MoCoGAN: Decomposing Motion and Content for Video Generation

Sergey Tulyakov, Snap Research
stulyakov@snap.com

Ming-Yu Liu, Xiaodong Yang, Jan Kautz
NVIDIA
{mingyul,xiaodongy,jkautz}@nvidia.com

Abstract

Visual signals in a video can be divided into content and motion. While content specifies which objects are in the video, motion describes their dynamics. Based on this prior, we propose the Motion and Content decomposed Generative Adversarial Network (MoCoGAN) framework for video generation. The proposed framework generates a video by mapping a sequence of random vectors to a sequence of video frames. Each random vector consists of a content part and a motion part. While the content part is kept fixed, the motion part is realized as a stochastic process. To learn motion and content decomposition in an unsupervised manner, we introduce a novel adversarial learning scheme utilizing both image and video discriminators. Extensive experimental results on several challenging datasets with qualitative and quantitative comparison to the state-of-the-art approaches, verify effectiveness of the proposed framework. In addition, we show that MoCoGAN allows one to generate videos with same content but different motion as well as videos with different content and same motion. Our code is available at https://github.com/sergeytulyakov/mocogan.

1. Introduction

Deep generative models have recently received an increasing amount of attention, not only because they provide a means to learn deep feature representations in an unsupervised manner that can potentially leverage all the unlabeled images on the Internet for training, but also because they can be used to generate novel images necessary for various vision applications. As steady progress toward better image generation is made, it is also important to study the video generation problem. However, the extension from generating images to generating videos turns out to be a highly challenging task, although the generated data has just one more dimension – the time dimension.

We argue video generation is much harder for the following reasons. First, since a video is a spatio-temporal recording of visual information of objects performing various actions, a generative model needs to learn the plausible physical motion models of objects in addition to learning their appearance models. If the learned object motion model is incorrect, the generated video may contain objects performing physically impossible motion. Second, the time dimension brings in a huge amount of variations. Consider the amount of speed variations that a person can have when performing a squat movement. Each speed pattern results in a different video, although the appearances of the human in the videos are the same. Third, as human beings have evolved to be sensitive to motion, motion artifacts are particularly perceptible.

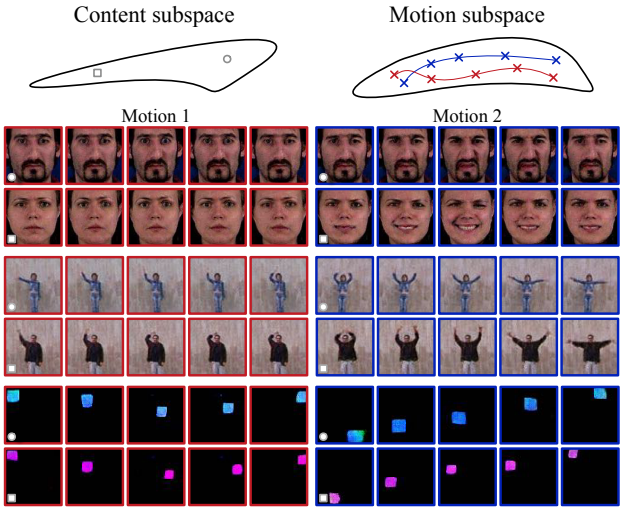

Figure 1: MoCoGAN adopts a motion and content decomposed representation for video generation. It uses an image latent space (each latent code represents an image) and divides the latent space into content and motion subspaces. By sampling a point in the content subspace and sampling different trajectories in the motion subspace, it generates videos of the same object performing different motion. By sampling different points in the content subspace and the same motion trajectory in the motion subspace, it generates videos of different objects performing the same motion.

Recently, a few attempts to approach the video generation problem were made through generative adversarial networks (GANs) [12]. Vondrick *et al*. [41] hypothesize that a video clip is a point in a latent space and proposed a VGAN

framework for learning a mapping from the latent space to video clips. A similar approach was proposed in the TGAN work [31]. We argue that assuming a video clip is a point in the latent space unnecessarily increases the complexity of the problem, because videos of the same action with different execution speed are represented by different points in the latent space. Moreover, this assumption forces every generated video clip to have the same length, while the length of real-world video clips varies. An alternative (and likely more intuitive and efficient) approach would assume a latent space of images and consider that a video clip is generated by traversing the points in the latent space. Video clips of different lengths correspond to latent space trajectories of different lengths.

In addition, as videos are about objects (content) performing actions (motion), the latent space of images should be further decomposed into two subspaces, where the deviation of a point in the first subspace (the content subspace) leads content changes in a video clip and the deviation in the second subspace (the motion subspace) results in temporal motions. Through this modeling, videos of an action with different execution speeds will only result in different traversal speeds of a trajectory in the motion space. Decomposing motion and content allows a more controlled video generation process. By changing the content representation while fixing the motion trajectory, we have videos of different objects performing the same motion. By changing motion trajectories while fixing the content representation, we have videos of the same object performing different motion as illustrated in Fig. 1.

In this paper, we propose the Motion and Content decomposed Generative Adversarial Network (MoCoGAN) framework for video generation. It generates a video clip by sequentially generating video frames. At each time step, an image generative network maps a random vector to an image. This vector consists of two parts where the first is sampled from a content subspace and the second is sampled from a motion subspace. Since content in a short video clip usually remains the same, we model the content space using a Gaussian distribution and use the same realization to generate each frame in a video clip. Sampling from the motion space is achieved through a recurrent neural network where the network parameters are learned during training. Despite lacking supervision regarding the decomposition of motion and content in natural videos, we show that MoCoGAN can learn to disentangle these two factors through a novel adversarial training scheme. Through extensive qualitative and quantitative experimental validations with comparison to the state-of-the-art approaches including VGAN [41] and TGAN [31], as well as the future frame prediction methods including Conditional-VGAN (C-VGAN) [41] and Motion and Content Network (MCNET) [40], we verify the effectiveness of MoCoGAN.

1.1. Related Work

Video generation is not a new problem. Due to limitations in computation, data, and modeling tools, early video generation works focused on generating dynamic texture patterns [35, 42, 9, 43]. In the recent years, with the availability of GPUs, Internet videos, and deep neural networks, we are now better positioned to tackle this problem.

Various deep generative models were recently proposed for image generation including GANs [12], variational autoencoders (VAEs) [20, 29, 37], and PixelCNNs [39]. In this paper, we propose the MoCoGAN framework for video generation, which is based on GANs.

Multiple GAN-based image generation frameworks were proposed. Denton *et al*. [8] showed a Laplacian pyramid implementation. Radford *et al*. [28] used a deeper convolution network. Zhang *et al*. [45] stacked two generative networks to progressively render realistic images. Coupled GANs [22] learned to generate corresponding images in different domains, later extended to translate an image from one domain to a different domain in an unsupervised fashion [21]. InfoGAN [5] learned a more interpretable latent representation. Salimans *et al*. [32] proposed several GAN training tricks. The WGAN [3] and LSGAN [23] frameworks adopted alternative distribution distance metrics for more stable adversarial training. Roth *et al*. [30] proposed a special gradient penalty to further stabilize training. Karras *et al*. [18] used progressive growing of the discriminator and the generator to generate high resolution images. The proposed MoCoGAN framework generates a video clip by sequentially generating images using an image generator. The framework can easily leverage advances in image generation in the GAN framework for improving the quality of the generated videos. As discussed in Section 1, [41, 31] extended the GAN framework to the video generation problem by assuming a latent space of video clips where all the clips have the same length.

Recurrent neural networks for image generation were previously explored in [14, 16, 27]. Specifically, some works used recurrent mechanisms to iteratively refine a generated image. Our work is different to [14, 16, 27] in that we use the recurrent mechanism to generate motion embeddings of video frames in a video clip. The image generation is achieved through a convolutional neural network.

The future frame prediction problem studied in [34, 26, 24, 17, 10, 38, 44, 40, 7] is different to the video generation problem. In future frame prediction, the goal is to predict future frames in a video given the observed frames in the video. Previous works on future frame prediction can be roughly divided into two categories where one focuses on generating raw pixel values in future frames based on the observed ones [34, 26, 24, 17, 44, 40], while the other focuses on generating transformations for reshuffling the pixels in the previous frames to construct fu-

ture frames [10, 38]. The availability of previous frames makes future frame prediction a conditional image generation problem, which is different to the video generation problem where the input to the generative network is only a vector drawn from a latent space. We note that [40] used a convolutional LSTM [15] encoder to encode temporal differences between consecutive previous frames for extracting motion information and a convolutional encoder to extract content information from the current image. The concatenation of the motion and content information was then fed to a decoder to predict future frames.

1.2. Contributions

Our contributions are as follows:

1. We propose a novel GAN framework for unconditional video generation, mapping noise vectors to videos.
2. We show the proposed framework provides a means to control content and motion in video generation, which is absent in the existing video generation frameworks.
3. We conduct extensive experimental validation on benchmark datasets with both quantitative and subjective comparison to the state-of-the-art video generation algorithms including VGAN[41] and TGAN [31] to verify the effectiveness of the proposed algorithm.

2. Generative Adversarial Networks

GANs [12] consist of a generator and a discriminator. The objective of the generator is to generate images resembling real images, while the objective of the discriminator is to distinguish real images from generated ones.

Let $\mathbf{x}$ be a real image drawn from an image distribution, p_X, and $\mathbf{z}$ be a random vector in $Z_{\mathrm{I}} \equiv \mathbb{R}^d$. Let G_{I} and D_{I} be the image generator and the image discriminator. The generator takes $\mathbf{z}$ as input and outputs an image, $\tilde{\mathbf{x}} = G_{\mathrm{I}}(\mathbf{z})$, that has the same support as $\mathbf{x}$. We denote the distribution of $G_{\mathrm{I}}(\mathbf{z})$ as $p_{G_{\mathrm{I}}}$. The discriminator estimates the probability that an input image is drawn from p_X. Ideally, $D_{\mathrm{I}}(\mathbf{x}) = 1$ if $\mathbf{x} \sim p_X$ and $D_{\mathrm{I}}(\tilde{\mathbf{x}}) = 0$ if $\tilde{\mathbf{x}} \sim p_{G_{\mathrm{I}}}$. Training of G_{I} and D_{I} is achieved via solving a minimax problem given by

$$\max_{G_{\mathrm{I}}} \min_{D_{\mathrm{I}}} \mathcal{F}_{\mathrm{I}}(D_{\mathrm{I}}, G_{\mathrm{I}}) \tag{1}$$

where the functional $\mathcal{F}_{\mathrm{I}}$ is given by

$$\begin{aligned} \mathcal{F}_{\mathrm{I}}(D_{\mathrm{I}}, G_{\mathrm{I}}) \;=\; & \mathbb{E}_{\mathbf{x}\sim p_X}[-\log D_{\mathrm{I}}(\mathbf{x})] + \\ & \mathbb{E}_{\mathbf{z}\sim p_{Z_{\mathrm{I}}}}[-\log(1 - D_{\mathrm{I}}(G_{\mathrm{I}}(\mathbf{z})))]. \end{aligned} \tag{2}$$

In practice, (1) is solved by alternating gradient update.

Goodfellow *et al*. [12] show that, given enough capacity to D_{I} and G_{I} and sufficient training iterations, the distribution $p_{G_{\mathrm{I}}}$ converges to p_X. As a result, from a random vector input $\mathbf{z}$, the network G_{I} can synthesize an image that resembles one drawn from the true distribution, p_X.

2.1. Extension to Fixed-length Video Generation

Recently, [41] extended the GAN framework to video generation by proposing a Video GAN (VGAN) framework. Let $\mathbf{v}^{\mathrm{L}} = [\mathbf{x}^{(1)}, ..., \mathbf{x}^{(\mathrm{L})}]$ be a video clip with L frames. The video generation in VGAN is achieved by replacing the vanilla CNN-based image generator and discriminator, G_{I} and D_{I}, with a spatio-temporal CNN-based video generator and discriminator, $G_{\mathrm{V^L}}$ and $D_{\mathrm{V^L}}$. The video generator $G_{\mathrm{V^L}}$ maps a random vector $\mathbf{z} \in Z_{\mathrm{V^L}} \equiv \mathbb{R}^d$ to a fixed-length video clip, $\tilde{\mathbf{v}}^{\mathrm{L}} = [\tilde{\mathbf{x}}^{(1)}, ..., \tilde{\mathbf{x}}^{(\mathrm{L})}] = G_{\mathrm{V^L}}(\mathbf{z})$ and the video discriminator $D_{\mathrm{V^L}}$ differentiates real video clips from generated ones. Ideally, $D_{\mathrm{V^L}}(\mathbf{v}^{\mathrm{L}}) = 1$ if $\mathbf{v}^{\mathrm{L}}$ is sampled from $p_{V^{\mathrm{L}}}$ and $D_{\mathrm{V^L}}(\tilde{\mathbf{v}}^{\mathrm{L}}) = 0$ if $\tilde{\mathbf{v}}^{\mathrm{L}}$ is sampled from the video generator distribution $p_{G_{\mathrm{V^L}}}$. The TGAN framework [31] also maps a random vector to a fixed length clip. The difference is that TGAN maps the random vector, representing a fixed-length video, to a fixed number of random vectors, representing individual frames in the video clip and uses an image generator for generation. Instead of using the vanilla GAN framework for minimizing the Jensen-Shannon divergence, the TGAN training is based on the WGAN framework [3] and minimizes the earth mover distance.

3. Motion and Content Decomposed GAN

In MoCoGAN, we assume a latent space of images $Z_{\mathrm{I}} \equiv \mathbb{R}^d$ where each point $\mathbf{z} \in Z_{\mathrm{I}}$ represents an image, and a video of K frames is represented by a path of length K in the latent space, $[\mathbf{z}^{(1)}, ..., \mathbf{z}^{(\mathrm{K})}]$. By adopting this formulation, videos of different lengths can be generated by paths of different lengths. Moreover, videos of the same action executed with different speeds can be generated by traversing the same path in the latent space with different speeds.

We further assume Z_{I} is decomposed into the content Z_{C} and motion Z_{M} subspaces: $Z_{\mathrm{I}} = Z_{\mathrm{C}} \times Z_{\mathrm{M}}$ where $Z_{\mathrm{C}} = \mathbb{R}^{d_{\mathrm{C}}}$, $Z_{\mathrm{M}} = \mathbb{R}^{d_{\mathrm{M}}}$, and $d = d_{\mathrm{C}} + d_{\mathrm{M}}$. The content subspace models motion-independent appearance in videos, while the motion subspace models motion-dependent appearance in videos. For example, in a video of a person smiling, content represents the identity of the person, while motion represents the changes of facial muscle configurations of the person. A pair of the person's identity and the facial muscle configuration represents a face image of the person. A sequence of these pairs represents a video clip of the person smiling. By swapping the look of the person with the look of another person, a video of a different person smiling is represented.

We model the content subspace using a Gaussian distribution: $\mathbf{z}_{\mathrm{C}} \sim p_{Z_{\mathrm{C}}} \equiv \mathcal{N}(\mathbf{z}|0, I_{d_{\mathrm{C}}})$ where $I_{d_{\mathrm{C}}}$ is an identity matrix of size $d_{\mathrm{C}} \times d_{\mathrm{C}}$. Based on the observation that the content remains largely the same in a short video clip, we use the same realization, $\mathbf{z}_{\mathrm{C}}$, for generating different frames in a video clip. Motion in the video clip is modeled by a

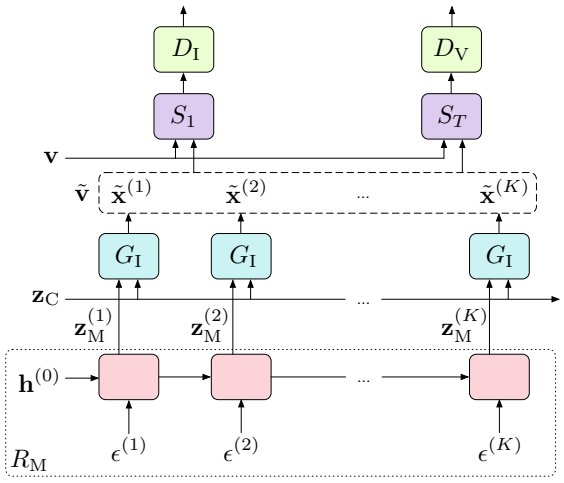

Figure 2: The MoCoGAN framework for video generation. For a video, the content vector, $\mathbf{z}_{\mathrm{C}}$, is sampled once and fixed. Then, a series of random variables $[\epsilon^{(1)}, ..., \epsilon^{(K)}]$ is sampled and mapped to a series of motion codes $[\mathbf{z}_{\mathrm{M}}^{(1)}, ..., \mathbf{z}_{\mathrm{M}}^{(K)}]$ via the recurrent neural network R_{M}. A generator G_{I} produces a frame, $\tilde{\mathbf{x}}^{(k)}$, using the content and the motion vectors $\{\mathbf{z}_{\mathrm{C}}, \mathbf{z}_{\mathrm{M}}^{(k)}\}$. The discriminators, D_{I} and D_{V}, are trained on real and fake images and videos, respectively, sampled from the training set $\mathbf{v}$ and the generated set $\tilde{\mathbf{v}}$. The function S_1 samples a single frame from a video, S_{T} samples T consequtive frames.

path in the motion subspace Z_{M}. The sequence of vectors for generating a video is represented by

$$[\mathbf{z}^{(1)}, ..., \mathbf{z}^{(K)}] = \left[\begin{bmatrix} \mathbf{z}_{\mathrm{C}} \\ \mathbf{z}_{\mathrm{M}}^{(1)} \end{bmatrix}, ..., \begin{bmatrix} \mathbf{z}_{\mathrm{C}} \\ \mathbf{z}_{\mathrm{M}}^{(K)} \end{bmatrix}\right] \tag{3}$$

where $\mathbf{z}_{\mathrm{C}} \in Z_{\mathrm{C}}$ and $\mathbf{z}_{\mathrm{M}}^{(k)} \in Z_{\mathrm{M}}$ for all k's. Since not all paths in Z_{M} correspond to physically plausible motion, we need to learn to generate valid paths. We model the path generation process using a recurrent neural network.

Let R_{M} to be a recurrent neural network. At each time step, it takes a vector sampled from a Gaussian distribution as input: $\epsilon^{(k)} \sim p_{\mathrm{E}} \equiv \mathcal{N}(\epsilon|0, I_{d_{\mathrm{E}}})$ and outputs a vector in Z_{M}, which is used as the motion representation. Let $R_{\mathrm{M}}(k)$ be the output of the recurrent neural network at time k. Then, $\mathbf{z}_{\mathrm{M}}^{(k)} = R_{\mathrm{M}}(k)$. Intuitively, the function of the recurrent neural network is to map a sequence of independent and identically distributed (i.i.d.) random variables $[\epsilon^{(1)}, ..., \epsilon^{(K)}]$ to a sequence of correlated random variables $[R_{\mathrm{M}}(1), ..., R_{\mathrm{M}}(K)]$ representing the dynamics in a video. Injecting noise at every iteration models uncertainty of the future motion at each timestep. We implement R_{M} using a one-layer GRU network [6].

Networks. MoCoGAN consists of 4 sub-networks, which are the recurrent neural network, R_{M}, the image generator, G_{I}, the image discriminator, D_{I}, and the video discriminator, D_{V}. The image generator generates a video clip by sequentially mapping vectors in Z_{I} to images, from a sequence of vectors $[\begin{bmatrix} \mathbf{z}_{\mathrm{C}} \\ \mathbf{z}_{\mathrm{M}}^{(1)} \end{bmatrix}, ..., \begin{bmatrix} \mathbf{z}_{\mathrm{C}} \\ \mathbf{z}_{\mathrm{M}}^{(K)} \end{bmatrix}]$ to a sequence of images, $\tilde{\mathbf{v}} = [\tilde{\mathbf{x}}^{(1)}, ..., \tilde{\mathbf{x}}^{(K)}]$, where $\tilde{\mathbf{x}}^{(k)} = G_{\mathrm{I}}(\begin{bmatrix} \mathbf{z}_{\mathrm{C}} \\ \mathbf{z}_{\mathrm{M}}^{(k)} \end{bmatrix})$ and $\mathbf{z}_{\mathrm{M}}^{(k)}$'s are from the recurrent neural network, R_{M}. We note that the video length K can vary for each video generation.

Both D_{I} and D_{V} play the judge role, providing criticisms to G_{I} and R_{M}. The image discriminator D_{I} is specialized in criticizing G_{I} based on individual images. It is trained to determine if a frame is sampled from a real video clip, $\mathbf{v}$, or from $\tilde{\mathbf{v}}$. On the other hand, D_{V} provides criticisms to G_{I} based on the generated video clip. D_{V} takes a fixed length video clip, say T frames, and decides if a video clip was sampled from a real video or from $\tilde{\mathrm{v}}$. Different from D_{I}, which is based on vanilla CNN architecture, D_{V} is based on a spatio-temporal CNN architecture. We note that the clip length T is a hyperparameter, which is set to 16 throughout our experiments. We also note that T can be smaller than the generated video length K. A video of length K can be divided into $K - \mathrm{T} + 1$ clips in a sliding-window fashion, and each of the clips can be fed into D_{V}.

The video discriminator D_{V} also evaluates the generated motion. Since G_{I} has no concept of motion, the criticisms on the motion part go directly to the recurrent neural network, R_{M}. In order to generate a video with realistic dynamics that fools D_{V}, R_{M} has to learn to generate a sequence of motion codes $[\mathbf{z}_{\mathrm{M}}^{(1)}, ..., \mathbf{z}_{\mathrm{M}}^{(K)}]$ from a sequence of i.i.d. noise inputs $[\epsilon^{(1)}, ..., \epsilon^{(K)}]$ in a way such that G_{I} can map $\mathbf{z}^{(k)} = [\mathbf{z}_{\mathrm{C}}, \mathbf{z}_{\mathrm{M}}^{(k)}]$ to consecutive frames in a video.

Ideally, D_{V} alone should be sufficient for training G_{I} and R_{M}, because D_{V} provides feedback on both static image appearance and video dynamics. However, we found that using D_{I} significantly improves the convergence of the adversarial training. This is because training D_{I} is simpler, as it only needs to focus on static appearances. Fig. 2 shows visual representation of the MoCoGAN framework.

Learning. Let p_{V} be the distribution of video clips of variable lengths. Let κ be a discrete random variable denoting the length of a video clip sampled from p_{V}. (In practice, we can estimate the distribution of κ, termed p_{K}, by computing a histogram of video clip length from training data). To generate a video, we first sample a content vector, $\mathbf{z}_{\mathrm{C}}$, and a length, κ. We then run R_{M} for κ steps and, at each time step, R_{M} takes a random variable ϵ as the input. A generated video is then given by

$$\tilde{\mathbf{v}} = \left[G_{\mathrm{I}}(\begin{bmatrix} z_{\mathrm{C}} \\ R_{\mathrm{M}}(1) \end{bmatrix}), ..., G_{\mathrm{I}}(\begin{bmatrix} z_{\mathrm{C}} \\ R_{\mathrm{M}}(\kappa) \end{bmatrix})\right]. \tag{4}$$

Recall that our D_{I} and D_{V} take one frame and T consecutive frames in a video as input, respectively. In order

to represent these sampling mechanisms, we introduce two random access functions S_1 and S_T. The function S_1 takes a video clip (either $\mathbf{v} \sim p_V$ or $\tilde{\mathbf{v}} \sim p_{\tilde{V}}$) and outputs a random frame from the clip, while the function S_T takes a video clip and randomly returns T consecutive frames from the clip. With this notation, the MoCoGAN learning problem is

$$\max_{G_I, R_M} \min_{D_I, D_V} \mathcal{F}_V(D_I, D_V, G_I, R_M) \tag{5}$$

where the objective function $\mathcal{F}_V(D_I, D_V, G_I, R_M)$ is

$$\mathbb{E}_{\mathbf{v}}[-\log D_I(S_1(\mathbf{v}))] + \mathbb{E}_{\tilde{\mathbf{v}}}[-\log(1 - D_I(S_1(\tilde{\mathbf{v}})))] + \mathbb{E}_{\mathbf{v}}[-\log D_V(S_T(\mathbf{v}))] + \mathbb{E}_{\tilde{\mathbf{v}}}[-\log(1 - D_V(S_T(\tilde{\mathbf{v}})))], \tag{6}$$

where $\mathbb{E}_{\mathbf{v}}$ is a shorthand for $\mathbb{E}_{\mathbf{v} \sim p_V}$, and $\mathbb{E}_{\tilde{\mathbf{v}}}$ for $\mathbb{E}_{\tilde{\mathbf{v}} \sim p_{\tilde{V}}}$. In (6), the first and second terms encourage D_I to output 1 for a video frame from a real video clip $\mathbf{v}$ and 0 for a video frame from a generated one $\tilde{\mathbf{v}}$. Similarly, the third and fourth terms encourage D_V to output 1 for T consecutive frames in a real video clip $\mathbf{v}$ and 0 for T consecutive frames in a generated one $\tilde{\mathbf{v}}$. The second and fourth terms encourage the image generator and the recurrent neural network to produce realistic images and video sequences of T-consecutive frames, such that no discriminator can distinguish them from real images and videos.

We train MoCoGAN using the alternating gradient update algorithm as in [11]. Specifically, in one step, we update D_I and D_V while fixing G_I and R_M. In the alternating step, we update G_I and R_M while fixing D_I and D_V.

3.1. Categorical Dynamics

Dynamics in videos are often categorical (e.g., discrete action categories: walking, running, jumping, etc.). To model this categorical signal, we augment the input to R_M with a categorical random variable, $\mathbf{z}_A$, where each realization is a one-hot vector. We keep the realization fixed since the action category in a short video remains the same. The input to R_M is then given by

$$\left[\begin{bmatrix} \mathbf{z}_A \\ \epsilon^{(1)} \end{bmatrix}, \ldots, \begin{bmatrix} \mathbf{z}_A \\ \epsilon^{(K)} \end{bmatrix} \right], \tag{7}$$

To relate $\mathbf{z}_A$ to the true action category, we adopt the InfoGAN learning [5] and augment the objective function in (6) to $\mathcal{F}_V(D_I, D_V, G_I, R_M) + \lambda L_I(G_I, Q)$ where L_I is a lower bound of the mutual information between the generated video clip and $\mathbf{z}_A$, λ is a hyperparameter, and the auxiliary distribution Q (which approximates the distribution of the action category variable conditioning on the video clip) is implemented by adding a softmax layer to the last feature layer of D_V. We use $\lambda = 1$. We note that when the labeled training data are available, we can train Q to output the category label for a real input video clip to further improve the performance [25].

4. Experiments

We conducted extensive experimental validation to evaluate MoCoGAN. In addition to comparing to VGAN [41] and TGAN [31], both quantitatively and qualitatively, we evaluated the ability of MoCoGAN to generate 1) videos of the same object performing different motions by using a fixed content vector and varying motion trajectories and 2) videos of different objects performing the same motion by using different content vectors and the same motion trajectory. We then compared a variant of the MoCoGAN framework with state-of-the-art next frame prediction methods: VGAN and MCNET [40]. Evaluating generative models is known to be a challenging task [36]. Hence, we report experimental results on several datasets, where we can obtain reliable performance metrics:

- **Shape motion.** The dataset contained two types of shapes (circles and squares) with varying sizes and colors, performing two types of motion: one moving from left to right, and the other moving from top to bottom. The motion trajectories were sampled from Bezier curves. There were $4,000$ videos in the dataset; the image resolution was 64×64 and video length was 16.
- **Facial expression.** We used the MUG Facial Expression Database [1] for this experiment. The dataset consisted of 86 subjects. Each video consisted of 50 to 160 frames. We cropped the face regions and scaled to 96×96. We discarded videos containing fewer than 64 frames and used only the sequences representing one of the six facial expressions: anger, fear, disgust, happiness, sadness, and surprise. In total, we trained on $1,254$ videos.
- **Tai-Chi.** We downloaded $4,500$ Tai Chi video clips from YouTube. For each clip, we applied a human pose estimator [4] and cropped the clip so that the performer is in the center. Videos were scaled to 64×64 pixels.
- **Human actions.** We used the Weizmann Action database [13], containing 81 videos of 9 people performing 9 actions, including jumping-jack and waving-hands. We scaled the videos to 96×96. Due to the small size, we did not conduct a quantitative evaluation using the dataset. Instead, we provide visual results in Fig. 1 and Fig. **??**.
- **UCF101** [33]**.** The database is commonly used for video action recognition. It includes $13,220$ videos of 101 different action categories. Similarly to the TGAN work [31], we scaled each frame to 85×64 and cropped the central 64×64 regions for learning.

Implementation. The details of the network designs are given in the supplementary materials. We used ADAM [19] for training, with a learning rate of 0.0002 and momentums of 0.5 and 0.999.

(a) Generated by MoCoGAN (b) Generated by VGAN [41] (c) Generated by TGAN [31]

Figure 3: Generated video clips used in the user study. The video clips were randomly selected.

Table 1: Video generation content consistency comparison. A smaller ACD means the generated frames in a video are perceptually more similar. We also compute the ACD for the training set, which is the reference.

ACD	Shape Motion	Facial Expressions
Reference	0	0.116
VGAN [41]	5.02	0.322
TGAN [31]	2.08	0.305
MoCoGAN	**1.79**	**0.201**

4.1. Video Generation Performance

Quantitative comparison. We compared MoCoGAN to VGAN and TGAN[1] using the shape motion and facial expression datasets. For each dataset, we trained a video generation model and generated 256 videos for evaluation. The VGAN and TGAN implementations can only generate fixed-length videos (32 frames and 16 frames correspondingly). For a fair comparison, we generated 16 frames using MoCoGAN, and selected every second frame from the videos generated by VGAN, such that each video has 16 frames in total.

For quantitative comparison, we measured content consistency of a generated video using the Average Content Distance (ACD) metric. For shape motion, we first computed the average color of the generated shape in each frame. Each frame was then represented by a 3-dimensional vector. The ACD is then given by the average pairwise L2 distance of the per-frame average color vectors. For facial expression videos, we employed OpenFace [2], which outperforms human performance in the face recognition task, for measuring video content consistency. OpenFace produced a feature vector for each frame in a face video. The ACD was then computed using the average pairwise L2 distance of the per-frame feature vectors.

[1]The VGAN and TGAN implementations are provided by their authors.

Figure 4: Comparison with TGAN on UCF101.

Table 2: Inception score for models trained on UCF101. All values except MoCoGAN's are taken from [31].

	VGAN	TGAN	MoCoGAN
UCF101	$8.18 \pm .05$	$11.85 \pm .07$	$\mathbf{12.42 \pm .03}$

Table 3: User preference score on video generation quality.

User preference, %	Facial Exp.	Tai-Chi
MoCoGAN / VGAN	**84.2** / 15.8	**75.4** / 24.6
MoCoGAN / TGAN	**54.7** / 45.3	**68.0** / 32.0

We computed the average ACD scores for the 256 videos generated by the competing algorithms for comparison. The results are given in Table 1. From the table, we found that the content of the videos generated by MoCoGAN was more consistent, especially for the facial expression video generation task: MoCoGAN achieved an ACD score of 0.201, which was almost 40% better than 0.322 of VGAN and 34% better than 0.305 of TGAN. Fig. 3 shows generated videos for competing algorithms.

Furthermore, we compared with TGAN and VGAN by training on the UCF101 database and computing the inception score as in Saito *et al*. [31]. Table 2 shows comparison results. In this experiment we used the same MoCoGAN model as in all other experiments. We noted that TGAN reached the inception score of 11.85 with WGAN training and Singular Value Clipping (SVC), while MoCoGAN showed a higher inception score of 12.42 without these tricks. Fig. 4 shows random samples generated by TGAN and MoCoGAN for this challenging task.

(a) Weizmann database

(b) Tai-Chi database

Figure 5: Examples of changing the motion code while fixing the content code. For (a) and (b) every row has fixed content, every column has fixed motion.

User study. We conducted a user study to quantitatively compare MoCoGAN to VGAN and TGAN using the facial expression and Tai-Chi datasets. For each algorithm, we used the trained model to randomly generate 80 videos for each task. We then randomly paired the videos generated by the MoCoGAN with the videos from one of the competing algorithms to form 80 questions. These questions were sent to the workers on Amazon Mechanical Turk (AMT) for evaluation. The videos from different algorithms were shown in random order for a fair comparison. Each question was answered by 3 different workers. The workers were instructed to choose the video that looks more realistic. Only the workers with a lifetime HIT (Human Intelligent Task) approval rate greater than 95% participated in the user study.

We report the average preference scores (the average number of times, a worker prefers an algorithm) in Table 3. From the table, we find that the workers considered the videos generated by MoCoGAN more realistic most of the times. Compared to VGAN, MoCoGAN achieved a preference score of 84.2% and 75.4% for the facial expression and Tai-Chi datasets, respectively. Compared to TGAN, MoCoGAN achieved a preference score of 54.7% and 68.0% for the facial expression and Tai-Chi datasets, respectively. In Fig. 3, we visualize the facial expression and Tai-Chi videos generated by the competing algorithms. We find that the videos generated by MoCoGAN are more realistic and contained less content and motion artifacts.

Qualitative evaluation. We conducted a qualitative experiment to demonstrate our motion and content decomposed representation. We sampled two content codes and seven motion codes, giving us 14 videos. Fig. 5 shows two examples randomly selected from this experiment. Every two rows share the same motion code while have different content codes. We observed that MoCoGAN generated the same motion sequences for two different content samples.

Table 4: Performance on categorical facial expression video generation with various MoCoGAN settings.

Settings		MCS	ACD
~~D_I~~	$\mathbf{z}_\mathrm{A} \rightarrow G_\mathrm{I}$	0.472	1.115
~~D_I~~	$\mathbf{z}_\mathrm{A} \rightarrow R_\mathrm{M}$	0.491	1.073
D_I	$\mathbf{z}_\mathrm{A} \rightarrow G_\mathrm{I}$	0.355	0.738
D_I	$\mathbf{z}_\mathrm{A} \rightarrow R_\mathrm{M}$	**0.581**	**0.606**

4.2. Categorical Video Generation

We augmented MoCoGAN with categorical variables and trained it for facial expression video generation as described in Section 3.1. The MUG dataset contains 6 different facial expressions and hence $\mathbf{z}_\mathrm{A}$ is realized as a 6 dimensional one-hot vector. We then generated 96 frames of facial expression videos. During generation, we changed the action category, $\mathbf{z}_\mathrm{A}$, every 16 frames to cover all 6 expressions. Hence, a generated video corresponded to a person performing 6 different facial expressions, one after another.

To evaluate the performance, we computed the ACD of the generated videos. A smaller ACD means the generated faces over the 96 frames were more likely to be from the same person. Note that the ACD reported in this subsection are generally larger than the ACD reported in Table 1, because the generated videos in this experiment are 6 times longer and contain 6 facial expressions versus 1. We also used the motion control score (MCS) to evaluate MoCoGAN's capability in motion generation control. To compute MCS, we first trained a spatio-temporal CNN classifier for action recognition using the labeled training dataset. During test time, we used the classifier to verify whether the generated video contained the action. The MCS is then given by testing accuracy of the classifier. A model with larger MCS offers better control over the action category.

In this experiment, we also evaluated the impact of different conditioning schemes to the categorical video generation performance. The first scheme is our default scheme where $\mathbf{z}_\mathrm{A} \rightarrow R_\mathrm{M}$. The second scheme, termed $\mathbf{z}_\mathrm{A} \rightarrow G_\mathrm{I}$, was to feed the category variable directly to the image generator. In addition, to show the impact of the image discriminative network D_I, we considered training the MoCoGAN framework without D_I.

Table 4 shows experimental results. We find that the models trained with D_I consistently yield better performances on various metrics. We also find that $\mathbf{z}_\mathrm{A} \rightarrow R_\mathrm{M}$ yields better performance. Fig. 6 shows two videos from the best model in Table 4. We observe that by fixing the content vector but changing the expression label, it generates videos of the same person performing different expressions. And similarly, by changing the content vector and providing the same motion trajectory, we generate videos

Figure 6: Generated videos of changing facial expressions. We changed the expression from smile to fear through surprise.

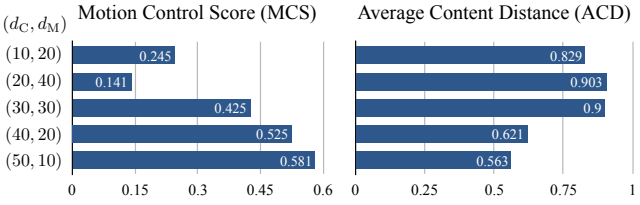

Figure 7: MoCoGAN models with varying (d_C, d_M) settings on facial expression generation.

Table 5: User preference score on the quality of the image-to-video-translation results.

User preference, %	Tai-Chi
MoCoGAN / C-VGAN	**66.9** / 33.1
MoCoGAN / MCNET	**65.6** / 34.4

of different people showing the same expression sequence.

We conducted an experiment to empirically analyze the impact of the dimensions of the content and motion vectors $\mathbf{z}_C$ and $\mathbf{z}_M^{(t)}$ (referred to as d_C and d_M) to the categorical video generation performance. In the experiment, we fixed the sum of the dimensions to 60 (i.e., $d_C + d_M = 60$) and changed the value of d_C from 10 to 50, with a step size of 10. Fig. 7 shows the results.

We found when d_C was large, MoCoGAN had a small ACD. This meant a video generated by the MoCoGAN resembled the same person performing different expressions. We were expecting a larger $\mathbf{z}_M$ would lead to a larger MCS but found the contrary. Inspecting the generated videos, we found when d_M was large (i.e. d_C was small), MoCoGAN failed to generate recognizable faces, resulting in a poor MCS. In this case, given poor image quality, the facial expression recognition network could only perform a random guess on the expression and scored poorly. Based on this, we set $d_C = 50$ and $d_M = 10$ in all the experiments.

4.3. Image-to-video Translation

We trained a variant of the MoCoGAN framework, in which the generator is realized as an encoder-decoder architecture [21], where the encoder produced the content code $\mathbf{z}_C$ and the initial motion code $\mathbf{z}_m^{(0)}$. Subsequent motion codes were produced by R_M and concatenated with the content code to generate each frame. That is the input was an image and the output was a video. We trained a MoCoGAN model using the Tai-Chi dataset. In test time, we sampled random images from a withheld test set to generate video sequences. In addition to the loss in (6), we have added the L_1 reconstruction loss for training the encoder-decoder architecture similar to Liu *et al*. [21]. Under this setting, MoCoGAN generated a video sequence starting from the first frame. We conducted a user study comparing our method with two state-of-the-art approaches: a Conditional-VGAN (C-VGAN) and Motion Content Network (MCNET) [40]. We note that MCNET used 4 frames to predict a video, while C-VGAN and MoCoGAN required a single frame only. The results are given in Table 5, which shows that the videos generated by our method were two-times more favored by the users. Fig. 8 shows that the videos generated by our method are temporally more consistent than those generated by MCNET.

Figure 8: Comparison with MCNET on image-to-video translation.

5. Conclusion

We presented the MoCoGAN framework for motion and content decomposed video generation. Given sufficient video training data, MoCoGAN automatically learns to disentangle motion from content in an unsupervised manner. For instance, given videos of people performing different facial expressions, MoCoGAN learns to separate a person's identity from their expression, thus allowing us to synthesize a new video of a person performing different expressions, or fixing the expression and generating various identities. This is enabled by a new generative adversarial network, which generates a video clip by sequentially generating video frames. Each video frame is generated from a random vector, which consists of two parts, one signifying content and one signifying motion. The content subspace is modeled with a Gaussian distribution, whereas the motion subspace is modeled with a recurrent neural network. We sample this space in order to synthesize each video frame. Our experimental evaluation supports that the proposed framework is superior to current state-of-the-art video generation and next frame prediction methods.

References

[1] N. Aifanti, C. Papachristou, and A. Delopoulos. The mug facial expression database. In *Image Analysis for Multimedia Interactive Services (WIAMIS), 2010 11th International Workshop on*, pages 1–4, 2010. 5

[2] B. Amos, B. Ludwiczuk, and M. Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016. 6

[3] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 2, 3

[4] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[5] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2, 5

[6] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Advances in Neural Information Processing Systems (NIPS) Workshop*, 2014. 4

[7] E. Denton and V. Birodkar. Unsupervised learning of disentangled representations from video. *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2

[8] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 2

[9] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision (IJCV)*, 2003. 2

[10] C. Finn, I. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2, 3

[11] I. Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016. 5

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 1, 2, 3

[13] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(12):2247–2253, 2007. 5

[14] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. In *International Conference on Machine Learning (ICML)*, 2015. 2

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 1997. 3

[16] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic. Generating images with recurrent adversarial networks. *arXiv preprint arXiv:1602.05110*, 2016. 2

[17] N. Kalchbrenner, A. v. d. Oord, K. Simonyan, I. Danihelka, O. Vinyals, A. Graves, and K. Kavukcuoglu. Video pixel networks. *arXiv preprint arXiv:1610.00527*, 2016. 2

[18] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2

[19] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 5

[20] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014. 2

[21] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2, 8

[22] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[23] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[24] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *International Conference on Learning Representations (ICLR)*, 2016. 2

[25] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier gans. In *International Conference on Machine Learning (ICML)*, 2017. 5

[26] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh. Action-conditional video prediction using deep networks in atari games. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 2

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

[28] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2016. 2

[29] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and variational inference in deep latent gaussian models. In *International Conference on Machine Learning (ICML)*, 2014. 2

[30] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2

[31] M. Saito, E. Matsumoto, and S. Saito. Temporal generative adversarial nets with singular value clipping. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 3, 5, 6

[32] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[33] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5

[34] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. In *International Conference on Machine Learning (ICML)*, 2015. 2

[35] M. Szummer and R. W. Picard. Temporal texture modeling.

In *International Conference on Image Processing (ICIP)*, 1996. 2

[36] L. Theis, A. v. d. Oord, and M. Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations (ICLR)*, 2016. 5

[37] S. Tulyakov, A. Fitzgibbon, and S. Nowozin. Hybrid vae: Improving deep generative models using partial observations. *Advances in Neural Information Processing Systems (NIPS) Workshop*, 2017. 2

[38] J. van Amersfoort, A. Kannan, M. Ranzato, A. Szlam, D. Tran, and S. Chintala. Transformation-based models of video sequences. *arXiv preprint arXiv:1701.08435*, 2017. 2, 3

[39] A. van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[40] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. Decomposing motion and content for natural video sequence prediction. In *International Conference on Learning Representations (ICLR)*, 2017. 2, 3, 5, 8

[41] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 1, 2, 3, 5, 6

[42] L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *ACM SIGGRAPH*, 2000. 2

[43] J. Xie, S.-C. Zhu, and Y. N. Wu. Synthesizing dynamic patterns by spatial-temporal generative convnet. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[44] T. Xue, J. Wu, K. Bouman, and B. Freeman. Probabilistic modeling of future frames from a single image. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[45] 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. *IEEE International Conference on Computer Vision (ICCV)*, 2016. 2