RV-GAN: Recurrent GAN for Unconditional Video Generation

Sonam Gupta Arti Keshari Sukhendu Das
Visualization and Perception Lab, Department of Computer Science
Engineering, Indian Institute of Technology, Madras, India
cs18d005@cse.iitm.ac.in, cs19s008@cse.iitm.ac.in, sdas@iitm.ac.in

Abstract

Generative models aiming to generate content from noise have achieved high-fidelity synthesis for image data. However, obtaining comparable performance in the field of unconditional video generation still remains challenging. In this work, we propose a recurrent GAN architecture to model the high-dimensional video data distribution. Recurrent networks by design are able to generate complex, long sequences in an autoregressive fashion. However, the standard LSTM unit for videos (ConvLSTM) is not ideally suited for the task of unconditional video generation. Therefore, we propose a simple yet effective LSTM variant called as TransConv LSTM (TC-LSTM) by modulating the conventional ConvLSTM to have a transpose convolutional structure in input-to-state transitions. This enables the network to model both spatial and temporal relationships across layers simultaneously inside the TC-LSTM unit. TC-LSTM unit acts as a building block of our generator. Extensive quantitative and qualitative analysis shows that RV-GAN outperforms state-of-the-art methods by a significant margin on Moving MNIST, MUG, Weizmann and UCF101 datasets. Additionally, owing to the recurrent structure, our method is able to generate high-quality videos, up to 2 times longer (32 frames) than training videos at inference time. Further analysis confirms that the proposed architecture is generic and can be easily adapted to other tasks like class-conditional video synthesis and text-to-video synthesis.

1. Introduction

Video generation is a complex task as it requires to model spatial as well as temporal dynamics simultaneously. Video generation task provides a means for unsupervised feature representation learning from the vast amount of unlabeled data available on the internet. Study [41] shows an improvement in the performance of downstream tasks like action classification, by using the learned weights of the discriminator.

Following the success of Convolutional GANs in image generation literature, many of the recent works in unconditional video synthesis have proposed 3D-CNN GANs [30,41,42]. These methods suffer from the following drawbacks: at inference, 3D CNN models can generate reasonable quality video sequences for a fixed length video on which it has been trained. The number of generated frames can be increased by increasing the time dimension of the input noise, but this leads to deteriorated results (see figure 4). Thus, to faithfully generate longer videos, more convolutional layers are required at the training time which leads to an increase in the number of parameters.

A few other works [37,39] modelled video synthesis as a two-step process. In the first step, latent vectors corresponding to each frame of the video are generated. In the second step, each frame is generated by using a 2D convolutional image generator. Although the two-step process reduces the complexity of the task, it lacks in the following: (1) It struggles to maintain the same appearance throughout the video, (2) spatio-temporal consistency is not modeled properly as temporal relationship is learned only in latent space and there is a lack of information transfer between consecutive frames (see figures 3, 4).

Videos are a sequence of frames, where consecutive frames will have high correlation. To exploit this inherent property of videos, we propose to use recurrent GAN namely, RV-GAN consisting of a recurrent generator with stacked TC-LSTM layers and two CNN based discriminators. Because of this hybrid design, the generator enjoys the sequential modelling capabilities of RNNs whereas discriminators leverage CNN properties to achieve good classification performance. In other words, our model is able to extract both local and global dynamics of the video. The architecture of the generator is inspired by video prediction models solving the task of future frame generation [36,44–47]. However, these networks are not designed to work for unconditional video generation setting where the generator learns a mapping between low-dimensional latent space and high dimensional video space. Thus, we propose an effective modification to Convolutional LSTM (ConvL-

STM) [47] that facilitates the interaction between low dimensional features (from the previous layer) with high dimensional features (of previous time step) via transpose convolution. This enables the vertical flow of low-level information from latent space input to RGB frame output. This modified LSTM block is the core component of RV-GAN and is referred to as TransConv LSTM (TC-LSTM). To summarize, following are the key contributions of our work:

1. We propose a recurrent GAN network, RV-GAN with recurrent generator and Convolutional Image and Video Discriminators. A novel TC-LSTM unit is used as the constituent unit for unconditional video generation.

2. We showcase the generalization capability of the proposed architecture on longer sequence generation (upto 32 frames) by training on shorter sequence. We further demonstrate the application of our model on two conditional video generation tasks, namely: class-conditional and text-to-video synthesis.

3. Extensive experimentation on benchmark datasets, both quantitatively and qualitatively along with the ablation studies demonstrates the superiority of our model over the state-of-the-art methods.

2. Related Work

Recently, the problem of video generation has attracted significant attention by research community. The task is challenging because it requires the network to generate realistic videos. Thus, to perform well, one of the critical requirements is to learn a good quality spatio-temporal representation from the training data. Two of the popular learning strategies used in the literature are GANs [9] and VAEs [25]. The complexity of the task varies based on the input that is used to condition the generation process. The methods that use some form of conditioning as input, for eg. first frame, few frames, etc are a bit easier to model than the one trained merely on latent noise. In this paper, we focus on the latter task, where no conditioning is provided as input i.e. unconditional video generation.

Conditional Video Generation One of the ways to reduce the complexity of modelling the high-dimensional video data is to provide additional information to the network such as class labels, captions, optical flow, few frames etc. This guides the network by revealing the spatial structure, content, and motion of the underlying video to be modelled. Few well-explored conditional video generation tasks are: Image to video generation [7,49], video-to-video translation [5,27] and future frame prediction [3,36,44–46]. The conditioning input can also be from other domains like text, audio etc. Some of these tasks are text-to-video generation [6,17,23], audio-to-video synthesis [12,19]. Although, our network does not require such conditioning, it can be generalized to these scenarios as illustrated in section 5.7.

Unconditional Video Generation refers to the generation of new video samples from training data distribution using latent noise vectors. To reduce the complexity of the task, several works try latent space decomposition into different video attributes, for example foreground, background, motion, content, objects, etc. VGAN [41] proposes a two-stream generator and a video discriminator for synthesizing the moving foreground and static background of the video separately. G3AN and G3AN++ [13, 42] aim to disentangle content and motion and introduce multi-stream convolutional architecture, where various branches are responsible for modelling spatial, spatio-temporal and temporal features. V3GAN [22] decomposes a video into foregound, background and motion using three branch convolutional generator and proposes a feature level masking strategy and shuffling loss to improve the decomposition. InMoDeGAN [43] assumes that the motion in a video can be represented in orthogonal basis vectors. They try to control video motion in latent space using a 3D Conv based generator and temporal pyramid discriminator.

Other direction of research is to utilize an image generator for generating each frame of the video. TGAN [30] proposes a dual generator approach where the temporal generator synthesizes the latent vectors corresponding to each frame of the video and an image generator maps these vectors to frames. MoCoGAN [39] replaces the temporal generator with a GRU for modelling motion in latent space. Similarly, [20, 37] attempt to generate higher resolution videos ranging from 128x128 to 512x512 and longer video generation ranging from 32 to 64 frames. In [34], Skorokhodov et al. builds on top of styleGAN2 model, and proposes a continuous time generator and a single hypernetwork based discriminator. In addition to these, [48] explore the use of transformer architecture for latent space, although high computational cost of transformer models for video data hinders the research in this direction. Limited works are present for higher resolution and longer video generation models due to unavailability of high-end GPU machines. We could not compare our model with these methods because of the high computational requirement for training these on our datasets.

RNN-GAN frameworks have been applied in diverse domains such as text [50], finance [33], sensors [2], music [28], medical [8] etc. However, most of these networks operate on same input and output size. Applying similar architecture in unconditional video generation domain is not straightforward due to the requirement of mapping low-dimensional latent noise vector into the high-dimensional video data. Our propose recurrent framework RV-GAN solves the task with a simplistic change in the existing ConvLSTM framework.

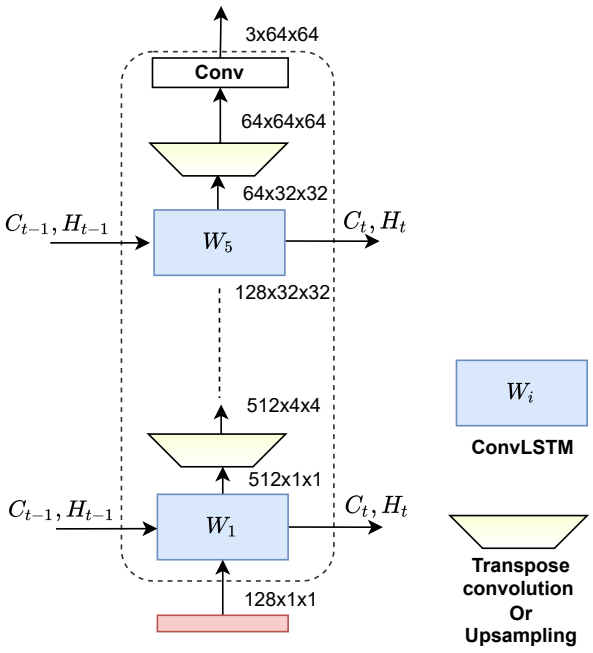

Figure 1. **Baseline:** Recurrent Generator with Transpose Convolution or Upsampling layer integrated with ConvLSTM [47] unit.

3. Preliminaries

3.1. Convolutional LSTM

LSTM [16] is a stable and powerful RNN module that has been proven to be successful for sequence modeling problems [4, 11, 29, 38]. It is able to model long-term dependencies by using cell state C_t that can accumulate information. The information in the cell-state is updated with the help of self-parameterized controlling gates. An LSTM consists of three gates, namely, input, output and forget gates. On arrival of a new input, the previous cell state C_{t-1} is updated depending upon which gate is activated. For instance, the information in the input will be accumulated if the input gate is activated. Similarly, the information in C_{t-1} can be forgotten if forget gate is activated. Finally, the output gates control the fraction of information of the latest cell state C_t that would be propagated to the final (hidden) state H_t. LSTM were designed to work with 1D data.

Convolutional LSTM (ConvLSTM) [47] is a variation of LSTM [16] that takes a 3D tensor as input, where two of the dimensions corresponds to height and width of the input data and third is the channel dimension. Thus, it can model the spatio-temporal features simultaneously by preserving the spatial information which would otherwise be lost in a standard LSTM. ConvLSTM determines the future state of a cell from the inputs and past states of its local neighbours, by using a convolutional structure in both the input-to-state and state-to-state transitions. In ConvLSTM, the input and output size are kept same by applying padding before convolution operation. The key equations of ConvLSTM are as follows:

$$\begin{aligned} g_t &= \tanh\left(W_{xg} \circledast X_t + W_{hg} \circledast H_{t-1} + b_g\right) \\ i_t &= \sigma(W_{xi} \circledast X_t + W_{hi} \circledast H_{t-1} + b_i) \\ f_t &= \sigma(W_{xf} \circledast X_t + W_{hf} \circledast H_{t-1} + b_f) \\ C_t &= f_t \odot C_{t-1} + i_t \odot g_t \\ o_t &= \sigma(W_{xo} \circledast X_t + W_{ho} \circledast H_{t-1} + b_o) \\ H_t &= o_t \odot \tanh\left(C_t\right) \end{aligned} \tag{1}$$

Here, σ represents the Sigmoid operation, $\circledast$ represents the convolution operation and $\odot$ shows Hadamard product. i_t, f_t, o_t, g_t are the input, forget, output and input-modulation gates respectively. $\{X_1, ..., X_t\}$, $\{C_1, ..., C_t\}$, $\{H_1, ..., H_t\}$ corresponds to the inputs, cell states and hidden states.

4. Proposed Method: RV-GAN

In this section, we first discuss two straightforward extensions of ConvLSTM networks for generative setting where the network learns to map an input noise vector to video. We then discuss the proposed RV-GAN architecture in detail.

4.1. Baselines

A stacked LSTM recurrent generator for video synthesis should satisfy following properties: (i) the spatial structure in the hidden states should be preserved. (ii) The size of the hidden states should increase as we move deeper across the layers in the network. In other words, more spatial context should be captured for higher resolution video synthesis. Keeping these in mind, we propose two plausible baselines using transpose convolution as illustrated in Figure 1. To satisfy property (ii), we introduce Transpose convolution or UpSampling block after each ConvLSTM layer.

However, we find that the model with upsampling layer introduces training instability and leads to mode collapse. On the other hand, the model with transpose convolutional layer generates samples where the appearance of the person gets distorted over time. This is because such a design fails to encode the change in spatial features between LSTM layers across time steps. To address above limitations, we propose to integrate the transpose convolution operation inside the LSTM unit. This leads to incorporation of learnable parameters of transpose convolution into the recurrent state transition over time. We call this modified LSTM unit as TransConv LSTM (TC-LSTM).

The overall architecture of our proposed RV-GAN model is shown in Figure 2(a). It consists of a recurrent generator and two convolutional discriminators, namely, image and

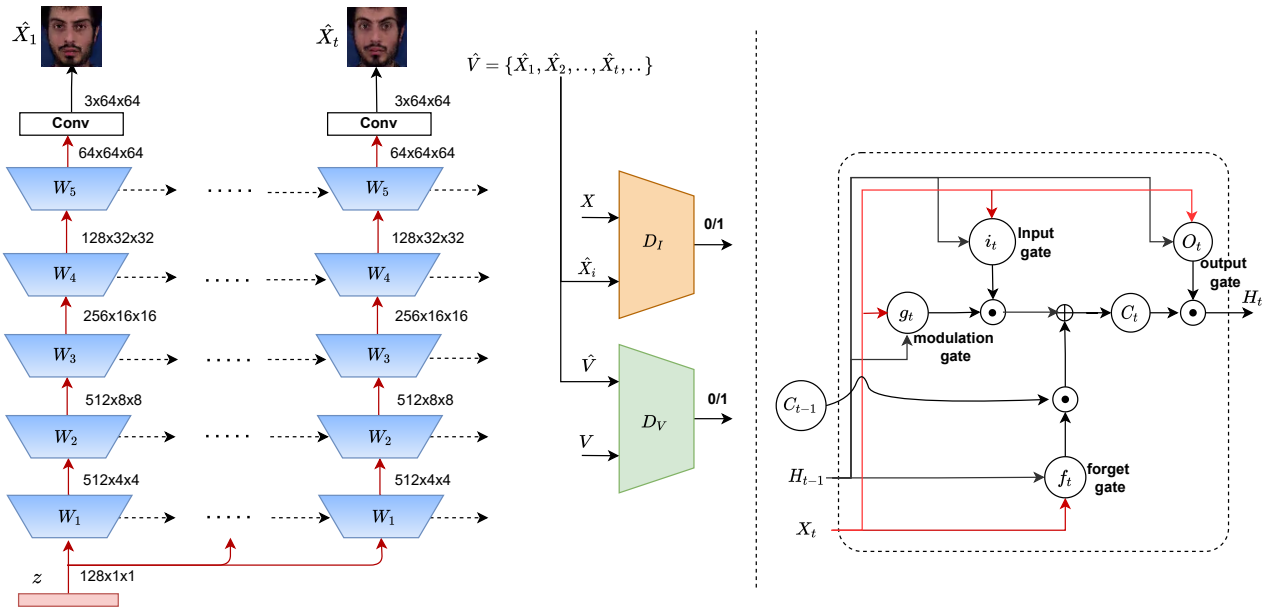

Figure 2. **RV-GAN (left):** The generator consists of a stack of five TC-LSTM layer with a 2D convolution layer at the end. It takes a random noise vector z as input to generate realistic video $\hat{V} = \{\hat{X}_1, \hat{X}_2, ..\hat{X}_T\}$ where X_i is frame at i^{th} time step in a video. Image Discriminator D_I accepts an image sampled from the video as input. Video Discriminator D_V accepts a video as input. **TC-LSTM Unit (right):** Diagrammatic representation of TC-LSTM. $H_{t-1}, C_{t-1}, H_t, C_t$ are the hidden states and cell states at previous and current timestamps. X_t is input to the TC-LSTM unit. Red arrows indicates the transpose convolution operation, black arrows represents convolution operation.

video. Let the training data consist of N video samples where each video is represented as $V = \{X_1, X_2, ..., X_T\}$. Here $X_i, 1 \leq i \leq T$ represents the i^{th} frame of the input video. Video generation problem can then be described as generating the output video $V = \{\hat{X}_1, \hat{X}_2, .., \hat{X}_T\}$ with T frames from an input noise vector z, randomly sampled from Normal distribution. The core of the recurrent generator architecture is the proposed TC-LSTM unit as discussed next.

4.2. TransConv-LSTM (TC-LSTM)

A single unit of the proposed TC-LSTM is illustrated in Figure 2 (right). It takes 3 inputs: X_t, the input noise vector or hidden state from previous TC-LSTM layer; H_{t-1}: the hidden state from previous time step; and C_{t-1}: cell state from previous time step. We use transpose convolution between input-to-state transitions in the input gate i_t, input-modulation gate g_t, forget gate f_t and output gate o_t. Thus, the computation of upsampled feature maps rely on two factors, cell states and hidden state of previous timestamp and input from previous layer. This design enhances the modeling capability of short-term spatio-temporal dynamics of the network. The key equations of TC-LSTM are given in equation 2, where $\circledast_T$ denotes the transpose convolution, $\circledast$ denotes the convolution and $\odot$ denotes the Hadamard product.

Choice of kernel size, stride and padding in transpose convolution operations decide the factor with which the input will be scaled up spatially. Thus, the same 5 layer network can be used for generating higher resolution video as well. Hidden states (H_t) and cell states (C_t) of the TC-LSTM are initialized to zero which corresponds to no past memory.

$$\begin{aligned} g_t &= \tanh\left(W_{xg} \circledast_T X_t + W_{hg} \circledast H_{t-1} + b_g\right) \\ i_t &= \sigma(W_{xi} \circledast_T X_t + W_{hi} \circledast H_{t-1} + b_i) \\ f_t &= \sigma(W_{xf} \circledast_T X_t + W_{hf} \circledast H_{t-1} + b_f) \\ C_t &= f_t \odot C_{t-1} + i_t \odot g_t \\ o_t &= \sigma(W_{xo} \circledast_T X_t + W_{ho} \circledast H_{t-1} + b_o) \\ H_t &= o_t \odot \tanh\left(C_t\right) \end{aligned} \tag{2}$$

4.3. Generator Architecture

Like a standard LSTM, TC-LSTM can also be used as a building block for complex architectures. For the unconditional video generation task, we use the recurrent generator architecture as illustrated in Figure 2. It consists of a stack of five TC-LSTM layers. We use a kernel size of 4 with a stride of 2 for input-to-state transition in TC-LSTM. The last TC-LSTM layer outputs 64 channels. Thus, a 2D convolution layer is used to obtain the final frame with 3 channels (RGB). The input at each time step is the same noise vector randomly sampled from the latent space.

Figure 3. **Comparison with SOTA methods** on Moving MNIST, Weizmann, MUG and UCF101 (left to right) datasets with MoCoGAN [39] (first row), G3AN [42] (second row), V3GAN [22] (third row) and our method (RV-GAN) (fourth row). More video sequences can be found in supplementary material. Sample videos can be found here. Frames are chosen at regular intervals for the purpose of visualization.

4.4. Discriminator Architecture

Similar to MoCoGAN [39], we use two separate discriminators, Image Discriminator D_I and Video Discriminator D_V. D_V has 5 convolutional 3D layers modelled as conv(2+1)D block. It accepts the entire video as input. D_I has 5 convolutional 2D layers and accepts a randomly sampled frame of the video as input. Image discriminator helps to maintain the quality of the individual frames. Video discriminator helps to improve the spatio-temporal consistency of the generated output.

4.5. Loss Functions

We use adversarial loss [9] to train the proposed RV-GAN network. Let the recurrent generator be denoted by G. The generator tries to generate realistic videos so that it can fool the discriminator whereas the discriminator tries to distinguish the generated videos from real ones by classifying them as fake or real. The loss functions for generator and discriminators are defined as follows:

$$\begin{aligned} &\min_{G} \max_{D_I, D_V} L_I(G, D_I) + L_V(G, D_V) \\ &\qquad where, \\ &L_I(G, D_I) = E_{s(v)\sim p_{data}}[\log(D_I(s(v)))] \\ &\qquad + E_{z\sim p_z}[\log(1 - D_I(s(G(z))))] \\ &L_V(G, D_V) = E_{v\sim p_{data}}[\log(D_V(v))] \\ &\qquad + E_{z\sim p_z}[\log(1 - D_V(G(z)))] \end{aligned} \tag{3}$$

RV-GAN tries to optimize the above loss function with respect to both D_I and D_V simultaneously. L_I refers to the loss associated with D_I whereas L_V refers to the loss associated with D_V. z is the random noise vector given as input to G to generate the video $G(z)$. v represents a video from the training data distribution. $s(X)$ with $X \in \{v, G(z)\}$ is a function that randomly samples one of the frames from real and generated video.

5. Experiments and Results

5.1. Datasets and Setup

We trained our model on four datasets as described below.

Moving MNIST is a synthetic dataset of handwritten digits consisting of 10,000 videos. Each video sequence contains 2 digits moving independently across the frame.

Weizmann [10] action dataset consists of 93 videos of 9 people performing 10 different actions, such as running, bending, jumping. The video frames have been flipped for augmentation purposes.

UCF101 [35] dataset contains 13,320 real-world video clips, categorized in 101 action classes. We rescale the frames to 85x64 followed by center-cropping to 64x64 for Weizmann and UCF101 datsets, similar to [30].

MUG [1] dataset contains 908 video sequences of 52 individuals performing facial expressions. We chose only 6 expressions- fear, anger, sadness, disgust, happiness and surprise. Faces are cropped in each frame on the basis of landmarks and then resized to 64x64.

For all our experiments, we randomly chose 16 frames with step sizes 1 and 2. The input noise (z) dimension is set to 128. The output videos contain 16 frames at a resolution of 64x64. A batch size of 16 is used. The learning rate for generator and discriminators is set to 10^{-4}. The networks are trained using Adam optimizer [24] with b_1 = 0.5 and b_2 = 0.999. Source code will be made public.

5.2. Evaluation metrics

We use **Fréchet Inception Distance (FID)** [15] as the evaluation metric to measure the quality of the generated videos. FID metric is the squared Wasserstein distance between two multidimensional Normal distributions ($\mathcal{N}(\mu, \Sigma)$). In the case of video data, the feature embedding of generated and real video samples are calculated us-

Method	MMNIST	WZM	MUG	UCF101	
	FID ↓	FID ↓	FID ↓	FID ↓	IS ↑
VGAN [41]	-	158.1	160.7	115.1	2.94
TGAN [30]	-	99.8	97.1	110.5	2.74
MoCoGAN [39]	44.1	92.2	87.1	104.1	3.06
G3AN [42]	16.1	86.0	67.1	86.7	3.62
V3GAN [22]	59.9	62.6	53.5	**80.2**	**3.88**
Baseline	31.3	63.0	MC	94.6	3.24
RV-GAN (Ours)	**14.8**	**57.3**	**23.3**	82.3	3.76

Table 1. Quantitative Comparison of our method with SOTA methods using FID (lower the better) on Moving-MNIST, Weizmann, MUG and UCF101 datasets. MC represents the mode collapse.

ing pretrained 3D-CNN network [14] and then the distributions are calculated by fitting a multivariate Gaussian on the feature embedding. The FID metric is then computed as : FID $= |\mu_r - \mu_g|^2 + \text{tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2})$ where μ_r and Σ_r correspond to the mean and covariance matrix of the real distribution and similarly μ_g and Σ_g correspond to the mean and covariance of the generated distribution. Lower values of FID metric correspond to better quality of the generated videos.

We also report **Fréchet Video Distance (FVD)** [40] metric to compare our method with the recently proposed self-supervised video GAN [18]. FVD uses similar settings as FID except for the 3D-CNN architecture. FVD uses the I3D [4] model for embedding whereas FID uses the Resnext101 [14] model pretrained on Kinetics dataset. [21].

For UCF101 dataset, we also report **Inception score (IS)**. [32]. IS is the KL divergence between class conditional probability distribution ($p(y|x)$) and marginal ($p(y)$) probability distribution which can be calculated as: $IS(G) = \exp(E_{x\sim G}(KL[p(y|x)||p(y)]))$. High values of IS indicate better diversity and quality of the generated samples.

5.3. Quantitative Evaluation

We compare our method with state-of-the-art methods [22, 25, 30, 39, 41] and the proposed baseline in section 4.1 using FID metric. Quantitative comparison of the four datasets is reported in Table 1. To compute the FID value, we have generated 5000 samples using the trained model. It can be seen that our method RV-GAN outperforms almost all other methods except V3GAN on UCF101. This might be because V3GAN generates background of the frame separately which is same throughout the video, which helps in enhancement of the metric. The high IS values obtained on UCF101 dataset implies that the generated samples are diverse and realistic. Table 1 indicates that our model is able to learn the spatio-temporal correlation well. RV-GAN also outperforms the baseline, suggesting that inclusion of

	Weizmann	MUG	UCF101
	FVD↓	FVD↓	FVD↓
MoCoGAN [39]	194.34	102.2	869.41
G3AN [42]	117.69	89.73	687.67
SVGAN [18]	105.51	67.62	643.55
Ours	**91.49**	**49.2**	**623.90**

Table 2. Comparison of the performance of our method with SOTA methods using FVD metric.

Architecture	WZM	MUG	UCF101
	FID ↓	FID ↓	FID ↓
3 - Layer	63.64	25.70	101.1
4 - Layer	61.08	24.78	88.7
5 - Layer (Ours)	**57.32**	**23.31**	**82.3**

Table 3. Ablation with Number of TC-LSTM Layers in generator.

transpose convolution inside the LSTM allows for improved visual quality and temporal consistency. We note that for Moving MNIST dataset, the FID values of V3GAN is the highest. This is because, for such fine digits moving independently in different directions, the network fails to learn foreground-background decomposition. MoCoGAN performs poorly on Moving MNIST. This suggests that the decomposition of video or latent space need not always result in reduced complexity of the task. Recently, self-supervised video GAN (SVGAN) [18] uses FVD for evaluation, hence we also compare our model with MoCoGAN, G3AN and SVGAN in Table 2 using FVD metric. It can be seen that our method consistently achieves the lowest FVD. This further confirms that RV-GAN is able to learn training data distribution.

5.4. Qualitative Results

We show a few samples of the generated videos on moving MNIST, Weizmann, UCF101 and MUG datasets for qualitative comparison with state-of-the-art methods in Figure 3. The generated samples by MoCoGAN, G3AN, V3GAN and RV-GAN are shown in the first, second, third and fourth rows respectively. We observe that our model is able to maintain temporal coherency for moving MNIST data better than other methods. On Weizmann dataset, RV-GAN is able to generate diverse videos. For instance, out of 5000 generated samples, jumping jack action was rarely found in the case of G3AN whereas our model generates sufficient number of jack action videos. For the MUG dataset, our model is able to generate realistic samples that capture the expressions well and generate good facial features. For UCF dataset, background is well modeled and the results are better than those of other methods. As Weizmann

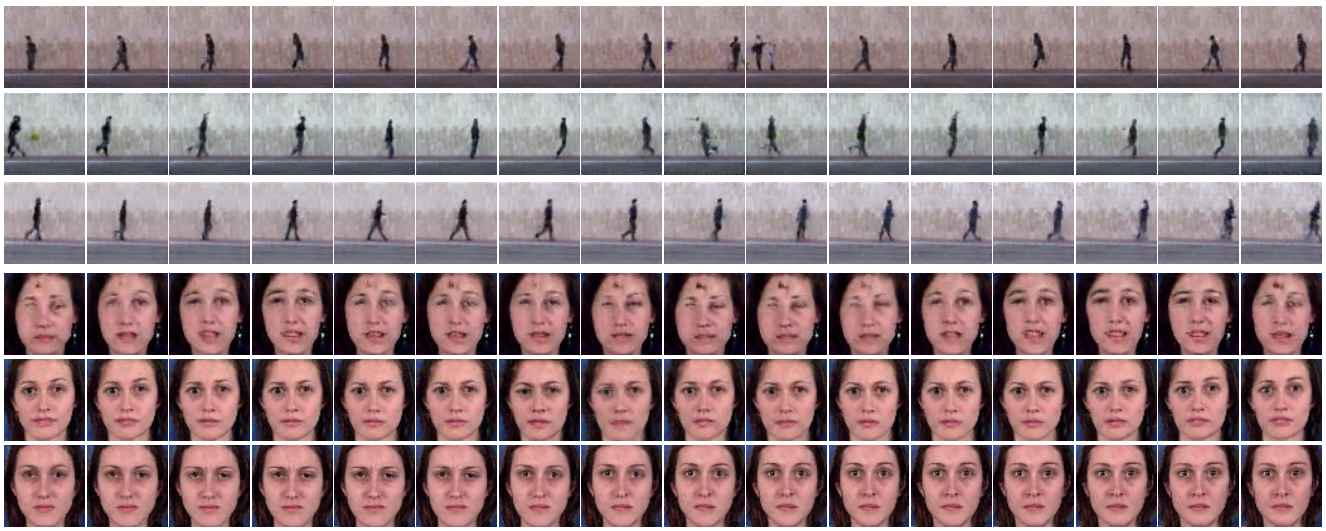

Figure 4. **Comparison with SOTA models for longer Sequence generation on Weizmann and MUG dataset:** Each row represents a video sequence generated by MoCoGAN (top), G3AN (middle) and Ours (bottom) respectively. Each of the model is trained on 16 frames and is used to generate 32 frames at the time of inference. Alternate frames are chosen for the purpose of visualization.

dataset consists of less number of videos, we perform linear interpolation similar to [31] to verify whether the network is memorizing the dataset. Qualitative results for this are available in supplementary material (SM). It can be seen that there is a smooth transition in generated videos with change in input noise. Thus, we can conclude that the network is not memorizing the dataset.

5.5. Ablation Study

Ablation of Architecture: We perform two ablation studies to assess the impact of each component of the proposed RV-GAN. Since there are no recurrent GANs for the task of unconditional video generation (to the best of our knowledge), we build a simple and reasonable baseline as specified in section 4.1. We use the same architecture as that of ours, Figure 2, but replace each TC-LSTM with a combination of ConvLSTM [47] and transpose convolutional layer in the generator as shown in Figure 1. The discriminator remains the same. This baseline emphasizes the need and significance of the proposed TC-LSTM.

Apart from this baseline, we also use the upsampling layer (using bilinear interpolation) after ConvLSTM layer, but this configuration proved to be unstable leading to mode collapse for all datasets. However, we did not come across mode collapse for TC-LSTM which shows that our model is stable. Results in Table 1 suggest that TC-LSTM have contributed to the improvement in the quality of the generated videos.

Study the number of layers: We further test the effect of the number of TC-LSTM layers using three variants as shown in Table 3. The three-layer variant contains three TC-LSTM layers with 512, 256 and 128 channels in hidden states, respectively. The four-layer variant has four TC-LSTM layers with 512, 256, 128 and 64 channels in hidden states, respectively, with a kernel size of 4x4. It can be seen from the Table 3 that our method consistently outperforms the present state-of-the-art methods even with a lesser number of layers. This also proves that our model is stable and maintains the quality of generated videos with a lower number of parameters. As expected, the performance of the network improves when more layers are used.

5.6. Longer Sequence Generation

One of the advantages of using a recurrent generator over 3D-CNN is that we can synthesize a longer sequence during inference time. This is useful when we have limited computational resources for training. Thus, we analyze the performance of our method for the generation of longer video sequences, both quantitatively in table 4 and qualitatively in figure 4. We used the model trained on sequences of 16 frames to generate videos for 24 and 32 time steps. We further compare it with MoCoGAN which uses the recurrent component to generate the motion noise vector and G3AN [42] which is a fully convolutional GAN. In MoCoGAN and G3AN a consistent discontinuity pattern is observed around 15th to 17th frame on Weizmann as well as MUG datasets (see rows 1,3 and 2,4 in Figure 4). However, we did not encounter such a pattern for our model. This shows that using a recurrent network for modelling motion only in latent space is not sufficient. More results for longer videos are available in SM.

Figure 5. **Class Conditional video generation:**. For both datasets, videos are generated by fixing the noise vector z and changing the class labels as shoown on top of each video sequence.

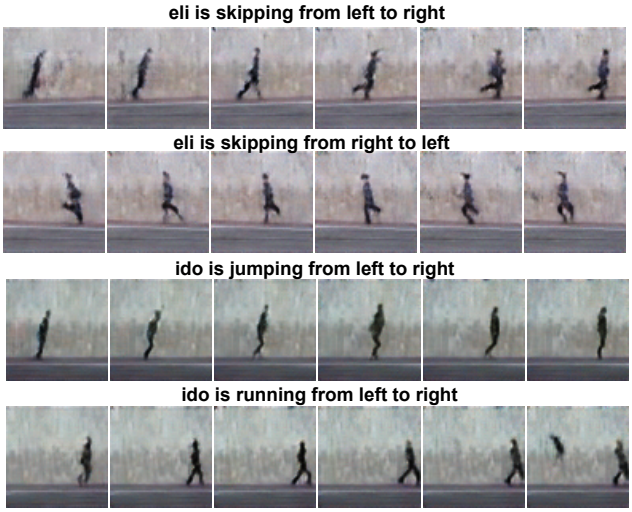

Figure 6. **Text-to-video generation on Weizmann dataset**. The caption used as input is shown on top of the video sequence. We observe that our model is able to learn the action and direction information correctly (first and second row). It also able to learn the appearance of the person corresponding to the input name.

5.7. Applications to Conditional Video Generation

We show that our recurrent model can easily be extended to other conditional video generation tasks. In particular, we trained a slightly modified version of RV-GAN for video generation conditioned on class label (also referred to as class conditional video generation) and text-to-video generation as described below. For both of these tasks, the experiments are performed on Weizmann Action dataset.

Class Conditional Video Generation : The task is to take a class label and random noise vector as input and generate a fake video which corresponds to the given class label. We concatenate one hot vector embedding of action label with input noise along the channel dimension and pass it through the generator. To condition both image and video discriminator, a repetition of one hot vector is concatenated as the fourth channel (after 3 RGB channels) in each frame of the video. The results can be visualized in Figure 5.

Text to Video Generation: Here, the conditioning factor for video generation is a caption (text). Since the text description of a video belongs to an entirely different modality, hence it is crucial to have sufficient number of data mapping from text to video. Text gives marginal information to create a video, which makes it a challenging problem. We chose to use an encoder-decoder model for this task. Pre-trained skip-thought [26] embedding is used to encode the caption and RV-GAN generator is used as a decoder. We choose five action categories of Weizmann action dataset namely: running, walking, side-walking, skipping and jumping because these classes have maximal motion along with an associated direction of movement. Some of the generated examples are shown in Figure 6. We can notice that the results are semantically correct but there is scope of improvement in the quality of the video which is hindered due to unavailability of the large size dataset.

Method	WZM		MUG	
	24 fr	32 fr	24 fr	32 fr
	FID ↓	FID↓	FID↓	FID↓
G3AN [42]	97.89	118.84	39.64	39.62
MoCoGAN [39]	101.05	103.25	39.94	41.26
Ours	**72.53**	**77.07**	**29.80**	**37.81**

Table 4. Quantitative comparison with SOTA methods for longer video generation at inference time. fr represents frame.

6. Conclusion

In this work, we present a novel recurrent framework for video synthesis. We extend ConvLSTM to build our novel TC-LSTM. Ablation analysis confirms that TC-LSTM is a stable LSTM unit for building complex generative architectures for spatio-temporal data. By incorporating a hybrid generator-discriminator architecture with adversarial learning, our framework is able to achieve results superior to the state-of-the-art methods on benchmark datasets. We believe that our work will open avenues for exploring even better recurrent architectures for the unconditional video generation task.

References

[1] Niki Aifanti, Christos Papachristou, and Anastasios Delopoulos. The mug facial expression database. In *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*, pages 1–4. IEEE, 2010. 5

[2] Moustafa Alzantot, Supriyo Chakraborty, and Mani Srivastava. Sensegen: A deep learning architecture for synthetic sensor data generation. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 188–193. IEEE, 2017. 2

[3] Prateep Bhattacharjee and Sukhendu Das. Temporal coherency based criteria for predicting video frames using deep multi-stage generative adversarial networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 4271–4280, 2017. 2

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 3, 6

[5] Yang Chen, Yingwei Pan, Ting Yao, Xinmei Tian, and Tao Mei. Mocycle-gan: Unpaired video-to-video translation. *Proceedings of the 27th ACM International Conference on Multimedia*, 2019. 2

[6] Kangle Deng, Tianyi Fei, Xin Huang, and Yuxin Peng. Irc-gan: Introspective recurrent convolutional gan for text-to-video generation. In *IJCAI*, 2019. 2

[7] Michael Dorkenwald, Timo Milbich, Andreas Blattmann, Robin Rombach, Konstantinos G Derpanis, and Bjorn Ommer. Stochastic image-to-video synthesis using cinns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3742–3753, 2021. 2

[8] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *ArXiv preprint arXiv:1706.02633*, 2017. 2

[9] 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. 2, 5

[10] 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. 5

[11] Alex Graves. Generating sequences with recurrent neural networks. *ArXiv preprint arXiv:1308.0850*, 2013. 3

[12] Yudong Guo, Keyu Chen, Sen Liang, Yongjin Liu, Hujun Bao, and Juyong Zhang. Ad-nerf: Audio driven neural radiance fields for talking head synthesis. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5764–5774, 2021. 2

[13] Sonam Gupta, Arti Keshari, and Sukhendu Das. G3an++ exploring wide gans with complementary feature learning for video generation. In *Proceedings of the Twelfth Indian Conference on Computer Vision, Graphics and Image Processing*, pages 1–9, 2021. 2

[14] 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. 6

[15] 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. 5

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 3

[17] Yaosi Hu, Chong Luo, and Zhenzhong Chen. Make it move: Controllable image-to-video generation with text descriptions. *ArXiv*, abs/2112.02815, 2021. 2

[18] Sangeek Hyun, Jihwan Kim, and Jae-Pil Heo. Self-supervised video gans: Learning for appearance consistency and motion coherency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10826–10835, 2021. 6

[19] Xinya Ji, Hang Zhou, Kaisiyuan Wang, Wayne Wu, Chen Change Loy, Xun Cao, and Feng Xu. Audio-driven emotional video portraits. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14075–14084, 2021. 2

[20] 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. 2

[21] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *ArXiv preprint arXiv:1705.06950*, 2017. 6

[22] Arti Keshari, Sonam Gupta, and Sukhendu Das. V3gan: Decomposing background, foreground and motion for video generation. 2021. 2, 5, 6

[23] Doyeon Kim, Donggyu Joo, and Junmo Kim. Tivgan: Text to image to video generation with step-by-step evolutionary generator. *IEEE Access*, 8:153113–153122, 2020. 2

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, abs/1412.6980, 2015. 5

[25] Diederik P. Kingma and M. Welling. Auto-encoding variational bayes. *ICLR*, abs/1312.6114, 2014. 2, 6

[26] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in Neural Pnformation Processing Systems*, pages 3294–3302, 2015. 8

[27] Kangning Liu, Shuhang Gu, Andrés Romero, and Radu Timofte. Unsupervised multimodal video-to-video translation via self-supervised learning. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1029–1039, 2021. 2

[28] Olof Mogren. C-rnn-gan: Continuous recurrent neural networks with adversarial training. *ArXiv preprint arXiv:1611.09904*, 2016. 2

[29] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *ArXiv preprint arXiv:1412.6604*, 2014. 3

[30] 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. 1, 2, 5, 6

[31] Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan, May 2020. 7

[32] Tim Salimans, I. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, 2016. 6

[33] Luca Simonetto. Generating spiking time series with generative adversarial networks: an application on banking transactions. *MS thesis - Univ. of Amsterdam*, 2018. 2

[34] Ivan Skorokhodov, S. Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. *ArXiv*, abs/2112.14683, 2021. 2

[35] 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. 5

[36] Jiahao Su, Wonmin Byeon, Jean Kossaifi, Furong Huang, Jan Kautz, and Anima Anandkumar. Convolutional tensor-train lstm for spatio-temporal learning. *Advances in Neural Information Processing Systems*, 33:13714–13726, 2020. 1, 2

[37] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N. Metaxas, and S. Tulyakov. A good image generator is what you need for high-resolution video synthesis. *ArXiv*, abs/2104.15069, 2021. 1, 2

[38] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015. 3

[39] 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. 1, 2, 5, 6, 8

[40] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *In Deep Generative Models for Highly Structured Data, ICLR 2019 Workshop,*, 2019. 6

[41] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. *Advances in Neural Information Processing Systems*, 29:613–621, 2016. 1, 2, 6

[42] 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. 1, 2, 5, 6, 7, 8

[43] Yaohui Wang, François Brémond, and Antitza Dantcheva. Inmodegan: Interpretable motion decomposition generative adversarial network for video generation. *ArXiv*, abs/2101.03049, 2021. 2

[44] Yunbo Wang, Zhifeng Gao, Mingsheng Long, Jianmin Wang, and S Yu Philip. Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning. In *International Conference on Machine Learning*, pages 5123–5132, 2018. 1, 2

[45] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *International Conference on Learning Representations*, 2018. 1, 2

[46] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 879–888, 2017. 1, 2

[47] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, pages 802–810, 2015. 1, 2, 3, 7

[48] Wilson Yan, Yunzhi Zhang, P. Abbeel, and A. Srinivas. Videogpt: Video generation using vq-vae and transformers. *ArXiv*, abs/2104.10157, 2021. 2

[49] Jiangning Zhang, Chao Xu, L. Liu, Mengmeng Wang, Xia Wu, Yong Liu, and Yunliang Jiang. Dtvnet: Dynamic time-lapse video generation via single still image. In *ECCV*, 2020. 2

[50] Yizhe Zhang, Zhe Gan, and Lawrence Carin. Generating text via adversarial training. In *NIPS Workshop on Adversarial Training*, volume 21, pages 21–32. academia. edu, 2016. 2