A Novel Adversarial Inference Framework for Video Prediction with Action Control*

Zhihang Hu and Jason T. L. Wang
New Jersey Institute of Technology
Newark, NJ 07102, USA
ashvegance@gmail.com, 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 often result in blur predictions and lack the ability of action control. To tackle these problems, we propose a framework, called VPGAN, which 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. Experimental results show that a combination of VPGAN with some pre-trained 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 [15, 16]. This task finds many applications ranging from autonomous driving, robotic planning, to object tracking. In practice, unlabelled 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 [2, 14].

*The first workshop on Statistical Deep Learning for Computer Vision, in *Seoul, Korea*, 2019.

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 [6]. A typical example is that the future trajectory of a ball hitting the ground is inherently random. Deterministic methods [16, 17, 20] are unable to handle this inherent uncertainty. Although adversarial based methods could predict more acute results, they lack the ability to produce specific 'future predictions.'

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.
- We incorporate a conformal mapping [1] network structure into our VPGAN framework to enable action control for generating desirable future frames.
- We combine pre-trained image segmentation models [3, 18] with our VPGAN framework to exploit their effectiveness in image understanding. Having more semantic understanding of the frames in video sequences would enable VPGAN to generate more accurate predictions.

The combination of our VPGAN framework with the pre-trained image segmentation models outperforms existing stochastic video prediction methods as shown in our experimental results reported in the paper.

2. 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 $\{X_T\}$.

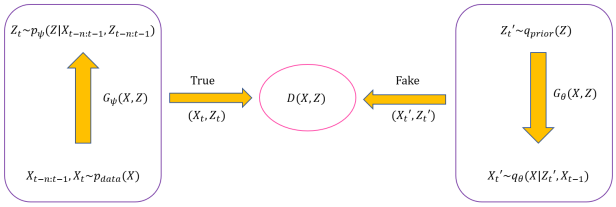

Figure 1. Illustration of our VPGAN learning process. Both G_ψ and G_θ are generators. Discriminator $D(X, Z)$ tries to discriminate between true pair (X, Z) and fake pair (X', Z').

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 [7, 21, 22], 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 constraints 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 $(X_0, Z_0), \ldots, (X_t, Z_t), 0 \leq t < T$.

2.1. Adversarial Inference

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 $p_{data}(X)$.

Figure 1 illustrates our 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'.

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:

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

To generate or predict the next frame X_t based on the past frames $X_{t-n:t-1}$, 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:

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

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

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 re-parameterization trick [11]. 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:

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

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

2.2. 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 [23]. However, our cycle consistency loss function is different from that in [23] 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. We generate the current frame X_t conditioned on the previous frame X_{t-1} 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

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

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:

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

Our multi k step cycle-consistent loss is generalized as summing up all single-multi losses:

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

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

Combining the multi-cycle loss and multi-reconstruction loss, our overall loss, denoted as L_{loss}, is calculated as follows,

$$L_{loss} = \alpha L_{adv} + \beta L^k_{cycle} + \lambda L^k_{recon} \tag{10}$$

The perceptual loss [10] is widely used 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 [19], ResNet [9] pre-trained on ImageNet [5]. In our paper, we applied simple ResNet [9] 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; furthermore it may suffer from gradient loss. Therefore, in our VPGAN framework, we only utilize the one-step cycle consistency loss given in (7). The evaluation of different k values on L^k_{cycle} will be in a future paper.

2.3. Action Control

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.

Thus, the angles between any two vectors in the latent variable space R^2 may not be preserved in the decoding process. 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 [4]. 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 2 illustrates this modified model.

The moving direction loss, denoted L_{moving}, is calcu-

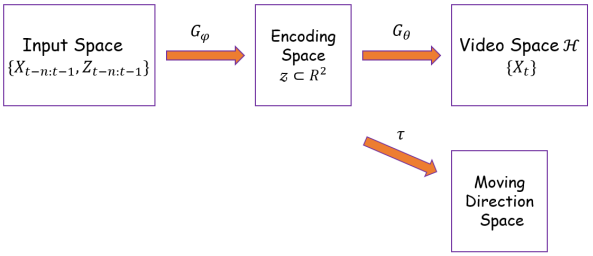

Figure 2. Illustration of our modified model for action control.

lated as:

$$L_{moving} = |\frac{< \tau(Z), v(X_{t-1}, X_t) >}{|\tau(Z)| \cdot |v(X_{t-1}, X_t)|} - 1| \quad (11)$$

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:

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

The Adam optimizer [12] is employed to optimize L_{loss}.

Based on a mathematical concept known as 'conformal mapping' [1], we introduce and add the network, τ, to our model. A mapping $\mathbf{f}$ is conformal iff it is homomorphic and its derivative is nowhere zero. In our VPGAN framework, the mapping τ is implemented using a 3-layer affine transform. It is very easy to prove that such an affine transform enforces τ to be conformal, therefore it preserves angles between any two vectors through $'0'$. In this way, if we know a latent variable Z moving toward a specific direction, we can 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.

The advantages of our proposed action control algorithm 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 2). It is not necessary to handle the latent variables $Z^1_t, \ldots, Z^n_t$ individually.

- Even when the latent variables accumulate many different factors, such as environmental changes, momentum information and so on, our action control algorithm is still able to generate objects moving in the desired direction.

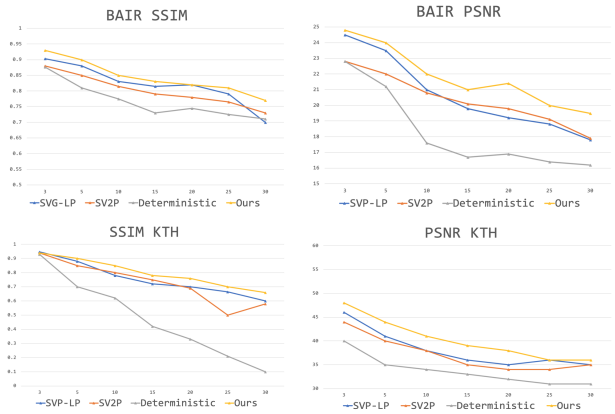

Figure 3. Performance results of our approach on BAIR and KTH datasets compaired with SVG-LP, SV2P and deterministic models.

3. Results

A series of experiments were conducted to evaluate the performance of our VPGAN framework on different datasets, including BAIR [8] and KTH Action Dataset [13]. The BAIR robot pushing dataset [8] involves a series of videos generated by a Sawyer robotic arm pushing a variety of objects. All of the videos have relatively similar surroundings (table settings) with a static background. Each video consists of actions taken by the robotic arm corresponding to the commanded gripper pose. The resolution of the videos is of 64×64, therefore, our input dimension is $64 \times 64 \times 3$. We conditioned on 10 frames to produce 30 frames.

The KTH action dataset [13] contains various types of videos collected in real-world cameras including a human subject doing one of six activities (walking, jogging, running, boxing, hand waving, and hand clapping). For the first three activities, the human enters and leaves the frame multiple times, leaving the frame empty with a mostly static background for multiple frames at a time.

Figure 3 shows performance results of our VPGAN model combined with pre-trained image segmentation models [3, 18] in comparison with existing SVG-LP [6], SV2P [2] and deterministic [16] methods. It can be seen from Figure 3 that our approach performs better than the related methods.

4. Conclusion

We presented a new approach for video prediction with action control. This approach consists of a new adversarial inference model, a novel cycle-consistency loss function, and a conformal mapping network structure for enabling action control. Our experimental results demonstrated good performance of the proposed approach and its superiority over existing methods.

References

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

[2] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine. Stochastic variational video prediction. In *6th International Conference on Learning Representations*, 2018.

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

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

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

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

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

[8] C. Finn, I. J. Goodfellow, and S. Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, pages 64–72, 2016.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

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

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

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

[14] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine. Stochastic adversarial video prediction. *arXiv preprint arXiv:1804.01523*, 2018.

[15] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. 2016.

[16] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. P. Singh. Action-conditional video prediction using deep networks in Atari games. In *Advances in Neural Information Processing Systems*, pages 2863–2871, 2015.

[17] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv preprint arXiv:1412.6604*, 2014.

[18] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, 2015.

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

[20] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 843–852, 2015.

[21] S. Tulyakov, M.-Y. Liu, X. Yang, and J. 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.

[22] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017.

[23] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision*, pages 2242–2251, 2017.