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

Nannan Li
Boston University
nnli@bu.edu

Kevin J Shih
NVIDIA

Bryan A. Plummer
Boston University
bplum@bu.edu

Figure 7: Additional pose transfer examples on DeepFashion. MUST, E2E, and ours are trained with unpaired images. Other methods are supervised by paired data.

Figure 8: Additional pose transfer examples on DeepFashion. MUST, E2E, and ours are trained with unpaired images. Other methods are supervised by paired data.

Figure 9: Additional pose transfer examples on DeepFashion. MUST, E2E, and ours are trained with unpaired images. Other methods are supervised by paired data.

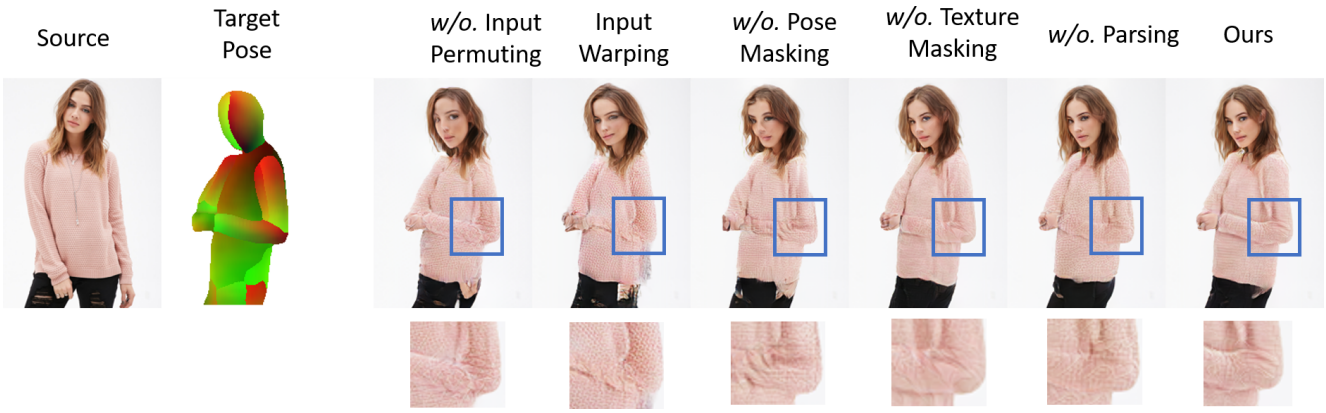
Figure 10: Failure cases in DeepFashion. Many failures are due to incorrect predictions of the source UV map and source parsing map.

		SSIM				IoU			
		Head	Clothes	Arms	Legs	Head	Clothes	Arms	Legs
(a)	*w/o.* Input Permuting (part-wise)	0.299	0.349	0.377	0.407	0.682	0.772	0.642	0.509
	Input Warping, *w/o.* Input Permuting	0.298	0.351	0.376	0.408	0.697	0.782	0.641	0.525
	w/o. Pose Masking	0.298	0.343	0.343	0.382	0.674	0.765	0.625	0.502
	w/o. Texture Masking	0.351	0.369	0.421	0.414	0.697	0.777	0.667	0.507
	w/o. Parsing	0.353	0.371	0.423	**0.441**	0.703	0.787	0.673	**0.552**
(b)	*w/o.*small kernel	0.305	0.368	0.394	0.395	0.674	0.778	0.649	0.505
	w/o. large kernel	0.335	0.365	0.396	0.407	0.689	0.783	0.653	0.522
	Patch Concat	0.331	0.365	0.398	0.407	0.686	0.778	0.653	0.525
	w. blur	0.239	0.354	0.310	0.343	0.625	0.784	0.601	0.454
(c)	*w/o.* Source Pose Branch	0.348	0.371	0.433	0.361	0.706	0.780	0.647	0.428
	Pose Concat	0.321	0.364	0.425	0.371	0.709	0.784	0.648	0.433
(d)	E2E [9]	0.362	0.307	0.337	0.350	0.709	0.761	0.635	0.501
	MUST [6]	**0.371**	0.312	0.404	0.297	0.707	0.737	0.559	0.424
	Ours	**0.371**	**0.377**	**0.452**	0.428	**0.717**	**0.791**	**0.688**	0.531

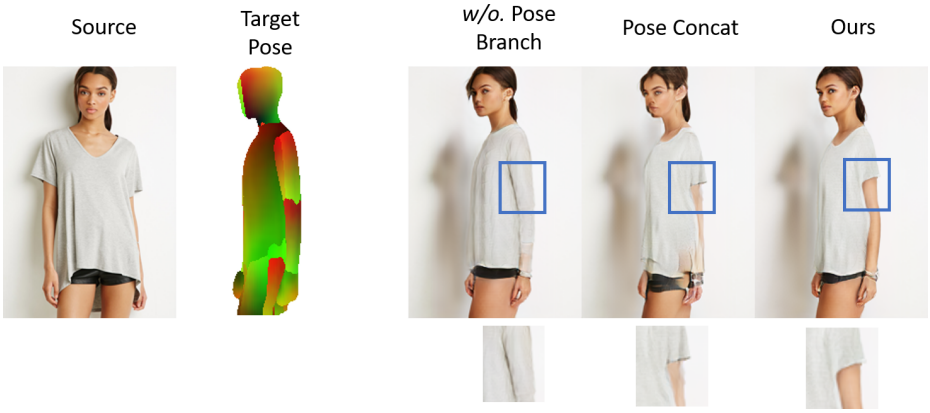
Table 6: Part-wise scores in DeepFashion. (a), (b) and (c) are our ablations. (d) includes self-driven methods trained without paired data.

A. Discussions

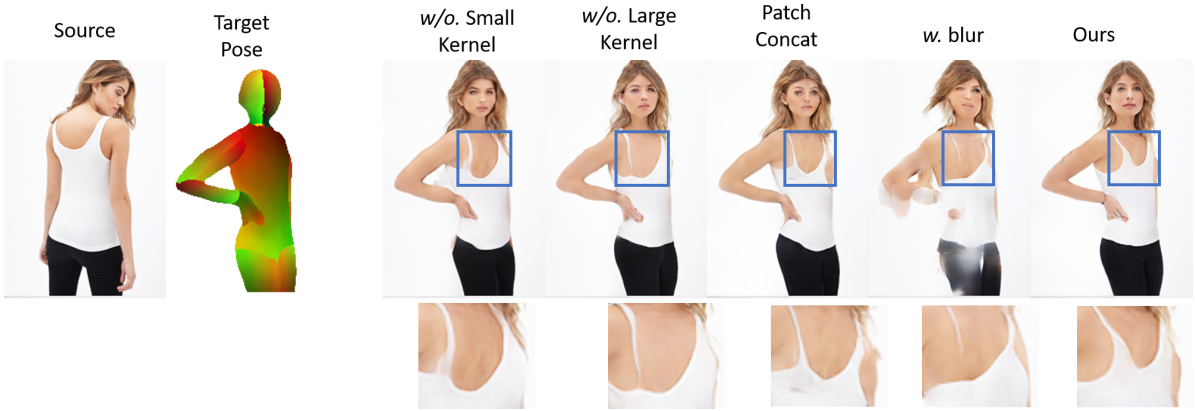
Failure case analysis. Figures 7-9 provide several successful examples generated by the proposed method on DeepFashion. We also obtain the part segments of the images in the test set using the human parser in [12], and then compute their part-wise scores. Table 6(d) shows that our model achieves better texture transfer and shape reconstruction than prior work [9, 6] in terms of part-wise SSIM and IoU. 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. We found the accuracy of the offline human parser and DensePose model greatly affects the transfer results. Figure 10 shows several failed examples due to this type of inaccuracy. In the first row, the coat wrapped around the dress was misclassified 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 the skirt is half-black and half-white because the skirt piece was not



(a) Examples of ablations on the input permutation function.



(b) Examples of ablations on the source pose branch.



(c) Examples of ablations on the dual kernel encoder.

Figure 11: 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.

identified in the parsing map.

Analysis on the ablations. We present the part-wise scores in Table 6 and some visualized examples in Figure 11 to show the functionality of each component of our model. IoU in Table 6 means the Intersection over Union score between the segmentation maps of the generated image and that of the target image. This metric evaluates the shape consistency of each body part after pose transfer.

From Table 6(a), we find that *w/o. Input Permuting* and *Input Warping* have larger drops on arms and heads compared to other body parts, which indicates

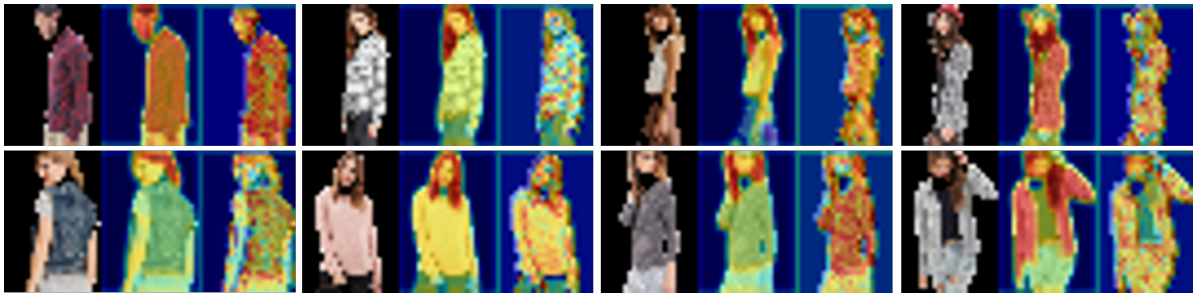


Figure 12: Visualized feature map of the encoded texture features. The feature map is overlaid with the source image. The source image is downsampled to the resolution of the feature map. Each triplet includes a downsampled source image, the feature map from the large-kernel encoder, and the feature map from the small-kernel encoder. Red indicates a higher value and blue means a smaller value.

their incapability of transferring human posture/shape. For example, in Figure 11a, both *w/o. Input Permuting* and *Input Warping* result in a distorted face and obvious edge blurring at the elbow after pose transfer. The ablation model *w/o. Pose Masking* overfits to an identity mapping function between the source and target pose, leading to the lowest IoU on all body parts in Table 6(a). The ablation model *w/o. Texture Masking* is much better than *w/o. Pose Masking* since not masking the texture can still get a correct pose transformation function from the source pose branch. *w/o. Parsing* has lower scores on most body parts compared to the full model, suggesting that including a parsing map in the input is overall beneficial to the pose transfer task.

In Table 6(b), *w/o. small kernel* has worse performance than *w/o. large kernel* and *Patch Concat*, indicating the importance of the small kernel encoder in reducing noise caused by input permutation. *w/o. large kernel* and *Patch Concat* show similar part-wise scores since their receptive fields are respectively limited by the small kernel size and patch size. As shown in Figure 11c, without the small-kernel encoder, the ablation model correctly transfers color, but blurs the edges of the strap. In *w/o.* large kernel and *Patch Concat*, the model has a limited receptive field and thus fails to recover the exact shape of the strap. To see if the large-kernel encoder is learning certain low-level information (*e.g.* color and shape) from permuted patches, we also tried replacing the inputs of the large-kernel encoder with heavily Gaussian blurred image without permutation (denoted by *w.* blur). From both Table 6(c) and the example in Figure 11c, we can see that images generated by *w.* blur are much worse compared to those of the full model. This suggests that features learned by the large-kernel encoder from the permuted image might include high-frequency information that is lost in the Gaussian blurred texture.

In Table 6(c), concatenating the source pose representation with the texture slightly improves IoU, but does not improve SSIM. This means merging the source pose and texture in one branch can provide some shape information, but does not have the ability to match the precise relative position between the source and target pose. For example, in Figure 11b, although *Pose Concat* reconstructs the short sleeve after pose transfer, it has artifacts around the edges of the arm. This is why our full model uses separate source pose and texture branches. By separating the source pose and texture, the source pose branch can directly learn the texture-agnostic geometry transformation between the two poses, and thus better recovers the shape.

To further explore the differences in 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 the range $[0, 1]$. Next, we downsample the image to the resolution of the feature map and overlay the normalized feature map with the downsampled source image. In Figure 12, each triplet includes the downsampled source image, the feature map from the large-kernel encoder, and the feature map from the small-kernel encoder. The feature map given by the 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 the large-kernel encoder might be learning coarse information from the clothing piece (*e.g.*, color and shape), while the small-kernel encoder is learning more fine-grained patterns (*e.g.*, stripe and pleat).

B. Training Details.

We use AdamW optimizer [4] for training with $\beta_1 = 0.5, \beta_2 = 0.999$. The initial learning rate is set to 10^{-3}

Method	FID↓	SSIM↑	M-SSIM↑	LPIPS↓	M-LPIPS↓	IS↑
Supervised by paired images						
GFLA [8]	20.194	0.286	0.815	0.274	0.138	2.546
SPIG [5]	22.043	**0.317**	0.819	**0.271**	0.129	2.761
DPTN [11]	17.929	0.289	0.820	0.266	0.125	2.479
NTED(DF) [7]	38.831	0.191	0.734	0.353	0.212	2.242
No paired images						
PT^2(Ours)	**17.389**	0.280	**0.820**	0.314	**0.122**	**2.789**

Table 7: Pose transfer results for 128×64 resolution images on Market-1501.

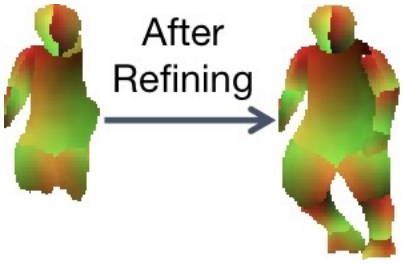


Figure 13: An example of the refined DensePose in Market-1501.

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 [10] to average the learned weights of the generator. We train on 256×176 images in DeepFashon and 128×64 images on Market-1501. Our pose representation is predicted by DensePose [2] and the parsing maps are obtained from CorrPM [12]. 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 [3] 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 for our pose transfer task. An example of the refined DensePose representation is shown in Figure 13. We also add human keypoints predicted from OpenPose [1] as part of the pose representation to improve the accuracy of predicted posture on Market-1501.

C. Analysis on Market-1501

As shown in Table 7, although our model outperforms fully-supervised approaches on M-SSIM and M-LPIPS, 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 part-wise SSIM scores. The 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.

References

[1] 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. 7

[2] 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. 7

[3] 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. 7

[4] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 6

[5] 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. 7

[6] 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. 4

[7] 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. 7

[8] 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. 7

[9] 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. 4

[10] 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. 7

[11] 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. 7

[12] Ziwei Zhang, Chi Su, Liang Zheng, and Xiaodong Xie. Correlating edge, pose with parsing. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2020. 4, 7