Learning Semantic Person Image Generation by Region-Adaptive Normalization (Supplementary Material)

Zhengyao Lv[1], Xiaoming Li[1], Xin Li[2], Fu Li[2], Tianwei Lin[2], Dongliang He[2], Wangmeng Zuo[1,3](✉)
[1]School of Computer Science and Technology, Harbin Institute of Technology, China
[2]Department of Computer Vision Technology (VIS), Baidu Inc.
[3]Pazhou Lab, Guangzhou, China
cszy98@gmail.com {csxmli, wmzuo}@hit.edu.cn

A. Training procedures

In order to explore the training manners of the two-stage model that may bring benefits to the final results, in this supplemental material, we conduct three schemes, (i) $S_1 \triangleright S_2$: pre-training the SPATN model in stage one, then fixing its parameters and training SPGNet in stage two; (ii) $S_1 \bowtie S_2$: pre-training the SPATN model in stage one, then jointly training with the SPGNet in stage two; (iii) $S_1 \parallel S_2$: training the SPATN and SPGNet simultaneously, in which the SPGNet takes the ground-truth target parsing map S_{p_t} as input. During the inference, the model takes the predicted target parsing map $\hat{S}_{p_t}$ from the SPATN model as input to generate the final results. The comparison results are shown in Table A. We find that though the joint training manner ($S_1 \bowtie S_2$) usually obtains great performance in many fields, it shows inferior results in this task. We analyze that 1) due to the long pathway of SPGNet, the gradient from the $\mathcal{L}_{full}$ to update the SPATN model tends to be zero, making the SPATN model seldomly benefits from the end-to-end training, 2) more importantly, there is a gap between the predicted semantic parsing map and the ground-truth. The inaccurate prediction may confuse the learning of the SPGNet, thus has a negative impact on the generation process. Therefore, we adopt the $S_1 \parallel S_2$ scheme to train our two-stage model and use the predicted target semantic maps in the first stage to guide the final generation of the target image in the inference.

Scheme	SSIM ↑	FID ↓	PCKh ↑	LPIPS ↓
$S_1 \triangleright S_2$	0.785	17.340	0.96	0.2240
$S_1 \bowtie S_2$	**0.792**	24.309	0.96	0.2435
$S_1 \parallel S_2$	0.782	**12.243**	**0.97**	**0.2105**

Table A. The quantitative comparison of different training schemes on our DeepFashion test set. ↑ (↓) means higher (lower) is better.

B. Network Architecture of SPGNet

Our SPGNet consists of a pose encoder, an appearance encoder and a decoder which is composed of several SPG-Blocks. Table B shows the details of SPGNet. Table C and Table D give the detailed network architecture of SPG-Block and Feature Deformation Module, respectively. Conv.(d, k, s) and ConvT.(d, k, s) denote convolution and transposed convolution layer, where d, k and s are output dimention, convolution kernel size and stride, respectively. And $dim(f_{t-1})$ is the dimention of feature map f_{t-1}. LReLU is leaky ReLU with negative slope c. BN and IN represent batch normalization and instance normalization, respectively. The denotation of input is the same as that in the paper. I_{p_s} denotes the source appearance image. S_{p_s} and $\hat{S}_{p_t}$ are source semantic parsing map and predicted target semantic parsing map. p is keypoint heat map of the target pose. Φ_{2D} and V denote the projection of the predicted 3D flow and visibility map from Intr-Flow [1], respectively.

C. Distance Map

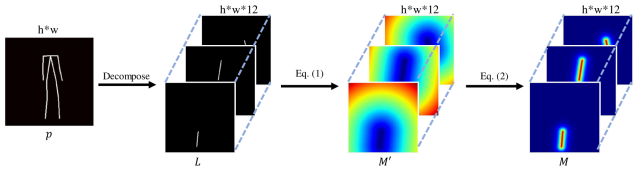

Figure A. The details of the distance map generation.

It may be not enough to use only keypoints skeleton as pose representations to generate semantic maps, especially when the poses are complex or rare in the dataset. The introduction of distance maps makes pose representation more robust to various poses. We generate 12 lines $\{L_m|_{m=1}^{12}\}$ distance map with 18-channels keypoint heat maps to represent the body skeleton. Each skeleton generates one channel distance map, thus we can generate a 12-channels distance map $\{M_m|_{m=1}^{12}\}$, which has the same width and height of the source image. The values in (x, y) of each M_m is calculated by the smallest distance between the point (x, y) and the skeleton L_m. The m-*th* distance map can be obtained by:

$$M'_m(x,y) = \min_{(x',y')\in L_m} \{\sqrt{(x-x')^2 + (y-y')^2}\}, \quad (1)$$

where (x', y') denotes the point on the skeleton L_m. Here, we further normalize these values by introducing a negative parameter κ. The final distance map is then defined as:

$$M_m(x,y) = exp(\kappa * M'_m(x,y)), \quad (2)$$

where $m \in \{1, 2, 3...12\}$ represents the m-*th* skeleton. In this way, the closer the point to the skeleton, the larger the value is. Thus it can well model the body structure. The process of distance map generation is shown in Fig. A. The experimental results on DeepFashion show that without using the distance map, the mIOU of the predicted semantic maps is 0.520, while when using the distance map, the mIOU is 0.539. By using the distance map, our method can generate plausible semantic parsing maps even though the poses are

Input	I_{p_s} $(3 \times \times 256 \times 256)$ Φ_{2D} $(2 \times \times 256 \times 256)$ V $(1 \times \times 256 \times 256)$	p $(18 \times 256 \times 256)$	I_{p_s}, $(3 \times 256 \times 256)$ S_{p_s}, $\hat{S}_{p_t}$, $(20 \times 256 \times 256)$
Feature Extraction	Conv. (32, 1, 1) ResidualBlock, ResidualBlock, FeatureWarp ReLU, Conv. (64, 3, 2), BN ResidualBlock, ResidualBlock, FeatureWarp ReLU, Conv. (128, 3, 2), BN ResidualBlock, ResidualBlock, FeatureWarp ReLU, Conv. (128, 3, 2), BN ResidualBlock, ResidualBlock, FeatureWarp ReLU, Conv. (128, 3, 2), BN ResidualBlock, ResidualBlock ReLU, Conv. (128, 3, 2), BN ResidualBlock, ResidualBlock ReLU, Conv. (128, 3, 2), BN	Conv. (32, 1, 1) ResidualBlock, ResidualBlock ReLU, Conv. (64, 3, 2), BN ResidualBlock, ResidualBlock ReLU, Conv. (128, 3, 2), BN ResidualBlock, ResidualBlock ReLU, Conv. (128, 3, 2), BN ResidualBlock, ResidualBlock ReLU, Conv. (128, 3, 2), BN ResidualBlock, ResidualBlock ReLU, Conv. (128, 3, 2), BN ResidualBlock, ResidualBlock ReLU, Conv. (128, 3, 2), BN	Conv. (32, 3, 1), IN, LReLU(0.2) Conv. (64, 3, 2), IN, LReLU(0.2) Conv. (128, 3, 2), IN, LReLU(0.2) ConvT.(64, 3, 2), IN, LReLU(0.2) ConvT.(32, 3, 2), IN, LReLU(0.2) ConvT.(128, 3, 1), Tanh Region Average Pooling
output	warped appearance features f_a^w	pose features f_p	Style Codes ST
Feature Fusion	ReLU, Conv. (512, 3, 1) PixelShuffle(2), SPGBlock 1 ReLU, Conv. (512, 3, 1) PixelShuffle(2), SPGBlock 2 ReLU, Conv. (512, 3, 1) PixelShuffle(2), SPGBlock 3 ReLU, Conv. (512, 3, 1) PixelShuffle(2), SPGBlock 4 ReLU, Conv. (384, 3, 1) PixelShuffle(2), SPGBlock 5 ReLU, Conv. (256, 3, 1) PixelShuffle(2), SPGBlock 6 ReLU, Conv. (128, 3, 1) PixelShuffle(2), SPGBlock 7 Conv. (3, 7, 1) Tanh()		
output	$\hat{I}_{p_t}$ $(3 \times 256 \times 256)$		

Table B. Details of SPGNet Architecture.

Input	f_{t-1}	f_a^w	f_p	ST	$\hat{S}_{p_t}$
SPGBlock	f_{t-1}	Concat		broadcasting	
	f_{t-1}	SEAN, ReLU Conv. $(dim(f_{t-1}), 1, 1)$			
	Concat				
	SEAN, ReLU				
	Conv. $(dim(f_{t-1}), 3, 1)$				
	$+f_{t-1}$				
output	f_t				

Table C. Details of SPGBlock.

Input	f_a	Φ_{2D}	V
Feature Deformation	Warp		V
	Expand Feature $f_a^{w'} * (V == 1)$ $f_a^{w'} * (V == 0)$		
	Visible part		Invisible part
	Concat		
	ResidualBlock		
output	f_a^w		

Table D. Details of Feature Deformation Module.

complex or rare (*e.g.*, crossed hand in Figure B), which will further benefit the later target image generation.

D. More Qualitative Results

In this section, we show more visual comparison with the competing methods (*i.e.* PATN [5], Intr-Flow [1], GFLA [3], XingGAN [4], ADGAN [2]) on DeepFashion and Market-1501 in Figs. C and D, respectively. It can be seen that our method can generate more semantic, consistent, and photo-realistic results. Besides, we also show more visual comparison of different SPGNet variants in Fig. E.

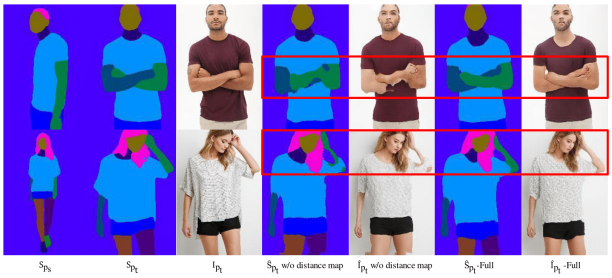

Figure B. The gain of the distance map to the image generation.

Figure C. More visual comparison with the competing methods on DeepFashion.

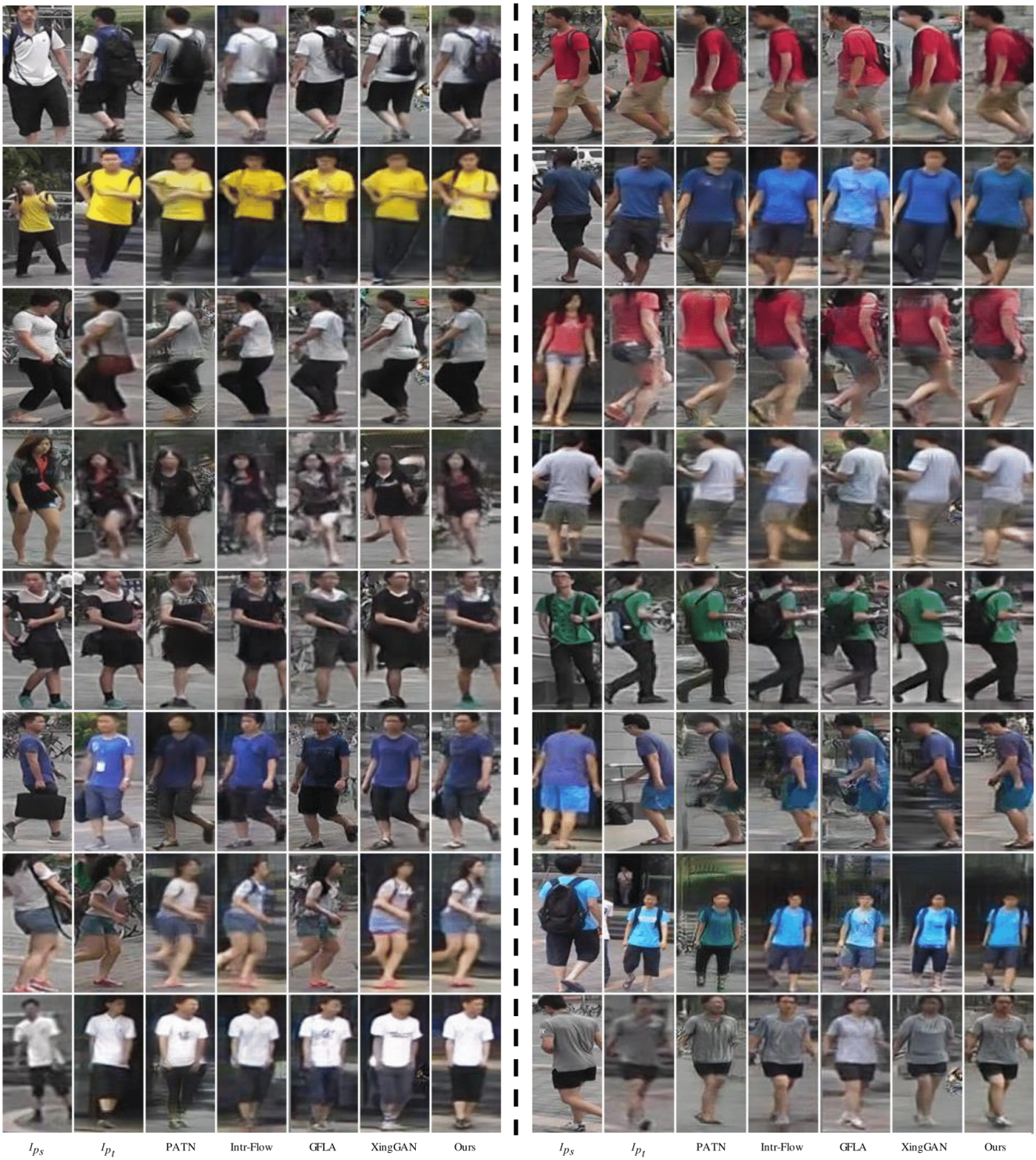

Figure D. More visual comparison with the competing methods on Market-1501.

Figure E. More visual comparison of different SPGNet variants.

References

[1] Yining Li, Chen Huang, and Chen Change Loy. Dense intrinsic appearance flow for human pose transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3693–3702, 2019. 1, 2

[2] Yifang Men, Yiming Mao, Yuning Jiang, Wei-Ying Ma, and Zhouhui Lian. Controllable person image synthesis with attribute-decomposed gan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[3] 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*, pages 7690–7699, 2020. 2

[4] Hao Tang, Song Bai, Li Zhang, Philip HS Torr, and Nicu Sebe. Xinggan for person image generation. *arXiv preprint arXiv:2007.09278*, 2020. 2

[5] 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*, pages 2347–2356, 2019. 2