the firsts to utilize the pixel flow for garment transfer. Thanks to its high degrees of freedom, 2D pixel flow is suitable for dressing both skin-tight and loose garments, however, it tends to fail when faced with large pose variations due to the ignorance of underlying rigid body information. [27], [48] and [24] thus turn to **3D vertex flow** based on 3D SMPL model [29] that is kinematics-aware to cope with complex poses. While outperform on diverse poses, these methods sacrifice the deformation freedom compared to the pixel flow, generating unsatisfactory results of loose clothes. Given clear advantages of both sides, we make the first attempt to leverage both 2D pixel flow and 3D vertex flow to facilitate in-the-wild virtual dressing.

3. Methodology

As shown in Fig. 1, given a source image I^s of a person and a query image I^q, the goal of our method is to generate a synthetic image O^q preserving the query's identity and background but now dressing the source's garment. While training the model with (I^s, I^q, O^q) triplets is straightforward, collecting such a dataset is beyond laborious since O^q is usually unavailable. Instead, we use (I^s, I^t, O^t) where the query (hereafter denoted as *target* for the training phase) and the source are the same person under different poses, which is easily accessible from different frames in dance videos. Fig. 2 illustrates the overall architecture that exploits the designed wFlow in a multi-stage manner described in the following subsections.

3.1. Stage 1: Conditional Person Segmentation

Our training approach in summary is similar to the pose transfer prototype [24, 27]. However, directly extending pose transfer to garment transfer is prone to overfitting, seeing that during training we use multi-pose images of the same person while at testing the query can be arbitrary identity. To provide more reliable synthesis guidance for inference, a Conditional Segmentation Network (CSN) is employed to predict person segmentation layout that conforms to the target shape as well as the source garment, as shown in Fig. 2.(a).

Specifically, with two separate encoders, the CSN first extracts features from two image collections respectively: (1) the 20-channel source person segmentation S^s (obtained by applying [25] on the source image I^s) together with the person representation R^s including a 3-channel RGB image, a 1-channel person mask (obtained by binarizing S^s), a 3-channel densepose (obtained by projecting fitted SMPL mesh [20] to 2D UV space), and a 18-channel body joints (obtained by applying OpenPose [3] on the RGB); (2) the target densepose D^t and body joints J^t. We additionally condition on D^t to give the network the flexibility to learn a rough target shape which can not be easily perceived from the joints J^t. Thereafter the two extracted bottleneck feature will be concatenated and sent to a series of residual blocks followed by a decoder, yielding the target person mask and segmentation layout (M^t, S^t). The formulation of this stage can thus be summarized as:

$$\left(M^t, S^t\right) = f_{csn}\left(R^s \oplus S^s, D^t \oplus J^t\right) \tag{1}$$

which is optimized by the combined L1 (for M^t) and pixel-level cross-entropy (for S^t) loss defined between the predictions and the ground truth from the target frame. The CSN also plays a preparatory role for the following flow estimation stage, where the predicted segmentation can ease the network at capturing spatial structure similarities.

3.2. Stage 2: Pixel Flow Estimation

As proved in [12], the 2D pixel flow mainly reflects structure and texture correspondence between images and thus is object-agnostic, showing its promise of generalization to arbitrary garment types. Motivated by this, we estimate through the PixelFlow Network (PFN) the pixel flow F^p indicating which locations in the target frame the source frame pixels should be mapped to. The inputs of PFN are similar to the CSN, except its target branch additionally receives the predicted segmentations from the CSN.

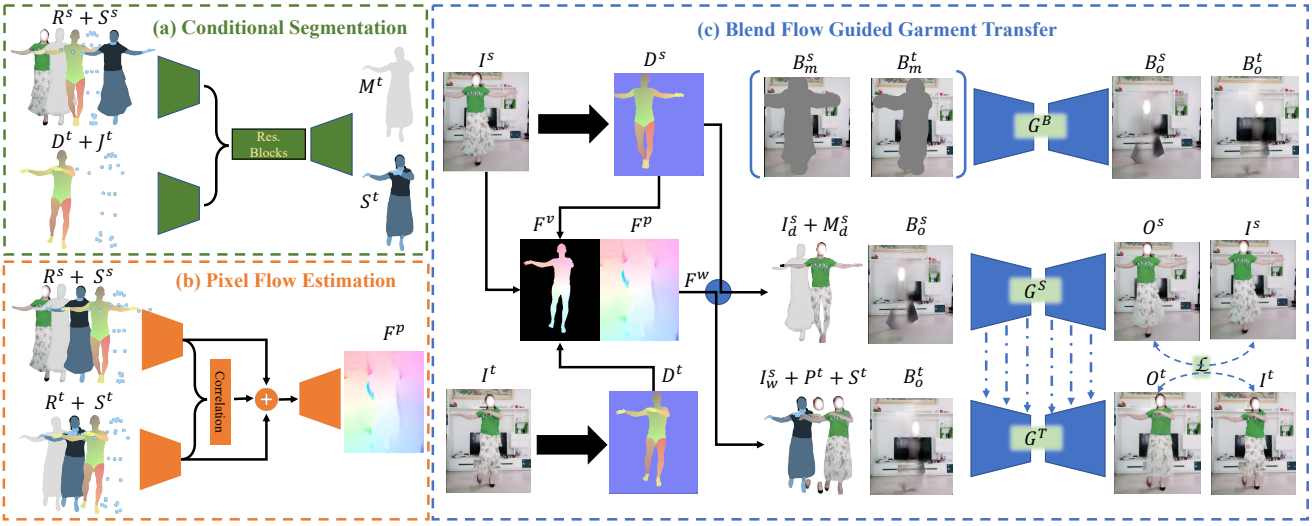

Figure 2. Architecture Overview. Our training pipeline contains: (a) Conditional Person Segmentation (Sec. 3.1): generates the person layouts (M^t, S^t) providing the source garment shape under the target pose; (b) Pixel Flow Estimation (Sec. 3.2): leverages (M^t, S^t) and other person representations to estimate a per-pixel flow map F^p. (c) wFlow-Guided Garment Transfer (Sec. 3.3): computes first the 3D vertex flow F^v from the source and target person frame (I^s, I^t) and fuses it with F^p to form the wFlow F^w. Thereafter the warped person I^s_w is fitted to the G^B-inpainted background B^t_o by a feature distillation generator G^T that incorporates the texture completion ability of the reconstruction generator G^S, producing respectively the output O^t/O^s supervised by I^t/I^s.

To give more details, as depicted in Fig. 2.(b), two identical encoders build the contractive steams of PFN, which extract appearance and structure features from the input groups. It might be conceptually simple to directly contrive pixel-wise correspondences from the concatenated downsampling features through chained deconvolutions, as the related work [12] does. However, in case of large deformation that usually occurs in real world scenes, such a vallina pipeline tends to produce artifacts for the intractable process of the network itself finding the cross-feature association. We hence adopt the feature correlation layer of FlowNetCorr proposed in [8] in-between the downsampling and the upsampling parts to impart the network stronger regulation when correlating discrepant features, i.e., a matching score quantizing the feature similarity between images. Furthermore, we skip-connect the encoding features to the same-level decoding layers for direct high-level feature transmission that speeds up the learning process. In this way, the decoder now gets a concatenation of the two bottleneck features and their correlation tensor as input, and progressively upsample it to the original image size with a layer-wise flow refinement mechanism, i.e., each flow map estimated by a certain decoding layer will be bilinearly upsampled and concatenated with the flow predicted by the next layer to jointly refine the estimation result.

The overall process can now be formulated as

$$F^p = f_{pfn}\left(R^s \oplus S^s, R^t \oplus S^t\right) \tag{2}$$

With F^p, we can map the source frame texture to the target. Note since we train by reposing the same person in different time instance, we thus can self-supervise the flow estimation with cross-frame consistency, i.e., similarity between the mapped texture and the ground truth texture in the target frame. We choose equally weighted perceptual [18] and L1 losses as the cost function for this stage.

The main difference between our PFN and the related ClothFlow [12] are three fold: (1) we additionally leverage the rigid densepose D^t as input to partially neutralize the high degrees of flow freedom, which is especially beneficial for tight-dressed persons; (2) we exploit a correlation layer that provides explicit feature matching guidance while in ClothFlow they implicitly explore this via a cascaded flow rectification network, of which the process as aforementioned is relatively hard to control. (3) The ClothFlow warps each encoding feature according to the estimated flow to account for the feature misalignment which we do not, since this has a risk of accumulating errors if the initial predicted flow is inaccurate. With the aid of feature correlation layer and the rich input information, our PFN is preferable to video data taken in the wild.

3.3. Stage 3: Garment Transfer with wFlow

The core contribution of this stage is the **wFlow**. We will first explain the process of obtaining this novel flow, and then dive into detail of the proposed Garment Transfer Network (GFN) as well as the objective functions.

wFlow. Having the predicted 2D pixel flow from stage 2, we boost its capability by injecting informative 3D SMPL vertex flow. The new blended flow is kinematic-aware that has more pose transfer potential when faced with in-the-

wild scenarios, therefore we name it *wFlow* (i.e., flow in the wild). Specifically, we fit SMPL body mesh to I^s and I^t using [20] respectively, and then project the fitted meshes to the densepose representation (D^s, D^t) in 2D UV space (i.e., image space). Since the SMPL mesh topology is fixed, we can immediately calculate a (2D) vertex flow F^v between D^s and D^t, given the barycentric coordinates of each densepose pixel with respect to its unprojected mesh face. We denote the binary mask derived from the vertex flow map as M^v, and apply the following formulation to obtain the wFlow F^w:

$$F^w = M^v \odot F^v + (1 - M^v) \odot F^p \tag{3}$$

This formulation has two nice properties. First, the vertex flow component has higher priority to guarantee the correctness of texture mapping for rigid body parts. Second, for non-rigid garment deformation, the pixel flow component will show accordingly its mastery. Thus, by Eq. 3, we factorize the texture mapping into articulation and non-rigid deformation, which we would argue is more flexible than solely using either one component. The ablation study on the wFlow in Sec. 4.4 further supports this point.

The F^w warps I^s to I^s_w complying with the target pose, which will then be concatenated with the S^t and the unchanged target person parts P^t (obtained by applying [25] on I^t), making ready the warped person representation (I^s_w, P^s, S^s) for the garment transfer network.

Garment Transfer Network (GTN). As shown in Fig. 2.(c), the GTN has three identical UNet-like generators named G^B, G^S and G^T, where (1) G^B inpaints the source and target backgrounds; (2) G^S reconstructs the source; and (3) G^T synthesizes the pose transfer result during training. Note at testing, the G^T will run garment transfer. Fundamentally, our GTN follows the overall architecture of [27], but adapts it for high fidelity garment transfer especially loose clothes by virtue of the proposed wFlow.

More specifically, the background inpainting generator G^B takes the dilated-masked source and target background (B^s_m, B^t_m) as batched input, and outputs the respectively inpainted backgrounds (B^s_o, B^t_o). Afterwards B^s_o will be added to the inputs of G^S together with the densepose-masked source RGB I^s_d and the source mask M^s, all of which are passed through G^S to reconstruct O^s as close to I^s. Note, since D^s comes from the garment-less SMPL mesh, some parts of loose garments may be masked out in I^s_d. This enforces the G^s to learn texture completion for the exterior part of M^s with respect to I^s_d. Concurrently, the B^t_o concatenated with the warped representation $(I^s_w + P^t + S^t)$ are aggregated by G^T to synthesize the pose transfer result O^t (or, the garment transfer result during testing). In practice, the G^S (resp. G^T) first predicts a fusion mask M^s_f (resp. M^t_f) and a coarse result $\widetilde{O^s}$ (resp. $\widetilde{O^t}$) and then fuse them with B^s_o (resp. B^t_o) to the final result O^s (resp. O^t):

$$O^{s(t)} = \widetilde{O^{s(t)}} \odot M_f^{s(t)} + B_o^{s(t)} \odot (1 - M_f^{s(t)}) \tag{4}$$

It is worth noting that we warp and distillate features of G^S to G^T inspired by [27], but has two main differences from it: (1) the features are now transformed by the blended wFlow instead of the single vertex flow as in [27], which provides non-rigid deformation ability; (2) in [27] they focus on transmitting structural and texture information between two generators, while we aim to distillate the *inpainted* features of G^S to G^T, which can ease the G^T at inference to inpaint textures of loose garment. This is again the reason why G^s needs the capability of texture completion realized by setting its input to M^s instead of D^s used by the other work. Besides, the G^T's dependency on G^S also shows the necessity of leveraging two generators instead of one (G^T-only). Please refer to the supplementary for more detail of this feature distillation operation.

Loss functions. The training losses of GTN are computed against three products: the fusion mask $M_f^{s(t)}$, the reconstructed O^s, and the pose transfer result O^t. In particular, we adopt the BCELoss $\mathcal{L}_{BCE}$ for M^s_f and M^t_f under the regularization of Total Variation constraints [26], which is formulated as (same for M^s_f):

$$\mathcal{L}_m\left(M^t_f\right) = \mathcal{L}_{BCE}\left(M^t_f, M^t\right) + TV\left(M^t_f\right),$$

where the TV loss in detail is:

$$TV(M^t_f) = \sum_{i,j}[(M^t_f(i,j) - (M^t_f(i-1,j)]^2 + [(M^t_f(i,j) - (M^t_f(i,j-1)]^2.$$

As for the reconstructed O^s and the synthesized O^t, we use L1 ($\mathcal{L}_1$) and perceptual loss ($\mathcal{L}_{perc}$)to measure their difference to the ground truth frame images. An adversarial loss based on the Pix2Pix discriminator [16] is further incorporated for G^T, narrowing the distribution gap between the synthesized and the real images. Thus, the total loss of the GTN generators is summarized as:

$$\mathcal{L}^G_{gtn} = \mathcal{L}_m + \mathcal{L}_1 + \mathcal{L}_{perc} + \mathcal{L}^G_{adv},$$

where $\mathcal{L}^G_{adv} = \sum D(O^t, S^t)^2$ in which the D denoting the discriminator and has its own loss

$$\mathcal{L}^D = \sum \left[D\left(O^t, S^t\right) + 1\right]^2 + \sum \left[D\left(I^t, S^t\right) - 1\right]^2.$$

3.4. Cyclic Online Optimization

The different setting of training and testing (pose transfer v.s. garment transfer) makes it challenge to directly confront with arbitrary query images especially those low-resolution or indistinct-foreground ones. We thus introduce an online optimization depicted in Fig. 3 that works

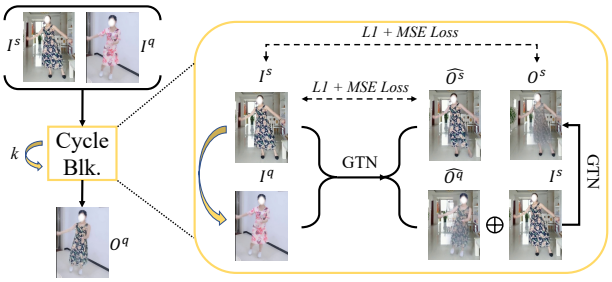

Figure 3. Illustration of the cycle online optimization.

on the cycle consistency to progressively refine the synthesized dressing result whenever the query image quality is unsatisfactory.

In specific, during inference for a pair of source and query images (I^s, I^q), we pass them k times through the carefully designed **Cycle Block**, where k is a tunable parameter that trades off the running time and the refinement degree. During the first pass, we first transfer the garment of I^s to I^q via the GTN, producing the reconstructed $\widehat{O^s}$ and the intermediate try-on image $\widehat{O^q}$. Thereafter the $\widehat{O^q}$ together with the input I^s will be once more processed by the same GTN, but here the transfer direction reversed $(\widehat{O^q} \rightarrow I^s)$, yielding the "dressed-back" image O^s where the "cycle" closed. The combined L1 and MSE Loss for $\widehat{O^s}$ and O^s with respect to I^s is used to guide this cyclic process. However, at the entrance of second pass, the role of I^s and I^q will exchange: I^q now becomes the "source" providing garments while the "query" I^s now wants to dress from I^q. Intuitively, by exchanging garments repeatedly between a given input pair, we want the GTN to overfit on them at inference. In this way, the output try-on result will be progressively refined to high quality with sharper edges and more realistic textures, largely mitigate the problem of dealing with low-quality input query image.

4. Experiments

4.1. Dataset: *Dance50k*

We learn realistic garment transfer by leveraging a collection of real world dance videos scraped from public internet. The dataset, named *Dance50k*, contains $50,000$ single-person dance sequences (about 15s duration) that feature varying poses and numerous garment types. Fig. 5 plots the garment distribution presented in *Dance50k* and the other popular try-on dataset named DeepFashion [28], revealing ours superiority in garment diversity, which is essential for virtual dressing tasks. Note, though the *Dance50k* in its modality is very similar to the TikTok Dataset [17], ours is two orders of magnitude larger than the TikTok (50k v.s. 300), showing the stronger suitability of $Dance50k$ for solving potential human-centric image/video problems that not limit to virtual try-on [2].

4.2. Implementation Details

We follow a multi-stage training regime for faster convergence and stable training. The proposed CSN, PFN and GTN are trained separately for $2.5M$ iterations with a batch size of 4 using the Adam optimizer ($\beta_1 = 0.5$, $\beta_2 = 0.999$), and the learning rate is set to 0.0002. We use Pytorch to implement the pipeline and train it on a single NVIDIA V100 GPU [3]. During training, we uniformly sample 10 frames from each video in *Dance50k* and get a combination of C_{10}^2 candidate (I^s, I^t) pairs. To guarantee validation of the flow direction, a simple assertion is further applied on the candidates to ensure that the number of joints in I^s is more than that in I^t. Consequently, a total of 949623 image pairs are finally used for training in a generalizable manner of pose transfer. While at testing, the identity of query image I^q can be different with the source I^s and the inference is performed in an end-to-end manner of garment transfer. If the cycle online optimization is further needed for some hard queries, we set the k to 20 for best time-quality trade-off.

4.3. Comparison with Baselines

Evaluation metric. To fully compare with baselines, we evaluate the performance in two aspects: (1) Fidelity of the pose transfer results; (2) Realism of the synthesized garment transfer images. We adopt the common GAN metrics to measure them respectively, i.e., the Structural SIMilarity index (SSIM) [44] for pose transfer, the Fréchet Inception Distance (FID) [14] and the Perceptual distance (LPIPS) [50] for garment transfer, as ground truth images are necessary for the SSIM and LPIPS while not for the FID. We further evaluate the accuracy of the generated garment shape by computing the Intersection over Union (IoU) between the generated and the real garment silhouette, where larger IoU means the generated shape is more consistent with the real. Note only loose garments (e.g., skirt, dresses) are taken into account for calculating IoU as tight clothing can be modeled as part of the human skin that presents small IoU variance. We additionally conduct a human evaluation to assess the results (please refer to the supplenmentary for detailed setting of this human evaluation), and besides *Dance50k*, we also report scores on the commonly-used DeepFashion dataset [28].

Qualitative comparison. As shown in Fig. 4, we conduct qualitative comparison on $Dance5k$ and DeepFashion [28], with the other three open source state-of-the-art dressing approaches. ADGAN [30] is incapable of fusing the background and fails to correctly preserve the garment attributes. DiOR [5] is a versatile model that can handle

[2] More detailed description and data examples of *Dance50k* are enclosed in the Supplementary.

[3] We provide the complete model details in the supplementary.

Figure 4. Qualitative comparisons on *Dance50k* (1-3rd rows) and DeepFashion Datset (4-5th rows). The first two columns represent the inputs, while the others are garment transfer results from our method and the other three baselines (LWG [27], ADGAN [30] and DiOR [5]). Our wFlow contains richer foreground and background texture details and more successfully transfer the loose garments. Please zoom in for more details and more visual results are provided in supplementary.

Dataset	Dance50k					DeepFashion				
Method	SSIM ↑	FID ↓	LPIPS ↓	IoU ↑	HE ↑	SSIM ↑	FID ↓	LPIPS ↓	IoU ↑	HE ↑
LWG [27]	0.891	13.080	0.107	0.484	0.271	0.778	59.239	0.225	0.508	0.284
ADGAN [30]	0.765	44.280	0.223	0.289	-	0.643	85.083	0.317	0.590	-
DiOR [5]	0.884	58.073	0.108	0.673	-	0.728	76.068	0.291	0.616	-
wFlow (Ours)	**0.920**	**8.809**	**0.090**	**0.719**	**0.729**	**0.844**	**57.652**	**0.187**	**0.687**	**0.716**

Table 1. Quantitative comparisons to other garment transfer methods, i.e., Liquid Warping GAN [27], ADGAN [30] and DiOR [5]. Note the SSIMs are reported for pose transfer results while FIDs and LPIPSs are for garment transfer. HE here refers to Human Evaluation and since only LWG and our wFlow can generate background, the HE is reported on these two methods excluding the other two baselines.

CO	F^p	F^v	Dance50k				DeepFashion			
			SSIM↑	FID↓	LPIPS ↓	IoU↑	SSIM↑	FID↓	LPIPS ↓	IoU↑
✓	✓	✗	0.920	12.077	0.096	0.685	0.842	63. 136	0.195	0.648
✓	✗	✓	**0.922**	9.455	0.093	0.699	**0.847**	59.214	0.191	**0.687**
✗	✓	✓	0.920	12.106	0.099	0.709	0.806	71.016	0.228	0.672
✓	✓	✓	0.920	**8.809**	**0.090**	**0.719**	0.844	**57.652**	**0.187**	**0.687**

Table 2. The ablation study on the Pixel Flow (F^p), Vertex Flow (F^v) and the cycle online optimization (CO).

pose/garment transfer and texture editing in an end-to-end pipeline, but it underperforms for in-the-wild scenes even with the attention-aided flow estimation [35], producing unrealistic texture distortion. While LWG [27] genenrates reasonable results, it tends to produce blurred body edges when faced with complex poses and can not model loose clothes. With the powerful wFlow, our garment transfer network can not only handle arbitrary clothing but also seamlessly synthesize the garment transfer foreground.

Quantitative comparison. As reported in Table 1, our **wFlow**-based garment transfer network leads all five metrics especially the highlighting FID calculated on the

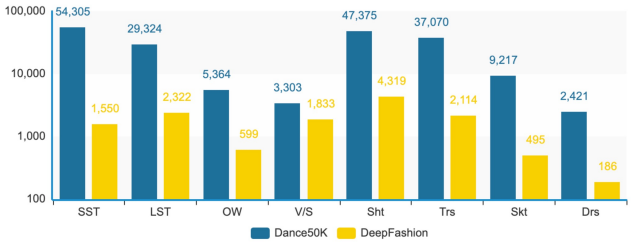

Figure 5. Garment distribution in Dance50k and DeepFashion Dataset. The terms in x-axis respectively denote Short Sleeve Top (SST), Long Sleeve Top (LST), Outwear (OW), Vest/Sling (V/S), Shorts (Sht), Trousers (Trs), Skirts (Skt) and Dress (Drs).

Dance50k dataset. The best FID and LPIPS indicates that our wFlow performs better garment transfer on in-the-wild imagery, without sacrificing the performance of high-fidelity pose transfer. Moreover, the IoU directly shows wFlow can obtain more accurate warping silhouettes, which is crucial for the subsequent texture fusion process. SSIM measures the structural and brightness similarity between pose transferred images and their ground truth. In our implementation, the vertex flow directly transmits source pixels without changing the pixel value, while the pixel flow result is further fused with an estimated alpha blend mask. As a result, injecting pixel flow into vertex flow will slightly change the brightness of transformed pixels, which will lower a bit the SSIM score for skin-tight clothes favored by the vertex flow. This is why SSIM drops with full method configuration. However, as we attend to in-the-wild garment transfer, combining these two types of flow can significantly improve the overall visual performance especially for loose outfits. This is also supported by the visualized results and other quantitative metrics including the human evaluation, as most volunteers appreciate the superiority of our wFlow in recovering sharp texture and preserving intact garment shape.

4.4. Ablation Study

We conduct three ablation studies to analyze the impact of the wFlow and the online Cycle Optimization. The corresponding evaluations are summarized in Table 2. Solely using F^p (resp. F^v) leads to inferior capability of capturing global texture features (resp. human kinematics information), harming either the SSIM or the FID score (Table 2, 1st-2nd row). since the cyclic online optimization helps refining overall quality of the synthesized images, removing it will also affect negatively (Table 2, 3rd row). By leveraging all the three building blocks, the full model (Table 2, 4th row) outperforms all metrics except SSIM reported for pose transfer. With wFlow, the key FID scores (reported for garment transfer) are improved by a large margin on both datasets, showing the success of our purposeful architecture design that pursues the in-the-wild dressing achievement.

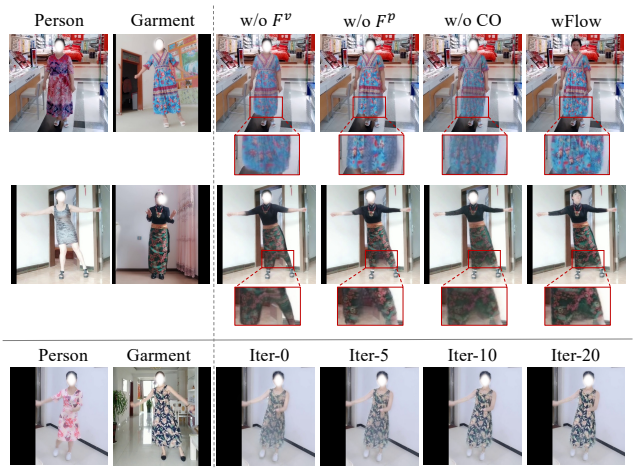

Figure 6. Ablation studies on the wFlow (upper part) and visualized effect of the online cycle optimization (lower part). Please zoom in for more details.

Accompanying the Table 2, the upper part of fig. 6 illustrates consequences of ablating the wFlow. Concretely, sorely using F^p or lacking the online CO procedure usually leads to blur garment texture, while sorely using F^v can not guarantee the consistency of garment texture and accuracy of garment shape. Furthermore, the lower part of fig. 6 demonstrates that the fidelity of garment texture can be progressively enhanced during the online CO procedure.

5. Conclusion

In this work, we propose a novel multi-stage garment transfer network that performs robustly on in-the-wild imagery. By leveraging easily accessible dance videos, our model predicts a blended flow called wFlow integrating both 2D and 3D body information to map garment textures between different persons. To further enhance the synthesized quality of substandard queries, a novel cyclic optimization is incorporated to iteratively refine the dressing result. We also envision the new dataset, *Dance50k*, can be used to facilitate related human-centric research areas that not limit to virtual try-on. The architecture presented in this paper provides a practical and reliable solution for real world virtual dressing application.

6. Acknowledgement

This work was supported in part by National Natural Science Foundation of China (NSFC) under Grant No.61976233, Guangdong Province Basic and Applied Basic Research (Regional Joint Fund-Key) Grant No.2019B1515120039, Guangdong Outstanding Youth Fund (Grant No.2021B1515020061), Shenzhen Fundamental Research Program (Project No.RCYX20200714114642083, No.JCYJ20190807154211365).

References

[1] Badour AlBahar, Jingwan Lu, Jimei Yang, Zhixin Shu, Eli Shechtman, and Jia-Bin Huang. Pose with Style: Detail-preserving pose-guided image synthesis with conditional stylegan. ACM Transactions on Graphics, 2021. 2

[2] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(6):567–585, 1989. 2

[3] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019. 3

[4] Seunghwan Choi, Sunghyun Park, Minsoo Lee, and Jaegul Choo. Viton-hd: High-resolution virtual try-on via misalignment-aware normalization. In Proc. of the IEEE conference on computer vision and pattern recognition (CVPR), 2021. 2

[5] Aiyu Cui, Daniel McKee, and Svetlana Lazebnik. Dressing in order: Recurrent person image generation for pose transfer, virtual try-on and outfit editing. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 14638–14647, October 2021. 6, 7

[6] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bochao Wang, Hanjiang Lai, Jia Zhu, Zhiting Hu, and Jian Yin. Towards multi-pose guided virtual try-on network. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9026–9035, 2019. 2

[7] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bowen Wu, Bing cheng Chen, and Jian Yin. Fw-gan: Flow-navigated warping gan for video virtual try-on. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 1161–1170, 2019. 3

[8] A. Dosovitskiy, P. Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. 2015 IEEE International Conference on Computer Vision (ICCV), pages 2758–2766, 2015. 3, 4

[9] Xin Gao, Zhenjiang Liu, Zunlei Feng, Chengji Shen, Kairi Ou, Haihong Tang, and Mingli Song. Shape controllable virtual try-on for underwear models. ArXiv, abs/2107.13156, 2021. 2

[10] Chongjian Ge, Yibing Song, Yuying Ge, Han Yang, Wei Liu, and Ping Luo. Disentangled cycle consistency for highly-realistic virtual try-on. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 16928–16937, June 2021. 2

[11] Yuying Ge, Yibing Song, Ruimao Zhang, Chongjian Ge, Wei Liu, and Ping Luo. Parser-free virtual try-on via distilling appearance flows. arXiv preprint arXiv:2103.04559, 2021. 2, 3

[12] Xintong Han, Xiaojun Hu, Weilin Huang, and Matthew R. Scott. Clothflow: A flow-based model for clothed person generation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10471–10480, 2019. 2, 3, 4

[13] Xintong Han, Zuxuan Wu, Zhe Wu, Ruichi Yu, and Larry S. Davis. Viton: An image-based virtual try-on network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7543–7552, 2018. 2

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Advances in Neural Information Processing Systems, pages 6626–6637, 2017. 6

[15] Eddy Ilg, N. Mayer, Tonmoy Saikia, Margret Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1647–1655, 2017. 3

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. CVPR, 2017. 5

[17] Yasamin Jafarian and Hyun Soo Park. Learning high fidelity depths of dressed humans by watching social media dance videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 12753–12762, June 2021. 6

[18] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In Proceedings of the European Conference on Computer Vision, pages 694–711, 2016. 4

[19] Rico Jonschkowski, Austin Stone, J. T. Barron, A. Gordon, K. Konolige, and A. Angelova. What matters in unsupervised optical flow. In ECCV, 2020. 3

[20] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In Computer Vision and Pattern Recognition (CVPR), 2018. 3, 5

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020. 3

[22] Hyug Jae Lee, Rokkyu Lee, Minseok Kang, Myounghoon Cho, and Gunhan Park. La-viton: A network for looking-attractive virtual try-on. In 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pages 3129–3132, 2019. 2

[23] Kathleen M Lewis, Srivatsan Varadharajan, and Ira Kemelmacher-Shlizerman. Tryongan: Body-aware try-on via layered interpolation. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2021), 40(4), 2021. 3

[24] Yining Li, Chen Huang, and Chen Change Loy. Dense intrinsic appearance flow for human pose transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019. 2, 3

[25] Xiaodan Liang, Ke Gong, Xiaohui Shen, and Liang Lin. Look into person: Joint body parsing & pose estimation network and a new benchmark. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018. 3, 5

[26] Jiaming Liu, Yu Sun, Xiaojian Xu, and U. Kamilov. Image restoration using total variation regularized deep image prior.

ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 7715–7719, 2019. 5

[27] Wen Liu, Zhixin Piao, Jie Min, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019. 2, 3, 5, 7

[28] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016. 6

[29] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: a skinned multi-person linear model. ACM Transactions on Graphics, 34(6), 2015. 2, 3

[30] Yifang Men, Yiming Mao, Yuning Jiang, Wei-Ying Ma, and Zhouhui Lian. Controllable person image synthesis with attribute-decomposed gan. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020. 3, 6, 7

[31] Matiur Rahman Minar and Heejune Ahn. Cloth-vton: Clothing three-dimensional reconstruction for hybrid image-based virtual try-on. In Asian Conference on Computer Vision (ACCV), 2020. 2

[32] Matiur Rahman Minar, Thai Thanh Tuan, Heejune Ahn, Paul Rosin, and Yu-Kun Lai. Cp-vton+: Clothing shape and texture preserving image-based virtual try-on. In CVPRW, 2020. 2

[33] Assaf Neuberger, Eran Borenstein, Bar Hilleli, Eduard Oks, and Sharon Alpert. Image based virtual try-on network from unpaired data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5184–5193, 2020. 3

[34] N. Neverova, Riza Alp Güler, and I. Kokkinos. Dense pose transfer. In ECCV, 2018. 2

[35] Yurui Ren, Ge Li, Shan Liu, and Thomas H Li. Deep spatial transformation for pose-guided person image generation and animation. IEEE Transactions on Image Processing, 2020. 7

[36] Kripasindhu Sarkar, Vladislav Golyanik, Lingjie Liu, and Christian Theobalt. Style and pose control for image synthesis of humans from a single monocular view, 2021. 3

[37] Kripasindhu Sarkar, Lingjie Liu, Vladislav Golyanik, and C. Theobalt. Humangan: A generative model of humans images. ArXiv, abs/2103.06902, 2021. 3

[38] Kripasindhu Sarkar, Dushyant Mehta, Weipeng Xu, Vladislav Golyanik, and C. Theobalt. Neural re-rendering of humans from a single image. In ECCV, 2020. 3

[39] Yibing Song, Jiawei Zhang, Lijun Gong, Shengfeng He, Linchao Bao, Jinshan Pan, Q. Yang, and Ming-Hsuan Yang. Joint face hallucination and deblurring via structure generation and detail enhancement. International Journal of Computer Vision, 127:785–800, 2019. 3

[40] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8934–8943, 2018. 3

[41] Zachary Teed and Jun Deng. Raft: Recurrent all-pairs field transforms for optical flow. ArXiv, abs/2003.12039, 2020. 3

[42] Bochao Wang, Huabin Zheng, Xiaodan Liang, Yimin Chen, Liang Lin, and Meng Yang. Toward characteristic-preserving image-based virtual try-on network. In Proceedings of the European Conference on Computer Vision, pages 589–604, 2018. 2

[43] Jiahang Wang, Tong Sha, Wei Zhang, Zhoujun Li, and Tao Mei. Down to the last detail: Virtual try-on with fine-grained details. Proceedings of the 28th ACM International Conference on Multimedia, 2020. 2

[44] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4):600–612, 2004. 6

[45] Zhenyu Xie, Zaiyu Huang, Fuwei Zhao, Haoye Dong, Michael Kampffmeyer, and Xiaodan Liang. Towards scalable unpaired virtual try-on via patch-routed spatially-adaptive GAN. In Thirty-Fifth Conference on Neural Information Processing Systems, 2021. 3

[46] Zhenyu Xie, Xujie Zhang, Fuwei Zhao, Haoye Dong, Michael C. Kampffmeyer, Haonan Yan, and Xiaodan Liang. Was-vton: Warping architecture search for virtual try-on network. ArXiv, abs/2108.00386, 2021. 2

[47] Han Yang, Ruimao Zhang, Xiaobao Guo, Wei Liu, Wangmeng Zuo, and Ping Luo. Towards photo-realistic virtual try-on by adaptively generating-preserving image content. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7850–7859, 2020. 2

[48] Jae Shin Yoon, Lingjie Liu, Vladislav Golyanik, Kripasindhu Sarkar, Hyun Soo Park, and Christian Theobalt. Pose-guided human animation from a single image in the wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 15039–15048, June 2021. 3

[49] Ruiyun Yu, Xiaoqi Wang, and Xiaohui Xie. Vtnfp: An image-based virtual try-on network with body and clothing feature preservation. In The IEEE International Conference on Computer Vision, pages 10511–10520, 2019. 2

[50] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In CVPR, 2018. 6

[51] Fuwei Zhao, Zhenyu Xie, Michael Kampffmeyer, Haoye Dong, Songfang Han, Tianxiang Zheng, Tao Zhang, and Xiaodan Liang. M3d-vton: A monocular-to-3d virtual try-on network. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 13239–13249, October 2021. 2

[52] Shengyu Zhao, Yilun Sheng, Yue Dong, E. Chang, and Yan Xu. Maskflownet: Asymmetric feature matching with learnable occlusion mask. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 6277–6286, 2020. 3