Collecting The Puzzle Pieces: Disentangled Self-Driven Human Pose Transfer by Permuting Textures

Nannan Li
Boston University
nnli@bu.edu

Kevin J Shih
NVIDIA

Bryan A. Plummer
Boston University
bplum@bu.edu

Abstract

Human pose transfer synthesizes new view(s) of a person for a given pose. Recent work achieves this via self-reconstruction, which disentangles a person's pose and texture information by breaking the person down into parts, then recombines them for reconstruction. However, part-level disentanglement preserves some pose information that can create unwanted artifacts. In this paper, we propose Pose Transfer by Permuting Textures (PT^2), an approach for self-driven human pose transfer that disentangles pose from texture at the patch-level. Specifically, we remove pose from an input image by permuting image patches so only texture information remains. Then we reconstruct the input image by sampling from the permuted textures for patch-level disentanglement. To reduce noise and recover clothing shape information from the permuted patches, we employ encoders with multiple kernel sizes in a triple branch network. Extensive experiments on DeepFashion and Market-1501 PT^2 reports significant gains on automatic metrics over other self-driven methods, and even outperforms some fully-supervised methods. A user study also reports images generated by our method are preferred in 68% of cases over self-driven approaches from prior work. Code is available at
https://github.com/NannanLi999/pt_square

1. Introduction

The goal of human pose transfer is to change the pose of a person while preserving the person's appearance and clothing textures. It has wide applications such as virtual try-on [42, 3, 43], controllable person image manipulation [3, 17] and person re-identification [46]. Recent work focused on using paired image data (*i.e.*, two images of the same person before and after reposing) [49, 45], but collecting such data can be very labor intensive. Alternatively, self-driven meth-
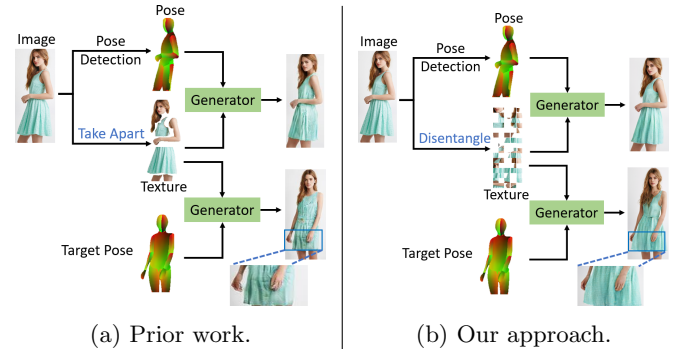

(a) Prior work.   (b) Our approach.

Figure 1: Self-driven pose transfer methods extract texture and pose representations and then learn to reconstruct the original image. **(a)** Most recent work breaks down the person into several parts to disentangle textures [23, 24, 39]. However, pose information may still appear in the part-wise texture features. Without supervision, disentangling them is difficult [19]. **(b)** Our approach disentangles textures from pose by permuting the image patches, effectively eliminating pose information, which enables more effective separation of pose and texture features than prior work.

ods train pose transfer models without paired images [24, 35]. However, they face two major challenges: effectively disentangling texture and pose, and preserving texture details across changes in pose. As illustrated in Fig. 1a, most prior works attempt to achieve the pose and texture disentanglement at a part-level [23, 41, 24, 39], but each body part may retain some pose information. Without direct supervision from pose-invariant textures, disentangling pose and textures in the image is difficult [19].

To address the above issues, we propose Pose Transfer by Permuting Textures (PT^2), a self-driven pose transfer method that uses input permutation to transfer clothing patterns to the target pose. The input permutation function geometrically disentangles texture from pose in raw images without requiring supervision. As shown in Fig. 1b, this creates a disentangled

texture sample space by randomly reordering the texture patches on the person so that the source pose is not easily recovered from the permuted textures. We find this patch-level disentanglement can transfer detailed clothing patterns to any target pose. However, our permutation function does introduce two new issues that we must address to effectively transfer pose.

The first drawback we address is that in addition to losing pose information after permutation, we also lose shape information. To recover some shape information of clothing items (*e.g.*, short dress or long dress), we use a dense pose representation [10] to model the geometric transformation between pose representations. While prior work also leverages pose information, this is typically a sparse representation (*e.g.*, [24, 39]). As our experiments show, PT^2 more effectively leverages the denser representations than prior work.

The second issue we address arises from the unnatural boundaries between newly neighboring permuted patches. Specifically, we use a dual-scale encoder in the pose and texture branches for handling permuted inputs. The small-scale encoder provides a more accurate representation of the texture details without the permutation noise, whereas the large-scale encoder enables us to capture longer range visual patterns. Our experiments show that each encoder provides a unique and necessary representation of the texture patterns, resulting in best performance when they are used jointly.

Our paper is organized as follows. Sec. 3 introduces our model pipeline. Sec. 3.1.1 details our paper's main contribution, the input permutation function. Sec. 3.1.2 describes the dual-scale encoder for recovering the clothing shape from the permuted textures. Extensive experiments in Sec. 4 show that PT^2 significantly improves the image quality of self-driven approaches on DeepFashion [18] and Market-1501 [48]. The user study in Sec. 4.1 also reports that our synthesized images are preferred in 68% over prior work.

2. Related Work

Pose transfer with unpaired images. Transferring pose without losing texture details when paired data is absent is challenging [39]. Early attempts produced poor quality images due to overfitting to an identity mapping in the self-supervised training [26, 6, 23]. More recent methods use cycle-GAN to include both source and target poses in training [35, 30], or disentangle pose and textures to prevent overfitting [1, 39, 7, 24]. These include using the StyleGAN framework to learn implicitly disentangled features [1, 7], or incorporating a part-wise texture encoder [39, 21]. However, we find that input permuting is more effective at disentangling texture and pose than part-wise

methods. This remains true even when incorporating methods that erase pose information in part-wise features by using their mean and variance as style vectors (*e.g.*, as done in [24]). As we will show, such transformations results in washed-out clothing details in the style representation. In contrast to these methods, our approach disentangles the texture at a patch level, and then restores the clothing details using dual-scale encoders in a triple branch network, which achieves better texture transfer with large pose variations.

Pose transfer with paired images. Methods trained with paired images learn the clothing deformation via soft attention that aggregates source person features with weighted sampling [50, 45, 27, 8], or via sparse attention that approximates a dense flow field from the source to the target person [11, 37, 28]. Some methods also use a semantic parsing map to provide further guidance to control the style of each body part [44, 22]. To help infer occluded parts of the person after pose transfer, prior work has also explored inpainting 2D partial texture to 3D full texture in the UV space, and then projecting it back to the 2D pose [9, 32, 1, 31]. However, the 2D to 3D projection could lose partial texture information. Another line of work [40, 33] uses multiple views of the same person to achieve 3D reconstruction with various poses. These methods all require full supervision from paired data, which might be difficult to collect in some real-world scenarios. Our approach only needs unpaired data for training, which enables direct fitting to an in-the-wild target domain.

Jigsaw Puzzle Solving. Our approach is also similar to some self-supervised representation learning methods that jigsaw puzzle solving, where the pretext task divides the image into large patches and then infers their relative positions by using each patch's inherent geometry information (*e.g.*, [25, 2]). However, our goal is to *sample* relevant patches based on the target posture. Thus, with the target posture as guidance, we use much smaller image patches to remove position information and to disentangle pose and texture.

3. Self-Driven Pose Transfer by Permuting Textures

Let I_s be the source image with posture P_s. Our goal is to synthesize a new view I_t of the same person wearing the same clothes in a target posture P_t. We assume we only have *unpaired* data for training, which enables direct fitting to an in-the-wild target domain as our experiments will show. Following [5, 17], we separate the image I_s into a foreground person E_s and the background B. The foreground person is generated by our pose transfer network (shown in Fig. 2 and dis-
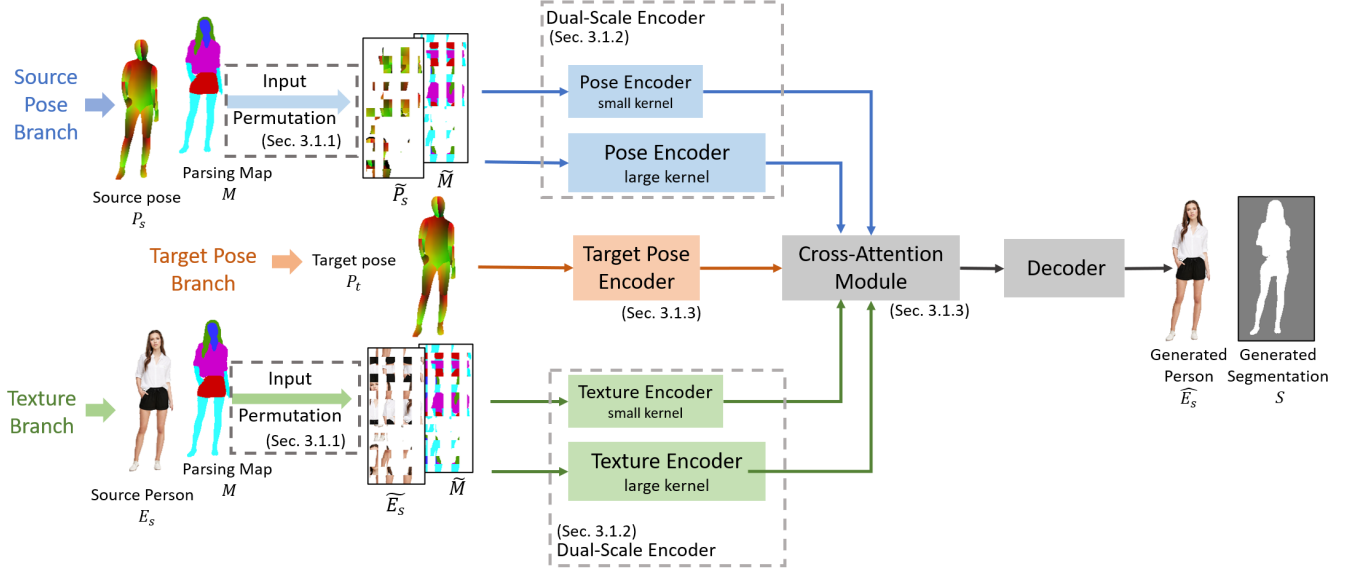
Figure 2: Overview of our pose transfer network in PT^2. The network takes the source person E_s, source parsing map M, and source pose P_s as inputs. In the source pose branch and texture branch, P_s and E_s are first permuted (Sec. 3.1.1) to create the corresponding sample space, which is encoded with dual-scale encoders (Sec. 3.1.2). Then the encoded features are sampled in a cross-attention module (Sec. 3.1.3) to be decoded into the generated person $\hat{E}_s$ and its segmentation S. The output of the pose transfer network is combined with the output of the background inpainting network (Sec. 3.2) to produce the final image.

cussed Sec. 3.1). The background is produced by the background inpainting network in Sec. 3.2. They are combined to create the final reconstructed image $\hat{I}_s$. We will describe our training objectives in Sec. 3.3.

3.1. Pose Transfer Network

The goal of the pose transfer network is to generate a new view of the foreground person in a target pose. As illustrated in Fig. 2, we use three branches: a source pose branch, a target pose branch, and a texture branch. The target pose branch encodes the given target pose. The source pose and texture branch first randomly permutes the source pose and texture, respectively (Sec. 3.1.1), then uses a dual-scale encoder (Sec. 3.1.2) to encode the permuted patches. The features of the permuted patches are sampled via cross-attention based on the target pose features (Sec. 3.1.3). The resulting outputs in the source pose branch and texture branch are then merged and decoded into the generated person $\hat{E}_s$ and its segmentation S. The rest of the image, namely the background for $\hat{E}_s$ and S, will be generated from a specialized background inpainting network (Sec. 3.2) to create the final image. As shown in Fig. 2, the target pose is identical to the source pose during training. However, during inference we simply need to replace the target pose with a new (different) posture to enable pose transfer.

3.1.1 Input Permutation

The input permutation in Fig. 2 disentangles the pose and textures at patch-level by randomly shuffling image patches, which is introduced below.

Inputs to the Texture Branch. As illustrated for the texture branch in Fig. 2, the texture representation should not simply be the source person E_s itself, as it is entangled with its posture. To erase the pose information from E_s, we create a texture sample space by dividing the image into $p \times p$ squares, referred to as "patches," then shuffling their locations. Intuitively the original pose cannot easily be retrieved from the permuted patches when the patch size p is sufficiently small. Additionally, we mask 20% of the patches so the model can learn to reconstruct occluded regions. Formally, let RandMask($\cdot$) be the input permutation function. The inputs of the texture branch become $[\tilde{E}_s; \tilde{M}] = \text{RandMask}([E_s; M], m_t)$, where $[;]$ is concatenation and m_t is the masking rate. We set $m_t = 0.2$ in our experiments (see Sec. 4.3 for an ablation study). The permuted textures $[\tilde{E}_s; \tilde{M}]$ are given as inputs to the dual-scale texture encoder (Sec. 3.1.2).

Inputs to the Pose Branches. Prior work uses a single branch to encode pose and texture [26, 23, 24, 39], but we find separate source pose and texture branches (as shown in Fig. 2) improves shape and texture recovery after pose transfer. These pose branches do not
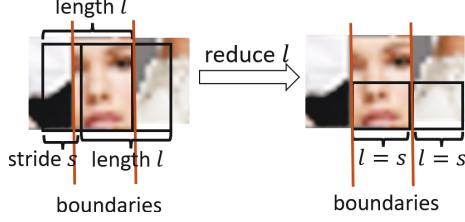
Figure 3: Comparing the receptive field in the large-kernel encoder (left) and small-kernel encoder (right) on a closeup of two permuted patches. The kernel size l is reduced to avoid boundary-crossing (see Sec. 3.1.2).
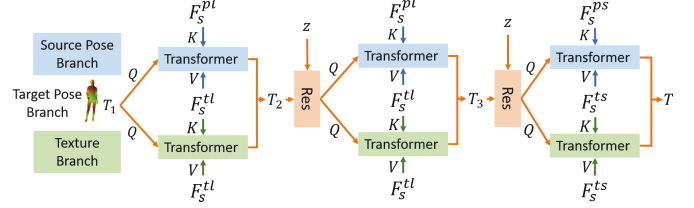

Figure 4: Cross-attention module in Sec. 3.1.3. Res represents a residual layer. The cross-attention mechanism aligns the permuted texture with the target pose.

contain texture information as postures are represented as 2D UV coordinates obtained from DensePose [10] pretrained on COCO [16]. We also permute source pose the same way as textures to learn a pose transformation function that supports large pose variations. In addition, we mask 50% of the source pose features to force the model to learn the inherited symmetry in the human body. Thus, inputs to the source pose branch are $[\tilde{P}_s; \tilde{M}] = \text{RandMask}([P_s; M], m_p)$, where $m_p = 0.5$. During training, the target pose is the original pose, but it is replaced with a new pose during inference. Note that the inputs of the texture and source pose branches are permuted the same way so they are spatially aligned in the attention module (see Sec. 3.1.3). The permuted source pose representations $[\tilde{P}_s; \tilde{M}]$ are then fed to a dual-scale pose encoder (Sec. 3.1.2).

3.1.2 Dual-Scale Encoder

We encode the permuted source pose/texture inputs from Sec. 3.1.1 with a dual-scale encoder containing twelve convolutional layers. When compared to the single-scale encoder in prior work [26, 23, 24, 39], we use different kernel sizes in the encoder's convolutional layers to address issues that arise due to the boundaries in permuted patches. Specifically, each feature vector has a certain receptive field in the image. For a receptive field that crosses the boundary between two permuted image patches, the pixels within the field could be spatially distant and irrelevant to each other due to the permutation. For example, in the left picture of Fig. 3, the black squares denote two adjacent receptive fields. Both squares crossing the boundary are seeing two distinct patterns (*e.g.*, face and shoes) within their own receptive fields. This could introduce a high volume of noise to the feature vector, preventing the model from recognizing true clothing patterns.

A simpler solution to avoid boundary-crossing is to encode each patch separately. However, this would limit the effective receptive field to be at most the patch

size and thus prevent the model from learning more styles and patterns. To solve the issue, we combine the large-kernel encoder with an additional encoder with a reduced kernel size such that the length of the receptive field l equals its stride s and is a divisor of the patch size p. We select this small kernel size such that the kernel does not cross the boundary of the permuted inputs from Sec. 3.1.1. As shown in the right picture of Fig. 3, this design avoids the boundary-crossing problem, enabling the convolutional kernel to learn a consistent pattern within its receptive field. Note that a large kernel size is still necessary as it has more parameters and a larger receptive field for learning larger image patterns. By combining encoders with large and small kernels, our model is capable of learning better feature representations from permuted patches.

3.1.3 Cross-Attention Module

The cross-attention module takes the output feature map of the dual-kernel encoders (Sec. 3.1.2) as inputs and learns to sample texture based on the target pose. As shown in Fig. 4, the attention module consists of three vision transformers [36, 45] formulated as $\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d}}) \cdot V$.

Sampling Source Pose Features. The first two transformers of the source pose branch sample permuted pose features based on the attention between the poses. Let F_s^{pl}, F_s^{tl} represent the output feature map of the large-kernel source pose encoder and texture encoder, respectively. T_1 is the feature map of the target pose P_t, which is encoded by the target pose encoder. The attention is learned from the correlation between the source pose feature F_s^{pl} and the target person feature T_i, which is computed using the transformer's attention function as:

$$Q = W_i^{pq} T_i, K = W_i^{pk} F_s^{pl}, V = W_i^{pv} F_s^{tl}, i = 1, 2. \quad (1)$$

Here, $W_i^{pq}, W_i^{pk}, W_i^{pv}$ are learnable projection matrices. After each transformer layer, the output features are concatenated and passed through a residual

layer [12] as in Fig. 4. A random noise vector z is injected into the residual layer as the affine transformation parameters of the feature map T_i to prevent mode collapse [15]. The residual layer can be written as: $\text{Res}(T_i) = f_i(z) \cdot h_i(T_i) + g_i(z) + h_i^{skip}(T_i)$, where $f_i, g_i, h_i, h_i^{skip}$ are learnable functions.

Note that the first transformer learns a texture-agnostic attention map for the target pose. Starting from T_2, information from the texture branch is gradually integrated into the target person representation T_2 and T_3.

Sampling Textures. The texture branch keeps refining the sampled textures based on the target person representation T_i. In order to produce spatially aligned feature maps in the sample space, this branch uses the same architectural design as the source pose branch. The attention map for texture sampling is formulated as the correlation between the texture feature F_s^{tl} and the target person feature T_i:

$$Q = W_i^{tq} T_i, K = W_i^{tk} F_s^{tl}, V = W_i^{tv} F_s^{tl}, i = 1, 2. \quad (2)$$

Fusing features from the small kernel encoder. Let F_s^{ps}, F_s^{ts} represent the output feature map of the small-kernel source pose encoder and texture encoder, respectively. In the last transformer layer, we replace F_s^{pl}, F_s^{tl} in the above equations with features produced by small-kernel encoders (*i.e.*, F_s^{ps}, F_s^{ts}) to learn more fine-grained image details. The output feature map T of the last transformer layer is then fed to the decoder, where T is gradually upsampled to the target person $\hat{E}_s$ and its segmentation mask S.

The cross-attention between two pose branches learns a geometric transformation between different postures. Then the texture branch samples the given textures based on the target pose. Fusing the two sources of information provides a more accurate match between the given pose and textures. By filling in more clothing details that are learned through small kernels, our pose transfer network can then faithfully recover the appearance of clothing items after pose transfer.

3.2. Background Inpainting Network

Following [5, 17], we use a separate UNet [20]-based background inpainting network to infer the background pixels of the masked foreground region. During training, we mask the bounding box containing the source person so the model learns to infer the entire background area during pose transfer. We combine the generated person with the output of the background inpainting network to obtain the final image. Formally, let $\hat{B}$ be the inpainted background output. Given the generated person $\hat{E}_s$ and its segmentation mask S pro-

duced by the pose transfer network, the final reconstructed image is:

$$\hat{I}_s = S \odot \hat{E}_s + (1 - S) \odot \hat{B} \quad (3)$$

3.3. Training Objectives

We train PT^2 using an adversarial loss using a pair of discriminators D and D_p [24]. D penalizes the distribution difference between the synthesized image $\hat{I}_s$ and the ground truth I_s. D_p evaluates that if the posture in $\hat{I}_s$ matches the source pose P_s. Thus, our adversarial loss is written as:

$$\begin{aligned} L_{adv} = &D(\hat{I}_s)^2 + (1 - D(I_s))^2 \\ &+ D_p([\hat{I}_s; P_s])^2 + (1 - D_p([I_s; P_s]))^2. \end{aligned} \quad (4)$$

To ensure the correctness of our image generation, we first add a simple reconstruction loss,

$$L_{rec} = ||\hat{I}_s - I_s||_1. \quad (5)$$

We also include a perceptual loss [14] that encourages both the ground truth and reconstructed image to have similar semantic properties,

$$L_{perc} = \sum_i ||\phi^i(\hat{I}_s) - \phi^i(I_s)||_1, \quad (6)$$

where ϕ^i is the ith layer of a VGG model [34] pretrained on ImageNet [4]. Finally, we use a style loss that penalizes discrepancies on colors and textures using the Gram matrix $\mathbb{G}(\cdot)$ of the features,

$$L_{style} = \sum_i ||\mathbb{G}(\phi^i(\hat{I}_s)) - \mathbb{G}(\phi^i(I_s))||_1. \quad (7)$$

Thus, our total loss can be written as

$$L_{total} = \lambda_1 L_{adv} + \lambda_2 L_{rec} + \lambda_3 L_{perc} + \lambda_4 L_{style}. \quad (8)$$

where λ_{1-4} are scalar hyperparameters.

4. Experiments

Datasets. We evaluate our proposed model on two benchmarks: DeepFashion [18] and Market1501 [48]. DeepFashion contains 52,712 high-quality images with a clean background. Market-1501 has 32,668 low-resolution images with various lighting conditions and noisy background. Following [45, 39], we select 8,570 test pairs on DeepFashion and 12,000 test pairs on Market-1501. As in prior self-driven methods [24, 39], we use 37,332 training images for DeepFashion and 12,112 training images for Market-1501.

Metrics. Following [35, 45], we train our PT^2 model on 256×176 images and pad the generated images to

Method	FID↓	SSIM↑	LPIPS↓	IS↑
Supervised by paired images				
GFLA*[28]	10.513	0.798	0.146	3.246
PISE [44]	13.623	0.777	0.189	3.257
SPIG [22]	12.243	0.790	0.186	3.097
DPTN [45]	13.466	0.789	0.176	3.253
CASD*[49]	11.360	**0.813**	0.154	3.056
NTED [27]	**6.786**	0.808	**0.133**	**3.264**
NTED(Market)	37.481	0.712	0.307	3.065
No paired images				
DPIG [23]	44.615	0.705	0.275	3.176
VU-Net [6]	23.012	0.789	0.312	3.012
E2E [35]	16.350	0.761	0.206	3.431
MUST-GAN [24]	15.012	0.773	0.190	3.470
MUST-GAN(DP)	12.157	0.781	0.183	3.422
SCM-Net [39]	12.180	0.751	0.182	**3.632**
PT^2(Ours)	**8.338**	**0.795**	**0.158**	3.469

Table 1: Pose transfer results on DeepFashion. (Market) means the method is trained on Market-1501 and (DP) means the method uses DensePose features. Results from prior work are evaluated using our code except SCM-Net. Generated images of methods marked with ∗ are resized due to different training image sizes.

Method	FID↓	M-SSIM↑	M-LPIPS↓	IS↑
Supervised by paired images				
GFLA [28]	20.194	0.815	0.138	2.546
SPIG [22]	22.043	0.819	0.129	2.761
DPTN [45]	17.929	**0.820**	0.125	2.479
NTED(DF)	38.831	0.734	0.212	2.242
No paired images				
PT^2(Ours)	**17.389**	**0.820**	**0.122**	**2.789**

Table 2: Pose transfer results on Market-1501. Results from prior work are evaluated using our code. (DF) means the method is trained on DeepFashion.

included in Supplementary Sec. B.

4.1. Quantitative Results

Table 1 compares methods on the pose transfer task using DeepFashion, where our approach significantly outperforms prior unpaired approaches on FID, SSIM, and LPIPS. Notably, we found our FID to be 4 points better than that of SCM-Net. Our method also outperforms most supervised (paired) methods *without paired training data*. Similar behavior is seen on Market-1501 (Table 2), where PT^2 outperforms paired methods in Masked-SSIM and Masked-LPIPS.

We perform an additional test to rule out dense pose as the primary contributing factor. For this, we retrained MUST-GAN [24], replacing their OpenPose with DensePose (denoted as MUST-GAN(DP) in Table 1). MUST-GAN(DP) improves the original OpenPose variant, suggesting dense pose as a better representation. However, our proposed approach still outperforms MUST-GAN(DP).

Benefits of Using Unpaired Data. In addition to requiring less annotation, our self-driven model can also be trained and directly fit to in-the-wild unpaired datasets while supervised methods can not. To simulate this scenario, we use NTED(Market) [27] trained on paired Market-1501 as our supervised model and test on DeepFashion as our in-the-wild dataset in Table 1. Our PT^2 model was trained only on DeepFashion using unpaired data. Similarly, in Table 2, NTED(DF) is trained on paired DeepFashion and tested on Market-1501 as the in-the-wild dataset. In Tables 1 & 2, PT^2 obtains better evaluation scores than NTED(Market) and NTED(DF) by a significant margin, which demonstrates an advantageous use case where unpaired training enables direct fitting on the target domain.

User Study. To verify the quality of generated images, we also conducted human evaluations on DeepFashion using Amazon Mechanical Turk. We collected

256×256 for evaluation in DeepFashion. In Market-1501, we use the same image size (*i.e.*, 128×64) for training and evaluation. We use Structural Similarity Index Measure (SSIM) [38], Frechet Inception Distance (FID) [13], Learned Perceptual Image Patch Similarity (LPIPS) [47], and Inception Score (IS) [29] to evaluate image synthesis quality. IS assesses the quality of images generated by adversarial training. In Market-1501, we add Masked-SSIM and Masked-LPIPS computed on the target person region to exclude the irrelevant background from contributing to the metrics.

All the methods are compared using our code under the same image sizes and padding method, except SCM-Net [39], which does not have publicly available code. Specifically, following [35], we pad the original/generated images of size 256 × 176 to 256 × 256 for our evaluation in DeepFashion. We pad all the generated images from other methods [28, 44, 22, 45, 49, 35, 24] in the same manner, and report their scores under our evaluation code in Tables 1 & 2. Note that due to a different training image size in [28, 49], we resized their generated images from 256 × 256 to 256 × 176 using PIL bilinear resampling and then added the padding. In Market-1501, following [22], we keep the size of the original image (*i.e.*, 128 × 64), for all comparisons. More training details are
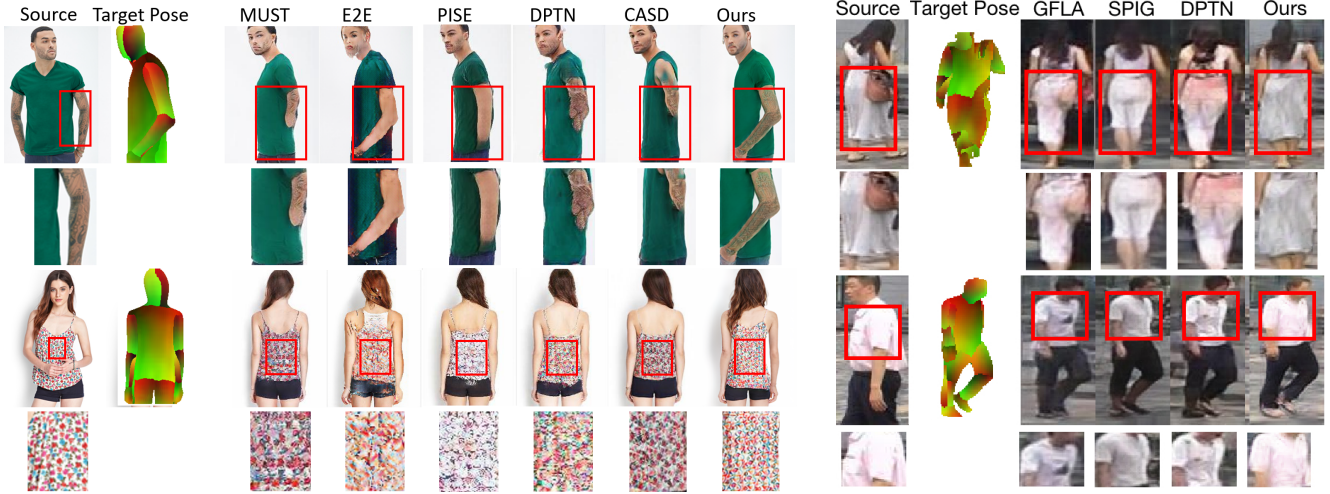
Figure 5: Qualitative pose transfer results on DeepFashion (left) and Market-1501 (right). We enlarged the marked area for a better view of clothing details. MUST, E2E, and our PT^2 are trained with unpaired data, while the rest are supervised by paired data. Our approach transfers the high-frequency texture patterns better than prior work.

	Supervised by paired images			No paired images	
	PISE [44]	DPTN [45]	CASD [49]	E2E [35]	MUST [24]
Facial Identity Preservation	53.2%±3.53	65.1%±3.37	50.4%±3.53	76.9%±2.98	69.2%±3.26
Image Plausibility	58.3%±3.49	66.7%±3.33	51.2%±3.53	76.6%±2.99	63.5%±3.40
Texture Correctness	68.7%±3.27	66.2%±3.35	53.8%±3.53	79.7%±2.84	72.6%±3.15
Average	60.1%±3.46	66.0%±3.27	51.8%±3.53	77.7%±2.93	68.4%±3.29

Table 3: A/B user preferences on DeepFashion. We report how often our approach was selected as most like the reposed image with respect to facial identity, image plausibility, and texture correctness. The number that follows ± is the corresponding standard deviation. Our results are even preferred over some fully supervised methods

3 judgments for 50 images (150 total). Each worker was presented 3 pictures: the true image, a PT^2 generated image, and an image generated by a method from prior work [24, 35, 44, 45, 49]. The worker was asked to evaluate the quality of the generated images in three aspects: clothing texture correctness, facial identity preservation, and image plausibility. Table 3 shows that among self-driven methods, more than 68% of workers believe our method achieves higher fidelity in the generated images in terms of clothing texture correctness, facial identity preservation, and image plausibility. Compared with approaches supervised by paired images, our method achieves comparable performance with an average of over 64% user preference, demonstrating the effectiveness of our proposed approach.

4.2. Qualitative Results

Pose Transfer. Fig. 5 visualizes pose transfer results. We enlarged the area marked with a red bounding box for a better view of clothing details. In the first row (left), our method transferred the arm tattoos to the target pose while other methods either ignored this detail or failed to reconstruct the arm. Our model also reconstructs the color pattern in the second row (left) better than other approaches. This is likely due to the small-scale encoder in our model capturing detailed texture and thus reconstructing it based on the target pose. On the right side of Fig. 5, compared with supervised pose transfer methods, our approach faithfully recovered the shape and color of the dress and shirt in the two examples. More examples and failure analysis are included in the supplementary.

Garment Replacement. Our approach can also be used to swap clothing pieces between people. Due to differences in the shape of the source and target garments (*e.g.*, jeans and shorts), the model can synthesize the texture from the source garment in various ways on the target. For instance, in Fig. 6, we can see

	Method	FID↓	SSIM↑	LPIPS↓	IS↑
(a)	Input Warping, *w/o.* Input Permuting	10.011	0.781±0.072	0.178±0.060	**3.579±0.086**
	w/o. Input Permuting (part-wise)	10.279	0.780±0.069	0.169±0.059	3.525±0.095
	w/o. Pose Masking	9.451	0.776±0.069	0.176±0.059	3.516±0.073
	w/o. Texture Masking	8.780	0.791±0.067	0.164±0.059	3.432±0.083
	w/o. Parsing	8.460	0.791±0.068	0.161±0.159	3.302±0.075
(b)	*w/o.* small kernel	8.905	0.782±0.067	0.170±0.060	3.442±0.119
	w/o. large kernel	9.275	0.785±0.068	0.166±0.060	3.401±0.078
	Patch Concat	8.514	0.784±0.068	0.168±0.059	3.459±0.090
(c)	Pose Concat	9.236	0.775±0.067	0.170±0.059	3.438±0.081
	w/o. Texture Matching	10.786	0.787±0.067	0.172±0.060	3.489±0.085
	PT^2 + Input Warping	**8.332**	0.792±0.059	0.160±0.060	3.470±0.090
	PT^2(Ours)	8.338	**0.795±0.067**	**0.158±0.059**	3.469±0.098

Table 4: Ablation study on DeepFashion reporting the contribution of each component of our model PT^2.

Figure 6: Qualitative results of garment replacement. The left column is the source person and the top row is the reference(target) garment. The target garments are marked with red bounding boxes. Our approach captures the interactions within outfit configurations when transferring appearance.

that the person in the third row has an untucked green shirt, which is tucked in when applying the shape of the shorts in the last column. Similarly, the shorts in the second row are occluded by the camel t-shirt and

pink jackets, but visible in other columns. Overall, this demonstrates that the model captures the interactions within outfit configurations, using this to synthesize plausible outputs.

4.3. Ablation Study

We evaluate the effectiveness of each component of our model in an ablation study. More part-wise analyses are provided in Sec. A of the supplementary.

Effects of input permutation. In Table 4(a), *w/o. Input Permuting (part-wise)* uses a part-wise encoder but does not permute the inputs, which results in entangled pose and textures. *Input Warping, w/o. Input Permuting* uses Thin Plate Spline (TPS) transformation to warp the source image, which can be viewed as a mild way of disentangling the pose and texture at the image-level [21]. *w/o. Pose Maksing* does not mask the source pose, which could cause the model to overfit to visible pose regions. *w/o. Texture Masking* does not mask the textures in the texture branch. *w/o. Parsing* removes the parsing map in the input. As shown in Table 4, our complete model PT^2 improves all the metrics, demonstrating the effectiveness of the proposed input permuting function. To further verify that the arm/body orientation in the small patches will not fail the disentanglement, we combine TPS warping with input permutation using large rotation and scaling in PT^2 + Input Warping. Our results show no additional improvement, suggesting significant disentanglement is already being achieved by our permutation function.

Effects of using dual-scale encoder. In Table 4(b), *w/o. small kernel* uses only the large kernel in the feature encoders, which leads to a loss of clothing detail. *w/o. large kernel* uses only the small kernel encoder,
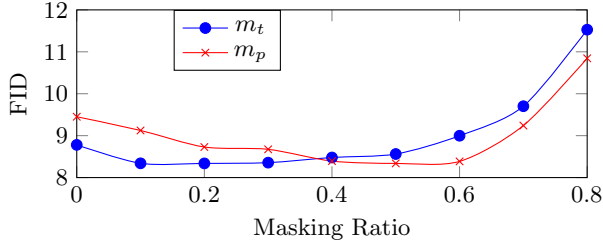
Figure 7: Masking ratio vs. FID. We set $m_p = 0.5$ when varying m_t, and $m_t = 0.2$ when varying m_p. The masked patches help the model learn how to reconstruct occluded regions.

Patch Size	FID↓	SSIM↑	LPIPS↓	IS↑
8	8.565	0.789±0.068	0.161±0.059	3.386±0.101
16	**8.338**	**0.795±0.067**	**0.158±0.059**	3.469±0.098
32	8.401	0.784±0.067	0.168±0.058	3.503±0.100
64	8.898	0.776±0.640	0.169±0.056	**3.555±0.139**

Table 5: Ablations over patch sizes on DeepFashion. We find patch size 16 works best for our model.

degrading the model's capacity for long-range texture patterns. The baseline approach *Patch Concat* avoids the boundary issue by encoding each patch separately, limiting the receptive field to be at most the patch size. This approach performs well, but the dual-scale approach still yields the best overall performance.

Effects of using three branches. In Table 4(c), *Pose Concat* removes the source pose branch and concatenates the source pose representation channel-wise to the texture input. Although *Pose Concat* has provided the geometry information for the permuted source pose, not directly learning the geometric transformation between postures through separate branches still leads to a significant drop in perceptual metrics. We also provide the results of *w/o. Texture Match*, which removes the texture-to-pose match in our cross-attention module (Fig. 4). Without texture match, pose match alone is not sufficient to learn the spatial support of clothing. Consider the task of transferring a flared skirt: much of the skirt's surface cannot be directly associated with a position on the human pose map. Therefore, adding the texture-to-pose correspondence (texture match) can help the model learn this type of spatial support. As reported in Table 4, our full model PT^2 that incorporates the texture match shows better performance on all perpetual metrics.

Effects of patch size. The amount of pose information within a patch is directly related to its size, which can affect our model's performance. Table 5 re-

ports the effect of varying the patch size of our image permutation function from Sec. 3.1.1. We find that smaller patches improve disentanglement, but result in more boundary-crossings. Larger patch sizes have fewer crossings, but results in more entangling of pose and texture. Table 5 shows that a patch size of 16 reflects the best balance of these two factors. In addition, we observe that even the suboptimal large patch sizes does not deteriorate downstream task performance too significantly. This arises due to the high masking rate of 16×16 patches in both the image and source pose in addition to the permutations (Sec. 3.1.1). Thus, the network is forced to learn to sample correct textures for the masked pixels based on the target pose, helping to avoid an identity mapping of the input person when the patch size is closer to the image size.

Ablations on masking ratios. Patch masking helps the model learn to reconstruct occluded regions. Fig. 7 illustrates this effect by varying the masking ratios of the source pose branch (m_p) and the texture branch (m_t), respectively. We find that when $0.1 \leq m_t \leq 0.3$ and $0.4 \leq m_p \leq 0.6$, the results are mostly flat, demonstrating that we can obtain gains with minimal tuning. However, lower values cause the model to overfit to visible patches, and higher values hurts performance due to many missing texture/pose details.

5. Conclusion

We propose PT^2, a self-driven human pose transfer method that randomly permutes image patches, helping to disentangle pose from texture, then reconstructs the original image from the permuted input. The approach helps to achieve patch-level disentanglement and detail-preserving texture transfer. Thus, PT^2 is capable of more accurately reconstructing garment textures in pose transfer tasks than prior work. We also show that using a dual-scale encoder can be resilient to permutation noise while aligning pose and textures. Experiments on DeepFashion and Market-1501 show that our model improves the image quality of self-driven approaches, with higher user preference scores than prior work. Moreover, PT^2 obtains comparable objective and subjective results to most pose transfer methods supervised by paired data.

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*, 40(6):1–11, 2021. 2

[2] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019. 2

[3] 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 International Conference on Computer Vision*, 2021. 1

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2009. 5

[5] Aysegul Dundar, Kevin J Shih, Animesh Garg, Robert Pottorf, Anrew Tao, and Bryan Catanzaro. Unsupervised disentanglement of pose, appearance and background from images and videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3883–3894, 2021. 2, 5

[6] 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*, 2018. 2, 6

[7] Jianglin Fu, Shikai Li, Yuming Jiang, Kwan-Yee Lin, Chen Qian, Chen Change Loy, Wayne Wu, and Ziwei Liu. StyleGAN-human: A data-centric odyssey of human generation. In *Proceedings of the European Conference on Computer Vision*, 2022. 2

[8] Chen Gao, Si Liu, Ran He, Shuicheng Yan, and Bo Li. Recapture as you want. *arXiv preprint arXiv:2006.01435*, 2020. 2

[9] Artur Grigorev, Artem Sevastopolsky, Alexander Vakhitov, and Victor Lempitsky. Coordinate-based texture inpainting for pose-guided human image generation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019. 2

[10] 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*, 2018. 2, 4

[11] Xintong Han, Xiaojun Hu, Weilin Huang, and Matthew R Scott. Clothflow: A flow-based model for clothed person generation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016. 5

[13] 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*, 2017. 6

[14] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and superresolution. In *Proceedings of the European Conference on Computer Vision*, 2016. 5

[15] Tero Karras, Samuli Laine, and Timo Aila. A stylebased generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019. 5

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, 2014. 4

[17] Wen Liu, Zhixin Piao, Zhi Tu, Wenhan Luo, Lin Ma, and Shenghua Gao. Liquid warping GAN with attention: A unified framework for human image synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5114–5132, 2022. 1, 2, 5

[18] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016. 2, 5

[19] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, 2019. 1

[20] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2015. 5

[21] Dominik Lorenz, Leonard Bereska, Timo Milbich, and Bjorn Ommer. Unsupervised part-based disentangling of object shape and appearance. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019. 2, 8

[22] Zhengyao Lv, Xiaoming Li, Xin Li, Fu Li, Tianwei Lin, Dongliang He, and Wangmeng Zuo. Learning semantic person image generation by region-adaptive normalization. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2021. 2, 6

[23] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. Disentangled person image generation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 3, 4, 6

[24] Tianxiang Ma, Bo Peng, Wei Wang, and Jing Dong. MUST-GAN: Multi-level statistics transfer for selfdriven person image generation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2021. 1, 2, 3, 4, 5, 6, 7

[25] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proceedings of the European Conference on Computer Vision*, 2016. 2

[26] Albert Pumarola, Antonio Agudo, Alberto Sanfeliu, and Francesc Moreno-Noguer. Unsupervised person image synthesis in arbitrary poses. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018. 2, 3, 4

[27] Yurui Ren, Xiaoqing Fan, Ge Li, Shan Liu, and Thomas H Li. Neural texture extraction and distribution for controllable person image synthesis. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022. 2, 6

[28] 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 conference on Computer Vision and Pattern Recognition*, 2020. 2, 6

[29] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, 2016. 6

[30] Soubhik Sanyal, Alex Vorobiov, Timo Bolkart, Matthew Loper, Betty Mohler, Larry S Davis, Javier Romero, and Michael J Black. Learning realistic human reposing using cyclic self-supervision with 3d shape, pose, and appearance consistency. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 2

[31] Kripasindhu Sarkar, Vladislav Golyanik, Lingjie Liu, and Christian Theobalt. Style and pose control for image synthesis of humans from a single monocular view. In *arXiv preprint arXiv:2102.11263*, 2021. 2

[32] Kripasindhu Sarkar, Dushyant Mehta, Weipeng Xu, Vladislav Golyanik, and Christian Theobalt. Neural re-rendering of humans from a single image. In *Proceedings of the European Conference on Computer Vision*, 2020. 2

[33] Ruizhi Shao, Hongwen Zhang, He Zhang, Mingjia Chen, Yan-Pei Cao, Tao Yu, and Yebin Liu. Doublefield: Bridging the neural surface and radiance fields for high-fidelity human reconstruction and rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2014. 5

[35] Sijie Song, Wei Zhang, Jiaying Liu, and Tao Mei. Unsupervised person image generation with semantic parsing transformation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 5, 6, 7

[36] Hao Tang, Song Bai, Li Zhang, Philip HS Torr, and Nicu Sebe. XingGAN for person image generation. In *Proceedings of the European Conference on Computer Vision*, 2020. 4

[37] Jilin Tang, Yi Yuan, Tianjia Shao, Yong Liu, Mengmeng Wang, and Kun Zhou. Structure-aware person image generation with pose decomposition and semantic correlation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. 2

[38] 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. 6

[39] Zijian Wang, Xingqun Qi, Kun Yuan, and Muyi Sun. Self-supervised correlation mining network for person image generation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022. 1, 2, 3, 4, 5, 6

[40] Tianhan Xu, Yasuhiro Fujita, and Eiichi Matsumoto. Surface-aligned neural radiance fields for controllable 3d human synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2

[41] Han Yang, Ruimao Zhang, Xiaobao Guo, Wei Liu, Wangmeng Zuo, and Ping Luo. Towards photorealistic virtual try-on by adaptively generating-preserving image content. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2020. 1

[42] Hongtao Yang, Tong Zhang, Wenbing Huang, Xuming He, and Fatih Porikli. Towards purely unsupervised disentanglement of appearance and shape for person images generation. In *Proceedings of the International Workshop on Human-Centric Multimedia Analysis*, 2020. 1

[43] Ruiyun Yu, Xiaoqi Wang, and Xiaohui Xie. Vtnfp: An image-based virtual try-on network with body and clothing feature preservation. In *Proceedings of the IEEE international conference on computer vision*, 2019. 1

[44] Jinsong Zhang, Kun Li, Yu-Kun Lai, and Jingyu Yang. Pise: Person image synthesis and editing with decoupled GAN. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2021. 2, 6, 7

[45] Pengze Zhang, Lingxiao Yang, Jian-Huang Lai, and Xiaohua Xie. Exploring dual-task correlation for pose guided person image generation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022. 1, 2, 4, 5, 6, 7

[46] Quan Zhang, Jianhuang Lai, Zhanxiang Feng, and Xiaohua Xie. Seeing like a human: Asynchronous learning with dynamic progressive refinement for person reidentification. *IEEE Transactions on Image Processing*, 31:352–365, 2021. 1

[47] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018. 6

[48] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-

identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, 2015. 2, 5

[49] Xinyue Zhou, Mingyu Yin, Xinyuan Chen, Li Sun, Changxin Gao, and Qingli Li. Cross attention based style distribution for controllable person image synthesis. In *Proceedings of the European Conference on Computer Vision*, 2022. 1, 6, 7

[50] 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 conference on Computer Vision and Pattern Recognition*, 2019. 2