V3GAN: Decomposing Background, Foreground and Motion for Video Generation

Arti Keshari
cs19s008@cse.iitm.ac.in
Sonam Gupta
cs18d005@cse.iitm.ac.in
Sukhendu Das
sdas@iitm.ac.in

Visualization and Perception Lab
Dept. of CSE
Indian Institute of Technology, Madras
Chennai, India

Abstract

Video generation is a challenging task that requires modeling plausible spatial and temporal dynamics in a video. Inspired by how humans perceive a video by grouping a scene into moving and stationary components, we propose a method that decomposes the task of video generation into the synthesis of foreground, background and motion. Foreground and background together describe the appearance, whereas motion specifies how the foreground moves in a video over time. We propose V3GAN, a novel three-branch generative adversarial network where two branches model foreground and background information, while the third branch models the temporal information without any supervision. The foreground branch is augmented with our novel feature-level masking layer that aids in learning an accurate mask for foreground and background separation. To encourage motion consistency, we further propose a shuffling loss for the video discriminator. Extensive quantitative and qualitative analysis on synthetic as well as real-world benchmark datasets demonstrates that V3GAN outperforms the state-of-the-art methods by a significant margin.

1 Introduction

Unsupervised feature representation learning from unlabeled data has been a problem of great interest in Computer Vision. Other tasks like classification, clustering, etc. can benefit from the knowledge of content and dynamics present in the learned feature representation. Deep Generative models have achieved great success in unsupervised learning by generating images [2, 5, 11, 20] from latent noise vectors. In contrast, a similar level of success has not yet been achieved in video generation. This is primarily because the video data is more complex due to the presence of the temporal dimension. For generating photorealistic videos, the model must learn the abstraction of different objects along with the evolution of their motion over time.

Many of the existing works [21, 26, 27, 30] have proposed Generative adversarial networks (GANs) to address this task. However, the videos generated by these models are still subpar from the real videos, specifically for complex datasets like UCF101. VGAN

© 2021. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

[27] decomposed the task of video generation into foreground and background, but it lacked motion consistency. MoCoGAN [26] and G3AN [30] disentangled motion and appearance in the latent space but suffered in visual quality because they focus on the foreground and background together. Studies have shown that infants learn the physical dynamics in an unsupervised way by connecting moving things as a single object and things moving separately from one another as multiple objects [25]. We utilize this idea and propose a novel generative method, V3GAN, which divides the task of video generation into foreground, background and motion generation, as illustrated in figure 1 (left). Here, the topmost branch generates the motion; the middle and bottom branches learn to separate the foreground and background using the notion of a mask. We argue that the content of a video can be described using these three key components. The foreground provides information about the main object(s) in the video, the background informs about where they are, and the motion says what they are doing.

To maintain the frame quality and motion consistency in the generated video, similar to existing works, we also use image and video discriminators, as shown in figure 1 (left). Despite using these, motion across the frames of a video is not smooth. These inconsistencies can be reduced if the video discriminator can pay attention to the temporal information. To enforce this, we propose a shuffling loss where the discriminator tries to distinguish between the real video and shuffled video. We empirically demonstrate the efficiency of the proposed method by evaluating our method on a synthetic dataset (Shapes) and real-world datasets (Weizmann Action, UCF101). To summarize, the key contributions of the proposed method are as follows:(i) a novel framework V3GAN, which maps latent noise vectors to the background, foreground and motion for video generation. (ii) a novel feature-level masking layer that learns the mask for intermediate convolution features to obtain a refined foreground mask. (iii) a novel shuffling loss for video discriminator which complements the 3D convolution-based video discriminator by penalizing the incorrect order of frames irrespective of how realistic the individual frames are.

2 Related Work

Limited work has been done in video generation problems because of high computation requirements and enormous possibilities of variations in a video. A generated video may vary in content, speed, color, and intensity, but it has to be highly correlated along the temporal dimension. To tackle this problem, existing works have tried VAEs [13] and GAN [5] based architectures. To reduce the complexity of the video generation task, many existing approaches have focused on conditional video generation. For instance, the tasks of future frame prediction [14, 17, 32], video generation from single image [15], video interpolation [1, 16] are all conditioned on images. Presence of appearance and structure information make these tasks simpler from unconditional video generation where the input is a noise vector. Some of these methods also use additional cues like human keypoints [3, 10], optical flow [15] etc. learned implicitly or explicitly from the input data.

Unconditional Video Generation: Early video generation approaches [27] use basic 3D spatio-temporal convolution networks to capture spatial as well as temporal information. VGAN [27] attempted to disentangle the background and foreground with two-stream 3D convolution architecture. TGAN [21] proposed to use two generators. Temporal generator outputs a sequence of noise vectors corresponding to the frames in a video. The image generator then maps these noise vectors to the RGB frames. MoCoGAN [26] extends TGAN

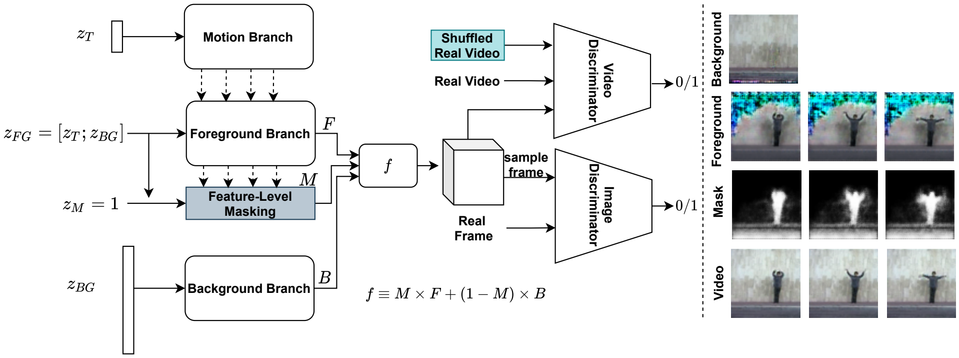

Figure 1: Left: Overview of proposed V3GAN architecture. Right: Illustration of background and foreground, the mask estimated, and the final generated video frames obtained by combining the three former components.

by replacing the temporal generator with the recurrent GRU network. It further introduced another noise called content noise and tried to disentangle the content (appearance) and motion in a video. Learning motion information in latent space makes the video inconsistent temporally. Few works [4, 8] have also tried object-centric approaches. Follow-up work of [26], G3AN [30] proposed a three-stream generator which takes two noise vectors for motion and appearance, respectively. Third stream takes the concatenation of both noise vectors and passes it through spatio-temporal network to generate the output video. Disentangling the appearance and motion is difficult for complex real-world data like UCF101. Thus, methods like MoCoGAN, G3AN fail to generate high-quality videos for such distribution, see fig. 4.

We observe that decomposing the video generation into foreground, background and motion can significantly reduce the learning complexity, as the model now focuses on learning the temporal dynamics only for the moving object. We propose a framework that generates foreground, background and motion simultaneously to create high-quality diverse videos from the underlying data distribution. Closest to our work in the literature is VGAN [27]. But unlike VGAN which shares the weight of foreground and mask generation network except the last layer, our model decomposes foreground and background at the feature level as well and propagates the mask to the next layer. This enables the network to generate good quality foreground masks.

Several works have used the shuffle based self-supervision [18, 28, 29] on videos as it does not require manual annotation. In [29], Wang and Gupta proposed a Siamese network [18] trained to sort the input sequence in correct order. Wang et al. in [28] proposed a shuffle discriminator which learned whether the input optical flow maps generated by the model are shuffled or not. Unlike [28], we propose a shuffling loss which is applied to the raw RGB images from the training data. Shuffled real videos helps the discriminator to focus on temporal information even if the individual frames are realistic.

3 Proposed Method

We propose a three-branch deep generative model that addresses the problem of video generation. The model itself splits the problem into three sub-tasks, namely, foreground generation

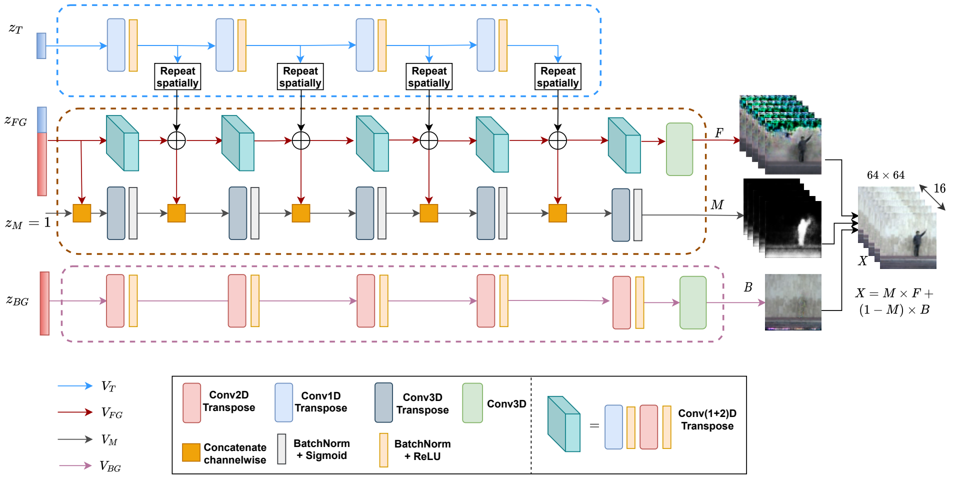

Figure 2: Architecture diagram of proposed V3GAN. V_T, V_{FG}, V_{BG} correspond to temporal, foreground, and background branches shown in blue, red, pink respectively. V_M is the feature level masking layer shown in black. Input z_T and z_{BG} are sampled from the Gaussian noise.

along with the foreground mask, background generation, and motion modelling. For learning both spatial and spatio-temporal dynamics, the model uses a frame discriminator and a video discriminator. We propose to use shuffling loss so that even small temporal inconsistencies in the model are penalized. In section 3.1, we will elaborate upon the Generator of V3GAN framework along with the proposed feature level masking layer. In section 3.2, we discuss about the discriminator. In Section 3.3, we present the proposed shuffling loss and the strategy used to generate the shuffled video. Lastly, in Section 3.4, we discuss the objective function used for training V3GAN.

3.1 Generator

V3GAN generator consists of three branches, which are the temporal branch V_T, the foreground branch V_{FG} augmented with the feature level masking layer V_M and the background branch V_{BG} as shown in figure 2. The inputs to the generator are two noise vectors z_{BG} and z_T corresponding to the processing pipeline of background and motion respectively. We notice that foreground is highly entangled with the motion, and all possible combinations of foreground and backgrounds are also not semantically meaningful. For instance, a surfer surfing on the road is an unrealistic scenario. Therefore, the foreground noise vector is chosen to be the concatenation of z_{BG} and z_T.

The choice of convolution operations is one of the key elements to enforce that each branch learns unique features. Given the assumption that there is no appreciable camera motion, i.e., the foreground object is the moving component, the background can be treated as a fixed $2D$ bitmap layer shared over the entire video. Hence, the background branch V_{BG} is designed to be independent of other branches. V_{BG} upsamples the input only spatially using the Conv2D transpose layer. The output of V_{BG} is a single $2D$ image which corresponds to the background of the generated video.

The V_T branch is tightly coupled with the V_{FG} branch, as the moving foreground will

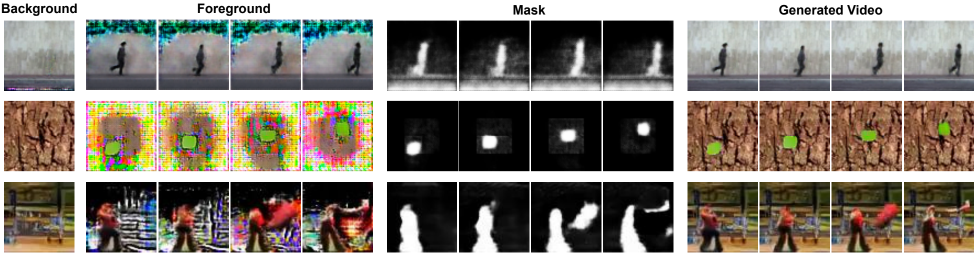

Figure 3: From left to right: Illustrations of background, foreground, mask; and the final generated video obtained by combining these components on Weizmann (top), Shapes (middle), UCF101 (bottom) datasets.

be highly correlated with the temporal dynamics. We use a noise vector z_T as input which evolves in temporal dimension by upsampling the input with Conv1D transpose layers. This branch learns the global motion information and guides the foreground stream V_{FG} by providing clues on various motion aspects like how to move, with what speed, direction of motion, etc. The output of each Conv1D transpose layer is repeated spatially to match the dimension of the foreground features. The resulting features are then combined with the foreground features using element-wise addition.

Foreground branch V_{FG} seeks to generate the spatio-temporal features. This branch consists of five Conv(1+2)D transpose layers. Conv(1+2)D operation [31] is factorized form of Conv3D transpose which is Conv1D transpose followed by Conv2D transpose. Conv1D transpose helps the foreground to learn the temporal component and Conv2D transpose helps to learn the spatial content of the foreground. The output features of Conv(1+2)D layer are added with the corresponding temporal features. The resulting features are then concatenated with the feature level masking layer as shown in figure 2. Conv3D layer is introduced at the output of V_{FG} to get the desired foreground in RGB domain from the learned features.

Feature level masking: Another key element for achieving the successful decomposition of foreground and background is the quality and accuracy of the generated foreground mask. Inspired by [27], we attempt to learn the mask from the foreground branch V_{FG} in an unsupervised manner by using the following relationship between the foreground (F), background (B) and mask (M) for video X.

$$X_i = M_i \times F_i + (1 - M_i) \times B_i, \qquad M_i \in [0,1] \tag{1}$$

where X_i, M_i, F_i and B_i are the i^{th} pixel in the video, mask, foreground and background respectively.

The input to the masking layer is a fixed vector $z_M = 1$ concatenated with z_{FG} along the channel dimension. Considering $z_M = 1$ implies that z_{FG} completely belongs to the foreground, which enforces the branch V_{FG} to focus on foreground features. The implicit assumption here is that at the initial layer, all channels of z_{FG} represent the foreground. The input is then passed through five Conv3D transpose layers to learn the spatio-temporal features. Each intermediate masking feature is concatenated with foreground features. Output of each stage has a single channel with sigmoid activation function. Empirical results show that feature level masking improves the performance of the video generation substantially. The evaluation results are reported later in section 4.

3.2 Discriminator

V3GAN consists of two discriminators similar to MoCoGAN [26], a video discriminator and an image discriminator. Unlike MoCoGAN, we also pass shuffled video as input to video discriminator. Video discriminator contains Conv(2+1)D layers and the image discriminator contains Conv2D layers. We used spectral normalization [19] in the discriminator as well as in generator to stabilize the training.

3.3 Shuffling Loss

The video discriminator D_V uses (2+1)D convolution to focus on both spatial and temporal aspects of the input. We propose a shuffling loss that exploits the sequential ordering of frames of a video. This loss enforces D_V to emphasize on the correctness of motion. The knowledge of incorrect ordering of frames vs. the correct ordering of frames is imparted to D_V, explicitly, by training it to classify a shuffled real video sequence as fake. Let sh be the shuffling function which takes the real video X and shuffling parameter α as input to generate the shuffled video that is hard for the discriminator to classify. The shuffling loss is then defined as follows:

$$L_{shuffle}(X,\alpha) = E[\log(1 - D_V(sh(X,\alpha))] \tag{2}$$

The shuffling parameter controls the fraction of frames to be shuffled. The function sh is defined as follows:
Step 1: Select αN frames uniformly at random from the real video where N (here 16) is the length of the video. **Step 2:** Swap a selected frame with its first or second neighbouring frame. The empirical analysis for shuffling loss can be found in Section 4.3.

3.4 Loss Function

The feedback from both D_V and D_I are used to train the network. Both D_V and D_I use adversarial loss functions as proposed in [20]. The learning problem of V3GAN is given in eq. 3 where F_I is the loss function of image discriminator and F_V is the loss function of video discriminator. The definitions for these losses are given as:

$$\min_{G} \max_{D_I, D_V} (F_I(G, D_I) + F_V(G, D_V)) \tag{3}$$

$$F_I = E[\log D_I(X_f)] + E[\log(1 - D_I(G(z_{BG}, z_T)_f))] \tag{4}$$

$$F_V = E[\log D_V(X)] + E[\log(1 - D_V(G(z_{BG}, z_T)))] + L_{shuffle}(X,\alpha) \tag{5}$$

G represents the generator. Subscript f in the loss function of D_I refers to the randomly sampled frame from the video. $L_{shuffle}$ represents the shuffling loss as described in eq. 2.

4 Experiments

We evaluate our method on the following datasets:

Shapes Dataset [26] is a synthetic dataset, containing 4000 videos of circles and squares of different colors and sizes on a black background. To understand the ability of the model to decompose foreground, background and motion, we randomly sample a texture as background from 7 different textures and apply it to each moving shape video.

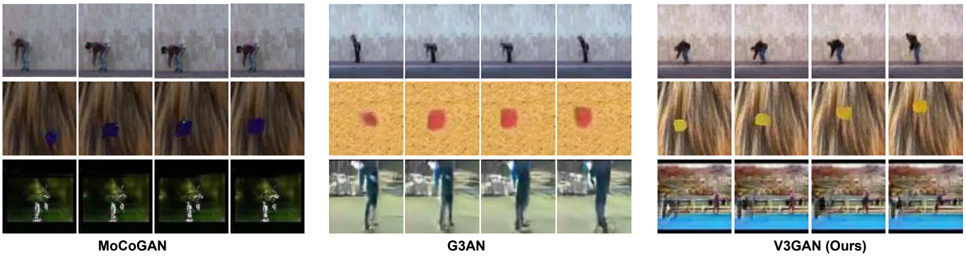

Figure 4: Qualitative comparison of performance with state-of-the-art methods: MoCoGAN and G3AN, on: Weizmann (top), Shapes (middle) and UCF101 (bottom) datasets. For the sake of illustration, frames are sampled at equal intervals from the video.

Weizmann Action Dataset [6] contains 93 videos of 9 people performing 10 actions, including running, jumping jack, etc. We augment the data by horizontal flipping.

UCF101 [24] is a commonly used dataset for video generation, containing more complex and variety of videos. It includes 13,220 videos of 101 different action categories. Since UCF101 contains some videos with moving background, we stabilize the camera motion by adopting the stabilization process as in [27]. For both Action and UCF101 datasets, the video frames are rescaled to 85x64 then centre cropped to 64x64, same as in [21].

Implementation Details: For all our experiments, the dimension of z_{BG} is set to 128 and that of z_T is set to 10. Shuffling parameter is set to $\alpha = 0.5$ and batch size is 16. The input videos contain 16 frames with resolution of 64×64. Adam optimizer [12] is used to train the network with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. Learning rate for generator and discriminators is set to 10^{-4}. The details of the network architecture design can be found in the supplementary document (SD). Source code will be made public.

4.1 Quantitative Evaluation

We compare the proposed method quantitatively with four state-of-the-art (SOTA) methods, VGAN, TGAN, MoCoGAN and G3AN using Frechet Inception Distance (FID) [9] and Inception Score(IS) [23]. FID metric is the squared Wasserstein distance between two multidimensional Gaussian distributions: $\mathcal{N}(\mu, \Sigma)$. We used a deep 3d CNN network of [7] to extract the mean and covariance of the distributions, which is then used to calculate the FID metric using $\text{FID} = |\mu - \mu_w|^2 + \text{tr}(\Sigma + \Sigma_w - 2(\Sigma\Sigma_w)^{1/2})$. Lower FID indicates better quality of the generated videos. For evaluation, we generated 5000 video samples using the trained model and calculate FID value. IS is the KL divergence between class conditional and marginal probability distribution $\exp(E_{x \sim p_g}(KL(p(y|x)||p(y))))$. Since the inception model must be pretrained on the data for which IS is calculated, we reported IS only for UCF101.

Table 1 contains the FID and IS for four prior published works along with ours. Our method outperforms all other baseline methods suggesting that the learned data distribution is closer to the real data distribution. V3GAN outperforms the SOTA even without using shuffling loss indicating that the decomposed representation is able to learn the spatio-temporal dynamics better. We note that, for Shapes dataset, the FID of our method is significantly lower than G3AN model. On visualizing the generated videos using both methods, we found that the videos generated by G3AN lack diversity. It generates a combination of only 3 backgrounds out of 7, whereas V3GAN generates videos with all 7 backgrounds. Moreover,

Method	Shapes	Weizmann	UCF101		UCF101 (Stable)
	(FID↓)	(FID↓)	(FID↓)	(IS↑)	(FID↓)
VGAN [27]	-	158.04	115.06	2.94	-
TGAN [21]	-	99.85	110.58	2.74	-
MoCoGAN [26]	144.87†	92.18	104.14	3.06	218.59†
G3AN [30]	-	86.01	91.21	3.62	-
G3AN*	168†	68.19†	86.73†	3.44	102.13†
V3GAN (w/o shuffle)	62.59	64.33	**78.71**	3.84	75.64
V3GAN + shuffle (Ours)	**28.07**	**62.65**	80.18	**3.88**	**74.36**

Table 1: Quantitative comparison with SOTA methods using FID metric for Shapes, Action, UCF101 including stabilized UCF101 datasets. † indicates that the values are obtained by training the official codes provided by the authors. G3AN* contains value after running official code with modified discriminator, which uses Conv(2+1)D instead of Conv3D.

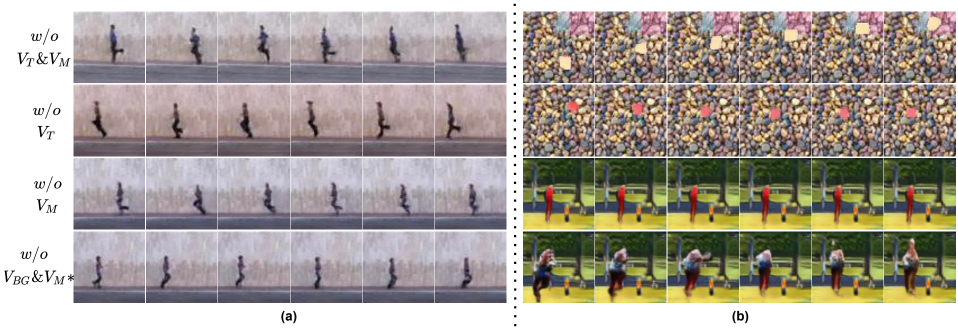

Figure 5: (a) Ablation study: Generated videos obtained by removing each branch of V3GAN. (b) Examples of videos generated by V3GAN on Shapes (first two rows and UCF101 (last two rows) datasets.

by design, V3GAN uses the same background over all frames in a video leading to a much better temporal consistency. Since the background of synthetic data does not contain illumination, shadows and other inherent variations over frames, it gives an impetus to our model for a drastic improvement for Shapes dataset. UCF101 dataset contains small background motion. Therefore, we have trained our model on the original data as well as stabilized data. Our method outperforms SOTA methods in both scenarios. In the last row of table 1, we show the effect of using the proposed shuffling loss. It is evident that shuffling improves the temporal consistency of the video resulting in better FID values on all datasets.

4.2 Qualitative Evaluation

In figure 3, we show the background, foreground and mask generated by our model along with the generated video for the three datasets. Our network is able to decompose the foreground and background with the help of the mask without any supervision. To verify the improvement in the visual quality of the generated videos, we compare our results with that of MoCoGAN and G3AN qualitatively. Figure 4 shows that, our method is able to generate in-line appearance of the front content. In the first row, the hands of the person are clearly

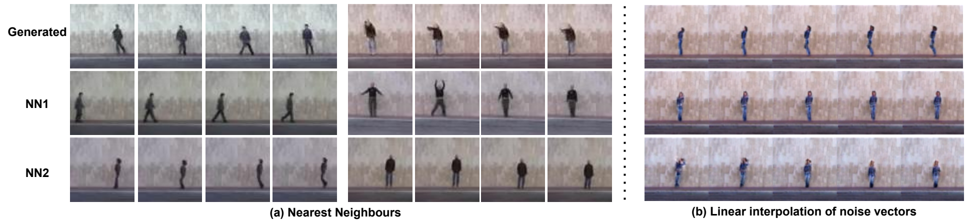

Figure 6: **(a) Nearest neighbours:** Top row contains generated video sequences. NN1 and NN2 represent first and second nearest neighbours of the generated video. **(b) Linear interpolation:** Top and bottom rows represent video generated from two noise vectors. Middle row is generated from an intermediate noise vector between them.

Architecture	**Shapes (FID↓)**	**Weizmann (FID↓)**	**UCF101 (FID↓)**
w/o V_T & V_M	40.93	73.44	81.60
w/o V_{BG} & V_M*	168.56	70.9	198.37
w/o V_M	34.19	74.45	76.33
w/o V_T	66.5	70.68	80.29
V3GAN (Ours)	**28.07**	**62.65**	**74.36**

Table 2: Effect of different components of V3GAN, studied for all three datasets. Here, V_M* means masking has also been removed from the last layer.

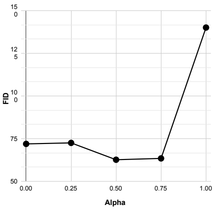

Figure 7: Effect of changing α in shuffling loss, on FID value for Action Dataset.

visible, and in the second row the shape of the rectangle is maintained. In the third row, the background is of good quality. Although there is scope of improvement in the foreground, but it is modelled better than the existing methods. Figure 5 (b) shows the videos generated on Shapes and UCF101 datasets by keeping the background noise fixed and only altering the foreground noise. It shows that motion pattern is highly correlated with the foreground.

Nearest neighbours and Linear interpolation Analysis: As the Weizmann dataset has a small size, we further examine whether or not the proposed method has memorized the training data using nearest neighbour and linear interpolation similar to [1, 22]. To find the nearest neighbours, we computed the euclidean distance between the feature embeddings of the generated videos with that of the training dataset. From figure 6 (a), we observe that the first two nearest neighbours chosen from the training dataset are distinct from the generated samples. In case of interpolation, we generate video corresponding to the intermediate noise vectors between two sampled noise vectors. It can be seen from figure 6 (b) that there is a graceful transition between videos with change in input noise. Above observations suggest that the network has not memorized the training dataset.

4.3 Additional Analysis

Ablation study: In table 2, we show the importance of each component of our architecture by removing different branches. We first remove V_T branch and V_M layer while retaining the mask generation at the last layer of the model. This configuration leads to a significant drop in performance proving that both feature level masking and motion branch are important to get good quality videos. We then removed V_{BG} and V_M (including masking at the last

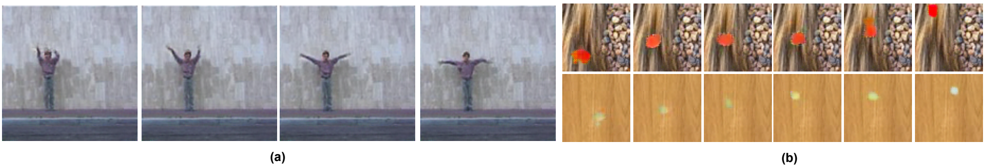

Figure 8: (a) Video generation at 128x128 resolution on Weizmann dataset. (b) Longer video sequence (32 frames) generation on Shapes dataset. Frames are chosen at regular interval for visualization purpose.

layer). After this, V_M and V_T were removed individually. Refer SD for complete architecture. It can be observed in figure 5 (a), that the quality of generated videos degraded when removing each component individually. For Weizmann dataset, removing V_M leads to the worst performance, confirming the contribution of the feature level masking layer. Interestingly, for Shapes and UCF101 dataset, removing V_{BG} and V_M* causes the maximum drop in performance due to mode collapse.

Effect of shuffling parameter (α): It can be seen in figure 7, that the learning is hindered at both lower and higher values of α. The best results are obtained at $\alpha = 0.5$. This means that higher amount of shuffling makes it obvious for the discriminator to identify as fake video. On the other hand, shuffling fewer frames generates difficult examples that contributes to overall learning.

Longer Video Generation and Higher Resolution Video Generation: We marginally modify V3GAN to check the scalability of the proposed method. First, we added one more layer in each branch of our architecture to generate the videos at 128x128 resolution with 16 frames. The qualitative results on Weizmann dataset are demonstrated in figure 8(a). Then, we generated longer videos of 32 frames on Weizmann and Shapes datasets. Figure 8(b) shows the generated samples for Shapes data, while results on Weizmann dataset can be found in the SD. We found that our model can be adopted for higher resolution and longer video generation without significant modifications and parameter overhead.

Class Conditional Video generation: Apart from the above experiments, we also test our model for class conditional video generation on Weizmann dataset. We append one-hot vector that corresponds to the class label with z_T noise vector and find that our model can correctly generate videos for given input class. Qualitative examples can be found in SD.

5 Conclusion

In this work, we have presented a generative model V3GAN, that learns to synthesize videos from a latent Gaussian space by generating background, foreground and motion simultaneously. We have introduced a novel feature masking layer and shuffling loss. Qualitative and quantitative evaluations on three datasets show that our method outperforms all SOTA methods. Ablation studies prove the individual contribution of each component in the architecture towards achieving the best results. The assumption that the background is static works reasonably well, as we target to generate short video clips of only 16 frames duration. For future work, it will be an interesting direction to explore if the background and foreground can be separated irrespective of camera motion, multiple moving objects and clutter.

References

[1] Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3703–3712, 2019.

[2] Andrew Brock, Jeff Donahue, and K. Simonyan. Large scale gan training for high fidelity natural image synthesis. *ICLR*, abs/1809.11096, 2019.

[3] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5933–5942, 2019.

[4] Sébastien Ehrhardt, O. Groth, Aron Monszpart, Martin Engelcke, I. Posner, N. Mitra, and A. Vedaldi. Relate: Physically plausible multi-object scene synthesis using structured latent spaces. *NeurIPS*, abs/2007.01272, 2020.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

[6] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, 2007.

[7] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[8] Paul Henderson and Christoph H. Lampert. Unsupervised object-centric video generation and decomposition in 3d. *NeurIPS*, abs/2007.06705, 2020.

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017.

[10] Yunseok Jang, Gunhee Kim, and Yale Song. Video prediction with appearance and motion conditions. In *International Conference on Machine Learning*, pages 2225–2234. PMLR, 2018.

[11] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *ArXiv preprint arXiv:1710.10196*, 2017.

[12] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, abs/1412.6980, 2015.

[13] Diederik P. Kingma and M. Welling. Auto-encoding variational bayes. *ICLR*, abs/1312.6114, 2014.

[14] Yong-Hoon Kwon and Min-Gyu Park. Predicting future frames using retrospective cycle gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1811–1820, 2019.

[15] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Flow-grounded spatial-temporal video prediction from still images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 600–615, 2018.

[16] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4463–4471, 2017.

[17] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *ICLR*, 2016.

[18] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016.

[19] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *ICLR*, 2018.

[20] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*, abs/1511.06434, 2016.

[21] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2830–2839, 2017.

[22] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan. *International Journal of Computer Vision*, 128:2586–2606, 2020.

[23] Tim Salimans, I. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016.

[24] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *ArXiv preprint arXiv:1212.0402*, 2012.

[25] Elizabeth S Spelke. Principles of object perception. *Cognitive science*, 14(1):29–56, 1990.

[26] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1526–1535, 2018.

[27] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. *Advances in Neural Information Processing Systems*, 29:613–621, 2016.

[28] Junyan Wang, Bingzhang Hu, Yang Long, and Yu Guan. Order matters: Shuffling sequence generation for video prediction. *BMVC*, 2019.

[29] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2015.

[30] Yaohui Wang, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. G3an: Disentangling appearance and motion for video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5264–5273, 2020.

[31] Yaohui Wang, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. Imaginator: Conditional spatio-temporal gan for video generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1160–1169, 2020.

[32] Wei Yu, Yichao Lu, Steve Easterbrook, and Sanja Fidler. Efficient and information-preserving future frame prediction and beyond. In *International Conference on Learning Representations*, 2019.