Deformable GANs for Pose-based Human Image Generation: Supplementary Material

Aliaksandr Siarohin[1], Enver Sangineto[1], Stéphane Lathuilière[2], and Nicu Sebe[1]
[1]DISI, University of Trento, Italy, [2] Inria Grenoble Rhone-Alpes, France
{aliaksandr.siarohin,enver.sangineto,niculae.sebe}@unitn.it, stephane.lathuiliere@inria.fr

1. Introduction

In this Supplementary Material we report some additional implementation details and we show other quantitative and qualitative results. Specifically, in Sec. 2 we explain how Eq. 8 (main paper) can be efficiently implemented using GPU-based parallel computing, while in Sec. 3 we show how the human-body symmetry can be exploited in case of missed limb detections. In Sec. 4 we train state-of-the-art Person Re-IDentification (Re-ID) systems using a combination of real and generated data, which, on the one hand, shows how our images can be effectively used to boost the performance of discriminative methods and, on the other hand, indirectly shows that our generated images are realistic and diverse. In Sec. 5 we show a direct (qualitative) comparison of our method with the approach presented in [2] and in Sec. 6 we show other images generated by our method, including some failure cases. Note that some of the images in the DeepFashion dataset have been manually cropped (after the automatic generation) to improve the overall visualization quality.

2. Nearest-neighbour loss implementation

Our proposed nearest-neighbour loss is based on the definition of $L_{NN}(\hat{x}, x_b)$ given in Eq. 8 (main paper). In that equation, for each point $\mathbf{p}$ in $\hat{x}$, the "most similar" (in the C_x-based feature space) point $\mathbf{q}$ in x_b needs to be searched for in a $n \times n$ neighborhood of $\mathbf{p}$. This operation may be quite time consuming if implemented using sequential computing (i.e., using a "`for-loop`"). We show here how this computation can be sped-up by exploiting GPU-based parallel computing in which different tensors are processed simultaneously. For the terminology, we refer to the main paper.

Given C_{x_b}, we compute n^2 shifted versions of C_{x_b}: $\{C_{x_b}^{(i,j)}\}$, where (i,j) is a translation offset ranging in a relative $n \times n$ neighborhood ($i, j \in \{-\frac{n-1}{2}, ..., +\frac{n-1}{2}\}$) and $C_{x_b}^{(i,j)}$ is filled with the value $+\infty$ in the borders. Using this translated versions of C_{x_b}, we compute n^2 corresponding difference tensors $\{D^{(i,j)}\}$, where:

$$D^{(i,j)} = |C_{\hat{x}} - C_{x_b}^{(i,j)}| \tag{1}$$

and the difference is computed element-wise. $D^{(i,j)}(\mathbf{p})$ contains the channel-by-channel absolute difference between $C_{\hat{x}}(\mathbf{p})$ and $C_{x_b}(\mathbf{p} + (i,j))$. Then, for each $D^{(i,j)}$, we sum all the channel-based differences obtaining:

$$S^{(i,j)} = \sum_c D^{(i,j)}(c), \tag{2}$$

where c ranges over all the channels and the sum is performed pointwise. $S^{(i,j)}$ is a matrix of scalar values, each value representing the L_1 norm of the difference between a point $\mathbf{p}$ in $C_{\hat{x}}$ and a corresponding point $\mathbf{p} + (i,j)$ in C_{x_b}:

$$S^{(i,j)}(\mathbf{p}) = ||C_{\hat{x}}(\mathbf{p}) - C_{x_b}(\mathbf{p} + (i,j))||_1. \tag{3}$$

For each point $\mathbf{p}$, we can now compute its best match in a local neighbourhood of C_{x_b} simply using:

$$M(\mathbf{p}) = min_{(i,j)} S^{(i,j)}(\mathbf{p}). \tag{4}$$

Finally, Eq. 8 (main paper) becomes:

$$L_{NN}(\hat{x}, x_b) = \sum_{\mathbf{p}} M(\mathbf{p}). \tag{5}$$

Since we do not normalize Eq. 2 by the number of channels nor Eq. 5 by the number of pixels, the final value $L_{NN}(\hat{x}, x_b)$ is usually very high. For this reason we use a small value $\lambda = 0.01$ in Eq. 10 of the main paper when weighting $\mathcal{L}_{NN}$ with respect to $\mathcal{L}_{cGAN}$.

3. Exploiting the human-body symmetry

As mentioned in Sec. 3.1 of the main paper, we decompose the human body in 10 rigid sub-parts: the head, the torso and 8 limbs (left/right upper/lower arm, etc.). When one of the joints corresponding to one of these body-parts has not been detected by the HPE, the corresponding region and affine transformation are not computed and the region-mask is filled with 0. This can happen because of either

that region is not visible in the input image or because of false-detections of the HPE.

However, when the missing region involves a limb (e.g., the right-upper arm) whose symmetric body part has been detected (e.g., the left-upper arm), we can "copy" information from the "twin" part. In more detail, suppose for instance that the region corresponding to the right-upper arm in the conditioning image is R^a_{rua} and this region is empty because of one of the above reasons. Moreover, suppose that R^b_{rua} is the corresponding (non-empty) region in x_b and that R^a_{lua} is the (non-empty) left-upper arm region in x_a. We simply set: $R^a_{rua} := R^a_{lua}$ and we compute f_{rua} as usual, using the (now, no more empty) region R^a_{rua} together with R^b_{rua}.

4. Improving person Re-ID via data-augmentation

The goal of this section is to show that the synthetic images generated with our proposed approach can be used to train discriminative methods. Specifically, we use Re-ID approaches whose task is to recognize a human person in different poses and viewpoints. The typical application of a Re-ID system is a video-surveillance scenario in which images of the same person, grabbed by cameras mounted in different locations, need to be matched to each other. Due to the low-resolution of the cameras, person re-identification is usually based on the colours and the texture of the clothes [3]. This makes our method particularly suited to automatically populate a Re-ID training dataset by generating images of a given person with identical clothes but in different viewpoints/poses.

In our experiments we use Re-ID methods taken from [3, 4] and we refer the reader to those papers for details about the involved approaches. We employ the Market-1501 dataset that is designed for Re-ID method benchmarking. For each image of the Market-1501 training dataset ($\mathcal{T}$), we randomly select 10 target poses, generating 10 corresponding images using our approach. Note that: (1) Each generated image is labeled with the *identity* of the conditioning image, (2) The target *pose* can be extracted from an individual different from the person depicted in the conditioning image (this is different from the other experiments shown here and in the main paper). Adding the generated images to $\mathcal{T}$ we obtain an augmented training set $\mathcal{A}$. In Tab. 1 we report the results obtained using either $\mathcal{T}$ (standard procedure) or $\mathcal{A}$ for training different Re-ID systems. The strong performance boost, orthogonal to different Re-ID methods, shows that our generative approach can be effectively used for synthesizing training samples. It also indirectly shows that the generated images are sufficiently realistic and different from the real images contained in $\mathcal{T}$.

5. Comparison with previous work

In this section we directly compare our method with the results generated by Ma et al. [2]. The comparison is based on the pairs conditioning image-target pose used in [2], for which we show both the results obtained by Ma et al. [2] and ours.

Figs. 1-2 show the results on the Market-1501 dataset. Comparing the images generated by our full-pipeline with the corresponding images generated by the full-pipeline presented in [2], most of the times our results are more realistic, sharper and with local details (e.g., the clothes texture or the face characteristics) more similar to the details of the conditioning image. For instance, in the first and the last row of Fig. 1 and in the last row of Fig. 2, our results show human-like images, while the method proposed in [2] produced images which can hardly be recognized as humans.

Figs. 3-4 show the results on the DeepFashion dataset. Also in this case, comparing our results with [2], most of the times ours look more realistic or closer to the details of the conditioning image. For instance, the second row of Fig. 3 shows a male face, while the approach proposed in [2] produced a female face (note that the DeepFashion dataset is strongly biased toward female subjects [2]). Most of the times, the clothes texture in our case is closer to that depicted in the conditioning image (e.g., see rows 1, 3, 4, 5 and 6 in Fig. 3 and rows 1 and 6 in Fig. 4). In row 5 of Fig. 4 the method proposed in [2] produced an image with a pose closer to the target; however it wrongly generated pants while our approach correctly generated the appearance of the legs according to the appearance contained in the conditioning image.

We believe that this qualitative comparison *using the pairs selected in [2]*, shows that the combination of the proposed deformable skip-connections and the nearest-neighbour loss produced the desired effect to "capture" and transfer the correct local details from the conditioning image to the generated image. Transferring local information while simultaneously taking into account the global pose deformation is a difficult task which can more hardly be implemented using "standard" U-Net based generators as those adopted in [2].

6. Other qualitative results

In this section we present other qualitative results. Fig. 5 and Fig. 6 show some images generated using the Market-1501 dataset and the DeepFashion dataset, respectively. The terminology is the same adopted in Sec. 6.2 of the main paper. Note that, for the sake of clarity, we used a skeleton-based visualization of $P(\cdot)$ but, as explained in the main paper, only the point-wise joint locations are used in our method to represent pose information (i.e., no joint-connectivity information is used).

Table 1: Accuracy of Re-ID methods on the Market-1501 test set (%)

Model	Standard training set ($\mathcal{T}$)		Augmented training set ($\mathcal{A}$)	
	Rank 1	*mAP*	*Rank 1*	*mAP*
IDE + Euclidean [3]	73.9	48.8	**78.5**	**55.9**
IDE + XQDA [3]	73.2	50.9	**77.8**	**57.9**
IDE + KISSME [3]	75.1	51.5	**79.5**	**58.1**
Discriminative Embedding [4]	78.3	55.5	**80.6**	**61.3**

Similarly to the results shown in Sec. 6.2 of the main paper, also these images show that, despite the pose-related general structure is sufficiently well generated by all the different versions of our method, most of the times there is a gradual quality improvement in the detail synthesis from Baseline to DSC to PercLoss to Full.

Finally, Fig. 7 and Fig. 8 show some failure cases (badly generated images) of our method on the Market-1501 dataset and the DeepFashion dataset, respectively. Some common failure causes are:

- Errors of the HPE [1]. For instance, see rows 2, 3 and 4 of Fig. 7 or the wrong right-arm localization in row 2 of Fig. 8.
- Ambiguity of the pose representation. For instance, in row 3 of Fig. 8, the left elbow has been detected in x_b although it is actually hidden behind the body. Since $P(x_b)$ contains only 2D information (no depth or occlusion-related information), there is no way for the system to understand whether the elbow is behind or in front of the body. In this case our model chose to generate an arm considering that the arm is in front of the body (which corresponds to the most frequent situation in the training dataset).
- Rare poses. For instance, row 1 of Fig. 8 shows a girl in an unusual rear view with a sharp 90 degree profile face (x_b). The generator by mistake synthesized a neck where it should have "drawn" a shoulder. Note that rare poses are a difficult issue also for the method proposed in [2].
- Rare object appearance. For instance, the backpack in row 1 of Fig. 7 is light green, while most of the backpacks contained in the training images of the Market-1501 dataset are dark. Comparing this image with the one generated in the last row of Fig. 5 (where the backpack is black), we see that in Fig. 5 the colour of the shirt of the generated image is not blended with the backpack colour, while in Fig. 7 it is. We presume that the generator "understands" that a dark backpack is an object whose texture should not be transferred to the clothes of the generated image, while it is not able to generalize this knowledge to other backpacks.
- Warping problems. This is an issue related to our specific approach (the deformable skip connections). The texture on the shirt of the conditioning image in row 2 of Fig. 8 is warped in the generated image. We presume this is due to the fact that in this case the affine transformations need to largely warp the texture details of the narrow surface of the profile shirt (conditioning image) in order to fit the much wider area of the target frontal pose.

References

[1] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *CVPR*, 2017.

[2] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool. Pose guided person image generation. In *NIPS*, 2017.

[3] L. Zheng, Y. Yang, and A. G. Hauptmann. Person re-identification: Past, present and future. *arXiv:1610.02984*, 2016.

[4] Z. Zheng, L. Zheng, and Y. Yang. A discriminatively learned CNN embedding for person reidentification. *ACM Trans. on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 14(1):13:1–13:20, 2018.

Figure 1: A qualitative comparison on the Market-1501 dataset between our approach and the results obtained by Ma et al. [2]. Columns 1 and 2 show the conditioning and the target image, respectively, which are used as reference by both models. Columns 3 and 4 respectively show the images generated by our full-pipeline and by the full-pipeline presented in [2].

Figure 2: More qualitative comparison on the Market-1501 dataset between our approach and the results obtained by Ma et al. [2].

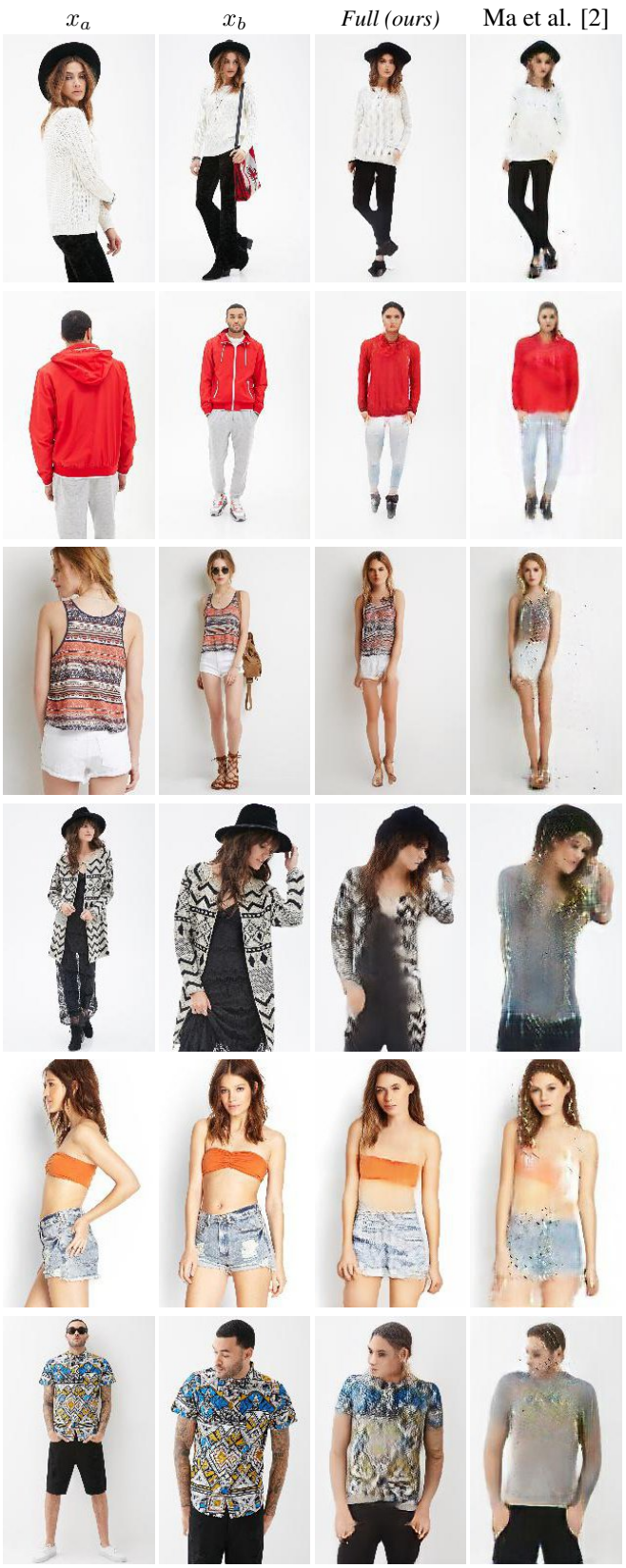

Figure 3: A qualitative comparison on the DeepFashion dataset between our approach and the results obtained by Ma et al. [2].

Figure 4: More qualitative comparison on the DeepFashion dataset between our approach and the results obtained by Ma et al. [2].

Figure 5: Other qualitative results on the Market-1501 dataset.

Figure 6: Other qualitative results on the DeepFashion dataset.

Figure 7: Examples of *badly* generated images on the Market-1501 dataset. See the text for more details.

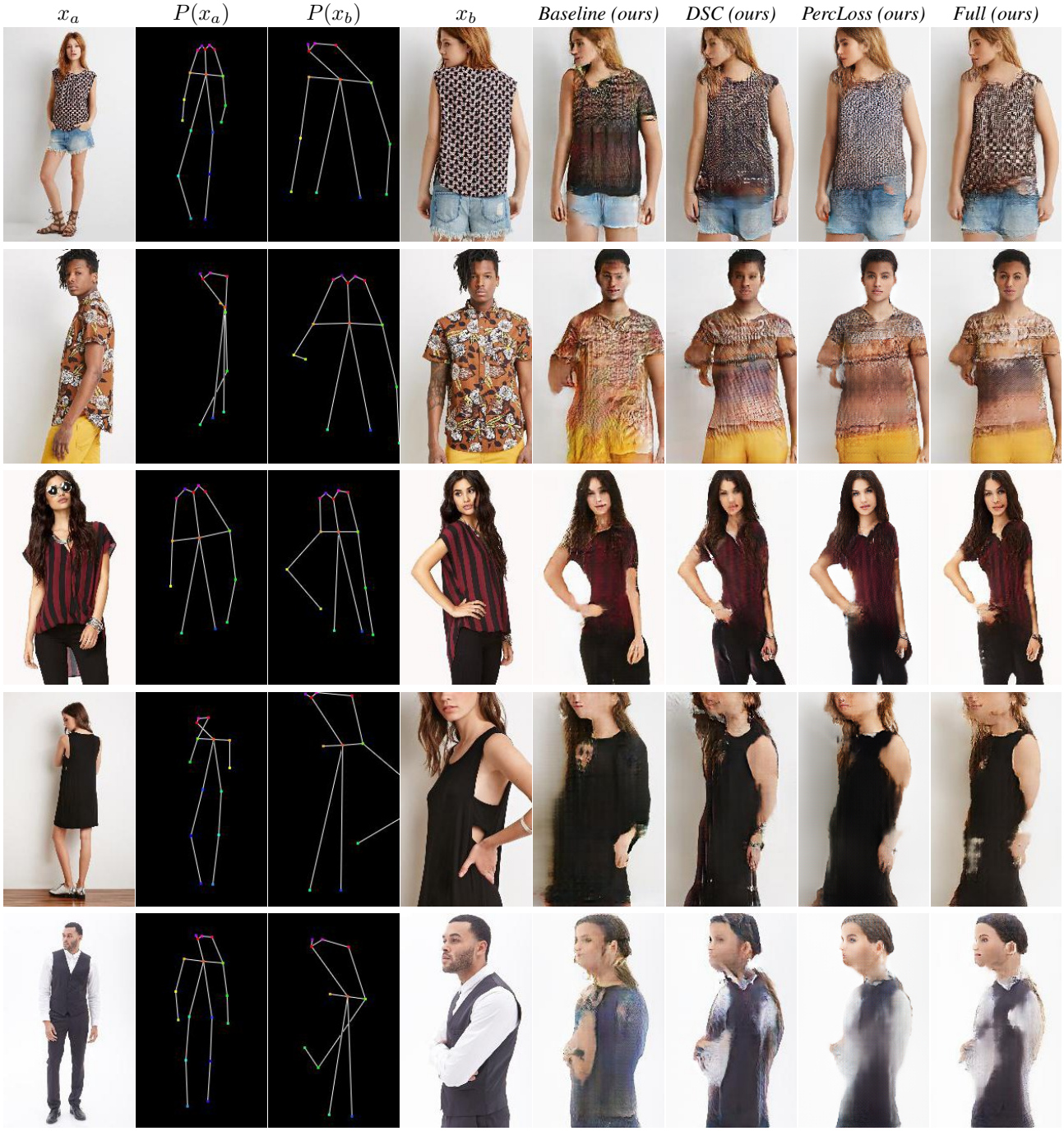

Figure 8: Examples of *badly* generated images on the DeepFashion dataset.