

PoNA: Pose-guided Non-local Attention for Human Pose Transfer

Kun Li[†], *Member, IEEE*, Jinsong Zhang[†], Yebin Liu, *Member, IEEE*, Yu-Kun Lai, *Member, IEEE*,
and Qionghai Dai, *Senior Member, IEEE*

Abstract—Human pose transfer, which aims at transferring the appearance of a given person to a target pose, is very challenging and important in many applications. Previous work ignores the guidance of pose features or only uses local attention mechanism, leading to implausible and blurry results. We propose a new human pose transfer method using a generative adversarial network (GAN) with simplified cascaded blocks. In each block, we propose a pose-guided non-local attention (PoNA) mechanism with a long-range dependency scheme to select more important regions of image features to transfer. We also design pre-posed image-guided pose feature update and post-posed pose-guided image feature update to better utilize the pose and image features. Our network is simple, stable, and easy to train. Quantitative and qualitative results on Market-1501 and DeepFashion datasets show the efficacy and efficiency of our model. Compared with state-of-the-art methods, our model generates sharper and more realistic images with rich details, while having fewer parameters and faster speed. Furthermore, our generated images can help to alleviate data insufficiency for person re-identification.

Index Terms—Human pose transfer, generative adversarial network (GAN), attention

I. INTRODUCTION

HUMAN pose transfer, which synthesizes a new image for a target person in a new pose, is a very significant task in many applications such as data augmentation for person re-identification [1], image processing [2], and video generation [3]. Given a condition image of a person and an arbitrary pose, human pose transfer system generates realistic images of the same person in the specified pose, as illustrated in Figure 1.

Many promising frameworks have been proposed for human pose transfer [3]–[9]. In order to generate realistic images, three main ideas are used in previous work. The first kind of methods [4]–[6], [9] apply an encoder-decoder framework to implicitly accomplish the transformation. However, they fuse the pose and image information through simple guide mechanism without utilizing the pose information for guidance, which leads to blurry and implausible results. The second

This work was supported in part by Disruptive Technology of the Ministry of Science and Technology (2020AAA0130000), the National Key Research and Development Program of China (2018YFB2100500), and Tianjin Research Program of Application Foundation and Advanced Technology (18JCY-BJC19200).

[†] Equal contribution.

Kun Li and Jinsong Zhang are with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, and the Key Research Center for Surface Monitoring and Analysis of Cultural Relics (SMACR), State Administration of Cultural Heritage, China.

Yebin Liu and Qionghai Dai are with the Department of Automation, Tsinghua University, Beijing 10084, China.

Yu-Kun Lai is with the School of Computer Science and Informatics, Cardiff University, Cardiff CF24 3AA, United Kingdom.

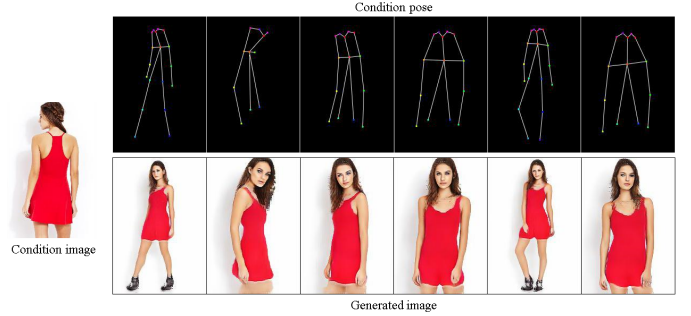


Figure 1: Generated examples by our method based on different target poses.

kind of methods [7] break up the whole body into parts (each body part as a rectangular region), transfer each body part respectively by computing a set of affine transformations, and finally combine the information of each body part to deliver the final results. However, the rectangular regions of image features are not precise for complex background and large pattern, which sometimes causes implausible images to be generated. The third kind of methods [8] propose to guide the pixel-to-pixel transfer and texture transfer by 3D prior knowledge, which can generate promising results. But their results depend on the accuracy of the appearance flow and their strategies require high computational cost and complicated training procedure.

The aforementioned existing methods fail to synthesize photo-realistic images due to the challenges of human pose transfer. One challenge is how to transfer the information of the condition image, including the style of clothes and appearance of the person, from the condition pose to the target pose. By comparing the image information to a student and pose information to a teacher, the student should study under the guidance of the teacher, while the teacher also needs to change teaching methods according to the aptitude and feedback of the student. On the one hand, under the guidance of the teacher, the student will become stronger and stronger. On the other hand, with the feedback from the student, the teacher will adjust the way of guidance according to the state of student to better guide the progress of the student. In fact, this is a chicken and egg problem between pose features and image features: good pose features will help to generate good image information while good image features will contribute to extracting relevant and important pose features. In previous network architectures, they ignore the guidance function between image features and pose features or use simple attention mechanism to deal with this

chicken and egg problem, which is difficult to make the utmost of image and pose features. Besides, human pose transfer is further compounded by self-occlusion and high variance in poses, which induce ambiguities in inferring unobserved pixels. Some methods deploy deeper networks or use 3D prior knowledge to cope with this challenge. However, their strategies require large computation budget and generate blurry images especially when there is a lot of regions to be inferred due to significant difference between poses.

The insight that takes human pose transfer as a chicken and egg problem motivates us to design a cross-modal block, named as **P**ose-guided **N**on-local **A**ttention (PoNA) block, with pre-posed image-guided pose feature update and post-posed pose-guided image feature update to better deal with the chicken and egg problem. With simplified cascaded cross-modal blocks, the model contributes to gradually transferring image features from the condition pose to the target pose. In the pre-posed image-guided pose feature update, we use self-attention module to merge pose feature and image feature. In the post-posed pose-guided image feature update, we propose a pose-guided non-local attention mechanism to alleviate ambiguities in inferring unobserved pixels, which also helps to reduce the required number of blocks. With our non-local attention mechanism, more important regions of image features can be selected and deformed, which is useful for inferring unobserved pixels and transferring image features from the condition pose to the target pose. Experimental results demonstrate that our method achieves more photo-realistic human pose transfer results with fewer parameters and faster speed, compared with five state-of-the-art methods. Some examples generated by our method are shown in Fig. 1. The code is available for research purposes at <https://github.com/Zhangjinso/PoNA>.

Our main contributions are summarized as follows:

- We propose a simple yet effective generator with simplified cascaded blocks for human pose transfer, which is easy to train with fewer parameters. We will make the code publicly available online.
- We propose a cross-modal block with pre-posed image-guided pose feature update and post-posed pose-guided image feature update, to better deal with the chicken and egg problem. This copes well with high variance between the source image and target image, because the pre-posed image-to-pose transfer gives a better initialization for image-based transfer, which is similar to the effect of rigging for model-based animation.
- We propose a pose-guided non-local attention mechanism in the image feature update to help select and deform important regions of image features, which deals well with the information missing and self-occlusion problems.
- We demonstrate the advantage of our method over the state-of-the-arts by quantitative and qualitative evaluation, and show the capability to alleviate data insufficiency for person re-identification.

The rest of this paper is organized as follows. Section II presents a brief review of related work. Section III describes the proposed network with pose-guided non-local attention

blocks. Experimental results are presented in Section IV, and the paper is concluded in Section V.

II. RELATED WORK

A. Generative Adversarial Networks (GANs)

The Generative adversarial network (GAN) [10] is composed of a generator and a discriminator where the discriminator tries to identify real images and synthesized images while the generator tries to fool the discriminator by generating realistic images. GANs usually generate realistic images through training in an adversarial way [10]–[12]. Conditional generative adversarial networks (CGANs) [13], which have achieved remarkable success, are proposed for the purpose of synthesizing realistic images with some condition constraints, *e.g.*, generating images at new viewpoints. Besides, as a commonly used structure in generators, U-Net [14] captures the input information of encoder and conveys it to decoder using skip connection, which is suitable for pixel-wise aligned tasks. However, human pose transfer is an unaligned task due to the deformation between the condition pose and target pose. Self-attention GAN [15] applies long-range dependency instead of local spatial dependency, solving the limitation of convolution operator. It calculates the attention map through transforming image features into two feature spaces to gain global information, and then acts on image features to get new image features with long-range dependency. However, self-attention mechanism can only gain the long-range dependency of itself but can not guide the transformation of other features, *i.e.*, guiding the transformation of image features using pose features. In this paper, we propose an improved attention mechanism for human pose transfer, which helps to deal with missing information and self-occlusions.

B. Person Image Generation

Lassner *et al.* [16] combine variational auto-encoder [17] and GAN to generate random person images with different appearance for the full body. Xian *et al.* [18] present a model for controlling the texture of synthesized image with input sketch and texture patches. Zhu *et al.* [19] propose a novel pipeline to synthesize novel views of human body from a monocular image. Balakrishnan *et al.* [20] decompose the person image generation task into multiple foregrounds with different body parts and background generation. Si *et al.* [21] adopt multi-stage adversarial losses for pose transformer network, foreground transformer network and background transformer network to generate more realistic images. Several methods [22]–[24] focus on the virtual try-on application and make remarkable progress in transferring clothes of a given person image while containing the condition pose and shape of that person. Previous existing researches ignore the chicken and egg problem in conditional generation. In this paper, we propose a cross-modal block with pre-posed image-guided pose feature update and post-posed pose-guided image feature update to cope with the chicken and egg problem in human pose transfer.

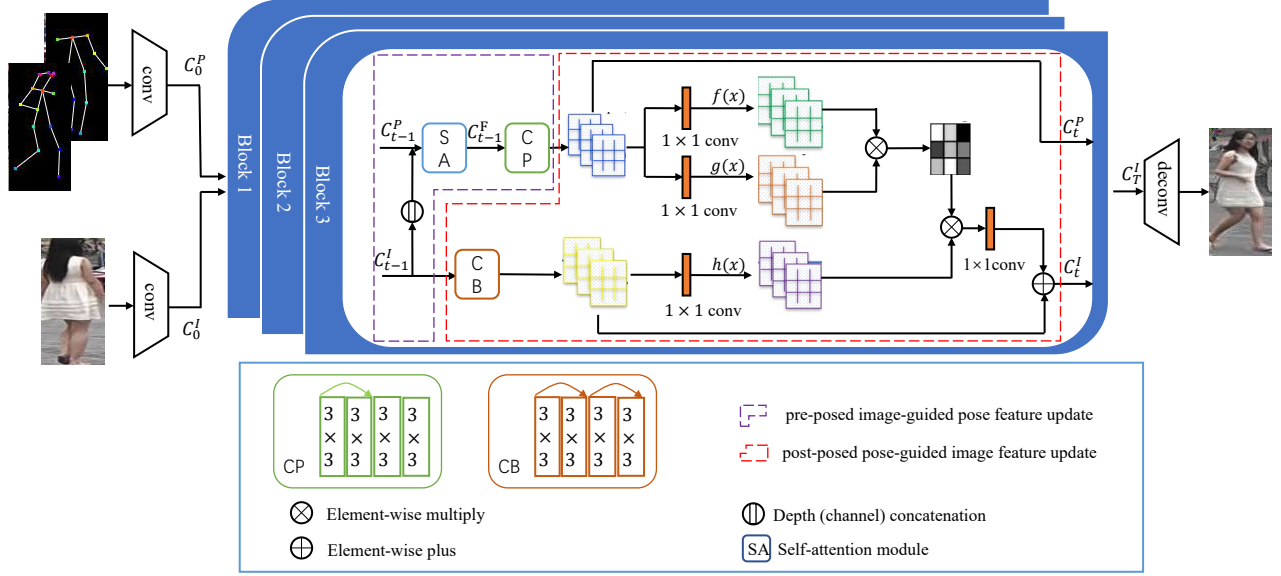


Figure 2: Structure of our proposed generator.

C. Human Pose Transfer

Human pose transfer is an important part of person image generation. Ma *et al.* [5] propose a coarse-to-fine framework, which first generates a coarse image and then refines it. However, this two-stage model is inefficient in computation and complex to train. Ma *et al.* [6] improve their previous work using a decomposition strategy. Esser *et al.* [4] combine a variational auto-encoder to sample appearance and a U-Net [14] to preserve shape information, modeling the interplay of shape and appearance. Neverova *et al.* [25] propose to form a warping module to preserve texture and a prediction module to generate plausible images and then use a blending module to deliver final results. Their work exploits a dense pose estimation system, which maps body pixels to UV surface coordinates, to generate pose presentation. Li *et al.* [8] use a 3D flow map and a visibility map from the condition pose and target pose to guide the transformation of image features and pixels. However, they need a flow regression module pre-trained by the dataset obtained through fitting a SMPL [26], a 3D human model with 6890 vertices and 13766 faces. Liu *et al.* [27] use a 3D body mesh recovery module [28] to disentangle the pose and shape, and propose a novel block to propagate the source information. However, the 3D body mesh recovery method cannot estimate pose and shape precisely, which leads to some blurry results. Zhu *et al.* [9] propose a progressive pose attention transfer network to utilize pose features to guide the image features transfer. They consider the guidance role of pose features, but local attention mechanism cannot capture long-range dependency to transfer the precise regions of the image features. In this paper, we cascade several pose-guided non-local attention blocks for better pose transfer using fewer parameters.

III. METHOD

A. Notations

Given a person image, we aim at generating an image for the person in another pose. To transfer pose arbitrarily, we adopt the commonly used pose representations to guide the transfer. Specifically, we use 18 human keypoints extracted by Human Pose Estimator (HPE) [29] and represented by heatmaps. The heatmap includes 18 channels, and each channel encodes the location of each joint of human body. Denote P_c and P_t as the heatmaps of the conditional pose and the target pose, I_c and I_t as the condition image and the target image.

Instead of directly using the keypoints K_c and K_t that are extracted from the condition image I_c and the target image I_t using HPE, we encode the pose as 18 heatmaps to provide widespread information about each joint location. Specifically, the condition pose P_c and the target pose P_t are represented as two tensors $P_c = \mathcal{P}(K_c)$ and $P_t = \mathcal{P}(K_t)$, where the slice \mathcal{P}_i ($1 \leq i \leq 18$) is a 2D matrix with the same dimension as I_c and I_t . Mathematically,

$$\mathcal{P}_i = \exp - \frac{(k - k_i)^2}{\sigma^2}, \quad (1)$$

where k_i is the i -th joint location and $\sigma = 6$ pixels, which is the same as that in Deform [7] and PATN [9]. During training, the model takes a pair of the condition and the target images (I_c, I_p) and a pair of the condition and the target pose (P_c, P_t) as inputs. The generator is fed with a triplet (I_c, P_c, P_t) and outputs I_g as close as possible to I_t .

B. Encoders

Figure 2 shows the architecture of our generator. The generator has two encoders, several pose-guided non-local attention blocks and one decoder. The input of the generator is (I_c, P_c, P_t), a triplet of the condition image, condition pose

and target pose. The output is our generated image I_g , which is expected to be similar to the target image I_t . The task is transferring the information in condition image I_c , including texture, body shape, clothes style, *etc.*, from condition pose P_c to target pose P_t . The encoders, pose encoder and appearance encoder, have the same structure with two down-sampling convolution layers. We concatenate condition pose and target pose as the input of the pose encoder. The pose encoder encodes the transformation between condition pose and target pose and preserves the information of both. The appearance encoder takes a condition image as input and encodes the information of the condition image. After going through two encoders, pose code C_0^P and image code C_0^I are obtained.

C. Pose-guided Non-local Attention Block

We propose pose-guided non-local attention (PoNA) blocks, which are cross-modal blocks, to make pose features truly guide the transformation of image features. With several PoNA blocks, image features can be transferred by pose features from the condition pose to the target pose gradually. Each PoNA block is separate and has the same structure. The inputs of the t^{st} PoNA block are image code C_{t-1}^I and pose code C_{t-1}^P from the $(t-1)^{st}$ PoNA block, and the outputs are image code C_t^I and pose code C_t^P . Several PoNA blocks finally output the final image code C_T^I which is put into the decoder to generate the final image, while the final pose code C_T^P is discarded.

As explained before, PoNA block is used to make pose features guide the transformation of image features. In each PoNA block, two pathways, image pathway and pose pathway, are designed for pose code and image code, respectively. Image pathway and pose pathway are connected with our improved non-local attention mechanism, which can accomplish the task of pose-guided transformation. As shown in Figure 2, image code C_{t-1}^I and pose code C_{t-1}^P from the $t-1^{st}$ PoNA block are first concatenated along the depth (channel) axis to get the fusion code. Pose code is updated with the fusion code as C_t^P , which contains both image features and pose features, and then attention map is computed using the updated pose code. Besides, image code is updated by going through four convolution layers. To get the final image code C_t^I , the attention map is modulated on the updated image features, which accomplishes the transformation of image features guided by pose features. Details about PoNA block is described in the following.

Image-guided Pose Code Update. Before going through image pathway and pose pathway, we concatenate the image code and the pose code along the depth (channel) dimension as fusion features to make pose features know the information about transformation of image features. With fusion code, we deploy a self-attention module [15] to better integrate fusion features and select more important regions for the guidance. Mathematically:

$$C_{t-1}^F = self_attention(C_{t-1}^P || C_{t-1}^I). \quad (2)$$

where $||$ is cited as concatenation along the depth (channel) dimension.

For pose pathway, a block with four convolution layers is used to encode the information of fusion features and prepare

for the following guidance. The four convolution layers (each layer with a normalization layer [30], [31] and a ReLU [32]) help the pose code know about the transformation of image features, which benefits for the following guidance. These four convolution layers are capable of extracting useful features from the fusion code. One of these layers also reduces the number of channels to half, making the size of the output equal to the input. The pose code is updated by:

$$C_t^P = conv_P(C_{t-1}^F). \quad (3)$$

Pose-guided Non-local Attention. The pose transfer is to move patches from the condition pose to the target pose and to deal with the relationship between different patches. From this point of view, the pose guides the transfer by finding where to extract condition patches and where to put target patches and meanwhile maintaining the relationship between patches. In our PoNA block, such transformation is realized by the attention map denoted as α_t , which are values between 0 and 1 calculated by softmax, indicating the importance of every element in the updated pose code and the extent to which the model attends to one location when synthesizing other locations.

Traditional non-local attention mechanism embeds query, key and value from the same feature, calculates the attention map by computing the similarity between key and query, and updates the value with this attention map to select more important regions. For the guided attention mechanism, key and value are embedded from the feature to be guided, and query is embedded from the feature as guidance. However, in our task, the pose feature and the image feature are in the different latent spaces, and it is hard to compute similarity between image feature and pose feature. To get a reliable attention map, we embed key and query from the updated pose feature. The map α_t is calculated from the pose code C_{t-1}^P . The pose code C_{t-1}^P is transformed into two feature spaces, key and query space, by two different 1×1 convolutional layers, denoted as mappings f and g , where $f(x) = W_f x$ and $g(x) = W_g x$. Mathematically:

$$\alpha_{t,j,i} = softmax(m_{t,i,j}), \quad (4)$$

where $\alpha_{t,j,i}$ indicates the relationship between the i^{th} location and the j^{th} location in the t^{th} PoNA block, and $m_{t,i,j}$ is calculated as

$$m_{t,i,j} = f(C_{t-1}^P)^T \cdot g(C_{t-1}^P). \quad (5)$$

Pose-guided Image Code Update. Image code is updated by going through four convolution layers and embedded into value space by a 1×1 convolutional layer. The attention map α_t from the pose-guided non-local attention mechanism can select important regions from the image code and deform the original image code by rearranging the image feature. The output of the attention layer is $o_t = (o_{t,1}, o_{t,2}, \dots, o_{t,j}, \dots, o_{t,N})$, where

$$o_t = \sum_{i=1}^N \alpha_{t,j,i} W_t(conv_I(C_{t-1}^I)_i), \quad (6)$$

$$h(x_i) = W_h x_i.$$

In the above formulation, N is the number of locations of the features from the previous hidden layer, and W_h and W_t are the learned weight matrices, which are implemented as 1×1 convolution layers.

In addition, we combine the output of the attention layer with the input image code by a learnable parameter γ which is initialized as 0. Therefore, the final output is given by

$$C_t^I = \gamma o_t + \text{conv}_I(C_{t-1}^I). \quad (7)$$

The learnable coefficient γ enables the block first to rely on local features and then gradually to learn to combine the non-local evidences.

D. Discriminator

We adopt two discriminators, appearance discriminator D_A and pose discriminator D_P , to judge how likely I_g contains the same person as I_c (appearance consistency) and how well I_g aligns with the target pose P_t (pose consistency). The two discriminators are similar in structures, where the inputs of them are I_g concatenated with either condition image I_c or target pose P_t along the depth axis. These inputs go through a convolutional layer (with normalization and ReLU after it) and several residual blocks (with self-attention module) after down-sampling. The outputs of the discriminators are the appearance consistency score S^A and the pose consistency score S^P , calculated by softmax layers.

E. Loss Function

Previous methods on human pose transfer [4]–[9] utilize multiple loss functions to supervise the training process. In this work, we adopt a combination of three loss functions as the full loss function \mathcal{L}_{full} , including a conditional adversarial loss \mathcal{L}_{CGAN} , an L1 loss \mathcal{L}_{L1} and a perceptual loss \mathcal{L}_{percep} , which are described in details as follows.

The full loss function is defined as

$$\mathcal{L}_{full} = \lambda_1 \mathcal{L}_{CGAN} + \lambda_2 \mathcal{L}_{L1} + \lambda_3 \mathcal{L}_{percep}, \quad (8)$$

where λ_1 , λ_2 , λ_3 represent the weights of \mathcal{L}_{CGAN} , \mathcal{L}_{L1} , \mathcal{L}_{percep} that contribute to \mathcal{L}_{full} , respectively.

Conditional adversarial loss. The conditional adversarial loss is defined as

$$\begin{aligned} \mathcal{L}_{CGAN} = & E_{P_t \in \mathcal{P}, (I_c, I_t) \in \mathcal{I}} \{\log[D_A(I_c, I_t) \cdot D_P(I_t, P_t)]\} + \\ & E_{P_t \in \mathcal{P}, I_c \in \mathcal{I}, I_g \in \hat{\mathcal{I}}} \{\log[(1 - D_A(I_c, I_g)) \cdot (1 - D_P(I_g, P_t))]\}, \end{aligned} \quad (9)$$

where \mathcal{P} , \mathcal{I} and $\hat{\mathcal{I}}$ denote the distributions of person poses, real person images and fake person images, respectively.

L1 loss. \mathcal{L}_{L1} loss represents the pixel-wise differences between the generated image I_g and the ground truth I_t , which is defined as

$$\mathcal{L}_{L1} = \|I_g - I_t\|_1. \quad (10)$$

Perceptual loss. Previous work [4]–[9] shows that L1-distance between feature maps extracted from two images by a pre-trained CNN could make the generated images look more natural and reduce the structural differences, which performs

well in style transfer [11] and pose transfer [5]–[8]. The perceptual loss is defined as

$$\mathcal{L}_{percep} = \sum_{i=1}^{C_\rho} \|\phi_\rho(I_g)_i - \phi_\rho(I_t)_i\|_1, \quad (11)$$

where ϕ_ρ denotes the outputs of the ρ^{th} layer from the VGG-19 model [33] pre-trained on ImageNet [34], and $\phi_\rho(\cdot)_i$ denotes the i^{th} feature map of the outputs of ϕ_ρ .

In practice, we find that it is good enough to use the features from *Conv1_2* of VGG-19.

F. Training Procedures

We train the generator and the discriminators alternately. When training, the input of the generator is a triplet (I_c, P_c, P_t) , and the output is a generated image I_g which has the same pose as the target image I_t . Specifically, I_c is fed to the image stream and (P_c, P_t) are fed to the pose stream. To train the discriminators, the appearance discriminator D_A takes (I_c, I_t) and (I_c, I_g) as inputs to calculate the appearance consistency score S^A , and the pose discriminator D_P takes (P_t, I_t) and (P_t, I_g) as inputs to calculate the pose consistency score S^P .

G. Implementation Details

We use 3 PoNA blocks in the generator. For images from Market-1501 dataset that are low resolution, we apply batch normalization [30] in the generator for all the normalization layers. Instance normalization [31] which is a better choice for the transfer task, is applied for DeepFashion dataset. The coefficients in the loss function $(\lambda_1, \lambda_2, \lambda_3)$ are set to be (5,10,10) for Market-1501 dataset and (5,1,1) for DeepFashion dataset. Leaky-ReLU [32] is adopted after each convolution layer or normalization layer in the discriminators, and its negative slope coefficient is set to be 0.2. For all models, we adopt the Adam optimizer [35] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ to train for 90k iterations.

IV. EXPERIMENTAL RESULTS

In this section, we first introduce the datasets and the metrics in Section IV-A, and compare our method with the state-of-the-art methods in Section IV-B, and then we perform an ablation evaluation to study the importance of the different components of our approach in Section IV-C. We show the application on data augmentation for person re-identification which helps to improve the performance in Section IV-D and finally analyze the limitations of our model in Section IV-E.

A. Datasets and Metrics

1) Datasets: Person re-identification dataset Market-1501 [36] and the In-shop Clothes Retrieval Benchmark DeepFashion [37] are commonly used datasets for the evaluation of human pose transfer. Images in Market-1501 are low resolution (128×64) with large variation in pose and background, which makes human pose transfer more challenging. Compared with Market-1501, the images in DeepFashion are higher resolution (256×176) with clean background. Zhu *et al.* [9] use HPE [38]

Table I: Quantitative comparison with five state-of-the-art methods on Market-1501 and DeepFashion.

Model	Market-1501						DeepFashion			
	SSIM	IS	mask-SSIM	mask-IS	PSNR	PCKh	SSIM	IS	PSNR	PCKh
PG ² [5]	0.261	3.495	0.782	3.367	28.135	0.73	0.773	3.163	30.785	0.89
VUnet [4]	0.266	2.965	0.793	3.549	28.071	0.92	0.763	3.440	30.856	0.93
Deform [7]	0.291	3.230	0.807	3.502	28.227	0.94	0.760	3.362	31.022	0.94
PATN [9]	0.311	3.323	0.811	3.773	28.228	0.94	0.773	3.209	31.160	0.96
LWG [27]	-	-	-	-	-	-	0.696	3.478	28.396	0.80
Ours (3 PoNA blocks)	0.315	3.487	0.814	3.867	28.257	0.94	0.775	3.338	31.382	0.95
Real Data	1.000	3.890	1.000	3.706	∞	1.00	1.000	4.053	∞	1.00

as pose joints detector and collect 263632 training pairs and 12000 testing pairs for Market-1501 and 101966 training pairs and 8570 testing pairs for DeepFashion, when proposing their PATN method. In both of these datasets, the person identities in the test set are different from those in the training set. In order to ensure the fairness of the comparison results, we adopt the training pairs and the testing pairs used in PATN [9] for both datasets.

2) *Metrics*: Ma *et al.* [5] use Structural SIMilarity (SSIM) and Inception Score (IS) as their metrics. SSIM can measure the similarity between synthesized images and ground-truth images, and IS is a common method used to measure the quality of image generation. For Market-1501 dataset, Ma *et al.* [6] further introduce the mask version of SSIM (mask-SSIM) and the mask version of IS (mask-IS) by masking the background out, which reduces the influence of blurry background of images. Besides, Zhu *et al.* [9] introduce the slightly modified version of Percentage of Correct Keypoints (PCKh) [39], which measures shape consistency by computing pose joints alignment using pre-trained Human Pose Estimator (HPE) [29]. Moreover, we use Peak Signal to Noise Ratio (PSNR) to measure the difference between the synthesized image and the ground-truth image in pixel level.

B. Comparison with Previous Work

1) *Qualitative Evaluation*: We evaluate the visual results of our method on Market-1501 and DeepFashion datasets, compared with five state-of-the-art methods: PG² [5], VUnet [4], Deform [7], PATN [9], and LWG [27]. Because LWG [27] does not provide pre-trained model on the Market-1501 dataset, we compare with LWG [27] only on the DeepFashion dataset.

For poor quality images in Market-1501, as shown in Figure 3, our method generates clearer images than the other methods. It is worth noting that our method gives the correct leg layouts even when the legs are crossed in the target pose (in the first and third rows). Besides, even if the condition image is blurred (in the second row) or has complex clothing patterns (in the third row), our method can learn the style of the garments and maintain these features in the generated images. Moreover, our method also keeps appearance consistency, *e.g.*, the bag is preserved in our results (in the first and third rows). For high quality images in DeepFashion, as shown in Figure 4, our method yields the sharpest person images with better

facial details while the generated images of the other methods have some blur contents (in the third and fifth rows). Besides, the texture (in the first and third rows) and clothing styles (in the first, second and fifth rows) in condition images are preserved in our generated images, which indicates that our model has the power of capturing global styles and local details for generation. Although LWG preserves fine details in the condition image (in the fifth row), it fails to generate sharp and plausible images with precise pose. Moreover, our method keeps appearance consistency, *e.g.*, the hat in the third row.

To further validate the performance of our model, we try to generate person images based on the same condition image and several different poses from the testing sets of Market-1501 and DeepFashion, respectively. Some examples are shown in Figure 5-7. We only compare with PATN [9] and LWG [27] since they are the latest state-of-the-art methods and the space is limited. LWG [27] cannot generate good results on the Market-1501 dataset, and hence we do not compare this method on that dataset. Because LWG [27] fails to generate sharp image when the condition image and reference image have different camera coordinates, we select some full-body images that have similar camera coordinates as condition images and reference images from DeepFashion dataset. As shown in Figure 5 and Figure 6, for images in Market-1501 dataset, our model can generate more realistic and plausible images even for poses with large variation. Moreover, our generated images keep shape consistency while the results of PATN [9] lose some details and are blurry. Compared with PATN [9] on DeepFashion dataset, our model generates sharper images with better facial details and less noise. As shown in Figure 7, the results of LWG are blurry and sensitive to different camera coordinates, although LWG can preserve some details. Besides, due to its dependency on 3D mesh recovery module to get pose and shape information, the persons in the generated images from different condition images have different shapes and inaccurate pose when the results of 3D mesh recovery module are not precise. As shown in the third and last columns of the second example, LWG cannot cope well with large differences of camera coordinates between the condition image and the estimated image, which explains why the quantitative results of LWG are not good (see Table I). Besides, we compare with five state-of-the-art methods on DeepFashion dataset in Figure 8

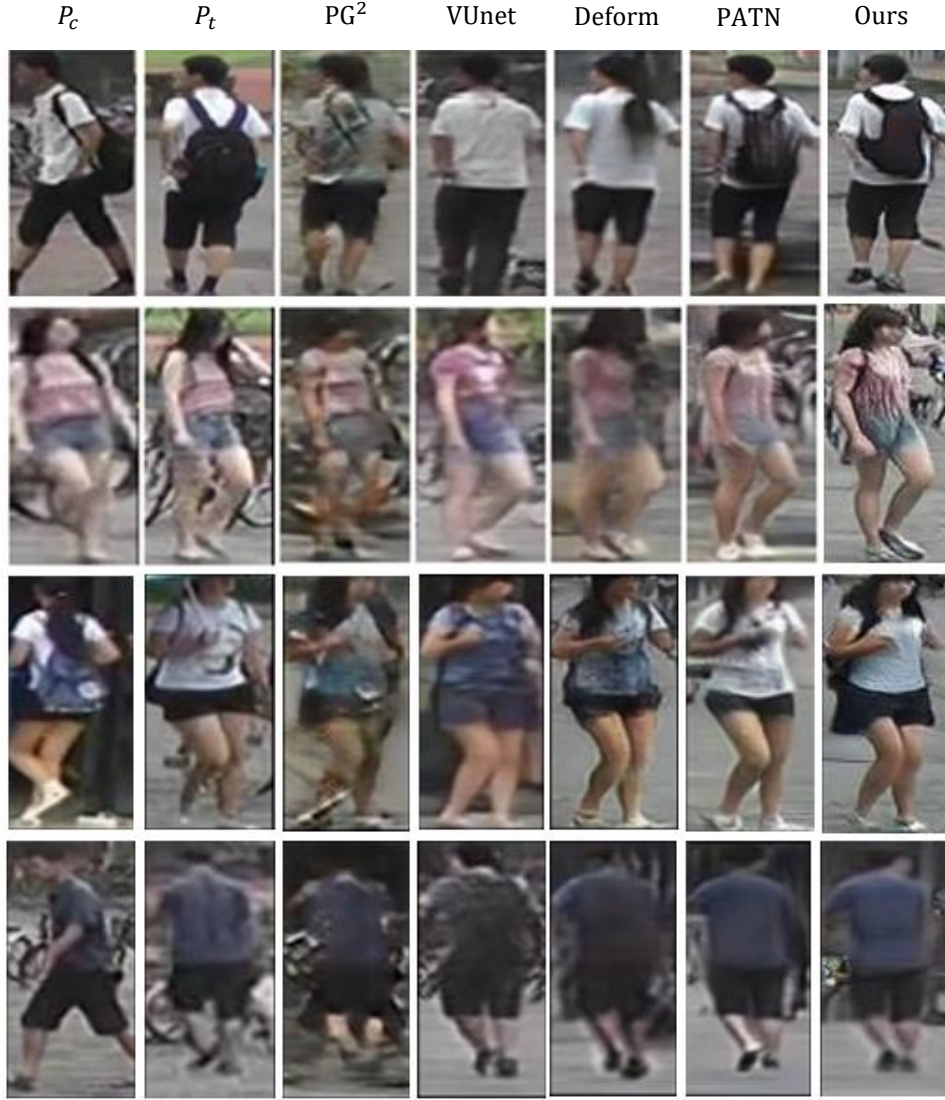


Figure 3: Qualitative comparisons on Market-1501 dataset. From left to right are the results of PG^2 [5], VUnet [4], Deform [7], PATN [9] and ours, respectively.

to show the performance conditioned by different poses for the same person. Our model generates more realistic images and preserves the pattern of clothes. Moreover, the unseen regions synthesized by our model are more reasonable with less artifacts.

2) *Quantitative Evaluation*: Table I gives quantitative results compared with five state-of-the-art methods: PG^2 [5], VUnet [4], Deform [7], PATN [9] and LWG [27]. We use the same training set and testing set used in PATN. Because other methods do not give their data split scheme, we run their well-trained models on our testing set. It is inevitable to have some test images in their training sets. Even in this case, our method still outperforms them on most metrics. For Market-1501, although our IS metric is slightly lower than Ma *et al.* [5], our mask-IS metric, which is more convincing for Market-1501, is the highest score. The quantitative results on Market-1501 demonstrate that the images generated by our method maintain structure similarity and shape consistency even if

the condition images are low resolution and vary significantly in the pose and background. For DeepFashion, our method has the best results in terms of SSIM, which means that our generated images keep structure similarity to the ground truth. Our PSNR value is the highest on both Market-1501 and DeepFashion datasets, which demonstrates that our generated images have the minimum pixel-level errors.

3) *Efficiency Evaluation*: Table II shows our computation complexity and efficiency compared with five state-of-the-art methods: PG^2 [5], VUnet [4], Deform [7], PATN [9], and LWG [27]. We test all the methods on the same desktop with one NVIDIA Titan Xp graphics card. We discard the time of data preparation, and compute the testing time on DeepFashion dataset. Through the comparison of model size and speed, our model significantly outperforms the other five state-of-the-art methods. Compared with PATN [9], our method has fewer parameters and faster running speed, but get better results, especially for our model with 2 PoNA blocks. Note that the

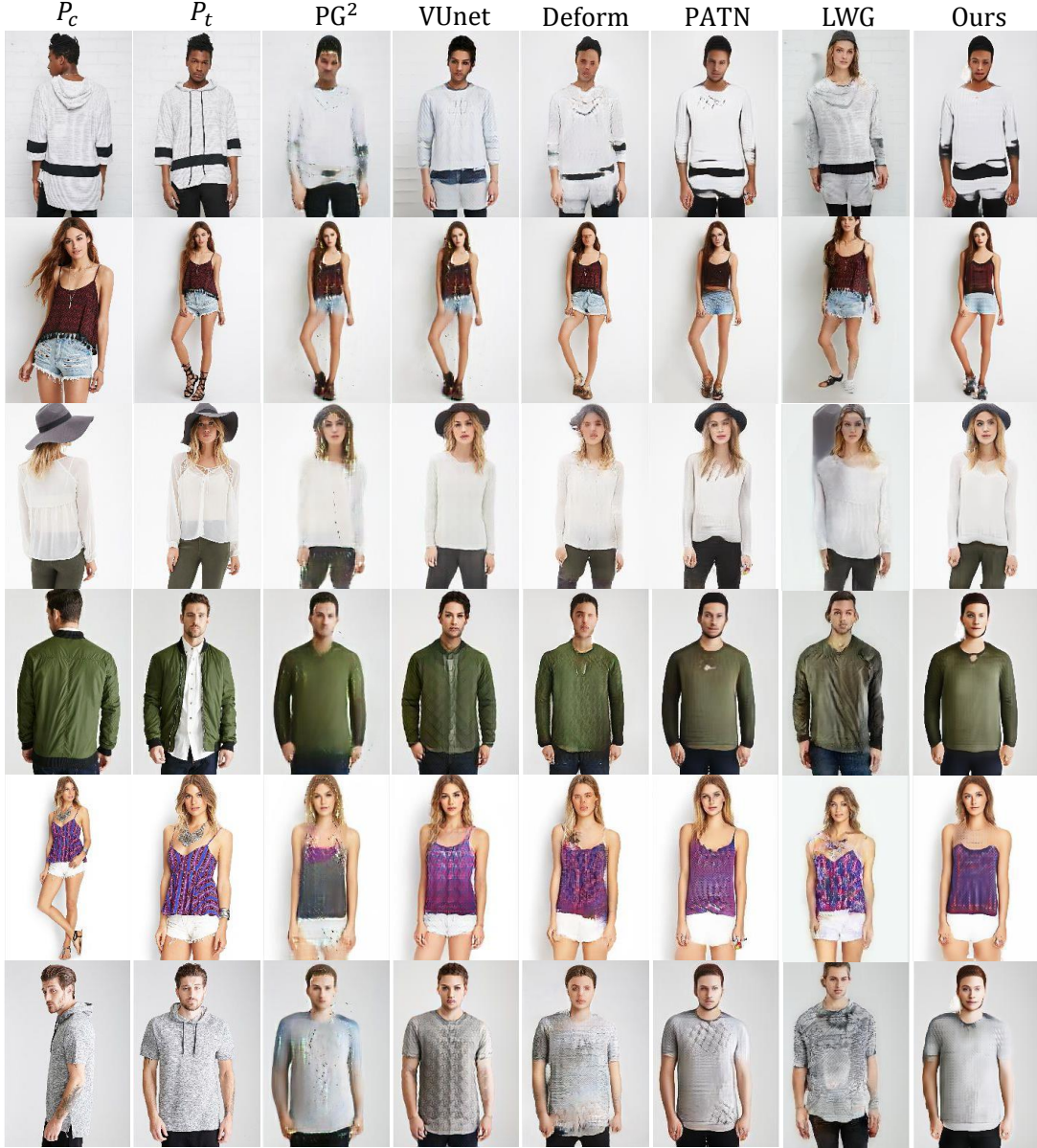


Figure 4: Qualitative comparisons on DeepFashion dataset. From left to right are the results of PG^2 [5], VUnet [4], Deform [7], PATN [9], LWG [27] and ours, respectively.

Table II: Comparison of model size and speed on DeepFashion dataset.

Method	Parameters	Speed
PG^2 [5]	437.09 M	16.36 fps
VUnet [7]	139.36 M	44.27 fps
Deform [4]	82.08 M	25.49 fps
PATN [9]	41.36 M	97.37 fps
LWG [27]	97.45 M	7.31 fps
Ours with 3 PoNA blocks	34.33 M	105.18 fps
Ours with 2 PoNA blocks	23.31 M	179.32 fps

results of our model with 2 PoNA blocks are slightly lower than PATN (with 9 blocks) in SSIM and mask-IS metrics, but higher than their model in IS and mask-SSIM metrics, as shown in Table I and Table III. In a word, our model has the fewest parameters and the fastest speed, even when using 3 PoNA blocks.

C. Ablation Study

1) *Effectiveness of Cascaded PoNA Blocks*: The generator we proposed has several cascaded PoNA blocks with improved non-local attention mechanism, to deal with the challenging cases of human pose transfer. In order to demonstrate the effectiveness of cascaded PoNA blocks, we test different numbers of PoNA blocks on Market-1501 dataset. Table II



Figure 5: Image generation results conditioned by different poses on the Market-1501 dataset. For each condition image, the first row shows the images generated by PATN [9], and the second row shows our results.

and Figure 9 show the quantitative and qualitative results using different numbers of PoNA blocks, respectively. When we only use 1 PoNA block in our model, the quantitative results decrease in all metrics except mask-IS, and slightly lower than the other state-of-the-art methods (please refer to Table I). This verifies the effectiveness of our pose-guided non-local attention mechanism. As shown in Figure 9, even using 1 PoNA block, our model can still obtain reasonable results. For simple clothing patterns, *e.g.*, uniform color skirt (in the third row) and knapsack (in the fourth row), 1 PoNA block is enough. Using 2 or more PoNA blocks, our model can capture fine details of complex appearance and generate plausible images, especially for 3 PoNA blocks. With the increase of PoNA blocks, some artifacts may appear (in the second and third rows). Therefore, in our experiments, we use 3 PoNA blocks to generate plausible and robust results. The images in the last column of Figure 9 are generated by PATN [9] with 9 blocks, which has the best performance in their paper. Compared with PATN [9], our PoNA blocks can capture more details and generate more realistic images. With non-local attention mechanism, our network can transfer the details in the condition image, even for the images with complex clothing patterns, and hence our network generates sharper and more realistic images.

2) Effectiveness of different components of PoNA Block:

The PoNA block is composed of self-attention module and cross-modal attention module. To validate the effectiveness of these two components, we conduct experiments on Market-1501 dataset by training the model without cross-modal module and the model without self-attention module. Table IV and Figure 10 show the quantitative and qualitative results. It can be seen that the cross-modal attention module plays a key role in improving the performance. The cross-modal attention

Table III: Results of different numbers of PoNA blocks.

Number	SSIM	IS	mask-SSIM	mask-IS	PCKh
1	0.298	3.367	0.806	3.806	0.91
2	0.309	3.419	0.812	3.745	0.94
3	0.315	3.487	0.814	3.867	0.94
4	0.309	3.481	0.809	3.783	0.92
5	0.306	3.449	0.808	3.872	0.92

module is designed for post-posed pose-guided image feature update, and it is the only way that pose feature plays a role in guiding the transformation of image feature. Without the cross-modal attention module, our model cannot obtain pose information, which causes that the generated person does not have the similar pose to the ground truth. Besides, there are some different target poses and target images for the same source image as paired training data, which means that the same source image has different target images. Therefore, without pose information, the model may learn to synthesize images with mean shape and mean pose to minimize the loss. In our model, the self-attention module is used to merge pose feature and image feature to select more important features. Without self-attention module, the pre-posed image-guided pose feature update cannot select important features to embed key and value to obtain the attention map, and hence generates some wrong estimates in the final image. Our full model achieves the best performance.

3) *Place and Importance of Fusion Code:* This section will demonstrate our design of the cross-modal block with pre-posed image-guided pose feature update and post-posed pose-guided image feature update. The improved attention mechanism can make pose features guide the transfer of image

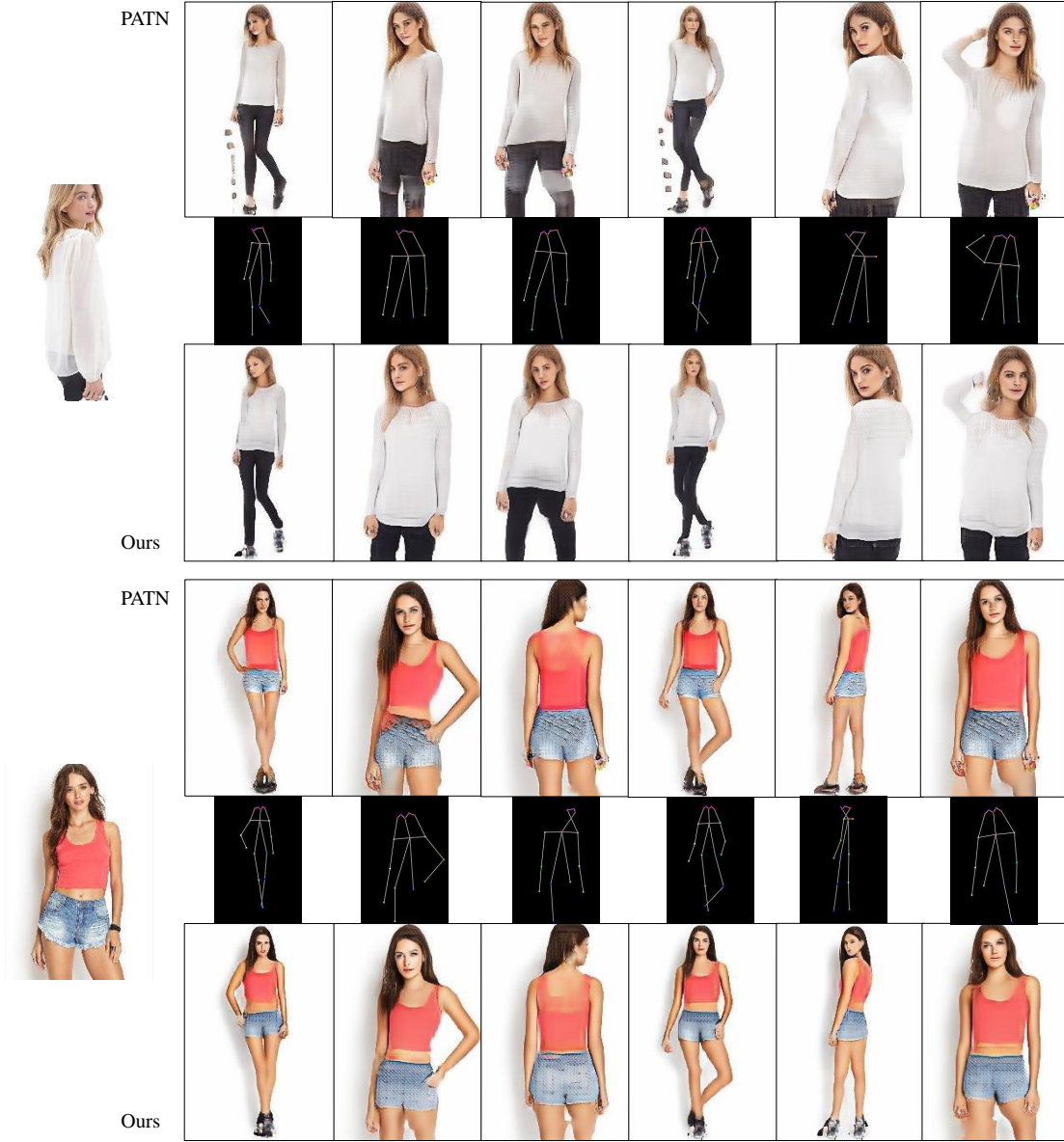


Figure 6: Image generation results conditioned by different poses on the DeepFashion dataset. For each condition image, the first row shows the images generated by PATN [9], and the second row shows our results.

Table IV: Quantitative results without different components of the PoNA block.

	SSIM	IS	mask-SSIM	mask-IS	PCKh
w/o Cro.	0.155	3.034	0.729	3.628	0.15
w/o Self.	0.307	3.331	0.809	3.755	0.93
Full model	0.315	3.487	0.814	3.867	0.94

features after getting fusion code and updating pose code, as illustrated in Section III. However, because concatenating pose code and image code is the only way to let pose features know the process of image feature transfer, when and where to concatenate image code and pose code to update fusion code is important for the follow-up operations. Images in Market-

1501 dataset are more challenging for human pose transfer and suitable for validating the place and importance of fusion code. We call the fusion code before and after pose code updating as the head and tail fusion, and deploy fusion code in the middle of four convolution layers in pose code updating as middle fusion. To validate the importance of fusion code, we also remove it from PoNA block as none fusion. We test our model with 3 PoNA blocks. Figure 11 shows the structures of head, middle and tail fusions.

Table V shows the quantitative results of different places of fusion code. It can be seen that the head fusion, *i.e.*, *pre-posed* image-guided pose feature update, has the best performance, because the head fusion can fuse pose features and image features before pose code updating and help to understand the structured information of image features. The tail fusion cannot maintain the structured information and understand



Figure 7: Image generation results conditioned by different poses on the DeepFashion dataset. For each condition image, the first row shows the images generated by LWG [27], and the second row shows our results.

Table V: Results of different places of fusion code.

Fusion place	SSIM	IS	mask-SSIM	mask-IS	PCKh
Head	0.315	3.487	0.814	3.867	0.94
Middle	0.310	3.501	0.811	3.897	0.93
Tail	0.307	3.680	0.807	3.884	0.93
None	0.297	3.427	0.804	3.705	0.91

the image features compared with head fusion and middle fusion. Even though we remove the fusion code, owing to our improved attention mechanism, our model can obtain promising results, which are higher or slightly lower than the other state-of-the-art methods illustrated in Table I in all metrics. This verifies the effectiveness of our *post-posed* pose-guided image feature update.

The qualitative results are shown in Figure 12. The model with head fusion generates more plausible results based on condition images (in the fourth and fifth rows), especially when the legs in the target image are crossed (in the second row). PATN [9] with 9 blocks (in the last column), with local attention mechanism, is not able to capture the details in the condition image, *e.g.*, the white hat (in the first row), the bag (in the third and fourth rows) and the white collar (in the

fifth row). Note that the images generated by our model are sharper and have rich details, even if we remove the fusion code updating. This further proves the effectiveness of our improved attention mechanism.

4) Visualization of Features in Our Model: In this section, we visualize all the core features of our model to get an intuitive understanding on the transformation of pose feature and image feature. Figure 13 shows the visualization of features in our model. In pre-posed image-guided pose feature update, the images represent the features before and after the encoder and the features before and after the pose code update in each PoNA block from top to down. It can be seen that the pose feature becomes denser, which is beneficial to obtain a reliable attention map to deform the image feature. In post-posed pose-guided image feature update, the first row gives the input image and the output feature of image encoder, and the middle three rows show the image features before and after deforming by cross-modal attention module. As shown in Figure 13, the cross-modal attention module deforms the image feature from condition pose to target pose gradually and focuses on different regions in each block.

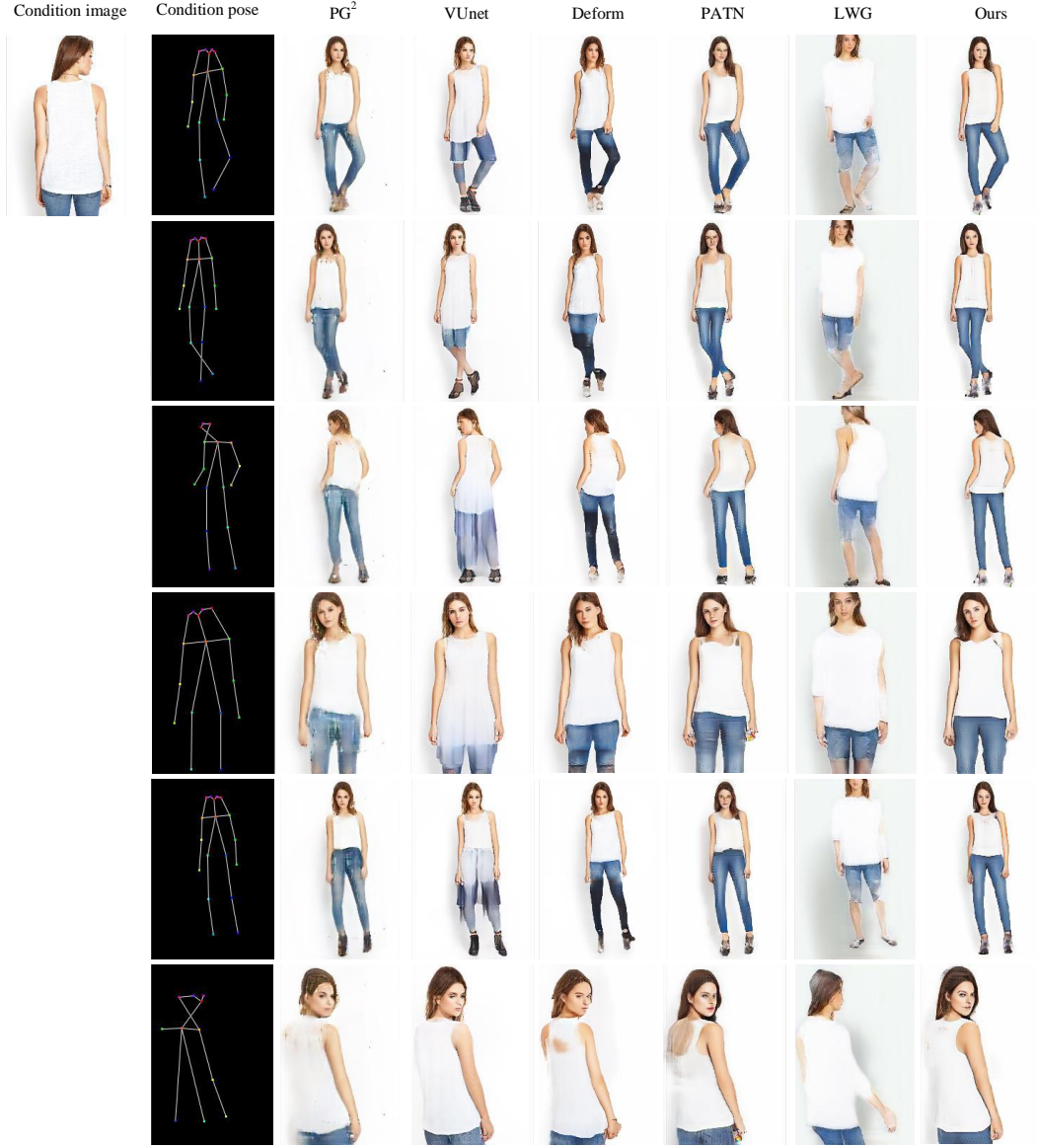


Figure 8: Image generation results conditioned by different poses on the DeepFashion dataset. From left to right are the results of PG² [5], VUnet [4], Deform [7], PATN [9], LWG [27] and ours, respectively.

D. Data Augmentation for Person Re-identification

Human pose transfer is able to generate images of the same person in different poses, which is useful to augment person re-identification (re-ID) [40] dataset to solve the problem of lacking training data and improve the performance of person re-ID. To some degree, the performance of augmenting dataset depends on the performance of human pose transfer model. In order to illustrate the performance of our model, we test on Market-1501 dataset [36], which is a main person re-ID dataset. We exploit our generated images to replace the images in Market-1501. Specifically, we first obtain a reduced training set by selecting a portion p to randomly preserve the images in the whole training set. Then, we use our human pose transfer method to generate missing images conditioned on preserved images and the pose of missing images. Finally, we combine

the reduced training set and the generated training set to obtain the new training set. Note that the images in the new training set has the same identities and each identity has the same number of images with the same pose as original images. We select the portion from 20% to 80% at intervals of 20%, and get four reduced training sets and four new augmented training sets. We use the Person re-ID baseline Framework [41] based on ResNet-50 [42] as our training and testing protocols.

The training sets we created are suitable for showing the performance of data augmentation using our human pose transfer method. First, the reduced training set provides an environment with insufficient data and helps us know the lower bound of performance. Second, the original training set with realistic images gives the upper bound of performance. With lower bound and upper bound, we can measure the performance of data augmentation using the human pose transfer method.



Figure 9: Results on Market-1501 dataset with different numbers of PoNA blocks.

Table VI shows the re-ID results using the reduced training sets (referred to as None) and the new training sets generated by VUnet [4], Deform [7], PATN [9], and our method. We did not compare PG² [5] and LWG [27] since they cannot generate good results on this Market-1501 dataset. For fair comparison, we use the same settings (*e.g.*, condition images and target poses) to generate the images for all the methods. We use mean Average Precision (mAP) as the metric to measure the re-ID performance. With the same model and the same

parameters, the re-ID performance relies on the photorealism of generated images and the texture consistency of the same identity. As shown in the table, the model augmented by our method achieves the most accurate re-ID estimation, which indicates that our method generates more realistic images with consistent textures.

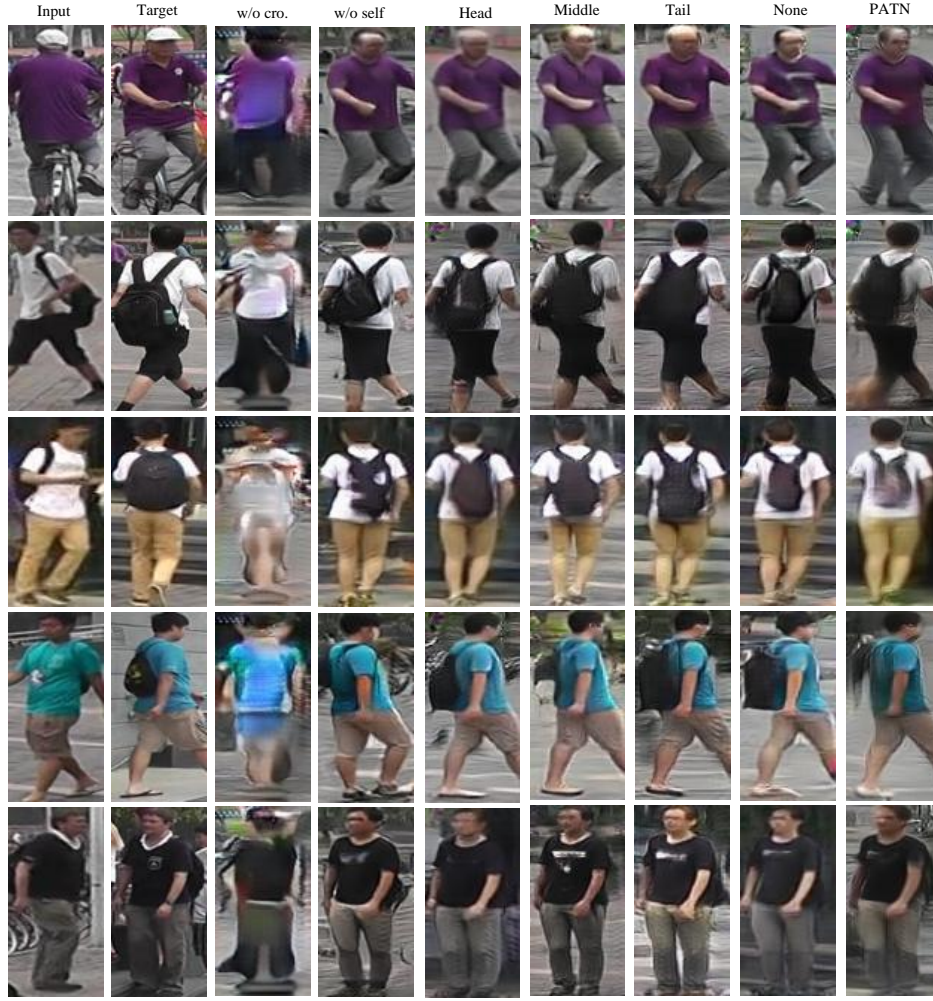


Figure 12: Generated images using different places of fusion code.

Table VI: Results of re-ID.

Aug. Model	Portion p of the original dataset				
	0.2	0.4	0.6	0.8	1.0
None	42.3	58.9	66.7	69.5	71.8
VUnet [4]	54.1	59.6	66.2	68.6	71.8
Deform [7]	55.8	60.7	67.4	69.4	71.8
PATN [9]	57.2	61.4	67.9	69.7	71.8
Ours	58.8	63.8	68.7	70.4	71.8

E. Limitations

Our model can deform the image feature from the condition pose to the target pose by pose-guided attention mechanism, which also alleviates the negative effects of occlusion. Although our model generates impressive results, the quality of generated images can be further improved, especially for occlusion cases. Figure 14 shows some failure cases using our method. Our model treats large areas of occlusion as part of the texture, resulting in blurry areas and incorrect textures (*e.g.*, dress and shorts). Pose-guided attention mechanism, as a non-

local attention mechanism, can select and deform important regions, but cannot cope well with the invisible areas in the condition image by the weighted sum. In future work, we will try to use human parsing map to extract semantic information to deal with occlusion and add local attention mechanism to enhance the quality of texture.

V. CONCLUSION

In this paper, we propose a pose-guided non-local attention (PoNA) block with an improved attention mechanism to deal with the challenging human pose transfer. With the improved attention mechanism, each block selects precise regions of image features to transfer based on pose features. The generator of our network is composed of several PoNA blocks and transfers image features progressively. Compared with previous work, our network generates more realistic and sharper images with rich details, and get the highest scores in the reasonable mask-SSIM and mask-IS metrics. At the same time, our network has fewer parameters and faster speed. Moreover, the proposed network can generate training images for person re-identification to alleviate the data



Figure 10: Generated images without different components of the PoNA block.

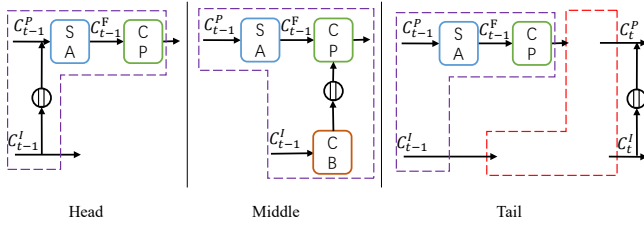


Figure 11: Structures of different places of fusion code. Note that the legend is as the same as Figure 2.

insufficiency. Our improved attention mechanism with pre-posed and post-posed fusion is suitable for other conditioned generation tasks. In the future, we will deploy the improved attention mechanism to other conditioned generation tasks, such as facial animation.

REFERENCES

- [1] Q. Zhou, H. Fan, H. Yang, H. Su, S. Zheng, S. Wu, and H. Ling, “Robust and efficient graph correspondence transfer for person re-identification,” *IEEE Trans. Image Processing*, 2019.
- [2] H. Yue, J. Yang, X. Sun, F. Wu, and C. Hou, “Contrast enhancement based on intrinsic image decomposition,” *IEEE Trans. Image Processing*, vol. 26, no. 8, pp. 3981–3994, 2017.

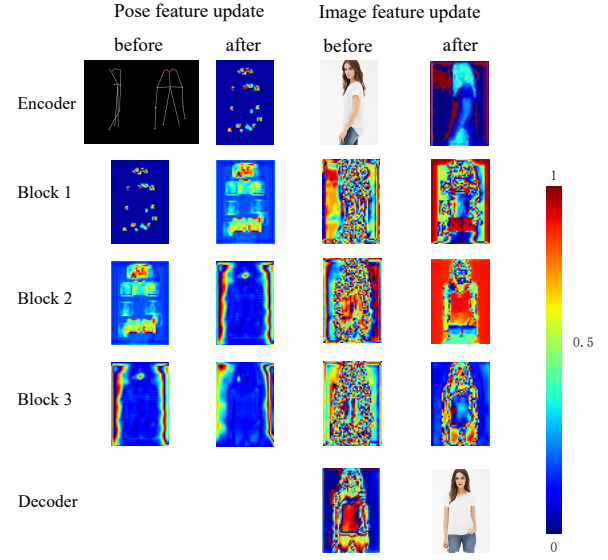


Figure 13: Visualization of features in our model.



Figure 14: Examples of failure cases using our method.

- [3] J. Walker, K. Marino, A. Gupta, and M. Hebert, “The pose knows: Video forecasting by generating pose futures,” in *CVPR*, 2017, pp. 3332–3341.
- [4] P. Esser, E. Sutter, and B. Ommer, “A variational u-net for conditional appearance and shape generation,” in *CVPR*, 2018, pp. 8857–8866.
- [5] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, “Pose guided person image generation,” in *NIPS*, 2017, pp. 406–416.
- [6] L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele, and M. Fritz, “Disentangled person image generation,” in *CVPR*, 2018, pp. 99–108.
- [7] A. Siarohin, E. Sangineto, S. Lathuilliere, and N. Sebe, “Deformable gans for pose-based human image generation,” in *CVPR*, June 2018.
- [8] Y. Li, C. Huang, and C. C. Loy, “Dense intrinsic appearance flow for human pose transfer,” in *CVPR*, 2019, pp. 3693–3702.
- [9] Z. Zhu, T. Huang, B. Shi, M. Yu, B. Wang, and X. Bai, “Progressive pose attention transfer for person image generation,” in *CVPR*, 2019, pp. 2347–2356.
- [10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *NIPS*, 2014, pp. 2672–2680.
- [11] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *ECCV*. Springer, 2016, pp. 694–711.
- [12] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.
- [13] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014.
- [14] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *MICCAI*. Springer, 2015, pp. 234–241.
- [15] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” *arXiv preprint arXiv:1805.08318*, 2018.
- [16] C. Lassner, G. Pons-Moll, and P. V. Gehler, “A generative model of people in clothing,” in *ICCV*, 2017, pp. 853–862.

- [17] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [18] W. Xian, P. Sangkloy, V. Agrawal, A. Raj, J. Lu, C. Fang, F. Yu, and J. Hays, “Texturegan: Controlling deep image synthesis with texture patches,” in *CVPR*, June 2018.
- [19] H. Zhu, H. Su, P. Wang, X. Cao, and R. Yang, “View extrapolation of human body from a single image,” in *CVPR*, 2018, pp. 4450–4459.
- [20] G. Balakrishnan, A. Zhao, A. V. Dalca, F. Durand, and J. Guttag, “Synthesizing images of humans in unseen poses,” in *CVPR*, 2018, pp. 8340–8348.
- [21] C. Si, W. Wang, L. Wang, and T. Tan, “Multistage adversarial losses for pose-based human image synthesis,” in *CVPR*, 2018, pp. 118–126.
- [22] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis, “Viton: An image-based virtual try-on network,” in *CVPR*, 2018, pp. 7543–7552.
- [23] Z. Lahner, D. Cremers, and T. Tung, “Deepwrinkles: Accurate and realistic clothing modeling,” in *ECCV*, 2018, pp. 667–684.
- [24] B. Wang, H. Zheng, X. Liang, Y. Chen, L. Lin, and M. Yang, “Toward characteristic-preserving image-based virtual try-on network,” in *ECCV*, 2018, pp. 589–604.
- [25] N. Neverova, R. Alp Guler, and I. Kokkinos, “Dense pose transfer,” pp. 123–138, 2018.
- [26] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, “Smpl: A skinned multi-person linear model,” *ACM Trans. Graphics*, vol. 34, no. 6, pp. 1–16, 2015.
- [27] W. Liu, Z. Piao, J. Min, W. Luo, L. Ma, and S. Gao, “Liquid warping gan: A unified framework for human motion imitation, appearance transfer and novel view synthesis,” in *CVPR*, 2019, pp. 5904–5913.
- [28] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, “End-to-end recovery of human shape and pose,” in *CVPR*, 2018, pp. 7122–7131.
- [29] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime multi-person 2d pose estimation using part affinity fields,” in *CVPR*, 2017, pp. 7291–7299.
- [30] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.
- [31] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” *arXiv preprint arXiv:1607.08022*, 2016.
- [32] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *ICML*, vol. 30, no. 1, 2013, p. 3.
- [33] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.
- [35] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [36] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, “Scalable person re-identification: A benchmark,” in *CVPR*, 2015, pp. 1116–1124.
- [37] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, “Deepfashion: Powering robust clothes recognition and retrieval with rich annotations,” in *CVPR*, 2016, pp. 1096–1104.
- [38] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime multi-person 2d pose estimation using part affinity fields,” in *CVPR*, 2017, pp. 7291–7299.
- [39] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, “2d human pose estimation: New benchmark and state of the art analysis,” in *CVPR*, 2014, pp. 3686–3693.
- [40] L. Zheng, Y. Yang, and A. G. Hauptmann, “Person re-identification: Past, present and future,” *arXiv preprint arXiv:1610.02984*, 2016.
- [41] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz, “Joint discriminative and generative learning for person re-identification,” in *CVPR*, 2019, pp. 2138–2147.
- [42] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016, pp. 770–778.