

# Everybody Dance Now

Caroline Chan\*

Shiry Ginosar

Tinghui Zhou†

Alexei A. Efros

UC Berkeley



Figure 1: **“Do as I Do” motion transfer:** given a YouTube clip of a ballerina (top), and a video of a graduate student performing various motions, our method transfers the ballerina’s performance onto the student (bottom). Video: <https://youtu.be/mSaIrz8lM1U>

## Abstract

*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 approach this problem as video-to-video translation using pose as an intermediate representation. To transfer the motion, we extract poses from the source subject and apply the learned pose-to-appearance mapping to generate the target subject. We predict two consecutive frames for temporally coherent video results and introduce a separate pipeline for realistic face synthesis. Although our method is quite simple, it produces surprisingly compelling results (see video). This motivates us to also provide a forensics tool for reliable synthetic content detection, which is able to distinguish videos synthesized by our system from real data. In addition, we release a first-of-its-kind open-source dataset of videos that can be legally used for training and motion transfer.*

## 1. Introduction

Consider the two video sequences on Figure 1. The top row is the input – it is a YouTube clip of a ballerina (the *source* subject) performing a sequence of motions. The bottom row is the output of our algorithm. It corresponds to frames of a different person (the *target* subject) apparently performing the same motions. The twist is that the target person never performed the same exact sequence of motions as the source, and, indeed, knows nothing about ballet. He was instead filmed performing a set of standard moves, without specific reference to the precise actions of the source. And, as is obvious from the figure, the source and the target are of different genders, have different builds, and wear different clothing.

In this work, we propose a simple but surprisingly effective approach for “Do as I Do” video retargeting – automatically transferring the motion from a source to a target subject. 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 by learning a simple video-to-video translation. With our framework, we create a variety of videos, enabling un-

\*C. Chan is currently a graduate student at MIT CSAIL.

†T. Zhou is currently affiliated with Humen, Inc.

trained 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 [16] 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. Even if both subjects perform the same routine, it is still unlikely to have an exact frame to frame pose correspondence due to body shape and motion style unique to each subject.

We observe that keypoint-based pose preserves motion signatures over time while abstracting away as much subject identity as possible and can serve as an intermediate representation between any two subjects. We therefore use pose stick figures obtained from off-the-shelf human pose detectors, such as OpenPose [6, 34, 43], as an intermediate representation for frame-to-frame transfer, as shown in Figure 2. We then learn an image-to-image translation model between pose stick figures and images of our target person. To transfer motion from source to target, we input the pose stick figures from the source into the trained model to obtain images of the target subject in the same pose as the source.

The central contribution of our work is a surprisingly simple method for generating compelling results on human motion transfer. We demonstrate complex motion transfer from realistic in-the-wild input videos and synthesize high-quality and detailed outputs (see Section 4.3 and our video for examples). Motivated by the high quality of our results, we introduce an application for detecting if a video is real or synthesized by our method. We strongly believe that it is important for work in image synthesis to explicitly address the issue of fake detection (Section 5).

Furthermore, we release a two-part dataset: First, five long single-dancer videos which we filmed ourselves that can be used to train and evaluate our model, and second, a large collection of short YouTube videos that can be used for transfer and fake detection. We specifically designate the single-dancer data to be high-resolution open-source data for training motion transfer and video generation methods. The subjects whose data we release have all consented to allowing the data to be used for research purposes. For more details, see our project website [https://carolineec.github.io/everybody\\_dance\\_now](https://carolineec.github.io/everybody_dance_now).

## 2. Related Work

Over the last two decades there has been extensive work dedicated to motion transfer. Early methods focused on creating new content by manipulating existing video footage [5, 12, 31]. For example, Video Rewrite [5] creates videos of a subject saying a phrase they did not originally

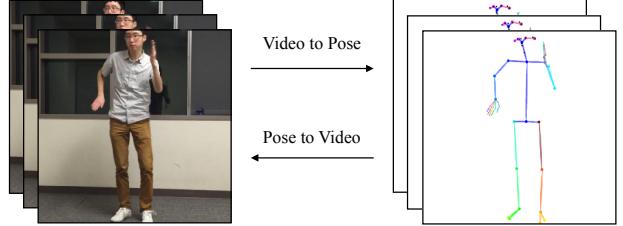


Figure 2: Our method creates correspondences by detecting poses in video frames (Video to Pose) and then learns to generate images of the target subject from the estimated pose (Pose to Video).

utter by finding frames where the mouth position matches the desired speech. Efros et al. [12] use optical flow as a descriptor to match different subjects performing similar actions allowing “Do as I do” and “Do as I say” retargeting. Classic computer graphics approaches to motion transfer attempt to perform this in 3D. Ever since the retargeting problem was proposed between animated characters [14], solutions have included the use of inverse kinematic solvers [23] and retargeting between significantly different 3D skeletons [15]. Our approach is similarly designed for in-the-wild video subjects, although we learn to synthesize novel motions rather than manipulating existing frames and we use 2D representations.

Several approaches rely on calibrated multi-camera setups to ‘scan’ a target actor and manipulate their motions in a new video through a fitted 3D model of the target. To obtain 3D information, Cheung et al. [9] 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. Xu et al. [45] use multi-view captures of a target subject performing simple motions to create a database of images and transfer motion through a fitted 3D skeleton and corresponding surface mesh for the target. Work by Casas et al. use 4D Video Textures [7] to compactly store a layered texture representation of a scanned target person and use their temporally coherent mesh and data representation to render video of the target subject performing novel motions. In contrast, our approach explores motion transfer between 2D video subjects and avoid data calibration and lifting into 3D space.

Similarly to our method, recent works have applied deep learning for reanimation in different applications and rely on more detailed input representations. Given synthetic renderings, an interior face model, and a gaze map as input, Kim et al. [19] transfer head position and facial expressions between human subjects and render their results in detailed portrait videos. Our problem is analogous to this work except we retarget full body motion, and the inputs to our model as 2D pose stick figures as opposed to more detailed 3D representations. Similarly, Martin-Brualla et al. [29] apply neural re-rendering to enhance rendering of human motion capture for VR/AR purposes. The primary

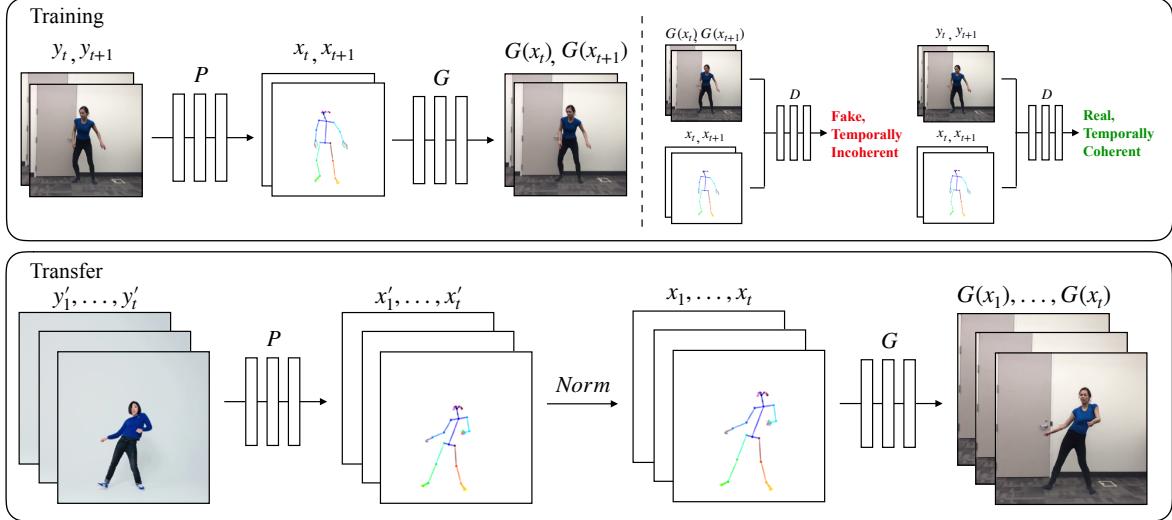


Figure 3: (Top) **Training**: Our model uses a pose detector  $P$  to create pose stick figures from video frames of the target subject. We learn the mapping  $G$  alongside an adversarial discriminator  $D$  which attempts to distinguish between the “real” correspondences  $(x_t, x_{t+1}), (y_t, y_{t+1})$  and the “fake” sequence  $(x_t, x_{t+1}), (G(x_t), G(x_{t+1}))$ . (Bottom) **Transfer**: We use a pose detector  $P$  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  $G$ .

focus of this work is to render realistic humans in real time and similarly uses a deep network to synthesize their final result, but unlike our work does not address motion transfer between subjects. Villegas et al. [37] focus on retargeting motion between rigged skeletons and demonstrate reanimation in 3D characters without supervised data. Similarly, we learn to retarget motion using a skeleton-like intermediate representation, however we transfer full body motion between human subjects who are not rigged to the skeleton unlike animated characters.

Recent methods focus on disentangling motion from appearance and synthesizing videos with novel motion [36, 2]. MoCoGAN [36] employs unsupervised adversarial training to learn this separation and generates videos of subjects performing novel motions or facial expressions. This theme is continued in Dynamics Transfer GAN [2] 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 to different target subjects to generate new motions. However, in contrast to these methods we specialize on synthesizing detailed dance videos.

Modern approaches have shown success in generating detailed single images of human subjects in new poses [3, 10, 11, 18, 22, 27, 28, 33, 38, 13, 46]. Works including Ma et al. [27, 28] and Siarohin et al. [33] have introduced novel architectures and losses for this purpose. Furthermore, [39, 38] have shown pose is an effective supervisory signal for future prediction and video generation. However these works are not designed specifically for motion transfer. Rather than generating possible views of a previously unseen person from a single input image, we are interested

in learning the style of a single, known person from large amounts of personalized video data and synthesizing them dancing in a detailed high-resolution video.

Concurrent with our work, [1, 4, 24, 40] learn mappings between videos and demonstrate motion transfer between faces and from poses to body. Wang et al. [40] achieves results of similar quality to ours with a more complex method and significantly more computational resources.

Our work is made possible by recent rapid advances along two separate directions: robust pose estimation, and realistic image-to-image translation. Modern pose detection systems including OpenPose [6, 34, 43] and DensePose [32] allow for surprisingly reliable and fast pose extraction in a variety of scenarios. At the same time, the recent emergence of image-to-image translation models, pix2pix [16], CoGAN [26], UNIT [25], CycleGAN [48], DiscoGAN [20], Cascaded Refinement Networks [8], and pix2pixHD [41], have enabled high-quality single-image generation. We build upon these two building blocks by using pose detection as an intermediate representation and extending upon single-image generation to synthesize temporally-coherent, surprisingly realistic videos.

### 3. Method

Given a video of a source person and another of a target person, our goal is to generate a new video of the target 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. See Figure 3 for

an overview of our pipeline. In the pose detection stage we use a pre-trained 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 the frame. Finally, we design a system to learn the mapping from the pose stick figures to images of the target person using adversarial training. Next we describe each stage of our system.

### 3.1. Pose Encoding and Normalization

**Encoding body poses** To encode the body pose of a subject image, we use a pre-trained pose detector  $P$  (Open-Pose [6, 34, 43]) which accurately estimates 2D  $x, y$  joint coordinates. We then create a colored pose stick figure by plotting the keypoints and drawing lines between connected joints as shown in Figure 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 re-targeting 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 location as in the **Transfer** section of Figure 3. We find this transformation by analyzing the heights and ankle positions for the poses of each subject and use a linear mapping between the closest and farthest ankle positions in both videos. After gathering these positions, we calculate the scale and translation for each frame based on its corresponding pose detection. Details of this process are described in Section 8.5.

### 3.2. Pose to Video Translation

Our video synthesis method is based off of an adversarial single frame generation process presented by Wang et al. [41]. In the original conditional GAN setup, the generator network  $G$  engages in a minimax game against multi-scale discriminator  $D = (D_1, D_2, D_3)$ . The generator must synthesize images in order to fool the discriminator which must discern between “real” (ground truth) images and “fake” images produced by the generator. The two networks are trained simultaneously and drive each other to improve -  $G$  learns to synthesize more detailed images to deceive  $D$  which in turn learns differences between generated outputs and ground truth data. For our purposes,  $G$  synthesizes images of a person given a pose stick figure.

Such single-frame image-to-image translation methods are not suitable for video synthesis as they produce temporal artifacts and cannot generate the fine details important in perceiving humans in motion. We therefore add a learned model of temporal coherence as well as a module for high resolution face generation.

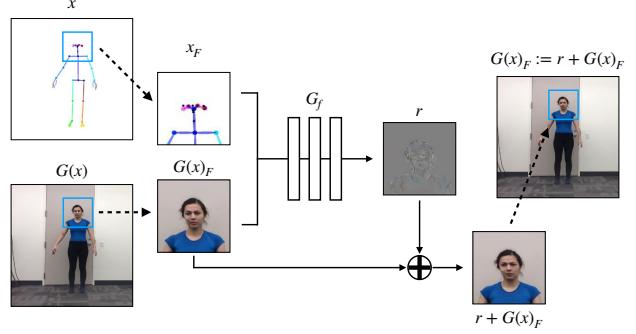


Figure 4: Face GAN setup. Residual is predicted by generator  $G_f$  and added to the original face prediction from the main generator.

**Temporal smoothing** To create video sequences, we modify the single image generation setup to enforce temporal coherence between adjacent frames as shown in Figure 3 (top right). 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

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

**Face GAN** We add a specialized GAN setup to add more detail and realism to the face region as shown in Figure 4. 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 (i.e.  $128 \times 128$  patch centered around the nose keypoint),  $G(x)_F$ , 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 synthesized face region is the addition of the residual with the face region of the main generator  $r + G(x)_F$ . A discriminator  $D_f$  then attempts to discern the “real” face pairs  $(x_F, y_F)$  from the “fake” face pairs  $(x_F, r + G(x)_F)$ , similarly to the original pix2pix [16] objective:

$$\begin{aligned} \mathcal{L}_{\text{face}}(G_f, D_f) = & \mathbb{E}_{(x_F, y_F)}[\log D_f(x_F, y_F)] \\ & + \mathbb{E}_{x_F}[\log(1 - D_f(x_F, G(x)_F + r))]. \end{aligned} \quad (2)$$

Here  $x_F$  is the face region of the original pose stick figure  $x$  and  $y_F$  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  $y_F$ .

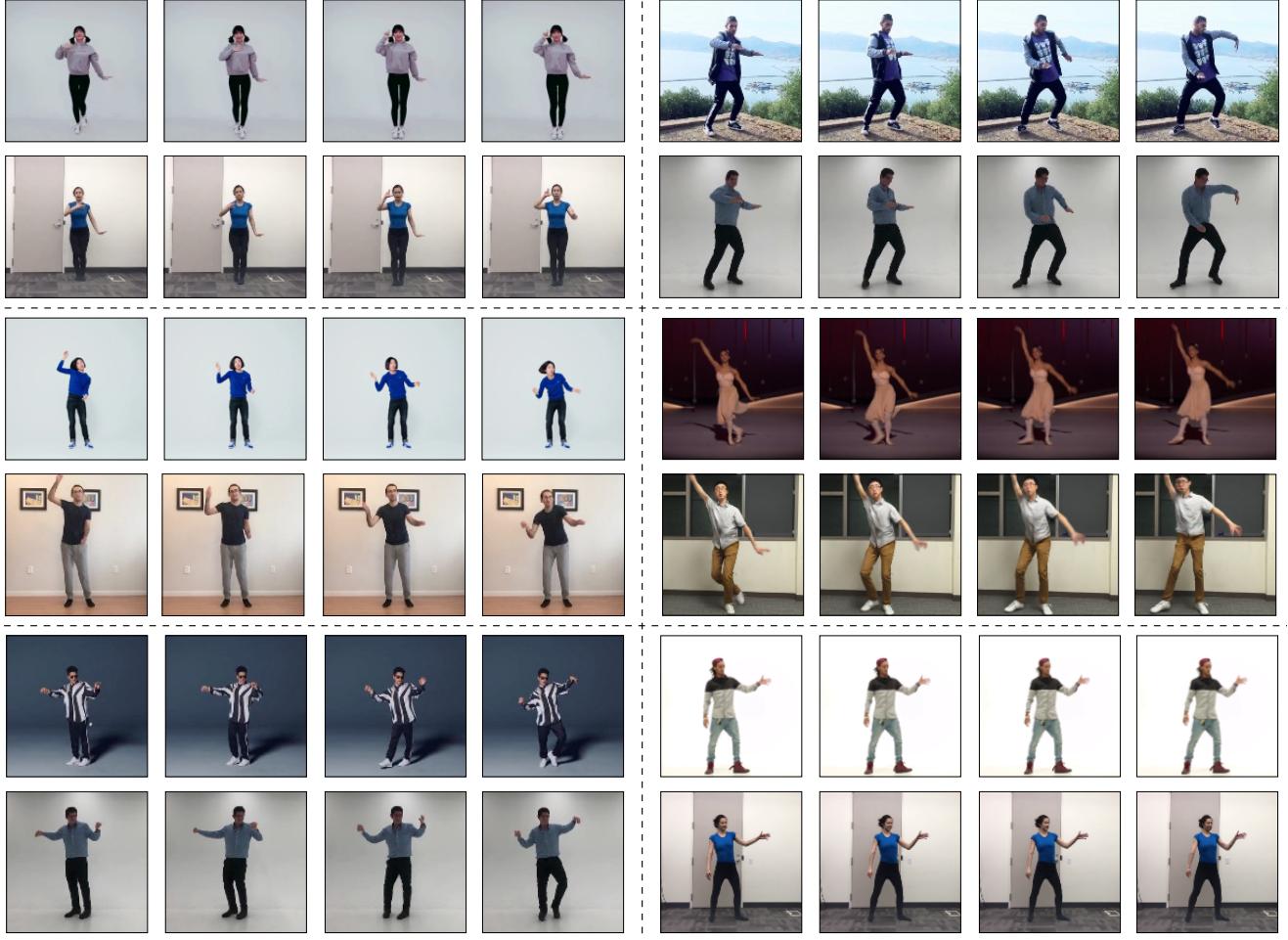


Figure 5: Transfer results. In each section we show four consecutive frames. The top row shows the source subject and the bottom row shows the synthesized outputs of the target person.

### 3.3. 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 -

$$\begin{aligned} \min_G & \left( \left( \max_{D_i} \sum_{k_i} \mathcal{L}_{\text{smooth}}(G, D_k) \right) + \lambda_{FM} \sum_{k_i} \mathcal{L}_{FM}(G, D_k) \right. \\ & \left. + \lambda_P (\mathcal{L}_P(G(x_{t-1}), y_{t-1}) + \mathcal{L}_P(G(x_t), y_t)) \right) \end{aligned} \quad (3)$$

Where  $i = 1, 2, 3$ . Here,  $\mathcal{L}_{\text{GAN}}(G, D)$  is the single image adversarial loss presented in the original pix2pix paper [16]:

$$\mathcal{L}_{\text{GAN}}(G, D) = \mathbb{E}_{(x, y)} [\log D(x, y)] + \mathbb{E}_x [\log(1 - D(x, G(x))] \quad (4)$$

$\mathcal{L}_{FM}(G, D)$  is the discriminator feature-matching loss presented in pix2pixHD, and  $\mathcal{L}_P(G(x), y)$  is the perceptual reconstruction loss [17] which compares pretrained VGGNet [35] features at different layers of the network (fully specified in the Section 8.2).

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

$$\min_{G_f} \left( \left( \max_{D_f} \mathcal{L}_{\text{face}}(G_f, D_f) \right) + \lambda_P \mathcal{L}_P(r + G(x)_F, y_F) \right) \quad (5)$$

where  $\mathcal{L}_{\text{FM}}(G, D)$  is the discriminator feature-matching loss presented in pix2pixHD, and  $\mathcal{L}_P$  is a perceptual reconstruction loss [17] which compares pretrained VGGNet [35] features at different layers of the network. For training details see Section 8.2.

## 4. Experiments

We compare our performance to baseline methods on multiple target subjects and source motions.

### 4.1. Setup

We collect two types of data long, open-source, single-dancer *target* videos which we film ourselves to train our

model on and make publicly available, and in-the-wild *source* videos collected online for motion transfer. The filming set-up for target videos and collection method for source videos are detailed in Section 8.3.

**Baseline methods** 1) **Nearest Neighbors**. For each source video frame, we retrieve the closest match in the training target sequence using the following pose distance metric: For two poses  $p, p'$  each with  $n$  joints  $p_1, \dots, p_n$  and  $p'_1, \dots, p'_n$ , we define the distance between them as the normalized sum of the L2 distances between the corresponding joints  $p_k = (x_k, y_k)$  and  $p'_k = (x'_k, y'_k)$ :

$$d(p, p') = \frac{1}{n} \sum_{k=1}^n \|p_k - p'_k\|_2 \quad (6)$$

The adjacent target matches frames are then concatenated into a frame-by-frame nearest neighbors sequence.

2) **Balakrishnan et al. (PoseWarp)** [3] generate images of a given target subject in a new pose. While, unlike ours, this method is designed for single image synthesis, we use it to synthesize a video frame-by-frame for comparison.

**Ablation conditions** 1) **Frame-by-frame synthesis** (FBF). In this condition we ablate our temporal smoothing setup and apply pix2pixHD [41] on a per-frame basis. 2) **Temporal smoothing** (FBF+TS). In this condition we ablate the Face GAN module to study the difference it makes on the final result. 3) **Our model** (FBF+TS+FG). uses both temporal smoothing and a Face GAN.

**Evaluation metrics** We use perceptual studies on Mechanical Turk for evaluating the video results of our final method in comparison to ablated conditions and baselines. For the ablation study, we further measure the quality of each synthesized frame using two metrics: 1) **SSIM**. Structural Similarity [42] and 2) **LPIPS** Learned Perceptual Image Patch Similarity [47]. We examined the pose distance seen in Equation 6 to measure the similarity between input and synthesized pose. However, we found this distance to be not very informative due to noisy detections.

## 4.2. Quantitative Evaluation

We quantitatively compare our approach against the baselines, and then against ablated versions of our method.

### 4.2.1 Comparison to Baselines

We compare our method to baselines on the same transfer task for all subjects for which we filmed longer videos. From a single out-of-sample source video, we synthesize a transfer video for every baseline-subject pair. We then crop the same 10-second snippets of video for each baseline and subject pair and use these for our perceptual studies.

| Method       | 1     | 2     | 3     | 4     | 5     | Total |
|--------------|-------|-------|-------|-------|-------|-------|
| NN           | 95.9% | 96.4% | 94.6% | 95.8% | 94.7% | 95.1% |
| PoseWarp [3] | 83.1% | 69.9% | 88.7% | 84.6% | 74.4% | 83.3% |

Table 1: Comparison to baselines using perceptual studies for subjects 1 through 5 and in total average. We report the percentage of time participants chose **our** method as more realistic than the baseline.

| Method       | 1     | 2   | 3   | 4   | 5     | Total |
|--------------|-------|-----|-----|-----|-------|-------|
| NN           | 85%   | 93% | 94% | 90% | 91%   | 91.2% |
| PoseWarp [3] | 77.5% | 70% | 80% | 90% | 78.7% | 79.1% |

Table 2: Comparison of our method without Face GAN (FBF+TS variant) to baselines for subjects 1 through 5 and in total average. We report the percentage of time participants chose the FBF+TS ablation as more realistic than the baseline.

Participants on MTurk watched a series of video pairs. In each pair, one video was synthesized using our method; the other by a baseline. They were then asked to pick the more realistic one. Videos of resolution  $144 \times 256$  (as this is the highest resolution that PoseWarp baseline can produce) were shown, and after each pair, participants were given unlimited time to respond. Each task consisted of 18 pairs of videos and was performed by 100 distinct participants. Table 1 displays the results of this study and shows that participants indicated our method is more realistic 95.1% and 83.3% of the time on average in comparison to the Nearest Neighbors and PoseWarp [3] baselines respectively.

We include an additional perceptual study to verify our method is not preferred over the others simply due to more emphasis on face synthesis. We compare the FBF+TS variant (without the Face GAN module) to both baselines in Table 2. We find that the FBF+TS ablation is consistently preferred, albeit slightly less than our full model, over the Nearest Neighbors and PoseWarp baselines 91.2% and 79.1% of the time on average respectively.

### 4.2.2 Ablation Study

We perform an ablation study on held-out test data of the target subject (the source and target are the same) since we do not have paired same-pose frames across subjects.

As shown in Table 3a(bottom), both SSIM and LPIPS scores are similar for all model variations on the body regions. Scores on full images are even more similar, as the ablated models have no difficulty generating the static background. However, Table 3a(top) demonstrates the effectiveness of our face residual generator by showing the improvement of our full model over the the FBF+TS condition.

As these comparisons are in a frame-by-frame fashion they do not emphasize the usefulness of our temporal smoothing setup. The effect of this module can be seen in the qualitative video results and in the perceptual studies

| Region | Metric | FBF   | FBF+TS       | FBF+TS+FG    |
|--------|--------|-------|--------------|--------------|
| Face   | SSIM   | 0.784 | 0.811        | <b>0.816</b> |
|        | LPIPS  | 0.045 | 0.039        | <b>0.036</b> |
| Body   | SSIM   | 0.828 | <b>0.838</b> | <b>0.838</b> |
|        | LPIPS  | 0.057 | 0.051        | <b>0.050</b> |

(a) Metric comparison for synthesized face (top) and full-body (bottom) regions. Metrics are averaged over the 5 subjects. For SSIM higher is better. For LPIPS lower is better.

| Condition | 1     | 2     | 3     | 4     | 5     | Total |
|-----------|-------|-------|-------|-------|-------|-------|
| FBF       | 54.1% | 69.7% | 62.4% | 53.8% | 60.0% | 58.8% |
| FBF+TS    | 59.6% | 56.4% | 50.3% | 53.0% | 53.1% | 53.9% |

(b) Perceptual study results for subjects 1 through 5 and in total average. We report the percentage of time participants chose our method as more realistic than the ablated conditions.

Table 3: Ablation studies. We compare frame-by-frame synthesis (FBF), adding temporal smoothing (FBF+TS) and our final model with temporal smoothing and Face GAN modules (FBF+TS+FG).

| Condition     | 1     | 2   | 3     | 4   | 5     | Total |
|---------------|-------|-----|-------|-----|-------|-------|
| Prefer FBF+TS | 60.5% | 62% | 57.5% | 50% | 62.5% | 58.5% |

Table 4: Comparison of our method without Face GAN (FBF+TS) to the FBF ablation for subjects 1 through 5 and in total average. We report the percentage of time participants chose the FBF+TS ablation over the FBF ablation.

results in Table 3b. Here we see that our method is preferred 58.8% and 53.3% of the time over frame-by-frame synthesis and the No Face GAN (FBF+TS) setup respectively. In general, this shows that incorporating temporal information at training time positively influences video results. Although the effect of the Face GAN can be somewhat subtle, overall this addition benefits our results, especially in the case of subject 1 whose training video is very sharp where facial details are easily visible.

We further compare our method without the Face GAN (FBF+TS) to the frame-by-frame (FBF) ablation to verify our temporal smoothing setup alone improves result quality. Table 4 reports that the FBF+TS ablation is preferred on average over the FBF alone. Note that for subject 4 FBF produced noticeable flickering, but FBF+TS introduced texture artifacts on his loose shirt (see Figure 9).

### 4.3. Qualitative Results

Transfer results for multiple source and target subjects can be seen in Figure 5. The advantage of using the Face GAN module can be seen in a single frame comparison in Figure 6. As mentioned, [3] is designed for single image synthesis. Nonetheless, even for a single frame transfer, we outperform [3] as we show in Figure 7.

While the above single-image and quantitative results

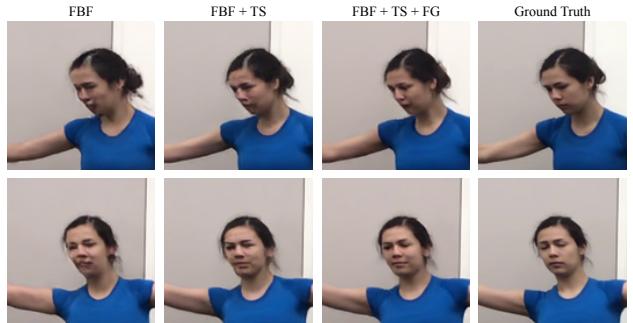


Figure 6: Face image comparison on held-out data. We compare frame-by-frame synthesis (FBF), adding temporal smoothing (FBF+TS) and our full model (FBF+TS+FG).

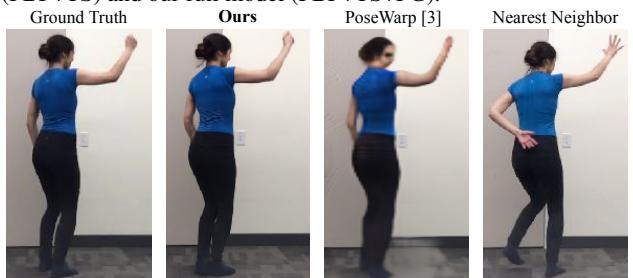


Figure 7: Comparison between our model, [3], and nearest neighbors on single-frame synthesis on held-out data.

(Section 4.2) suggest the superiority of our approach, more significant difference can be observed in our video. There we find the temporal modeling produces more frame to frame coherence than the frame-by-frame ablation, and that adding a specialized facial generator and discriminator adds considerable detail and realism.

## 5. Detecting Fake Videos

Recent progress on image synthesis and generative models has narrowed the gap between synthesized and real images and videos, which has raised legal and ethical questions on video authenticity (among many other social implications). Given the high quality of our results, it is important to investigate mechanisms for detecting computer-generated videos including ones generated by our model.

We train a fake-detector to identify fake videos created by our system — given a video, the fake-detector flags it as real or fake. We train the fake-detector in a parallel fashion to our synthesis process, to classify whether a sequence of 2 consecutive frames is real (from ground-truth frames) or fake (from our generation). This allows the fake-detector to exploit cues based on the fidelity of individual frames as well as consistency across time. To make a decision for the whole video in question, we multiply the decision probabilities for all consecutive frame pairs. Details of the network architecture are included in Section 8.2. For the purpose of training the fake-detector, we collect a 62-subject set of

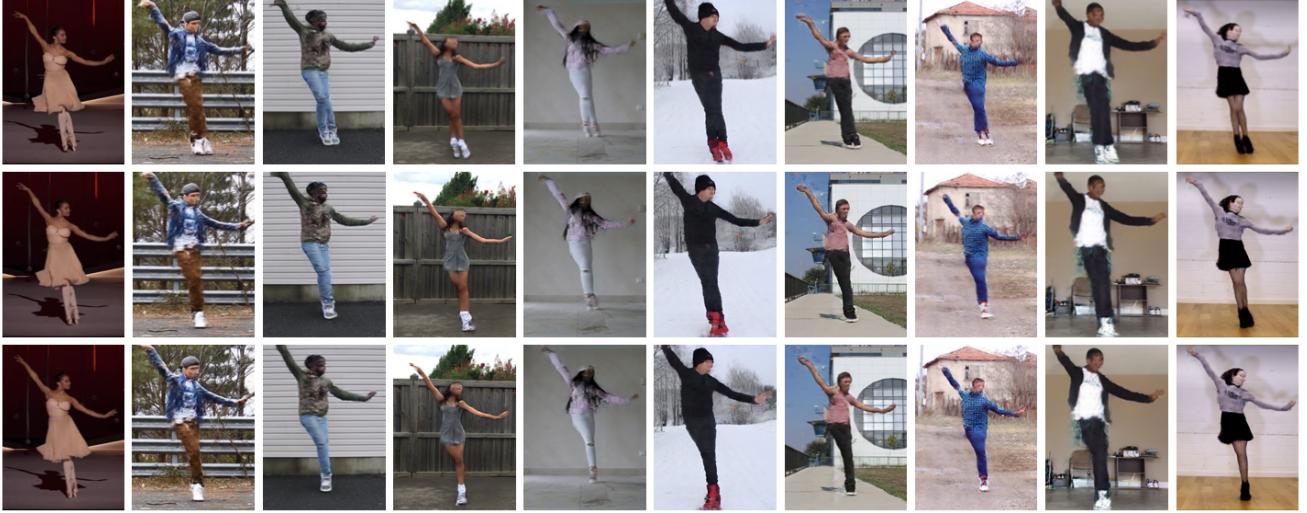


Figure 8: Multi-subject synchronized dancing. By applying the same source motion to multiple subjects, we can create the effect of them performing synchronized dance moves.



Figure 9: Failure cases. Ground truth appearance reference (left) followed by our results (right).

| Source Motion | Same subject | Mars   | Copeland |
|---------------|--------------|--------|----------|
| Accuracy      | 95.68%       | 96.70% | 97.00%   |

Table 5: Fake detection average accuracy for held-out target subjects. As seen in the rows, fake videos were created for each target subject using same-subject and different-subject source motions.

short  $1920 \times 1080$  resolution dancing videos. This larger dataset is collected from public YouTube videos where a subject dances in front of a static camera for an average of 3 minutes. We split this set into 48 subjects for training and 14 held-out subjects for testing.

We train a separate synthesis model for each of the 48 train subjects to produce fake content for detection. By training our fake-detector on multiple fake videos depicting a large set of subjects we ensure that it generalizes to detecting fakes of different people and does not over-fit to one or two individuals. We note that since each person dancing performs a rich set of motions we require less training data than for detecting fakes in still images.

We evaluate our fake-detector on synthesized videos for 14 held-out test subjects. We use both motion taken from the same subject (where the source and target are the same person) and motion driven by a different source subject (Bruno Mars and Misty Copeland) to synthesize fake videos for each held out subject. Our results are shown in Table 5. Overall, the fake-detector successfully distinguishes real and fake sequences regardless of where the source motion is from. As expected, our fake detection accuracy is

lowest for same-person motion transfer, and is highest for transfer of motion from a prima ballerina (Misty Copeland).

## 6. Potential Applications

One fun application of our system is to create a motion-synchronized dancing video with multiple subjects (say, for making a family reunion video). Given trained synthesis models for multiple subjects, we use the same source video to drive the motion of all target subjects — creating an effect of them performing the same dance moves in a synchronized manner. See Figure 8 and the video.

Several systems based on our prototype description were recently successfully employed commercially. One example is an augmented reality stage performance art piece where a 3D-rendered dancer appears to float next to a real dancer [30]. Another is an in-game entertainment application making NBA players dance [44].

## 7. Limitations and Discussion

Our relatively simple model is usually able to create arbitrarily long, good-quality videos of a target person dancing given the movements of a source dancer to follow. However, it suffers from several limitations.

We have included examples of visual artifacts in Figure 9. On the left, our model struggles with loose clothing or hair which is not conveyed well through pose. The middle columns show a missing right arm which was not detected by OpenPose. On the right we observe some texture artifacts in shirt creases. Further work could focus on improving results by combining target videos with different clothing or scene lighting, improving pose detection systems, and mitigating the artifacts caused by high frequency textures in loose/wrinkled clothing or hair.

Our pose normalization solution does not account for

different limb lengths or camera positions. These discrepancies additionally widen the gap between the motion seen in training and testing. However, our model is able to generalize to new motions fairly well from the training data. When filming a target sequence, we have no specific source motion in mind and do not require the target subject performing similar motions to any source. We instead learn a single model that generalizes to a wide range of source motion. However our model sometimes struggles to extrapolate to radically different poses. For example, artifacts can occur if the source motion contains extreme poses such as handstands if the target training data did not contain such upside-down poses. Future work could focus on the training data, i.e. what poses and how many are needed to learn an effective model. This area relates to work on understanding which training examples are most influential [21].

**Acknowledgements** We thank Andrew Owens for the catchy title. This work was supported, in part, by NSF grant IIS-1633310 and research gifts from Adobe, eBay, and Google.

## References

- [1] Kfir Aberman, Mingyi Shi, Jing Liao, D Lischinski, Baoquan Chen, and Daniel Cohen-Or. Deep video-based performance cloning. In *Computer Graphics Forum*, volume 38, pages 219–233. Wiley Online Library, 2019.
- [2] Wissam J Baddar, Geonmo Gu, Sangmin Lee, and Yong Man Ro. 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.
- [3] Guha Balakrishnan, Amy Zhao, Adrian V Dalca, Fredo Durand, and John Guttag. Synthesizing images of humans in unseen poses. In *CVPR*, 2018.
- [4] Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. Recycle-gan: Unsupervised video retargeting. In *ECCV*, 2018.
- [5] Christoph Bregler, Michele Covell, and Malcolm Slaney. Video rewrite: Driving visual speech with audio. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 353–360. ACM Press/Addison-Wesley Publishing Co., 1997.
- [6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Re-altime multi-person 2D pose estimation using part affinity fields. In *CVPR*, 2017.
- [7] Dan Casas, Marco Volino, John Collomosse, and Adrian Hilton. 4D Video Textures for Interactive Character Appearance. *Computer Graphics Forum (Proceedings of EUROGRAPHICS)*, 33(2):371–380, 2014.
- [8] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *IEEE International Conference on Computer Vision (ICCV)*, volume 1, page 3, 2017.
- [9] German KM Cheung, Simon Baker, Jessica Hodgins, and Takeo Kanade. Markerless human motion transfer. In *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, pages 373–378. IEEE, 2004.
- [10] Rodrigo de Bem, Arnab Ghosh, Thalaiyasingam Ajanthan, Ondrej Miksik, N Siddharth, and Philip Torr. A semi-supervised deep generative model for human body analysis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.
- [11] Rodrigo De Bem, Arnab Ghosh, Adnane Boukhayma, Thalaiyasingam Ajanthan, N Siddharth, and Philip Torr. A conditional deep generative model of people in natural images. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1449–1458. IEEE, 2019.
- [12] Alexei A. Efros, Alexander C. Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *IEEE International Conference on Computer Vision*, pages 726–733, Nice, France, 2003.
- [13] Patrick Esser, Ekaterina Sutter, and Björn Ommer. A variational u-net for conditional appearance and shape generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8857–8866, 2018.
- [14] Michael Gleicher. Retargetting motion to new characters. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 33–42. ACM, 1998.
- [15] Chris Hecker, Bernd Raabe, Ryan W Enslow, John DeWeese, Jordan Maynard, and Kees van Prooijen. Real-time motion retargeting to highly varied user-created morphologies. In *ACM Transactions on Graphics (TOG)*, volume 27, page 27. ACM, 2008.
- [16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.
- [17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016.
- [18] Donggyu Joo, Doyeon Kim, and Junmo Kim. Generating a fusion image: One’s identity and another’s shape. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1635–1643, 2018.
- [19] Hyeongwoo Kim, Pablo Carrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):163, 2018.
- [20] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1857–1865. JMLR.org, 2017.
- [21] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1885–1894. JMLR.org, 2017.
- [22] Christoph Lassner, Gerard Pons-Moll, and Peter V Gehler. A generative model of people in clothing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 853–862, 2017.

- [23] Jehee Lee and Sung Yong Shin. A hierarchical approach to interactive motion editing for human-like figures. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 39–48. ACM Press/Addison-Wesley Publishing Co., 1999.
- [24] Lingjie Liu, Weipeng Xu, Michael Zollhoefer, Hyeongwoo Kim, Florian Bernard, Marc Habermann, Wenping Wang, and Christian Theobalt. Neural rendering and reenactment of human actor videos. *ACM Transactions on Graphics 2019 (TOG)*, 2019.
- [25] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017.
- [26] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016.
- [27] Liqian Ma, Xu Jia, Qianru Sun, Bernt Schiele, Tinne Tuytelaars, and Luc Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017.
- [28] Liqian Ma, Qianru Sun, Stamatios Georgoulis, Luc Van Gool, Bernt Schiele, and Mario Fritz. Disentangled person image generation. In *CVPR*, pages 99–108, 2018.
- [29] Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskyi, Jonathan Taylor, Julien Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, Adarsh Kowdle, Christoph Rhemann, Dan B Goldman, Cem Keskin, Steve Seitz, Shahram Izadi, and Sean Fanello. Lookingood: Enhancing performance capture with real-time neural re-rendering. *ACM Trans. Graph.*, 37(6):255:1–255:14, Dec. 2018.
- [30] Kyle McDonald. Dance x Machine Learning: First Steps. <https://medium.com/@kcimc/discrete-figures-7d9e9c275c47>, 2019. [Online; accessed 21-March-2019].
- [31] Greg Mori, Alex Berg, Alexei Efros, Ashley Eden, and Jitendra Malik. Video based motion synthesis by splicing and morphing. Technical Report UCB//CSD-04-1337, University of California, Berkeley, June 2004.
- [32] Iasonas Kokkinos Riza Alp Güler, Natalia Neverova. Densepose: Dense human pose estimation in the wild. *arXiv*, 2018.
- [33] Aliaksandr Siarohin, Enver Sangineto, Stphane Lathuilière, and Nicu Sebe. Deformable gans for pose-based human image generation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [34] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.
- [35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [36] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [37] Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. Neural kinematic networks for unsupervised motion retargetting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [38] Ruben Villegas, Jimei Yang, Yuliang Zou, Sungryull Sohn, Xunyu Lin, and Honglak Lee. Learning to generate long-term future via hierarchical prediction. *arXiv preprint arXiv:1704.05831*, 2017.
- [39] Jacob Walker, Kenneth Marino, Abhinav Gupta, and Martial Hebert. The pose knows: Video forecasting by generating pose futures. In *International Conference on Computer Vision*, 2017.
- [40] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
- [41] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [42] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [43] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.
- [44] Xpire. Using AI to make NBA players dance. <https://tinyurl.com/y3bdj5p5>, 2019. [Online; accessed 21-March-2019].
- [45] Feng Xu, Yebin Liu, Carsten Stoll, James Tompkin, Gaurav Bharaj, Qionghai Dai, Hans-Peter Seidel, Jan Kautz, and Christian Theobalt. Video-based characters: creating new human performances from a multi-view video database. In *ACM Transactions on Graphics (TOG)*, volume 30, page 32. ACM, 2011.
- [46] Mihai Zanfir, Alin-Ionut Popa, Andrei Zanfir, and Cristian Sminchisescu. Human appearance transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5391–5399, 2018.
- [47] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.
- [48] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

## 8. Appendix

### 8.1. Video Demonstration

Our video demo can be found at <https://youtu.be/mSaIrz8lM1U> and examples from our comparison to baselines and ablation study can be found at <https://youtu.be/sQD0WVS0blg>.

### 8.2. Implementation Details

Our generator and discriminator architectures are modified from pix2pixHD [41] to handle the temporal set-

ting. We follow the progressive learning schedule from pix2pixHD and learn to synthesize at  $512 \times 256$  at the first (global) stage, and then upsample to  $1024 \times 512$  at the second (local) stage. For predicting face residuals, we use the global generator of pix2pixHD and a single  $70 \times 70$  PatchGAN discriminator [16]. We set hyperparameters  $\lambda_P = 5$  and  $\lambda_{VGG} = 10$  during the global and local training stages respectively. For the dataset collected in Section 4.1, we trained the global stage for 5 epochs, the local stage for 30 epochs, and the face GAN for 5 epochs.

For the perceptual loss  $\mathcal{L}_P$ , we compare the `conv1_1`, `conv2_1`, `conv3_1`, `conv4_1`, and `conv5_1` layer outputs of the VGG-19 network.

Our generator and discriminator architectures follow that presented by Wang et al. [41]. The fake-detector architectures matches that of the discriminator with a final fully connected layer.

### 8.3. Dataset Collection

Our dataset of long target videos consists of footage we filmed ourselves from 8 to 17 minutes with 4 videos at  $1920 \times 1080$  resolution and 1 at  $1280 \times 720$ . Our goal in collecting a dataset of target videos is to provide the community with open-source data for which we explicitly collect release forms in which subjects allow their data to be released to other researchers. We recruited target subjects from different sources: friends, professional dancers, reporters etc. 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. Similarly, we used a stationary camera to ensure a static background in all frames. To ensure the quality of the frames, we filmed our target subjects for between 8 and 30 minutes of real time footage at 120 frames per second using a modern cellphone camera, and use the first 20% of the footage for training and the last 80% for testing. Since our pose representation does not encode information about clothes and hair, we instructed our target subjects not to wear loose clothing and to tie up long hair.

In contrast, source videos can be easily collected online as we only require decent pose detections on these. We therefore use in-the-wild single-dancer videos where the only restriction we enforce is a static camera position.

### 8.4. Comparison with vid2vid

We also compare our model with a concurrent video synthesis framework called vid2vid [40]. The excessive requirement of memory and computing power of vid2vid prohibits us from comparing with their model in the high resolution setup. Instead, we train both our model and theirs in lower resolution ( $512 \times 256$ ). Our system and vid2vid generally perform similarly and produce results of comparable quality. We provide a qualitative comparison in Figure 10.



Figure 10: We compare a lower resolution version of our model without a Face GAN (top) with a lower resolution of vid2vid [41]. We find our results comparable.

### 8.5. Global Pose Normalization Details

In this section we describe our normalization method to match poses between the source and target. Consider a case where the source subject is significantly taller in frame than the target or is slightly elevated above the target subject's in frame position. If we directly input the unmodified poses to our system, we may generate images of the target person which are not congruent with the scene. In this example, the target person may appear large with respect to the background or surrounding objects, and may appear to be levitating since the input pose places the feet above the floor. Additionally, when generating an image from a very different pose from the in proportion and reasonably positioned poses in training, the overall quality of synthesis is expected to decline. Therefore we design a method to reasonably match the poses by finding a suitable transformation between the source and target poses. We parametrize this transformation in terms of a scale and translation factor applied to all pose keypoints for a given frame.

To find a suitable translation factor, we need to determine the position of both subjects within their respective frames. We first find the closest position  $s_{close}$  and farthest position  $s_{far}$  the source subject is away from the camera in their video. Similarly, we do the same for the target by determining  $t_{close}$  and  $t_{far}$  respectively. The goal is then to map the close and far range of the source to that of the target subject as to match the positions of both subjects, i.e.  $s_{far} \mapsto t_{far}$  and  $s_{close} \mapsto t_{close}$ . Given a frame where the source is at position  $y$ , we then translate the source's pose vertically by:

$$translation = t_{far} + \frac{y - s_{far}}{s_{close} - s_{far}}(t_{close} - t_{far}) \quad (7)$$

In practice, we use the average of the y coordinates of the subject's ankles to determine the position within a given frame.

To reasonably scale the source poses, we determine the

heights of each subject at their closest and farthest positions in their video - denote these quantities as  $h_{s_{close}}, h_{s_{far}}$  for the source and  $h_{t_{close}}, h_{t_{far}}$  for the target subjects respectively. We then determine separate scales for the close position given by  $c_{close} = \frac{h_{t_{close}}}{h_{s_{close}}}$  and similarly for the far position given by  $c_{far} = \frac{h_{t_{far}}}{h_{s_{far}}}$ . When given a frame where the source is at position  $y$ , we scale the source's pose (in both x, y directions) by:

$$scale = c_{far} + \frac{y - s_{far}}{s_{close} - s_{far}}(c_{close} - c_{far}) \quad (8)$$

We use the euclidean distance between the average ankle position and the nose keypoint of our given pose as the subject's height in a given frame.

After the translation and scale factors have been determined for a given source pose, we then add the translation to all keypoints and then apply the scale factor so that the ankle  $y$  positions remain the same (i.e. the ground is the x axis).

Given poses from a subject, we find the close position by taking the maximum y coordinate of their average ankle position over all frames.

$$s_{close} = \max \left\{ \frac{s_{ankle1} + s_{ankle2}}{2} \right\}$$

The far position is found by clustering the y ankle 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. If we denote  $S = \frac{s_{ankle1} + s_{ankle2}}{2}$  as the average ankle position in a given frame, then the clustering is as described by the set

$$\max\{S : ||S - s_{med}|| < \alpha |s_{close} - s_{med}| \} \cap \{S < s_{med}\} \quad (9)$$

where  $s_{med}$  is the median foot position,  $\max$  is the maximum ankle position, and  $\epsilon$  and  $\alpha$  are scalars. In practice we find setting  $\alpha = 0.7$  generally works well, although this scalar can be finetuned on a case by case basis since it depends highly on the camera height and the subject's range of motion.