IRC-GAN: Introspective Recurrent Convolutional GAN for Text-to-video Generation

Kangle Deng* , **Tianyi Fei*** , **Xin Huang** and **Yuxin Peng†**
Institute of Computer Science and Technology, Peking University, Beijing, China
pengyuxin@pku.edu.cn

Abstract

Automatically generating videos according to the given text is a highly challenging task, where visual quality and semantic consistency with text are two critical issues. In existing methods, when generating a specific frame, the information in those frames generated before is not fully exploited. And an effective way to measure the semantic consistency between videos and given text remains to be established. To address these issues, we present a novel Introspective Recurrent Convolutional GAN (IRC-GAN) approach. First, we propose a recurrent transconvolutional generator, where LSTM cells are integrated with 2D transconvolutional layers. As 2D transconvolutional layers put more emphasis on the details of each frame than 3D ones, our generator takes both the definition of each video frame and temporal coherence across the whole video into consideration, and thus can generate videos with better visual quality. Second, we propose mutual-information introspection to semantically align the generated video to text. Unlike other methods simply judging whether the video and the text match or not, we further take mutual information to concretely measure the semantic consistency. In this way, our model is able to introspect the semantic distance between the generated video and the corresponding text, and try to minimize it to boost the semantic consistency. We conduct experiments on 3 datasets and compare with state-of-the-art methods. Experimental results demonstrate the effectiveness of our IRC-GAN to generate plausible videos from given text.

1 Introduction

In computer vision, automatic visual content generation has experienced a remarkable evolution due to Generative Adversarial Networks (GANs) [Goodfellow *et al.*, 2014]. Many improvements are made to reach better results including CA-GAN [Ni *et al.*, 2018] , MEGAN [Park *et al.*, 2018] and so on [Zhang and Peng, 2018b]. GANs are capable of more complex tasks like muti-domain synthesis [Mao and Li, 2018; Hao *et al.*, 2018] and multi-view generation [Tian *et al.*, 2018; Song *et al.*, 2018].

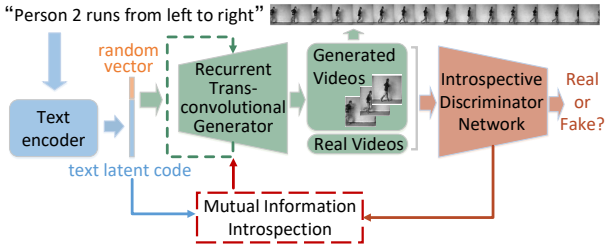


Figure 1: Overview of the proposed IRC-GAN approach.

In this paper, we focus on generating videos from text, namely text-to-video generation. There are two critical issues in such task. First, the generated frames need to be both realistic and temporally coherent. Second, the video content needs to be in accordance with text. In brief, text-to-video generation has two key components: **visual quality and semantic consistency**.

A straightforward idea to handle such issues is to use conditional generative models like c-GAN [Goodfellow *et al.*, 2014] and c-VAE [Doersch, 2016] as most current methods do, which are mainly adapted from image synthesis like [Zhang *et al.*, 2017; Zhang and Peng, 2018a]. When extending image generation to videos, there is an inevitable problem that videos have one more dimension than images. In order to tackle this, [Pan *et al.*, 2017] replaces the 2D convolutional layers [Zeiler *et al.*, 2010] with 3D layers, while [Mittal *et al.*, 2016] chooses to simply use 2D layers to generate the video frame by frame. However, 3D layers may have poorer frame quality than 2D layers [Saito *et al.*, 2017] while 2D layers fail to take temporal dependency into account. Due to these reasons, generated videos may suffer from either low frame quality or poor temporal coherence. On the other hand, simply treating text as conditions cannot concretely measure the semantic consistency between videos and text and thus may struggle to precisely coordinate videos with text.

In order to overcome the difficulties of video generation from text, we present a novel Introspective Recurrent Convolutional GAN as shown in Figure 1, which mainly has follow-

*These authors contributed equally.
†Corresponding author.

ing contributions:

- **Recurrent Transconvolutional Generator (RTG)**: Frame quality and temporal coherence are two aspects of visual quality of the generated videos. To improve these, our generator integrates LSTM cells with 2D transconvolutional networks. Such structure endows the generator with memory of history information so that each frame is generated on the basis of previous frames, which may lead to better coherence. Besides, as 2D transconvolutional layers put more emphasis on the details of each frame than 3D ones, the definition of each frame is boosted. In this way, our generator can synthesize videos with better quality.

- **Mutual-information Introspection (MI)**: Instead of simply judging matched or not, mutual information is introduced to measure semantic similarity quantitatively. In this way, our model introspects how far the generated video is semantically from the given text during a two-stage training process. In the first stage, the text encoder is trained with a seq2seq auto-encoder and an introspective network extracting mutual information between the text and the corresponding video. In this way, semantic distances among different text are established. In the second stage, our model introspects the semantic distance between the generated videos and the corresponding text and try to minimize it to boost the semantic consistency. By doing so, our model can generate videos precisely matched with the given text.

To verify the effectiveness of our IRC-GAN method, we collect several datasets and modify them into 3 datasets: Single Moving Mnist-4, Double Moving Mnist-4 and KTH-4, comparing with 4 state-of-the-art approaches.

2 Introspective Recurrent Convolutional GAN

Our IRC-GAN attempts to synthesize a temporal coherent and plausible frame sequence semantically aligned with the given text. As shown in figure 2, there are three components: the text encoder network, the recurrent transconvolutional generator network and the introspective discriminator network.

2.1 Text Encoder Network

The text encoder is designed to encode the given text into text latent codes z_{text} for video generation. First, each word is represented as a one-hot vector. So a sentence of length l can be denoted as $\{w_1, w_2, \ldots, w_l\}$. w_t is the *t*-th word's one-hot vector. The sentence is then fed into a bidirectional LSTM network [Schuster and Paliwal, 1997] to contextually embed each word into h_t. Finally, we input the contextually embedded word sequence $\{h_1, h_2, \ldots, h_l\}$ into a LSTM-based encoder and treat the final LSTM output as the text latent code $z_{text} \in \mathbb{R}^{d_{text}}$.

2.2 Recurrent Transconvolutional Generator Network

Overview of Structure

We prefer 2D-transconvolution-based networks to 3D networks for two reasons: 1) Flexibility. 2D networks can handle any arbitrary length of frame sequences. 2) Frame quality. 3D networks construct the whole video in one go, whose kernels are distracted, while 2D networks do the job frame by frame, whose kernels focus on the details of each frame. So the synthesized frames by 2D networks tend to be of higher definition.

But such method can't deal with temporal coherence, because each frame is generated independently. To break such independence, we introduce LSTM units into the network so that each frame can be created on the basis of previous frames. Furthermore, the text latent code z_{text} is additionally fed into the LSTM units at each time step to remind the generator of what the given text is.

Details of Network

First, an LSTM-based feature generator $G_f(z) : \mathbb{R}^{d_{text}+d_{norm}} \rightarrow \mathbb{R}^{d_l \times d_f}$ is designed to transform the latent code $z \in \mathbb{R}^{d_{text}+d_{norm}}$ into a frame-wise feature sequence $[y_1, y_2, \cdots, y_{d_l}]$. The latent code z is the concatenation of the text latent code z_{text} and a normal random vector z_{norm}. And $y_i \in \mathbb{R}^{d_f}$ denotes the frame-wise feature which will be utilized to generate the *i*-th frame later.

Then each y_i will go through a series of 2D transconvolutional layers, denoted as TransConv2D$_k$, $k = 1, 2, \cdots, 5$, and will finally turn into a video frame with the size of $d_c \times d_h \times d_w$. As is mentioned above, in order to take temporal dependency into account, we integrate LSTM cells with each TransConv2D$_i$ to endow the generator with the memory of history information. In particular, the output of TransConv2D$_i$, denoted as $V_i \in \mathbb{R}^{d_{c_i} \times d_{h_i} \times d_{w_i}}$, is first reshaped as $\hat{V}_i \in \mathbb{R}^{d_{c_i} \times (d_{h_i} \times d_{w_i})}$ and we send $\hat{V}_i$ to a LSTM-based memory unit $M_i : \mathbb{R}^{(d_{h_i} \times d_{w_i})} \rightarrow \mathbb{R}^{(d_{h_i} \times d_{w_i})}$ channel by channel. We collect all the channels that go through M_i and put together to form $L_i \in \mathbb{R}^{d_{c_i} \times d_{h_i} \times d_{w_i}}$. Then L_i, concatenated with the original V_i in the channel dim, is treated as the input of the next 2D transconvolution layer, TransConv2D$_{i+1}$. The states of these memory units are passed across frames and thus each frame can be constructed on the basis of history information which can contribute to the temporal coherence.

Besides, we additionally feed the text latent code z_{text} into the memory units at each time step to further guide the video generation so the mapping of the LSTM-based memory unit is actually $\mathbb{R}^{(d_{h_i} \times d_{w_i})+d_{text}} \rightarrow \mathbb{R}^{(d_{h_i} \times d_{w_i})}$. At the end, we put the generated frames together and join them into a whole video with the size of $d_l \times d_c \times d_h \times d_w$.

2.3 Introspective Discriminator Network

Basic Discriminators

Inspired by [Pan *et al.*, 2017], the discriminator network D is designed to distinguish real videos from synthetic ones from three perspectives: (1) the whole video, (2) each video frame, (3) the motion across adjacent frames. To implement these three points, two types of basic discriminators are required:

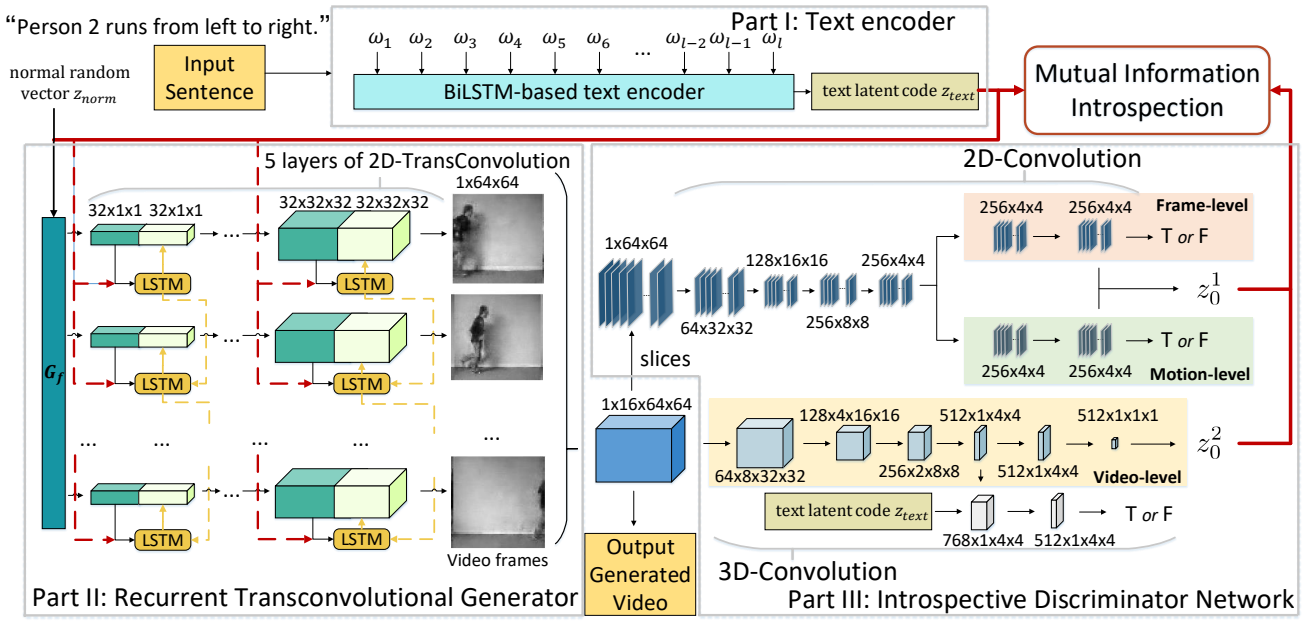


Figure 2: Our IRC-GAN framework consists of three parts: the text encoder network, the recurrent transconvolutional generator network and the introspective discriminator network.

- 3D discriminator $D_1(v) : \mathbb{R}^{d_l \times d_c \times d_h \times d_w} \rightarrow [0, 1]$. D_1 first extracts video-level features m_v from the input video $v \in \mathbb{R}^{d_l \times d_c \times d_h \times d_w}$ via 3D convolutional layers and then feeds m_v into a fully-connected layer with softmax to discriminate whether the input video is real or fake from the perspective of the whole video.
- 2D discriminator $D_2(v) : \mathbb{R}^{d_l \times d_c \times d_h \times d_w} \rightarrow [0, 1]$. D_2 also gets a video as its input but processes it in a frame-wise way. In order to obtain information about the temporal coherence, we get the output after four 2D convolutional layers and subtract the output of the previous frame f_{i-1} from that of the current one f_i. The details are the same as [Pan *et al.*, 2017].

Mutual-information Introspection

The discriminator network described above is only able to help synthesize videos that look like real ones. In order to semantically align the generated videos to the text, we propose mutual-information introspection, which is inspired by InfoGAN [Chen *et al.*, 2016]. We similarly argue that there should be high mutual information between the videos and the corresponding text. In information theory, mutual information between X and Y measures the "amount of information" learned from knowledge of random variable Y about the other random variable X. The mutual information can be expressed as the difference of two entropy terms:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \quad (1)$$

This definition has an intuitive interpretation: $I(X;Y)$ is the reduction of uncertainty in X when Y is observed. This interpretation makes it easy to formulate a cost in the text-to-video problem: give any $z' \sim P_{z_{text}}(z)$, we want $P_G(G(z_{text}, z_{norm})|z_{text} = z')$ to have a small entropy. In other words, the uncertainty of the generated video is minimized when the corresponding text is given. In formalization, mutual-information introspection can be expressed as the following optimization problem:

$$\max_G I(z_{text}; G(z_{text}, z_{norm})) \quad (2)$$

Similarly with [Chen *et al.*, 2016], the mutual information term $I(z_{text}; G(z_{text}, z_{norm}))$ is hard to maximize directly and we first find its lower bound:

$$\begin{aligned}
&I(z_{text}; G(z_{text}, z_{norm})) \\
&= H(z_{text}) - H(z_{text}|G(z_{text}, z_{norm})) \\
&= \mathbb{E}_{x \sim G(z_{text}, z_{norm})}[\mathbb{E}_{z' \sim P(z_{text}|x)} P(z'|x)] + H(z_{text}) \\
&= \mathbb{E}_{x \sim G(z_{text}, z_{norm})}[D_{KL}(P(\cdot|x)||Q(\cdot|x)) \\
&\quad + \mathbb{E}_{z' \sim P(z_{text}|x)} Q(z'|x)] + H(z_{text}) \\
&\geq \mathbb{E}_{x \sim G(z_{text}, z_{norm})}[\mathbb{E}_{z' \sim P(z_{text}|x)} Q(z'|x)] + H(z_{text}) \\
&= \mathbb{E}_{z' \sim P(z_{text}), x \sim G(z', z_{norm})} Q(z'|x) + H(z_{text})
\end{aligned} \quad (3)$$

The above inequation makes sense because the KL-divergence is always non-negative. It is obvious that this lower bound becomes tight as the auxiliary ditribution Q approaches the posterior distribution $P(\cdot|v)$ where v obeys the distribution of videos , making the KL-divergence between Q and P decreases to zero.

In implementation, we parameterize the auxiliary Q as a neural network named as introspective network and perform supervised learning with video-text pairs to make it converge to the true posterior distribution $P(\cdot|v)$. Assume we have trained Q to an approximation of P and then, as a result, since the entropy of z_{text} has nothing to do with the video gener-

ation, mutual-information introspection becomes the maximization of $\mathbb{E}_{z' \sim P(z_{text}), x \sim G(z', z_{norm})} Q(z'|x)$.

For simplicity, we assume the posterior distribution as a parameterized normal distribution. Then the problem becomes:

$$\begin{aligned} &\max_G \mathbb{E}_{z' \sim P(z_{text}), x \sim G(z', z_{norm})} Q(z'|x) \\ \Leftrightarrow &\max_G \mathbb{E}_{z' \sim P(z_{text}), x \sim G(z', z_{norm})} Q(\hat{z} = z'|v = x) \\ \Leftrightarrow &\max_G \mathbb{E}_{z' \sim P(z_{text}), x \sim G(z', z_{norm})} \log Q(\hat{z} = z'|v = x) \\ \Leftrightarrow &\min_G \mathbb{E}_{z' \sim P(z_{text}), x \sim G(z', z_{norm})} ||z' - z_0||^2 \end{aligned} \tag{4}$$

where z_0 is the mean of the normal posterior distribution $P(\cdot|v = x)$ which depends on the video x.

From the above mathematical derivation, an intuitive interpretation for mutual-information introspection is to introspect the generation by reconstructing what it is made from, namely the process of video-to-text, and use the reconstruction error as the cost function.

In order to reduce the computation cost, the Q introspective network can share all the convolutional layers with the discriminators. So we just modify the two discriminators described above by adding a fully-connected layer respectively to output the reconstructed text latent code.

We find by experiments that preserving the conditional structure of the 3D discriminator like [Pan *et al.*, 2017] can speed up the convergence of the model. Therefore, the integrated discriminator network is composed of the following two subnetworks:

- 3D-convolution-based subnet $D_1(v, z_{text}) : \mathbb{R}^{d_l \times d_c \times d_h \times d_w} \times \mathbb{R}^{d_{text}} \rightarrow [0, 1] \times \mathbb{R}^{d_{text}}$. From the perspective of the whole video, D_1 distinguishes whether the input video looks like a real one and matches the given text, as well as introspectively outputs the reconstructed text latent code z_0^1. Of course, the input z_{text} will not participate in the reconstruction of the latent code.
- 2D-convolution-based subnet $D_2(v) : \mathbb{R}^{d_l \times d_c \times d_h \times d_w} \times \mathbb{R}^{d_{text}} \rightarrow [0, 1] \times \mathbb{R}^{d_{text}}$. From the perspective of frame quality and motion, D_2 just distinguishes whether the input video is real or fake and also introspectively outputs the reconstructed text latent code z_0^2.

In the following paper, we still use $D_1(v, z_{text})$ and $D_2(v)$ to represent the true-or-false answer the discriminator network outputs, and we denote the introspectively reconstructed text latent codes from video v as $Q_1(v)$ and $Q_2(v)$ for convenience.

2.4 Optimization

According to the discussion above, the overall optimization problem of the discriminator network is as follows:

$$\min_{D_1, D_2, Q_1, Q_2} \mathcal{L}_{D_1} + \mathcal{L}_{D_2} + \mathcal{L}_{Q_1} + \mathcal{L}_{Q_2} \tag{5}$$

where $\mathcal{L}_{D_1}, \mathcal{L}_{D_2}, \mathcal{L}_{Q_1}, \mathcal{L}_{Q_2}$ are defined as:

$$\begin{aligned} \mathcal{L}_{D_1} = &- \log D_1(v_{real}, z_{matched}) \\ &- \log(1 - D_1(v_{syn}, z_{matched})) \\ &- \log(1 - D_1(v_{real}, z_{unmatched})) \end{aligned} \tag{6}$$

$$\mathcal{L}_{D_2} = - \log D_2(v_{real}) - \log(1 - D_2(v_{syn})) \tag{7}$$

$$\mathcal{L}_{Q_1} = \lambda_{info} \cdot ||Q_1(v_{real}) - z_{matched}||^2 \tag{8}$$

$$\mathcal{L}_{Q_2} = \lambda_{info} \cdot ||Q_2(v_{real}) - z_{matched}||^2 \tag{9}$$

v_{real} and v_{syn} denote real videos and synthetic videos respectively while $z_{matched}$ and $z_{unmatched}$ denote the text latent code that match and does not match the video.

Equations (6) and (7) are cost functions of typical GANs. Equations (8) and (9) are meant to learn the true posterior distribution $P(\cdot|v)$ by means of supervised learning, which is part of our mutual-information introspection.

On the other hand, the overall optimization problem of the generator network is as follows:

$$\min_G \mathcal{L}_{G_1} + \mathcal{L}_{G_2} + \mathcal{L}_{info} \tag{10}$$

where $\mathcal{L}_{G_1}, \mathcal{L}_{G_2}, \mathcal{L}_{info}$ are defined as:

$$\mathcal{L}_{G_1} = - \log D_1(G(z_{text}, z_{norm}), z_{text}) \tag{11}$$

$$\mathcal{L}_{G_2} = - \log D_2(G(z_{text}, z_{norm})) \tag{12}$$

$$\begin{aligned} \mathcal{L}_{info} = \lambda_{info} \cdot (&||Q_1(G(z_{text}, z_{norm})) - z_{text}||^2 \\ &+ ||Q_2(G(z_{text}, z_{norm})) - z_{text}||^2) \end{aligned} \tag{13}$$

Equations (11) and (12) are also cost functions of typical GANs. Equation (13) is the regularization that makes the generator synthesize videos from which the reconstructed text latent code can be as close to the ground-truth as possible, which means aligning the generated videos to the given text.

As discussed above, we can see that both the recurrent transconvolutional generator and the mutual-information introspection rely on an adequate distribution of the text latent code. So it is natural to perform a two-stage training process: first pre-training the text encoder and then training the whole model with the encoder fixed.

In the first stage, the text encoder is trained with a seq2seq auto-encoder [Dai and Le, 2015] and a "Q-style" introspective network which recovers the text latent code from videos. In order to better perform mutual-information introspection later, the cost function includes not only the term of the auto-encoder reconstruction error, but also the mean square distance between the encoded z_{text} and the reconstructed text latent code from the corresponding video. In the second stage, only the text encoder in the previous training process is preserved. And we train the generator and discriminator network with the cost functions of (5) and (10).

3 Experiments and Results

3.1 Datasets

We have adopted 3 datasets of progressively increasing complexity: Single Moving Mnist-4, Double Moving Mnist-4 and KTH-4. They are constructed according to [Mittal *et al.*, 2016]. Each video in these datasets has 16 frames and each frame has the size of 64×64.

- **Single Moving Mnist-4**: [Mittal *et al.*, 2016] first constructed Single Moving Mnist dataset for the text-to-video task. As there are merely two motions: up-down and left-right in the original dataset, we find it too simple to distinguish the effectiveness of the current methods. So we slightly improve the dataset by introducing **four moving directions** : move left then right, move right then left, move up then down and move down then up while the original dataset doesn't tell apart "up then down" and "down then up". Each video is accompanied with a single sentence describing the digit and its moving direction.
- **Double Moving Mnist-4**: [Mittal *et al.*, 2016] also constructed a more complicated version of Moving Mnist which contains two bouncing handwritten digits. Similarly, we extend the dataset by introducing four moving directions for each digit.
- **KTH-4**: Based on the original KTH dataset by [Laptev *et al.*, 2004], [Mittal *et al.*, 2016] first select some video clips and accompany each video clip with a descriptive caption. Since [Mittal *et al.*, 2016] didn't make open their dataset or source codes to construct it, we construct our own KTH-4 according to the method mentioned in [Mittal *et al.*, 2016]. The original KTH dataset contains 600 videos, classified into 6 actions and 25 people. We pick out 4 of the actions: walking, running, boxing, waving hands, and then extend them into 6 motions: walking from left to right, walking from right to left, running from left to right, running from right to left, waving hands and boxing.

3.2 Compared Methods

We compare our IRC-GAN with the following state-of-the-art methods: Sync-DRAW [Mittal *et al.*, 2016], VGAN-c [Vondrick *et al.*, 2016], TGANs-c [Pan *et al.*, 2017] and MoCoGAN-c [Tulyakov *et al.*, 2018]. Like [Pan *et al.*, 2017], we also modify VGAN and MoCoGAN as VGAN-c and MoCoGAN-c to adapt to the text-to-video task.

3.3 Evaluation Metrics

In text-to-video generation, we need to evaluate both the visual quality and the semantic match. In addition to visually examining the results, we adopt two quantified evaluation metrics to evaluate the effectiveness.

Generative Adversarial Metric (GAM)

Generative Adversarial Metric [Im *et al.*, 2016] can directly compare two generative adversarial models by having them engage in a "battle" against each other. Given two generative models $M_1 = (\tilde{G}_1, \tilde{D}_1)$ and $M_2 = (\tilde{G}_2, \tilde{D}_2)$, two kinds of ratios between the discriminators of the two models are calculated:

$$r_{test} = \frac{\epsilon(\tilde{D}_1(x_{test}))}{\epsilon(\tilde{D}_2(x_{test}))}, r_{sample} = \frac{\epsilon(\tilde{D}_1(\tilde{G}_2(z)))}{\epsilon(\tilde{D}_2(\tilde{G}_1(z)))} \quad (14)$$

where $\epsilon(\cdot)$ denotes the average classification error rate and x_{test} is the testing set. If r_{test} is close to 1, which means the two models have almost the same ability to recognize the real videos, the relationship between r_{sample} and 1 can reveal which model can fool the other model more easily. **For example, $r_{test} \approx 1$ and $r_{sample} < 1$ mean that G_1 is a better generative model than G_2.**

Since this method is restricted for GANs, Sync-DRAW is excluded from this comparison. As for our IRC-GAN, our discriminator network additionally takes the mutual information between the generated videos and the text into consideration. So we add a term about the mutual information when calculating the classification error rate of our discriminator network. In order to be fair, when comparing our IRC-GAN with other methods, we adjust the weights of the mutual information term to make sure r_{test} is close to 1 so that we can compare the two models by checking r_{sample}.

Human Evaluation

Additionally, we conduct a user study to evaluate both of the visual quality and semantic consistency. To compare the visual quality of our method and others, the testee will be shown 2 gifs, one randomly chosen from the generated videos of our method and the other from one of the compared methods. The testee is then required to point out the gif he thinks to have better quality. On the other hand, to compare the semantic match, we show the testee two gifs generated from the same text but from different methods and we require the testee to report which gif is more in line with the text. A total number of 20 evaluators are invited as testees. Each testee is given 30 inquiries on visual quality and 30 inquiries on semantic match. The testee can prefer either of the two given gifs or he can claim a draw, which means choosing the two at the same time. The percentage user preference shown in Tables 2 and 3 is the proportion of the chosen times of two methods.

3.4 Qualitative Analysis

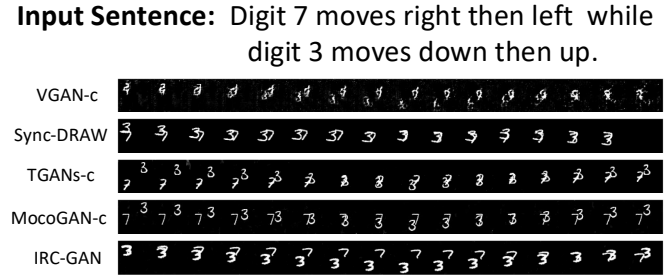
Figure 3 shows some samples of our results on these datasets. VGAN-c doesn't seem to reach convergence and performs poorly. By traditional 2D transconvolutional layers, Sync-DRAW can generate relatively clear frames but its temporal coherence remains to be improved: The shape of the digit varies across frames and its motion is not coherent. Although our IRC-GAN performs roughly the same as TGANs-c and MoCoGAN-c on the simple Moving Mnist dataset, our IRC-GAN generates better results on the real-world KTH-4 dataset. The results demonstrate that our IRC-GAN can generate videos of both fine definition and coherence.

In Table 1, the GAM r_{sample} scores are all less than 1, which means our IRC-GAN can generate videos that can fool the discriminators of other GAN-based methods. In other words, compared with the videos generated by other methods, those generated by ours are more consistent with the text and more similar to the real ones.

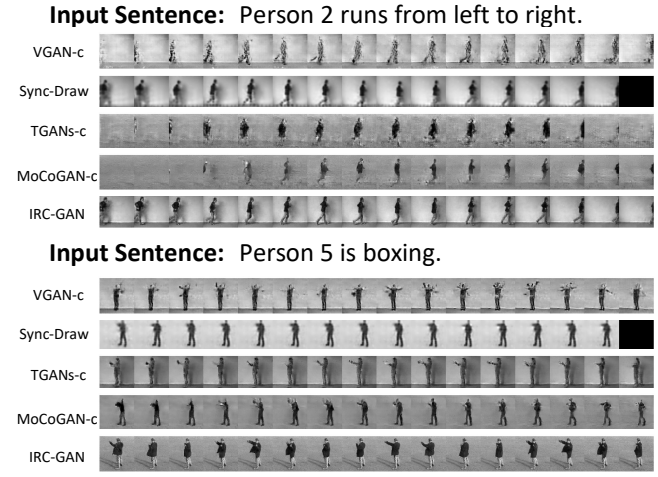
Battler	Single-4	Double-4	KTH-4
ours vs VGAN-c	0.532	0.220	0.391
ours vs TGANs-c	0.673	0.687	0.667
ours vs MoCoGAN-c	0.372	0.372	0.505

Table 1: GAM metric: the r_{sample} score with r_{test} balanced to 1

(a) Results on Single Moving Mnist-4



(b) Results on Double Moving Mnist-4



(c) Results on KTH-4

Figure 3: Examples of generated videos

User preference, %	Single-4	Double-4	KTH-4
ours / VGAN-c	**97.2**/2.8	**99.9**/0.1	**85.7**/14.3
ours / Sync-DRAW	**69.8**/30.2	**76.2**/23.8	**82.6**/17.4
ours / TGANs-c	**59.3**/40.7	**67.6**/32.4	**54.5**/45.5
ours / MoCoGAN-c	**52.0**/48.0	**57.1**/42.9	**58.3**/41.7

Table 2: User preference score on video generation quality

In Tables 2 and 3, we can see that more testees prefer the generated videos of our method both on visual quality and semantic match. The advantage of our model on visual quality suggests that our RTG is capable of synthesizing more coherent and more vivid videos by memorizing the previous generated frames. And exceeding in semantic match indicates that

User preference, %	Single-4	Double-4	KTH-4
ours / VGAN-c	**94.1**/5.9	**95.5**/4.5	**95.0**/5.0
ours / Sync-DRAW	**73.5**/26.5	**83.3**/16.7	**71.1**/28.9
ours / TGANs-c	**64.0**/36.0	**55.6**/44.4	**51.9**/48.1
ours / MoCoGAN-c	**58.6**/41.4	**63.2**/36.8	**64.4**/35.6

Table 3: User preference score on the semantic consistency between the generated videos and the text

Battler	Single-4	Double-4	KTH-4
ours vs RCGAN-c	0.500	0.768	0.778
ours vs TGANs-info	0.895	0.835	0.956

Table 4: GAM metrics of our model against the baselines

measuring the semantic distance between videos and text by MI can effectively improve the video-text consistency.

3.5 Ablation Analysis

Two ablation experiments are conducted to further show the effectiveness of the two components in our approach, namely MI and RTG. 1) RCGAN-c: Our approach without MI. 2) TGANs-info: RTG is replaced with the commonly-used generator of TGANs.

As shown in Table 4, these two baseline models are compared with our IRC-GAN via GAM metrics. The r_{sample} scores are less than 1 on all of the 3 datasets, which means the videos generated by our IRC-GAN can fool these two baseline models. In other words, our model is better.

4 Conclusion

In this paper, we have proposed an Introspective Recurrent Convolutional GAN (IRC-GAN) to generate videos from text. Such task needs to consider both visual quality and semantic consistency. To improve visual quality, we propose a recurrent transconvolutional generator where LSTM cells are integrated with 2D transconvolutional layers. Such generator boosts both the definition of each video frame and temporal coherence across the whole video. On the other hand, semantic consistency is ensured by mutual-information introspection. In this way, the semantic distances between videos and text can be learnt, which helps to synthesize corresponding videos. Experiments on three datasets compared with several state-of-art methods verify the effectiveness of our method. In the future work, we will introduce cross-media techniques to further establish relations between text and video.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61771025.

References

[Chen *et al.*, 2016] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

[Dai and Le, 2015] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087, 2015.

[Doersch, 2016] Carl Doersch. Tutorial on variational autoencoders. 2016.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[Hao *et al.*, 2018] Guang-Yuan Hao, Hong-Xing Yu, and Wei-Shi Zheng. Mixgan: learning concepts from different domains for mixture generation. *IJCAI*, 2018.

[Im *et al.*, 2016] Daniel Jiwoong Im, Chris Dongjoo Kim, Hui Jiang, and Roland Memisevic. Generating images with recurrent adversarial networks. *arXiv preprint arXiv:1602.05110*, 2016.

[Laptev *et al.*, 2004] Ivan Laptev, Barbara Caputo, et al. Recognizing human actions: a local svm approach. In *null*, pages 32–36. IEEE, 2004.

[Mao and Li, 2018] Xudong Mao and Qing Li. Unpaired multi-domain image generation via regularized conditional gans. *IJCAI*, 2018.

[Mittal *et al.*, 2016] Gaurav Mittal, Tanya Marwah, and Vineeth N Balasubramanian. Sync-draw: Automatic video generation using deep recurrent attentive architectures. 2016.

[Ni *et al.*, 2018] Yao Ni, Dandan Song, Xi Zhang, Hao Wu, and Lejian Liao. Cagan: Consistent adversarial training enhanced gans. In *IJCAI*, pages 2588–2594, 2018.

[Pan *et al.*, 2017] Yingwei Pan, Zhaofan Qiu, Ting Yao, Houqiang Li, and Tao Mei. To create what you tell: Generating videos from captions. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1789–1798. ACM, 2017.

[Park *et al.*, 2018] David Keetae Park, Seungjoo Yoo, Hyojin Bahng, Jaegul Choo, and Noseong Park. Megan: mixture of experts of generative adversarial networks for multimodal image generation. *IJCAI*, 2018.

[Saito *et al.*, 2017] 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.

[Schuster and Paliwal, 1997] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[Song *et al.*, 2018] Jingkuan Song, Jingqiu Zhang, Lianli Gao, Xianglong Liu, and Heng Tao Shen. Dual conditional gans for face aging and rejuvenation. In *IJCAI*, pages 899–905, 2018.

[Tian *et al.*, 2018] Yu Tian, Xi Peng, Long Zhao, Shaoting Zhang, and Dimitris N Metaxas. Cr-gan: learning complete representations for multi-view generation. *IJCAI*, 2018.

[Tulyakov *et al.*, 2018] 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.

[Vondrick *et al.*, 2016] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.

[Zeiler *et al.*, 2010] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. 2010.

[Zhang and Peng, 2018a] Chenrui Zhang and Yuxin Peng. Stacking vae and gan for context-aware text-to-image generation. In *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, pages 1–5. IEEE, 2018.

[Zhang and Peng, 2018b] Chenrui Zhang and Yuxin Peng. Visual data synthesis via gan for zero-shot video classification. *IJCAI*, 2018.

[Zhang *et al.*, 2017] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5907–5915, 2017.