PRELIMINARY VERSION: DO NOT CITE
The AAAI Digital Library will contain the published
version some time after the conference

Video Generation from Text

Yitong Li[†*], Martin Renqiang Min[‡], Dinghan Shen[†], David Carlson[†], Lawrence Carin[†]
[†] Duke University, Durham, NC, United States, 27708
[‡]NEC Laboratories America, Princeton, NJ, United States, 08540
{yitong.li, dinghan.shen, david.carlson, lcarin}@duke.edu, renqiang@nec-labs.com

Abstract

Generating videos from text has proven to be a significant challenge for existing generative models. We tackle this problem by training a conditional generative model to extract both static and dynamic information from text. This is manifested in a hybrid framework, employing a Variational Autoencoder (VAE) and a Generative Adversarial Network (GAN). The static features, called "gist," are used to sketch text-conditioned background color and object layout structure. Dynamic features are considered by transforming input text into an image filter. To obtain a large amount of data for training the deep-learning model, we develop a method to automatically create a matched text-video corpus from publicly available online videos. Experimental results show that the proposed framework generates plausible and diverse short-duration smooth videos, while accurately reflecting the input text information. It significantly outperforms baseline models that directly adapt text-to-image generation procedures to produce videos. Performance is evaluated both visually and by adapting the inception score used to evaluate image generation in GANs.

1 Introduction

Generating images from text is a well-studied topic, but generating video clips based on text has yet to be explored as extensively. Previous work on the generative relationship between text and a short video clip has focused on producing text captioning from video (Venugopalan et al., 2015; Donahue et al., 2015; Pan et al., 2016; Pu et al., 2017). However, the inverse problem of producing videos from text has more degrees of freedom, and is a challenging problem for existing methods. A key consideration in video generation is that both the broad picture and object motion must be determined by the text input. Directly adapting text-to-image generation methods empirically results in videos in which the motion is not influenced by the text.

In this work, we consider motion and background synthesis from text, which is related to video prediction. In video prediction, the goal is to learn a nonlinear transformation function between given frames to predict subsequent frames (Vondrick and Torralba, 2017) – this step is also required in video generation. However, simply predicting future frames is not enough to generate a complete video clip. Recent work on video generation has decomposed video into a static background, a mask and moving objects (Vondrick, Pirsiavash, and Torralba, 2016; Tulyakov et al., 2017). Both of the cited works use a Generative Adversarial Network (GAN) (Goodfellow et al., 2014), which has shown encouraging results on sample fidelity and diversity.

However, in contrast with these previous works on video generation, here we conditionally synthesize the motion and background features based on side information, specifically text captions. In the following, we call this procedure text-to-video generation. Text-to-video generation requires both a good conditional scheme and a good video generator. There are a number of existing models for text-to-image generation (Reed et al., 2016; Mansimov et al., 2016); unfortunately, simply replacing the image generator by a video generator provides poor performance (e.g. severe mode collapse), which we detail in our experiments. These challenges reveal that even with a well-designed neural network model, directly generating video from text is difficult.

In order to solve this problem, we breakdown the generation task into two components. First, a conditional VAE model is used to generate the "gist" of the video from the input text, where the gist is an image that gives the background color and object layout of the desired video. The content and motion of the video is then generated by conditioning on both the gist and text input. This generation procedure is designed to mimic how humans create art. Specifically, artists often draw a broad draft and then fill in the detailed information. In other words, the gist-generation step extracts static "universal" features from the text, while the video generator extracts the dynamic "detailed" information from the text.

One approach to combining the text and gist information is to simply concatenate the feature vectors from the encoded text and the gist, as was previously used in image generation (Yan et al., 2016). This method unfortunately struggles to balance the relative strength of each feature set, due to their vastly different dimensionality. Instead, our work computes a set of image filter kernels based on the input text and applies the generated filter on the gist picture to get an encoded text-gist feature vector. This combined vector better models the interaction between the text and the gist than simple concatenation. It is similar to the method used in De Brabandere et

[*]Most of this work was done when the first and third authors were summer interns at NEC Laboratories America.

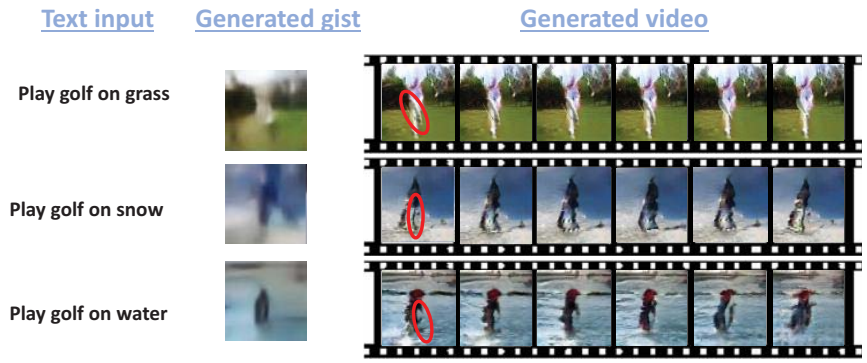


Figure 1: Samples of video generation from text. Universal background information (the gist) is produced based on the text. The text-to-filter step generates the action (e.g., "play golf"). The red circle shows the center of motion in the generated video.

al. (2016) for video prediction and image-style transformation, and Shen et al. (2017) for question answering. As we demonstrate in the experiments, the text filter better captures the motion information and adds detailed content to the gist.

Our contributions are summarized as follows: (i) By viewing the gist as an intermediate step, we propose an effective text-to-video generation framework. (ii) We demonstrate that using input text to generate a filter better models dynamic features. (iii) We propose a method to construct a training dataset based on YouTube (`www.youtube.com`) videos where the video titles and descriptions are used as the accompanying text. This allows abundant on-line video data to be used to construct robust and powerful video representations.

2 Related Work

2.1 Video Prediction and Generation

Video generation is intimately related to video prediction. Video prediction focuses on making object motion realistic in a stable background. Recurrent Neural Networks (RNNs) and the widely used sequence-to-sequence model (Sutskever, Vinyals, and Le, 2014) have shown significant promise in these applications (Villegas et al., 2017; De Brabandere et al., 2016; van Amersfoort et al., 2017; Kalchbrenner et al., 2017). A common thread among these works is that a convolutional neural network (CNN) encodes/decodes each frame and connects to a sequence-to-sequence model to predict the pixels of future frames. In addition, Liu et al. (2017) proposed deep voxel-flow networks for video-frame interpolation. Human-pose features have also been used to reduce the complexity of the generation (Villegas et al., 2017; Chao et al., 2017).

There is also significant work on video generation conditioned on a given image. Specifically, Vukotić et al. (2017); Chao et al. (2017); Walker et al. (2016); Chen et al. (2017); Xue et al. (2016) propose methods to generate videos based on static images. In these works, it is important to distinguish potential moving objects from the given image. In contrast to video prediction, these methods are useful for generating a variety of potential futures, based upon the current image. Xue et al. (2016) inspired our work by using a cross-convolutional layer. The input image is convolved with its image-dependent kernels to give predicted future frames. A similar approach has previously been used to generate future frames (De Brabandere et al., 2016). For our work, however, we do not have a matching frame for most possible text inputs. Thus, this is not feasible to feed in a first frame.

GAN frameworks have been proposed for video generation without the need for a priming image. A first attempt in this direction was made by separating scene and dynamic content (Vondrick, Pirsiavash, and Torralba, 2016). Using the GAN framework, a video could be generated purely from randomly sampled noise. Recently, Tulyakov et al. (2017) incorporated an RNN model for video generation into a GAN-based framework. This model can construct a video simply by pushing random noise into a RNN model.

2.2 Conditional Generative Networks

Two of the most popular deep generative models are the Variational Autoencoder (VAE) (Kingma and Welling, 2013) and the Generative Adversarial Network (GAN) (Goodfellow et al., 2014). A VAE is learned by maximizing the variational lower bound of the observation while encouraging the approximate (variational) posterior distribution of the hidden latent variables to be close to the prior distribution. The GAN framework relies on a minimax game between a "generator" and a "discriminator." The generator synthesizes data whereas the discriminator seeks to distinguish between real and generated data. In multi-modal situations, GAN empirically shows advantages over the VAE framework (Goodfellow et al., 2014).

In order to build relationships between text and videos, it is necessary to build conditionally generative models, which have received significant recent attention. In particular, (Mirza and Osindero, 2014) proposed a conditional GAN model for text-to-image generation. The conditional information was given to both the generator and the discriminator by concatenating a feature vector to the input and the generated image. Conditional generative models have been extended in several directions. Mansimov et al. (2016) generated images from captions with an RNN model using "attention" on the text. Liu and Tuzel (2016); Zhu et al. (2017) proposed conditional GAN models for either style or domain transfer learning. However, these methods focused on transfer from

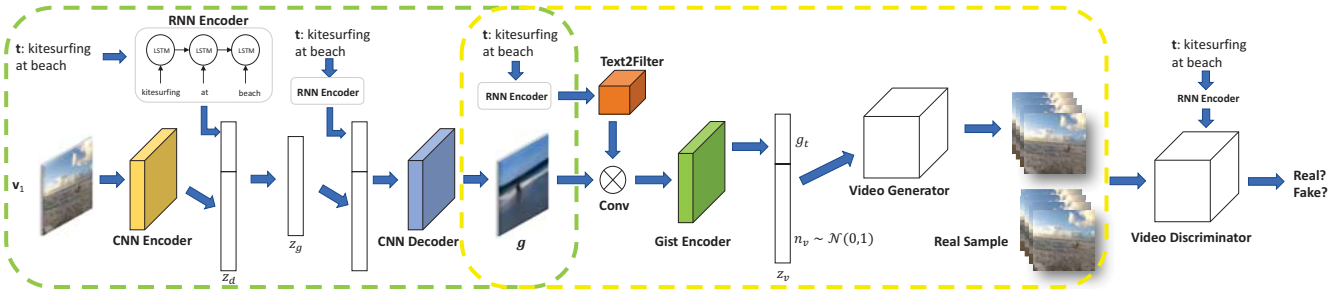


Figure 2: Framework of the proposed text-to-video generation method. The gist generator is within the green box. The encoded text is concatenated with the encoded frame to form the joint hidden representation $\boldsymbol{z}_d$, which is further transformed into $\boldsymbol{z}_g$. The video generator is within the yellow box. The text description is transformed into a filter kernel (Text2Filter) and applied to the gist. The generation uses the feature $\boldsymbol{z}_g$. Following this point, the flow chart forms a standard GAN framework with a final discriminator to judge whether a video and text pair is real or synthetic. After training, the CNN image encoder is ignored.

image to image. Converting these methods for application to text and image/video pairs is non-trivial.

The most similar work to ours is from Reed et al. (2016), which is the first successful attempt to generate natural images from text using a GAN model. In this work, pairs of data are constructed from the text features and a real or synthetic image. The discriminator tries to detect synthetic images or the mismatch between the text and the image. A direct adaptation unfortunately struggles to produce reasonable videos, as detailed in our experiments. Text-to-video generation requires a stronger conditional generator than what is necessary for text-to-image generation, due to the increased dimensionality. Video is a 4D tensor, where each frame is a 2D image with color information and spatiotemporal dependency. The increased dimensionality challenges the generator to extract both static and motion information from input text.

3 Model Description

We first introduce the components of our model, and then expand on each module in subsequent sections. The overall structure of the proposed model is given in Figure 2. There are three model components: the conditional gist generator (green box), the video generator (yellow box), and the video discriminator. The intermediate step of gist generation is developed using a conditional VAE (CVAE). Its structure is detailed in Section 3.1. The video generation is based on the scene dynamic decomposition with a GAN framework (Vondrick, Pirsiavash, and Torralba, 2016). The generation structure is detailed in Section 3.2. Because the proposed video generator is dependent on both the text and the gist, it is hard to incorporate all the information by a simple concatenation, as proposed by Reed et al. (2016). Instead, this generation is dependent on a "Text2Filter" step described in Section 3.3. Finally, the video discriminator is used to train the model in an end-to-end fashion.

The data are a collection of N videos and associated text descriptions, $\{\boldsymbol{V}_i, \boldsymbol{t}_i\}$ for $i = 1, \ldots, N$. Each video $\boldsymbol{V}_i \in \mathbb{R}^{T\times C\times H\times W}$ with frames $\boldsymbol{V}_i = \{\boldsymbol{v}_{1i}, \cdots, \boldsymbol{v}_{Ti}\}$, where C reflects the number of color bands (typically $C = 1$ or $C = 3$), and H and W are the number of pixels in the height and width dimensions, respectively, for each video frame. Note that all videos are cut to the same number of frames; this limitation can be avoided by using an RNN generator, but this is left for future work. The text description $\boldsymbol{t}$ is given as a sequence of words (natural language). The index i is only included when necessary for clarity.

The text input was processed with a standard text encoder, which can be jointly trained with the model. Empirically, the chosen encoder is a minor contributer to model performance. Thus for simplicity, we directly adopt the skip-thought vector encoding model (Kiros et al., 2015).

3.1 Gist Generator

In a short video clip, the background is usually static with only small motion changes. The gist generator uses a CVAE to produce the static background from the text (see example gists in Figure 1). Training the CVAE requires pairs of text and images; in practice, we have found that simply using the first frame of the video, $\boldsymbol{v}_1$, works well.

The CVAE is trained by maximizing the variational lower bound

$$\mathcal{L}_{CVAE}(\boldsymbol{\theta}_g, \boldsymbol{\phi}_g; \boldsymbol{v}, \boldsymbol{t}) = \mathbb{E}_{q_{\phi_g}(\boldsymbol{z}_g|\boldsymbol{v},\boldsymbol{t})}\left[\log p_{\theta_g}(\boldsymbol{v}|\boldsymbol{z}_g, \boldsymbol{t})\right] - \mathrm{KL}\left(q_{\phi_g}(\boldsymbol{z}_g|\boldsymbol{v}, \boldsymbol{t})||p(\boldsymbol{z}_g)\right). \quad (1)$$

Following the original VAE construction (Kingma and Welling, 2013), the prior $p(\boldsymbol{z}_g)$ is set as an isotropic multivariate Gaussian distribution; θ_g and ϕ_g are parameters related to the decoder and encoder network, respectively. The subscript g denotes gist. The encoder network $q_{\phi_g}(\boldsymbol{z}_g|\boldsymbol{v}, \boldsymbol{t})$ has two sub-encoder networks $\eta(\cdot)$ and $\psi(\cdot)$. $\eta(\cdot)$ is applied to the video frame $\boldsymbol{v}$ and $\psi(\cdot)$ is applied to the text input $\boldsymbol{t}$. A linear-combination layer is used on top of the encoder to combine the encoded video frame and text. Thus $\boldsymbol{z}_g \sim \mathcal{N}\left(\mu_{\phi_g}[\eta(\boldsymbol{v}); \psi(\boldsymbol{t})], \mathrm{diag}\left(\sigma_{\phi_g}[\eta(\boldsymbol{v}); \psi(\boldsymbol{t})]\right)\right)$. The decoding network takes $\boldsymbol{z}_g$ as an input. The output of this CVAE network is called "gist", which is then one of the inputs to the video generator.

At test time, the encoding network on the video frame is ignored, and only the encoding network $\psi(\cdot)$ on the text is applied. This step ensures the model sketches for the text-conditioned video. In our experiments, we demonstrate that

directly creating a plausible video with diversity from text is critically dependent on this intermediate generation step.

3.2 Video Generator

The video is generated by three entangled neural networks, in a GAN framework, adopting the ideas of Vondrick, Pirsiavash, and Torralba (2016). The GAN framework is trained by having a generator and a discriminator compete in a minimax game (Goodfellow et al., 2014). The generator synthesizes fake samples to confuse the discriminator, while the discriminator aims to accurately distinguish synthetic and real samples. This work utilizes the recently developed Wasserstein GAN formulation (Arjovsky, Chintala, and Bottou, 2017), given by

$$\min_{\theta_G \in \Theta_G} \max_{\theta_D} \mathbb{E}_{\boldsymbol{V} \sim p(\boldsymbol{V})} \left[D(\boldsymbol{V}; \theta_D) \right] - \mathbb{E}_{\boldsymbol{z}_v \sim p(\boldsymbol{z}_v)} \left[D(G(\boldsymbol{z}_v; \theta_G); \theta_D) \right]. \quad (2)$$

The function D discriminates between real and synthetic video-text pairs, and the parameters θ_D are limited to maintain a maximum Lipschitz constant of the function. The generator G generates synthetic samples from random noise that attempt to confuse the discriminator.

As mentioned, conditional GANs have been previously used to construct images from text (Reed et al., 2016). Because this work needs to condition on both the gist and text, it is unfortunately complicated to construct gist-text-video triplets in a similar manner. Instead, first a motion filter is computed based on the text $\boldsymbol{t}$ and applied to the gist, further described in Section 3.3. This step forces the model to use the text information to generate plausible motion; simply concatenating the feature sets allows the text information to be given minimal importance on motion generation. These feature maps are further used as input into a CNN encoder (the green cube in Figure 2), as proposed by Isola et al. (2016). The output of the encoder is denoted by the text-gist vector $\boldsymbol{g}_t$, which jointly considers the gist and text information.

To this point, there is no diversity induced for the motion in the text-gist vector, although some variation is introduced in the sampling of the gist based on the text information. The diversity of the motion and the detailed information is primarily introduced by concatenating isometric Gaussian noise $\boldsymbol{n}_v$ with the text-gist vector, to form $\boldsymbol{z}_v = [\boldsymbol{g}_t; \boldsymbol{n}_v]$. The subscript v is short for video. The random-noise vector $\boldsymbol{n}_v$ gives motion diversity to the video and synthesizes detailed information.

We use the scene dynamic decomposition (Vondrick, Pirsiavash, and Torralba, 2016). Given the vector $\boldsymbol{z}_v$, the output video from the generator is given by

$$G(\boldsymbol{z}_v) = \alpha(\boldsymbol{z}_v) \odot m(\boldsymbol{z}_v) + (1 - \alpha(\boldsymbol{z}_v)) \odot s(\boldsymbol{z}_v). \quad (3)$$

The output of $\alpha(\boldsymbol{z}_v)$ is a 4D tensor with all elements constrained in $[0, 1]$ and $\odot$ is element-wise multiplication. $\alpha(\cdot)$ and $m(\cdot)$ are both neural networks using 3D fully convolutional layers (Long, Shelhamer, and Darrell, 2015). $\alpha(\cdot)$ is a mask matrix to separate the static scene from the motion. The output of $s(\boldsymbol{z}_v)$ is a static background picture repeated through time to match the video dimensionality, where the values in $s(\cdot)$ are from an independent neural network with 2D convolutional layers. Therefore, the text-gist vector $\boldsymbol{g}_t$ and the random noise combine to create further details on the gist (the scene) and dynamic parts of the video.

The discriminator function $D(\cdot)$ in (2) is parameterized as a deep neural network with 3D convolutional layers; it has a total of five convolution and batch normalization layers. The encoded text is concatenated with the video feature on the top fully connected layer to form the conditional GAN framework.

3.3 Text2Filter

Simply concatenating the gist and text encoding empirically resulted in an overly reliant usage of either gist or text information. Tuning the length and relative strength of the features is challenging in a complex framework. Instead, a more robust and effective way to utilize the text information is to construct the motion-generating filter weights based on the text information, which is denoted by Text2Filter. This is shown as the orange cube in Figure 2.

The Text2Filter operation consists of only convolutional layers, following existing literature (Long, Shelhamer, and Darrell, 2015). We extend the 2D fully convolutional architecture to a 3D fully convolutional architecture for generating filters from text. The filter is generated from the encoded text vector by a 3D convolutional layer of size $F_c \times F_t \times kx \times ky \times kz$, where F_t is the length of the encoded text vector $\psi(\boldsymbol{t})$. F_c is number of output channels and $kx \times ky \times kz$ is filter kernel size. The 3D convolution is applied to the text vector. In our experiments, $F_c = 64$. $kx = 3$ in accordance with the RGB channels. ky and kz are set by the user, since they will become the kernel size of the gist after the 3D convolution. After this operation, the encoded text vector $\psi(\boldsymbol{t})$ of length F_t becomes a filter of size $F_c \times 3 \times ky \times kz$, which is applied on the RGB gist image $\boldsymbol{g}$. A deep network could also be adopted here if desired.

Mathematically, the text filter is represented as

$$f_g(\boldsymbol{t}) = \text{3Dconv}(\psi(\boldsymbol{t})). \quad (4)$$

Note that "3Dconv" represents the 3D full convolution operation and $\psi(\cdot)$ is the text encoder. The filter $f_g(\boldsymbol{t})$ is directly applied on the gist to give the text-gist vector

$$\boldsymbol{g}_t = \text{Encoder}\left(\text{2Dconv}\left(\boldsymbol{g}, f_g(\boldsymbol{t})\right)\right). \quad (5)$$

3.4 Objective Function, Training, and Testing

The overall objective function is manifested by the combination of $\mathcal{L}_{CVAE}$ and $\mathcal{L}_{GAN}$. Including an additional reconstruction loss $\mathcal{L}_{RECONS} = ||\boldsymbol{G} - \hat{\boldsymbol{V}}||_1$ empirically improves performance, where $\hat{\boldsymbol{V}}$ is the output of the video generator and $\boldsymbol{G}$ is T repeats of $\boldsymbol{g}$ in time dimension. The final objective function is given by

$$\mathcal{L} = \gamma_1 \mathcal{L}_{CVAE} + \gamma_2 \mathcal{L}_{GAN} + \gamma_3 \mathcal{L}_{RECONS}, \quad (6)$$

where γ_1, γ_2 and γ_3 are scalar weights for each loss term. In the experiments, $\gamma_1 = \gamma_2 = 1$ and $\gamma_3 = 10$, making the values of the three terms comparable empirically. The generator and discriminator are both updated once in each iteration. Adam (Kingma and Ba, 2014) is used as an optimizer.

When generating new videos, the video encoder before z_g in Figure 2 is discarded, and the additive noise is drawn $z_g \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. The text description and random noise are then used to generate a synthetic video.

4 Dataset Creation

Because there is no standard publicly available text-to-video generation dataset, we propose a way to download videos with matching text description. This method is similar in concept to the method in (Ye et al., 2015) that was used to create a large-scale video-classification dataset.

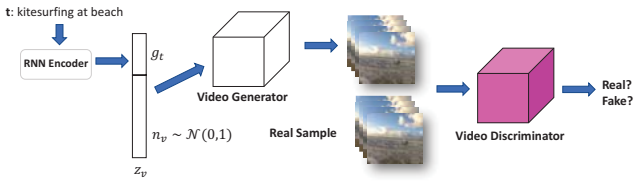
Retrieving massive numbers of videos from YouTube is easy; however, automatic curation of this dataset is not as straightforward. The data-collection process we have considered proceeds as follows. For each keyword, we first collected a set of videos together with their title, description, duration and tags from YouTube. The dataset was then cleaned by outlier-removal techniques. Specifically, the methods of (Berg, Berg, and Shih, 2010) were used to get the 10 most frequent tags for the set of video. The quality of the selected tags is further guaranteed by matching them to the words in existing categories in ImageNet (Deng et al., 2009) and ActionBank (Sadanand and Corso, 2012). These two datasets help ensure that the selected tags have visually detectable objects and actions. Only videos with at least three of the selected tags were included. Other requirements include (i) the duration of the video should be within the range of 10 to 400 seconds, (ii) the title and description should be in English, and (iii) the title should have more than four meaningful words after removing numbers and stop words.

Clean videos from the Kinetics Human Action Video Dataset (Kinetics) (Kay et al., 2017) are additionally used with the steps described above to further expand the dataset. The Kinetic dataset contains up to one thousand videos in each category, but the combined visual and text quality and consistency is mixed. For instance, some videos have non-English titles and others have bad video quality. In our experiments, we choose ten keywords as our selected categories: 'biking in snow', 'playing hockey', 'jogging', 'playing soccer ball', 'playing football', 'kite surfing', 'playing golf', 'swimming', 'sailing' and 'water skiing'. Note that the selected keywords are related to some categories in the Kinetic dataset. Most of the videos in the Kinetic dataset and the downloaded videos unfortunately have meaningless titles, such as a date indicating when the video was shot. After screening these videos, we end up with about 400 videos for each category. Using the YouTube8M (Abu-El-Haija et al., 2016) dataset for this process is also feasible, but the Kinetic dataset has cleaner videos than YouTube8M.

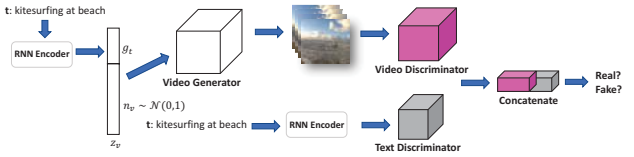
5 Experiments

5.1 Video Preprocessing

Current video-generation techniques only deal with smooth dynamic changes. A sudden change of shot or fast-changing background introduces complex non-linearities between frames, causing existing models to fail. Therefore, each video is cut and only qualified clips are used for the training (Vondrick, Pirsiavash, and Torralba, 2016). The clips were qualified as follows. Each video uses a sampling rate of 25 frames per second. SIFT key points are extracted for each frame, and the RANSAC algorithm determines whether continuous frames have enough key-point overlap (Lowe, 1999). This step ensures smooth motions in the background and objects in the used videos. Each video clip is limited to 32 frames, with 64×64 resolution. Pixel values are normalized to the range of $[-1, 1]$, matching the use of the $tanh$ function in the network output layer.



(a) Baseline with only text encoder.



(b) Baseline with pairing information.

Figure 3: Two baselines adapted from previous work. Figure 3(a) uses the conditional framework proposed by Vondrick, Pirsiavash, and Torralba (2016). The model was originally used for video prediction conditioned on a starting frame. The starting frame in the model is replaced with text description. Figure 3(b) uses a discriminator performing on the concatenation of encoded video and text vectors. This is inspired by Reed et al. (2016).

5.2 Models for Comparison

To demonstrate the effectiveness of our gist generation and conditional text filter, we compare the proposed method to several baseline models. The scene dynamic decomposition framework (Vondrick, Pirsiavash, and Torralba, 2016) is used in all the following baselines, which could be replaced with alternative frameworks. These baseline models are as follows:

- **Direct text to video generation (DT2V)**: Concatenated encoded text $\psi(\boldsymbol{t})$ and randomly sampled noise are fed into a video generator without the intermediate gist generation step. This also includes a reconstruction loss $\mathcal{L}_{RECONS}$ in (6). This is the method shown in Figure 3(a).
- **Text-to-video generation with pair information (PT2V)**: DT2V is extended using the framework of (Reed et al., 2016). The discriminator judges whether the video and text pair are real, synthetic, or a mismatched pair. This is the method in Figure 3(b). We use a linear concatenation for the video and text feature in the discriminator.
- **Text-to-video generation with gist (GT2V)**: The pro-

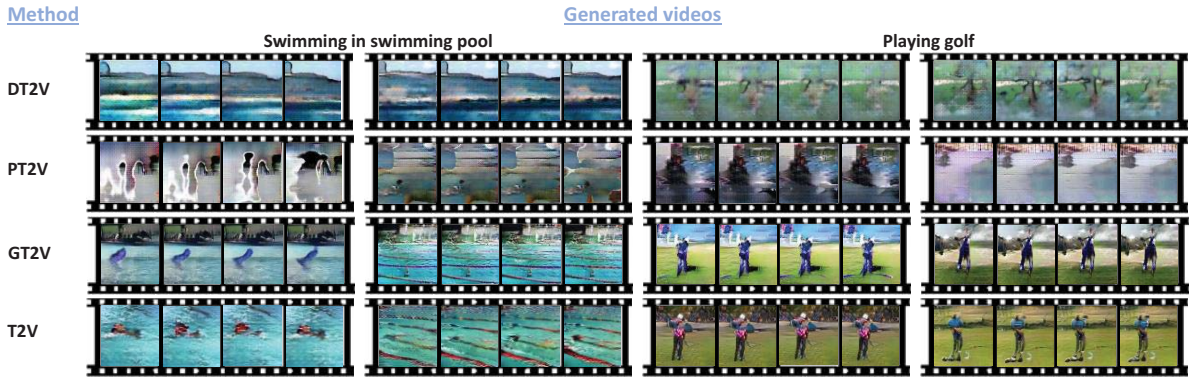


Figure 4: Comparison of generated videos with different methods. The generated movie clips are given as supplemental files (`http://www.cs.toronto.edu/pub/cuty/Text2VideoSupp`).

posed model, including only the conditional VAE for gist generation but *not* the conditional text filter (Text2Filter).

- **Video generation from text with gist and Text2Filter (T2V)** This is the complete proposed model in Section 3 with both gist generation and Text2Filter components.

Figure 4 presents samples generated by these four models, given text inputs "swimming in the swimming pool" and "playing golf". The DT2V method fails to generate plausible videos, implying that the model in Figure 3(a) does not have the ability to simultaneously represent both the static and motion features of the input. Using the "pair trick" (Reed et al., 2016; Isola et al., 2016) does not drastically alter these results. We hypothesize that because the video is a 4D tensor while the text is a 1D vector, balancing strength of each domain in the discriminator is rendered difficult. By using gist generation, GT2V gives a correct background and object layout but is deficient in motion generation. By concatenating the encoded gist vector, the encoded text vector, and the noise vector, the video generator of (3) is hard to control. Specifically, this method may completely ignore the encoded text feature when generating motion. This is further explained in Section 5.5.

In comparison, the T2V model provides both background and motion features. The intermediate gist-generation step fixes the background style and structure, and the following Text2Filter step forces the synthesized motion to use the text information. These results demonstrate the necessity of both the gist generator and the Text2Filter components in our model. In the following subsections, we intentionally generate videos that do not usually happen in real world. This is to address concerns of simply replicating videos in the training set.

5.3 Static Features

This section shows qualitative results of the gist generation, demonstrating that the gist reflects the static and background information from input text.

Figures 5(a) and 5(b) show sample gists of kite surfing at two different places. When generating videos with a grass field, the gist shows a green color. In contrast, when kite surfing on the sea, the background changes to a light blue. A black blurred shape appears in the gist in both cases, which is filled in with detail in the video generation. In Figure 5(c),

(a) Kitesurfing on the sea. (b) Kitesurfing on grass.

(c) Swimming in swimming pool. (d) Swimming in snow.

Figure 5: Input text with same motion and different background information. The input text is given as the figure caption.

Figure 6: Left is from text input "kitesurfing on the sea". Right is from text input "kitesurfing on grass"

(a) Left is "swimming at swimming pool". Right is "playing golf at swimming pool".

(b) Left is "sailing on the sea". Right is "running on the sea".

Figure 7: Same textual motion for different locations. These texts inputs show generalization, as the text in the right column does not exist in the training data.

the lanes of a swimming pool are clearly visible. In contrast, the gist for swimming in snow gives a white background. Note that for two different motions at the same location, the gists are similar (results not shown due to space).

One of the limitations of our model is the capacity of motion generation. In Figure 6, although the background color is correct, the kite-surfing motion on the grass is not consistent with reality. Additional samples can be found in Figure 1.

5.4 Motion Features

We further investigate motion-generation performance, which is shown by giving similar background and sampling the generated motion. The samples are given in Figure 7.

This figure shows that a different motion can be successfully generated with similar backgrounds. However, the greatest limitation of the current CNN video generator is its difficulty in keeping the object shape while generating a reasonable motion. Moving to specific features such as human pose or skeleton generation could provide improvements to this issue (Chao et al., 2017; Walker et al., 2017).

5.5 Quantitative Results

Following the idea of inception score (Salimans et al., 2016), we first train a classifier on six categories: 'kite surfing', 'playing golf', 'biking in snow', 'sailing', 'swimming' and 'water skiing.' Additional categories were excluded due to the low in-set accuracy of the classifier on those categories.

A relatively simple video classifier is used, which is a five-layer neural network with 3D full convolutions (Long, Shelhamer, and Darrell, 2015) and ReLU nonlinearities. The output of the network is converted to classification scores through a fully connected layer followed by a soft-max layer. In the training process, the whole video dataset is split with ratios $7:1:2$ to create training, validation and test sets. The trained classifier was used on the 20% left-out test data as well as the generated samples from the proposed and baseline models. The classification accuracy is given in Table 1.

	In-set	DT2V	PT2V	GT2V	T2V
Accuracy	0.781	0.101	0.134	0.192	0.426

Table 1: Accuracy on different test sets. 'In-set' means the test set of real videos. DT2V, PT2V, GT2V, and T2V (the full proposed model) are described in Section 5.2.

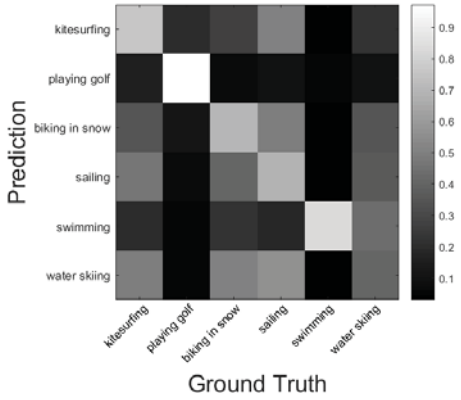


Figure 8: Classification confusion matrix on T2V generated samples.

We observe clear mode collapse when using DT2V and PT2V, explaining their poor performance. Further, it appears that directly generating video from a GAN framework fails because the video generator is not powerful enough to account for both the static and motion features from text. Using the gist generation in GT2V provides an improvement over the other baseline models. This demonstrates the usefulness of the gist, which alleviates the burden of the video generator. Notably, the full proposed model (including Text2Filter) performs best on this metric by a significant margin, showing the necessity of both the gist generation and Text2Filter.

Figure 8 shows the confusion matrix when the classifier is applied to the generated videos of our full model. Generated videos of swimming and playing golf are easier to classify than other categories. In contrast, both 'sailing' and 'kite surfing' are on the sea. Thus it is difficult to distinguish between them. This demonstrates that the gist generation step distinguishes different background style successfully.

6 Conclusion

This paper proposes a framework for generating video from text using a hybrid VAE-GAN framework. The intermediate gist-generation step greatly helps enforce the static background of video from input text. The proposed Text2Filter helps capture dynamic motion information from text. In the future, we plan to build a more powerful video generator by generating human pose or skeleton features, which will further improve the visual quality of generated human activity videos.

References

Abu-El-Haija, S.; Kothari, N.; Lee, J.; Natsev, P.; Toderici, G.; Varadarajan, B.; and Vijayanarasimhan, S. 2016. Youtube-8m: A large-scale video classification benchmark. *arXiv:1609.08675*.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein gan. *ICML*.

Berg, T. L.; Berg, A. C.; and Shih, J. 2010. Automatic attribute discovery and characterization from noisy web data. In *ECCV*.

Chao, Y.-W.; Yang, J.; Price, B.; Cohen, S.; and Deng, J. 2017. Forecasting human dynamics from static images. In *IEEE CVPR*.

Chen, B.; Wang, W.; Wang, J.; Chen, X.; and Li, W. 2017. Video imagination from a single image with transformation generation. *arXiv:1706.04124*.

De Brabandere, B.; Jia, X.; Tuytelaars, T.; and Van Gool, L. 2016. Dynamic filter networks. In *NIPS*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *IEEE CVPR*.

Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; and Darrell, T. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *IEEE CVPR*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2016. Image-to-image translation with conditional adversarial networks. *arXiv:1611.07004*.

Kalchbrenner, N.; Oord, A. v. d.; Simonyan, K.; Danihelka, I.; Vinyals, O.; Graves, A.; and Kavukcuoglu, K. 2017. Video pixel networks. *ICML*.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. 2017. The kinetics human action video dataset. *arXiv:1705.06950*.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv:1312.6114*.

Kiros, R.; Zhu, Y.; Salakhutdinov, R. R.; Zemel, R.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Skip-thought vectors. In *NIPS*.

Liu, M.-Y., and Tuzel, O. 2016. Coupled generative adversarial networks. In *NIPS*.

Liu, Z.; Yeh, R.; Tang, X.; Liu, Y.; and Agarwala, A. 2017. Video frame synthesis using deep voxel flow. *ICCV*.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

Lowe, D. G. 1999. Object recognition from local scale-invariant features. In *IEEE ICCV*, volume 2.

Mansimov, E.; Parisotto, E.; Ba, J.; and Salakhutdinov, R. 2016. Generating images from captions with attention. In *ICLR*.

Mirza, M., and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv:1411.1784*.

Pan, Y.; Mei, T.; Yao, T.; Li, H.; and Rui, Y. 2016. Jointly modeling embedding and translation to bridge video and language. In *IEEE CVPR*.

Pu, Y.; Min, M. R.; Gan, Z.; and Carin, L. 2017. Adaptive feature abstraction for translating video to language. *ICLR workshop*.

Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; and Lee, H. 2016. Generative adversarial text-to-image synthesis. In *ICML*.

Sadanand, S., and Corso, J. J. 2012. Action bank: A high-level representation of activity in video. In *IEEE CVPR*.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. In *NIPS*.

Shen, D.; Min, M. R.; Li, Y.; and Carin, L. 2017. Adaptive convolutional filter generation for natural language understanding. *arXiv preprint arXiv:1709.08294*.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Tulyakov, S.; Liu, M.-Y.; Yang, X.; and Kautz, J. 2017. Mocogan: Decomposing motion and content for video generation. *arXiv:1707.04993*.

van Amersfoort, J.; Kannan, A.; Ranzato, M.; Szlam, A.; Tran, D.; and Chintala, S. 2017. Transformation-based models of video sequences. *arXiv:1701.08435*.

Venugopalan, S.; Rohrbach, M.; Donahue, J.; Mooney, R.; Darrell, T.; and Saenko, K. 2015. Sequence to sequence-video to text. In *IEEE ICCV*.

Villegas, R.; Yang, J.; Hong, S.; Lin, X.; and Lee, H. 2017. Decomposing motion and content for natural video sequence prediction. *ICLR*.

Vondrick, C., and Torralba, A. 2017. Generating the future with adversarial transformers. In *CVPR*.

Vondrick, C.; Pirsiavash, H.; and Torralba, A. 2016. Generating videos with scene dynamics. In *NIPS*.

Vukotić, V.; Pintea, S.-L.; Raymond, C.; Gravier, G.; and Van Gemert, J. 2017. One-step time-dependent future video frame prediction with a convolutional encoder-decoder neural network. *arXiv:1702.04125*.

Walker, J.; Doersch, C.; Gupta, A.; and Hebert, M. 2016. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*.

Walker, J.; Marino, K.; Gupta, A.; and Hebert, M. 2017. The pose knows: Video forecasting by generating pose futures.

Xue, T.; Wu, J.; Bouman, K.; and Freeman, B. 2016. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *NIPS*.

Yan, X.; Yang, J.; Sohn, K.; and Lee, H. 2016. Attribute2image: Conditional image generation from visual attributes. In *ECCV*.

Ye, G.; Li, Y.; Xu, H.; Liu, D.; and Chang, S.-F. 2015. Eventnet: A large scale structured concept library for complex event detection in video. In *ACM Int. Conf. on Multimedia*.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*.