Everybody Dance Now

Chan et al., 2018

Presentation by Denis Tarasov


Source Video
Detected Pose
Source to Target 1 Result
Source to Target 2 Result

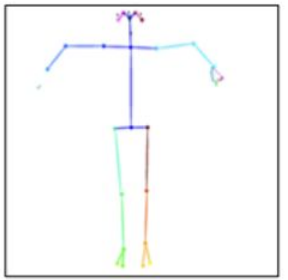
Problem

per-frame image-to-image translation with spatio-temporal smoothing

Given

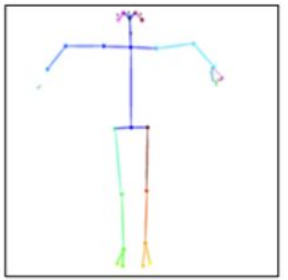
two videos:

- target person (use appearance)
- source person (use motions)



Method overview

- pose detection
- global pose normalization
- mapping to the target subject using GAN



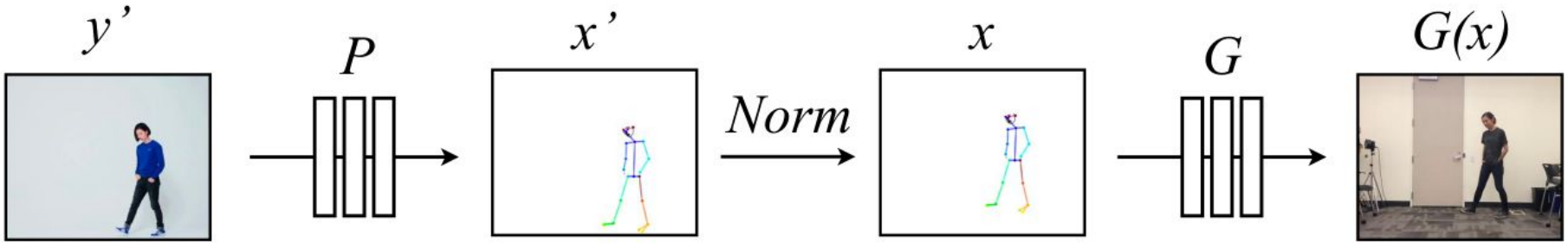
Pose detection

pretrained state-of-the-art pose detector

OpenPose

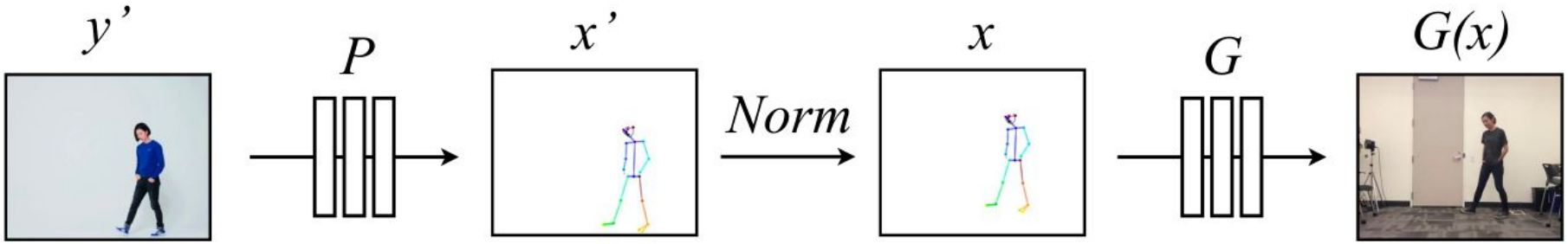
Global pose normalization

accounts for differences between source and target body shapes and locations within frame



Global pose normalization

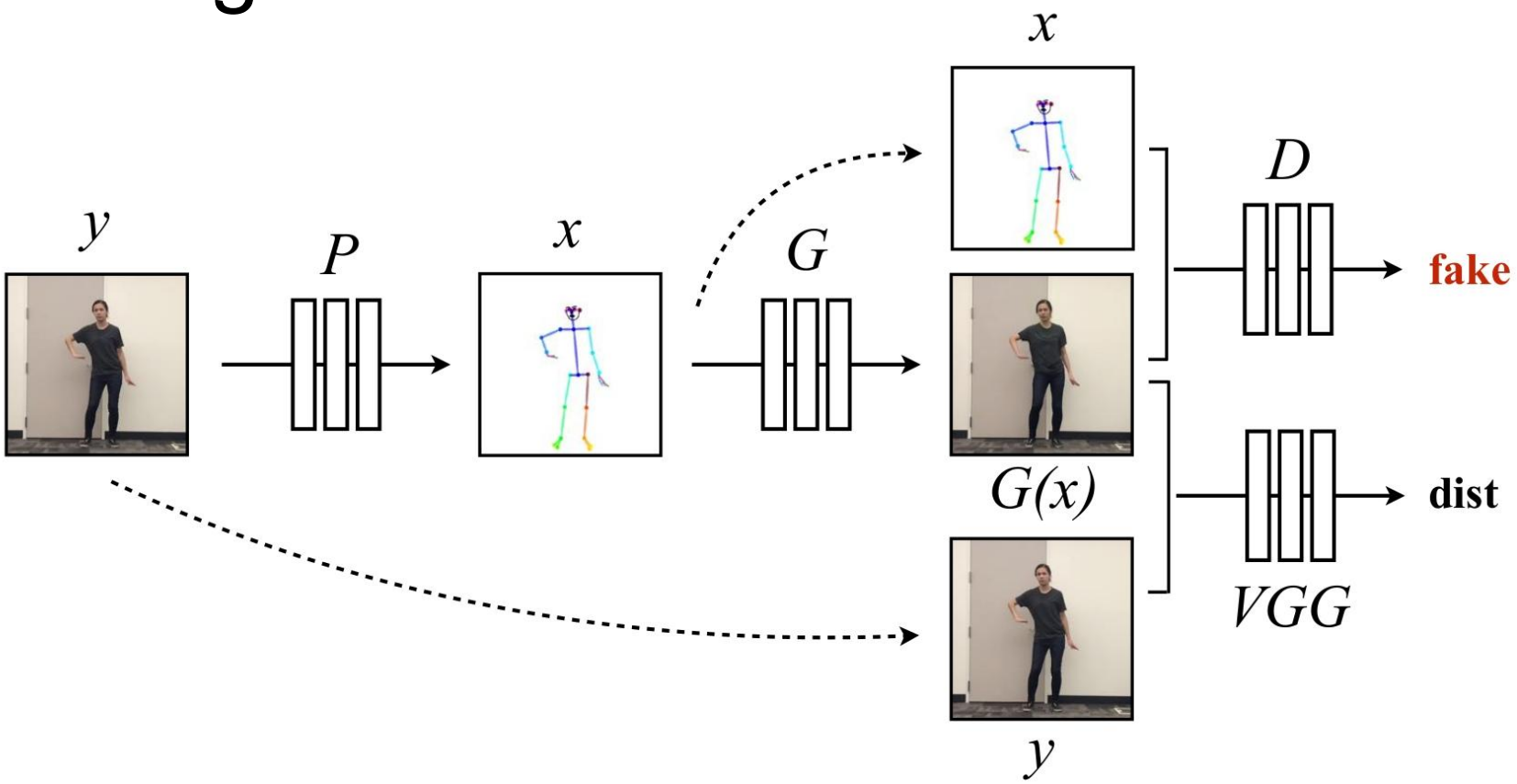
long story short: distance between feet of target and source person, then transformation



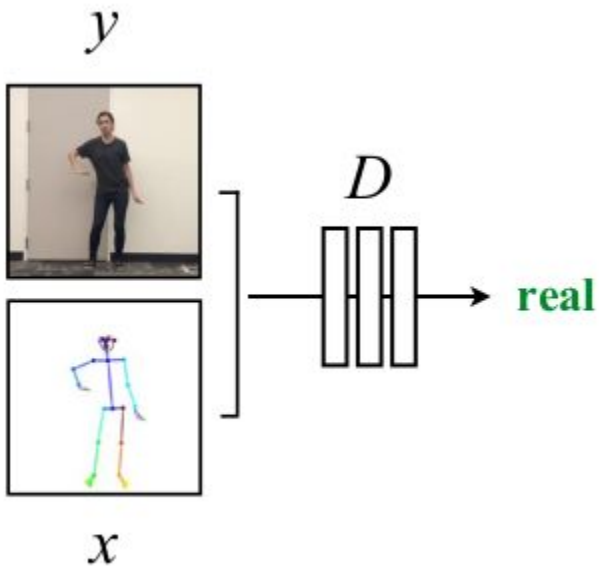
Mapping to the target subject

learn mapping from the normalized pose with adversarial learning

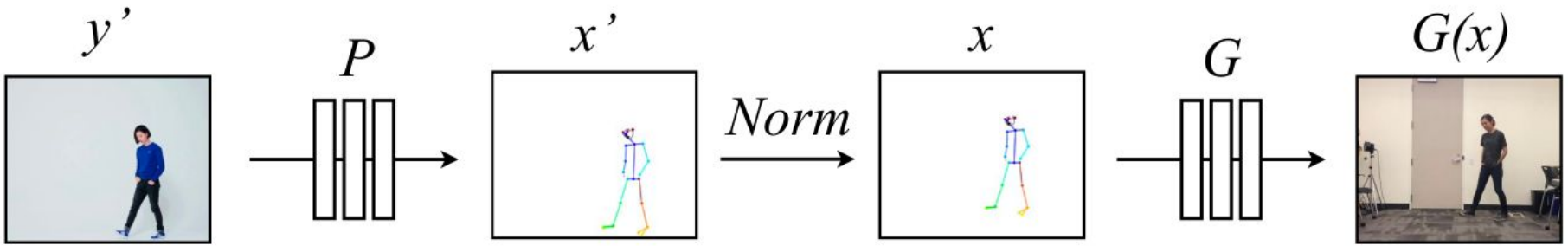
Training



Training



Transfer



pix2pixHD framework

$$\min_{G}\left(\left(\max_{D_1,D_2,D_3}\sum_{k=1,2,3}\mathcal{L}_{\mathrm{GAN}}(G,D_k)\right)+\lambda_{FM}\sum_{k=1,2,3}\mathcal{L}_{\mathrm{FM}}(G,D_k)\right.$$
$$\left.+\lambda_{VGG}\mathcal{L}_{VGG}(G(x),y)\right)$$

Good old GAN loss

$$\mathcal{L}_{\mathrm{GAN}}(G, D) = \mathbb{E}_{(x,y)}[\log D(x, y)] + \mathbb{E}_x[\log(1 - D(x, G(x))]$$

3 discriminators (pix2pixHD)

3 discriminators for different scales of image (1x, 2x and 4x smaller that the original one)

Feature matching loss (pix2pixHD)

$$\mathcal{L}_{\text{FM}}(G, D_k) = \mathbb{E}_{(\mathbf{s},\mathbf{x})} \sum_{i=1}^{T} \frac{1}{N_i} [||D_k^{(i)}(\mathbf{s}, \mathbf{x}) - D_k^{(i)}(\mathbf{s}, G(\mathbf{s}))||_1]$$

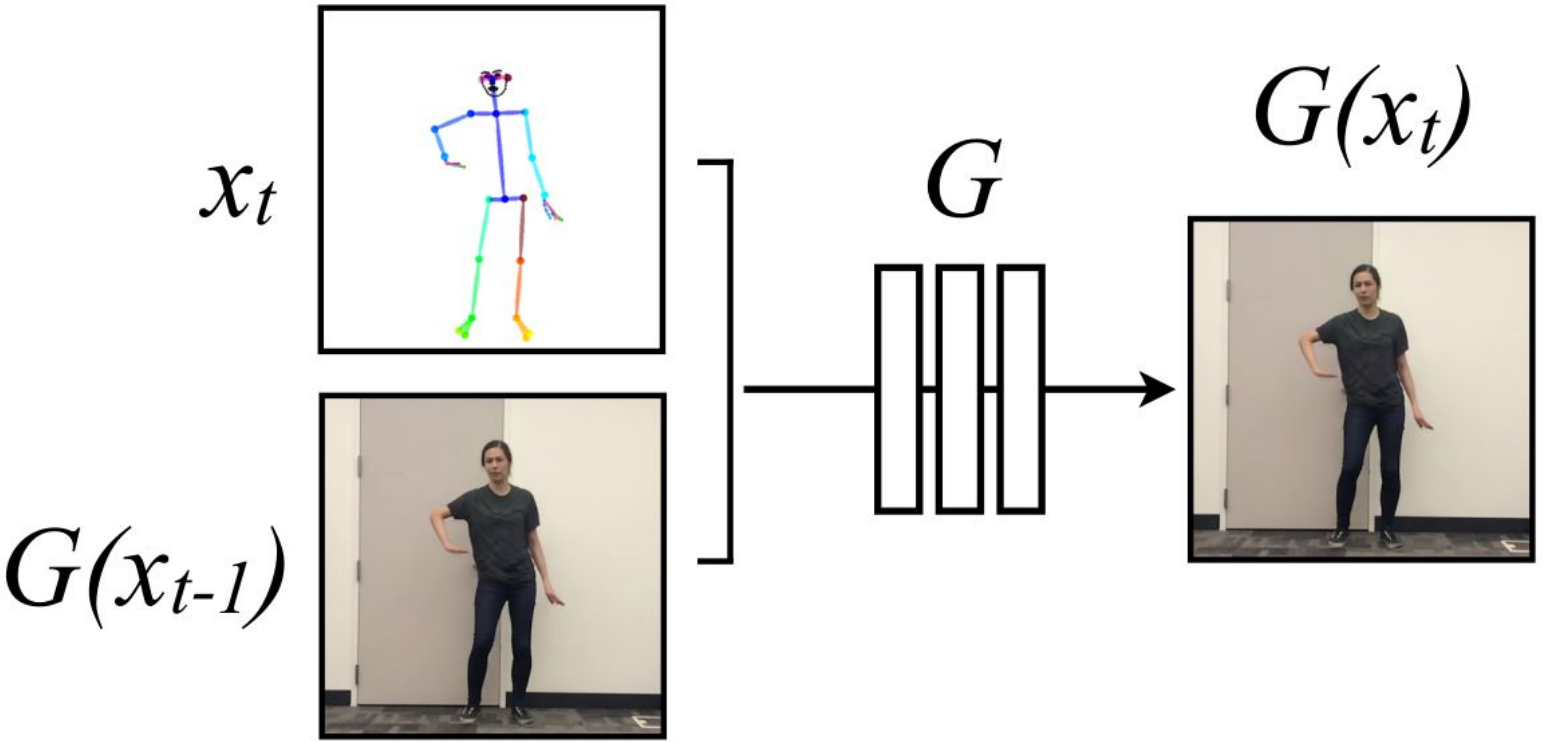
D_k^(i) is ith-layer feature extractor of discriminator D_k

N_i is the number of elements on this layer

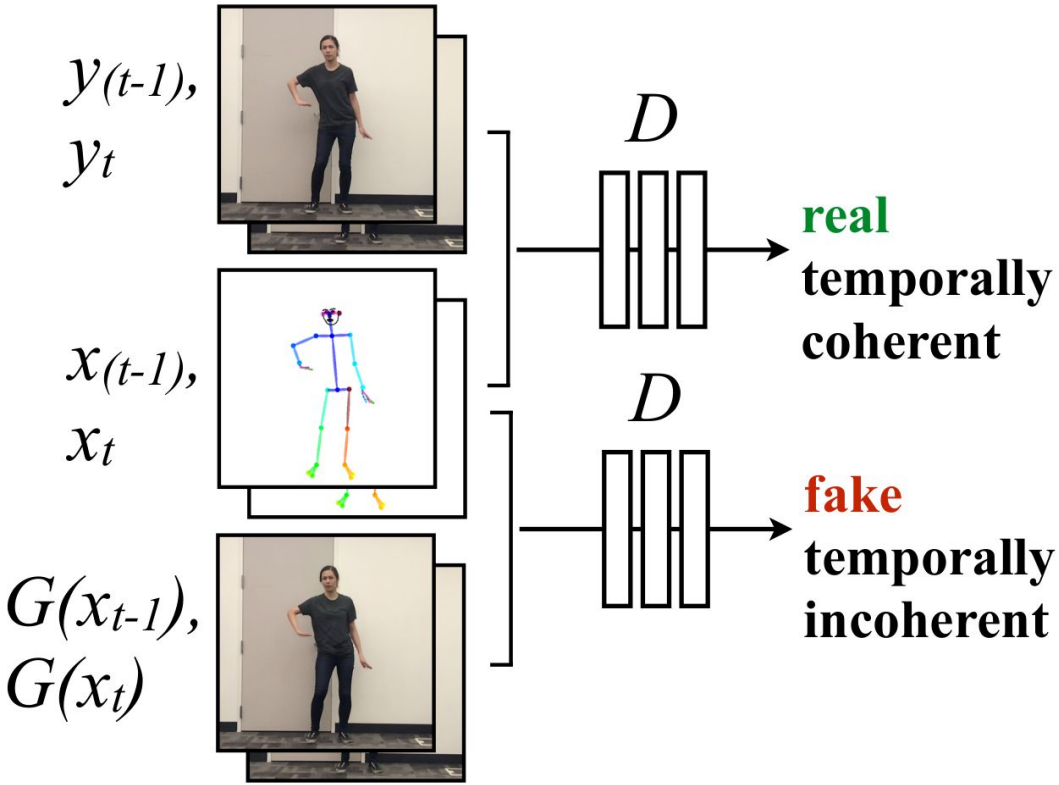
VGG loss

perceptual reconstruction loss which compares pretrained VGGNet features at different layers of the network

Temporal smoothing



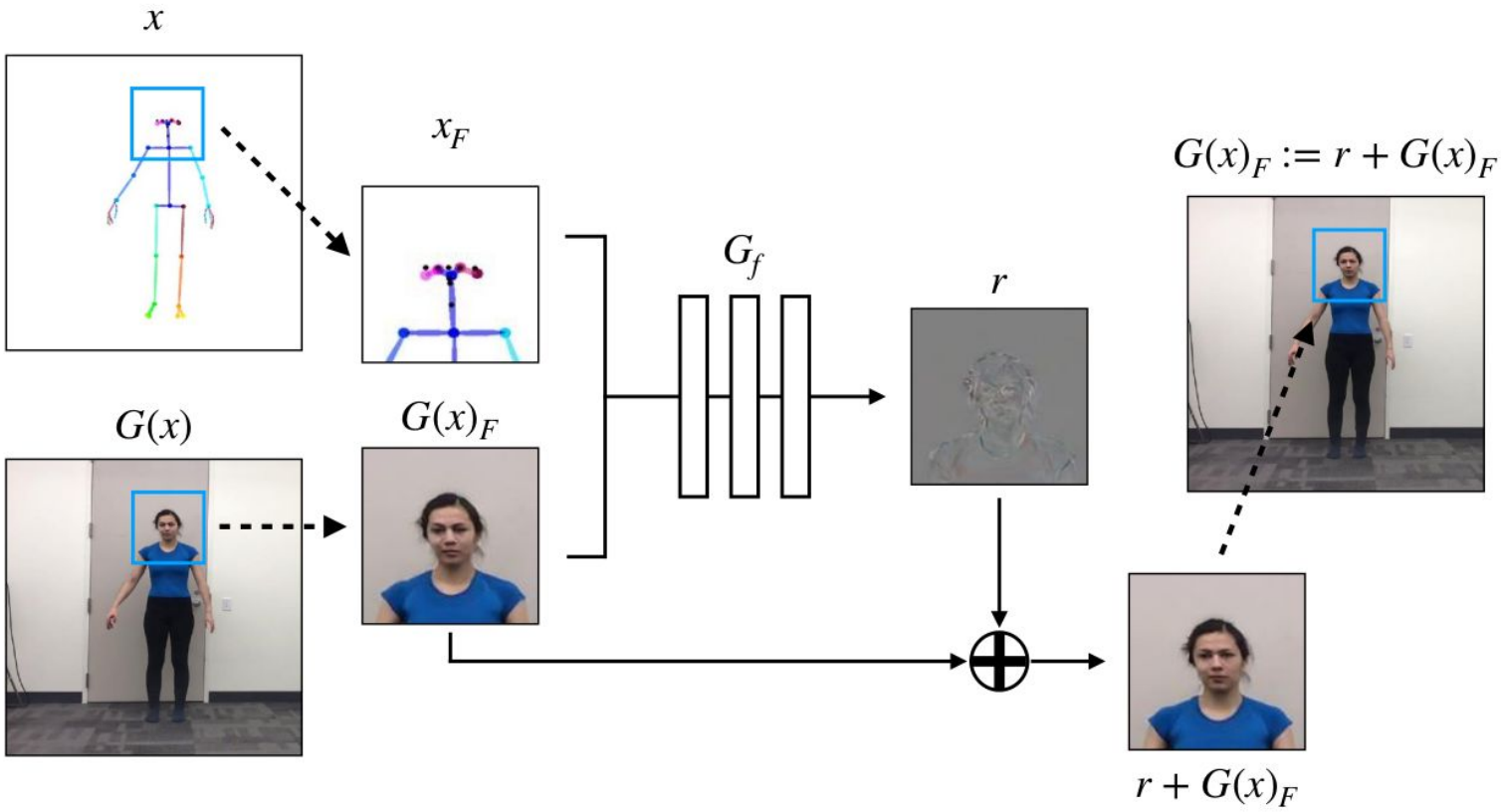
Temporal smoothing



Temporal smoothing

$$\mathcal{L}_{\text{smooth}}(G, D) = \mathbb{E}_{(x,y)}[\log D(x_{t-1}, x_t, y_{t-1}, y_t)]$$
$$+\mathbb{E}_x[\log(1 - D(x_{t-1}, x_t, G(x_{t-1}), G(x_t))]$$

Face GAN



Face GAN

$$\mathcal{L}_{\text{face}}(G_f, D_f) = \mathbb{E}_{(x_F, y_F)}[\log D_f(x_F, y_F)]$$
$$+\mathbb{E}_{x_F}[\log\left(1 - D_f(x_F, G(x)_F + r)\right)].$$

Finally, optimize full image GAN...

$$\min_G \left(\left(\max_{D_1, D_2, D_3} \sum_{k=1,2,3} \mathcal{L}_{\text{smooth}}(G, D_k) \right) + \lambda_{FM} \sum_{k=1,2,3} \mathcal{L}_{\text{FM}}(G, D_k) \right.$$

$$\left. + \lambda_{VGG} \left(\mathcal{L}_{VGG}(G(x_{t-1}), y_{t-1}) + \mathcal{L}_{VGG}(G(x_t), y_t) \right) \right)$$

…and then Face GAN

$$\min_{G_f}\left(\left(\max_{D_f}\mathcal{L}_{\text{face}}(G_f, D_f)\right) + \lambda_{VGG}\mathcal{L}_{VGG}(r + G(x)_F, y_F)\right)$$

Data collection

- target videos: ~20 minutes, 120 fps, tight clothes (for easier video generation)
- source videos: not such strict limitations (only pose detection needed)

Experiments

- SSIM
- LPIPS
- Pose-detector-on-outputs loss

Structural Similarity (SSIM)

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \quad s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \quad c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}$$

where mu and sigma - expectation and variance, sigma_xy - covariance, c_i - some constants, x and y - windows of size NxN

Structural Similarity (SSIM)

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \quad s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \quad c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}$$

where l - luminance, s - structure comparison, c - contrast

Structural Similarity (SSIM)

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1} \quad s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3} \quad c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}$$

$$SSIM(x,y) = [l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot s(x,y)^{\gamma})]$$

Structural Similarity (SSIM)

Let alpha = beta = gamma = 1, then:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Learned Perceptual Image Patch Similarity (LPIPS)

	Patch 0	Patch 1
Humans		✓
L2/PSNR, SSIM, FSIM	✓	
Random Networks	✓	
Unsupervised Networks		✓
Self-Supervised Networks		✓
Supervised Networks		✓

Learned Perceptual Image Patch Similarity (LPIPS)

Mean of euclidean distances between activations of images x and x’ on each layer of some network F

Pose-detector-on-outputs loss

Let coordinates of joints p_k = (x_k, y_k), then loss:

$$d(p, p') = \frac{1}{n} \sum_{k=1}^{n} \| p_k - p'_k \|_2$$

Experiment #1

Loss	SSIM mean	LPIPS mean
pix2pixHD	0.89564	0.03189
T.S.	0.89597	0.03137
T.S. + Face **[Ours]**	**0.89807**	**0.03066**

Table 1. Body output image comparisons - result cropped to bounding box around input pose. For all tables, T.S. denotes a model with our temporal smoothing setup, and T.S. + Face is our full model with both the temporal smoothing setup and Face GAN.

Experiment #2

Loss	SSIM mean	LPIPS mean
pix2pixHD	0.81374	0.03731
T.S.	0.8177	0.03662
T.S. + Face **[Ours]**	**0.83046**	**0.03304**

Table 2. Face output image comparisons - result cropped to bounding box around input face

Experiment #3

Loss	Body (23)	Face (70)	Hands (21)	Overall (135)
pix2pixHD	**2.39352**	1.1872	3.86359	2.0781
T.S.	2.63446	1.14348	3.76056	2.06884
T.S. + Face **[Ours]**	2.56743	**0.91636**	**3.29771**	**1.92704**

Table 3. Mean pose distances, using the pose distance metric described in Section 7. Lower pose distance is more favorable.

Experiment #4

Loss	Body (23)	Face (70)	Hands (21)	Overall (135)
pix2pixHD	0.17864	0.77796	1.67584	2.63244
T.S.	0.15989	0.56318	1.76016	2.48323
T.S. + Face **[Ours]**	**0.15578**	**0.47392**	**1.66366**	**2.29336**

Table 4. Mean number of missed detections per image, fewer missed detections is better.

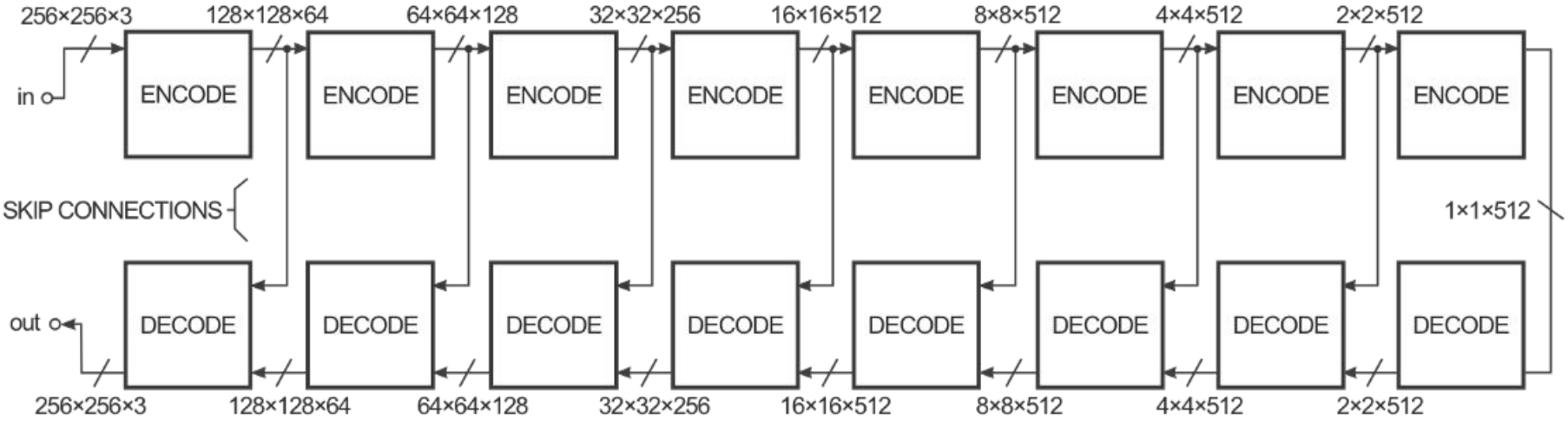
How to improve

- different normalization technique (currently doesn’t care about body differences)
- different GAN losses (many new approaches)

Sources

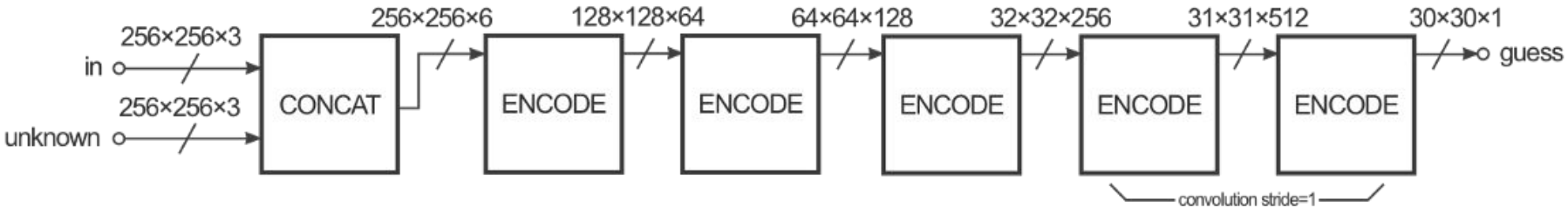
Article: https://arxiv.org/abs/1808.07371

Generator architecture (pix2pix)



(simplified version)

Generator architecture (pix2pix)



result: matrix of 0 and 1 of how “believable” corresponding window of input image