Generating Video of a Target Character Performing Given Actions

Christopher Liman Mohsen Mahvash Rishi N

Abstract

We investigate two general video synthesis methods that use conditional generative adversarial networks (GANs) to create artificial videos. The goal is to generate videos that contain a target character performing new actions. In the first method, the GANs are trained on the video of the target character to directly generate the images of the target for new actions. In the second method, we first learn the representations of the 3D geometry and the appearance of the target person from the target video and then apply a texture to the body using a GAN. We apply both methods to create videos of the same target for several selected poses taken from a human action dataset.
Project github `https://github.com/github-rishi/cs230_project`

1 Introduction

In this project, we address how to transfer actions, such as jumping jacks, to a target person by training the model using a short video of the person. Generating realistic personalized actions can be used for many applications such as gaming, personalized VR, and clothing. As an example, the users of a metaverse can create their realistic avatars similar to their actual appearance by training on their short video, then potentially modifying their avatar appearance with different clothing or accessories, and also moving their avatar with joint motions captured by their VR handsets.

We explore two general approaches to create personalized human body actions. In both approaches, the output of the action generator is a video of the target person performing a given action and the input of the action generator is a time sequence of the body key points for the given action. In the first approach, we directly create the images of the body motions for given body poses using a conditional Generative Adversarial Network (GAN)[1]. The GAN is trained with a video of the target user. In the second approach, we partially or fully integrate the graphic rendering pipeline into the video generator to create the output images. The joint points of the given action are converted to a body mesh representing the person's body structures in 3D and the actions. We then convert the mesh to images and train a conditional GAN to apply texture to the images. The second approach has the advantage that we could texture the body mesh using a conventional texture map or a neural renderer, which can be more robust to action change than image to image training.

2 Related work

Several methods for generating artificial videos have been researched. We divide them into two categories: (1):Direct image to image approach [2][3][4][7][17] and (2): Computer graphics based methods [8] [9] [11][12][15]

In direct image to image methods, conditional GANs [1][18] are used to create each image of the output video for the given pose. The given pose is extracted from the video of a different person

CS230: Deep Learning, Fall 2022, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

performing the action. Chen et al [2] used two Conditional GANs to generate the body and the face. They used two sequential frames instead of one frame to train body GANs. [17] also uses a GAN to convert the frames of video. They used a spatio-temporal adversarial objective to achieve coherent video frames. [7] proposes a network weight generation module that utilizes an attention mechanism. They claim their approach can generate unseen actions for the target person. Tulyakov et al [3] proposes MoCoGAN that decomposes the content and motion of a video. They showed that the motion of a person in a video can be changed without any change in the appearance of the person.

In computer graphics based methods, the 3D mesh and the appearance of the target person is derived from a video of the target person and then a neural renderer is used to generate the output video. Liu, Lingjie, et al.[8] synthesized residual deformations and the dynamic appearance of the target due to new action using a neural network model. They trained their neural rendering network using videos from many cameras, 3D scans. [11] also used a neural renderer to create texture. Most graphics based approaches use 3D scans to capture body shape and camera. In this project, we use the videos from a single camera to train our model.

The graphics based method requires body mesh of the target in the video generation pipeline. The Max Planck Institute for Intelligent Systems created Skinned Multi-Person Linear model (SMPL) and various extensions of it such as SMPLX to represent body meshes in different poses[13][14][15]. The parameters of the model include body appearance and body pose. The pose of the body can be changed without creating any artifacts in body shape and appearance. We used SMPLX to create our avatar for given actions. Different methods have been proposed to capture the SMPL from an image [12][14][16]. SMPLify-X [14] uses a regression approach to fit a SMPL mesh and OpenPose parameters [19] to an image. EasyMoCap [10][20] and VIBE [13] use GAN to estimate SMPL body model parameters for a given image, TCMR [16] extracts smooth SMPL mesh of video frames. Most of the related works use conditional GANs to convert the images[1][18]. These GANs learn a loss function to train the mapping between input and output images.

3 Dataset and Features

Our model is based on the image to image translation. Our target person video (used for training), is from: `https://youtu.be/XExv47DPIys`. This training video is from the youtube playlist mentioned as part of imaginaire dataset for vid2vid model. We used this youtube video for training since it has a single person dancing in front of a centrally located camera and the person stays in the center of the frame. The video length for our training needs to be around 3-5 mins and this playlist offers a wide range of 3-5 minutes videos with a wide range of poses. The static background in the video also assures that there are no challenges in regard to generating the proper background when we do pose translation.

The resolution of the images generated from the youtube video are 1920 x 1080. We use 2000 consecutive images from the beginning of the video for our training set. For our validation set, we use 500 consecutive images from a later part of the video, where the person performs different motions. For creating our action dataset required for the test, we use action videos that were found on youtube, as well a youtube video of another person dancing (appendix).

4 Methods

4.1 Image to Image Method

Our method is built upon the video generation method of [2] which is also our baseline for evaluation `https://github.com/stanleyshly/EverybodyDanceNow.git`. During training, we run OpenPose [19] to get pose images from a single target video and use them to train the pix2pixHD conditional GAN, see Figure 1. During the test, we feed the pose images of the source videos to the trained model to get the videos of the target person with the source poses. The pix2pix HD GAN model is trained using two sequential images and their open pose representations (sequential frames are not shown in Figure1. The GAN takes input of target poses for time t and t+1, and generates the image for time t, which also trains the model for the transitioning in between the images. The discriminator is trained using a combination of both generated images and the open pose images. The generator cost function is a combination of temporal smoothing loss function L_{smooth}, feature mapping loss L_{FM}, and perceptual VGG loss L_P[1][2]. We can use a different weight for each part of the body, and we decided to use a higher weight λ_F for the face.

$$Cost = \min_G(\max_{D_i} \sum_{k_i} L_{smooth}(G, D_k) + \lambda_{FM} \sum_{k_i} L_{FM}(G, D_k)$$
$$+ \lambda_P(L_P(G(x_{t-1}), y_{t-1}) + L_P(G(x_t), y_t))$$
$$+\lambda_F(L_P(G(x_{t-1})_F, (y_{t-1})_F) + L_P(G(x_t)_F, (y_t)_F)))$$

where $L_P(G_F, y_F)$ represents the pixels inside the face bounding box.

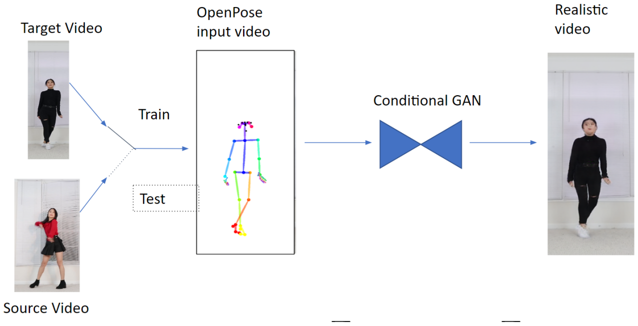


Figure 1: Transforming action of the source to the target by training a GAN with the target video. We switch the input to the source video during the test to generate the desired video.

4.2 Computer Graphics Based Method

Instead of using the OpenPose skeleton representation as input to the GAN model, we use SMPLify-X or EasyMoCap GAN to create SMPL-X representation of the poses as input to our GAN model 2. This input is a realistic 3D model of the human body that more accurately represents body shape compared to a skeleton. The different body parts are labeled with different colors. The pix2pixHD generator translates semantic label maps to realistic images, and having the pixels of the label maps aligned with the ground truth images should improve the generator performance. Also, the SMPL-X model includes more expressive face and hands compared to the SMPL model, which should help generate videos with a more expressive pose.

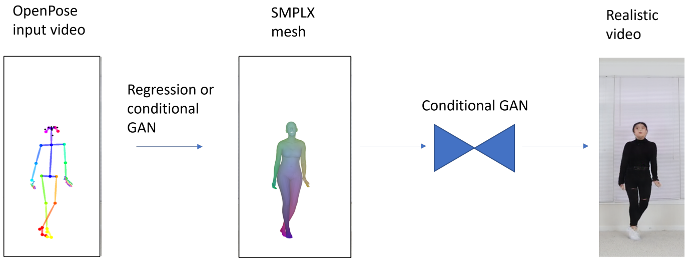


Figure 2: Block diagram of SMPL-X method

5 Experiments/Results/Discussion

There were many hyperparameters we considered varying when training our GAN, but due to the lengthy training time required, we focused on changing the method and the cost function. For all our experiments, we used a learning rate of 2e-4 for the Adam optimizer, and we used the same beta1=0.5, beta2=0.999, eps=1e-8 which is described in the pix2pixHD model. We used a mini-batch size of 1 due to running out of GPU memory if this is increased. We typically ran 30 epochs as this was more than enough for the cost function to converge, but ran 40 epochs for the face penalty cost function as it required more epochs to converge, due to the face penalty.

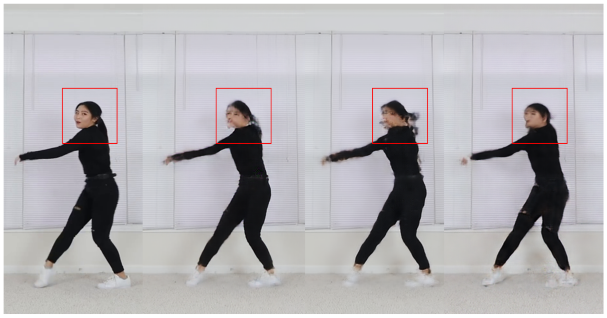


Figure 3: Comparing three OpenPose methods with ground truth on validation data: Left to right: Ground truth, Baseline, L1, Face penalty.

We performed various experiments based on the methods described earlier and evaluate the results when it is possible see tables 1 and 2:

1) Face GAN: We use the approach of [2] without the face GAN as our baseline. We tried including the face GAN of [2]. However, this did not improve the generated video, and caused discontinuities between the face and the body.

2) Output L1: In one experiment, we used an L1 loss on the pixel values between real and generated images, replacing the perceptual VGG loss used in pix2pixHD, which is an L1 loss on the outputs of specific convolutional layers between real and generated images. Our loss improved the objective quality metrics as shown in table 1.

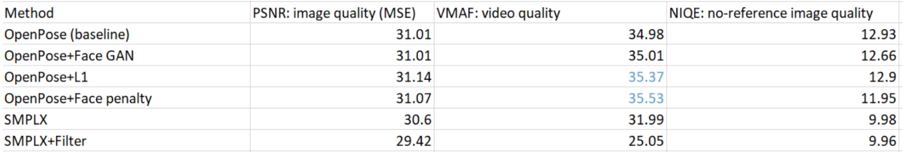
3) Face Penalty: Our cost function contains a VGG loss penalty for the face. With our model we found much better results for the face generation compared to the base model, figure 3. Because of the higher penalty for the face in the VGG loss function, VGG loss started at a much higher value in the initial iterations, compared to the base model. We also found that the generator and the discriminator didn't converge to a common loss with this method. However, with increased VGG loss penalty for face we found that overall perceptual quality of the video improved. We believe this is because the model is improving face quality with some compromise on other pixels. But since face quality is important for visual quality of the video, the overall quality looks better.

4) SMPL-X Input: We used the SMPL-X meshes and SMPLify-X[14] regression instead of OpenPose images to train our model. With SMPL-X and SMPLify-X we didn't observe much improvement over the baseline model in the overall generated video, however, in certain images we found that the model was able to represent the hand and body feature more cleanly compared to OpenPose based model. Since SMPL-X is generated using a regression model, we found that the generated images of SMPL-X had many outliers due to which we saw the generated video to be shaky. We created our own filter to remove these outliers from source SMPL-X images and obtained a much smoother target video.

5) Pose Normalization: We found that our method changed the shape of target character to become very tall or short when the body to image ratios of the target and source are very different. We implemented the pose normalization by modifying the json files created after running OpenPose. We found the max bounding box for the target pose based on all the training images. With the max bounding box, we created a reference coordinate which denotes the Xmin, Ymin of the target pose. With the length and width of the max bounding box along with Xmin, Ymin co-ordinate, we interpolated the source pose to these coordinates after finding the maximum bounding box of the source poses.

We chose three objective quality metrics to quantitatively assess our generated videos, see table 1. Two of them are full-reference methods, which compare the quality of the videos generated from our validation set (same target person but different actions as the training set) against the ground truth of the original validation videos. Note that we cannot apply these metrics to the test set, because we do not have the ground truth of the target person performing the different actions. The first metric is PSNR (peak signal-to-noise ratio), which operates on individual frames and is proportional to the negative log of the mean squared error of the RGB pixel values. The second metric is VMAF (Video Multimethod Assessment Fusion) [5], which is a fusion of different video quality metrics using support vector machine-based regression, and includes temporal aspects of quality, unlike PSNR.

The third metric is NIQE (Natural Image Quality Evaluator) [6], a no-reference image quality metric which makes use of deviations from regularities in natural images. We also run this third metric on the test dataset. For all of the metrics, higher is better, and VMAF has a range of 0-100.



Method	PSNR: image quality (MSE)	VMAF: video quality	NIQE: no-reference image quality
OpenPose (baseline)	31.01	34.98	12.93
OpenPose+Face GAN	31.01	35.01	12.66
OpenPose+L1	31.14	35.37	12.9
OpenPose+Face penalty	31.07	35.53	11.95
SMPLX	30.6	31.99	9.98
SMPLX+Filter	29.42	25.05	9.96

Table 1: objective quality metrics on validation dataset, average of frames

Overall, we found that the L1 and face penalty modifications to the OpenPose method slightly improved VMAF video quality, with the face penalty performing the best. The SMPLX method had worse performance but in certain frames it subjectively looked more realistic. To evaluate our model for various actions, we tested our face penalty model on another dance video of a person in different clothing and also tried three action videos of random people performing jumping jacks, lunges and punching, figure 4. We found that for the other dance video and for jumping jacks, the images generated with the source pose are much cleaner and have proper face representation, compared to the images generated with lunges and punching actions. It is due to the fact that the jumping jack and dance have poses which more resemble our training set of the person dancing, while lunges and punching are poses which don't resemble any poses in the training set.

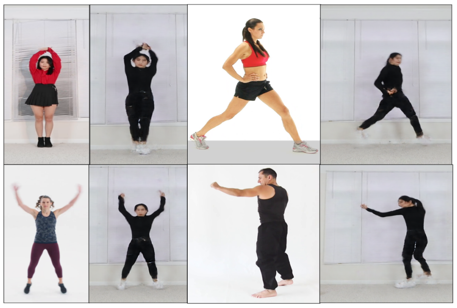
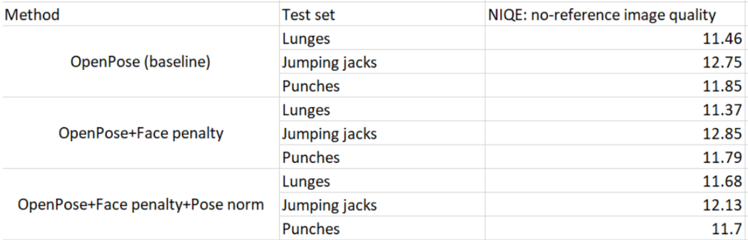


Figure 4: Inference of action videos on the trained model. Top left: different dance video, Top Right: Lunges, Bottom Left: Jumping Jack, Bottom Right: Punching



Method	Test set	NIQE: no-reference image quality
OpenPose (baseline)	Lunges	11.46
	Jumping jacks	12.75
	Punches	11.85
OpenPose+Face penalty	Lunges	11.37
	Jumping jacks	12.85
	Punches	11.79
OpenPose+Face penalty+Pose norm	Lunges	11.68
	Jumping jacks	12.13
	Punches	11.7

Table 2: no-reference objective quality metric on test dataset, average of frames

6 Conclusion/Future Work

In this work, we have demonstrated two general approaches that use conditional GANs to generate videos of a target person with new actions. The first approach is to train the GAN directly on the pose or pose image of the target person, while the second approach is to train the GAN on the 3D geometry or 3D geometry image of the target person. We obtain better results with the first approach, although the second approach has some advantages in that the 3D geometry has more information than just the pose. We would have liked to train on more target videos and tune more hyperparameters but were limited by computing resources. Future work could explore conditioning the GAN not on images but on pose or 3D geometry vectors, or for the second approach to train the generator to generate texture maps for the 3D geometry instead of directly generating realistic images.

7 Contributions

All group members contributed equally to this project.

References

[1] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, “High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs.” arXiv, Aug. 20, 2018. doi: 10.48550/arXiv.1711.11585.

[2] C. Chan, S. Ginosar, T. Zhou, and A. A. Efros, “Everybody Dance Now.” arXiv, Aug. 27, 2019. doi: 10.48550/arXiv.1808.07371.

[3] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, “MoCoGAN: Decomposing Motion and Content for Video Generation.” arXiv, Dec. 13, 2017. doi: 10.48550/arXiv.1707.04993.

[4] A. Clark, J. Donahue, and K. Simonyan, “Adversarial Video Generation on Complex Datasets.” arXiv, Sep. 25, 2019. doi: 10.48550/arXiv.1907.06571.

[5] T.-J. Liu, Y.-C. Lin, W. Lin, and C.-C. J. Kuo, “Visual quality assessment: recent developments, coding applications and future trends,” SIP, vol. 2, no. 1, Jul. 2013, doi: 10.1017/ATSIP.2013.5.

[6] A. Mittal, R. Soundararajan, and A. C. Bovik, “Making a ‘Completely Blind’ Image Quality Analyzer,” IEEE Signal Processing Letters, vol. 20, no. 3, pp. 209–212, Mar. 2013, doi: 10.1109/LSP.2012.2227726.

[7] T.-C. Wang, M.-Y. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro, “Few-shot Video-to-Video Synthesis.” arXiv, Oct. 28, 2019. doi: 10.48550/arXiv.1910.12713.

[8] Liu, Lingjie, et al. “Neural Actor: Neural Free-View Synthesis of Human Actors with Pose Control.” ACM Transactions on Graphics, vol. 40, no. 6, Dec. 2021, p. 219:1-219:16. December 2021, https://doi.org/10.1145/3478513.3480528.

[9] Nagano, Koki, et al. “PaGAN: Real-Time Avatars Using Dynamic Textures.” ACM Transactions on Graphics, vol. 37, no. 6, Dec. 2018, p. 258:1-258:12. December 2018, https://doi.org/10.1145/3272127.3275075

[10] Shuai, Qing, et al. “Novel View Synthesis of Human Interactions from Sparse Multi-View Videos.” ACM SIGGRAPH 2022 Conference Proceedings, Association for Computing Machinery, 2022, pp. 1–10. ACM Digital Library, https://doi.org/10.1145/3528233.3530704.

[11] Prokudin, Sergey, et al. SMPLpix: Neural Avatars from 3D Human Models. arXiv, 9 Nov. 2020. arXiv.org, https://doi.org/10.48550/arXiv.2008.06872.

[12] Kocabas, Muhammed, et al. VIBE: Video Inference for Human Body Pose and Shape Estimation. arXiv, 29 Apr. 2020. arXiv.org, https://doi.org/10.48550/arXiv.1912.05656.

[13] Loper, Matthew, et al. “SMPL: A Skinned Multi-Person Linear Model.” ACM Transactions on Graphics, vol. 34, no. 6, Nov. 2015, p. 248:1-248:16. November 2015, https://doi.org/10.1145/2816795.2818013.

[14] Pavlakos, Georgios, et al. Expressive Body Capture: 3D Hands, Face, and Body From a Single Image. 2019, pp. 10975–85.

[15] S. Saito, J. Yang, Q. Ma, and M. J. Black, “SCANimate: Weakly Supervised Learning of Skinned Clothed Avatar Networks.” arXiv, Apr. 08, 2021. doi: 10.48550/arXiv.2104.03313.

[16] Choi, Hongsuk, et al. Beyond Static Features for Temporally Consistent 3D Human Pose and Shape from a Video. arXiv, 27 Apr. 2021. arXiv.org, https://doi.org/10.48550/arXiv.2011.08627.

[17] Wang, Ting-Chun, et al. Video-to-Video Synthesis. arXiv, 3 Dec. 2018. arXiv.org, https://doi.org/10.48550/arXiv.1808.06601.

[18] Isola, Phillip, et al. Image-to-Image Translation with Conditional Adversarial Networks. arXiv, 26 Nov. 2018. arXiv.org, https://doi.org/10.48550/arXiv.1611.07004.

[19] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields.” arXiv, Apr. 13, 2017. doi: 10.48550/arXiv.1611.08050.

[20]https://github.com/zju3dv/EasyMocap

8 Appendix

Figure 5: Test set videos.

Other dance video: `https://youtu.be/vcClNX5zNIs`
Jumping jacks: `https://youtu.be/NeN8c-94EOo`
Punches: `https://youtu.be/M_4Vt5lfEUE`
Lunges: `https://youtu.be/AvBrsGNA7V8`

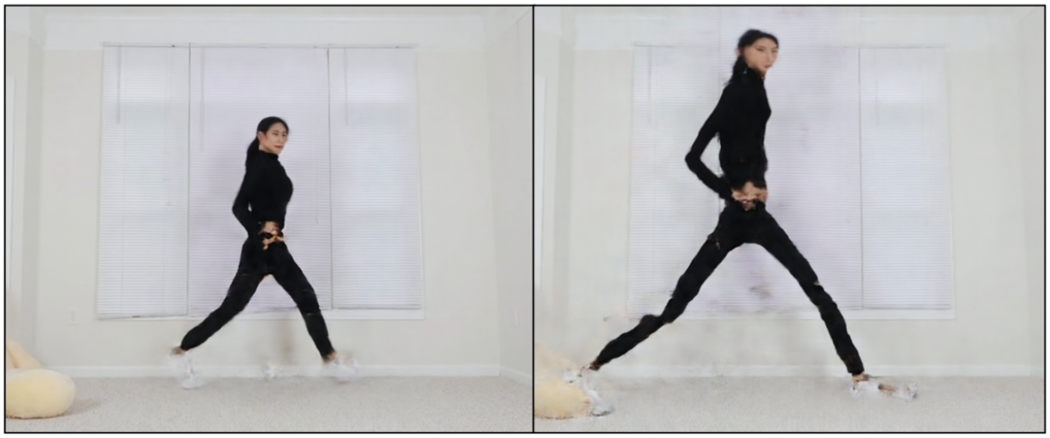


Figure 6: Inference of lunges video with and without pose normalization.