Everybody Dance Now

CAROLINE CHAN, UC Berkeley
SHIRY GINOSAR, UC Berkeley
TINGHUI ZHOU, UC Berkeley
ALEXEI A. EFROS, UC Berkeley

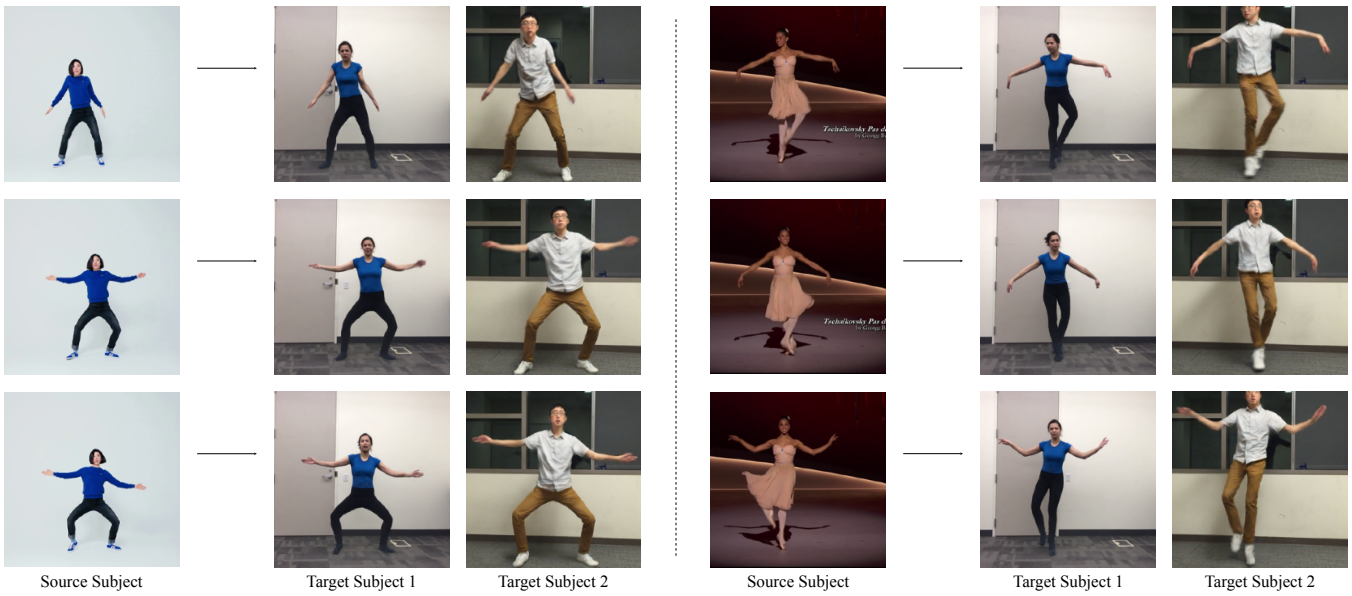

Fig. 1. Motion transfer from a source onto two target subjects.

This paper presents a simple method for "do as I do" motion transfer: given a source video of a person dancing we can transfer that performance to a novel (amateur) target after only a few minutes of the target subject performing standard moves. We pose this problem as a per-frame image-to-image translation with spatio-temporal smoothing. Using pose detections as an intermediate representation between source and target, we learn a mapping from pose images to a target subject's appearance. We adapt this setup for temporally coherent video generation including realistic face synthesis. Our video demo can be found at https://youtu.be/PCBTZh41Ris.

Additional Key Words and Phrases: Motion transfer, Video generation, Generative adversarial networks

1 INTRODUCTION

We propose a method to transfer motion between human subjects in different videos. Given two videos – one of a *target* person whose appearance we wish to synthesize, and the other of a *source* subject whose motion we wish to impose onto our target person – we transfer motion between these subjects via an end to end pixel-based pipeline. This is in contrast to approaches over the last two decades which employ nearest neighbor search [4, 9] or retarget motion in 3D [7, 13, 26, 30]. With our framework, we create a variety of videos, enabling untrained amateurs to spin and twirl like ballerinas, perform martial arts kicks or dance as vibrantly as pop stars.

To transfer motion between two video subjects in a frame-by-frame manner, we must learn a mapping between images of the two individuals. Our goal is therefore to discover an image-to-image translation [14] between the source and target sets. However, we do not have corresponding pairs of images of the two subjects performing the same motions to supervise learning this translation directly. Even if both subjects perform the same routine, it is still unlikely to have an exact frame to frame body-pose correspondence due to body shape and stylistic differences unique to each subject.

We observe that keypoint-based pose, which inherently encodes body position but not appearance, can serve as an intermediate representation between any two subjects. Compatible with our objective, poses preserve motion signatures over time while abstracting away as much subject identity as possible. We therefore design our intermediate representation to be pose stick figures such as in Figure 2. From the target video, we obtain pose detections [5, 27, 35] for each frame yielding a set of *(pose stick figure, target person image)* corresponding pairs. With this aligned data we are able to learn an image-to-image translation model between pose stick figures

Authors' addresses: Caroline Chan, UC Berkeley; Shiry Ginosar, UC Berkeley; Tinghui Zhou, UC Berkeley; Alexei A. Efros, UC Berkeley.

Fig. 2. Correspondence between pose stick figure and target person frame.

and images of our target person in a supervised way. Therefore, our model is trained to produce personalized videos of a specific target subject. Then to transfer motion from source to target, we input the pose stick figures into the trained model to obtain images of the target subject in the same pose as the source. We add two components to improve the quality of our results: To encourage the temporal smoothness of our generated videos, we condition the prediction at each frame on that of the previous time step. To increase facial realism in our results we include a specialized GAN trained to generate the target person's face.

Our method produces videos where motion is transferred between a variety of video subjects without the need for expensive 3D or motion capture data. Our main contributions are a learning-based pipeline for human motion transfer between videos, and the quality of our results which demonstrate complex motion transfer in realistic and detailed videos. We also conduct an ablation study on the components of our model comparing to a baseline framework.

2 RELATED WORK

Over the last two decades there has been extensive study dedicated towards motion transfer or retargeting. Early methods focused on creating new content by manipulating existing video footage [4, 9]. For example, Video Rewrite creates videos of a subject saying a phrase they did not originally utter by finding frames where the mouth position matches the desired speech [4]. Another approach uses optical flow as a descriptor matches different subjects performing similar actions allowing "Do as I do" and "Do as I say" retargeting [9]. Similarly, our approach is designed for video subjects which can be found online or captured in person, although we learn to synthesize novel motions rather than manipulating existing frames.

Other approaches using 3D transfer motion for graphics and animation purposes. Since the retargeting problem was first proposed between animated characters [11], solutions have included the introduction of inverse kinematic solvers to the problem [19] and retargeting between significantly different skeletons [13]. Recently, Villegas et al. [30] apply deep learning techniques to retarget motion without supervised data. Unlike these approaches, our work explores motion transfer between 2D video subjects where there is a lack of 3D information. To mitigate this problem, Cheung et al. [7] propose an elaborate multi-view system to calibrate a personalized kinematic model, obtain 3D joint estimations, and render images of a human subject performing new motions. In contrast, our approach avoids both source-target data calibration and lifting into 3D space.

Recent studies of motion in video have been able to learn to distinguish the movements from appearance and consequently synthesize novel motions in video [1, 29]. MoCoGAN [29] employs unsupervised adversarial training to learn this separation and generates videos of subjects performing novel motions or facial expressions. This theme is continued through subsequent work in Dynamics Transfer GAN [1] which transfers facial expressions from a source subject in a video onto a target person given in a static image. Similarly, we apply our representation of motion (pose stick figures) to different target subjects to generate new motions while in contrast our work specializes on synthesizing detailed dance movements.

Modern approaches have shown success in generating detailed images of human subjects in novel poses [10, 16, 22, 23, 31]. Furthermore, recent methods can synthesize such images for temporally coherent video [2] and future prediction [31]. Frameworks such as Recycle-GAN [3] and vid2vid [32] learn mappings between different videos and demonstrate motion transfer between faces and from poses to body respectively. Our method accounts for both video generation while preserving important details such as facial features.

We are able to learn a mapping from pose to target subject due to advances in image generation and substantial work on general image mapping frameworks. Since the recent emergence of Generative Adversarial Networks (GANs) for approximating generative models [12], GANs have been used for many purposes including image generation [8], especially because they can produce high quality images with sharp details [18]. These advances have led to use of Conditional GANs, in which the generated output is conditioned on a structured input [25]. In addition to specific applications or mappings, studies employed adversarial training to learn arbitrary image to image translations. Over the past few years there have been several frameworks, which often (but not all) use GANS, developed to solve such mappings including pix2pix [14], CoGAN [21], UNIT [20], DiscoGAN [17], CycleGAN [37], Cascaded Refinement Networks [6], and pix2pixHD [33]. Due to our approach toward motion transfer, we are able to choose from and adopt such frameworks for our purposes.

3 METHOD OVERVIEW

Given a video of a source person and another of a target person, our goal is to generate a new video of the target person enacting the same motions as the source. To accomplish this task, we divide our pipeline into three stages – pose detection, global pose normalization, and mapping from normalized pose stick figures to the target subject. In the pose detection stage we use a pretrained state of the art pose detector to create pose stick figures given frames from the source video. The global pose normalization stage accounts for differences between the source and target body shapes and locations within frame. Finally, we design a system to learn the mapping from the normalized pose stick figures to images of the target person with adversarial training.

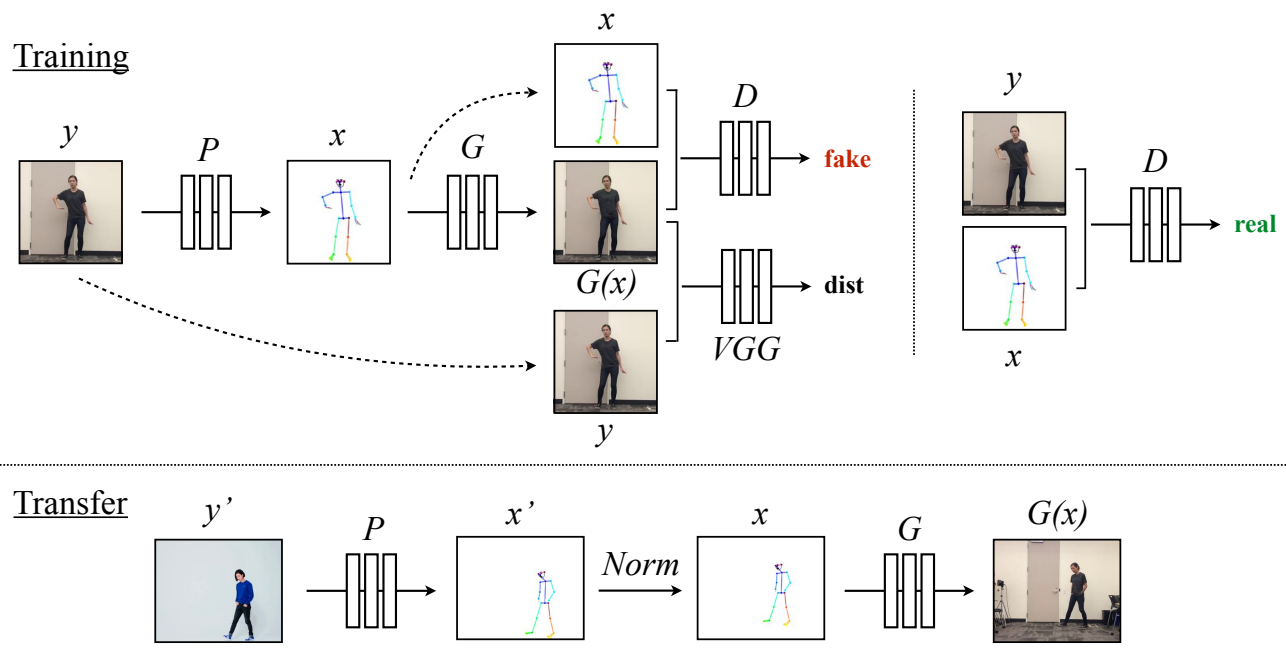

Fig. 3. (Top) **Training**: Our model uses a pose detector P to create pose stick figures from video frames of the target subject. During training we learn the mapping G alongside an adversarial discriminator D which attempts to distinguish between the "real" correspondence pair (x, y) and the "fake" pair $(G(x), y)$. (Bottom) **Transfer**: We use a pose detector $P : Y' \rightarrow X'$ to obtain pose joints for the source person that are transformed by our normalization process $Norm$ into joints for the target person for which pose stick figures are created. Then we apply the trained mapping G.

We now detail our full training system as seen in the **Training** setup of Figure 3. Given frame y from the original target video, we use pose detector P to obtain a corresponding pose stick figure $x = P(y)$. During training, we use corresponding (x, y) pairs to learn a mapping G which synthesizes images of the target person given pose stick x. Through adversarial training with discriminator D and a perceptual reconstruction loss **dist** using a pretrained VGGNet [15, 28], we optimize the generated output $G(x)$ to resemble the ground truth target subject frame y. D attempts to distinguish between "real" image pairs (i.e. (pose stick figure x, ground truth image y)) and "fake" image pairs (i.e. (pose stick figure x, model output $G(x)$).

Our transfer setup is shown in the **Transfer** setup of Figure 3. Similarly to training, pose detector P extracts pose information from source frame y' yielding pose stick figure x'. However, in their video the source subject likely appears bigger, or smaller, and standing in a different position than the subject in the target video. In order for the source pose to better align with the filming setup of the target, we apply a global pose normalization $Norm$ to transform the source's original pose x' to be more consistent with the poses in the target video x. We then pass the normalized pose stick figure x into our trained model G to obtain an image $G(x)$ of our target person which corresponds with the original image of the source y'.

We now describe every component of our system in detail.

4 POSE ESTIMATION AND NORMALIZATION

4.1 Pose estimation

In order to create images which encode body position, we use a pretrained pose detector P [5, 27, 35] which accurately estimates x,y joint coordinates. We draw a representation of the resulting pose stick figure by plotting the keypoints and drawing lines between connected joints as shown in Figure 2. During training, pose stick figures of the target person are inputs to the generator G. For transfer, P obtains joint estimates for the source subject which are then normalized as in Section 4.2 to better match the poses of the transfer subject seen in training. The normalized pose coordinates are used to create input pose stick figures for the generator G.

4.2 Global pose normalization

In different videos, subjects may have different limb proportions or stand closer or farther to the camera than one another. Therefore when transferring motion between two subjects, it may be necessary to transform the pose keypoints of the source person so that they appear in accordance with the target person's body shape and proportion as in the **Transfer** section of Figure 3. We find this transformation by analyzing the heights and ankle positions for poses of each subject and use a linear mapping between the closest and farthest ankle positions in both videos. After gathering these statistics we calculate the scale and translation for each frame based on its corresponding pose detection. Further details of the global pose normalization are described in the appendix in Section 9.

5 ADVERSARIAL TRAINING OF IMAGE TO IMAGE TRANSLATION

We modify the adversarial training setup of pix2pixHD [33] to (1) produce temporally coherent video frames and (2) synthesize realistic face image. We now describe the original objective and our modifications to it in detail.

5.1 pix2pixHD framework

We base our method on the objective presented in pix2pixHD [33]. In the original conditional GAN setup, the generator network G is engaged in a minimax game against multi-scale discriminators $D = (D_1, D_2, D_3)$. The generator's task is to synthesize realistic images in order to fool the discriminator which must discern between "real" (ground truth data) images from the "fake" images produced by the generator. These two networks are trained simultaneously and drive each other to improve, as the generator must learn to synthesize more realistic images to deceive the discriminator which in turn learns differences between generator outputs and ground truth data. The original pix2pixHD objective takes the form -

$$\min_G \Big(\Big(\max_{D_1,D_2,D_3} \sum_{k=1,2,3} \mathcal{L}_{\text{GAN}}(G, D_k) \Big) + \lambda_{FM} \sum_{k=1,2,3} \mathcal{L}_{\text{FM}}(G, D_k) + \lambda_{VGG} \mathcal{L}_{VGG}(G(x), y) \Big) \quad (1)$$

Here, $\mathcal{L}_{\text{GAN}}(G, D)$ is the adversarial loss presented in the original pix2pix paper [14]

$$\mathcal{L}_{\text{GAN}}(G, D) = \mathbb{E}_{(x,y)}[\log D(x, y)] + \mathbb{E}_x[\log(1 - D(x, G(x))] \quad (2)$$

$\mathcal{L}_{\text{FM}}(G, D)$ is the discriminator feature-matching loss presented in pix2pixHD, and $\mathcal{L}_{VGG}(G(x), y)$ is the perceptual reconstruction loss [15] which compares pretrained VGGNet [28] features at different layers of the network.

5.2 Temporal smoothing

To create video sequences, we modify the single image generation setup to enforce temporal coherence between adjacent frames as shown in Figure 4. We investigate the effects of this addition in our ablation study in Section 7.1. Instead of generating individual frames, we predict two consecutive frames where the first output $G(x_{t-1})$ is conditioned on its corresponding pose stick figure x_{t-1} and a zero image z (a placeholder since there is no previously generated frame at time $t-2$). The second output $G(x_t)$ is conditioned on its corresponding pose stick figure x_t and the first output $G(x_{t-1})$. Consequently, the discriminator is now tasked with determining both the difference in realism and temporal coherence between the "fake" sequence $(x_{t-1}, x_t, G(x_{t-1}), G(x_t))$ and "real" sequence $(x_{t-1}, x_t, y_{t-1}, y_t)$. The temporal smoothing changes are now reflected in the updated GAN objective –

$$\mathcal{L}_{\text{smooth}}(G, D) = \mathbb{E}_{(x,y)}[\log D(x_{t-1}, x_t, y_{t-1}, y_t)] + \mathbb{E}_x[\log(1 - D(x_{t-1}, x_t, G(x_{t-1}), G(x_t))] \quad (3)$$

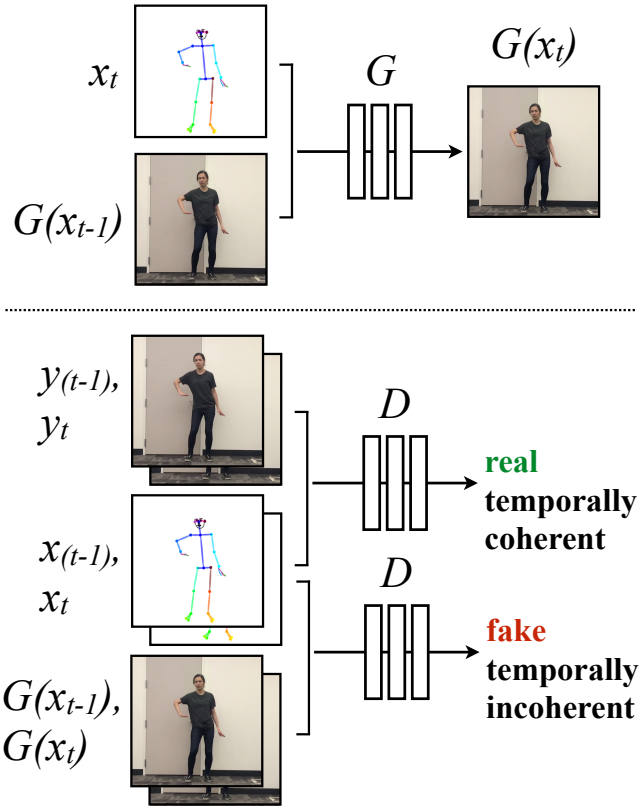

Fig. 4. Temporal smoothing setup. When synthesizing the current frame $G(x_t)$, we condition on its corresponding pose stick figure x_t and the previously synthesized frame $G(x_{t-1})$ to obtain temporally smooth outputs. Discriminator D then attempts differentiate the "real" temporally coherent sequence $(x_{t-1}, x_t, y_{t-1}, y_t)$ from the "fake" sequence $(x_{t-1}, x_t, G(x_{t-1}), G(x_t))$.

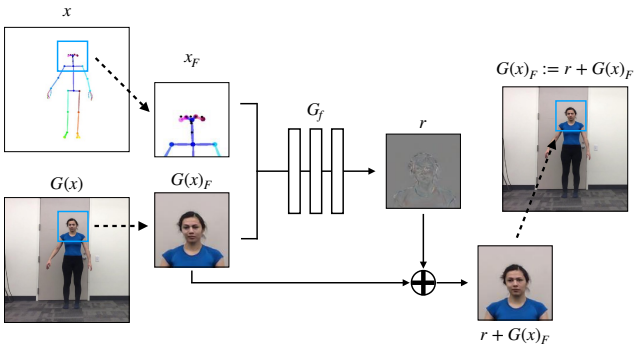

Fig. 5. Face GAN setup. Residual is predicted by generator G_f and added to the original face prediction from the main generator.

5.3 Face GAN

We add a specialized GAN setup designed to add more detail and realism to the face region as shown in Figure 5. We show that our face GAN produces convincing facial features and improves upon

the results of the full image GAN in our ablation studies detailed in Section 7.1.

After generating the full image of the scene with the main generator G, we input a smaller section of the image centered around the face $G(x)_F$ and the input pose stick figure sectioned in the same fashion x_F to another generator G_f which outputs a residual $r = G_f(x_F, G(x)_F)$. The final output is the addition of the residual with original face region $r + G(x)_F$ and this change is reflected in the relevant region of the full image. A discriminator D_f then attempts to discern the "real" face pairs (x_F, y_F) (face region of the input pose stick figure, face region of the ground truth target person image) from the "fake" face pairs $(x_F, r + G(x)_F)$ similarly to the original pix2pix objective -

$$\mathcal{L}_{\text{face}}(G_f, D_f) = \mathbb{E}_{(x_F, y_F)}[\log D_f(x_F, y_F)] + \mathbb{E}_{x_F}[\log\left(1 - D_f(x_F, G(x)_F + r)\right)]. \quad (4)$$

where x_F is the face region of the original pose stick figure x, y_F is the face region of ground truth target person image y. Similarly to the full image, we add a perceptual reconstruction loss on comparing the final face $r + G(x)_F$ to the ground truth target person's face y_F.

5.4 Full Objective

We employ training in stages where the full image GAN is optimized separately from the specialized face GAN. First we train the main generator and discriminator (G, D) during which the full objective is -

$$\min_G \left(\left(\max_{D_1, D_2, D_3} \sum_{k=1,2,3} \mathcal{L}_{\text{smooth}}(G, D_k) \right) + \lambda_{FM} \sum_{k=1,2,3} \mathcal{L}_{\text{FM}}(G, D_k) + \lambda_{VGG} \left(\mathcal{L}_{VGG}(G(x_{t-1}), y_{t-1}) + \mathcal{L}_{VGG}(G(x_t), y_t) \right) \right) \quad (5)$$

.

After this stage, the full image generator and discriminator weights are frozen and we optimize the face GAN with full objective

$$\min_{G_f} \left(\left(\max_{D_f} \mathcal{L}_{\text{face}}(G_f, D_f) \right) + \lambda_{VGG} \mathcal{L}_{VGG}(r + G(x)_F, y_F) \right) \quad (6)$$

.

6 IMPLEMENTATION

6.1 Data Collection

We collect source and target videos in slightly different manners. To learn the appearance of the target subject in many poses, it is important that the target video captures a sufficient range of motion and sharp frames with minimal blur. To ensure the quality of the frames, we filmed our target subject for around 20 minutes of real time footage at 120 frames per second which is possible with some modern cell phone cameras. Since our pose representation does not encode information about clothes, we had our target subjects wear tight clothing with minimal wrinkling.

In contrast to some of the preparation required for filming a target subject, source videos do not require the same (albeit still reasonable) quality as we only need decent pose detections from the source video. Without such limitations, many high quality videos of a subject performing a dance are abundant online.

We found pre-smoothing pose keypoints to be immensely helpful in reducing jittering in our outputs. For videos with a high framerate (120 fps), we gaussian smooth the keypoints over time, and we use median smoothing for videos with lower framerates.

6.2 Network architecture

We adapt architectures from various models for different stages of our pipeline. To extract pose keypoints for the body, face, and hands we use architectures provided by a state of the art pose detector OpenPose [5, 27, 35].

For the image translation stage of our pipeline, we adapt the architectures proposed by Wang et al. in the pix2pixHD model [33]. To create $128x128$ face image residuals, we do not need the full capability of the entire pix2pixHD generator and therefore we predict face residuals using the global generator of pix2pixHD. Similarly, we use a single $70x70$ Patch-GAN discriminator [14] for the face discriminator. In practice we use the LSGAN [24] objective during training similarly to pix2pixHD for both the full image and face GANs.

7 EXPERIMENTS

We explore the effects of our modifications to the pix2pixHD baseline and evaluate the quality of our results on our own dataset collected as in Section 6.1. Since we do not have ground truth data for retargeting between two different video subjects, we analyze the reconstruction of our target person (i.e. the source person is the target person) with validation data. We conduct an ablation study on the inclusion of our temporal smoothing setup and face GAN compared to a pix2pixHD baseline.

To assess the quality of our individual frames, we measure both Structural Similarity (SSIM) [34] and Learned Perceptual Image Patch Similarity (LPIPS) [36]. Since we do not have ground truth flows for our data, we rely on qualitative analysis to evaluate the temporal coherence of our output videos.

In addition, we run the pose detector P on the outputs of each system, and compare these reconstructed keypoints to the pose detections of the original input video. If all body parts are synthesized correctly, then the reconstructed pose should be close to the input pose on which the output was conditioned. Therefore, we can evaluate these pose reconstructions to analyze the quality of our results.

For a pose distance metric between two poses p, p' each with n joints $p_1, ..., p_n$ and $p'_1, ..., p'_n$, we sum the L2 distances between the corresponding joints $p_k = (x_k, y_k)$ and $p'_k = (x'_k, y'_k)$ normalized by the number of keypoints

$$d(p, p') = \frac{1}{n} \sum_{k=1}^{n} \|p_k - p'_k\|_2 \quad (7)$$

To avoid dealing with missing detections (i.e. without viewing the original image of the subject it can be hard to discern whether a "missed" detection is due to noise or occlusion), we only compare poses where all joints are detected.

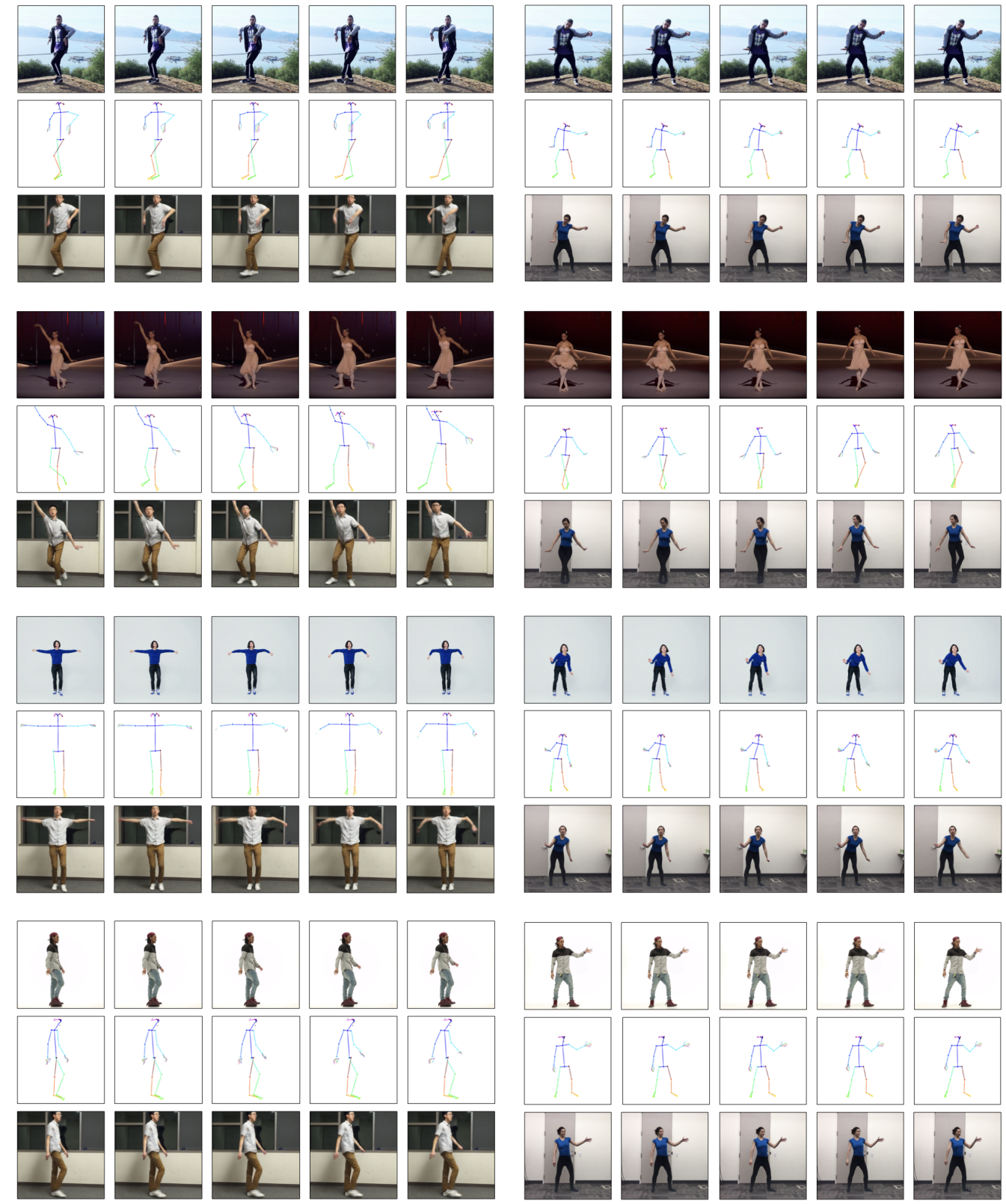

Fig. 6. Transfer results. In each section we show five consecutive frames. The top row shows the source subject, the middle row shows the normalized pose stick figures, and the bottom row shows the model outputs of the target person.

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.

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

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.

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.

7.1 Ablation Study

The results of our ablation study are presented in Tables 1 to 4. We compare results from the pix2pixHD baseline (pix2pixHD), a version of our model with just the temporal smoothing setup (T.S.), and our full model with both the temporal smoothing setup and the face GAN (T.S. + Face).

Table 1 contains mean image similarity measurements for a region around the body. Some example images are in Figure 7. Both SSIM and LPIPS scores are similar for all model variations. Qualitatively, the pix2pixHD baseline already reasonably synthesizes the target person as reflected by the similarity measurements. Scores on full images are even more similar between our ablations, as all ablations have no difficulty generating the static background. Table 2 shows mean scores for the face region (for the face GAN, this is the region for which the face residual is generated). Again, scores are generally favorable for all ablations, although the full model with both the temporal smoothing and face GAN setups obtains the best scores with the biggest discrepancy in the face region.

Table 3 shows the mean pose distance using the method described in Equation 7 for each ablation. We run the pose metric on particular sets of keypoints (body, face, hands) to determine the regions which incur the most error. Adding the temporal smoothing setup does not seem to decrease the reconstructed pose distances significantly, however including the face GAN adds substantial improvements overall, especially for the face and hand keypoints.

In Table 4 we count the number of missed detections (i.e. joints detected on ground truth frames but not on outputs) on various regions and the whole pose as the pose metric does not accurately depict missed detections. With the addition of our model parts, the number of missed detections generally decreases, especially for the face keypoints.

7.2 Qualitative Assessment

Although the ablation study scores for the temporal smoothing setup are generally comparable or an improvement to the pix2pixHD baseline, significant differences occur in video results where the temporal smoothing setup exhibits more frame to frame coherence than the pix2pixHD baseline. Qualitatively, the temporal smoothing setup helps with smooth motion, color consistency across frames, and also in individual frame synthesis.

Consistent with the ablation study, we find that adding a specialized facial generator and discriminator adds considerable detail and encourages synthesizing realistic body parts. We compare the face synthesis with and without the face GAN in Figure 8 and in our video results.

8 DISCUSSION

Overall our model is able to create reasonable and arbitrarily long videos of a target person dancing given body movements to follow through an input video of another subject dancing. Although our setup can produce plausible results in many cases, occasionally our results suffer from several issues.

Fundamentally, our input pose stick figures depend on noisy pose estimations which do not carry temporal information from frame to frame. Missing or incorrect keypoint locations from pose detection injects error into our inputs and these failures often carry over into our results, even though we attempt to mitigate these limitations through our temporal smoothing setup. Even though we try to inject temporal coherence through our setup and presmoothing keypoints, our results often still suffer from jittering. Errors occur particularly in transfer videos when the input motion or motion speed is different from the movements seen at training time. However, even when the target subject attempts to copy a dance from a source subject in the training sequence, our results still experience some jittering and shakiness when the motion from the source is transferred onto the target. Since normalized poses for transfer are often similar to those seen in training, we attribute this observation to the underlying difference between how our target and transfer subjects move given their unique body structure. In this way, we believe that motion is tied to identity which is still present in the pose detections.

Although our method for global pose normalization reasonably resizes the movements of any source subject to match the scale and location of the target person seen in training, our simple scale-and-translate solution does not account for different limb lengths and camera positions or angles. These discrepancies also contribute to a wider gap between the motion seen in training and at test time.

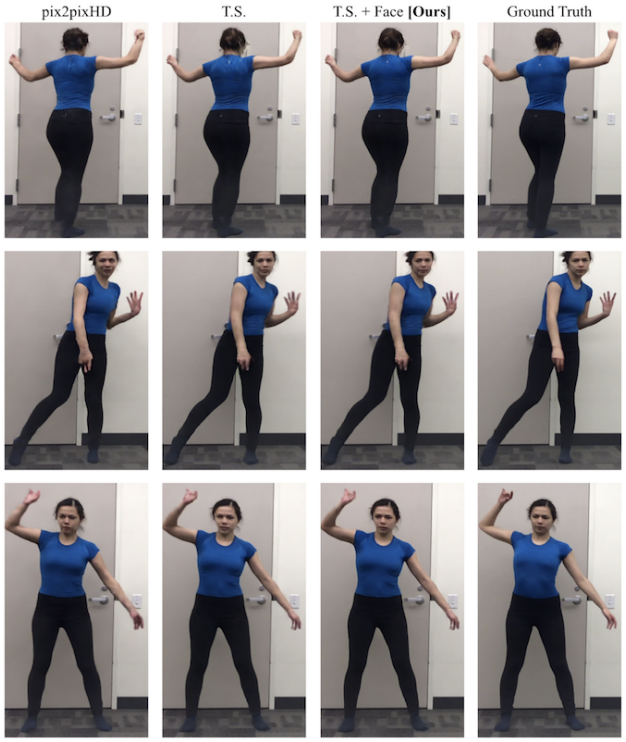

Fig. 7. Comparison of synthesis results for different models. Images have been cropped to a bounding box around the original pose. T.S. denotes a model with our temporal smoothing setup, and T.S. + Face is our full model with both the temporal smoothing setup and Face GAN. The temporal smoothing setup adds details to the hands, head, shirt, and shadows. These details are carried over in the full model includes additional detail to the face and surrounding area resulting in the most realistic synthesis.

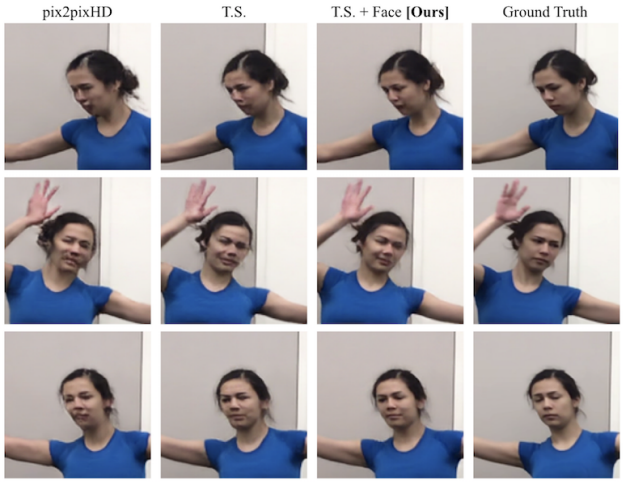

Fig. 8. Face image comparison from different models on the validation set. T.S. denotes a model with our temporal smoothing setup, and T.S. + Face is our full model with both the temporal smoothing setup and Face GAN. Details improve and distortions decrease upon the additions of the temporal smoothing setup and the face GAN.

Additionally, the 2D coordinates and missing detections constrict the number of ways we are able to retarget motion between subjects, which often work in 3D with perfect joint locations and temporally coherent motions.

To address these issues, more work is needed on temporally coherent video generation and on representations of human motion. Although overall the pose stick figures yielded convincing results, we would like to avoid the restrictions it presents by instead using temporally coherent inputs and representation specifically optimized for motion transfer in future work. Despite these challenges, our method is able to produce compelling videos given a variety of inputs.

Acknowledgements: This work was supported, in part, by NSF grant IIS-1633310 and research gifts from Adobe, eBay, and Google.

REFERENCES

[1] Wissam J Baddar, Geonmo Gu, Sangmin Lee, and Yong Man Ro. 2017. Dynamics Transfer GAN: Generating Video by Transferring Arbitrary Temporal Dynamics from a Source Video to a Single Target Image. *arXiv preprint arXiv:1712.03534* (2017).

[2] Guha Balakrishnan, Amy Zhao, Adrian V Dalca, Fredo Durand, and John Guttag. 2018. Synthesizing Images of Humans in Unseen Poses. *arXiv preprint arXiv:1804.07739* (2018).

[3] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. 2018. Recycle-GAN: Unsupervised Video Retargeting. In *ECCV*.

[4] Christoph Bregler, Michele Covell, and Malcolm Slaney. 1997. Video rewrite: Driving visual speech with audio. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 353–360.

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *CVPR*.

[6] Qifeng Chen and Vladlen Koltun. 2017. Photographic image synthesis with cascaded refinement networks. In *IEEE International Conference on Computer Vision (ICCV)*, Vol. 1. 3.

[7] German KM Cheung, Simon Baker, Jessica Hodgins, and Takeo Kanade. 2004. Markerless human motion transfer. In *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*. IEEE, 373–378.

[8] Emily L Denton, Soumith Chintala, Rob Fergus, et al. 2015. Deep Generative Image Models using aï£ij Laplacian Pyramid of Adversarial Networks. In *Advances in neural information processing systems*. 1486–1494.

[9] Alexei A. Efros, Alexander C. Berg, Greg Mori, and Jitendra Malik. 2003. Recognizing Action at a Distance. In *IEEE International Conference on Computer Vision*. Nice, France, 726–733.

[10] Patrick Esser, Ekaterina Sutter, and Björn Ommer. 2018. A Variational U-Net for Conditional Appearance and Shape Generation. (2018).

[11] Michael Gleicher. 1998. Retargetting motion to new characters. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 33–42.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[13] Chris Hecker, Bernd Raabe, Ryan W Enslow, John DeWeese, Jordan Maynard, and Kees van Prooijen. 2008. Real-time motion retargeting to highly varied user-created morphologies. In *ACM Transactions on Graphics (TOG)*, Vol. 27. ACM, 27.

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2016. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004* (2016).

[15] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*.

[16] Donggyu Joo, Doyeon Kim, and Junmo Kim. 2018. Generating a Fusion Image: One's Identity and Another's Shape. *arXiv preprint arXiv:1804.07455* (2018).

[17] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. 2017. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192* (2017).
[18] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2016. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802* (2016).
[19] Jehee Lee and Sung Yong Shin. 1999. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 39–48.
[20] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. 2017. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*. 700–708.
[21] Ming-Yu Liu and Oncel Tuzel. 2016. Coupled generative adversarial networks. In *Advances in neural information processing systems*. 469–477.
[22] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. 2017. Pose Guided Person Image Generation. *arXiv preprint arXiv:1705.09368* (2017).
[23] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. 2018. Disentangled person image generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 99–108.
[24] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least squares generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2813–2821.
[25] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).
[26] Ju Shen and Jianjun Yang. 2015. Automatic pose tracking and motion transfer to arbitrary 3d characters. In *International Conference on Image and Graphics*. Springer, 640–653.
[27] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. 2017. Hand Keypoint Detection in Single Images using Multiview Bootstrapping. In *CVPR*.
[28] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[29] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. 2018. Mocogan: Decomposing motion and content for video generation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018).
[30] Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. 2018. Neural Kinematic Networks for Unsupervised Motion Retargetting. *arXiv preprint arXiv:1804.05653* (2018).
[31] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. 2017. Learning to Generate Long-term Future via Hierarchical Prediction. *arXiv preprint arXiv:1704.05831* (2017).
[32] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. Video-to-Video Synthesis. *arXiv* (2018).
[33] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2017. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. *arXiv preprint arXiv:1711.11585* (2017).
[34] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.
[35] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional pose machines. In *CVPR*.
[36] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*.
[37] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *arXiv preprint arXiv:1703.10593* (2017).

9 APPENDIX

9.1 Global Pose Normalization Details

To find a transformation in terms of scale and translation between a source pose and a target pose, we find the minimum and maximum ankle positions in image coordinates of each subject while they are on the ground (i.e. feet raised in the air are not considered). These coordinates represent the farthest and closest distances to the camera respectively). The maximum ankle position is the y foot coordinate closest to the bottom of the image. The minimum foot position is found by clustering the y foot coordinates which are less than (or spatially above) the median ankle position and about the same distance as the maximum ankle position's distance to the median ankle position. The clustering is as described by this set

$$\{t : ||t - med| - \alpha * |max - med|| < \epsilon\} \cap \{t < med\} \tag{8}$$

where med is the median foot position, max is the maximum ankle position, and ϵ is a scalar. In practice we use $\epsilon = 0.7$ (although this scalar depends on the camera height) and take the maximum of this set to obtain the minimum foot position.

Once the minimum and maximum ankle positions of each subject are found, we carry out a linear mapping between the minimum and maximum ankle positions of each video (i.e. minimum of source mapped to minimum of target, and same for the maximum ankle positions). We characterize our transformation in terms of scale and translation in the y direction, which is calculated for each frame.

The translation is calculated according to the average of the left and right ankle y coordinates and its distance between the maximum and minimum ankle positions in the source frame. Then the new transformed food position is the coordinate between the maximum and minimum ankle position in the target video with the same relative/interpolated distance. Given an average ankle position a_{source} in the a source frame, the translation b is calculated for that frame according to the following equation -

$$b = t_{min} + \frac{a_{source} - s_{min}}{s_{max} - s_{min}}(t_{max} - t_{min}) - f_{source} \tag{9}$$

where t_{min} and t_{max} are the minimum and maximum ankle positions in the target video, and s_{min} and s_{max} are the minimum and maximum ankle positions in the source video.

To calculate the scale, we cluster the heights around the minimum ankle position and the maximum ankle position and find the maximum height for each cluster for each video. Call these maximum heights t_{close} for the maximum of the cluster near the target person's maximum ankle position, t_{far} for the maximum of the cluster near the target person's minimum ankle position, and s_{close} and s_{far} respectively. We obtain the close ratio by taking the ratio between the target's close height and the source's close height, and similarly for the far ratio. Given average ankle position a_{source}, the scale for this frame is interpolated between these two ratios in the same way as the translation is interpolated as described in the following equation -

$$scale = \frac{t_{far}}{s_{far}} + \frac{a_{source} - s_{min}}{s_{max} - s_{min}}(\frac{t_{close}}{s_{close}} - \frac{t_{far}}{s_{far}}) \tag{10}$$