COLLECTING THE PUZZLE PIECES: DISENTANGLED SELF-DRIVEN HUMAN POSE TRANSFER BY PERMUTING TEXTURES

Anonymous authors
Paper under double-blind review

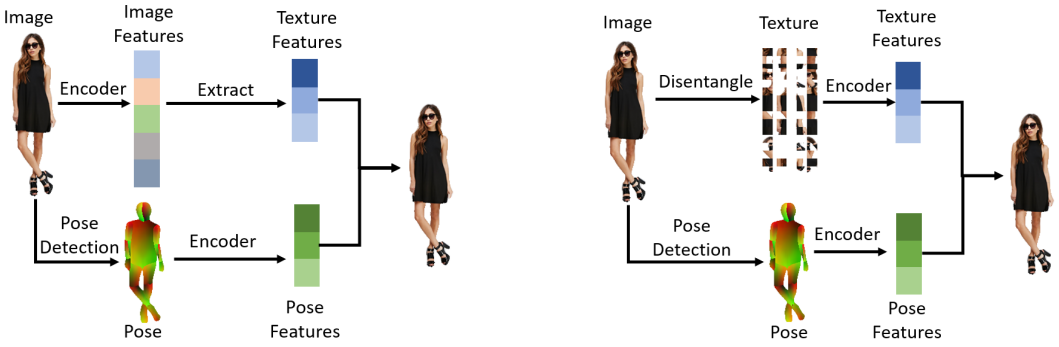
ABSTRACT

Human pose transfer aims to synthesize a new view of a person under a given pose. Recent works achieve this via self-reconstruction, which disentangles pose and texture features from the person image, then combines the two features to reconstruct the person. Such feature-level disentanglement is a difficult and ill-defined problem that could lead to loss of details and unwanted artifacts. In this paper, we propose a self-driven human pose transfer method that permutes the textures at random, then reconstructs the image with a dual branch attention to achieve image-level disentanglement and detail-preserving texture transfer. We find that compared with feature-level disentanglement, image-level disentanglement is more controllable and reliable. Furthermore, we introduce a dual kernel encoder that gives different sizes of receptive fields in order to reduce the noise caused by permutation and thus recover clothing details while aligning pose and textures. Extensive experiments on DeepFashion and Market-1501 shows that our model improves the quality of generated images in terms of FID, LPIPS and SSIM over other self-driven methods, and even outperforming some fully-supervised methods. A user study also shows that among self-driven approaches, images generated by our method are preferred in 72% of cases over prior work.

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 (Yang et al., 2020b; Cui et al., 2021; Yu et al., 2019), controllable person image manipulation (Cui et al., 2021; Liu et al., 2021) and person re-identification (Zhang et al., 2021b). Recent work has focused on using paired image data (*i.e.*, two images of the same person before and after reposing) (Zhou et al., 2022; Zhang et al., 2022), but collecting such data can be very labor intensive. Although self-driven methods have been proposed to train pose transfer models without paired data (Ma et al., 2021; Song et al., 2019), there still remains two major challenges: how to disentangle texture and pose, and how to preserve texture details across changes in pose. As illustrated in Figure 1a, prior research attempts to achieve the pose and texture disentanglement at a feature level (Ma et al., 2018; Yang et al., 2020a; Ma et al., 2021; Wang et al., 2022). However, without direct supervision from pose-invariant textures, disentangling texture features from the person image while also preserving specific clothing details in the disentangled features is a difficult and ill-defined problem (Locatello et al., 2019). Small imbalances between pose and texture could leave obvious artifacts in the generated images.

In this paper, we propose Pose Transfer by Permuting Textures (PT^2), a self-driven pose transfer model using image-level disentanglement to represent detailed clothing patterns in any target pose. As shown in Figure 1b, a key novelty is our input permutation function that disentangles the raw inputs of texture and pose. Our method does not need supervised pose-invariant textures because most pose information has been removed by the permutation. The input permutation function creates a disentangled texture sample space by randomly reordering the texture patches on the person such that the source pose cannot be recovered from the permuted textures. This approach is similar in spirit to self-supervised representation learning methods that use jigsaw puzzle solving to learn a good feature representation (Noroozi & Favaro, 2016; Carlucci et al., 2019), where the pretext task divides the image into large patches and attempts to infer their relative positions by using the inherent



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

Figure 1: Pose transfer methods trained without supervision extract disentangled texture and pose representations and then learn to reconstruct the original image. **(a)** Recent work uses separate encoders to disentangle texture and pose (Pumarola et al., 2018; Ma et al., 2018; 2021; Wang et al., 2022). However, pose information may still appear in the texture features, and without supervision, disentangling them is difficult (Locatello et al., 2019). **(b)** Our approach disentangles textures from pose by permuting the image patches, effectively eliminating pose information, which enables our approach to disentangle pose and texture features better than prior work.

geometry information within each patch. However, we differ in that our goal is to sample relevant patches based on the target posture. We make the patch much smaller in order to remove the position information and thus disentangle pose and texture. Furthermore, we mask some of the textures to force the generator to infer occluded and unseen regions, such as t-shirt occluded by crossed arms.

One challenge we face is that the permutation of textures causes loss of shape and relative position information, which has two significant consequences. First, the model cannot recognize different body parts and garments. For example, the generator could use texture from leggings to synthesize a top tank. Second, it makes the length of clothing items unknown because of lack of relative position of clothing pieces. The first issue can be easily solved by combining the person with a human parsing map to give a semantic identifier for each pixel (Ma et al., 2021). Whereas for the second problem, we need an additional sample space in the model that provides relative position information. This inspires us to add a pose branch, where we use the dense pose representation (Güler et al., 2018) as the sample space to provide position information for each pixel after permuting the textures. Each pixel value in the space indicates the position of that pixel under the texture coordinate system (Güler et al., 2018). In addition, we find that using different kernel sizes in the convolutional layers of our dual branch attention module provides a better representation for our task.

Our main contributions are:

- We propose Pose Transfer by Permuting Textures (PT^2), a self-driven pose transfer model that utilizes input permutation to transfer clothing patterns to the target pose without using paired images for supervision.
- The proposed pose branch in PT^2 provides relative geometry information for the permuted textures, which helps recover shape and length after pose transfer. In addition, different kernel sizes are introduced in the branch, which can reduce the noise caused by input permutation and thus preserve clothing details while aligning pose and textures.
- Extensive experiments on DeepFashion (Liu et al., 2016) and Market-1501 (Zheng et al., 2015) show that PT^2 significantly improves the image quality of self-driven approaches. A user study reports that our method are preferred in 72% of cases over the state-of-the-art.

2 Related Work

Pose transfer with paired images. Methods trained with paired images aim to learn the complex non-rigid deformation of clothing items. In (Zhu et al., 2019; Zhang et al., 2022; Ren et al., 2022; Gao et al., 2020), this transformation is learned via soft attention that aggregates source image features with weighted sampling. Zhang et al. (2021a); Lv et al. (2021) further use semantic parsing maps as guidance to control the style of each body part. The major difficulty in such feature-level

attention is that clothing details could be washed out in lower-resolution feature maps. To preserve these details after pose transfer, flow-based methods haven been proposed to approximate a dense flow field from the source to the target person. Han et al. (2019) introduced a pyramid feature network that outputs a pixel-level flow filed. Tang et al. (2021); Ren et al. (2020) further combined soft attention with dense flow to learn more accurate estimations. However, since the learned flow can only copy existing pixels in the source image to the target, it might fail at inferring occluded and unseen parts of the person. Grigorev et al. (2019); Sarkar et al. (2020); Albahar et al. (2021) explored inpainting 2D partial texture to 3D full texture in the UV space, and then projecting it back to the 2D pose. Although these methods can produce high-quality person images, they all require strong supervision from paired data, which might be difficult to collect in some real-world scenarios.

Pose transfer with unpaired images. In a self-driven setting where paired data is absent, it is more difficult to transfer the pose without losing texture details. Without supervision, the generated images tend to have repeated texture patterns and edge blurring (Wang et al., 2022). Prior work has addressed this problem by disentangling the texture and posture at a feature level. Early attempts produced poor quality images for large pose deformations (Pumarola et al., 2018; Esser et al., 2018; Ma et al., 2018). Song et al. (2019) introduced a generated target parsing map to a cycle-GAN pose transfer model, which requires paired segmentation maps for training the human parsing model. Sanyal et al. (2021) presents a 3D based reposing approach with appearance visibility inference. Wang et al. (2022) used part-wise encoder to learn texture features that are less correlated with pose, where global pose information can still be inferred from the texture features. The model in (Ma et al., 2021) first computes region-wise image features, and then takes their mean and variance as the texture features to be integrated with the pose representation. This helps erase the pose information in the texture features, but, as we show, it may miss clothing details. In contrast to these methods, our approach disentangles the texture at an image-level by permutation, and uses a dual kernel encoder with dual branch attention to transfer detailed clothing patterns to the target pose.

3 Self-Driven Pose Transfer by Permuting Textures (PT^2)

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 in I_s and wearing the same clothes in a target posture P_t. Models requiring paired data use the target pose P_t and the target image I_t for training, but our approach needs only information derived from the source image. Specifically, the target pose/image in training is identical to the source pose/image. In inference, replacing the target pose with a different one enables pose transfer.

PT^2 contains a pose transfer network (Sec. 3.1) that synthesizes a new view of the person in its target pose, and a background inpainting network that infers its full background (Sec. 3.2). The generated person and its full background are combined to create the final reconstructed image.

3.1 Pose Transfer Network

The objective of the pose transfer network is to take the foreground person in the source image and generate a new view of them in a different pose. Figure 2 gives the overall architecture, which contains two branches: a pose branch that learns the geometric transformation function from pose P_s to pose P_t, and a texture branch that learns to transfer the textures of the person E_s to pose P_t. The permuted inputs (Sec. 3.1.1) from the two branches are first encoded with dual kernel encoder (Sec. 3.1.2), and then merged in a dual branch attention module (Sec. 3.1.3) to be decoded into the generated person $\hat{E}_d$ and its segmentation S.

3.1.1 Input Permutation

Inputs to the Texture Branch. To guide the texture transfer with a posture in a self-driven way, we first need to disentangle the pose and textures in the image. The posture can be derived by a DensePose model (Güler et al., 2018) pretrained on COCO (Lin et al., 2014), which gives a 2D UV coordinate representation P_s. However, the texture representation should not simply be the source person E_s itself, as it is obviously entangled with posture. To erase the pose information from E_s, we create a texture sample space by dividing the image into $k \times k$ squares, referred to as "patches," and then shuffling their locations. Intuitively, when the patch size k is sufficiently small, the original posture cannot easily be retrieved from the permutation. Additionaly, 20% of the patches are masked

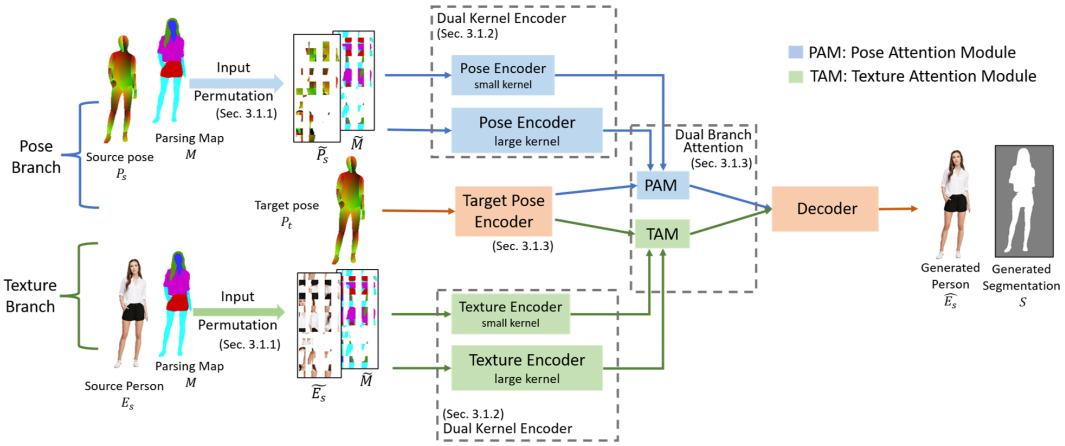


Figure 2: The pose transfer network in PT^2. During training, the target pose P_t is the same as the source pose P_s. The network takes the source person E_s, source parsing map M and source pose P_s as inputs. In the pose branch and texture branch, the inputs are first permuted (Sec. 3.1.1) to create the corresponding sample space, which is encoded with dual kernel encoders (Sec. 3.1.2). Then the encoded features are sampled in a dual branch 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.

to encourage the model to learn occluded regions. Formally, let RandMask$(\cdot)$ be the input permuting function. The inputs of the texture attention branch become $[\tilde{E}_s; \tilde{M}] = \text{RandMask}([E_s; M], m_t)$, where $[;]$ means concatenation and m_t is the masking rate. We set $m_t = 0.2$ in our experiments. The permuted tetxures $[\tilde{E}_s; \tilde{M}]$ are given as inputs to the dual-kernel texture encoder (Sec. 3.1.2).

Inputs to the Pose Branch. While prior work only uses a texture branch to transfer texture to the target pose (Pumarola et al., 2018; Ma et al., 2018; 2021; Wang et al., 2022), we propose a pose branch that provides relative geometry information for the permuted textures, which helps recover shape and length after pose transfer. To learn a powerful pose transformation function that supports large pose variations, the source pose in this branch is permuted the same way as the textures. In addition, we mask 50% of the source pose representation to force the model to learn the inherit symmetry in human body. The inputs to the pose branch become $[\tilde{P}_s; \tilde{M}] = \text{RandMask}([P_s; M], m_p)$, where $m_p = 0.5$. Note that the inputs of the texture and pose branches are permuted the same way so they are spatially aligned in the dual branch attention module (see Sec. 3.1.3). The permuted pose representations $[\tilde{P}_s; \tilde{M}]$ are given as inputs to the dual-kernel pose encoder (Sec. 3.1.2).

3.1.2 Dual Kernel Encoder

Following (Pumarola et al., 2018; Ma et al., 2018; 2021; Wang et al., 2022), we utilize separate encoders to learn both the texture and the pose information. However, in addition to providing permuted inputs from Sec. 3.1.1 to help further disentangle texture from pose, another way our approach differs is that we use multiple kernel sizes in the encoder's convolutional layers. More formally, the texture/pose encoder learns a multi-dimensional feature map $F \in \mathbb{R}^{H \times W \times d}$ from the permuted texture/posture, where each vector $v \in \mathbb{R}^d$ in the feature map has a certain receptive field in the image. Let l denote the length of the receptive field and s be the stride of F (both are measured by number of pixels in the image). Generally, larger receptive field is capable of learning more diverse features, and thus l is usually much larger than s. However, for a receptive field that crosses the boundary between two permuted image patches, the pixels within the field could be spatially faraway and irrelevant in the original image. In the left picture of Figure 3, the two black squares denote two adjacent receptive fields. While the left square lying within an image patch are seeing a consistent pattern (*e.g.*, face), the right square crossing the boundary are seeing two distinct patterns (*e.g.*, face and shoes). This could introduce high volume of noise to the feature vector v, preventing the model from recognizing true clothing patterns.

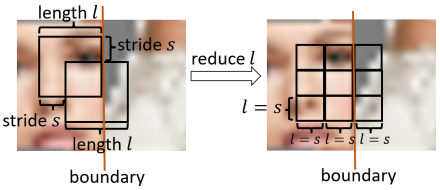


Figure 3: Illustration of the receptive field in the large-kernel encoder (left) and small-kernel encoder (right). The kernel size l is reduced to avoid overlap between receptive fields (see Sec. 3.1.2).

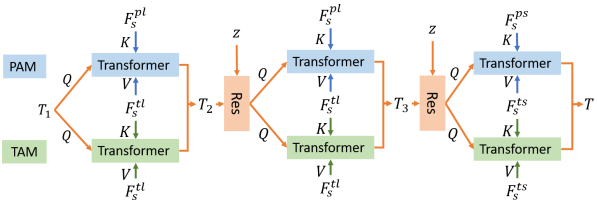


Figure 4: Dual branch attention module in Sec. 3.1.3. The top flow is PAM and the bottom flow is TAM. Res represents a residual layer. The cross-attention mechanism aligns the permuted texture with the target pose.

To solve the above issue, we introduce an additional pose encoder with reduced kernel size such that $l = s$, which can be regarded as a Multi Layer Perceptron over image patches. We select our 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 Figure 3, this design avoids the overlap between receptive fields, enabling the convolutional kernel to learn a consistent pattern within its own receptive field. Note that large kernel size is still necessary as it has more parameters and larger receptive filed for learning larger image patterns. Therefore, by combining encoders with a large and small kernel sizes our model is capable of learning more diverse features. The outputs of the dual kernel encoders in both texture branch and pose branch are fed to the dual branch attention module in Sec. 3.1.3.

3.1.3 Dual Branch Attention

We use a cross-attention transformer (Tang et al., 2020; Tan et al., 2021; Zhang et al., 2022) to align texture and pose features. Specifically, we use a Pose Attention Module (PAM) for the pose branch and a Texture Attention Module (TAM) for the texture branch. Let F_s^{pl}, F_s^{ps} represent the output feature map of the large-kernel pose encoder and the small-kernel pose encoder in the pose branch, respectively. Similarly, F_s^{tl}, F_s^{ts} are the encoded features from the texture encoders in the texture branch. T_1 is the feature map of the target pose P_t encoded by the target pose encoder, which is implemented with six convolutional layers. As shown in Figure 4, both PAM and TAM are composed of three cross-attention vision transformers formulated as $\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d}}) \cdot V$. With cross-attention the model can sample textures based on the target pose to generate a person image. In the first two transformer layers, the pose attention in the pose branch is learned from the correlation between the source pose and the target pose, which is computed as:

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

Here, $W_i^{pq}, W_i^{pk}, W_i^{pv}$ are learnable projection matrices. Similarly, the texture attention in the texture branch is formulated as the correlation between the texture and the target pose:

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

After each transformer layer, a residual layer is appended as in Figure 4. A random noise vector z is injected to the residual layer as the affine transformation parameters of the feature map T_i to prevent mode collapse (Karras et al., 2019).

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}) in order to reconstruct more detailed information. 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.

PAM in the dual branch attention module learns geometric transformation between different postures, and TAM samples the given textures based on the target pose. Fusing the two source 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

As in (Dundar et al., 2021; Liu et al., 2021), we use a separate background inpainting network, implemented using UNet (Long et al., 2015), to infer the background pixels of the masked foreground region. However, we found if we only mask out the foreground segmentation S, the model would ignore the unknown background in the mask and reconstruct only known pixels during inference. This is because the background area is always visible in the source image in self-supervised training, so the network does not learn how to infer missing areas. Therefore, we expand the mask to the whole bounding box of the detected person in order to create invisible background areas during training. Let $\hat{B}$ be the inpainted background. Given the generated person $\hat{E}_s$ and its segmentation mask S produced by the pose transfer network, the final reconstructed image is: $\hat{I}_s = S \odot \hat{E}_s + (1-S) \odot \hat{B}$.

3.3 LOSS FUNCTIONS

We train our model using an adversarial loss that can be written as:

$$L_{adv} = D(\hat{I}_s)^2 + (1 - D(I_s))^2 + D_p([\hat{I}_s; P_s])^2 + (1 - D_p([I_s; P_s]))^2. \tag{3}$$

where D and D_p represent different discriminators. 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. To ensure correctness of our image generation, we use three different loss functions that capture different desired properties. The first is a simple reconstruction loss,

$$L_{rec} = ||\hat{I}_s - I_s||_1. \tag{4}$$

In addition, we use a perceptual loss (Johnson et al., 2016) that encourages both the ground truth and reconstructed image have similar semantic properties,

$$L_{perc} = \sum_i ||\phi^i(\hat{I}_s) - \phi^i(I_s)||_1, \tag{5}$$

where ϕ^i is the ith layer of a VGG model (Simonyan & Zisserman, 2014) pretrained on ImageNet Deng et al. (2009). Finally, we use a style loss that penalizes discrepancies on colors and textures using the Gram matrix $\mathbb{G}(\cdot)$ of the features,

$$L_{style} = \sum_i ||\mathbb{G}(\phi^i(\hat{I}_s)) - \mathbb{G}(\phi^i(I_s))||_1. \tag{6}$$

Thus, our total loss can be written as,

$$L_{total} = \lambda_1 L_{adv} + \lambda_2 L_{rec} + \lambda_3 L_{perc} + \lambda_4 L_{style}. \tag{7}$$

where λ_{1-4} are scalar hyperparameters. Additional training details are provided in Appendix A.

4 EXPERIMENTS

Datasets. We evaluate our proposed model on two benchmarks: DeepFashion (Liu et al., 2016) and Market1501 (Zheng et al., 2015). DeepFashion contains 52,712 high-quality images with a clean background. Market-1501 has 32,668 low-resolution images with various lighting conditions and a noisy background. Following Zhang et al. (2022); Wang et al. (2022), we select 8,570 test pairs with a resolution of 256×256 on DeepFashion, and 12,000 test pairs with a resolution of 128×64 on Market-1501. As in prior self-driven methods (Ma et al., 2021; Wang et al., 2022), we use 37,332 training images for DeepFashion and 12,112 training images for Market-1501.

Metrics. Following (Ma et al., 2021; Wang et al., 2022), we use Structural Similarity Index Measure (SSIM) (Wang et al., 2004), Frechet Inception Distance (FID) (Heusel et al., 2017) , Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al., 2018) and Inception Score (IS) (Salimans et al., 2016) to evaluate the quality of the synthesized images. Among these metrics, SSIM measures structural similarity in the pixel space. FID computes Wasserstein-2 distance between two distributions. LPIPS evaluates perceptual similarity in deep network's feature space. We use the default AlexNet as LPIPS's backbone. IS assesses the quality of images generated by adversarial training. On Market-1501, we add Masked-SSIM and Masked-LPIPS computed on the target person region to exclude the influence of the irrelevant background.

Table 1: Pose transfer results for 256×256 resolution images on DeepFashion. All results for prior work are taken from the original papers or produced with the author's source code.

Method	FID↓	SSIM↑	LPIPS↓	IS↑
Supervised by paired images				
PATN (Zhu et al., 2019)	24.071	0.770	0.299	3.141
GFLA (Ren et al., 2020)	10.573	0.707	0.234	**3.635**
PISE (Zhang et al., 2021a)	13.610	-	0.206	-
SPIG (Lv et al., 2021)	12.243	0.782	0.211	-
DPTN (Zhang et al., 2022)	11.466	0.778	0.196	-
CASD (Zhou et al., 2022)	11.373	0.725	0.194	-
NTED (Ren et al., 2022)	**6.786**	**0.808**	**0.133**	3.264
No paired images				
VU-Net (Esser et al., 2018)	23.580	0.786	0.321	3.087
E2E (Song et al., 2019)	29.900	0.736	0.238	3.441
DPIG (Ma et al., 2018)	48.200	0.614	0.284	3.228
MUST (Ma et al., 2021)	15.902	0.742	-	**3.692**
SCM-Net Wang et al. (2022)	12.180	0.751	0.182	3.632
PT^2(Ours)	**8.338**	**0.795**	**0.158**	3.469

Table 2: Pose transfer results for 128×64 resolution images on Market-1501. All results for prior work are taken from the original papers or produced with the author's source code.

Method	FID↓	SSIM↑	M-SSIM↑	LPIPS↓	M-LPIPS↓	IS↑
Supervised by paired images						
PATN (Zhu et al., 2019)	22.657	0.311	0.811	0.320	0.159	-
GFLA (Ren et al., 2020)	19.751	0.281	0.796	0.282	0.148	-
SPIG (Lv et al., 2021)	23.331	**0.315**	0.818	**0.278**	0.139	-
DPTN (Zhang et al., 2022)	18.995	0.285	-	0.271	-	-
No paired images						
PT^2(Ours)	**17.389**	0.280	**0.820**	0.314	**0.122**	**2.789**

4.1 Quantitative Results

Table 1 compares methods on the pose transfer task using DeepFashion, where our approach outperforms most methods on FID, SSIM, and LPIP by a large margin. For example, we improve FID by 4 points over the state-of-the-art. Notably, our model, *which requires no paired training data*, also achieves better performance than most supervised methods trained with paired data. Similar behavior is seen on Market-1501 (Table 2), where our self-driven PT^2 gains in Masked-SSIM and Masked-LPIPS over supervised methods. However, we note that we do perform worse according to SSIM and LPIPS, which is computed over the entire image rather than just the target person region.

To investigate the reason behind the discrepancy when we use masked regions for evaluation, we computed the SSIM scores on different body parts of the person. The average scores for background, arms, legs, clothes and head for PT^2 are: 0.237, 0.263, 0.283, 0.323, 0.337, respectively. The lowest SSIM is on the background because the dataset is collected from surveillance videos, where the background can change drastically in different time frames. This violates our assumption that the background does not change, explaining the relatively poor performance. That said, since our goal is pose transfer, the improved performance using M-SSIM and M-LPIPS demonstrates we are more successful than even the supervised methods on Market-1501 at that task.

User Study. To verify the quality of generated images, we also conducted a human evaluation on DeepFashion using Amazon Mechanical Turk. We collected 3 judgements for 50 images (150 total). Each worker was presented 3 pictures: the true image, a PT^2 generated image, and a image generated by a method from prior work. The worker was asked to pick a picture that looks most similar to the true image. Table 3 shows that among self-driven methods, more than 72% workers believe our method achieves higher fidelity in the generated images. Compared with approaches supervised by paired images, our method achieves comparable performance with an average of over 62% user preference, demonstrating the effectiveness of our proposed approach.

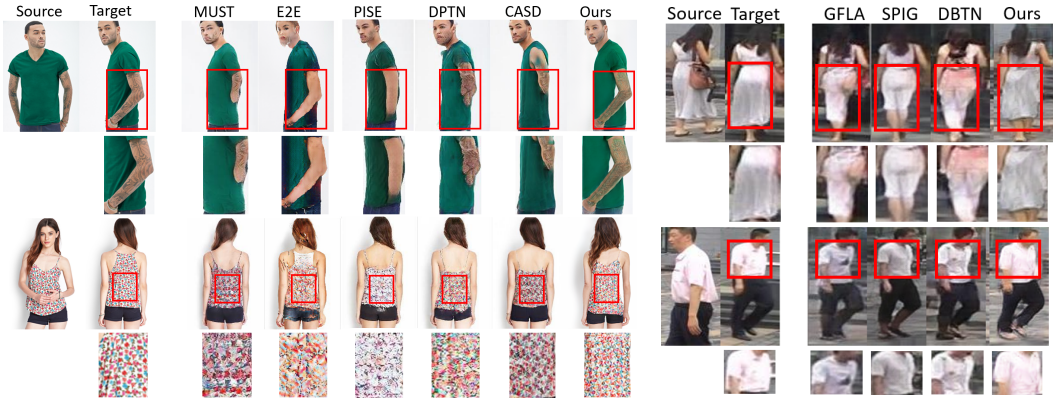


Figure 5: Qualitative pose transfer results on DeepFashion (left) and Market-1501 (right). We enlarged the area marked with a red bounding box for a better view of clothing details. Examples from prior work are generated with the author's code and pretrained models. MUST, E2E, and our PT^2 are trained with unpaired data, while the rest are supervised by paired data. These results show our approach transfers the original clothing patterns onto the target pose better than prior work.

Table 3: A/B user preferences on DeepFashion. We report how often our approach was selected as most like the ground truth image. The number that follows $\pm$ is the corresponding standard deviation. We significantly outperform methods trained without paired images. Our results were also preferred over fully supervised methods.

	Supervised by paired images			**No paired images**	
	PISE (Zhang et al., 2021a)	DPTN (Zhang et al., 2022)	CASD (Zhou et al., 2022)	E2E (Song et al., 2019)	MUST (Ma et al., 2021)
PT^2	68.7%$\pm$3.27	66.2%$\pm$3.35	53.8%$\pm$3.53	79.7%$\pm$2.84	72.6%$\pm$3.15

4.2 Qualitative Results

Pose Transfer. Figure 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. Similarly, our model learns the color pattern in the second row (left) better than other approaches. This is because the small kernel encoder in our model can capture such detailed texture and thus reconstruct it based on the target pose. On the right side of Figure 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, including failure cases, are in Appendix B.

Garment Replacement. With a given parsing map, our approach can also switch the clothing pieces on two persons. Let $I_A = (A_{pose}, A_{clt})$ be an image of person A wearing clothes A_{clt} under posture A_{pose}. To replace A_{clt} with B_{clt} in $I_B = (B_{pose}, B_{clt})$, we first align A, B's pose to A_{pose} using the proposed pose transfer method, and then replace A_{clt} to B_{clt} using their parsing maps. To fix small mis-alignment after copy-paste B_{clt} using the parsing map, the image is fed to PT^2 again for a more plausible reconstruction. Due to the shape difference of the source and reference garments (*e.g.*, jeans and shorts), the model could give different ways of combining all the clothing pieces after replacing a specific garment. For example, in Figure 6, for the person in the top second source image, the upper clothes are tucked into the shorts but untucked to the jeans. Similarly, the shorts in the top first source image are occluded by the camel t-shirt and pink jackets, but are visible when combined with other shirts. Overall, Figure 6 shows that the proposed method successfully replaces various types of garments in the given images while preserving their patterns and details.

4.3 Ablation Study

We performed an ablation study to evaluate the effectiveness of each component of our model. In Table 4, ***w/o.*** **Input Permuting** does not permute the inputs, which results in entangled pose and

Figure 6: Examples of garment replacement. The left column is the source image and the top row is reference image. All reference garments are marked with red bounding boxes.

Table 4: Ablations in DeepFashion. Compare with each ablation model, our full model PT^2 that combines all the components improves the overall performance.

Method	FID↓	SSIM↑	LPIPS↓	IS↑
Input Warping, *w/o.* Input Permuting	10.011	0.781±0.072	0.178±0.060	**3.579±0.086**
w/o. Input Permuting	10.279	0.780±0.069	0.169±0.059	3.525±0.095
w/o. Pose Branch	11.391	0.785±0.068	0.177±0.068	3.485±0.095
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
PT^2(Ours)	**8.338**	**0.795±0.067**	**0.158±0.059**	3.469±0.098

textures. **Input Warping, w/o. Input Permuting** uses Thin Plate Spline transformation (TPS) to warp the source image, which can be viewed as a mild way of disentangling the pose and texture at the image-level. ***w/o.* Pose Branch** removes the pose branch in our method. ***w/o.* small kernel** uses only large kernel in the feature encoders, which should lead to loss of clothing details. ***w/o.* large kernel** uses only small kernel in the feature encoders. We train all these ablation models under the same configuration. As shown in Table 4, our complete model PT^2 improves all the metrics, demonstrating the effectiveness of each component.

5 Conlusion

We propose PT^2, a self-driven human pose transfer method that permutes the textures at random and then reconstructs the image with dual branch attention to achieve image-level disentanglement and detail-preserving texture transfer. The introduced dual kernel encoder in the model gives different sizes of receptive fields, which can reduce the noise caused by permutation and thus recovers clothing details while aligning pose and texture. Extensive experiments on DeepFashion and Market-1501 shows that our model improves the image quality of self-driven approaches, where a user study shows our images are preferred over prior work in 72% of cases. Moreover, it obtains comparable objective and subjective results to most pose transfer methods supervised by paired data.

6 REPRODUCIBILITY STATEMENT

We include our source code in the Supplementary for other researchers to easily reproduce our results in this paper. The code has a README file with detailed instructions of running and evaluating our model. The training details and all the hyperparameters we used in our approach are provided in Appendix A.

7 ETHICS STATEMENT

The proposed method introduces image-level disentanglement of pose and texture, and provides a self-driven framework for the human pose transfer task. The results of this research could be broadly disseminated by exploiting the publicly available source code. The research is beneficial to the research community in that it builds a unified framework for self-driven pose transfer, and gives insights to other exemplar-guided image generation tasks. From the perspective of ethical considerations, our method has the potential to be used as a tool through the spread of misinformation, which echos concerns have been addressed in related machine learning research (Ramesh et al., 2022; Karnouskos, 2020). It is of utmost importance to follow certain policies and regulations against misinformation (Pennycook et al., 2020) when using these AI technologies, as well as highlight the importance of developing methods for detecting misinformation, including for media created using artificial intelligence (*e.g.*, Wang et al. (2020); Tan et al. (2020)).

REFERENCES

Badour Albahar, Jingwan Lu, Jimei Yang, Zhixin Shu, Eli Shechtman, and Jia-Bin Huang. Pose with style: Detail-preserving pose-guided image synthesis with conditional styleGAN. *ACM Transactions on Graphics*, 2021.

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

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.

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.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition*, 2009.

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.

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.

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

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.

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.

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.

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.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European conference on computer vision*, 2016.

Stamatis Karnouskos. Artificial intelligence in digital media: The era of deepfakes. *IEEE Transactions on Technology and Society*, 1(3):138–147, 2020.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

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.

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*, 2021.

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.

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.

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.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

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.

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.

Tianxiang Ma, Bo Peng, Wei Wang, and Jing Dong. MUST-GAN: Multi-level statistics transfer for self-driven person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proceedings of the European conference on computer vision*, 2016.

Gordon Pennycook, Ziv Epstein, Mohsen Mosleh, Antonio Arechar, Dean Eckles, and David Rand. Understanding and reducing the spread of misinformation online. *ACR North American Advances*, 2020.

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.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

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.

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.

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.

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.

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.

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

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.

Reuben Tan, Bryan A. Plummer, and Kate Saenko. Detecting cross-modal inconsistency to defend against neural fake news. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Reuben Tan, Bryan Plummer, Kate Saenko, Hailin Jin, and Bryan Russell. Look at what i'm doing: Self-supervised spatial grounding of narrations in instructional videos. In *Advances in Neural Information Processing Systems*, 2021.

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.

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.

Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot...for now. In *IEEE conference on computer vision and pattern recognition*, 2020.

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.

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.

Han Yang, Ruimao Zhang, Xiaobao Guo, Wei Liu, Wangmeng Zuo, and Ping Luo. Towards photo-realistic virtual try-on by adaptively generating-preserving image content. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2020a.

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*, 2020b.

Yasin Yaz, Chuan-Sheng Foo, Stefan Winkler, Kim-Hui Yap, Georgios Piliouras, Vijay Chandrasekhar, et al. The unusual effectiveness of averaging in GAN training. In *International Conference on Learning Representations*, 2019.

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.

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*, 2021a.

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.

Quan Zhang, Jianhuang Lai, Zhanxiang Feng, and Xiaohua Xie. Seeing like a human: Asynchronous learning with dynamic progressive refinement for person re-identification. *IEEE Transactions on Image Processing*, 31:352–365, 2021b.

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.

Ziwei Zhang, Chi Su, Liang Zheng, and Xiaodong Xie. Correlating edge, pose with parsing. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

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.

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.

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.

Figure 7: Additional pose transfer examples on DeepFashion.

A Training Details.

We use AdamW optimizer (Loshchilov & Hutter, 2019) for training with $\beta_1 = 0.5, \beta_2 = 0.999$. The initial learning rate is set to 10^{-3} and decays to 2×10^{-4} after five starting epochs. The trade-off parameters are set to $\lambda_1 = 2.0, \lambda_2 = 5.0, \lambda_3 = 0.5, \lambda_4 = 150$ in all experiments. The patch size is 16×16 for DeepFashion and 8×8 for Market-1501. To stabilize the training, we use the EMA strategy Yaz et al. (2019) to average the learned weights of the generator. Our pose representation is predicted by DensePose (Güler et al., 2018) and the parsing maps are obtained from CorrPM (Zhang et al., 2020). We found that the predicted dense pose in Market-1501 has poor quality as the image resolution is too low (128×64) for the DensePose model. Therefore, we use an offline super resolution model Liang et al. (2021) to upsample the Market-1501 images to 512×256, get dense pose from these images, and then downsample the pose to the original image resolution (128×64) for our pose transfer task. We also add human keypoints predicted from OpenPose (Cao et al., 2019) as part of the pose representation to improve the accuracy of predicted posture on Market-1501.

B Discussions

Failure case analysis. Figure 7 provides several successful examples generated by the propsoed method on DeepFashion. However, one limitation of our model is that it relies on the segmentation map and DensePose prediction of the source image to obtain semantic and position information for the permuted textures. Thus, we found the accuracy of human parser and DensePose model greatly affects the transfer results. Figure 8 shows several failed examples due to this type of inaccuracy. In

Figure 8: Failure cases in DeepFashion. Many failures are due to incorrect predictions of the source UV map and source parsing map.

Figure 9: Generated images of ablations of our model. Each component of our model improves the transfer of shape information and detailed clothing patterns, resulting in our full model obtaining the best results.

the first row, the coat wrapped around the dress was misrecognized as part of the dress in the parsing map, for which our generated back view incorrectly mixes up their textures. Similarly, the skirt in the second row was classified as shorts in the parsing map. As a result, our generator infers the occluded clothing piece as shorts in the front view. In the last row, the color of skirt is half-black and half-white because the skirt piece was not identified in the parsing map. **More analysis on ablations.** We present some visualized examples in Figure 9 to show the functionality of each component of our model. It's clear that models with less perturbation of the input texture (*i.e.*, *w/o. Input Permuting* and *Input Warping*) fail at large pose changes (*e.g.*, from back view to front full view in the bottom row). Removing the pose branch (*w/o.* Pose Branch) causes loss of length information, resulting in

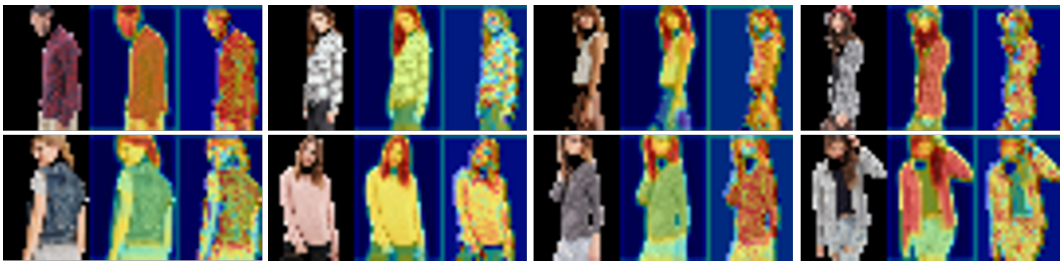


Figure 10: Visualized feature map of the encoded texture features. The feature map is overlaid with the source image. The source image is downsampled to the resolution of the feature map. Each triplet includes a downsampled source image, the feature map from large-kernel encoder, and the feature map from small-kernel encoder. Red indicates higher value and blue means smaller value.

extended dress in the second row. Without the small-kernel encoder (*w/o.* small kernel), the ablation model correctly transfers color and shape, but fails to recover complex clothing patterns and details in the third row. Without large kernel (*w/o.* large kernel), the model can correctly reconstruct clothes with singular color, but is less capable of transferring detailed textures (see the third and fourth row). To see if the large-kernel encoder is learning certain low-level information from permuted patches, we also tried replacing the inputs of the large-kernel encoder with heavily Gaussian blurred image without permutation. From the examples, we can see that images generated by *w.* blur are much worse compared to the full model. This suggests that features learned by large-kernel encoder from the permuted image might have richer information than Gaussian blurred texture.

To further explore the differences of texture features learned by the large-kernel encoder and the small-kernel encoder, we sum up the encoded feature maps across all channels in the texture branch, and normalize their values to be in range $[0, 1]$. Then we downsample the image to the resolution of the feature map and overlay the normalized feature map with the downsampled source image. In Figure 10, each triplet includes the downsampled source image, the feature map from large-kernel encoder, and the feature map from small-kernel encoder. Feature map given by large-kernel encoder (middle image in each triplet) appears to be much smoother than that of the small-kernel encoder (right image in each triplet). This suggests that large-kernel encoder might be learning coarse information from the clothing piece (*e.g.*, color and shape), while small-kernel encoder is learning more fine-grained patterns (*e.g.*, stripe and pleat).