Conformal-Cycle-Consistent Adversarial Model for Video Prediction with Action Control

Zhihang Hu and **Jason T. L. Wang**
New Jersey Institute of Technology
{zh245,wangj}@njit.edu,

Abstract

The ability of predicting future frames in video sequences, known as video prediction, is an appealing yet challenging task in computer vision. This task requires an in-depth representation of video sequences and a deep understanding of real-word causal rules. Existing approaches for tackling the video prediction problem can be classified into two categories: deterministic and stochastic methods. Deterministic methods lack the ability of generating possible future frames and often yield blurry predictions. On the other hand, although current stochastic approaches can predict possible future frames, their models lack the ability of action control in the sense that they cannot generate the desired future frames conditioned on a specific action. In this paper, we propose new generative adversarial networks (GANs) for stochastic video prediction. Our framework, called VPGAN, employs an adversarial inference model and a cycle-consistency loss function to empower the framework to obtain more accurate predictions. In addition, we incorporate a conformal mapping network structure into VPGAN to enable action control for generating desirable future frames. In this way, VPGAN is able to produce fake videos of an object moving along a specific direction. Finally, our experimental results show that the combination of VPGAN with some pretrained image segmentation models outperforms existing stochastic video prediction methods.

1 Introduction

Acquiring an in-depth understanding of videos has been a cornerstone problem in computer vision. This problem has been studied by various researchers from different perspectives, among which video prediction has attracted much attention. Video prediction aims to generate the pixels of future frames given a sequence of context frames [Mathieu *et al.*, 2016]. This task finds many applications ranging from autonomous driving, robotic planning, to object tracking. In practice, unlabeled video sequences can be gathered autonomously from a sensor or recording device. A machine capable of predicting future events using these video sequences in an unsupervised manner will have gained extensive and deep knowledge about its physical environment and surroundings [Babaeizadeh *et al.*, 2018; Lee *et al.*, 2018].

However, despite its appealing prospects, accurate video prediction remains an open problem. The major challenge is the inherent uncertainty in the dynamics of the world [Denton and Fergus, 2018]. A typical example is that the future trajectory of a ball hitting the ground is inherently random. Deterministic methods [Ranzato *et al.*, 2014; Srivastava *et al.*, 2015] are unable to handle this inherent uncertainty. Furthermore, the improper loss functions adopted in many of the deterministic methods often result in blurry predictions.

With the advent of models such as generative adversarial networks (GANs) [Goodfellow *et al.*, 2014] and variational autoencoders (VAEs) [Pu *et al.*, 2016][Hu *et al.*, 2018], the quality of prediction results has been improved. Furthermore, stochastic methods based on these models are able to generate multiple future frames using some randomly sampled noise [Babaeizadeh *et al.*, 2018; Denton and Fergus, 2018; Lee *et al.*, 2018]. However, the adversarial loss functions in GANs tend to be difficult to tune, and these networks suffer from the mode collapse problem, i.e., they select a few prominent modes without being able to adequately cover the space of possible predictions [Babaeizadeh *et al.*, 2018; Lee *et al.*, 2018]. Moreover, the stochastic methods lack the understanding and control of latent-variable space, rendering action control impossible. Hence, they are unable to generate desirable future frames.

To tackle these problems, we present in this paper a new GAN-based framework, named VPGAN, for stochastic video prediction. The main contributions of our work include the following:

- We introduce a new adversarial inference model designed for stochastic video prediction and incorporate a novel cycle-consistency loss into the model to better learn actions that take place in video sequences for enhancing the quality of predicted frames.
- We incorporate a conformal mapping [Ahlfors, 1973] network structure into our VPGAN framework to enable action control for generating desirable future frames.

- We combine pretrained image segmentation models [Badrinarayanan *et al.*, 2017; Ronneberger *et al.*, 2015] with our VPGAN framework to exploit their effectivenes in image understanding. Having more semantic understanding of the frames in video sequences would enable VPGAN to generate more acccurate predictions.

To the best of our knowledge, this is the first work to incorporate action control into the video prediction task so as to generate specific future frames. In addition, the combination of our VPGAN framework with the pretrained image segmentation models outperforms existing stochastic video prediction methods as shown in our experimental results reported in the paper.

2 Related Work

While deterministic models such as LSTM based performed very poor on video prediction task due to the inherent mean squared error (MSE) loss function, Mathieu *et al.* [Mathieu *et al.*, 2016] developed a stochastic GAN-based model for video prediction. This model used an adversarial loss function instead of least absolute deviations (L_1 loss) and least square errors (L_2 loss). Tulyakov *et al.* [Tulyakov *et al.*, 2018] described a MoCoGAN framework, which decomposed a video into a content part and a motion part. They trained two GANs, one of which was a motion GAN and the other was a content GAN, together for video generation. Similar decomposition schemes have been used in [Denton and Birodkar, 2017; Villegas *et al.*, 2017]. These decomposition methods borrow the idea from background subtraction techniques [Hu *et al.*, 2018] and work well in rather simple scenarios.

Apart from GAN-based models, another popular technique for generative models is the VAE-based framework [Pu *et al.*, 2016]. This framework aims to minimize the reconstruction loss and regularization term (KL divergence between a posterior distribution and a prior). It employs a Bayesian support vector machine, permitting efficient Bayesian inference. VAE-based models have acquired great success in image generation [Pu *et al.*, 2016]. However, they suffer from the same problem when used in video prediction. Specifically, the L_2 reconstruction loss function used by these models tends to produce blurry results as it generates the expected value of all the possibilities for each pixel independently [Babaeizadeh *et al.*, 2018; Lee *et al.*, 2018]. Hence, few works have applied VAEs directly to video prediction.

The fact that GANs lack Bayesian inference while VAEs suffer from an inappropriate loss function leads to combining GANs with VAEs. State-of-the-art VAE-GAN hybrids include SVG-LP [Denton and Fergus, 2018] and SV2P [Babaeizadeh *et al.*, 2018], both of which perform stochastic video prediction. Notice that VAE-GAN hybrids are not the only type of models that incorporate inference mechanisms into GANs. There are many other efforts such as ALI [Dumoulin *et al.*, 2017] and BiGANs [Donahue *et al.*, 2017]. In particular, the adversarially learned inference (ALI) model jointly learns a generation network and an inference network using an adversarial process, and has been successfully applied to some semi-supervised learning tasks.

In this paper, we propose a new adversarial inference model in the spirit of ALI and designed specifically for video prediction, and incorporate it into our VPGAN framework. Furthermore, VPGAN employs a cycle-consistency loss function to enhance the quality of prediction results. We show experimentally that the combination of our VPGAN with pretrained image segmentation models [Badrinarayanan *et al.*, 2017; Ronneberger *et al.*, 2015] outperforms the existing VAE-GAN hybrids including SVG-LP and SV2P.

3 Method

The task of stochastic video prediction can be formalized as learning a multivalued function $f : R^{N\times M\times T} \mapsto R^{N\times M}$ from a collection of T context frames $X_0, \ldots, X_{T-1}$, each of which is a matrix of N rows and M columns of pixels, to some possible future frames $\{X_T\}$.

It is natural to think that the transformation from frame X_{t-1} to frame X_t is caused by some variation Z_t. In [Denton and Birodkar, 2017; Tulyakov *et al.*, 2018; Villegas *et al.*, 2017], the latent variable Z_t is considered as the motion of objects. However, in practice, Z_t contains not only object motion, but also variations of the physical environment and surroundings. In fact, due to adding some contraints to the latent variable, Z_t is the accumulation of multiple factors, i.e., $Z_t = Z_t^1 + Z_t^2 + \ldots + Z_t^k$. Furthermore, because the variation between frames is small as environmental changes usually don't take place in a sudden, we assume the prior distribution of Z_t is a standard Gaussian $N(\mathbf{0}, \mathbf{I})$. Based on this assumption, the video data can be described as a sequence of pairs $(X_0, Z_0), \ldots, (X_t, Z_t), 0 \leq t < T$.

3.1 Adversarial Inference

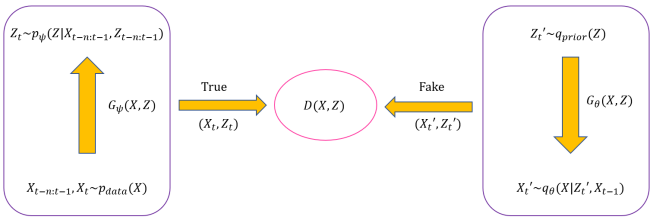

Figure 1: Illustration of the VPGAN learning process.

Let X represent the frames and let Z represent the variations under consideration. Let $p_{data}(X)$ represent the true distribution of X. We wish to construct a joint distribution $q(X, Z)$ such that $q(X, Z)$ is a good approximation of $p_{data}(X)$. In practice, it is hard to match $q(X, Z)$ with $p_{data}(X)$. On the other hand, because the video data can be described as a sequence of pairs $(X_0, Z_0), \ldots, (X_t, Z_t)$, we can consider matching $q(X, Z)$ with the joint distribution of (X_t, Z_t), denoted $p(X, Z)$. When $q(X, Z)$ and $p(X, Z)$ are matched, their marginal distributions are also matched. However, performing the matching using a traditional loss such as MSE and L_1 would result in blurry predictions. Instead, we incorporate a new adversarial inference model into our framework. By playing a min-max game between the true evidence

(X_t, Z_t) and generated fake sample (X'_t, Z'_t), we can match $q(X, Z)$ with $p(X, Z)$.

Figure 1 illustrates the VPGAN learning process during training. VPGAN employs two generators: $p_\psi = G_\psi(X, Z)$ and $q_\theta = G_\theta(X, Z)$. Let $X_{t-n:t-1}$ denote the frames $X_{t-n}, \ldots, X_{t-1}$ and let $Z_{t-n:t-1}$ denote the variations $Z_{t-n}, \ldots, Z_{t-1}$. Intuitively, past variations should have a 'momentum impact' on the present variation. Thus, we generate the variation at time t, Z_t, conditioned on the past frames $X_{t-n}, \ldots, X_{t-1}$ and past variations $Z_{t-n}, \ldots, Z_{t-1}$, i.e., $Z_t \sim p_\psi(Z|X_{t-n:t-1}, Z_{t-n:t-1})$. Variations $Z_{t-n:t-1}$ are contained in the frames $X_{t-n:t-1}$ but putting them explicitly through the input would help p_ψ focus more on the 'momentum impact'. The generator p_ψ in this case could be viewed as an encoder that encodes the past variations $Z_{t-n}, \ldots, Z_{t-1}$ into the latent variable space.

On the other hand, we generate the fake frame at time t, X'_t, conditioned on variation Z'_t sampled from a prior, $q_{prior}(Z)$, and a single past frame X_{t-1}, i.e., $X'_t \sim q_\theta(X|Z'_t, X_{t-1})$. Here, conditioning on one single past frame is reasonable as Z represents the changes between frames, and conditioning on less information would enforce Z to learn the 'true' variation efficiently. Thus, the generator q_θ serves as a decoder in our framework, which decodes the variation Z'_t and generates new frame X'_t.

The symbol $D(X, Z)$ in Figure 1 represents the discriminator, which tries to distinguish between the true evidence (X_t, Z_t) and generated fake sample (X'_t, Z'_t). VPGAN keeps on generating fake samples until the discriminator $D(X, Z)$ can not distinguish between the true evidence and generated fake sample, at which moment the training process terminates. When the training is completed, the two joint distributions $q(X, Z)$ and $p(X, Z)$ match with each other.

Denote $p_\psi(Z|X_{t-n:t-1}, Z_{t-n:t-1})$ by $G_\psi(X_{t-n:t-1}, Z_{t-n:t-1})$ and $q_\theta(X|Z'_t, X_{t-1})$ by $G_\theta(Z'_t, X_{t-1})$. The adversarial loss function used in the training is calculated as:

$$\begin{aligned} L_{adv} &= E_{X_t \sim p_{data}(X)}[\\ &\quad \log D(X_t, G_\psi(X_{t-n:t-1}, Z_{t-n:t-1}))] \\ &+ E_{Z'_t \sim q_{prior}(Z)}[\\ &\quad 1 - \log D(G_\theta(Z'_t, X_{t-1}), Z'_t)] \end{aligned} \tag{1}$$

Figure 2 illustrates the VPGAN feed-forward inference process during testing. The figure shows how to generate or predict the next frame X_t based on the past frames $X_{t-n:t-1}$. First, the past frames $X_{t-n:t-1}$ and past encoded vectors $Z_{t-n:t-1}$ are sent to the encoder p_ψ, which generates the next encoded vector (variation) Z_t. Then the decoder q_θ takes X_{t-1} and Z_t together, and predicts the next frame X_t. Depending on different variations (latent variables) Z_t, q_θ can predict multiple possible next (future) frames $\{X_T\}$.

During training and inference, we calculate p_ψ and q_θ as follows:

$$p_\psi(Z|X_{t-n:t-1}, Z_{t-n:t-1}) \sim N(\mu_\psi(X, Z), \sigma_\psi(X, Z)\mathbf{I}) \tag{2}$$

$$q_\theta(X|Z_t, X_{t-1}) \sim N(\mu_\theta(X, Z), \sigma_\theta(X, Z)\mathbf{I}) \tag{3}$$

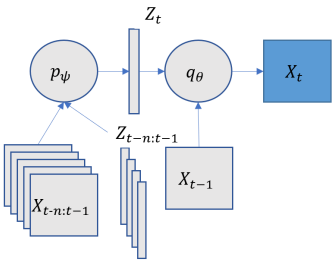

Figure 2: Illustration of the VPGAN inference process.

Based on the assumption that the prior distribution of Z is a standard Gaussian, we have

$$q_{prior}(Z) \sim N(\mathbf{0}, \mathbf{I}) \tag{4}$$

The sampling procedure used in calculating p_ψ and q_θ can be computed by the reparameterization trick [Kingma, 2013]. Specifically, instead of sampling directly from the Gaussian function with the complicated parameters, we treat the sampling procedure as a deterministic transformation of some noise such that the transformation's distribution is computable. Thus, we calculate Z_t as:

$$Z_t = \mu_\psi(X, Z) + \sigma_\psi(X, Z) \odot \xi, \;\; \xi \sim N(\mathbf{0}, \mathbf{I}) \tag{5}$$

where $\odot$ denotes the Hadamard product (element-wise product).

3.2 Cycle Consistency

Cycle consistency is based on the idea of using transitivity as a way to regularize structured data. Here we propose a new cycle consistency loss function for video prediction. With the same generator in (3), we generate the frame at time $t-1$, X_{t-1}, conditioned on the opposite of Z_t and X_t. That is, we generate $\bar{X}_{t-1}$ conditioned on $-Z_t$ and X_t where $\bar{X}_{t-1}$ is approximately equal to X_{t-1} as expressed in (6):

$$X_{t-1} \approx \bar{X}_{t-1} \sim q_\theta(X| - Z_t, X_t) \tag{6}$$

This is reminiscent of the cycle consistency loss used for image-to-image translation in [Zhu *et al.*, 2017]. However, our cycle consistency loss function is different from that in [Zhu *et al.*, 2017] because our loss function is mainly designed for video prediction rather than image translation. Since the prior Z_t follows a standard Gaussian distribution (cf. (4)), it is natural to consider the opposite variation to be the negative of Z_t. Figure 3 illustrates how cycle consistency works in our framework. As shown in the figure, we generate the current frame X_t (right) conditioned on the previous frame X_{t-1} (left) and variation Z_t. On the other hand, with the same generator, we generate the previous frame X_{t-1} conditioned on the current frame X_t and the negative of Z_t.

Mathematically, denote $q_\theta(X|Z_t, X_{t-1})$ by $G_\theta(Z_t, X_{t-1})$ and $q_\theta(X| - Z_t, X_t)$ by $G_\theta(-Z_t, X_t)$. Our cycle consistency loss is calculated as

$$\begin{aligned} L^1_{cycle} &= E_{X_t, X_{t-1} \sim p_{data}(X)}\{ \\ &\quad \| X_t - G_\theta(Z_t, G_\theta(-Z_t, X_t)) \|_1 \\ &+ \| X_{t-1} - G_\theta(-Z_t, G_\theta(Z_t, X_{t-1})) \|_1\} \end{aligned} \tag{7}$$

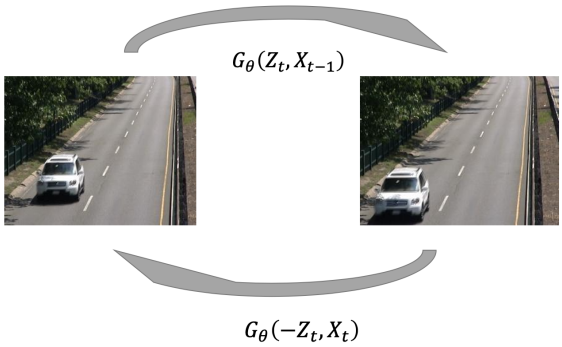

Figure 3: Illustration of cycle consistency in our framework.

Here, we utilize L_1 loss as the reconstruction loss. The loss function L_{cycle} in (7) only considers one-step cycle consistency. We can generalize the formula in (7) to take into account cycle consistency of multiple steps (more precisely, k steps) for video prediction, we first define a single-multi loss as follows (8):

$$\begin{aligned} l^k_{cycle} &= E_{X_t, X_{t-k} \sim p_{data}(X)} \{ \| X_t - G_\theta(Z_t, \\ &\quad G_\theta(Z_{t-1}, \ldots, G_\theta(-Z_t, X_t))) \|_1 \\ &+ \| X_{t-k} - G_\theta(-Z_t, \\ &\quad G_\theta(-Z_{t-1}, \ldots, G_\theta(Z_t, X_{t-k}))) \|_1 \} \end{aligned} \quad (8)$$

And our multi k step cycle-consistent loss is generalized as summing up all single-multi loss (9):

$$L^k_{cycle} = \sum_1^k a_i \cdot l^i_{cycle} \quad (9)$$

It could be very time consuming to calculate L^k_{cycle} for $k \geq 2$ since the procedure includes iteratively calculation of Generator G_θ. For instance, l^2_{cycle} would require to 8 times, and overall L^2_{cycle} would require 12 times.

When applying multi-cycle consistent loss, it is nature to take multi-reconstruction loss into consideration. Since the computational costs mainly involves in calculating L^k_{cycle}, multi-reconstruction loss only results in linear difference time. Define l^k_{recon} to be,

$$\begin{aligned} l^k_{recon} &= E_{X_t, \ldots, X_{t-k} \sim p_{data}(X)} \{ \\ &\quad \varphi(X_t, G_\theta(Z_t, ..., G_\theta(Z_{t-k+1}, X_{t-k}))) \} \end{aligned} \quad (10)$$

Here, φ stands for the reconstruction distance between the ground truth X_t and calculated $G_\theta(Z_t, ..., G_\theta(Z_{t-k+1}, X_{t-k}))$. Generally, L_2 loss is applied, but in order to obtain more acute results and reduce the impact of 'future averaging', we make φ to be the L_1 plus perceptual loss. Therefore, L^k_{recon} is naturally defined as,

$$L^k_{recon} = \sum_1^k b_i \cdot l^i_{recon} \quad (11)$$

Combing the multi-cycle loss and mult-reconstruction loss defined in this section, our overall loss, denoted as L_{loss}, is calculated as follows,

$$L_{loss} = \alpha L_{adv} + \beta L^k_{cycle} + \lambda L^k_{recon} \quad (12)$$

The perceptual loss [?] is widely applied in evaluating the reconstruction quality of images, it could be the distance on the K-th feature map, for some K, of some convolutional neural networks, such as VGG16 [Simonyan and Zisserman, 2014], ResNet [**?**] pretrained on ImageNet [**?**]. In our paper, we applied simple ResNet [**?**] to our model.

Note that, although the multi-step cycle loss enforcing long-dependency consistency likely enables more accurate action learning and predictions, its training and inference time would be approximately k times larger than that for the 1-step cycle loss, and furthermore it may suffer from gradient loss. Therefore, in our VPGAN framework, we only utilize the one-step cycle consistency loss given in (7).

3.3 Action Control

In practice, it is natural to consider the generation of desirable images or videos using GANs. Since GANs generally generate data from a random sample of the latent variable space $\mathcal{Z}$, it is hard to control the behaviour of GANs. In this subsection, we propose new techniques for generating desirable frames using GANs.

In the previous subsection, we use Z_t and $-Z_t$ to represent the opposite variations in the video space $\mathcal{H}$. Specifically, for a movement dataset, Z_t should be able to learn the moving direction of an object, and then $-Z_t$ should mainly represent the object's moving in the opposite direction. That is, from the encoding space (i.e., latent variable space) $\mathcal{Z}$ to the video space $\mathcal{H}$, we preserve what we call a 'symmetry' property, meaning that if Z_1, Z_2 are symmetric in the encoding space $\mathcal{Z}$, then the corresponding generated movements should be symmetric in the video space $\mathcal{H}$.

In addition, we wish to manipulate the latent variable space $\mathcal{Z}$ so as to generate desirable moving directions, through preserving 'orthogonality,' or more precisely, through preserving angles between the encoding space $\mathcal{Z}$ and the moving directions of an object. This orthogonality property can be preserved by first enforcing the latent variable space $\mathcal{Z}$ to be R^2. Although the moving direction of an object in a video sequence is in R^2, the latent variable $Z \in R^2$ may not simply represent the moving direction of the object, for the following reasons:

- The moving direction and Z may not be in the same coordinate system, as the decoder from the latent variable space $\mathcal{Z}$ to the video space $\mathcal{H}$ may contain rotation operations.
- The latent variable Z may contain not only direction information, but also velocity, momentum, and other information.
- The latent variable Z may contain information related to environmental changes.

Thus, the angles between any two vectors in the latent variable space R^2 may not be preserved in the decoding process.

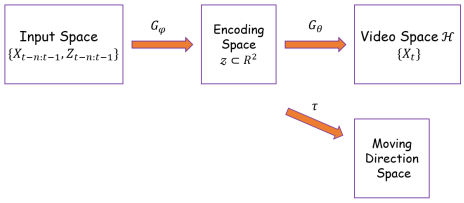

Figure 4: Illustration of our modified model for action control.

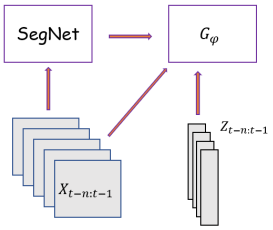

Figure 5: Applying Image Segmentation Pretrained Network as feature extractor in our model leverages the accuracy of our prediction.

To overcome this problem, we add a network to our framework to preserve such angles. This network acts as a mapping, denoted τ, which maps a latent variable from the latent variable space $\mathcal{Z}$ to the moving direction space $\mathcal{D}$. The moving direction $v(X_{t-1}, X_t)$ of an object between frames X_{t-1} and X_t can be computed by running an optical flow algorithm [Beauchemin and Barron, 1995]. Thus, our modified model consists of two decoders: one from the latent variable space $\mathcal{Z}$ (i.e., R^2) to the video space $\mathcal{H}$ as discussed in the previous subsections, and the other decoder, τ, from the latent variable space $\mathcal{Z}$ to the moving direction space $\mathcal{D}$. Figure 4 illustrates this modified model.

The moving direction loss, denoted L_{moving}, is calculated as:

$$L_{moving} = |\frac{< \tau(Z), v(X_{t-1}, X_t) >}{|\tau(Z)| \cdot |v(X_{t-1}, X_t)|} - 1| \quad (13)$$

where $< \cdot >$ represents the inner product of two vectors. Such a loss function penalizes the angle difference between two vectors. Our overall training loss is updated to take into account the moving direction loss, and is calculated as:

$$L_{loss} = \alpha L_{adv} + \beta L_{cycle} + \lambda L_{feature} + \mu L_{moving} \quad (14)$$

The Adam optimizer [Kingma and Ba, 2015] is employed to optimize L_{loss}.

Based on a mathematical concept known as 'conformal mapping' [Ahlfors, 1973], we introduce and add the network, τ, to our model. Formally, a mapping $\mathbf{f} = (f_1, \ldots, f_n)$ where $\mathbf{f} : U \to V, U, V \subset R^n$, is conformal (or angle-preserving) at a point $u_0 \in U$ if it preserves orientation and angles between directed curves through u_0. A mapping $\mathbf{f}$ is conformal iff it is homomorphic and its derivative is nowhere zero, i.e.,

$$\mathbf{J} = \frac{d\mathbf{f}}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \neq \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \quad (15)$$

In our VPGAN framework, the mapping τ is implemented using a 3-layer affine transform.

$$\begin{aligned} f^i &= A^i \cdot X + B^i \\ f &= f^1 \circ f^2 \circ f^3 \end{aligned} \quad (16)$$

It is very easy to prove that such affine transform enforced τ to be conformal, therefore it preserved angles between any two vectors through $'0'$. In this way, if we know a latent variable Z moving toward a specific direction, we can then control the generated moving direction by manipulating the latent variable Z (through rotating with some angle since the angle is preserved between the latent variable space $\mathcal{Z}$ and the moving direction space $\mathcal{D}$). Under this circumstance, we actually do not need to know details concerning Z such as velocity, momentum and other information. Algorithm 1 depicts our action control procedure.

Algorithm 1 Action Control

1: Sample n sets of continuous frames $\{X_{t-1}, X_t\}_n$ in which objects move toward the same direction.
2: Encode the frames into $Z_t^1, \ldots, Z_t^n$ in the latent variable space $\mathcal{Z}$.
3: Calculate the mean of $Z_t^1, \ldots, Z_t^n$ and denote the mean by $\bar{Z}$.
4: Compute the angle ς between the desired direction and sampled direction.
5: Rotate $\bar{Z}$ by the angle ς in the latent variable space $\mathcal{Z}$.
6: Decode $\bar{Z}$ into the video space $\mathcal{H}$.

The advantages of our proposed action control alogrithm are the following:

- It suffices to enforce a conformal mapping τ from the latent variable space $\mathcal{Z}$ to the moving direction space $\mathcal{D}$ (see Figure 4). It is not necessary to handle the latent variables $Z_t^1, \ldots, Z_t^n$ individually.
- Even when the latent variables accumulate many different factors, such as environmental changes, momentumn information and so on, our action control algorithm is still able to generate objects moving in the desired direction.

4 Experiments

We conducted a series of experiments to evaluate the performance of our VPGAN framework on different datasets, including Moving Mnist [Srivastava *et al.*, 2015], BAIR [Finn *et al.*, 2016], KTH Action Dataset [Laptev *et al.*, 2004].

We generated frames about object moving towards a specific direction on Moving Mnist dataset, as it contains object moving between various direction. Experiments on BAIR dataset illustrated our potential on generating multiple futures. We also show quantitative results by comparing structural similarity(SSIM) and Peak-Signal-to-Noise Ratio(PSNR) scores between ground truth and our generated future.

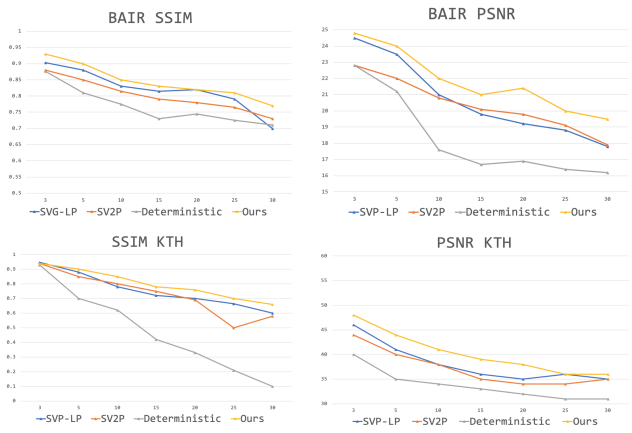

Figure 6: Quantitive results of our model on BAIR and KTH datasets comparing with SVG-LP, SV2P and deterministic models. (a) and (b) showed results on BAIR, while (c) and (d) showed results on KTH.

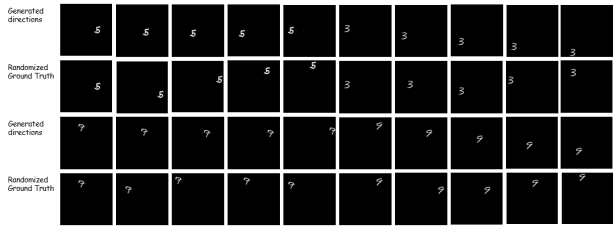

Figure 7: Generating desired movements of Mnist Character.As shown, '5' in moving leftwards, '3' is moving downwards, '7' is moving rightwards, '9' is moving 'rightdown'.

Note that in 'low-complexity' task like Moving Mnist, we're mainly exploring the potential of our action control approach, therefore we're using simple model architecture. In task such as 'BAIR', 'KTH', we applied image-segmentation models as our feature-extractor in generator just as in Figure 5.

4.1 Generating Desired Futures

In this experiment, we applied the Moving Mnist dataset [Srivastava *et al.*, 2015]. This dataset contains Mnist characters randomly moving towards each direction, we modified the background to be as large as 150×150. Training sequence were generated by per character moving towards various directions, to generate character moving in a specific direction. We follow the algorithm in 3.3 and figure 7 shows our generated results. We managed to produce Mnist character moving towards one specific direction, in this scenario, we did not apply a complicated structure for our generator $G_{theta}(Z, X)$ or $G_{\psi}(X, Z)$. Simple LSTM stacked above CNN would be enough.

4.2 BAIR dataset

The BAIR robot pushing dataset [Finn *et al.*, 2016] consists of a robotic arm pushing various objects around a table. Moving directions of the arm are highly random, thus it is suitable for generating possible futures conditioned on the past. The architecture we applied to our model is G_{theta} being a LSTM + pretrained Image segmentation model. Since the complexity of the Image space is not too large, we applied UNet [Ronneberger *et al.*, 2015] as our pretrained Image segmentation model. Figure 5 reveals that our quantitative results reached the state-of-the-art in BAIR dataset compared to SVL-LP [Denton and Fergus, 2018], SV2P [Lee *et al.*, 2018], and deterministic models, while figure 8 shows some of our examples generated.

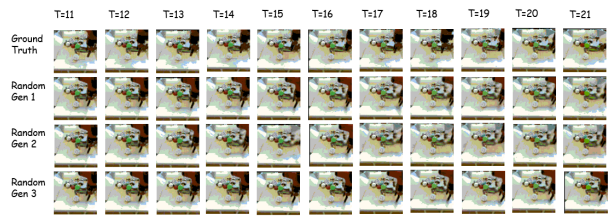

Figure 8: Examples of generated frames on BAIR dataset. We showed our potential of generating possible futures.

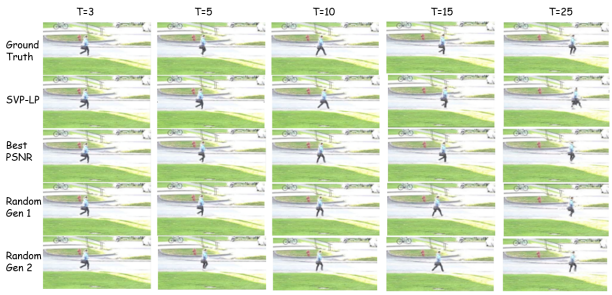

Figure 9: Examples of our generation in KTH action data. As shown in the last column, our result generate more acute results compared to SVP-LP due to the introduced of cycle-consistency.

4.3 KTH dataset

KTH action data [Laptev *et al.*, 2004] contains various types of videos collected in real-world cameras. We tested our model applying the similar structure as in BAIR dataset, except for we switch UNet to SegNet [Badrinarayanan *et al.*, 2017], since SegNet is more powerful and is pretrained on real-world images, the object mask given by SegNet could leverage our results. Figure 6 gives our quantitative results on KTH dataset. Examples of generated frames and comparison with SVP-LP was done in Figure 9.

Conclusion

In this paper, we presented an adversarial inference framework for video prediction, and introduced cycle consistency and conformal mapping into our framework. Experiments revealed that cycle-consistency relieved the problem of blur predictions and preserve long-time acute results, while conformal mapping enabled us to control our prediction futures through manipulating the latent variables.

References

[Ahlfors, 1973] Lars V. Ahlfors. *Conformal Invariants: Topics in Geometric Function Theory*. McGraw-Hill, New York, 1973.

[Babaeizadeh *et al.*, 2018] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. In *6th International Conference on Learning Representations*, 2018.

[Badrinarayanan *et al.*, 2017] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(12):2481–2495, 2017.

[Beauchemin and Barron, 1995] Steven S. Beauchemin and John L. Barron. The computation of optical flow. *ACM Comput. Surv.*, 27(3):433–467, 1995.

[Denton and Birodkar, 2017] Emily L. Denton and Vighnesh Birodkar. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems*, pages 4417–4426, 2017.

[Denton and Fergus, 2018] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1182–1191, 2018.

[Donahue *et al.*, 2017] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *5th International Conference on Learning Representations*, 2017.

[Dumoulin *et al.*, 2017] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martín Arjovsky, Olivier Mastropietro, and Aaron C. Courville. Adversarially learned inference. In *5th International Conference on Learning Representations*, 2017.

[Finn *et al.*, 2016] Chelsea Finn, Ian J. Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, pages 64–72, 2016.

[Goodfellow *et al.*, 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[Hu *et al.*, 2018] Zhihang Hu, Turki Turki, Nhathai Phan, and Jason T. L. Wang. A 3D atrous convolutional long short-term memory network for background subtraction. *IEEE Access*, 6:43450–43459, 2018.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

[Kingma, 2013] Diederik P. Kingma. Fast gradient-based inference with continuous latent variable models in auxiliary form. *CoRR*, abs/1306.0733, 2013.

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

[Lee *et al.*, 2018] Alex X Lee, Richard Zhang, Frederik Ebert, Pieter Abbeel, Chelsea Finn, and Sergey Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

[Mathieu *et al.*, 2016] Michaël Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. 2016.

[Pu *et al.*, 2016] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. In *Advances in Neural Information Processing Systems*, pages 2352–2360, 2016.

[Ranzato *et al.*, 2014] MarcAurelio Ranzato, Arthur Szlam, Joan Bruna, Michael Mathieu, Ronan Collobert, and Sumit Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Srivastava *et al.*, 2015] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 843–852, 2015.

[Tulyakov *et al.*, 2018] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526–1535, 2018.

[Villegas *et al.*, 2017] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017.

[Zhu *et al.*, 2017] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision*, pages 2242–2251, 2017.