To Create What You Tell: Generating Videos from Captions*

Yingwei Pan, Zhaofan Qiu, Ting Yao, Houqiang Li and Tao Mei
University of Science and Technology of China, Hefei, China
Microsoft Research, Beijing, China
{panyw.ustc,zhaofanqiu}@gmail.com;{tiyao,tmei}@microsoft.com;lihq@ustc.edu.cn

ABSTRACT

We are creating multimedia contents everyday and everywhere. While automatic content generation has played a fundamental challenge to multimedia community for decades, recent advances of deep learning have made this problem feasible. For example, the Generative Adversarial Networks (GANs) is a rewarding approach to synthesize images. Nevertheless, it is not trivial when capitalizing on GANs to generate videos. The difficulty originates from the intrinsic structure where a video is a sequence of visually coherent and semantically dependent frames. This motivates us to explore semantic and temporal coherence in designing GANs to generate videos. In this paper, we present a novel Temporal GANs conditioning on Captions, namely TGANs-C, in which the input to the generator network is a concatenation of a latent noise vector and caption embedding, and then is transformed into a frame sequence with 3D spatio-temporal convolutions. Unlike the naive discriminator which only judges pairs as fake or real, our discriminator additionally notes whether the video matches the correct caption. In particular, the discriminator network consists of three discriminators: video discriminator classifying realistic videos from generated ones and optimizes video-caption matching, frame discriminator discriminating between real and fake frames and aligning frames with the conditioning caption, and motion discriminator emphasizing the philosophy that the adjacent frames in the generated videos should be smoothly connected as in real ones. We qualitatively demonstrate the capability of our TGANs-C to generate plausible videos conditioning on the given captions on two synthetic datasets (SBMG and TBMG) and one real-world dataset (MSVD). Moreover, quantitative experiments on MSVD are performed to validate our proposal via Generative Adversarial Metric and human study.

*This work was performed at Microsoft Research Asia. The first two authors made equal contributions to this work.

MM '17, October 23–27, 2017, Mountain View, CA, USA

ACM ISBN 978-1-4503-4906-2/17/10...$15.00
https://doi.org/10.1145/3123266.3127905

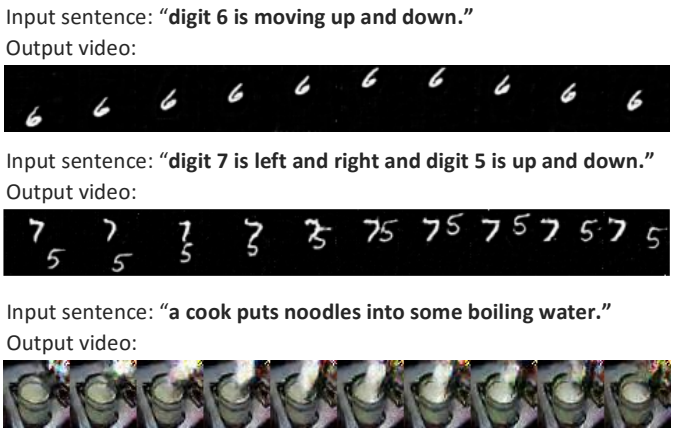

Figure 1: Examples of video generation from captions on Single-Digit Bouncing MNIST GIFs, Two-Digit Bouncing MNIST GIFs and Microsoft Research Video Description Corpus, respectively.

CCS CONCEPTS

• **Information systems** → **Multimedia information systems**; • **Computing methodologies** → *Machine translation*; *Vision for robotics*;

KEYWORDS

Video Generation; Video Captioning; GANs; CNNs

1 INTRODUCTION

Characterizing and modeling natural images and videos remains an open problem in computer vision and multimedia community. One fundamental issue that underlies this challenge is the difficulty to quantify the complex variations and statistical structures in images and videos. This motivates the recent studies to explore Generative Adversarial Nets (GANs) [5] in generating plausible images [4, 18]. Nevertheless, a video is a sequence of frames which additionally contains temporal dependency, making it extremely hard to extend GANs to video domain. Moreover, as videos are often accompanied by text descriptors, e.g., tags or captions, learning video generative models conditioning on text then reduces sampling uncertainties and has a great potential real-world applications. Particularly, we are interested in producing videos from captions in this work, which is a brave new and timely problem. It aims to generate a video which is semantically aligned with the given descriptive sentence as illustrated in Figure 1.

In general, there are two critical issues in video generation employing caption conditioning: temporal coherence across video frames and semantic match between caption and the generated video. The former yields insights into the learning of generative model that the adjacent video frames are often visually and semantically coherent, and thus should be smoothly connected over time. This can be regarded as an intrinsic and generic property to produce a video. The later pursues a model with the capability to create realistic videos which are relevant to the given caption descriptions. As such, the conditioned treatment is taken into account, on one hand to create videos resembling the training data, and on the other, to regularize the generative capacity by holistically harnessing the relationship between caption semantics and video content.

By jointly consolidating the idea of temporal coherence and semantic match in translating text in the form of sentence into videos, this paper extends the recipe of GANs and presents a novel Temporal GANs conditioning on Caption (TGANs-C) framework for video generation, as shown in Figure 2. Specifically, sentence embedding encoded by the Long-Short Term Memory (LSTM) networks is concatenated to the noise vector as an input of the generator network, which produces a sequence of video frames by utilizing 3D convolutions. As such, temporal connections across frames are explicitly strengthened throughout the progress of video generation. In the discriminator network, in addition to determining whether videos are real or fake, the network must be capable of learning to align videos with the conditioning information. In particular, three discriminators are devised, including video discriminator, frame discriminator and motion discriminator. The former two classify realistic videos and frames from the generated ones, respectively, and also attempt to recognize the semantically matched video/frame-caption pairs from mismatched ones. The latter one is to distinguish the displacement between consecutive real or generated frames to further enhance temporal coherence. As a result, the whole architecture of TGANs-C is trained end-to-end by optimizing three losses, i.e., video-level and frame-level matching-aware loss to correct label of real or synthetic video/frames and align video/frames with correct caption, respectively, and temporal coherence loss to emphasize temporal consistency.

The main contribution of this work is the proposal of a new architecture, namely TGANs-C, which is one of the first effort towards generating videos conditioning on captions. This also leads to the elegant views of how to guarantee temporal coherence across generated video frames and how to align video/frame content with the given caption, which are the problems not yet fully understood in the literature. Through an extensive set of quantitative and qualitative experiments, we validate the effectiveness of our TGANs-C model on three different benchmarks.

2 RELATED WORK

We briefly group the related work into two categories: natural image synthesis and video generation. The former draws upon research in synthesizing realistic images by utilizing deep generative models, while the latter investigates generating image sequence/video from scratch.

Image Synthesis. Synthesizing realistic images has been studied and analyzed widely in AI systems for characterizing the pixel level structure of natural images. There are two main directions on automatically image synthesis: Variational Auto-Encoders (VAEs) [10] and Generative Adversarial Networks (GANs) [5]. VAEs is a directed graphical model which firstly constrains the latent distribution of the data to come from prior normal distribution and then generates new samples through sampling from this distribution. This direction is straightforward to train but introduce potentially restrictive assumptions about approximate posterior distribution, always resulting in overly smoothed samples. Deep Recurrent Attentive Writer (DRAW) [9] is one of the early works which utilizes VAEs to generate images with a spatial attention mechanism. Furthermore, Mansimov *et al.* extend this model to generate images conditioning on captions by iteratively drawing patches on a canvas and meanwhile attending to relevant words in the description [12].

GANs can be regarded as the generator network modules learnt with a two-player minimax game mechanism and has shown the distinct ability of producing plausible images [4, 18]. Goodfellow *et al.* propose the theoretical framework of GANs and utilize GANs to generate images without any supervised information in [5]. Although the earlier GANs offer a distinct and promising direction for image synthesis, the results are somewhat noisy and blurry. Hence, Laplacian pyramid is further incorporated into GANs in [4] to produce high quality images. Later in [15], GANs is expended with a specialized cost function for classification, named auxiliary classifier GANs (AC-GANs), for generating synthetic images with global coherence and high diversity conditioning on class labels. Recently, Reed *et al.* utilize GANs for image synthesis based on given text descriptions in [19], enabling translation from character level to pixel level.

Video Generation. When extending the existing generative models (e.g., VAEs and GANs) to video domain, very few works exploit such video generation from scratch task as both the spatial and temporal complex variations need to be characterized, making the problem very challenging. In the direction of VAEs, Mittal *et al.* employ Recurrent VAEs and an attention mechanism in a hierarchical manner to create a temporally dependent image sequence conditioning on captions [13]. For video generation with GANs, a spatio-temporal 3D deconvolutions based GANs is firstly proposed in [25] by untangling the scene's foreground from the background. Most recently, the 3D deconvolutions based GANs is further decomposed into temporal generator consisting of 1D deconvolutional layers and image generator with 2D deconvolutional layers for video generation in [20].

In short, our work in this paper belongs to video generation models capitalizing on adversarial learning. Unlike the aforementioned GANs-based approaches which mainly focus on video synthesis in an unconditioned manner, our

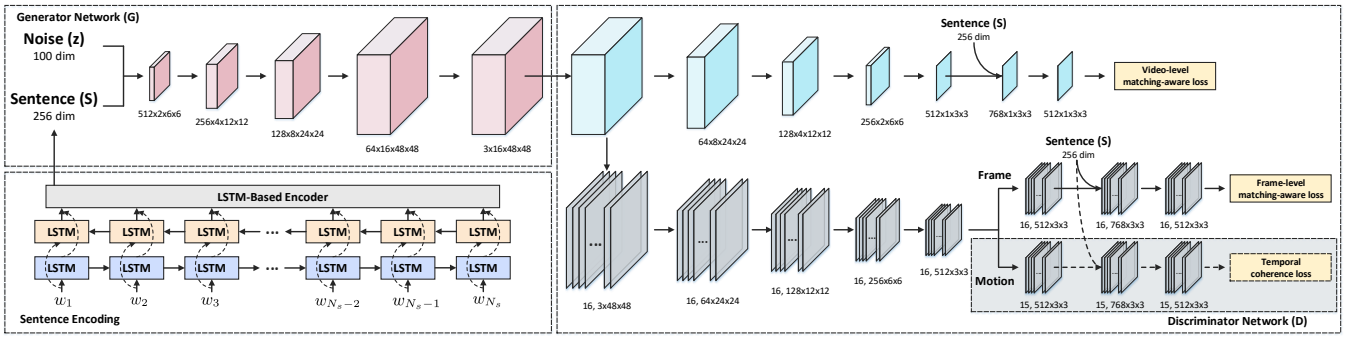

Figure 2: Temporal GANs conditioning on Captions (TGANs-C) framework mainly consists of a generator network G and a discriminator network D (better viewed in color). Given a sentence $\mathcal{S}$, a bi-LSTM is first utilized to contextually embed the input word sequence, followed by a LSTM-based encoder to obtain the sentence representation S. The generator network G tries to synthesize realistic videos with the concatenated input of the sentence representation S and random noise variable z. The discriminator network D includes three discriminators: video discriminator to distinguish real video from synthetic one and align video with the correct caption, frame discriminator to determine whether each frame is real/fake and semantically matched/mismatched with the given caption, and motion discriminator to exploit temporal coherence between consecutive frames. Accordingly, the whole architecture is trained with the video-level matching-aware loss, frame-level matching-aware loss and temporal coherence loss in a two-player minimax game mechanism.

research is fundamentally different in the way that we aim at generating videos conditioning on captions. In addition, we further improve video generation from the aspects of involving frame-level discriminator and strengthening temporal connections across frames.

3 VIDEO GENERATION FROM CAPTIONS

The main goal of our Temporal GANs conditioning on Captions (TGANs-C) is to design a generative model with the ability of synthesizing a temporal coherent frame sequence semantically aligned with the given caption. The training of TGANs-C is performed by optimizing the generator network and discriminator network (video and frame discriminators which simultaneously judge synthetic or real and semantically mismatched or matched with the caption for video and frame) in a two-player minimax game mechanism. Moreover, the temporal coherence prior is additionally incorporated into TGANs-C to produce temporally coherent frame sequence in two different schemes. Therefore, the overall objective function of TGANs-C is composed of three components, i.e., video-level matching-aware loss to correct the label of real or synthetic video and align video with matched caption, frame-level matching-aware loss to further enhance the image reality and semantic alignment with the conditioning caption for each frame, and temporal coherence loss (i.e., temporal coherence constraint loss/temporal coherence adversarial loss) to exploit the temporal coherence between consecutive frames in unconditional/conditional scheme. The whole architecture of TGANs-C is illustrated in Figure 2.

3.1 Generative Adversarial Networks

The basic generative adversarial networks (GANs) consists of two networks: a generator network G that captures the data distribution for synthesizing image and a discriminator network D that distinguishes real images from synthetic ones. In particular, the generator network G takes a latent variable $\mathbf{z}$ randomly sampled from a normal distribution as input and produces a synthetic image $x_{syn} = G(\mathbf{z})$. The discriminator network D takes an image x as input stochastically chosen (with equal probability) from real images or synthetic ones through G and produces a probability distribution $P(S|x) = D(x)$ over the two image sources (i.e., synthetic or real). As proposed in [5], the whole GANs can be trained in a two-player minimax game. Concretely, given an image example x, the discriminator network D is trained to minimize the adversarial loss, i.e., maximizing the log-likelihood of assigning correct source to this example:

$$\begin{aligned} l_a(x) = &-I_{(S=real)} \log\big(P(S = real|x)\big) \\ &-(1 - I_{(S=real)}) \log\big(1 - P(S = real|x)\big), \end{aligned} \tag{1}$$

where the indicator function $I_{condition} = 1$ if *condition* is true; otherwise $I_{condition} = 0$. Meanwhile, the generator network G is trained to maximize the adversarial loss in Eq.(1), targeting for maximally fooling the discriminator network D with its generated synthetic images $\{x_{syn}\}$.

3.2 Temporal GANs Conditioning on Captions (TGANs-C)

In this section, we elaborate the architecture of our TGANs-C, the GANs based generative model consisting of two networks: a generator network G for synthesizing videos conditioning on captions, and a discriminator network D that simultaneously distinguishes real videos/frames from synthetic ones and

aligns the input videos/frames with semantically matching captions. Moreover, two different schemes for modeling temporal coherence across frames are incorporated into TGANs-C for video generation.

3.2.1 Generator Network. Suppose we have an input sentence $\mathcal{S}$, where $\mathcal{S} = \{w_1, w_2, ..., w_{N_s-1}, w_{N_s}\}$ including N_s words. Let $\mathbf{w}_t \in \mathbb{R}^{d_w}$ denote the d_w-dimensional "one-hot" vector (binary index vector in a vocabulary) of the t-th word in sentence $\mathcal{S}$, thus the dimension of the textual feature $\mathbf{w}_t$, i.e., d_w, is the vocabulary size. Taking the inspiration from recent success of Recurrent Neural Networks (RNN) in image/video captioning [16, 17, 26–28], we first leverage the bidirectional LSTM (bi-LSTM) [21] to contextually embed each word and then encode the embedded word sequence into the sentence representation $\mathbf{S}$ via LSTM. In particular, the bi-LSTM consisting of forward and backward LSTMs [7] is adopted here. The forward LSTM reads the input word sequence in its natural order (from w_1 to w_{N_s}) and then calculates the forward hidden states sequence $\{\overrightarrow{h}_1, \overrightarrow{h}_2, ..., \overrightarrow{h}_{N_s}\}$, whereas the backward LSTM produces the backward hidden states sequence $\{\overleftarrow{h}_1, \overleftarrow{h}_2, ..., \overleftarrow{h}_{N_s}\}$ with the input sequence in the reverse order (from w_{N_s} to w_1). The outputs of forward LSTM and backward LSTM are concatenated as the contextually embedded word sequence $\{h_1, h_2, ..., h_{N_s}\}$, where $h_t = \left[\overrightarrow{h}_t^\top, \overleftarrow{h}_t^\top\right]^\top$. Then, we feed the embedded word sequence into the next LSTM-based encoder and treat the final LSTM output as the sentence representation $\mathbf{S} \in \mathbb{R}^{d_s}$. Note that both bi-LSTM and LSTM-based encoder are pre-learnt with sequence auto-encoder [3] in an unsupervised learning manner. Concretely, a LSTM-based decoder is additionally attached on the top of LSTM-based encoder for reconstructing the original word sequence. Such LSTM-based decoder will be removed and only the bi-LSTM and LSTM-based encoder are reserved for representing sentences with improved generalization ability after pre-training over large quantities of sentences.

Next, given the input sentence $\mathbf{S}$ and random noise variable $\mathbf{z} \in \mathbb{R}^{d_z} \sim \mathcal{N}(0, 1)$, a generator network G is devised to synthesize a frame sequence: $\{\mathbb{R}^{d_s}, \mathbb{R}^{d_z}\} \rightarrow \mathbb{R}^{d_c \times d_l \times d_h \times d_d}$ where d_c, d_l, d_h and d_d denote the channels number, sequence length, height and width of each frame, respectively. To model the spatio-temporal information within videos, the most natural way is to utilize the 3D convolutions filters [24] with deconvolutions [29] which can simultaneously synthesize the spatial information via 2D convolutions filters and provide temporal invariance across frames. Particularly, the generator network G first encapsulates both the random noise variable $\mathbf{z}$ and input sentence $\mathbf{S}$ into a fixed-length input latent variable $\mathbf{p}$, which is applied with feature transformation and concatenation, and then synthesizes the corresponding video $v_{syn} = G(\mathbf{z}, \mathbf{S})$ based on the input $\mathbf{p}$ through 3D deconvolutional layers. The fixed-length input latent variable $\mathbf{p}$ is computed as

$$\mathbf{p} = \left[\mathbf{z}^\top, \mathbf{S}^\top \mathbf{W}_s\right]^\top \in \mathbb{R}^{d_z + d_p}, \tag{2}$$

where $\mathbf{W}_s \in \mathbb{R}^{d_s \times d_p}$ is the transformation matrix for sentence representation. Accordingly, the generator network G produces the synthetic video $v_{syn} = \{f_{syn}^1, f_{syn}^2, ..., f_{syn}^{d_l}\}$ conditioning on sentence $\mathcal{S}$ where $f_{syn}^i \in \mathbb{R}^{d_c \times d_h \times d_d}$ represents i-th synthetic frame.

3.2.2 Discriminator Network. The discriminator network D is designed to enable three main abilities: (1) distinguishing real video from synthetic one and aligning video with the correct caption, (2) determining whether each frame is real/fake and semantically matched/mismatched with the conditioning caption, (3) exploiting the temporal coherence across consecutive real frames. To address the three crucial points, three basic discriminators are particularly devised:

- Video discriminator $D_0(v, \mathcal{S})$ ($\{\mathbb{R}^{d_v}, \mathbb{R}^{d_s}\} \rightarrow [0, 1]$): D_0 first encodes input video $v \in \mathbb{R}^{d_v}$ into a video-level tensor $\mathbf{m}_v$ with a size of $d_{c_0} \times d_{l_0} \times d_{h_0} \times d_{d_0}$ via 3D convolutional layers. Then, the video-level tensor $\mathbf{m}_v$ is augmented with the conditioning caption $\mathbf{S}$ for discriminating whether the input video is real and simultaneously semantically matched with the given caption.
- Frame discriminator $D_1\left(f^i, \mathcal{S}\right)$ ($\{\mathbb{R}^{d_f}, \mathbb{R}^{d_s}\} \rightarrow [0, 1]$): D_1 transforms each frame $f^i \in \mathbb{R}^{d_f}$ in v into a frame-level tensor $\mathbf{m}_{f^i} \in \mathbb{R}^{d_{c_0} \times d_{h_0} \times d_{d_0}}$ through 2D convolutional layers and then augments frame-level tensor $\mathbf{m}_{f^i}$ with the conditioning caption $\mathbf{S}$ to recognize the real frames with matched caption.
- Motion discriminator $D_2\left(f^i, f^{i-1}\right)$ ($\{\mathbb{R}^{d_f}, \mathbb{R}^{d_f}\} \rightarrow \mathbb{R}^{d_{c_0} \times d_{h_0} \times d_{d_0}}$): D_2 distills the 2D motion tensor $\overrightarrow{\mathbf{m}}_{f^i}$ to represent the temporal dynamics across consecutive frames f^i and f^{i-1}. Please note that we adopt the most direct way to measure such motion variance between two consecutive frames by subtracting previous frame-level tensor from current one (i.e., $\overrightarrow{\mathbf{m}}_{f^i} = \mathbf{m}_{f^i} - \mathbf{m}_{f^{i-1}}$).

Specifically, in the training epoch, we can easily obtain a set of real-synthetic video triplets $\mathcal{T}$ according to the prior given captions, where each tuple $\{v_{syn^+}, v_{real^+}, v_{real^-}\}$ consists of one synthetic video v_{syn^+} conditioning on given caption $\mathcal{S}$, one real video v_{real^+} described by the same caption $\mathcal{S}$, and one real video v_{real^-} described by different caption from $\mathcal{S}$. Therefore, three video-caption pairs are generated based on the caption $\mathcal{S}$ and its corresponding video tuple: the synthetic and semantically matched pair $\{v_{syn^+}, \mathcal{S}\}$, real and semantically matched pair $\{v_{real^+}, \mathcal{S}\}$, and another real but semantically mismatched pair $\{v_{real^-}, \mathcal{S}\}$. Each video-caption pair $\{v, \mathcal{S}\}$ is then set as the input to the discriminator network D, followed by three kinds of losses to be optimized and each for one discriminator accordingly.

Video-level matching-aware loss. Noticing that the input video-caption pair $\{v, \mathcal{S}\}$ might not only be from distinctly sources (i.e., real or synthetic), but also contain matched or mismatched semantics. However, the conventional discriminator network can only differentiate the video sources

without any explicit notion of the semantic relationship between video content and caption. Taking the inspiration from the matching-aware discriminator in [19], we elaborate the video-level matching-aware loss for video discriminator D_0 to learn better alignment between video and the conditioning caption. In particular, for the video discriminator D_0, the conditioning caption $\mathcal{S}$ is first transformed with the embedding function $\varphi_0(\mathbf{S}) \in \mathbb{R}^{d_{s_0}}$ followed by rectification. Then the embedded sentence representation is spatially replicated to construct a $d_{s_0} \times d_{l_0} \times d_{h_0} \times d_{d_0}$ tensor, which is further concatenated with the video-level tensor $\mathbf{m}_v$ along the channel dimension. Finally the probability of recognizing real video with matched caption $D_0(v, \mathcal{S})$ is measured via a $1 \times 1 \times 1$ convolution followed by rectification and a $d_{l_0} \times d_{h_0} \times d_{d_0}$ convolution. Hence, given the real-synthetic video triplet $\{v_{syn^+}, v_{real^+}, v_{real^-}\}$ and the conditioning caption $\mathcal{S}$, the video-level matching-aware loss is measured as

$$\begin{aligned}\mathcal{L}_v = -\tfrac{1}{3}\big[&\log\left(D_0\left(v_{real^+}, \mathcal{S}\right)\right) + \log\left(1 - D_0\left(v_{real^-}, \mathcal{S}\right)\right) \\ &+ \log(1 - D_0(v_{syn^+}, \mathcal{S}))\big]\end{aligned}. \quad (3)$$

By minimizing this loss over positive video-caption pair (i.e., $\{v_{real^+}, \mathcal{S}\}$) and negative video-caption pairs (i.e., $\{v_{syn^+}, \mathcal{S}\}$ and $\{v_{real^-}, \mathcal{S}\}$), the video discriminator D_0 is trained to not only recognize each real video from synthetic ones but also classify semantically matched video-caption pair from mismatched ones.

Frame-level matching-aware loss. To further enhance the frame reality and semantic alignment with the conditioning caption for each frame, a frame-level matching-aware loss is involved here which enforces the frame discriminator D_1 to discriminate whether each frame of the input video is both real and semantically matched with the caption. For the frame discriminator D_1, similar to D_0, an embedding function $\varphi_1(\mathbf{S}) \in \mathbb{R}^{d_{s_0}}$ is utilized to transform the conditioning caption $\mathcal{S}$ into the low-dimensional representation. Then we replicate the sentence embedding spatially to concatenate it with the frame-level tensor of each frame along the channel dimension. Accordingly, the final probability of recognizing real frame with matched caption $D_0\left(f^i, \mathcal{S}\right)$ is achieved through a 1×1 convolution followed by rectification and a $d_{h_0} \times d_{d_0}$ convolution. Therefore, given the real-synthetic video triplet $\{v_{syn^+}, v_{real^+}, v_{real^-}\}$ and the conditioning caption $\mathcal{S}$, we calculate the frame-level matching-aware loss as

$$\begin{aligned}\mathcal{L}_f = -\tfrac{1}{3d_l}\big[&\sum_{i=1}^{d_l}\log(D_1(f^i_{real^+}, \mathcal{S})) + \sum_{i=1}^{d_l}\log(1 - D_1(f^i_{real^-}, \mathcal{S})) \\ &+ \sum_{i=1}^{d_l}\log(1 - D_1(f^i_{syn^+}, \mathcal{S}))\big]\end{aligned}, \quad (4)$$

where $f^i_{real^+}$, $f^i_{real^-}$ and $f^i_{syn^+}$ denotes the i-th frame in v_{real^+}, v_{real^-} and v_{syn^+}, respectively.

Temporal coherence loss. Temporal coherence is one generic prior for video modeling, which reveals the intrinsic characteristic of video that the consecutive video frames are usually visually and semantically coherent. To incorporate this temporal coherence prior into TGANs-C for video generation, we consider two kinds of schemes on the basis of motion discriminator $D_2\left(f^i, f^{i-1}\right)$.

(1) *Temporal coherence constraint loss.* Motivated by [14], the similarity of two consecutive frames can be directly defined according to the Euclidean distances between their frame-level tensors, i.e., the magnitude of motion tensor:

$$\mathcal{D}\left(f^i, f^{i-1}\right) = \left\|\mathbf{m}_{f^i} - \mathbf{m}_{f^{i-1}}\right\|_2^2 = \left\|\overrightarrow{\mathbf{m}}_{f^i}\right\|_2^2. \quad (5)$$

Then, given the real-synthetic video triplet, we characterize the temporal coherence of the synthetic video v_{syn^+} as a constraint loss by accumulating the Euclidean distances over every two consecutive frames:

$$\mathcal{L}_t^{(1)} = \frac{1}{d_l - 1}\sum_{i=2}^{d_l}\mathcal{D}(f^i_{syn^+}, f^{i-1}_{syn^+}). \quad (6)$$

Please note that the temporal coherence constraint loss is designed only for optimizing generator network G. By minimizing this loss of synthetic video, the generator network G is enforced to produce temporally coherent frame sequence.

(2) *Temporal coherence adversarial loss.* Different from the first scheme formulating temporal coherence as a monotonous constraint in an unconditional manner, we further devise an adversarial loss to flexibly emphasize temporal consistency conditioning on the given caption. Similar to frame discriminator D_1, the motion tensor $\overrightarrow{\mathbf{m}}_{f^i}$ in motion discriminator D_2 is first augmented with embedded sentence representation $\varphi_2(\mathbf{S})$. Next, such concatenated tensor representation is leveraged to measure the final probability $\Phi_2(\overrightarrow{\mathbf{m}}_{f^i}, \mathcal{S})$ of classifying the temporal dynamics between consecutive frames as real ones conditioning on the given caption. Thus, given the real-synthetic video triplet $\{v_{syn^+}, v_{real^+}, v_{real^-}\}$ and the conditioning caption $\mathcal{S}$, the temporal coherence adversarial loss is measured as

$$\begin{aligned}\mathcal{L}_t^{(2)} = -\tfrac{1}{3(d_l-1)}\big[&\sum_{i=2}^{d_l}\log(\Phi_2(\overrightarrow{\mathbf{m}}_{f^i_{real^+}}, \mathcal{S})) \\ &+ \sum_{i=2}^{d_l}\log(1 - \Phi_2(\overrightarrow{\mathbf{m}}_{f^i_{real^-}}, \mathcal{S})) \\ &+ \sum_{i=2}^{d_l}\log(1 - \Phi_2(\overrightarrow{\mathbf{m}}_{f^i_{syn^+}}, \mathcal{S}))\big]\end{aligned}, \quad (7)$$

where $\overrightarrow{\mathbf{m}}_{f^i_{real^+}}$, $\overrightarrow{\mathbf{m}}_{f^i_{real^-}}$ and $\overrightarrow{\mathbf{m}}_{f^i_{syn^+}}$ denotes the motion tensor in v_{real^+}, v_{real^-} and v_{syn^+}, respectively. By minimizing the temporal coherence adversarial loss, the temporal discriminator D_2 is trained to not only recognize the temporal dynamics across synthetic frames from real ones but also align the temporal dynamics with the matched caption.

3.2.3 Optimization. The overall training objective function of TGANs-C integrates the video-level matching-aware loss in Eq.(3), frame-level matching-aware loss in Eq.(4) and temporal coherence constraint loss/temporal coherence adversarial loss in Eq.(6)/Eq.(7). As our TGANs-C is a variant of the GANs architecture, we train the whole architecture in a two-player minimax game mechanism. For the discriminator network D, we update its parameters according to the

following overall loss

$$\hat{\mathcal{L}}_D^{(1)} = \sum_{\mathcal{T}} \frac{1}{2} \left(\mathcal{L}_v + \mathcal{L}_f \right), \tag{8}$$

$$\hat{\mathcal{L}}_D^{(2)} = \sum_{\mathcal{T}} \frac{1}{3} \left(\mathcal{L}_v + \mathcal{L}_f + \mathcal{L}_t^{(2)} \right), \tag{9}$$

where $\mathcal{T}$ is the set of real-synthetic video triplets, $\hat{\mathcal{L}}_D^{(1)}$ and $\hat{\mathcal{L}}_D^{(2)}$ denotes the discriminator network D's overall adversarial loss in unconditional scheme (i.e., TGANs-C with temporal coherence Constraint loss (TGANs-C-C)) and conditional scheme (i.e., TGANs-C with temporal coherence Adversarial loss (TGANs-C-A)), respectively. By minimizing this term, the discriminator network D is trained to classify both videos and frames with correct sources, and simultaneously align videos and frames with semantically matching captions. Moreover, for TGANs-C-A, the discriminator network D is additionally enforced to distinguish the temporal dynamics across frames with correct sources and also align the temporal dynamics with the matched captions.

For the generator network G, its parameters are adjusted with the following overall loss

$$\hat{\mathcal{L}}_G^{(1)} = - \sum_{v_{syn^+} \in \mathcal{T}} \frac{1}{3} \big[\log(D_0(v_{syn^+}, \mathcal{S})) + \frac{1}{d_l} \sum_{i=1}^{d_l} \log(D_1(f_{syn^+}^i, \mathcal{S})) - \frac{1}{d_l - 1} \sum_{i=2}^{d_l} \mathcal{D}(f_{syn^+}^i, f_{syn^+}^{i-1}) \big], \tag{10}$$

$$\hat{\mathcal{L}}_G^{(2)} = - \sum_{v_{syn^+} \in \mathcal{T}} \frac{1}{3} \big[\log(D_0(v_{syn^+}, \mathcal{S})) + \frac{1}{d_l} \sum_{i=1}^{d_l} \log(D_1(f_{syn^+}^i, \mathcal{S})) + \frac{1}{d_l - 1} \sum_{i=2}^{d_l} \log(\Phi_2(\vec{\mathbf{m}}_{f_{syn^+}^i}, \mathcal{S})) \big], \tag{11}$$

where $\hat{\mathcal{L}}_G^{(1)}$ and $\hat{\mathcal{L}}_G^{(2)}$ denotes the generator network G's overall adversarial loss in TGANs-C-C and TGANs-C-A, respectively. The generator network G is trained to fool the discriminator network D on videos/frames source prediction with its synthetic videos/frames and meanwhile align synthetic videos/frames with the conditioning captions. Moreover, for TGANs-C-C, the consecutive synthetic frames are enforced to be similar in an unconditional scheme, while for TGANs-C-A, it additionally aims to fool D on temporal dynamics source prediction with the synthetic videos in a conditional scheme. The training process of TGANs-C is given in Algorithm 1.

Algorithm 1 The training of Temporal GANs conditioning on Captions (TGANs-C)

Given the number of maximum training iteration T.
for $t = 1$ to T **do**
  Fetch input batch with sampled video-sentence pairs $\{(\mathcal{S}, v_{real^+})\}$.
  for Each video-sentence pair $(\mathcal{S}, v_{real^+})$ **do**
    Get the random noise variable $\mathbf{z} \sim \mathcal{N}(0, 1)$.
    Produce the synthetic video $v_{syn^+} = G(\mathbf{z}, \mathbf{S})$ conditioning on the caption $\mathcal{S}$ via the generator network G.
    Randomly select one real video v_{real^-} described by a different caption from $\mathcal{S}$.
  end for
  Obtain all the real-synthetic tuple $\{v_{syn^+}, v_{real^+}, v_{real^-}\}$ with the corresponding caption $\mathcal{S}$, denoted as $\mathcal{T}$ in total.
  Compute video-level matching-aware loss via Eq. (3).
  Compute frame-level matching-aware loss via Eq. (4).
  -Scheme 1: TGANs-C-C
    Compute temporal coherence constraint loss via Eq. (6).
    Update the discriminator network D w.r.t loss in Eq. (8).
    Update the generator network G w.r.t loss in Eq. (10).
  -Scheme 2: TGANs-C-A
    Compute temporal coherence adversarial loss via Eq. (7).
    Update the discriminator network D w.r.t loss in Eq. (9).
    Update the generator network G w.r.t loss in Eq. (11).
end for

3.3 Testing Epoch

After the optimization of TGANs-C, we can obtain the learnt generator network G. Thus, given a test caption $\hat{\mathcal{S}}$, the bi-LSTM is first utilized to contextually embed the input word sequence, followed by a LSTM-based encoder to achieve the sentence representation $\hat{\mathbf{S}}$. The sentence representation $\hat{\mathbf{S}}$ is then concatenated with the random noise variable $\mathbf{z}$ as in Eq.(2) and finally fed into the generator network G to produce the synthetic video $\hat{v}_{syn} = \{\hat{f}_{syn}^1, \hat{f}_{syn}^2, ..., \hat{f}_{syn}^{d_l}\}$.

4 EXPERIMENTS

We evaluate and compare our proposed TGANs-C with state-of-the-art approaches by conducting video generation task on three datasets of progressively increasing complexity: Single-Digit Bouncing MNIST GIFs (SBMG) [13], Two-digit Bouncing MNIST GIFs (TBMG) [13], and Microsoft Research Video Description Corpus (MSVD) [2]. The first two are recently released GIF-based datasets consisting of MNIST [11] digits moving frames and the last is a popular video captioning benchmark of YouTube videos.

4.1 Datasets

SBMG. Similar to priors works [22, 23] in generating synthetic dataset, SBMG is produced by having single handwritten digit bouncing inside a 64×64 frame. It is composed of 12,000 GIFs and every GIF is 16 frames long, which contains a single 28×28 digit moving left-right or up-down. The starting position of the digit is chosen uniformly at random. Each GIF is accompanied with single sentence describing the digit and its moving direction, as shown in Figure 3(a).

TBMG. TBMG is an extended synthetic dataset of SBMG which contains two handwritten digits bouncing. The generation process is the same as SBMG and the two digits within each GIF move left-right or up-down separately. Figure 3(b) shows two exemplary GIF-caption pairs in TBMG.

MSVD. MSVD contains 1,970 video snippets collected from YouTube. There are roughly 40 available English descriptions per video. In experiments, we manually filter out the videos about cooking and generate a subset of 518 cooking videos. Following the settings in [6], our cooking subset is split with 363 videos for training and 155 for testing. Since video generation is a challenging problem, we assembled this subset with cooking scenario to better diagnose pros and

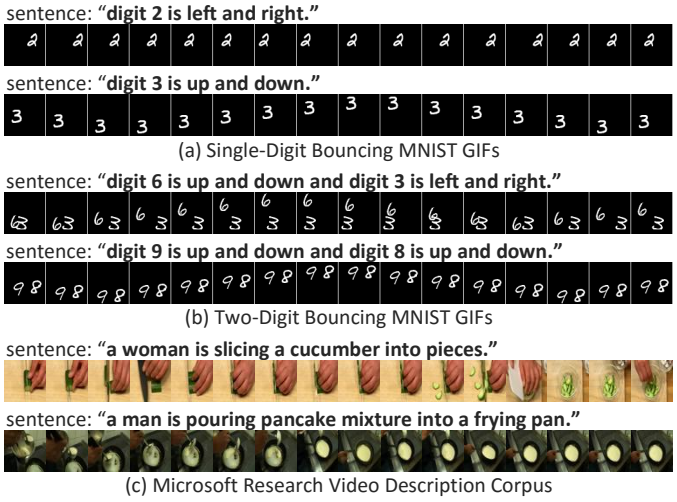

Figure 3: (a)—(c): Exemplary video-caption pairs from three benchmarks: (a) Single-Digit Bouncing MNIST GIFs; (b) Two-Digit Bouncing MNIST GIFs; (c) Microsoft Research Video Description Corpus.

cons of models. We randomly select two examples from this subset and show them in Figure 3(c).

4.2 Experimental Settings

Parameter Settings. We uniformly sample $d_l = 16$ frames for each GIF/video and each word in the sentence is represented as "one-hot" vector. The architecture of our TGANs-C is mainly developed based on [18, 19]. We resize all the GIFs/videos in three datasets with 48×48 pixels. In particular, for sentence encoding, the dimension of the input and hidden layers in bi-LSTM and LSTM-based encoder are all set to 256. For the generator network G, the dimension of random noise variable $\mathbf{z}$ is 100 and the dimension of sentence embedding in generator network d_p is 256. For the discriminator network D, we set the size of video-level tensor $\mathbf{m}_v$ in video discriminator D_0 as $512 \times 1 \times 3 \times 3$ and the size of frame-level tensor $\mathbf{m}_{fi}$ in frame discriminator D_1 is as $512 \times 3 \times 3$.

Implementation Details. We mainly implement our proposed method based on Theano [1], which is one of widely adopted deep learning frameworks. Following the standard settings in [18], we train our TGANs-C models on all datasets by utilizing Adam optimizer with a mini-batch size of 64. All weights were initialized from a zero-centered Normal distribution with standard deviation 0.02 and the slope of the leak was set to 0.2 in the LeakyReLU. We set the learning rate and momentum as 0.0002 and 0.9, respectively.

Evaluation Metric. For the quantitative evaluation of video generation, we adopt Generative Adversarial Metric (GAM) [8] which can directly compare two generative adversarial models by having them engage in a "battle" against each other. Given two generative adversarial models $M_1 = \{(\widetilde{G}_1, \widetilde{D}_1)\}$ and $M_2 = \{(\widetilde{G}_2, \widetilde{D}_2)\}$, two kinds of ratios between the discriminative scores of the two models are measured as:

$$r_{test} = \frac{\epsilon\left(\widetilde{D}_1\left(\mathbf{x}_{test}\right)\right)}{\epsilon\left(\widetilde{D}_2\left(\mathbf{x}_{test}\right)\right)} \text{ and } r_{sample} = \frac{\epsilon\left(\widetilde{D}_1\left(\widetilde{G}_2\left(\mathbf{z}\right)\right)\right)}{\epsilon\left(\widetilde{D}_2\left(\widetilde{G}_1\left(\mathbf{z}\right)\right)\right)}, \quad (12)$$

where $\epsilon(\bullet)$ denotes the classification error rate and $\mathbf{x}_{test}$ is the testing set. The test ratio r_{test} shows which model generalizes better on test data and the sample ratio r_{sample} reveals which model can fool the other model more easily. Finally, the GAM evaluation metric judges the winner as:

$$\text{winner} = \begin{cases} M_1 & \text{if } r_{sample} < 1 \text{ and } r_{test} \simeq 1 \\ M_2 & \text{if } r_{sample} > 1 \text{ and } r_{test} \simeq 1 \\ \text{Tie} & \text{otherwise} \end{cases}. \quad (13)$$

4.3 Compared Approaches

To empirically verify the merit of our TGANs-C, we compared the following state-of-the-art methods.

(1) Synchronized Deep Recurrent Attentive Writer (Sync-DRAW) [13]: Sync-DRAW is a VAEs-based model for video generation conditioning on captions which utilizes Recurrent VAEs to model spatio-temporal relationship and a separate attention mechanism to capture local saliency.

(2) Generative Adversarial Network for Video (VGAN) [25]: The original VGAN attempts to leverage the spatio-temporal convolutional architecture to design a GANs-based generative model for video generation in an unconditioned manner. Here we additionally incorporate the matching-aware loss into the discriminator network of basic VGAN and enable this baseline to generate videos conditioning on captions.

(3) Generative Adversarial Network with Character-Level Sentence encoder (GAN-CLS) [19]: GAN-CLS is originally designed for image synthesis from text descriptions by utilizing DC-GAN and a hybrid character-level convolutional-recurrent neural network for text encoding. We directly extend this architecture by replacing 2D convolutions with 3D spatio-temporal convolutions for text-conditional video synthesis.

(4) Temporal GANs conditioning on Captions (TGANs-C) is our proposal in this paper which includes two runs in different schemes: TGANs-C with temporal coherence constraint loss (TGANs-C-C) and TGANs-C with temporal coherence adversarial loss (TGANs-C-A). Two slightly different settings of TGANs-C are named as TGANs-C_1 and TGANs-C_2. The former is trained with only video-level matching-aware loss, while the latter is more similar to TGANs-C that only excludes the temporal coherence loss.

4.4 Optimization Analysis

Different from the traditional discriminative models which have a particularly well-behaved gradient, our TGANs-C is optimized with a complex two-player minimax game. Hence, we depict the evolution of the generator network G at the training stage to illustrate the convergence of our TGANs-C. Concretely, we randomly sample one random noise variable $\mathbf{z}$ and caption $\mathcal{S}$ before training, and then leverage them to produce synthetic videos via the generator networks G of TGANs-C-A at different iterations on TBMG. As shown in Figure 4, the quality of synthetic videos does improve

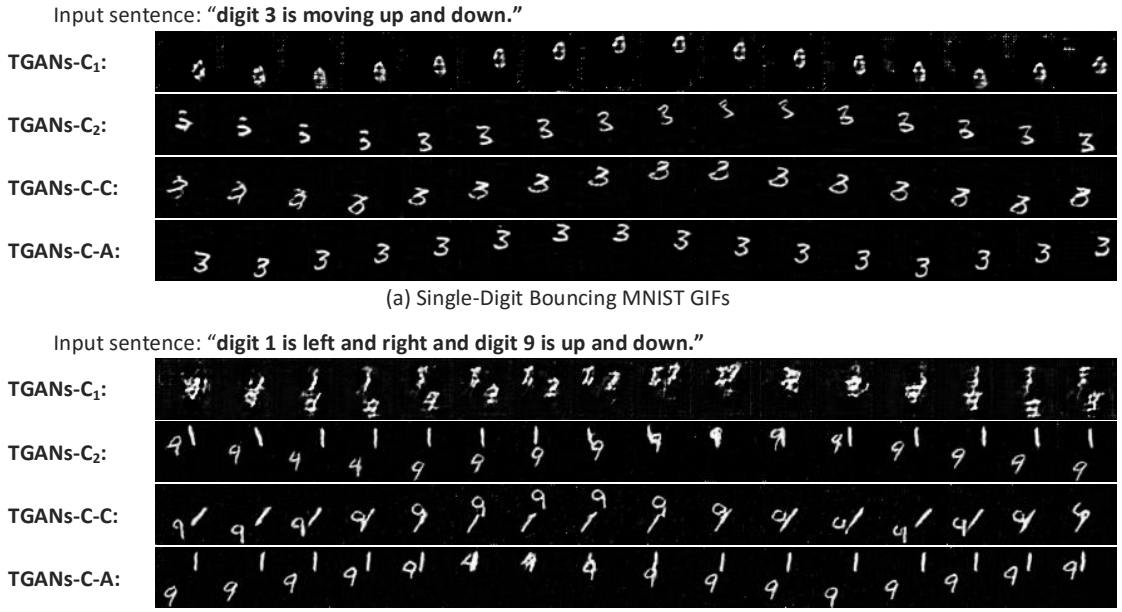

Figure 5: Examples of generated videos by our four TGANs-C runs on (a) Single-Digit Bouncing MNIST GIFs and (b) Two-digit Bouncing MNIST GIFs.

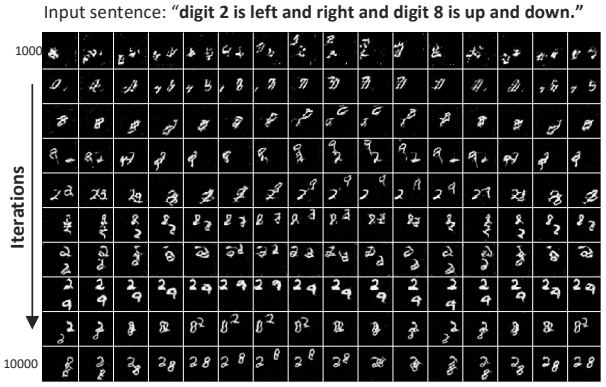

Figure 4: Evolution of synthetic results of the generator network G with the increase of the iteration on TBMG dataset. Both of the input random noise variable z and caption $\mathcal{S}$ are fixed. Each row denotes one synthetic video and the results are shown every 1,000 iterations.

as the iterations increase. Specifically, after 9,000 iterations, the generater network G consistently synthesizes plausible videos by reproducing the visual appearances and temporal dynamics of handwritten digits conditioning on the caption.

4.5 Qualitative Evaluation

We then visually examine the quality of the results and compare among our four internal TGANs-C runs on SBMG and TBMG datasets. The examples of generated videos are shown in Figure 5. Given the input sentence of "digit 3 is moving up and down" in Figure 5(a), all the four runs can interpret the temporal track of forming single-digit bouncing videos. TGANs-C_1 which only judges real or fake on video level and aligns video with the caption performs the worst among all the models and the predicted frames tend to be blurry. By additionally distinguishing frame-level realness and optimizing frame-caption matching, TGANs-C_2 is capable of producing videos in which each frame is clear but the shape of the digit sometimes changes over time. Compared to TGANs-C_2, TGANs-C-C emphasizes the coherence across adjacent frames by further regularizing the similarity in between. As a result, the frames generated by TGANs-C-C are more consistent than TGANs-C_2 particularly of the digit in the frames, but on the other hand, the temporal coherence constraint exploited in TGANs-C-C is in a brute-force manner, making the generated videos monotonous and not that real. TGANs-C-A, in comparison, is benefited from the mechanism of adversarially modeling temporal connections. The chance that a video is gradually formed as real is better.

Figure 5(b) shows the generated videos by our four TGANs-C runs conditioning on the caption of "digit 1 is left and right and digit 9 is up and down." Similar to the observations on single-digit bouncing videos, the four runs could also model the temporal dynamics of two-digit bouncing scenarios. When taking temporal smoothness into account, the quality of the videos generated by TGANs-C-C and TGANs-C-A is enhanced, as compared to the videos produced by TGANs-C_1 and TGANs-C_2. In addition, TGANs-C-A generates more realistic videos than TGANs-C-C, verifying the effectiveness of learning temporal coherence in an adversarial fashion.

Next, we compare with the three baselines on MSVD dataset. In view that TGANs-C-A consistently performs the best in our internal comparisons, we refer to this run as TGANs-C in the following evaluations. The comparisons of generated videos by different approaches are shown in Figure

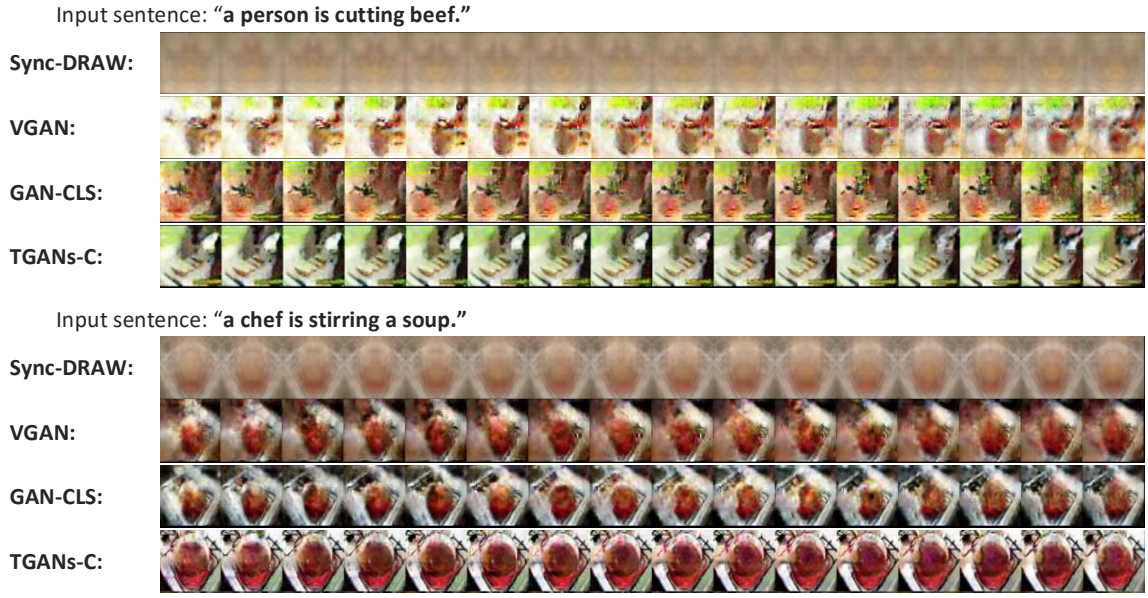

Figure 6: Examples of generated videos by different approaches on MSVD dataset.

6. We can easily observe that the videos generated by our TGANs-C have higher quality compared to the other models. The created frames by Sync-DRAW are very blurry since VAEs are biased towards generating smooth frames and the method does not present all the objects in the frames. The approach of VGAN generates the frames which tend to be fairly sharp. However, the background of the frames is stationary as VGAN enforces a static background and moving foreground, making it vulnerable to produce videos with background movement. Compared to GAN-CLS which only involves video-level matching-aware discriminator, our TGANs-C takes the advantages of additionally exploring frame-level matching-aware discriminator and temporal coherence across frames, and thus generates more realistic videos.

4.6 Human Evaluation

To better understand how satisfactory are the videos generated from different methods, we also conducted a human study to compare our TGANs-C against three approaches, i.e., Sync-DRAW, VGAN and GAN-CLS. A total number of 30 evaluators (15 females and 15 males) from different education backgrounds, including computer science (8), management (4), business (4), linguistics (4), physical education (1), international trade (1) and engineering (8), are invited and a subset of 500 sentences is randomly selected from testing set of MSVD dataset for the subjective evaluation.

We show all the evaluators the four videos generated by each approach plus the given caption and ask them to rank all the videos from 1 to 4 (good to bad) with respect to the three criteria: 1) Reality: how realistic are these generated videos? 2) Relevance: whether the videos are relevant to the given caption? 3) Coherence: judge the temporal connection and readability of the videos. To make the annotation as objective as possible, the four generated videos conditioning on each sentence are assigned to three evaluators and the final ranking is averaged on the three annotations. Furthermore, we average the ranking on each criterion of all the generated videos by each method and obtain three metrics. Table 1 lists the results of the user study on MSVD dataset. Overall, our TGANs-C is clearly the winner across all the three criteria.

Table 1: The user study on three criteria: 1) Reality - how realistic are these generated videos? 2) Relevance - whether the videos are relevant to the given caption? 3) Coherence - judge the temporal connection and readability of the videos. The average ranking (lower is better) on each criterion of all the generated videos by each approach is reported.

Methods	Reality	Relevance	Coherence
Sync-DRAW	3.95	3.93	3.90
VGAN	2.21	2.29	2.23
GAN-CLS	2.08	1.97	2.01
TGANs-C	**1.76**	**1.81**	**1.86**

4.7 Quantitative Evaluation

To further quantitatively verify the effectiveness of our proposed model, we compare our TGANs-C with two generative adversarial baselines (i.e., VGAN and GAN-CLS) in terms of GAM evaluation metric on MSVD dataset. As the method of Sync-DRAW produces videos by VAEs-based architecture rather than generative adversarial scheme, it is excluded in this comparison. The quantitative results are summarized in Table 2. Overall, considering the "battle" between our TGANs-C and the other two baselines, the sample ratios r_{sample} are both less than one, indicating that TGANs-C can produce more authentic synthetic videos and fool the other two models more easily. The results basically verify the advantages of exploiting frame-level realness, frame-caption matching and the temporal coherence across adjacent frames for video generation. Moreover, when comparing between the two 3D-based baselines, GAN-CLS beats VGAN easily.

Table 2: Model Evaluation with GAM metric on MSVD.

Battler	$\mathbf{r_{test}}$	$\mathbf{r_{sample}}$	Winner
GAN-CLS vs VGAN	1.08	0.89	GAN-CLS
TGANs-C vs VGAN	1.09	0.39	TGANs-C
TGANs-C vs GAN-CLS	0.96	0.53	TGANs-C

This somewhat reveals the weakness of VGAN, where the architecture is devised with the brute-force assumption that the background is stationary and only foreground moves, making it hard to mimic the real-word videos with dynamic background. Another important observation is that for the "battle" between each two runs, the test ratio r_{test} is consistently approximately equal to one. This assures that none of the discriminator networks D in these runs is overfitted more than the other, i.e., the corresponding sample ratios r_{sample} are applicable and not biased for evaluating generative adversarial models.

5 CONCLUSIONS

Synthesizing images or videos will be crucial for the next generation of multimedia systems. In this paper, we have presented the Temporal GANs conditioning on Captions (TGANs-C) architecture, succeeded in generating videos that correspond to a given input caption. Our model expands on adversarial learning paradigm from three aspects. First, we extend 2D generator network to 3D for explicitly modeling spatio-temporal connections in videos. Second, in addition to naive discriminator network which only judges fake or real, ours further evaluate whether the generated videos or frames match the conditioning caption. Finally, to guarantee the adjacent frames coherently formed over time, the motion information between consecutive real or generated frames is taken into account in the discriminator network. Extensive quantitative and qualitative experiments conducted on three datasets validate our proposal and analysis. Moreover, our approach creates videos with better quality by a user study from 30 human subjects.

Future works will focus, first of all, on improving visual discriminability of our model, i.e., synthesize higher resolution videos. A promising route to explore will be that of decomposing the problem into several stages, where the shape or basic color based on the given caption is sketched in the primary stages and the advanced stages rectify the details of videos. Second, how to generate videos conditioning on open-vocabulary caption is expected. Last but not least, extending our framework to audio domain should be also interesting.

Acknowledgments. This work was supported in part by 973 Program under contract No. 2015CB351803 and NSFC under contract No. 61325009.

REFERENCES

[1] Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien, Justin Bayer, Anatoly Belikov, Alexander Belopolsky, and others. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688* (2016).

[2] David L Chen and William B Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *ACL*.

[3] Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *NIPS*.

[4] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. 2015. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In *NIPS*.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NIPS*.

[6] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2013. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*.

[7] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural Computation* (1997).

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

[9] G Karol, I Danihelka, A Graves, D Rezende, and D Wierstra. 2015. DRAW: a recurrent neural network for image generation. In *ICML*.

[10] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. In *ICLR*.

[11] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* (1998).

[12] Elman Mansimov, Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. 2016. Generating images from captions with attention. In *ICLR*.

[13] Gaurav Mittal, Tanya Marwah, and Vineeth Balasubramanian. 2016. Sync-DRAW: Automatic GIF Generation using Deep Recurrent Attentive Architectures. *arXiv preprint arXiv:1611.10314* (2016).

[14] Hossein Mobahi, Ronan Collobert, and Jason Weston. 2009. Deep learning from temporal coherence in video. In *ICML*.

[15] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2016. Conditional Image Synthesis With Auxiliary Classifier GANs. *arXiv preprint arXiv:1610.09585* (2016).

[16] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. 2016. Jointly modeling embedding and translation to bridge video and language. In *CVPR*.

[17] Yingwei Pan, Ting Yao, Houqiang Li, and Tao Mei. 2017. Video Captioning with Transferred Semantic Attributes. In *CVPR*.

[18] Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*.

[19] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. Generative adversarial text to image synthesis. In *ICML*.

[20] Masaki Saito and Eiichi Matsumoto. 2016. Temporal Generative Adversarial Nets. *arXiv preprint arXiv:1611.06624* (2016).

[21] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* (1997).

[22] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *NIPS*.

[23] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. 2015. Unsupervised Learning of Video Representations using LSTMs.. In *ICML*.

[24] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

[25] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Generating videos with scene dynamics. In *NIPS*.

[26] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*.

[27] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. 2017. Incorporating Copying Mechanism in Image Captioning for Learning Novel Objects. In *CVPR*.

[28] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. 2017. Boosting Image Captioning with Attributes. In *ICCV*.

[29] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. 2010. Deconvolutional networks. In *CVPR*.