CS230

DNCEGAN

Allison Lettiere
Stanford University
agl12@stanford.edu

Michelle Xing
Stanford University
mxing621@stanford.edu

Abstract

We explored video production using Generative Adversarial Networks (GANs), presenting a novel structure that incorporates a single generator network and three discriminator networks: one to capture the content of a single frame, a second to capture overall video content, and a third to capture changes in motion and figures in nearby frames. We examined the ability for GANs to reproduce video clips containing large quantities of motion, specifically exploring the motion patterns in dance videos. We trained our conditional model on 15 different dance classes. While previous exploration with content-motion models looks largely at video production for training sets in which motion is localized to one person or one area of the frame, we determined that our model can also produce frames representative of motion when learning sequences with significantly higher motion quantities.

1 Introduction

Generative Adversarial Networks employ the unique and effective method of training two separate deep neural networks—the generator and the discriminator–which compete against each other. It is out of this competition that drives both neural networks to improve their ability to reach their individual goals: while the generator wants to produce realistic data from vectors of noise, the discriminator wants to distinguish between samples from the generator distribution and the real data.

Video production using deep generative models, and more specifically deep generative adversarial networks, is a problem that has only recently been explored in the field of deep learning research. Evolving out of image production with GANs, video production offers a new challenge, requiring the computational device to learn to reproduce more than just the content of examples in the dataset, but also sequences of possible motion once a starting frame has been developed. Previous research has focused on using fairly uniform sequences of training video input, with motion standardized between different samples. Instead, in our work, we explore how well a novel GANs model, built to capture additional sequential motion data, can perform on a dataset containing videos of dance performances, in which the quantity of motion and the number of people in a typical frame is much higher that previous video GAN inputs.

The substantial motion in our dance dataset causes this task to be more difficult than the facial expression and gesture video production featured in many past video GAN implementations, as the the GANs may misconceive the input data's motion and produce a series of blurry, discombobulated frames. Consequently, in our project, in order to capture the numerous dimensions of dance videos and generate vidoes that reflect our complex dataset, we use not only an image and a video discriminator, but also an additional motion discriminator.

While incorporating additional motion information, we implemented categorical content production through a conditional GAN (Mirza & Osindero, 2014). Our model produces frames (and furthermore videos) of dances from the fifteen dance categories in our dataset. The input to our algorithm is a collection of dance videos, divided by dance type category. Each video is represented as a horizontally concatenated sequence of frames. We produce new sequences of frames, which can then be converted to gif or mp4 format. We aimed to analyze what the GAN determines is valuable for differentiating between each of the dance categories. We compared the output of the different input categories to determine if the GAN either truly views these categories as distinct, or if the GAN is not able to extract the differences as a result of the "visually overlapping" nature of the dance types in the dataset. This approach provided an additional avenue for producing conditioned outputs, based on increasingly complex input datasets. We determined that incorporating additional motion information, specifically for conditioned dance

CS230: Deep Learning, Spring 2020, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

classes, produces visually overlapping contiguous frames, that while emulating some dance patterns, do not yield class-divided, clearly defined dance videos.

2 Related work

There are numerous previous instances of exploration with content production via GANs. One notable example is *Conditional generative adversarial nets for convolutional face generation* (Gauthier, 2014). This paper is based off of the The author explores using specific categories for image production, much like how we aim to explore specific dance category production. A particularly clever implementation step was the decision to apply random shifts to the condtion axes in the data in order to determine how each axis impacted the output data, as the changes in output data would only be as a result of the specific axis shift. This conditional image exploration, which is also outlined in Mirza & Osindero, is the basis for the video conditional GAN analysis we explore.

One realm of video-GAN analysis lies in the future frame prediction problem referenced in Lee, et al. The authors' model combined two techniques, integrating the underlying stochastic nature of the data and adversarially training a model to produce realistic images. The authors aimed to learn the underlying motion patterns in the data to be able to predict future frames, much like how we aim to manipulate a GAN to learn the underlying motion patterns in the different dances. An additional frame prediction task is outlined in Mathieu, Couprie, & LeCun. The authors employed a discriminator model that receives an input sequence of frames and predicts the probability that the last frames have been generated by the generator. The generative model is a multi-scale network that takes an input of four frames and outputs a single frame. While providing an interesting starting point for understanding models that integrate information about sequential frames, the overall frame prediction problem complicates the ability the produce an entire sequence of novel frames.

The motion and content decomposed GAN (MoCoGAN) model (Tulyakov et al., 2018) is a generative model that can be used specifically on videos. The model uses two different subspaces, one for video content and one for video motion. Unlike previous models, MoCoGAN uses deep convolutional networks for both image and video classification. The model employs two discriminators trained on videos/images from the training and generated set. The generator and discriminator compete to minimize their individual loss, with the generator aiming to produce images and videos that fool the discriminator, and the discriminator aiming to correctly classify images and videos as real or fake from the generator. Our model adds an additional facet to the discriminators explored in MoCoGAN, with a supplement to incorporate further motion and variety in input data.

In *DCVGAN: Depth Conditional Video Generation* (Nakahira & Kawamoto, 2019), authors Nakahira and Kawamoto outline how GANs such as the MoCoGAN only examine the "color information" of the frames. Nakahira and Kawamoto developed a model that uses depth information in addition to color information which outperforms MoCoGAN on the measures of quality and diversity of the generated videos. To achieve this, the model draws on the geometrical contents of a given scene in order to generate depth videos. We aimed to emulate parts of this work in order to produce dance videos of relatively high quality and diversity, while introducing a new motion discriminator.

In *Everybody Dance Now* (Chan, et al., 2019), Chan et al. created a model to impose the movements of a video of one person dancing onto another person who performed different motion such that they appear to be dancing in sync when compared side-by-side. They utilized pose detector, specifically OpenPose, to generate stick-figure representation of the dance. The clips used in this research only featured a single dancer, while our clips involve often multiple dancers in a single frame, requiring a deeper representation of contiguous motion for multiple figures.

3 Dataset and Features

We used the Let's Dance Dataset (Castro, et al., 2016), which contains a collection of video frames of people performing different categories of dance. The dancer(s) are wearing performance dress in some of the videos and normal, practicing clothes in others.

We used 15 classes (ballet, break, chacha, flamenco, foxtrot, jive, latin, pasodoble, quickstep, rumba, square, swing, tango, tap, waltz), with class distributions listed in Table 1 in the Appendix. We limited the length of the input clips to 32 frames instead of the 250-300 frames in the original dance dataset. As a preprocessing step, we first filtered the frames to only include 1080p (1920x1080 px) images. We then cropped these frames to 1080x1080 px, using the upper leftmost 1080x1080. After cropping the frames, we horizontally concatenated all of the frames from a given video into a single image with width of 34560 pixels (1080 times 32) and height of 1080 pixels.

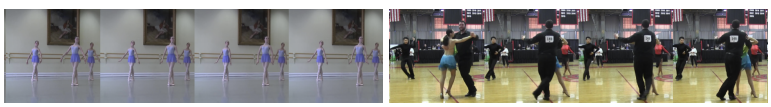

Figure 1: Samples of four consecutive frames from horizontally concatenated training examples

The output data from the model is in the same format as the input, with each video represented as a single horizontal long image, of size 34560x1080 px to be converted to a gif or mp4 format.

4 Methods

We initially pursued future frame prediction to compile an entire video, specifically employing the structure outlined in Mathieu, Couprie, & LeCun. We determined that it would be more productive to use a conditional GAN to produce the entire frame sequence without specific preceding image input. We transitioned away from the frame prediction to a different video GAN model. DNCEGAN is represented by five networks, a recurrent neural network, a generator, and three discriminators.

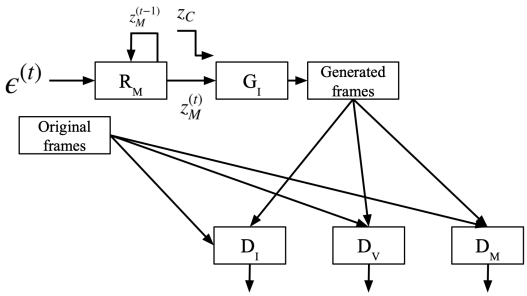

Figure 2: DNCEGAN Architecture

Figure 3: Frame 1, Frame 2, Difference Frame

4.1 Generator

Similar to in MoCoGAN, the generator (G_I) creates new frame sequences by mapping vectors in the latent image space, $z^{(t)}$ to a sequence of images. The generator focuses on the content of the images, which are sampled as z_c from a Guassian distribution with mean 0, and cannot perceive motion. Thus the recurrent neural network, R_M integrates later feedback from the discriminators regarding motion. R_M learns about the different types of motion provided in the training data set. R_M learns to generate a sequence of motion codes $\boldsymbol{Z_M}$from a sequence of noise inputs for a time point t, represented as $\epsilon^{(t)}$ This understanding of different motion representations helps the generator properly order sequential frames to feed into the discriminators.

4.2 Discriminators

The first discriminator, D_I, provides feedback to G_I based on individual images, and determines if an image if from a real or generated sequence of frames. The second discriminator, D_V, determines if a video clip is real or comes from a generated clip. The discriminator incorporates a spatio-temporal architecture to provide feedback about the time sequence video information.

Our novel discriminator D_M helps G_I and R_M encodes additional motion information, beyond that captured in the previous two discriminators. We captured information about additional motion and an increased number of figures by examining sets of consecutive frames and determining the pixel-wise distance between these frames. More specifically, we take two frames that separated by one time step, like Image 1 and Image 2 in Figure 3, and create a "difference" image in which every pixel is the result of the subtraction of the color values of that pixel in the second image from the first image. D_M then tries to determine if the motion capture is from a real or generated video set.

4.3 One-Hot Conditional Vector Representation

For the conditional implementation, each dance category is represented as a one hot vector for the class label. This one hot category vector is connected to the content and motion vectors for the specific class video.

4.4 Objective Function

DNCEGAN's conditional objective function is defined by

$$\max_{G_I,R_M}\ \min_{D_I,D_V,D_M} V(G_I,R_M,D_I,D_V,D_M,Q)$$

$$\begin{aligned} V(G_I,R_M,D_I,D_V,D_M,Q) = {} & \mathbb{E}_{\mathbf{v}p_{data}}[-logD_I(S_1(\mathbf{v}))] + \mathbb{E}_{\tilde{\mathbf{v}}p_{gen}}[-log(1-D_I(S_1(\tilde{\mathbf{v}})))]+ \\ & \mathbb{E}_{\mathbf{v}p_{data}}[-logD_V(S_2(\mathbf{v}))] + \mathbb{E}_{\tilde{\mathbf{v}}p_{gen}}[-log(1-D_V(S_2(\tilde{\mathbf{v}})))]+ \\ & \mathbb{E}_{\mathbf{v}p_{data}}[-logD_M(S_3(\mathbf{v}))] + \mathbb{E}_{\tilde{\mathbf{v}}p_{gen}}[-log(1-D_M(S_3(\tilde{\mathbf{v}})))] \\ & \lambda I(G_I,A) \end{aligned} \tag{1}$$

where S_1 provides a random video frame, S_2 provides a random consecutive frames from a video, S_3 returns random consecutive motion representation images, I is the lower bound on mutual information between generated video clips and the categorical variable representation, and A is an approximation of the dance presence conditioned on the input video frame sequence.

5 Experiments

We conducted multiple experiments in order to evaluate our novel approach of adding a 3rd discriminator to map out the motion subspace. We first trained MoCoGAN to generate dance videos only in the ballet category. Then, we trained a separate MoCoGAN that was conditional this time, trained on all 15 different dance categories in order to see its performance on our particular dataset. Finally, we trained DNCEGAN conditionally on all 15 different dance categories with the additional motion discriminator. This allowed us to see the benefits of conditional GANs with regards to our highly complex dataset and to compare DNCEGAN to MoCoGAN both qualitatively and quantitatively.

For our conditional GAN, we used Adam optimization for gradient descent with learning rate of 0.002 and momentums of 0.5 and 0.99. We selected a low learning rate because we wanted to ensure that the GAN did not miss the loss minimum and overshoot. We used an image batch, motion batch, and video batch size of 32 because each of the videos consists of 32 frames. The code for our project can be found at https://github.com/aglettiere/CS230-Final-Project.

It is challenging to compare different generative adversarial networks' performances to each other, especially for video GANs. Therefore, in this study, we use the inception score metric to evaluate the quality of frames from the original and generated videos (Salimans, et al. 2016). Inception score is an automatic method to evaluate samples and seeks to reflect human qualitative evaluation. As shown below, the inception score is calculated by applying two requirements: containing meaningful objects and diversity in the images.

$$IS = exp(\mathbb{E}_x KL(p(y|x)||p(y)))$$

Since evaluating generative models is widely known to be a difficult task, we also calculated the inception score on the preprocessed dataset, which is what the models received as input, to have a point of comparison.

6 Results and Discussion

All of the inception scores, noted in Table 2 in the Appendix, are generally low because inception score primarily values diversity and variety in the images of the generator. When calculating the inception score, we examined all of the produced class frames and took an average over all of the frames or images. This means that the inception score was calculated over a total of (32 videos * 32 frames/video) = 1024 frames. Because 32 frames from any one video may be relatively similar, the videos in general (both the original and the generated) have lower inception scores.

Figure 4: DNCEGAN generated videos for categories: chacha, pasodoble, tango, waltz, tap, swing (clockwise order)

Examining the results, it appears that our hypothesis about how to properly adjust the discriminators for increased movement was only partially correct. DNCEGAN does appear to capture highly complex motion on a multi-person level, as shown in the frames produced in Figure 4. Nonetheless, the overall mean inception score for the videos produced by Conditional DNCEGAN, listed in Table 2, is only 60% of the overall mean inception score on the original dataset. Our DNCEGAN images are fairly diverse in color and content; however, we believe that the model performs less well in regards to the evaluation of quality. We believe this occurs specifically because the inception score determines quality based on a correlate with human perception of quality and as seen in the videos generated by DNCEGAN, sometimes the setting and the dancers are hard to discern. Ultimately, the DNCEGAN frames are not as clearly defined as the original frames, explaining the discrepancy in inception scores.

The videos produced in the 15 dance categories provided additional insight on the conditional nature of the "overlapping" dance categories. DNCEGAN generated videos often shared similar features such as background features or positioning of people in the frames. For example, the videos categorized as pasodoble and tango in Figure 6 share similar coloring as well as framing of dancers and audience. Since we discovered that there were

only minor differences in backgrounds between the generated frames for the different classes, we believe that the generator was learning larger trends about the dance data, heavily influenced by overarching trends in the dataset. The similarities amongst the videos produced in the 15 dance classes may have occurred because during the early training epochs of DNCEGAN, the generator was trying to simultaneously manage producing quality frames and extracting specific motion information to connect the frames. The tuning of motion production specifically involved the ability of the motion and video discriminators to provide feedback to the generator about proper sequences of video frames. It's interesting to note that while the different categories in the dataset had a wider variety of inception scores, the videos produced by DNCEGAN all generally lie near each other regardless of dance category, an indication of this blending of dance types.

Figure 5: MoCoGAN sequential concatenated frames of generated ballet videos after 10,000 training iterations

Figure 6: DNCEGAN generated videos for the ballet category after 10,000 training iterations

More specifically, when we trained DNCEGAN exclusively on the ballet dataset, the resulting frames contained common features–see figure 6 and the video labeled swing in Figure 4. Our first experiment with MoCoGAN was able to learn the generalized ballet outfits and patterns (Figure 5). But when we added in the additional 14 classes, the generated ballet frames contained more general ballroom dance patterns, outfits, and backgrounds, indicating that the overall dataset may have influenced the class-based production (Figure 6). This is attributed to the fact that while ballet videos in the dataset as reflected by MoCoGAN's generated videos were often on a stage, which helped isolate and focus on the dancers in question while for other types of dances, the setting of the videos were noisy, filled with audience members in a ballroom. The dances themselves may have had "overlapping" styles but the variety of backgrounds harmed the quality of the ballet videos generated by the conditional DNCEGAN. The mean DNCEGAN ballet inception score is 0.48 lower than the mean MoCoGAN ballet inception score, yet the gap in inception score between DNCEGAN ballet category and MoCoGAN ballet is much smaller than that between the original dataset and MoCoGAN ballet. This trend was also reflected in the loss values achieved by both models. The generator losses for both MoCoGAN (Appendix Figure 7) and DNCEGAN (Appendix Figure 8) experienced a similar change, increasing after an initial sharp decrease, indicating that the generator for both models was struggling to produce content that would fool the discriminator. But after 10,000 iterations, MoCoGAN's generator loss was 9.25, the image discriminator loss was 0.29, and the video discriminator loss was 0.069. Conversely, DNCEGAN's generator loss was 15.16, the image discriminator loss was 0.65, the video discriminator loss was 0.036, and the motion discriminator loss was 1.13.

Ultimately, the clips produced, while lacking fine grained pixel precision, look like a potential collection of individuals dancing. DNCEGAN appears to have been borrowing themes from other categories when producing class-specific content. Furthermore, the generator may have potentially been learning patterns about dance backgrounds, using this information to discriminate between the classes instead of differentiating based on the type of motion. Adding in a larger variety of training samples may help remediate this potential class-specific overfitting in the future.

7 Conclusion and Future Work

We proposed the DNCEGAN network to use additional motion information and conditional class information to generate dance-type specific videos. Our project was motivated by the MoCoGAN, however we proposed a new design of the architecture that incorporates an additional discriminator and class distinctions with hopes of encoding better information for dances with high motion volume. While our proposed method did not produce a similar level of precision of results as MoCoGAN, we believe that this exploration provided a starting step for developing models that can accommodate larger levels of motion. We believe that the network in DNCEGAN may not have been able to learn the desired motion information during training time, producing a more scattered variety of results. We deterined that capturing background noise, specifically from a colorfully decorated dance platform or a crowd of spectators is an extremely difficult task, especially when integrating conditional information because the background noise in the dataset appeared to be connected to the type of dance the data originated from. In the future, we would like to explore additional datasets with a similar quantity of motion and number of figures as in the Let's Dance Dataset. We would also like to further investigate potential hyperparameter variations to see if we can improve DNCEGAN inception score results.

8 Appendix

dance	ballet	break	chacha	flam.	foxtrot	jive	latin	pas.	quickst.	rumba	square	swing	tango	tap	waltz
videos	48	56	48	49	47	70	42	58	39	50	63	59	46	52	75

Table 1: Number of Training Samples Per Dance Category

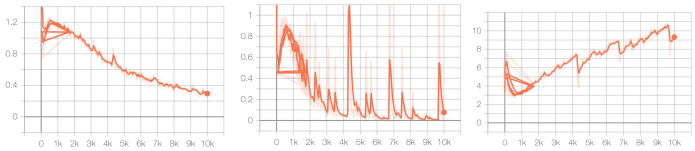

Figure 7: MoCoGAN: Image Discriminator Loss vs. Training Iteration, Video Discriminator Loss vs. Training Iteration, Generator Loss vs. Training Iteration (In order)

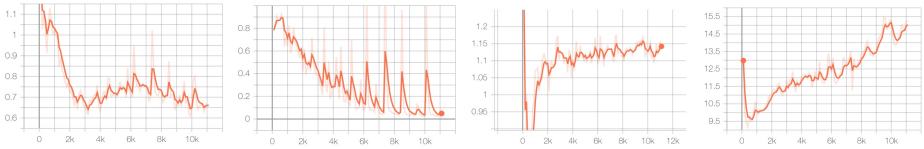

Figure 8: DNCEGAN: Image Discriminator Loss vs. Training Iteration, Video Discriminator Loss vs. Training Iteration, Motion Discriminator Loss vs. Training Iteration, Generator Loss vs. Training Iteration (In order)

	Dataset	MoCoGAN Ballet	Conditional DNCEGAN
ballet	4.27 ± 0.28	3.12 ± 0.16	2.64 ± 0.17
break	4.49 ± 0.22	-	2.53 ± 0.18
chacha	3.48 ± 0.14	-	2.37 ± 0.18
flamenco	4.89 ± 0.36	-	2.36 ± 0.09
foxtrot	3.79 ± 0.26	-	2.32 ± 0.12
jive	3.95 ± 0.23	-	2.28 ± 0.14
latin	4.75 ± 0.23	-	2.29 ± 0.11
pasodoble	3.65 ± 0.17	-	2.37 ± 0.17
quickstep	3.85 ± 0.19	-	2.33 ± 0.13
rumba	3.43 ± 0.14	-	2.27 ± 0.14
square	4.41 ± 0.25	-	2.69 ± 0.10
swing	3.95 ± 0.19	-	2.52 ± 0.15
tango	2.76 ± 0.06	-	2.33 ± 0.10
tap	5.79 ± 0.48	-	2.46 ± 0.16
waltz	2.76 ± 0.13	-	2.45 ± 0.21
overall	4.01 ± 0.29	3.12 ± 0.16	2.45 ± 0.20

Table 2: Inception scores for original data and models trained on Let's Dance Dataset

Figure 9: Ballet videos generated at 500 (upper left), 1500 (upper right), 6000 (lower left), and 10,000 (lower right) training iterations

9 Contributions

We worked in parallel on most components of the project. We found that it was most productive for our learning to open the EC2 instance and work on data manipulation tasks and model production together.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016). Tensorflow: A system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16) (pp. 265-283).

Castro, D., Hickson, S., Sangkloy, P., Mittal, B., Dai, S., Hays, J., & Essa, I. (2018). Let's Dance: Learning From Online Dance Videos. arXiv preprint arXiv:1801.07388.

Chan, C., Ginosar, S., Zhou, T., & Efros, A. A. (2019). Everybody dance now. In Proceedings of the IEEE International Conference on Computer Vision (pp. 5933-5942).

Dyelax. "Dyelax/Adversarial_Video_Generation." GitHub, 5 May 2017, github.com/dyelax/Adversarial_Video_Generation.

Gauthier, J. (2014). Conditional generative adversarial nets for convolutional face generation. Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester, 2014(5), 2.

Lee, A. X., Zhang, R., Ebert, F., Abbeel, P., Finn, C., & Levine, S. (2018). Stochastic adversarial video prediction. arXiv preprint arXiv:1804.01523.

Mathieu, M., Couprie, C., & LeCun, Y. (2015). Deep multi-scale video prediction beyond mean square error. arXiv preprint arXiv:1511.05440.

Mirza, M., Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

Nakahira, Y., & Kawamoto, K. (2019, September). DCVGAN: Depth Conditional Video Generation. In 2019 IEEE International Conference on Image Processing (ICIP) (pp. 749-753). IEEE.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... & Lerer, A. (2017). Automatic differentiation in pytorch.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X. (2016). Improved techniques for training gans. In Advances in neural information processing systems (pp. 2234-2242).

Sergeytulyakov. "Sergeytulyakov/Mocogan." GitHub, 30 Jan. 2019, github.com/sergeytulyakov/mocogan.

Tulyakov, S., Liu, M. Y., Yang, X., & Kautz, J. (2018). Mocogan: Decomposing motion and content for video generation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1526-1535).