Dressing in Order: Recurrent Person Image Generation for Pose Transfer, Virtual Try-on and Outfit Editing

Aiyu Cui Daniel McKee Svetlana Lazebnik
University of Illinois at Urbana-Champaign
{aiyucui2,dbmckee2,slazebni}@illinois.edu

Abstract

We proposes a flexible person generation framework called Dressing in Order (DiOr), which supports 2D pose transfer, virtual try-on, and several fashion editing tasks. The key to DiOr is a novel recurrent generation pipeline to sequentially put garments on a person, so that trying on the same garments in different orders will result in different looks. Our system can produce dressing effects not achievable by existing work, including different interactions of garments (e.g., wearing a top tucked into the bottom or over it), as well as layering of multiple garments of the same type (e.g., jacket over shirt over t-shirt). DiOr explicitly encodes the shape and texture of each garment, enabling these elements to be edited separately. Joint training on pose transfer and inpainting helps with detail preservation and coherence of generated garments. Extensive evaluations show that DiOr outperforms other recent methods like ADGAN [28] in terms of output quality, and handles a wide range of editing functions for which there is no direct supervision.

1. Introduction

Driven by increasing power of deep generative models as well as commercial possibilities, person generation research has been growing fast in recent years. Popular applications include virtual try-on [3, 9, 14, 16, 29, 40, 42], fashion editing [4, 11], and pose-guided person generation [5, 8, 20, 23, 26, 34, 35, 36, 37, 38, 45]. Most existing work addresses only one generation task at a time, despite similarities in overall system designs. Although some systems [8, 28, 35, 36] have been applied to both pose-guided generation and virtual try-on, they lack the ability to preserve details [28, 35] or lack flexible representations of shape and texture that can be exploited for diverse editing tasks [8, 28, 35, 36].

This paper proposes a flexible 2D person generation pipeline applicable not only to pose transfer and virtual try-on, but also fashion editing, as shown in Figure 1. The architecture of our system is shown in Figure 2. We separately encode pose, skin, and garments, and the garment encodings are further separated into shape and texture. This allows us to freely play with each element to achieve different looks.

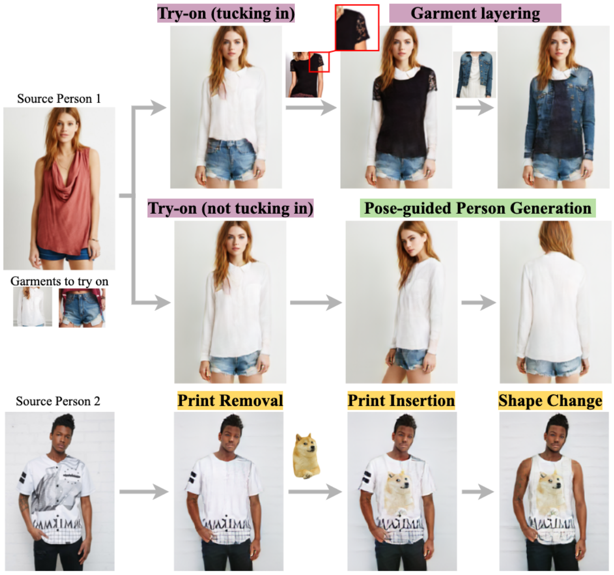

Figure 1. Applications supported by our DiOr system: Virtual try-on supporting different garment interactions (tucking in or not) and overlay; pose-guided person generation; and fashion editing (texture insertion and removal, shape change). Note that the arrows indicate possible editing sequences and relationships between images, *not* the flow of our system.

In real life, people put on garments one by one, and can layer them in different ways (e.g., shirt tucked into pants, or worn on the outside). However, existing try-on methods start by producing a mutually exclusive garment segmentation map and then generate the whole outfit in a single step. This can only achieve one look for a given set of garments, and the interaction of garments is determined by the model. By contrast, our system incorporates a novel recurrent generation module to produce different looks depending on the order of putting on garments. This is why we call our system **DiOr**, for **Dressing in Order**.

After a survey of related work in Section 2, we will describe our system in Section 3. Section 3.1 will introduce our encoding of garments into 2D shape and texture, enabling each to be edited separately. The shape is encoded using soft masks that can additionally capture transparency. A flow field estimation component at encoding time allows for a more accurate deformation of the garments to fit the target pose. Section 3.2 will describe our recurrent generation scheme that does not rely on garment labels and can handle a variable number of garments. Section 3.3 will discuss our training approach, which combines pose transfer with inpainting to enable preservation of fine details. Section 4 will present experimental results (including comparisons and user study), and Section 5 will illustrate the editing functionalities enabled by our system.

2. Related Work

Virtual try-on. Generation of images of a given person with a desired garment on is a challenging task that requires both capturing the garment precisely and dressing it properly on the given human body. The simplest try-on methods are aimed at replacing a single garment with a new one [3, 8, 9, 14, 16, 17, 40, 42]. Our work is more closely related methods that attempt to model all the garments worn by a person simultaneously, allowing users to achieve multiple garment try-on [18, 28, 29, 33, 36]. SwapNet [33] works by transferring all the clothing from one person's image onto the pose of another target person. This is done by first generating a mutually exclusive segmentation mask of the desired clothing on the desired pose. O-VITON [29] also starts by producing a mutually exclusive segmentation mask for all try-on garments, and then injects the garment encodings into the associated regions. Unlike our work, O-VITON cannot change the pose of the target person. Attribute-decomposed GAN (ADGAN) [28] encodes garments in each class into a 1D style code and feeds a concatenation of codes into a StyleGAN [15] generator. It additionally conditions on 2D pose, enabling pose transfer as well as try-on. Our system adopts a similar kind of conditioning. However, as will be seen in our comparative evaluation, ADGAN's 1D garment encoding, which does not separate shape from texture, is severely limited in its fidelity of garment reproduction.

Sarkar et al. [35, 36] achieve high-quality try-on results by aligning the given human images with a 3D mesh model (SMPL [25]) via DensePose [7], estimating a UV texture map corresponding to the desired garments, and rendering this texture onto the desired pose. The focus of our work is different, as we avoid explicit 3D human modeling.

All of the above methods assume a pre-defined set of garment classes (e.g., tops, jackets, pants, skirts, etc.) and allow at most one garment in each class. This precludes the ability to layer garments from the same class (e.g., one top over another). By contrast, while we rely on an off-the-shelf clothing segmenter, our generation pipeline does not make use of garment classes, only masks. Moreover, in all previous work, when there is overlap between two garments (e.g. top and bottom), it is up to the model to decide the interaction of the two garments, (e.g., whether a top is tucked into the bottom). Unlike these methods, ours produce different results for different dressing orders.

Pose transfer requires changing the pose of a given person while keeping that person's identity and outfit the same. Several of the virtual try-on methods above [8, 28, 33, 35, 36] are explicitly conditioned on pose, making them suitable for pose transfer. Our method is of this kind. An advantage of pose transfer is that there exist datasets featuring people with the same clothing in multiple poses [24], making it easier to obtain supervision than for virtual try-on.

Most relevant to us are pose transfer methods that represent poses using 2D keypoints [5, 26, 28, 37, 38, 45]. However, these methods have a limited ability to capture garment details and result in blurry textures. Global Flow Local Attention (GFLA) [34] and Clothflow [8] compute dense 2D flow fields to align source and target poses. We adopt the global flow component of GFLA as part of our system, obtaining comparable results on pose transfer while adding a number of try-on and editing functions.

Other pose transfer methods [6, 20, 23, 30, 36] rely on 3D human modeling via DensePose [7] and SMPL [25]. They work either by completing the UV map and re-rendering [6, 30, 36], or learning a flow from the rich 3D information [20, 23]. Such methods represent a different philosophy from ours and are therefore less comparable.

Fashion editing. Fashion++ [11] learns to minimally edit an outfit to make it more fashionable, but there is no way for the user to control the changes. Dong et al. [4] edits outfits guided by user's hand sketches. Instead, our model allows users to edit what they want by making garment selections, and changing the order of garments in a semantic manner.

3. Method

This section describes our DiOr pipeline (Fig. 2). We introduce our person representation in Section 3.1, then describe our pipeline in Section 3.2, our training strategy in Section 3.3, and relationship to prior work in Section 3.4.

3.1. Person Representation

We represent a person as a (pose, body, {garments}) tuple, each element of which can come from a different source image. Unlike other works (e.g., [28, 29]) the number of garments can vary and garment labels are not used. This allows us to freely add, remove and switch the order of garments. Following prior work [28, 34], we represent pose P as the 18 keypoint heatmaps defined in OpenPose [1].

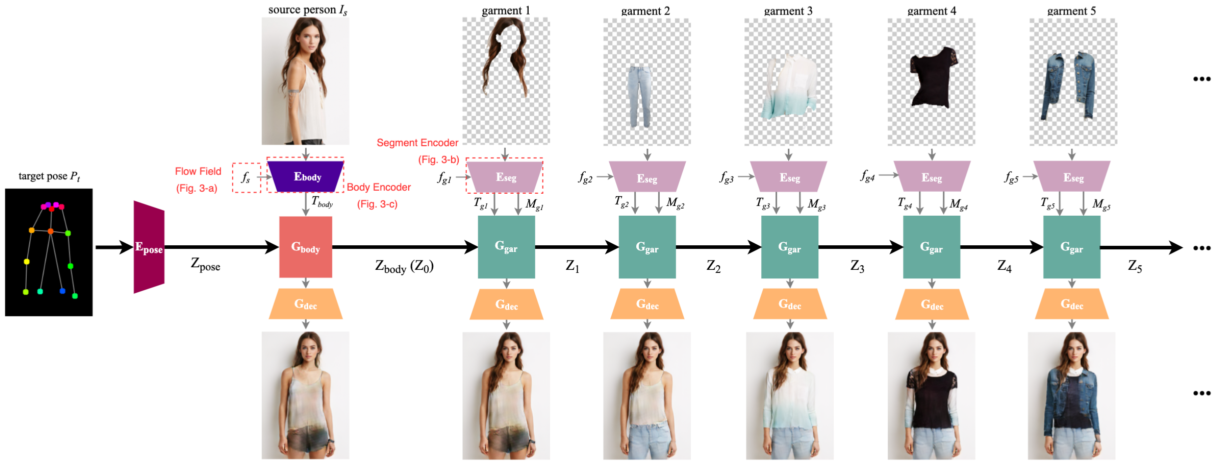

Figure 2. **DiOr generation pipeline** (see Section 3 for details). We represent a person as a (*pose*, *body*, {*garments*}) tuple. Generation starts by encoding the target pose as Z_{pose} and the source body as texture map T_{body}. Then the body is generated as Z_{body} by the generator module $\mathbf{G}_{\text{body}}$. Z_{body} serves as Z_0 for the recurrent garment generator $\mathbf{G}_{\text{gar}}$, which receives the garments in order, each encoded by a 2D texture feature map T_{gk} and soft shape mask M_{gk}. In addition to masked source images, the body and garment encoders take in estimated flow fields f to warp the sources to the target pose. We can decode at any step to get an output showing the garments put on so far.

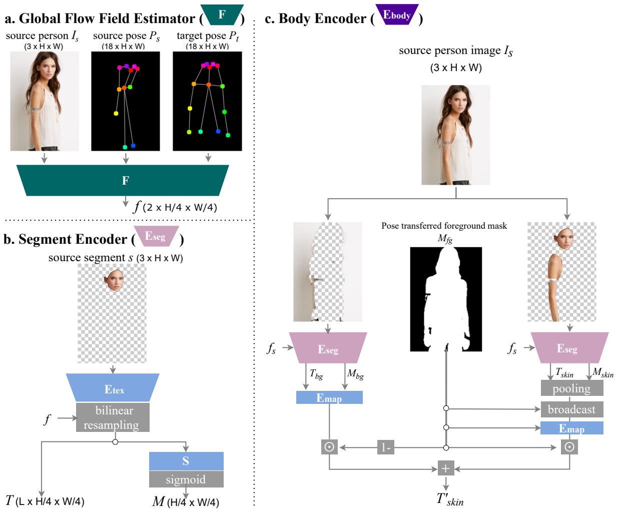

Figure 3. System details. (a) Global flow field estimator $\mathbf{F}$ adopted from GFLA [34], which is modified to only yield a flow field f. (b) Segment encoder $\mathbf{E}_{\text{seg}}$ that produces a spatially aligned texture feature map T and a soft shape mask M. (c) Body encoder $\mathbf{E}_{\text{body}}$ that broadcasts a mean skin vector to the entire foreground region (union of the masks of pose-transferred foreground parts) and maps it to the correct dimension by $\mathbf{E}_{\text{map}}$ for later style blocks.

Garment representation. Given a source garment g_k worn by a person in an image $I_{g_k} \in \mathbb{R}^{3\times H\times W}$, we first run an off-the-shelf human parser [19] to obtain the masked garment segment s_{g_k}. We also obtain a pose estimate P_{g_k} for the person in I_{g_k} by OpenPose [1]. Because P_{g_k} is different from the desired pose P, we need to infer a flow field f_{g_k} to align the garment segment s_{g_k} with P. We do this using the Global Flow Field Estimator $\mathbf{F}$ from GFLA [34] (Fig. 3(a)). $\mathbf{F}$ can also work on the shop images of garments only (without a person wearing them), in which case P_{g_k} will just be empty heatmaps (see second example in Fig. 5).

Next, as shown in Fig. 3(b), we encode the garment segment s_{g_k} by the segment encoder module $\mathbf{E}_{\text{seg}}$. This starts with a texture encoder $\mathbf{E}_{\text{tex}}$, which consists of the first three layers of the VGG encoder in ADGAN [28] (for a downsampling factor of 4) with leaky ReLU[27]. The output of $\mathbf{E}_{\text{tex}}$ is warped by the flow field f_{g_k} using bilinear interpolation, yielding a **texture feature map** denoted T_{g_k}. We also compute a **soft shape mask** of the garment segment as $M_{g_k} = \mathbf{S}(T_{g_k})$, where $\mathbf{S}$ is a segmenter consisting of three convolutional layers. The texture map T_{g_k} and shape mask M_{g_k} are both outputs of the segment encoder $\mathbf{E}_{\text{seg}}$.

Because the texture feature map T_{g_k} will be used as style input for later style blocks, we map T_{g_k} to the correct dimension of style blocks as $T'_{g_k} = \mathbf{E}_{\text{map}}(T_{g_k}, M_{g_k})$, where the mapping module, $\mathbf{E}_{\text{map}}$, consists two convolutional layers and takes the stacked T_{g_k} and M_{g_k} as input.

Body representation. Fig. 3(c) shows the process of encoding the body of the source person from image $I_s \in \mathbb{R}^{3\times H\times W}$. Based on the human segmenter [19], we form masks corresponding to background s_{bg} and skin s_{skin} (the latter consisting of arms, legs and face). These are encoded by the a above-described segment encoder $\mathbf{E}_{\text{seg}}$ to get $(T_{\text{bg}}, M_{\text{bg}})$ and $(T_{\text{skin}}, M_{\text{skin}})$, respectively.

To ensure that the body feature map spans the entire body region regardless of any garments that would cover it later, we compute a mean body vector b of T_{skin} over the ROI defined by M_{skin}. Then we broadcast b to the pose-transferred foreground region M_{fg} (the union of the masks of all pose-transferred foreground parts), map the broadcasted feature map to the correct dimension by $\mathbf{E}_{\text{map}}$, and combine with the mapped background feature T'_{bg} to get

$$T'_{\text{body}} = M_{\text{fg}} \odot \mathbf{E}_{\text{map}}(M_{\text{fg}} \otimes b, M_{\text{fg}}) + (1 - M_{\text{fg}}) \odot T'_{\text{bg}} \quad (1)$$

where $\otimes$ and $\odot$ denote broadcasting and elementwise multiplication, respectively.

3.2. Generation Pipeline

In the main generation pipeline (Fig.2), we start by encoding the "skeleton" P, next generating the body from T'_{body} (eq. 1), and then the garments from encoded texture and shape masks $(T'_{g_1}, M_{g_1}), ..., (T'_{g_K}, M_{g_K})$ in sequence.

Pose and skin generation. To start generation, we encode the desired pose P using the pose encoder $\mathbf{E}_{\text{pose}}$, implemented as three convolutional layers, each followed by instance normalization [39] and leaky ReLU [27]. This results in hidden pose map $Z_{\text{pose}} \in \mathbb{R}^{L \times H/4 \times W/4}$, where L is the latent channel size. Next, we generate the hidden body map Z_{body} given Z_{pose} and the body texture map T'_{body} by a body generator $\mathbf{G}_{\text{body}}$, implemented by two style blocks in ADGAN [28]. Because our body texture map T'_{body} is in 2D, ADGAN's adaptive instance normalization [12] in the style block is replaced by SPADE [31].

Recurrent garment generation. Next, we generate the garments, treating Z_{body} as Z_0. For the k-th garment, the garment generator $\mathbf{G}_{\text{gar}}$ takes its mapped texture map T'_{g_k} and soft shape mask M_{g_k}, together with the previous state Z_{k-1}, and produces the next state Z_k as

$$Z_k = \mathbf{\Phi}(Z_{k-1}, T'_{g_k}) \odot M_{g_k} + Z_{k-1} \odot (1 - M_{g_k}) \,, \quad (2)$$

where $\mathbf{\Phi}$ is a conditional generation module with the same structure as $\mathbf{G}_{\text{body}}$ above. Note that the soft shape mask M_{g_k} effectively controls garment opacity – a novel feature of our representation. More details are in supp. materials.

After the encoded person is finished dressing, we get the final hidden feature map Z_K and output image $I_{\text{gen}} = \mathbf{G}_{\text{dec}}(Z_K)$, where $\mathbf{G}_{\text{dec}}$ is the decoder implemented as the same as the final decoder in ADGAN [28], consisting of residual blocks, upsampling and convolutional layers followed by layer normalization and ReLU.

3.3. Training

Similar to ADGAN [28], we train our model on pose transfer: given a person image I_s in a source pose P_s, generate that person in a target pose P_t. As long as reference images I_t of the same person in the target pose are available, this is a supervised task. To perform pose transfer, we set the body image and the garment set to be those of the source person, and render them in the target pose. There could be up to four separately encoded garments for a person to be added in order, so the recurrent generator gets ample training examples of various layering types and garment combinations.

We started by training a model solely on pose transfer, but observed that it gives imprecise or inconsistent results for try-on and overlay (see Fig. 6). To improve the realism of our model, we next experimented with training it for reconstruction as well as transfer, i.e., setting $P_t = P_s$ for a fraction of the training examples. Although this helped to preserve details and improved handling of garment overlaps, the resulting model could not complete missing regions in garments (e.g., regions covered by hair in the source). At length, we found inpainting, or recovery of a partially masked-out source image I'_s, to be a better supplementary training task, enabling detail preservation while filling in missing regions. We combine the tasks by use a percentage α of the training data for inpainting and the rest for pose transfer. In our implementation, $\alpha = 0.2$, and inpainting masks are generated by the free-form algorithm of Yu et al. [43].

To train on both pose transfer and inpainting, we use all the six loss terms from GFLA [34]. Two of these are correctness and regularization loss for the predicted flow field, which are combined into the geometric loss L_{geo}. Another three GFLA terms encourage consistency of generated and real target pairs: L1 loss, perceptual loss, and style loss. These are combined into the content loss L_{content}. The final GFLA term is a GAN loss L_{GAN}, for which GFLA uses a single discriminator conditioned on pose, but we use two discriminators, one conditioned on the pose and the other on segmentation, as in ADGAN [28]. Our discriminators have the same architecture as GFLA's. We set the coefficients of these six loss terms by following GFLA.

In addition, to ensure that our shape masks capture the shape correctly, we use a pixel-level binary cross-entropy loss between the soft shape mask M_g and its associated "ground truth" segmentation (extracted by the parser [19] from the target image) for each garment. This loss is denoted as L_{seg}. Our final, combined loss is thus given by

$$L = L_{\text{content}} + L_{\text{geo}} + \lambda_{GAN} L_{GAN} + \lambda_{\text{seg}} L_{\text{seg}} \,, \quad (3)$$

where we set λ_{seg} to 0.1 and λ_{GAN} to 1.

3.4. Relationship to Prior Work

Our system was most closely inspired by ADGAN [28]. Like ADGAN, we separately encode each garment, condition the generation on 2D pose, and train on pose transfer. We also borrow the architecture of some ADGAN blocks, as explained above. However, ADGAN encodes a garment into single 1D vector, but we encode a garment in shape and texture separately in 2D. Thus, DiOr allows shape and texture of individual garments to be edited separately, which is impossible in ADGAN. Our 2D encoding is better than ADGAN's 1D encoding at capturing complex spatial patterns, giving us superior results on virtual try-on, as shown in next section. Besides, In ADGAN, after garments are separately encoded, all the embeddings are fused into a single vector, so the number and type of garments are fixed, and garment order is not preserved. By contrast, in our recurrent pipeline, garments are injected one at a time, and their number, type, and ordering can vary.

Our method also builds on GFLA [34] by adopting its global flow component and most of the loss terms. Our experiments will show that we achieve similar performance without GFLA's local attention component. Plus, GFLA can only handle pose transfer, while our model can solve a number of additional tasks.

A few previous methods have also recognized the potential of inpainting to help with human image generation, though they use it differently than we do. ACGPN [42] is a single garment try-on method that features an inpainting module to fuse the elements of the person to be rendered. In 3D-based person re-rendering literature, two recent approaches [6, 36] use an inpainting loss to complete unseen regions of the UV texture map.

4. Experiments

4.1. Implementation Details

We train our model on the DeepFashion dataset [24] with the same training/test split used in PATN [45] for pose transfer at 256×176 resolution. In implementation, we run Eq. 2 twice for each garment for better performance. For the first 20k iterations, we use the same procedure as GFLA to warm up the global flow field estimator $\mathbf{F}$. Meanwhile, we warm up the texture encoder $\mathbf{E}_{\text{tex}}$ and final decoder $\mathbf{G}_{\text{dec}}$ with L_{content} and L_{GAN} losses by encoding a masked input image with $\mathbf{E}_{\text{tex}}$ and recovering the complete image using $\mathbf{G}_{\text{dec}}$. Then for the next 150k iterations, we train the network with $\mathbf{F}$ frozen using Adam optimizer with learning rate $1e-4$. Finally, we unfreeze $\mathbf{F}$ and train the entire network end-to-end with learning rate $1e-5$ until the model converges. We train a **small model** with L, the latent dimension of Z, set to 128 and a **large model** with L=256, on one and two TITAN Xp cards, respectively.

4.2. Automatic Evaluation of Pose Transfer

We run automatic evaluation on the pose transfer task, which is the only one that has reference images available. Table 1 shows a comparison of our results with GFLA [34] and ADGAN [28], both of which use the same 2D keypoints to represent the pose, have code and model publicly available, and use the same train/val split.

We compute several common metrics purporting to measure the structural, distributional, and perceptual similarity between generated and real reference images: SSIM [41], FID [10], and LPIPS [44]. Plus, we propose a new metric **sIoU**, which is the mean IoU of the segmentation masks produced by the human segmenter [19] for real and generated images, to measure the shape consistency. This metric is inspired by the FCN scores used in Isola et al. [13] to evaluate the consistency of label maps for the labels-to-photos generation task. To mitigate any possible bias from using the same segmenter to obtain garment masks in our pipeline, we compute this metric using ATR human parse labels [21] instead of the LIP labels [22] used in our pipeline. We further process the ATR label sets by merging left and right body parts, which gives more stable results. Without over-interpreting the automatic metrics (which are found to be sensitive to factors like resolution, sharpness, and compression quality of the reference images [32]) we can conservatively conclude that our pose transfer performance is at least comparable to that of GFLA and ADGAN, which is confirmed by the user study reported in Section 4.4. Our large model has the highest sIoU, which suggests its ability to preserve the structure of generated garments, and is consistent with the example outputs shown in Fig. 4. There, our output is qualitatively similar to GFLA (not surprising, since we adopt part of their flow mechanism), and consistently better than ADGAN, whose inability to reproduce garment textures or structured patterns is not obvious from the automatic metrics.

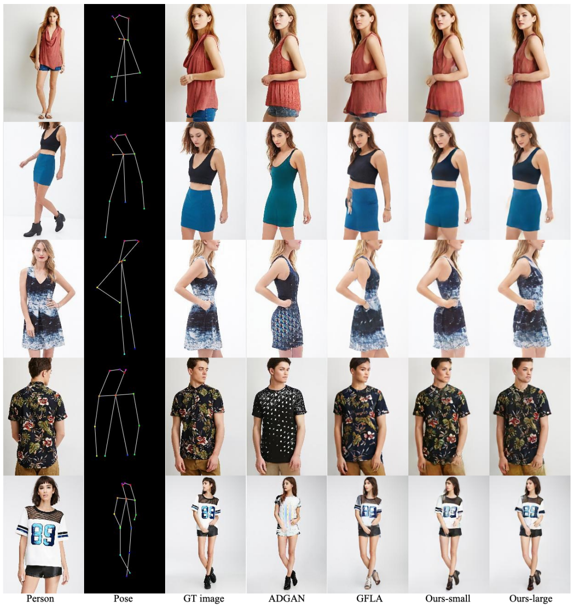

Figure 4. Pose transfer results.

4.3. Ablation Studies

Recurrent generation. We ablate our recurrent mechanism by merging the feature maps of all garments based on the softmax of their soft shape masks (without sigmoid taken) and injecting the merged feature map into a single-shot garment generator $\mathbf{G}_{\text{gar}}$. As shown in Tab. 2(a), the non-recurrent model gets a considerably lower sIoU than the full one. Fig. 5 shows the reason: when there is overlap between garments, the non-recurrent model tends to blend garments together, resulting in ghosting artifacts.

Joint training on pose transfer and inpainting. Tab. 2(b) reports the results of our model trained without inpainting, and trained with reconstruction instead of inpainting (with

	size	SSIM ↑	FID ↓	LPIPS ↓	sIoU ↑
Def-GAN* [37]	82.08M	-	18.46	0.233	-
VU-Net* [5]	139.4M	-	23.67	0.264	-
Pose-Attn* [45]	41.36M	-	20.74	0.253	-
Intr-Flow* [20]	49.58M	-	16.31	**0.213**	-
GFLA* [34]	14.04M	0.713	**10.57**	0.234	57.32
ours-small	11.26M	0.720	12.97	0.236	57.22
ours-large	24.41M	**0.725**	13.10	0.229	**58.63**

(a) Comparisons at 256 × 256 resolution

	size	SSIM↑	FID↓	LPIPS↓	sIoU↑
ADGAN [28]	32.29M	0.772	18.63	0.226	56.54
ours-small	11.26M	0.804	14.34	0.182	58.99
ours-large	24.41M	**0.806**	**13.59**	**0.176**	**59.99**

(b) Comparisons at 256×176 resolution

Table 1. Pose transfer evaluation. (a) Comparison with GFLA [34] (and other methods reported in [34]) at 256×256 resolution (our model is initially trained at 256×176 and then fine-tuned to 256×256). FID and LPIPS scores for methods with * are reproduced from GFLA, all other scores are computed by us using the same reference images used in [34] (provided by the authors). Note that Intr-Flow is the only method leveraging 3D information. (b) Comparison with ADGAN [28] at 256×176 resolution. Arrows indicate whether higher (↑) or lower (↓) values of the metric are considered better.

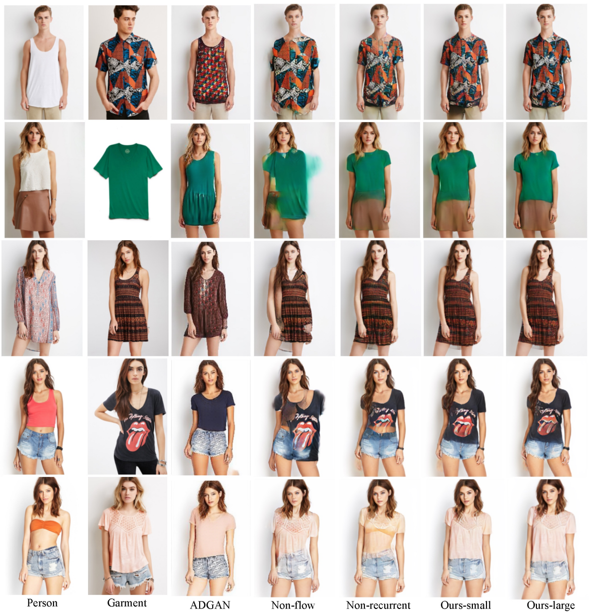

Figure 5. Virtual try-on results. We set the try-on order as (hair, bottom, top, jacket). Note that ADGAN is often unable to preserve the shape and texture of the transferred garment, while the non-recurrent version of our model creates ghosting in areas of garment overlap. In the second example, the garment is transferred from a "shop" image without a person, which is not our expected setting.

$\alpha = 0.2$). Although the differences from the full model are not apparent in the table for the pose transfer task, we can observe distinctive artifacts associated with different training choices in the rest applications like virtual try-on and layering, as described in section 3.3 and shown in Fig. 6.

Encoding: separate vs. single, 2D vs. 1D. We evaluate the effect of separate shape and texture encodings vs. a single garment encoding (i.e., joint shape and texture), as well as 2D vs. 1D encoding in Tab. 2(c). To encode a garment by a single representation in 2D, we change Eq. (3) to $Z_k = \mathbf{\Phi}(Z_{k-1}, T_{g_k}) + Z_{k-1}$ to remove the shape factor. To further reduce this single encoding to 1D, we follow ADGAN's scheme (SPADE is switched to AdaIn [12] in this case). For the version with separate 1D shape and texture codes, we attempt to decode the 1D shape vector into a mask by learning a segmenter consisting of a style block taking pose as input and generating the segmentation conditioned on the 1D shape vector, followed by broadcasting the texture vector into the shape mask to get the texture map. As an additional variant, we trained a model combining 1D texture encoding with 2D shape encoding, where we also get the texture map by broadcasting. From Tab. 2(c) and Fig. 7, we can see that the ablated versions, especially the 1D ones, are blurrier and worse at capturing details. This is consistent with our intuition that it is hard to recover spatial texture from a 1D vector. The 2D single encoding looks plausible but is still less sharp than the full model, plus it does not permit separate editing of shape and texture.

		SSIM↑	FID↓	LPIPS↓	sIoU↑
	Full	0.804	14.34	**0.182**	58.99
(a)	Non-recurrent	0.804	14.85	0.183	58.44
(b)	Pose transfer only training	0.801	**13.77**	0.186	**59.01**
	Joint training with reconst.	0.803	14.35	0.184	58.33
(c)	Single 2D encoding	**0.806**	15.18	0.183	58.71
	Single 1D encoding	0.797	16.14	0.200	56.22
	1D texture+1D shape	0.798	20.07	0.204	55.17
	1D texture+2D shape	0.802	16.05	0.192	57.40
(d)	Non-flow	0.800	16.47	0.196	56.28

Table 2. Ablation studies for the small model at 256×176 on the pose transfer task (see text for details).

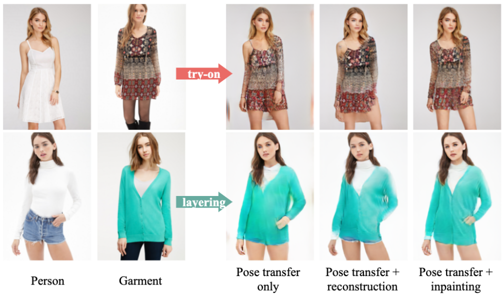

Figure 6. Comparison of our model trained on pose transfer only, joint with reconstruction, and joint with inpainting. The two ablated models have poorer ability to fill in holes (e.g., those created by hair on top of the source garment) or to create clear and coherent garment boundaries.

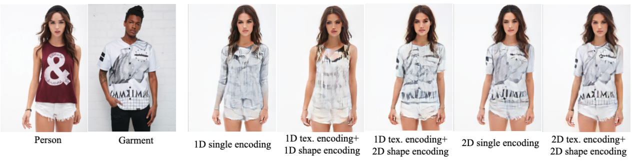
Person
Garment
1D single encoding
1D tex. encoding+ 1D shape encoding
1D tex. encoding+ 2D shape encoding
2D single encoding
2D tex. encoding+ 2D shape encoding

Figure 7. Comparison of different garment encodings from Table 2(c). The 1D encoding causes blurry texture. The single encoding in 2D is plausible in try-on but limits flexibility for editing tasks.

Compared method	Task	Prefer other vs. ours
GFLA [34]	pose transfer	47.73% vs. **52.27%**
ADGAN [28]	pose transfer	42.52% vs. **57.48%**
ADGAN [28]	virtual try-on	19.36% vs. **80.64%**

Table 3. User study results (see text). For fairness, ADGAN is compared with our large model trained at 256×176, while GFLA is compared with our large model fine-tuned to 256×256, with all outputs resized to 256×176 before being displayed to users.

Flow field for garment localization. To prove the necessity of flow field f, which transforms the body part or garment from the source pose to its desired pose, we ablate the flow field f by removing the global flow field estimator F and the bilinear interpolation step in segment encoder $\mathbf{G}_{\text{seg}}$. As evident from Fig. 5 and Tab. 2(d), the flow field f is essential for placing a garment in its right position and rendering the output realistically.

4.4. User Study

Next, we report the results of a user study comparing our model to ADGAN and GFLA on pose transfer, and ADGAN on virtual try-on. We show users inputs and outputs from two unlabeled models in random order, and ask them to choose which output they prefer.

For pose transfer, we randomly select 500 pairs from the test subset as the question pool. For virtual try-on, we randomly select 700 image pairs of person and garment (restricted to tops only), with the person facing front in both the person image and the garment image. We manually filter out the pairs which have no person shown in the person image, and which have no top shown in the garment image. For fairness, we also exclude all pairs with a jacket present, because our model considers jackets as separate garments from tops but ADGAN treats them as tops. When we run our model, the garment dressing order is set to (hair, top, bottom, jacket). Altogether, 22 questions for either pose transfer or try-on are given to each user. The first two questions are used as warm-up and not counted. We collected responses from 53 users for transfer, and 45 for try-on.

The results are shown in Table 3. For pose transfer, our model is comparable to or slightly better than GFLA and ADGAN. Interestingly, ADGAN does not come off too badly on pose transfer despite its poor texture preservation because it tends to produce well-formed humans with few pose distortions, and generates nice shading and folds for textureless garments (see top example in Fig/ 4). For virtual try-on, the advantage of our model over ADGAN is decisive due to our superior ability to maintain the shape and texture of transferred garments (see Fig. 5).

5. Editing Applications

In this section, we demonstrate the usage of our model for several fashion editing tasks. With the exception of garment reshaping, which we found to need fine-tuning (see below), all the tasks can be done directly with the model trained as described in Section 3. See Supplemental Materials for additional qualitative examples.

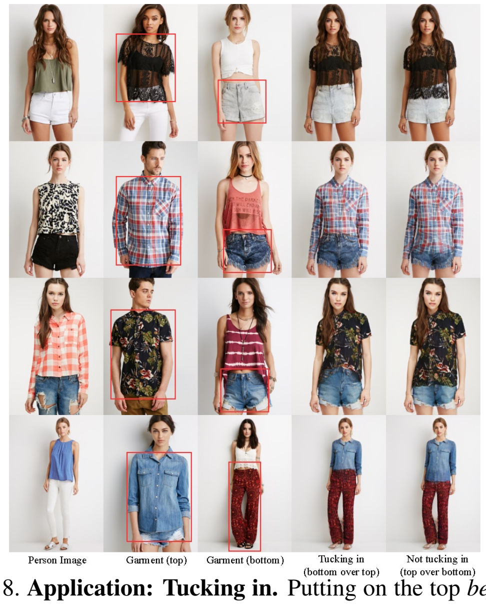
Person Image
Garment (top)
Garment (bottom)
Tucking in (bottom over top)
Not tucking in (top over bottom)

Figure 8. **Application: Tucking in.** Putting on the top *before* the bottom tucks it in, and putting it on *after* the bottom lets it out.

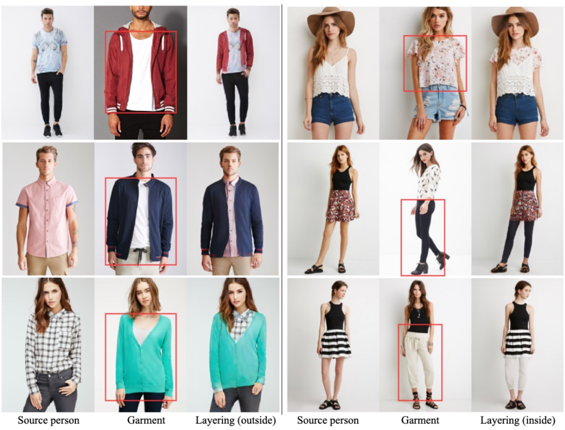

Figure 9. **Application: Single layering.** Layer a garment outside (left) or inside (right) another garment in the same category.

Tucking in. As shown in Fig. 8, our model allows users to decide whether they want to tuck a top into a bottom by specifying dressing order.

Garment layering. Fig. 9 shows the results of layering garments from the same category (top or bottom). Fig. 10 shows that we can also layer more than two garments in the same category (e.g., jacket over sweater over shirt).

Content removal. To remove an unwanted print/pattern on a garment, we can mask the corresponding region in the texture map T_g while keeping the shape mask M_g unchanged, and the generator will fill in the missing part. (Fig. 11).

Print insertion. To insert an external print, we treat the masked region from an external source as an additional "garment". In this case, the generation module is responsible for the blending and deformation, which limits the realism but produces plausible results as shown in Fig. 12.

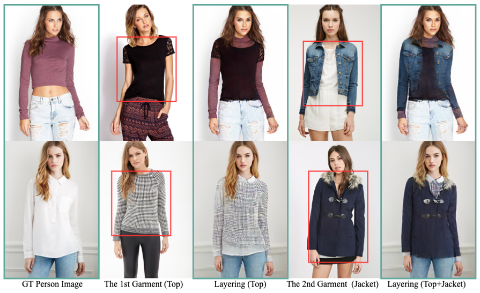

Figure 10. **Application: Double layering.** Layering two garments on top of the existing outfits in sequence.

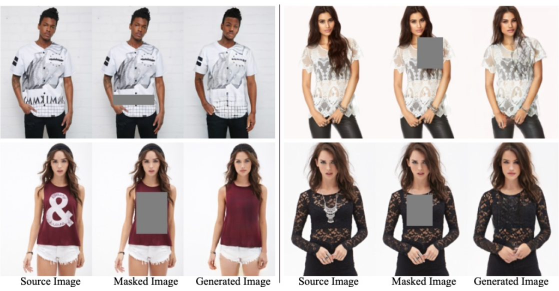

Figure 11. **Application: Content removal.**

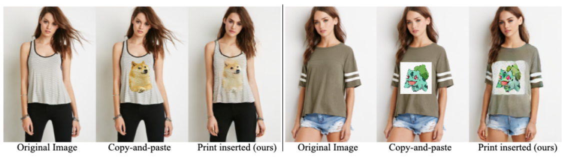

Figure 12. **Application: Print insertion.**

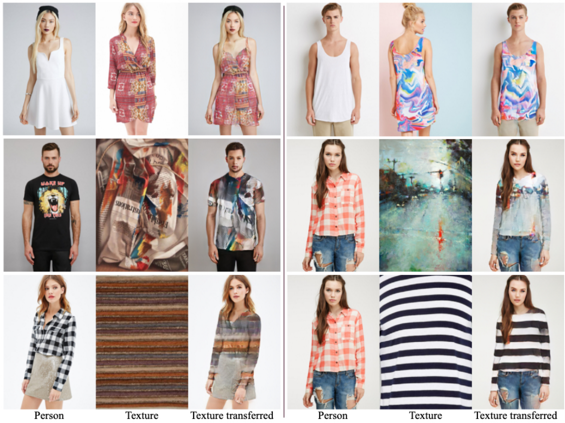

Figure 13. **Application: Texture transfer**.

Texture transfer. To transfer textures from other garments or external texture patches, we simply replace the garment texture map T_g with the desired feature map encoded by $\mathbf{E}_{\text{tex}}$. Fig.13 shows the results of transferring textures from source garments (top row) and the DTD dataset [2] (bottom two rows). In the latter case, the texture does not deform over the body realistically, but the shading added by the generation module is plausible, and the results for less structured prints can be striking.

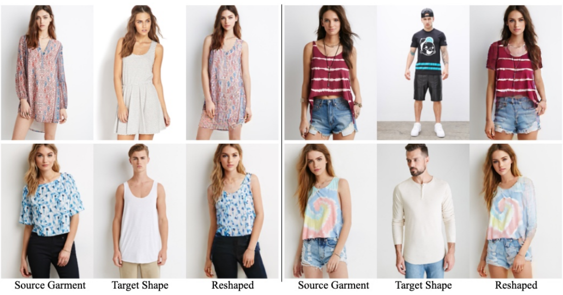

Figure 14. **Application: Reshaping.**

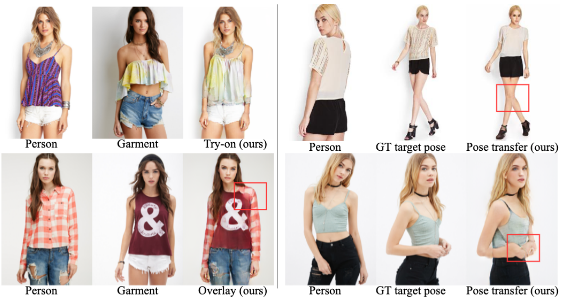

Figure 15. Failure cases.

Reshaping. We can reshape a garment by replacing its shape mask with that of another garment (Fig. 14). Our default model can easily handle removals (e.g. changing long sleeves to short), but not extensions (making sleeves longer). To overcome this, we fine-tuned the model with a larger inpainting ratio ($\alpha = 0.5$). The resulting model does a reasonable job of adding short sleeves to sleeveless garments (top right example of Fig. 14), but is less confident in hallucinating long sleeves (bottom right).

6. Limitations and Future Work

This paper introduced DiOr, a flexible person generation pipeline trained on pose transfer and inpainting but capable of diverse garment layering and editing tasks for which there is no direct supervision. While our results are promising, there remain a number of limitations and failure modes. Some of these are illustrated in Fig.15: complex or rarely seen poses are not always rendered correctly, unusual garment shapes are not preserved, some ghosting artifacts are present, and holes in garments are not always filled in properly. More generally, the shading, texture warping, and garment detail preservation of our method, while better than those of other recent methods, are still not entirely realistic. In the future, we plan to work on improving the quality of our output through more advanced warping and higher-resolution training and generation.

Acknowledgments. This work was supported in part by NSF grants IIS 1563727 and IIS 1718221, Google Research Award, Amazon Research Award, and AWS Machine Learning Research Award.

References

[1] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019.

[2] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[3] 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.

[4] Haoye Dong, Xiaodan Liang, Yixuan Zhang, Xujie Zhang, Xiaohui Shen, Zhenyu Xie, Bowen Wu, and Jian Yin. Fashion editing with adversarial parsing learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8120–8128, 2020.

[5] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8857–8866, 2018.

[6] Artur Grigorev, Artem Sevastopolsky, Alexander Vakhitov, and Victor Lempitsky. Coordinate-based texture inpainting for pose-guided human image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12135–12144, 2019.

[7] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018.

[8] 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.

[9] 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.

[10] 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. *Advances in neural information processing systems*, 30, 2017.

[11] Wei-Lin Hsiao, Isay Katsman, Chao-Yuan Wu, Devi Parikh, and Kristen Grauman. Fashion++: Minimal edits for outfit improvement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5047–5056, 2019.

[12] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[14] Nikolay Jetchev and Urs Bergmann. The conditional analogy gan: Swapping fashion articles on people images. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2287–2292, 2017.

[15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[16] Kathleen M Lewis, Srivatsan Varadharajan, and Ira Kemelmacher-Shlizerman. Vogue: Try-on by stylegan interpolation optimization. *arXiv preprint arXiv:2101.02285*, 2021.

[17] Kedan Li, Min Jin Chong, Jingen Liu, and David Forsyth. Toward accurate and realistic virtual try-on through shape matching and multiple warps. *arXiv preprint arXiv:2003.10817*, 2020.

[18] Kedan Li, Min Jin Chong, Jeffrey Zhang, and Jingen Liu. Toward accurate and realistic outfits visualization with attention to details. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15546–15555, 2021.

[19] Peike Li, Yunqiu Xu, Yunchao Wei, and Yi Yang. Self-correction for human parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[20] 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*, pages 3693–3702, 2019.

[21] Xiaodan Liang, Si Liu, Xiaohui Shen, Jianchao Yang, Luoqi Liu, Jian Dong, Liang Lin, and Shuicheng Yan. Deep human parsing with active template regression. *IEEE transactions on pattern analysis and machine intelligence*, 37(12):2402–2414, 2015.

[22] Xiaodan Liang, Chunyan Xu, Xiaohui Shen, Jianchao Yang, Si Liu, Jinhui Tang, Liang Lin, and Shuicheng Yan. Human parsing with contextualized convolutional neural network. In *Proceedings of the IEEE international conference on computer vision*, pages 1386–1394, 2015.

[23] 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*, pages 5904–5913, 2019.

[24] 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.

[25] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

[26] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 405–415, 2017.

[27] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Citeseer, 2013.

[28] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 5084–5093, 2020.

[29] 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[30] Natalia Neverova, Riza Alp Guler, and Iasonas Kokkinos. Dense pose transfer. In *Proceedings of the European conference on computer vision (ECCV)*, pages 123–138, 2018.

[31] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[32] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On buggy resizing libraries and surprising subtleties in fid calculation. *arXiv preprint arXiv:2104.11222*, 2021.

[33] Amit Raj, Patsorn Sangkloy, Huiwen Chang, Jingwan Lu, Duygu Ceylan, and James Hays. Swapnet: Garment transfer in single view images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 666–682, 2018.

[34] Yurui Ren, Xiaoming Yu, Junming Chen, Thomas H Li, and Ge Li. Deep image spatial transformation for person image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7690–7699, 2020.

[35] Kripasindhu Sarkar, Vladislav Golyanik, Lingjie Liu, and Christian Theobalt. Style and pose control for image synthesis of humans from a single monocular view. *arXiv preprint arXiv:2102.11263*, 2021.

[36] Kripasindhu Sarkar, Dushyant Mehta, Weipeng Xu, Vladislav Golyanik, and Christian Theobalt. Neural re-rendering of humans from a single image. In *European Conference on Computer Vision*, pages 596–613. Springer, 2020.

[37] Aliaksandr Siarohin, Enver Sangineto, Stéphane Lathuiliere, and Nicu Sebe. Deformable gans for pose-based human image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3408–3416, 2018.

[38] Hao Tang, Song Bai, Li Zhang, Philip HS Torr, and Nicu Sebe. Xinggan for person image generation. In *European Conference on Computer Vision*, pages 717–734. Springer, 2020.

[39] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6924–6932, 2017.

[40] 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 (ECCV)*, pages 589–604, 2018.

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

[42] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 7850–7859, 2020.

[43] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4471–4480, 2019.

[44] 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.

[45] Zhen Zhu, Tengteng Huang, Baoguang Shi, Miao Yu, Bofei Wang, and Xiang Bai. Progressive pose attention transfer for person image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2347–2356, 2019.