Conditional Video Generation Using Action-Appearance Captions

S. Yamamoto[1] A. Tejero-de-Pablos[1] Y. Ushiku[1] T. Harada[1,2]

[1]The University of Tokyo [2]RIKEN

Abstract

The field of automatic video generation has received a boost thanks to the recent Generative Adversarial Networks (GANs). However, most existing methods cannot control the contents of the generated video using a text caption, losing their usefulness to a large extent. This particularly affects human videos due to their great variety of actions and appearances. This paper presents Conditional Flow and Texture GAN (CFT-GAN), a GAN-based video generation method from action-appearance captions. We propose a novel way of generating video by encoding a caption (e.g., "a man in blue jeans is playing golf"*) in a two-stage generation pipeline. Our CFT-GAN uses such caption to generate an optical flow (action) and a texture (appearance) for each frame. As a result, the output video reflects the content specified in the caption in a plausible way. Moreover, to train our method, we constructed a new dataset for human video generation with captions. We evaluated the proposed method qualitatively and quantitatively via an ablation study and a user study. The results demonstrate that CFT-GAN is able to successfully generate videos containing the action and appearances indicated in the captions.*

1 Introduction

The field of multimedia content creation has experienced a remarkable evolution since the application of Generative Adversarial Networks (GANs) to image and video generation. Radford et al. [18] succeeded in generating realistic images using Deep Convolutional Generative Adversarial Networks (DCGANs). As an improvement, Wang et al. [25] proposed a GAN that takes into account structure and style. Using a two-stage architecture, they achieved more plausible images. These methods are able to generate images randomly but do not allow to specify the image content. To approach this problem, Zhang et al. [28] proposed StackGAN to generate photo-realistic images from a description sentence or *caption* (e.g., "*a man standing in a park*").

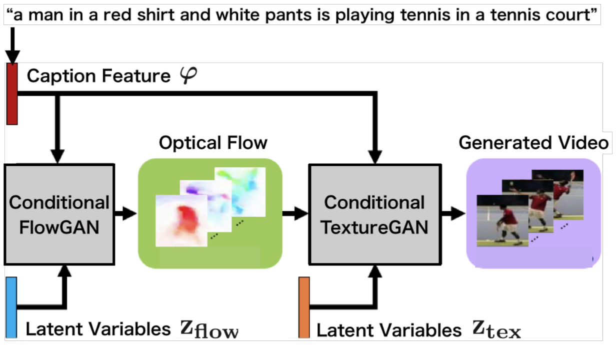

"a man in a red shirt and white pants is playing tennis in a tennis court"
Caption Feature φ
Optical Flow
Generated Video
Conditional FlowGAN
Conditional TextureGAN
Latent Variables z_{flow}
Latent Variables z_{tex}

Figure 1: Overview of the proposed method, CFT-GAN. We employ a two-stage generative architecture: Conditional FlowGAN generates optical flow from latent variables $\boldsymbol{z}_{\text{flow}}$ and features φ extracted from the input caption; Conditional TextureGAN generates video from latent variables $\boldsymbol{z}_{\text{tex}}$, features φ extracted from the input caption and the generated optical flow.

In addition to images, automatic generation of video content using GANs has also been studied. Video generation is a more difficult task, since the content between frames has to be consistent, and the motion of objects has to be plausible. This is particularly challenging in the case of human video, due to the complexity of actions and appearances. Vondrick et al. [24] proposed a scene-consistent video generation method (VGAN). The videos generated by VGAN have a consistent background, but the motion of humans is usually distorted and not realistic. To improve this, Ohnishi et al. [15] proposed a method for generation of realistic video by employing optical flow in a two-stage pipeline to improve the plausibility of actions. To control the content of the generated video, many works provide an image (the first frame) as a condition [5, 27], but few of them use captions [16, 12]. Providing captions requires almost no effort, and results are potentially more creative than using an input image. Moreover, previous methods show little variety of human actions and appearances. In order to overcome this problem, we explored the way captions are encoded into the generation pipeline, aiming for a video generator that can be controlled to reflect a variety of actions and appearances.

In this paper, we present a novel video generation method from action-appearance captions: Conditional Flow and Texture GAN (CFT-GAN). An action-appearance caption is a sentence describing a subject, the action performed, and the background (e.g., *a man in blue jeans is playing golf in a field*). We propose a way of encoding caption features as a condition to generate both the optical flow and the final video. In order for our videos to show plausible actions, we first generate an optical flow to represent the motion in the scene, as in [15]. Figure 1 shows system overview of CFT-GAN. Our method consists of two components: Conditional FlowGAN generates the motion of the scene using an action-appearance caption as a condition; Conditional TextureGAN generates the output video using the same caption and the motion generated by Conditional FlowGAN as a condition. To the best of our knowledge, this is the first GAN-based method for video generation from action-appearance captions.

We also constructed a new dataset[1] for video generation from action-appearance captions, and used it to evaluate our method. To the best of our knowledge, there is no such dataset available, so we captioned a human action video dataset for our generation purposes.

The contributions of this paper are as follows.

- We propose a novel method for automatic video generation from captions, CFT-GAN. Our way of encoding caption features into a two-stage architecture allows us to control the action and appearance of the generated video.
- We constructed a new video dataset with action-appearance captions, and used it to train our method. We also explain how to properly train the complex architecture of CFT-GAN.
- We provide an evaluation of different caption encodings via an ablation study, and a verification via a user study.

2 Related work

The field of automatic image and video generation has experienced a boost due to the emergence of Generative Adversarial Networks (GANs) [3, 18]. Unlike previous methods (i.e., Variational Auto-Encoders), GANs allow generating frames not contained in the original dataset. In image generation, Pix2Pix [8] proposed an architecture based on a U-net network [21] to convert an input image to a target image that shares the same edges. The bypass between the upper-layers and lower-layers of U-net allows the output image to reflect spatial information from the input (e.g, edges). In order to improve the realism of the generated images, Style and Structure GAN (S^{2}GAN) [25] relies on a two-stage generation method to preserve the structure of the objects. First, Structure-GAN generates the underlying 3D model; then, the 3D model is input to Style-GAN, which generates the output 2D image. Since the aforementioned works do not provide a way of controlling the generated image, methods to impose a condition in the output content have also been studied [28, 19, 14, 9, 4]. StackGAN [28] is able to generate realistic images from captions. It extracts text features from captions, and uses them as a condition for a two-stage generator. The first-stage generator generates a low resolution image from a set of latent variables and the caption features. Then, the second-stage generator generates a high resolution image using the same latent variables and caption features, and the low resolution image generated by the first-stage generator.

The task of automatic video generation has been also approached using GANs. However, video generation is more challenging, since it requires consistency between frames and motion should be plausible. This is particularly challenging in the case of human motion generation. Video GAN (VGAN) [24] achieves scene-consistent videos by generating the foreground and background separately. This method consists of 3D convolutions that

[1] The dataset will be released after publication.

learn motion information and appearance information simultaneously. However, capturing both motion and appearance using single-stream 3D convolutional networks causes generated videos to have problems with either their visual appearance or motion. Recent methods [15, 23] explore the fact that videos consist of motion and appearance. In [15], a hierarchical video generation system is proposed: Flow and Texture GAN (FTGAN). FTGAN consists of two components: FlowGAN generates the motion of the video, which is used by TextureGAN to generate videos. This method is able to successfully generate realistic video that contains plausible motion and consistent scenes.

Although in [24, 15, 23] there is no way to control the content of the videos, some other methods have attempted to condition video generation. In [5, 27, 30, 2, 6], the video is generated by providing the first frame of the sequence as the reference. These works also fall into the category of video prediction, since they require providing the initial state of the scene. While providing an image constricts the degrees of freedom of the generated video, this may be undesirable or impractical in certain creative applications.

We believe that captions can be leveraged as an alternative, more practical way of controlling the content of the generated video. To the best of our knowledge, there is comparatively few works on automatic video generation from captions [16, 12, 22]. However, whereas [16] does not tackle the challenge of generating human actions, [12] only handles simple movements such as “walking left” or “running right” in a black-and-white scene. TGAN [22] handles more complex human actions, using an action-class word (i.e., *golf*) to condition the generated video. These methods’ architecture is single-stage.

While image generation methods became able to reflect the content of an input caption, video generation has not achieved that level of control yet. We believe that, by leveraging the two-stage architecture for texture and motion, it is possible to condition, not only the type of action, but also human appearance and background. In this paper, we propose a method for automatic video generation based on the encoding caption features into the video’s motion and appearance. To the best of our knowledge, this is the first method capable of generating a video that reflects the human action and appearance specified in an input caption.

3 Preliminaries

3.1 Generative adversarial networks

Generative Adversarial Networks (GANs) [3] consist of two networks: a generator (G) and a discriminator (D). G attempts to generate data that looks similar to the given dataset. The input for G is a latent variable $\boldsymbol{z}$, which is randomly sampled from a probability distribution $p_{\boldsymbol{z}}$ (e.g., a Gaussian distribution).

D attempts to distinguish between data from the dataset (real data) and data generated from G (generated data). During training, a GAN simultaneously updates these two networks according to the following objective function V:

$$\begin{aligned}\min_G \max_D V(G,D) = \mathbb{E}_{\boldsymbol{x}\sim dataset}[\log(D(\boldsymbol{x}))] \\ + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[\log(1-D(G(\boldsymbol{z})))]\end{aligned} \tag{1}$$

where $\boldsymbol{x}$ is the data from the dataset (real data).

3.2 Single-stage GAN for video generation

Generative Adversarial Network for Video (VGAN) [24] is a network for video generation based on the concept of GANs; it consists of a generator and a discriminator. The VGAN generator comprises a mask architecture to generate separately a static background and a moving foreground from latent variables z:

$$G(\boldsymbol{z}) = m(\boldsymbol{z}) \odot f(\boldsymbol{z}) + (1 - m(\boldsymbol{z})) \odot b(\boldsymbol{z}) \tag{2}$$

where $\odot$ represents the element-wise multiplication, and $m(\boldsymbol{z})$ is a spatiotemporal matrix with values in the range of 0 to 1. For each pixel $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t})$, the mask selects whether the foreground $f(\boldsymbol{z})$ or the background $b(\boldsymbol{z})$ appears in the video. To generate a consistent background, $b(\boldsymbol{z})$ produces a spatial static image replicated over time. During training, in order to emphasize the background image, L1 regularization $\lambda\|m(\boldsymbol{z})\|_1$ for $\lambda = 0.1$ is added to the GAN objective function.

3.3 Two-stage GAN for video generation

Hierarchical video generation networks (e.g., FTGAN) [15] are based on the fact that videos consist of two elements: motion and appearance. Likewise, FTGAN

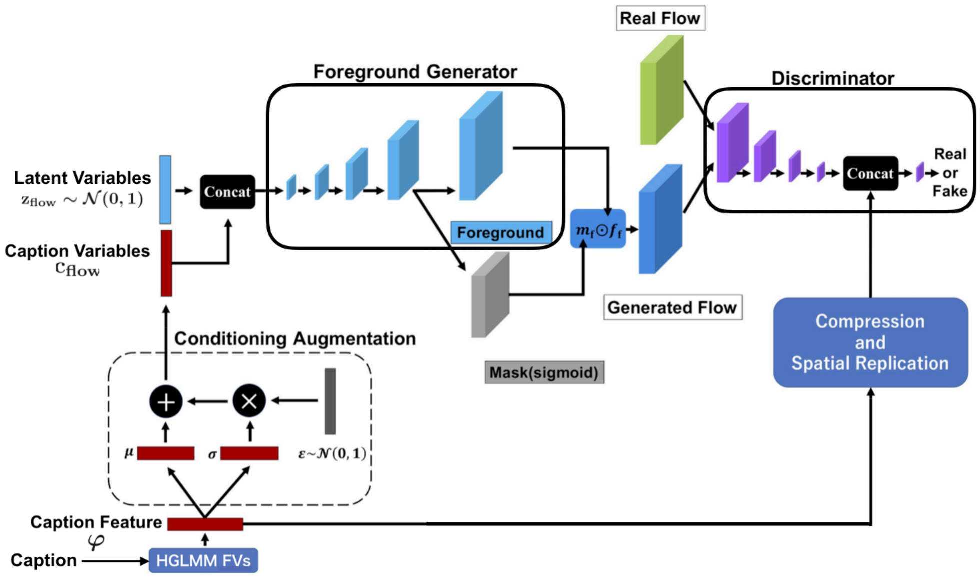


Figure 2: Architecture of Conditional FlowGAN (training): Given the latent variables $\boldsymbol{z}_{\text{flow}}$ sampled from Gaussian distributions and the caption as input, the generator learns to generate the optical flow that represents the action specified in the caption. Real and generated optical flows have a resolution of 64×64 *pixels and a duration of 32 frames.*

consists of two components: FlowGAN and TextureGAN. FlowGAN generates motion in the form of optical flow from latent variables. Then, TextureGAN generates videos from latent variables and the optical flow generated by FlowGAN.

3.3.1 FlowGAN

FlowGAN generates optical flow from latent variables $\boldsymbol{z}_{\text{flow}}$. The architecture of FlowGAN is based on VGAN [24]. VGAN is able to generate scene-consistent videos by generating the foreground and background separately. However, considering that the value of the optical flow should be zero for a static background, the FlowGAN generator does not need to learn a background stream. Instead, they make $b(\boldsymbol{z})$ a zero matrix, which is equivalent to using only the foreground stream of VGAN. Therefore, in FlowGAN, optical flow G_{flow} is generated as follows:

$$G_{\text{flow}}\left(\boldsymbol{z}_{\text{flow}}\right) = m\left(\boldsymbol{z}_{\text{flow}}\right) \odot f\left(\boldsymbol{z}_{\text{flow}}\right) \tag{3}$$

3.3.2 TextureGAN

TextureGAN takes the optical flow generated by FlowGAN and latent variables $\boldsymbol{z}_{\text{tex}}$ as input and generates the output video. The architecture of the generator is based on Pix2Pix [8] and VGAN, which generates foreground and background separately. The foreground generator is based in the U-net architecture [21], as in Pix2Pix. The bypasses between upper and lower layers in U-net allow reflecting the spatial information of the input into the output. Thus, the sharp edges of the input optical flow are reflected as the shapes of the moving objects/humans in the foreground of generated video. In TextureGAN, video G_{tex} is generated as follows:

$$\begin{aligned} G_{\text{tex}}\left(\boldsymbol{z}_{\text{tex}}, \boldsymbol{f}\right) = m\left(\boldsymbol{z}_{\text{tex}}, \boldsymbol{c}\right) \odot f\left(\boldsymbol{z}_{\text{tex}}, \boldsymbol{f}\right) \\ + \left(1 - m\left(\boldsymbol{z}_{\text{tex}}, \boldsymbol{f}\right)\right) \odot b\left(\boldsymbol{z}_{\text{tex}}\right) \end{aligned} \tag{4}$$

where $\boldsymbol{f}$ is the input optical flow. Apart from using the optical flow generated by FlowGAN in the foreground generator, the *ground truth* optical flow is used to train the discriminator.

3.4 Conditional GAN for image generation

One of the most prominent methods for including a caption as a condition to generate images is StackGAN [28]. Given an input caption (e.g., "*a gray bird with white on its chest and a very short beak*"), StackGAN extracts a feature embedding from the text and uses it, along with the latent variables, for generating the image (a two-stage generator, see Section 2). However, the limited number of training pairs of images and captions often results in sparsity in the text conditioning manifold. Such sparsity makes training a GAN difficult. To solve this problem, StackGAN introduces a conditioning augmentation technique. Instead of directly using the raw caption embedding φ as caption features, they use latent variables randomly sampled from an independent Gaussian distribution $\mathcal{N}(\mu(\varphi), \sigma(\varphi))$, where the mean $\mu(\varphi)$ and diagonal covariance matrix $\sigma(\varphi)$ are functions of the caption embedding φ. This conditioning augmentation technique smooths the distribution of caption features and makes StackGAN relatively easy to train.

4 Video generation from action-appearance captions

We propose Conditional Flow and Texture GAN (CFT-GAN), a novel method for video generation. As in [28], our method uses features extracted from an input caption as a condition for the generated content. To ensure the presence of motion and visual details in the generated video, we employ action-appearance captions, that is, captions that express an action, the appearance of the subject, and the background (e.g., "*a lady in a black dress doing sit ups in the gym*"). In order for the video to reflect the action in a plausible way, CFT-GAN separates the video generation hierarchically in two stages, as in [15]: Conditional FlowGAN generates optical flow motion based on the input caption; Conditional TextureGAN generates the output video using both the input caption and the generated optical flow as a condition.

4.1 Conditional FlowGAN

Conditional FlowGAN generates optical flow from caption features φ, and latent variables $\boldsymbol{z}_{\text{flow}}$. Figure 2 shows the architecture of Conditional FlowGAN.

First, to extract features φ from the input caption, we use Fisher Vectors (FVs) [17] based on a hybrid Gaussian-Laplacian mixture model (HGLMM) [11]. Then, as in [28], we extract caption variables $\boldsymbol{c}_{\text{flow}}$ using conditioning augmentation. That is, we calculate the mean $\mu(\varphi)$ and diagonal covariance matrix $\sigma(\varphi)$ from our caption features φ, and randomly sample latent variables $\boldsymbol{c}_{\text{flow}}$ from an independent Gaussian distribution $\mathcal{N}(\mu(\varphi), \sigma(\varphi))$.

Then, our latent variables $\boldsymbol{z}_{\text{flow}}$ are sampled from an independent Gaussian distribution, where $\mu = 0$ and $\sigma = 1$, i.e. $\mathcal{N}(0, 1)$. After that, we concatenate our caption variables $\boldsymbol{c}_{\text{flow}}$ and our latent variables $\boldsymbol{z}_{\text{flow}}$, and generate the foreground of the optical flow $\boldsymbol{f}_{\text{f}}$ and the mask of the optical flow $\boldsymbol{m}_{\text{f}}$. As in [24, 15], the mask $\boldsymbol{m}_{\text{f}}$ is a spatiotemporal matrix with each value ranging from 0 to 1; it selects either the foreground $\boldsymbol{f}_{\text{f}}$ or the background for each pixel $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t})$. According to [15], the background of optical flow should be zero if the camera is fixed, so Conditional FlowGAN does not require to learn a background generator. Instead, we use a zero matrix as background. We merge the foreground $\boldsymbol{f}_{\text{f}}$ and the background (zero matrix) based on the mask $\boldsymbol{m}_{\text{f}}$ as follows:

$$G_{\text{flow}}(\boldsymbol{z}_{\text{flow}}, \boldsymbol{c}_{\text{flow}}) = \boldsymbol{m}_{\text{f}}(\boldsymbol{z}_{\text{flow}}, \boldsymbol{c}_{\text{flow}}) \odot \boldsymbol{f}_{\text{f}}(\boldsymbol{z}_{\text{flow}}, \boldsymbol{c}_{\text{flow}}) \tag{5}$$

Finally, for the discriminator, we compressed the caption features φ using a fully-connected layer and then replicated them spatially, and concatenated the result to one of the middle layers of the discriminator, as in [28]. By doing this, the discriminator can judge not only whether the input optical flow is real or generated, but also whether the pair of input optical flow and caption are plausible.

4.2 Conditional TextureGAN

As shown in Figure 3, Conditional TextureGAN generates video from caption features, latent variables $\boldsymbol{z}_{\text{tex}}$, and the optical flow $\boldsymbol{f}$ generated by Conditional FlowGAN.

First, we extract caption features φ via HGLMM FVs as in Conditional FlowGAN, and then extract caption variables $\boldsymbol{c}_{\text{tex}}$ using conditioning augmentation. After that, we calculate the spatiotemporal matrix from caption variables $\boldsymbol{c}_{\text{tex}}$ via two up-sampling blocks.

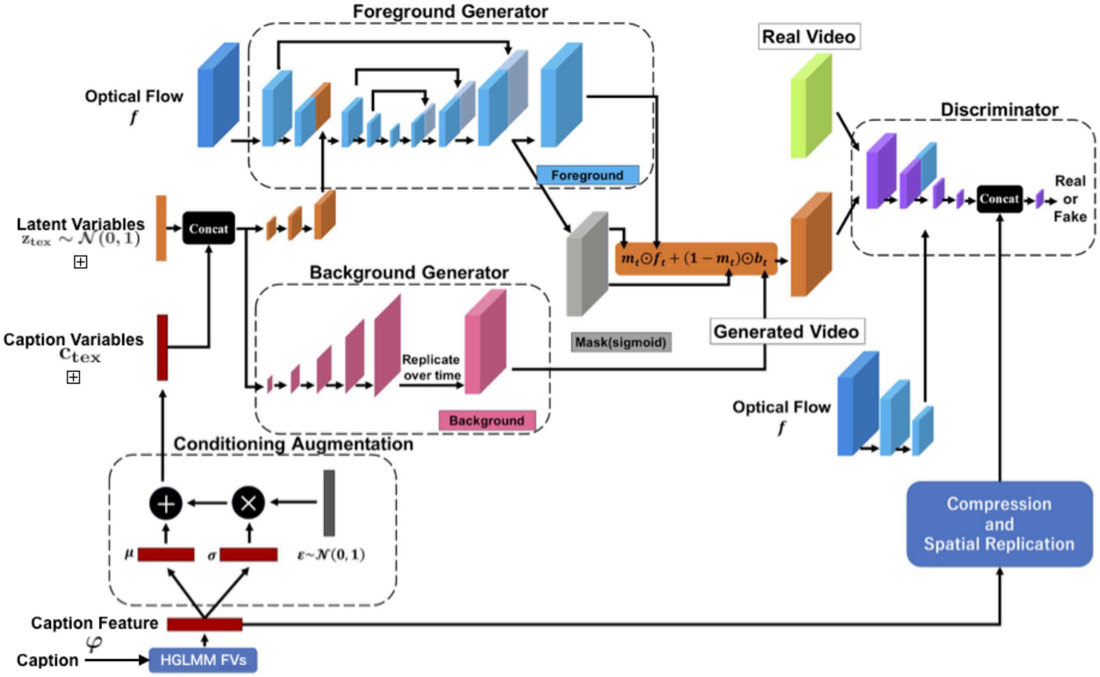


Figure 3: Architecture of Conditional TextureGAN (training): Given the optical flow f, the latent variables z_{tex} sampled from Gaussian distributions, and the caption as input, the generator learns to generate the video that represents the action and appearance specified in the caption. The input video, the input optical flow, and the output video have a resolution of 64×64 *pixels and a duration of 32 frames.*

We input the optical flow generated by FlowGAN to the foreground generator, which has a U-net structure [21]. The U-net structure contains a bypass between upper and lower layers that allows reflecting the spatial information of the input (i.e., the sharp edges of the optical flow) into the output (i.e., the foreground and mask).

We input the spatiotemporal matrix calculated from caption variables $\boldsymbol{c}_{\text{tex}}$ to the foreground generator at one of the middle layers to generate the foreground of the video $\boldsymbol{f}_{\text{t}}$ and the mask of the video $\boldsymbol{m}_{\text{t}}$. The mask $\boldsymbol{m}_{\text{t}}$ is a spatiotemporal matrix with values in the range of 0 to 1; it selects either the foreground $\boldsymbol{f}_{\text{t}}$ or the background $\boldsymbol{b}_{\text{t}}$ for each pixel (x, y, t). Simultaneously, the background generator generates the background $\boldsymbol{b}_{\text{t}}$ from the concatenation of $\boldsymbol{c}_{\text{tex}}$ and $\boldsymbol{z}_{\text{tex}}$. Finally, we merge the foreground $\boldsymbol{f}_{\text{t}}$ and the background $\boldsymbol{b}_{\text{t}}$ according to the mask $\boldsymbol{m}_{\text{t}}$ as follows:

$$\begin{aligned} &G_{\text{tex}}(\boldsymbol{z}_{\text{tex}}, \boldsymbol{c}_{\text{tex}}, \boldsymbol{f}) = \\ &\quad \boldsymbol{m}_{\text{t}}(\boldsymbol{z}_{\text{tex}}, \boldsymbol{c}_{\text{tex}}, \boldsymbol{f}) \odot \boldsymbol{f}_{\text{t}}(\boldsymbol{z}_{\text{tex}}, \boldsymbol{c}_{\text{tex}}, \boldsymbol{f}) \\ &\quad + (1 - \boldsymbol{m}_{\text{t}}(\boldsymbol{z}_{\text{tex}}, \boldsymbol{c}_{\text{tex}}, \boldsymbol{f})) \odot \boldsymbol{b}_{\text{t}}(\boldsymbol{z}_{\text{tex}}, \boldsymbol{c}_{\text{tex}}) \end{aligned} \quad (6)$$

Calculating the foreground and the background of the videos separately allows generating scene-consistent videos. Also, using motion information (optical flow) increases the plausibility of the actions contained in the generated video.

For the discriminator, we concatenated the spatiotemporal matrix extracted from the optical flow to the second layer of the discriminator. By doing this, the discriminator can judge not only whether the input video is real or generated, but also whether the pair of input video and optical flow are plausible. Also, as in Conditional FlowGAN, we compressed the caption features φ using a fully-connected layer and then replicated them spatially. Then, we concatenated them to one of the middle layers of the discriminator.

In this way, including the caption features in both Conditional FlowGAN and Conditional TextureGAN allows us to control the motion and the appearance of the video respectively.

4.3 Implementation details

To compute the HGLMM-based FVs, we trained an HGLMM with 30 centers using 300-dimensional word vectors [13] to extract text descriptors of the caption. Next, we compute the FVs of the descriptors using the learned HGLMM, and then apply principal components analysis (PCA) to reduce their size from 18000 to 256 dimensions. The size of our caption variables $\boldsymbol{c}_{\text{flow}}$ and $\boldsymbol{c}_{\text{tex}}$ is 128 dimensions, and the latent variables $\boldsymbol{z}_{\text{tex}}$ and $\boldsymbol{z}_{\text{flow}}$ are sampled from Gaussian distributions with 100 dimensions.

Conditional FlowGAN and Conditional TextureGAN contain up-sampling blocks and down-sampling blocks. Up-sampling blocks consist of the nearest-neighbor upsampling followed by 4×4 stride 2 convolutions. Conditional FlowGAN has 4 up-sampling blocks in the foreground generator. Conditional TextureGAN has 4 upsampling blocks in the foreground generator, background generator, and has 2 up-sampling blocks for concatenating $\boldsymbol{c}_{\text{tex}}$ and $\boldsymbol{z}_{\text{tex}}$ to the foreground generator. Batch normalization [7] and Rectified linear unit (ReLU) activation are applied after every up-sampling convolution, except for the last layer. The down-sampling blocks consist of 4×4 stride 2 convolutions, and we apply batch normalization [7] and LeakyReLU [26] to all layers but only apply batch normalization to the first layer. Conditional FlowGAN has 4 down-sampling blocks in the discriminator. Conditional TextureGAN has 4 down-sampling blocks in both the foreground generator and the discriminator, and 2 down-sampling blocks for extracting the spatiotemporal matrix from the input optical flow in the discriminator.

When training the networks, we use the Adam [10] optimizer with an initial learning rate $\alpha = 0.0002$ and momentum parameter $\beta_1 = 0.5$. The learning rate is decayed to $1/2$ from its previous value every 10,000 iterations during the training. We set a batch size of 32.

Although it is desirable to train Conditional FlowGAN and Conditional TextureGAN simultaneously, Conditional TextureGAN cannot be trained unless Conditional FlowGAN has been trained to some extent. For this, we use real optical flow calculated from real videos. Thus, at the beginning of the training, Conditional TextureGAN is updated mainly based on the loss obtained using real optical flow. Then, the network is updated gradually based on the loss obtained using the optical flow generated by Conditional FlowGAN. Loss functions are as follows:

$$\begin{aligned} L_{D_{\text{flow}}} &= \log D_{\text{flow}}(\boldsymbol{f}) \\ &+ \log(1 - D_{\text{flow}}(G_{\text{flow}}(\boldsymbol{z}_{\text{flow}}, \boldsymbol{c}_{\text{flow}}))) \end{aligned} \tag{7}$$

$$\begin{aligned} L_{G_{\text{flow}}} &= \log(1 - D_{\text{flow}}(G_{\text{flow}}(\boldsymbol{z}_{\text{flow}}, \boldsymbol{c}_{\text{flow}}))) \\ &+ \frac{\mathrm{k}}{\mathrm{K}} \log(1 - D_{\text{tex}}(G_{\text{tex}}(\boldsymbol{z}_{\text{tex}}, \boldsymbol{c}_{\text{tex}}, G_{\text{flow}}(\boldsymbol{z}_{\text{flow}}, \boldsymbol{c}_{\text{flow}})))) \end{aligned} \tag{8}$$

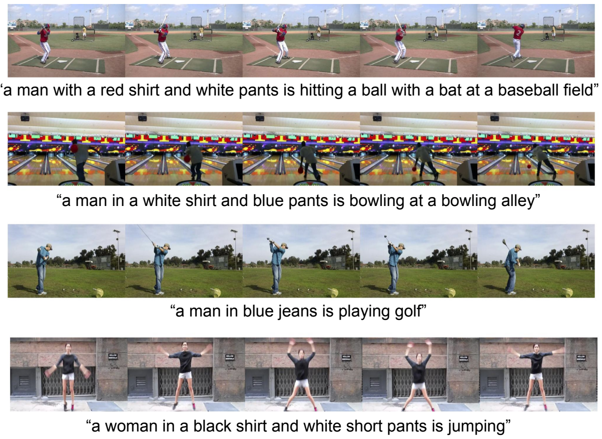
$$\begin{aligned} L_{D_{\text{tex}}} &= \log D_{\text{tex}}(\boldsymbol{x}) \\ &+ (1 - \frac{\mathrm{k}}{\mathrm{K}}) \log(1 - D_{\text{tex}}(G_{\text{tex}}(\boldsymbol{z}_{\text{tex}}, \boldsymbol{c}_{\text{tex}}, \boldsymbol{f}))) \\ &+ \frac{\mathrm{k}}{\mathrm{K}} \log(1 - D_{\text{tex}}(G_{\text{tex}}(\boldsymbol{z}_{\text{tex}}, \boldsymbol{c}_{\text{tex}}, G_{\text{flow}}(\boldsymbol{z}_{\text{flow}}, \boldsymbol{c}_{\text{flow}})))) \end{aligned} \tag{9}$$

$$\begin{aligned} L_{G_{\text{tex}}} &= (1 - \frac{\mathrm{k}}{\mathrm{K}}) \log(1 - D_{\text{tex}}(G_{\text{tex}}(\boldsymbol{z}_{\text{tex}}, \boldsymbol{c}_{\text{tex}}, \boldsymbol{f}))) \\ &+ \frac{\mathrm{k}}{\mathrm{K}} \log(1 - D_{\text{tex}}(G_{\text{tex}}(\boldsymbol{z}_{\text{tex}}, \boldsymbol{c}_{\text{tex}}, G_{\text{flow}}(\boldsymbol{z}_{\text{flow}}, \boldsymbol{c}_{\text{flow}})))) \end{aligned} \tag{10}$$

where k is the number of the current iteration, and K is the total number of iterations. We train CFT-GAN for $\mathrm{K} = 60000$ iterations.

5 Dataset and settings

We evaluated our method for the task of human video generation due to the variety of actions and appearances featured. In order to train our method for this task, a dataset containing videos and their corresponding descriptive captions are necessary. However, to the best of our knowledge, such video dataset does not exist at the moment. Thus, in this study, we constructed a new video dataset for video generation from action-appearance captions by using an existing video dataset and adding captions that describe its content. We used the Penn Action



"a man with a red shirt and white pants is hitting a ball with a bat at a baseball field"

"a man in a white shirt and blue pants is bowling at a bowling alley"

"a man in blue jeans is playing golf"

"a woman in a black shirt and white short pants is jumping"

Figure 4: Sample instances of our self-constructed dataset for video generation from action-appearance captions. We added descriptive captions to the Penn Action dataset [29] via Amazon Mechanical Turk [1].

dataset [29], which contains 2326 videos of 15 different classes, amounting to a total of 163841 frames. This dataset also contains the position of the body joints of the human shown in each frame; for our dataset, we used these positions to crop the area containing only the human. Then, we used Amazon Mechanical Turk (AMT [1]) to obtain one descriptive action-appearance caption per video in the dataset. An action-appearance caption contains an action (e.g., "*is jumping*"), the appearance of the person doing the action (e.g., "*a man in blue jeans*"), and in some cases the background (e.g., "*at a baseball field*"). Figure 4 shows some sample instances of our dataset.

We used Epic flow [20] as the ground truth optical flow to train our networks, as in [15]. We resized all frames and optical flow images to a resolution of 76 × 76 pixels, and augmented them by cropping them into 64 × 64 resolution images and randomly applying horizontal flips during training. In addition, since the length of the videos generated by our method and the baseline is 32 frames, we randomly cut 32 frames of each video for training.

6 Evaluation

We evaluated the proposed method qualitatively, via visual inspection of the results, and quantitatively, via an ablation study and a user study. We investigated the effectiveness of our encoding of action-appearance captions using different configurations of the two-stage architecture of CFT-GAN, as well as a single-stage architecture baseline, namely Conditional VGAN (CV-GAN). CV-GAN is a conditional video generation method based on VGAN (Section 2), in which a caption is encoded in the same way as CFT-GAN (feature extraction and concatenation with latent variables). Then, we used our self-constructed dataset to train our method and the baseline.

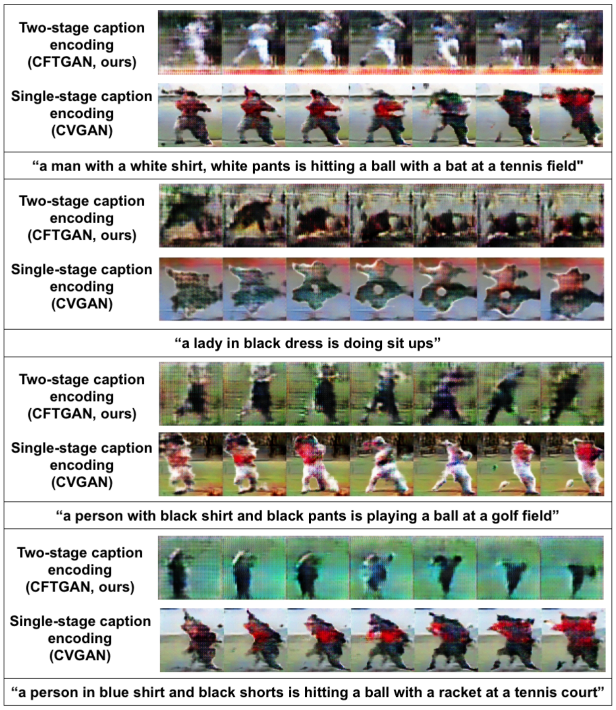

Two-stage caption encoding (CFTGAN, ours)
Single-stage caption encoding (CVGAN)
"a man with a white shirt, white pants is hitting a ball with a bat at a tennis field"
Two-stage caption encoding (CFTGAN, ours)
Single-stage caption encoding (CVGAN)
"a lady in black dress is doing sit ups"
Two-stage caption encoding (CFTGAN, ours)
Single-stage caption encoding (CVGAN)
"a person with black shirt and black pants is playing a ball at a golf field"
Two-stage caption encoding (CFTGAN, ours)
Single-stage caption encoding (CVGAN)
"a person in blue shirt and black shorts is hitting a ball with a racket at a tennis court"

Figure 5: Examples of videos generated by our method (CFT-GAN) and the baseline (CV-GAN). Each video was generated using the same caption for both methods. The generated videos have a resolution of 64×64 *pixels and a duration of 32 frames.*

6.1 Qualitative evaluation

Figure 5 shows examples of the generated videos compared to the single-stage baseline. The videos generated by CV-GAN do not reflect the content of the caption properly, that is, the appearance of the person, the action the person is executing, and the background. On the other hand, the proposed method is able to better reflect the contents of the caption. We believe this is because our Conditional TextureGAN uses the caption to generate the background and the foreground separately and, therefore, the appearance of the video can be better controlled. Also, we believe that using Conditional FlowGAN to generate the optical flow from the caption allows us to control better the action we want to reflect in the video. In spite of processing the caption in a similar manner, the baseline is not able to fully use the contents of the action-appearance caption for video generation. Visually, in the videos generated by CV-GAN, contours tend to be distorted and motion looks less plausible than the videos generated by CFT-GAN. Thus, we can infer the importance of dividing the video generation process into motion and appearance, not only for reflecting the contents of the caption, but also for improving the realism of the generated videos.

6.2 Quantitative evaluation

To evaluate objectively to what extent our method reflects the captions content, we used a distance metric between our videos and the original video with the same caption in the dataset. Since the GAN-based generated videos are, by definition, different from the original, we do not expect low distance values; instead we will focus on the difference between each configuration. The root-mean-square distance (RMSD) between two videos is defined as

$$\text{RMSD} = \sqrt{\sum_{f=1}^{F}\sum_{p=1}^{P}\frac{(p_d - p_g)^2}{P \times F}} \qquad (11)$$

where F is the number of frames in the video, P is the number of pixels in one frame, p_d is the RGB value of pixel p in the dataset frame, and p_g is the RGB value of pixel p in the generated frame. This measure is also similar to the endpoint error used to compare two optical flows. In our case, this distance increases if there are incoherences in the appearance and motion of the compared videos. Since the dataset and generated videos do not always have the same F, when comparing two videos, the longer video is subsampled uniformly to match the lower F. Similarly, the video with the higher resolution gets downscaled to the lowest P.

Firstly, we performed an ablation study in order to determine the contribution of each caption encoding to the generated video. We randomly selected fifty captions from the dataset and generated the videos using different configurations. We repeated this process five times. Table 1 summarizes the obtained distance values (averaged for all videos), and their respective frame examples can be found in Figure 6. When the encoding of caption features is omitted (i.e., replaced by zeros) in the texture generator (b), although motion is present, the generated videos do not reflect the indicated appearance. Furthermore, omitting the caption separately in the foreground (c) and background (d), leads to inconsistent human appearance and background appearance respectively. On the other hand, when the caption encoding is omitted in the flow generator (e), our architecture is not able to generate a plausible

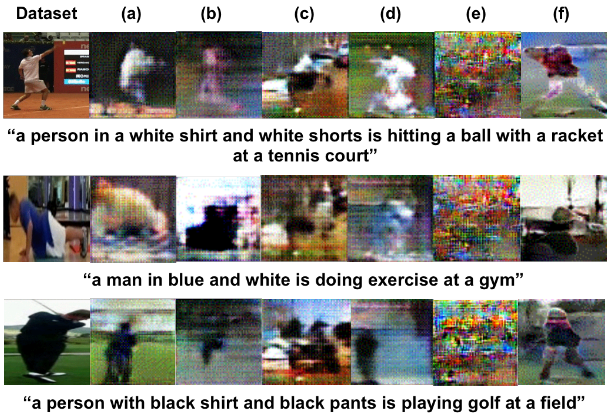

Dataset
(a)
(b)
(c)
(d)
(e)
(f)
"a person in a white shirt and white shorts is hitting a ball with a racket at a tennis court"
"a man in blue and white is doing exercise at a gym"
"a person with black shirt and black pants is playing golf at a field"

Figure 6: Ablation study. Examples of videos generated with a different way of encoding caption features. Notation of (a-f) is the same as in Table 1

Configuration	Distance
(a) Proposed (CFT-GAN)	**111.03**
(b) No caption in texture generator	193.06
(c) No caption in tex. gen. foreground only	170.26
(d) No caption in tex. gen. background only	172.15
(e) No caption in flow generator	221.15
(f) Single-stage caption encoding (CV-GAN)	150.68

Table 1: Ablation study of our method. We analyze the contribution of each caption encoding by measuring the distance (RMSD, lower is better) between the generated videos and the original videos in the dataset (averaged).

Question	Prefer CFT-GAN over CV-GAN
A	**58.88%**
B	**55.48%**

Table 2: Quantitative evaluation results of our method via AMT [1]. We show the generated videos and its corresponding captions to the AMT workers, and asked them to select between our method (CFT-GAN) and the baseline (CV-GAN) in two questions. Question A: "Which human video looks more realistic?", Question B: "Which human video looks more appropriate for the caption?"

video. We believe this is because the texture generator depends on the output of the flow generator. Finally, when trying our encoding in a single-stage generation method, the videos cannot reflect the caption successfully (see also Section 6.1). This shows the effectiveness of our caption encoding for two-stage video generation.

Lastly, we conducted a user study through Amazon Mechanical Turk [1] to compare the results between the proposed method and the baseline. As in our qualitative evaluation, we compared our method and the baseline in terms of the capability of the method to reflect the content of the input captions, and the realism of the generated videos. For this, we asked 50 unique workers to visualize 50 pairs of videos (CFT-GAN and CV-GAN) generated using same caption and answer the following two questions: (A) "Which human video looks more realistic?", (B) "Which human video looks more appropriate for the caption?". In total, we obtained 5000 opinions. The results of the survey (Table 2) show that, for both questions, more participants preferred the videos generated by our method instead of the baseline. This is coherent with the results of our ablation study, since our videos are able to reflect the content of our action-appearance captions. However, for the general user, videos are still not realistic enough, and thus, there is not a huge difference between both methods. In order to improve the realism of our method, the frame resolution could be improved by repeating the generation process taking the low resolution video as an input, similarly to StackGAN [28]. The length of our videos could be also increased, while preserving temporal coherency in motion and appearance. These would require extending our current dataset. We plan to tackle this issues in our future work.

7 Conclusions

We proposed CFT-GAN, a novel automatic video generation method, which encodes action-appearance captions into a two-stage pipeline. CFT-GAN consists of two GANs; Conditional FlowGAN and Conditional TextureGAN, to reflect a variety of actions (i.e., *doing sit ups*, *playing tennis*) and appearances (i.e., *blue shorts*, *white shirt*...) according to the caption introduced as a condition. Our experimental results demonstrate that a two-stage structure allows reflecting better the contents of the input action-appearance caption. To the best of our knowledge, this is the first GAN-based video generation method capable of controlling both the appearance and motion of human action videos using a text caption. In addition, we constructed a new video dataset for video generation from action-appearance captions, which will serve as source data for future research in this field. Our future work includes improving the resolution and duration of the generated videos to increase their realism, as well as other improvements such as video generation with moving backgrounds.

Acknowledgement

We would like to thank Takuhiro Kaneko, Takuya Nanri, Hiroaki Yamane, and Yusuke Kurose for helpful discussions. This work was partially supported by JST CREST Grant Number JPMJCR1403, Japan, and partially supported by the Ministry of Education, Culture, Sports, Science and Technology (MEXT) as "Seminal Issue on Post-K Computer."

References

[1] Amazon Mechanical Turk. https://www.mturk.com. 8, 10

[2] H. Cai, C. Bai, Y.-W. Tai, and C.-K. Tang. Deep video generation, prediction and completion of human action sequences. In *Proc. European Conference on Computer Vision*, pages 366–382, 2018. 3

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proc. Conference on Neural Information Processing Systems*, pages 2672–2680, 2014. 2, 3

[4] Z. Hao, X. Huang, and S. Belongie. AttnGAN: Fine-grained text to image generation with attentional generative adversarial networks. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 1316–1324, 2018. 2

[5] Z. Hao, X. Huang, and S. Belongie. Controllable video generation with sparse trajectories. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 7854–7863, 2018. 2, 3

[6] J. He, A. Lehrmann, J. Marino, G. Mori, and L. Sigal. Probabilistic video generation using holistic attribute control. In *Proc. European Conference on Computer Vision*, pages 466–483, 2018. 3

[7] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. International Conference on Machine Learning*, pages 448–456, 2015. 7

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2, 4

[9] T. Kaneko, K. Hiramatsu, and K. Kashino. Generative adversarial image synthesis with decision tree latent controller. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 6606–6615, 2018. 2

[10] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 7

[11] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 4437–4446, 2015. 5

[12] T. Marwah, G. Mittal, and V. N. Balasubramanian. Attentive semantic video generation using captions. In *Proc. International Conference on Computer Vision*, pages 1435–1443, 2017. 2, 3

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proc. Conference on Neural Information Processing Systems*, pages 3111–3119, 2013. 7

[14] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proc. International Conference on Machine Learning*, pages 2642–2651, 2017. 2

[15] K. Ohnishi, S. Yamamoto, Y. Ushiku, and T. Harada. Hierarchical video generation from orthogonal information: Optical flow and texture. In *Proc. AAAI Conference on Artificial Intelligence*, pages 1–9, 2018. 1, 2, 3, 5, 8

[16] Y. Pan, Z. Qiu, T. Yao, H. Li, and T. Mei. To create what you tell: Generating videos from captions. In *Proc. ACM on Multimedia Conference*, pages 1789–1798, 2017. 2, 3

[17] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proc. Conference on computer vision and pattern recognition*, pages 1–8, 2007. 5

[18] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proc. International Conference on Learning Representations*, pages 1–16, 2016. 1, 2

[19] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *Proc. International Conference on Machine Learning*, pages 1060–1069, 2016. 2

[20] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. EpicFlow: Edge-preserving interpolation of correspondences for optical flow. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 1164–1172, 2015. 8

[21] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. 2, 4, 6

[22] M. Saito, E. Matsumoto, and S. Saito. Temporal generative adversarial nets with singular value clipping. In *Proc. International Conference on Computer Vision*, pages 2830–2839, 2017. 3

[23] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz. Mocogan: Decomposing motion and content for video generation. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 1526–1535, 2018. 3

[24] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Proc. Conference on Neural Information Processing Systems*, pages 613–621, 2016. 1, 2, 3, 4, 5

[25] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. In *Proc. European Conference on Computer Vision*, pages 318–335, 2016. 1, 2

[26] B. Xu, N. Wang, T. Chen, and M. Li. Empirical evaluation of rectified activations in convolutional network. *arXiv:1505.00853*, 2015. 7

[27] C. Yang, Z. Wang, X. Zhu, C. Huang, J. Shi, and D. Lin. Pose guided human video generation. In *Proc. European Conference on Computer Vision*, pages 204–219, 2018. 2, 3

[28] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proc. International Conference on Computer Vision*, pages 5908–5916, 2017. 1, 2, 5, 10

[29] W. Zhang, M. Zhu, and K. G. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *Proc. International Conference on Computer Vision*, pages 2248–2255, 2013. 8

[30] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. Metaxas. Learning to forecast and refine residual motion for image-to-video generation. In *Proc. European Conference on Computer Vision*, pages 387–403, 2018. 3