Deformable GANs for Pose-based Human Image Generation

CVPR 2018 Poster

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

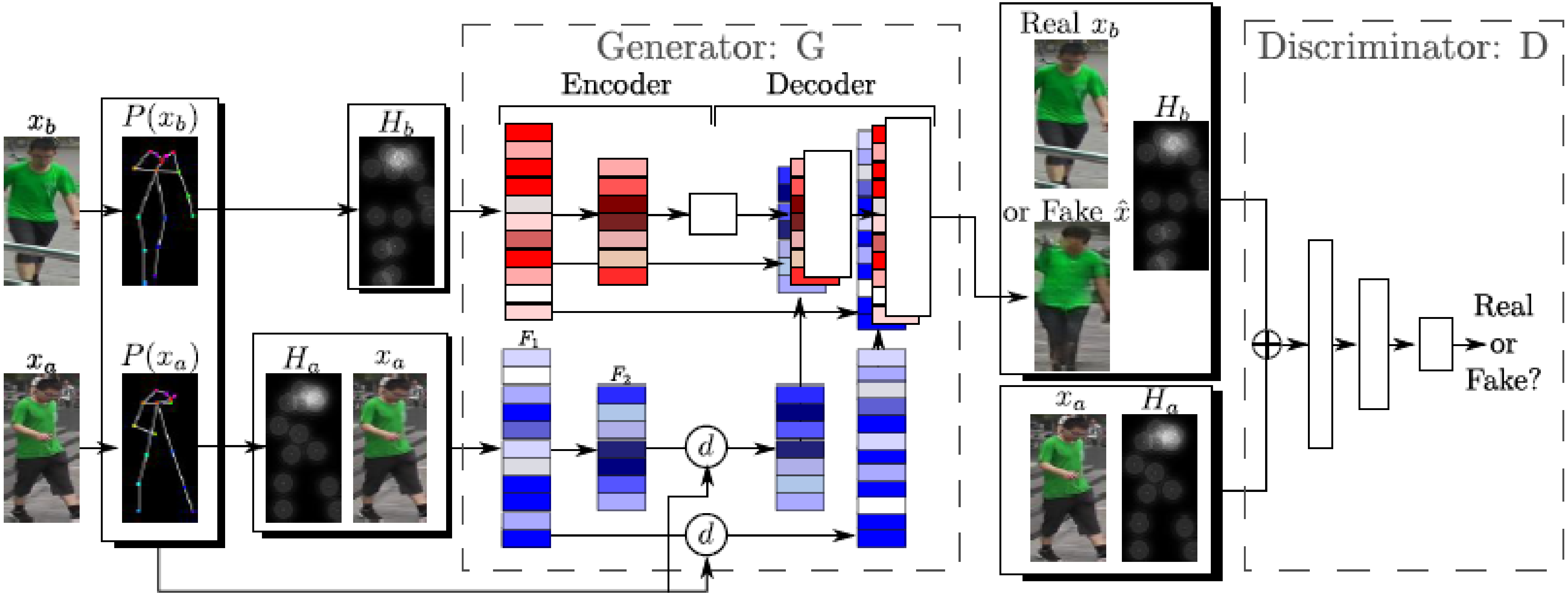
Problem Formulation

- Input: Reference image x_a + pose $P(x_a)$ + Target Pose $P(x_b)$
- Output: x_b
- Training samples:

$$\{(x_a^{(i)}, x_b^{(i)})\}_{i=1,\dots,N}$$

(a) Aligned task (b) Unaligned task

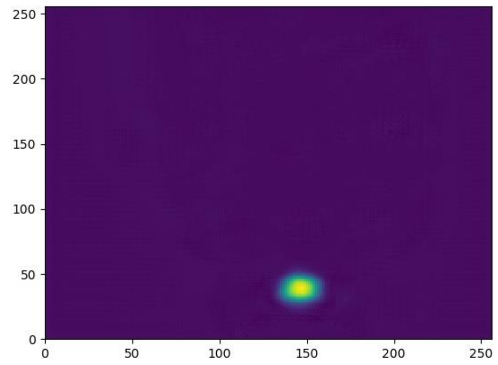
Network Structure

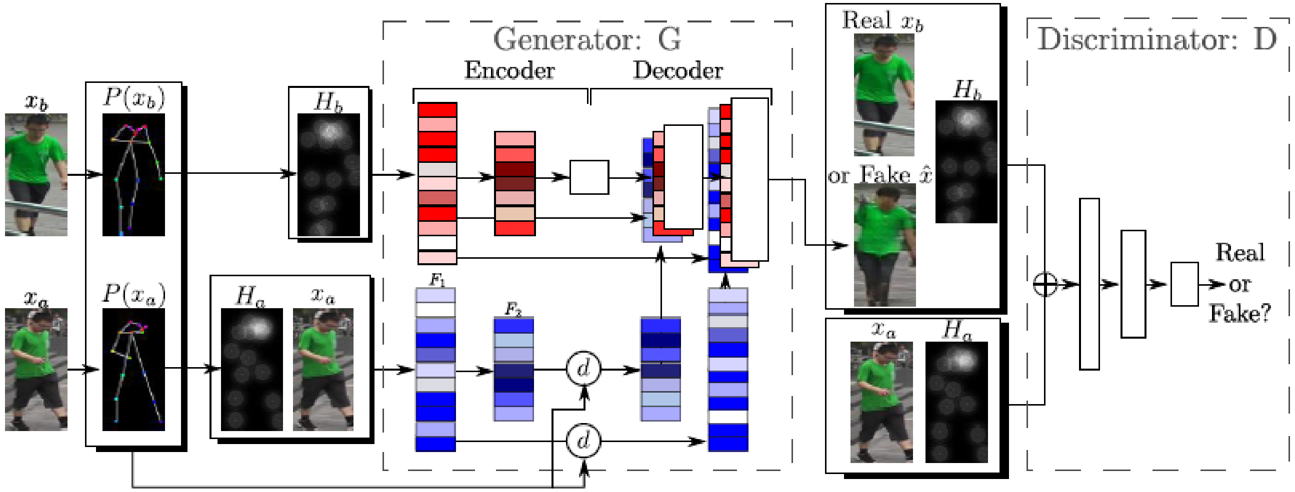


Network Structure

- The representation of pose
  - Belief map in Gaussian peak
  - Extracted with OpenPose (18 joints)

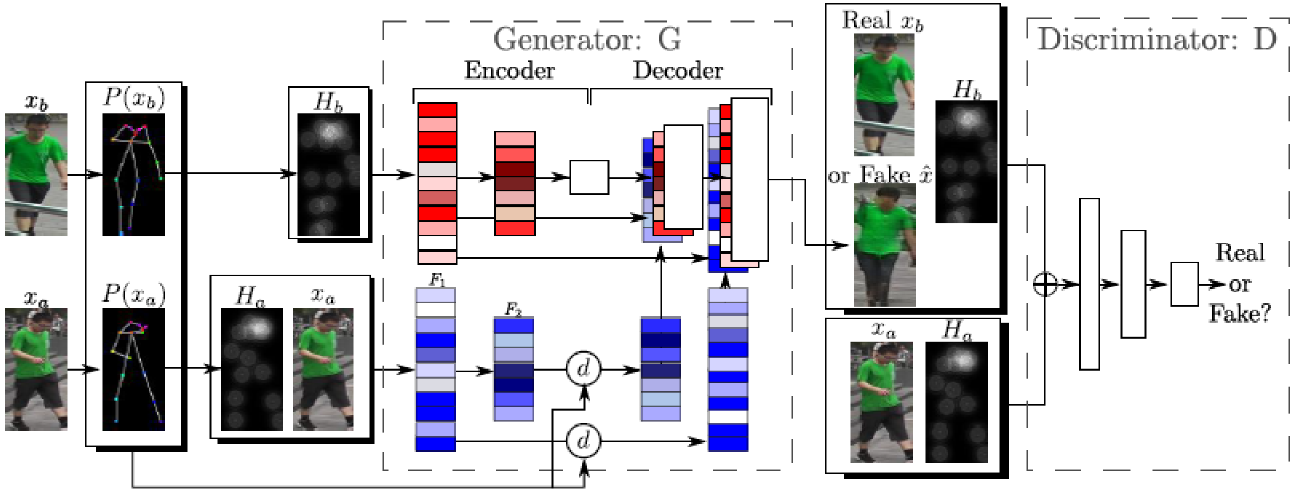
$$H_j(\mathbf{p}) = exp\left(-\frac{\|\mathbf{p}-\mathbf{p}_j\|}{\sigma^2}\right)$$





Network Structure

- Two-Stream Encoder
  - Pose encoder: UNet
  - Appearance encoder:
    - Unet + Deformable skip
- Decoder
- Discriminator



Network Structure

- Deformable Skip Connections
  - Decomposing an body in 10 rigid subparts ➡ 10 masks

    head, torso, left/right upper/lower arm, left/right upper/lower leg
  - Computing affine transformations

$$\min_{\mathbf{k}_h} \sum_{\mathbf{p}_j \in R_h^a, \mathbf{q}_j \in R_h^b} ||\mathbf{q}_j - f_h(\mathbf{p}_j; \mathbf{k}_h)||_2^2$$

  - Approximate the object deformation

$$F_h' = f_h(F \odot M_h),$$

$$d(F(\mathbf{p}, c)) = max_{h=1,...,10} F_h'(\mathbf{p}, c),$$

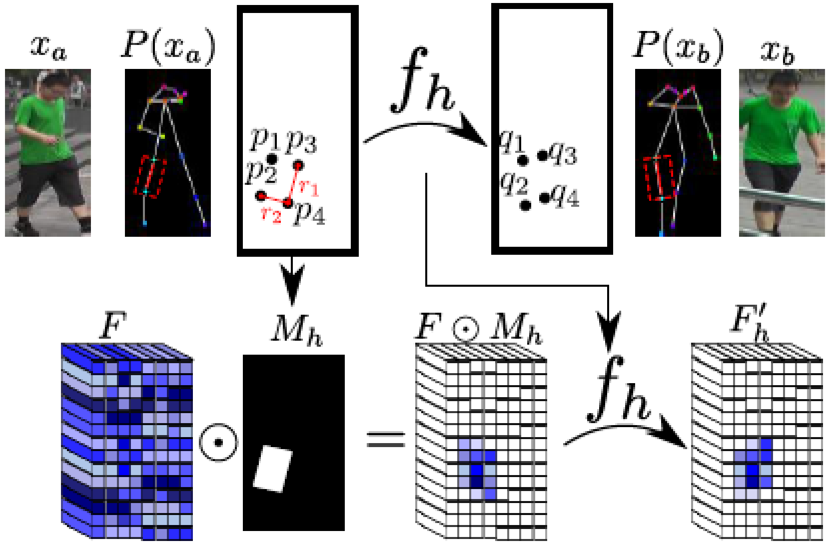


Figure 3: For each specific body part, an affine transformation f_h is computed. This transformation is used to "move" the feature-map content corresponding to that body part.

Training

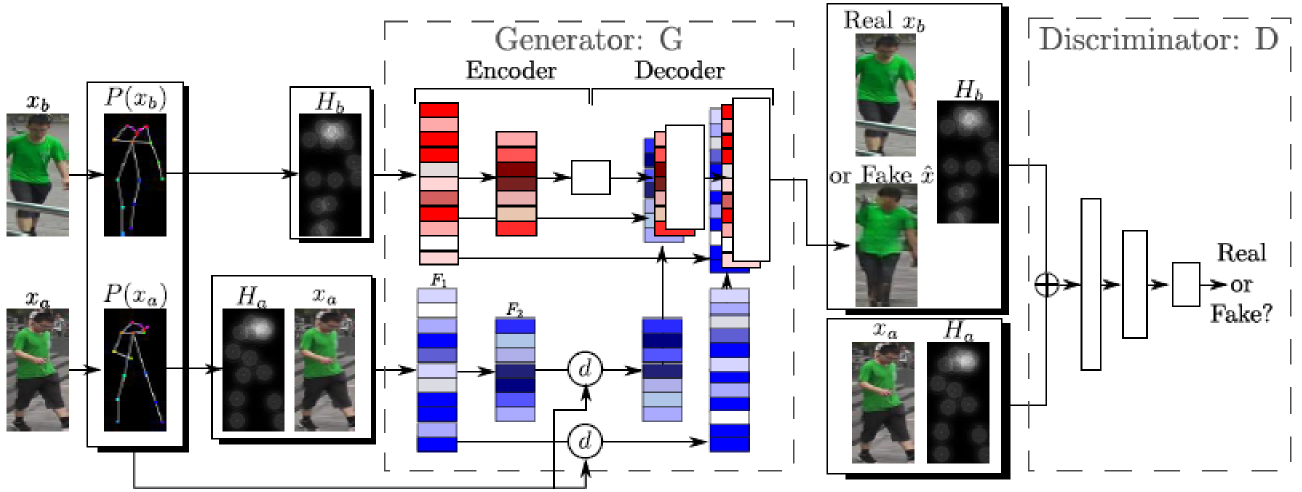
$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{NN}(G)$$

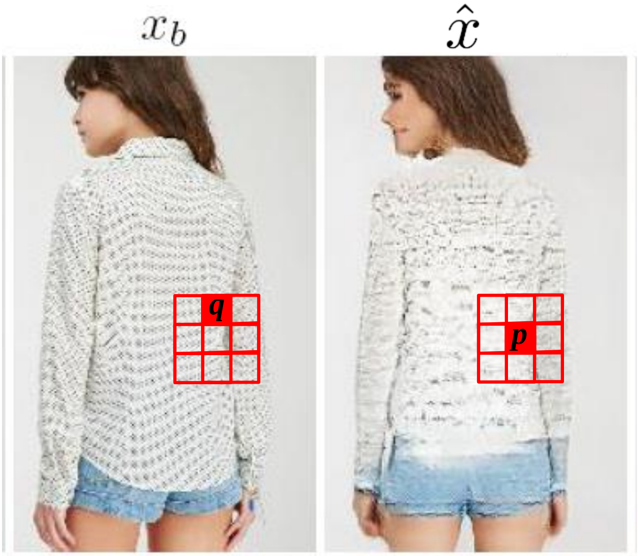
- Adversarial Loss

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{(x_a, x_b) \in \mathcal{X}}[\log D(x_a, H_a, x_b, H_b)] + \mathbb{E}_{(x_a, x_b) \in \mathcal{X}, z \in \mathcal{Z}}[\log(1 - D(x_a, H_a, \hat{x}, H_b))],$$

- Nearest Neighbor Loss
  - Don't have to be pixel aligned with the GT

$$L_{NN}(\hat{x}, x_b) = \sum_{\mathbf{p} \in \hat{x}} min_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} ||g(\hat{x}(\mathbf{p})) - g(x_b(\mathbf{q}))||_1,$$





Ground Truth Output

Experiments

- Dataset
  - Market-1501: 32668 images of 1501 people, 128 x 64
  - Deep Fashion: In-shop retrieval: 52712 images, 256 x 256
- Compared with State-of-the-Art
  - DS: detection score from SSD

Table 1: Comparison with the state of the art. (∗) These values have been computed using the code and the network weights released by Ma et al. [12] in order to generate new images.

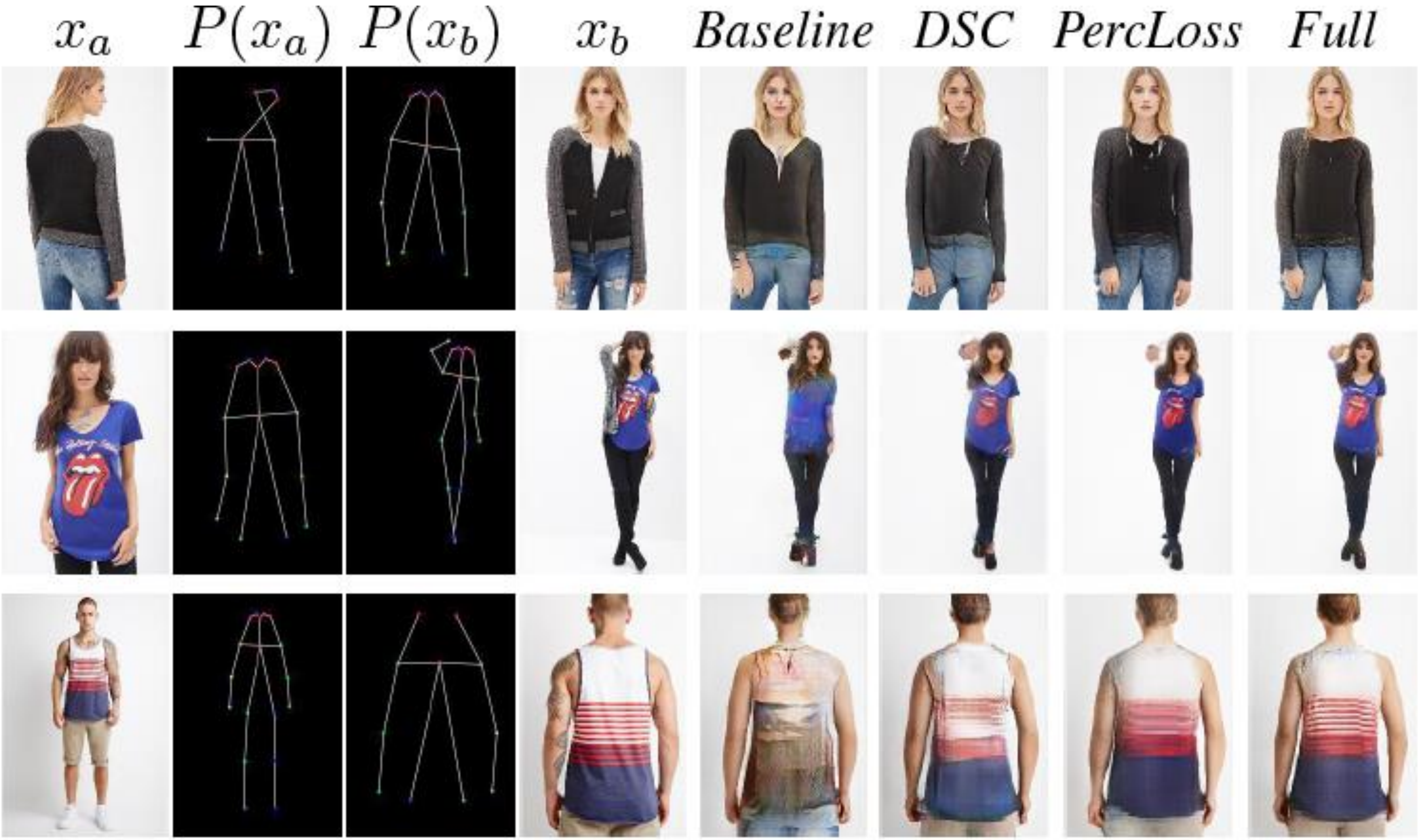
	Market-1501					DeepFashion		
Model	*SSIM*	*IS*	*mask-SSIM*	*mask-IS*	*DS*	*SSIM*	*IS*	*DS*
Ma et al. [12]	0.253	**3.460**	0.792	3.435	0.39*	**0.762**	3.090	0.95*
Ours	**0.290**	3.185	**0.805**	**3.502**	**0.72**	0.756	**3.439**	**0.96**
Real-Data	1.00	3.86	1.00	3.36	0.74	1.000	3.898	0.98

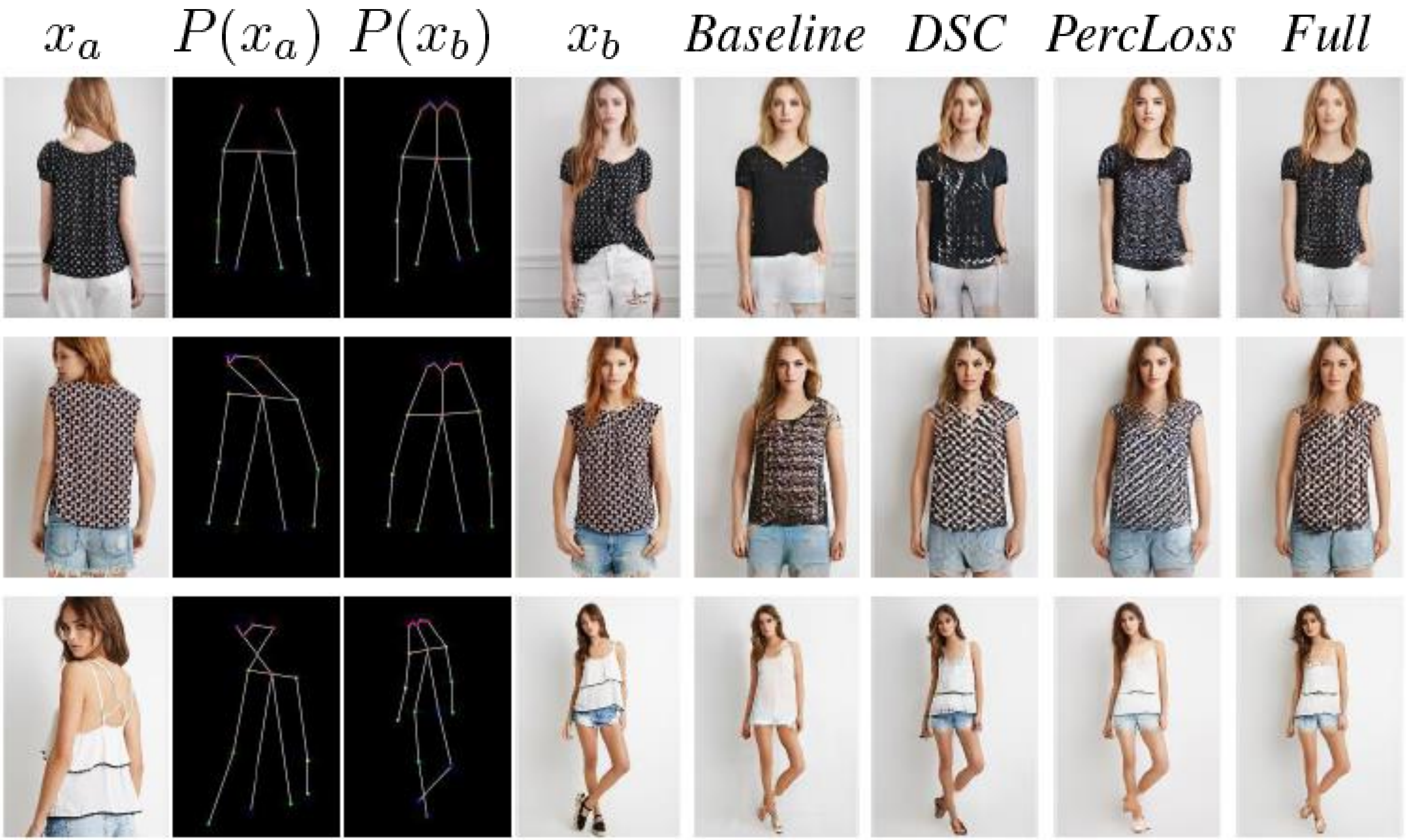
Experiments

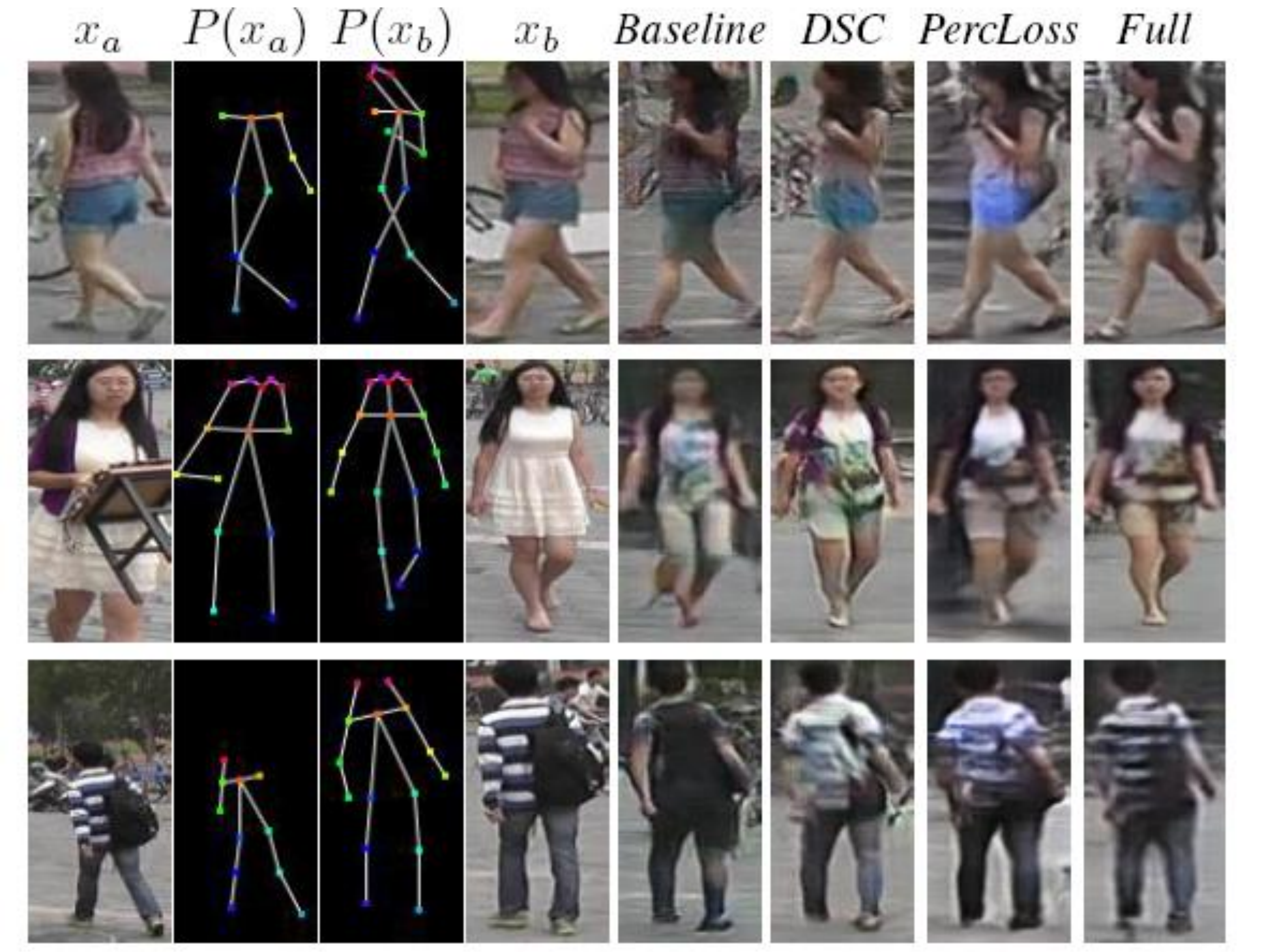
- Ablation Study
  - **Baseline**: Unet **w/o** deformable skip
  - **DSC**: Unet **with** deformable skip + **L1** loss
  - **PerLoss**: Unet **with** deformable skip + **Perceptual** loss
  - **Full**: Unet **with** deformable skip + **NN** loss

Table 3: Quantitative ablation study on the Market-1501 and the DeepFashion dataset.

	Market-1501					DeepFashion	
Model	*SSIM*	*IS*	*mask-SSIM*	*mask-IS*	*DS*	*SSIM*	*IS*
Baseline	0.256	3.188	0.784	3.580	0.595	0.754	3.351
DSC	0.272	**3.442**	0.796	**3.666**	0.629	0.754	3.352
PercLoss	0.276	3.342	0.788	3.519	0.603	0.744	3.271
Full	**0.290**	3.185	**0.805**	3.502	**0.720**	**0.756**	**3.439**
Real-Data	1.00	3.86	1.00	3.36	0.74	1.000	3.898

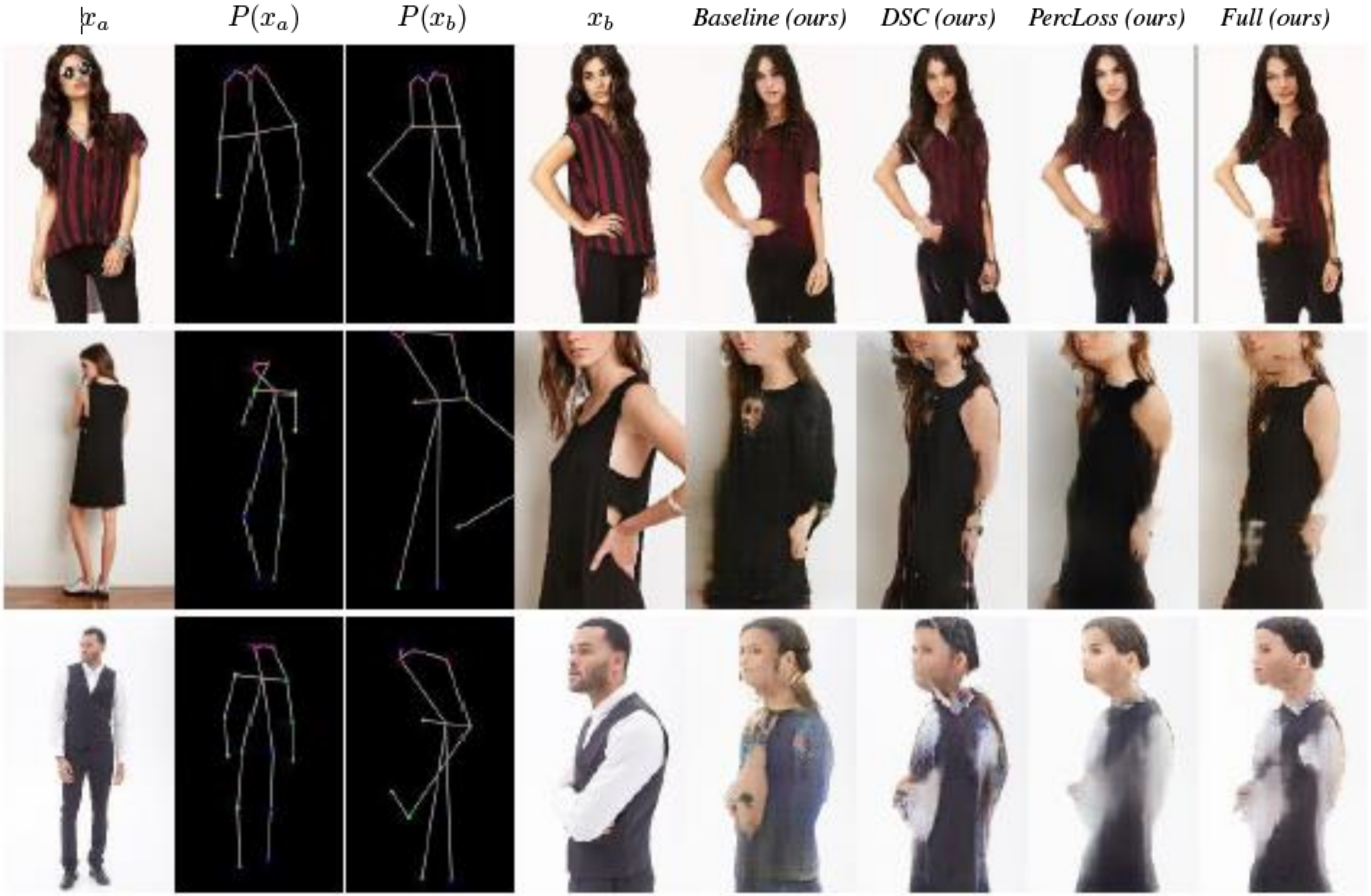

x_a
$P(x_a)$
$P(x_b)$
x_b
Baseline
DSC
PercLoss
Full


x_a
$P(x_a)$
$P(x_b)$
x_b
Baseline
DSC
PercLoss
Full


x_a
$P(x_a)$
$P(x_b)$
x_b
Baseline
DSC
PercLoss
Full

Experiments

- Failure cases


x_a
$P(x_a)$
$P(x_b)$
x_b
Baseline (ours)
DSC (ours)
PercLoss (ours)
Full (ours)

Thanks

Q&A

Sijie Song