Long-Term Video Prediction via Criticization and Retrospection

Xinyuan Chen, Chang Xu, *Member, IEEE*, Xiaokang Yang, *Fellow, IEEE*, and Dacheng Tao, *Fellow, IEEE*

***Abstract*— Video prediction refers to predicting and generating future video frames given a set of consecutive frames. Conventional video prediction methods usually criticize the discrepancy between the ground-truth and predictions frame by frame. As the prediction error accumulates recursively, these methods would easily become out of control and are often confined to the short-term horizon. In this paper, we introduce a retrospection process to rectify the prediction errors beyond criticizing the future prediction. The introduced retrospection process is designed to look back what have been learned from the past and rectify the prediction deficiencies. To this end, we build a retrospection network to reconstruct the past frames given the currently predicted frames. A retrospection loss is introduced to push the retrospection frames being consistent with the observed frames, so that the prediction error is alleviated. On the other hand, an auxiliary route is built by reversing the flow of time and executing a similar retrospection. These two routes interact with each other to boost the performance of retrospection network and enhance the understanding of dynamics across frames, especially for the long-term horizon. An adversarial loss is employed to generate more realistic results in both prediction and retrospection process. In addition, the proposed method can be used to extend many state-of-the-art video prediction methods. Extensive experiments on the natural video dataset demonstrate the advantage of introducing the retrospection process for long-term video prediction.**

***Index Terms*— Video prediction, generative adversarial networks.**

I. Introduction

VIDEO prediction refers to predicting future video frames by observing a sequence of video frames. Accurately predicting future frames is important in video coding, video completion, robotics, autonomous driving and intelligent agents that interact with their environment. For example, self-driving cars need to make a prediction of the passing vehicles. This prediction capability is also vital for autonomous systems in tasks of path planning and interaction with humans. In recent years, video prediction has attracted increasingly attention from the computer vision community and improved the performance based the deep neural networks and generative adversarial networks [1]–[4].

To predict reasonable future frame in natural videos, different distance metrics have been introduced to measure the discrepancy between predicted frames and ground-truth frames, (e.g., Euclidean distance [5]–[7]), as well as the discrepancy between the distribution of predicted frames and ground-truth frames (e.g., KL distance [2], and Wasserstein distance [1]) by adopting the generative adversarial networks. For instance, [5]–[7] train predicted models by minimizing mean squared error (MSE) between the next ground-truth frame and the predicted next frame. Since there are many possible futures, the model with MSE tends to output an average over many possible images, and causes a blurry result. To achieve sharper and more realistic results, MCNET [2] combines adversarial loss [8] and MSE loss in the video prediction framework. DRNET [9] designs an adversarial training strategy to disentangle the motion and content representations, so that the video prediction is achieved by a combination of the extracted content features and the predicted motion features.

These existing methods often execute in a recursive manner by taking the newly generated frames as observations to generate subsequent frames. They usually produce high quality predictions for the first few steps. But the prediction would then dramatically degrade, and could even lead to totally missing of the video context or keeping a stationary frame. Looking at the future, these methods continuously criticize the discrepancy between predicted and ground truth frames in a recursive generation process without pause, and the prediction error would accumulate and gradually become difficult to control. In addition to criticizing the future prediction, we are better to retrospect what have been learned from the past and rectify any deficiencies. The proposed retrospection process is used to alleviate the prediction error accumulation between the predicted frames and the ground-truth frames. The introduced retrospection process provides an auxiliary loss to improve the quality of predicted frames. If the predictions are not accurate or in high quality, the retrospection network is hard to well reconstruct past frames. As a result, to minimize the difference between the retrospection frames and the original frames in the

Manuscript received July 18, 2019; revised March 28, 2020 and May 24, 2020; accepted May 24, 2020. Date of publication June 3, 2020; date of current version July 8, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1001003, in part by the National Natural Science Foundation of China (NSFC) under Grant U19B2035 and Grant 61527804, in part by the Science and Technology Commission of Shanghai Municipality (STCSM) under Grant 18DZ1112300, and in part by the Australian Research Council Projects (ARC) under Grant FL-170100117, Grant DP-180103424, Grant IH-180100002, and Grant DE-180101438. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Chia-Kai Liang. *(Corresponding author: Chang Xu.)*

Xinyuan Chen is with the MoE Key Laboratory of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China, also with the Centre for Artificial Intelligence, University of Technology Sydney, Ultimo, NSW 2007, Australia, and also with the Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: xychen91@sjtu.edu.cn).

Chang Xu and Dacheng Tao are with the UBTech Sydney Artificial Intelligence Centre, The University of Sydney, Darlington, NSW 2008, Australia, and also with the Faculty of Engineering, School of Computer Science, The University of Sydney, Darlington, NSW 2008, Australia (e-mail: c.xu@sydney.edu.au; dacheng.tao@gmail.com).

Xiaokang Yang is with the MoE Key Laboratory of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: xkyang@sjtu.edu.cn).

Digital Object Identifier 10.1109/TIP.2020.2998297

retrospection module, the predictive network is encouraged to produce high-quality predicted frames.

In this paper, we propose to achieve long-term video predictions by criticizing the future and retrospecting the past. Instead of taking video prediction as a single feed-forward process, we suggest that a qualified generator would also be able to predict videos in a backward manner. A new retrospection network is developed to reconstruct the past frames given the current predicted frames. During the prediction process, a prediction loss is employed to minimize the discrepancy between the predicted and the ground-truth frames. After predicting in a few time steps, we pause the prediction process and look back to rectify the prediction deficiencies in a backward manner. A retrospection loss is introduced to minimize the distance between the retrospected frames and the original frames. Also, we build an auxiliary route to train the retrospection network, where retrospection network first generates a few past frames in backward, then pauses to check if the generated frames could predict the future accurately by the prediction network. Standard feed-forward prediction networks can be flexibly integrated with the proposed video retrospection network to improve the training stage of video generator. The retrospection operation will be dropped in the test, so that test efficiency can be preserved.

We conducted both qualitative and quantitative experiments on three natural video datasets, e.g., the KTH [10], Weizmann [11] and UCF-101 datasets [12]. Experiments demonstrate that the proposed algorithm can boost visual quality of generated videos and lead to more precise results in long-term prediction, which significantly outperforms prior arts. We apply the proposed retrospection process to many state-of-the-art video prediction methods. Experiments reveal the generality of our approach.

II. Related Work

Many deep learning techniques for video prediction have emerged recently. We review relevant studies related to video prediction using deep neural networks. Over the last few years, CNNs and recurrent neural networks (RNNs) have gained huge popularity and a number of studies [13]–[16] applied CNNs and RNNs to predict future frames from an image sequence. The video prediction problem was initially studied at the patch level containing synthetic motions [17]–[19]. Reference [20] adopted a discrete vector quantization approach to performing patch-level video prediction. However, predicting patches encounters the well-known aperture problem that causes blockiness as prediction advances in time.

In pixel level, a lot of works have emerged on video prediction since convolutional LSTM (Conv-LSTM) [21] has successfully been applied in a large variety of computer vision research area [22]–[24]. Several studies [5], [13], [15] adopted convolutional LSTM to take spatial and temporal contexts into account. Reference [25] proposed a video generation framework which utilized the Conv-LSTM to encode short-term and long-term spatial-temporal context for semantic video generation using captions. Reference [26] combined the ConvLSTM into a deep generative model which modeled the factorization of the joint likelihood of inputs in the form of video data. References [5], [13], [15] adopted convolutional LSTM to take spatial and temporal contexts into account. Reference [25] proposed a video generation framework which utilized the Conv-LSTM to encode short-term and long-term spatial-temporal context for semantic video generation using captions. Reference [26] combined the ConvLSTM into a deep generative model which modeled the factorization of the joint likelihood of inputs in the form of video data.

On the other hand, some methods managed to ease the task by introducing various prior knowledge. One popular hypothesis is that a video sequence could be decomposed as content and motion. By independently modeling the motion and content, MCNET [2] predicted the next frame by combining the predicted motion feature and the extracted content feature. DRNET [9] designed an adversarial training strategy to disentangle the motion and content representations. Another assumption is that the outcome of an event is stochastic as a consequence of the latent events, so different possible future for each sample of its latent variables can be predicted [27]. Reference [4] incorporated a two steam generation architecture to deal with high frequency and low-frequency video content separately so as to output structured prediction. Reference [28] produced a mask that outlines the predicted foreground object to achieve a better quality of prediction.

Other methods exploited external labeled notation to facilitate future prediction. For example, [7] developed an action-conditioned video prediction framework that utilized action prior knowledge as well as previous appearance information to predict futures in the game. Reference [1] utilized labeled optical flow to simultaneously solve both video prediction and optical flow estimation. Reference [29] took advantage of the scene geometry and use the predicted depth for generating the next frame prediction. Reference [30] made use of geometry-aware deep network model by accessing camera intrinsic parameters to predict next frames of cityscapes. The mentioned methods used annotations to facilitate video prediction, such as action annotations, optical flow, and skeleton information. The annotated labels are required manually annotations. Some of the labels such as optical flow can be estimated by algorithms, but the estimation error would decrease the quality of the generated frame.

Traditional video prediction networks are trained by minimizing mean square error (MSE) between predicted and ground truth frames [5]–[7]. However, since there are many possible futures, the model with MSE tends to output an average over many possible images and causes a blurry result. Generative adversarial networks (GANs) [8] is an implicit generative model, implemented by a two-player game: a generator and a discriminator. The generator aims to generate realistic images to fool the discriminator, while the discriminator aims to classify whether the images are real or fake. The idea of an adversarial loss that forces the generated images to be, in principle, indistinguishable from real images. After the invention of adversarial training, many studies applied this scheme to generate images in the context of image generation [31], [32], image-to-image translation [33]–[35], text-to-image generation [36], etc. In video prediction area, MCNET [37] proposed

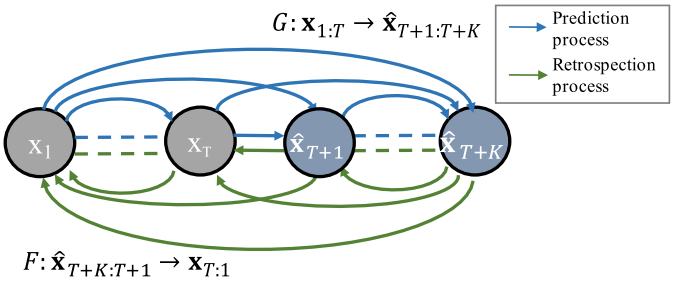

Fig. 1. The illustration of our model. The generative model predicts the next frame conditioned on the previous frames (blue lines), while our model introduces a retrospection process to reconstruct the original input frames given predictions in a reverse chronological order (green lines).

adversarial training with multi-scale convolutional networks to generate sharper predictions. Reference [38] exploited spatial and motion constraints in addition to intensity and gradient losses. They computed optical flow through FlowNet [39] and the flow information is used to predict temporally consistent frames. Reference [3] proposed a long-term prediction method in a hierarchical approach. On the other hand, stochastic video prediction methods have emerged that output samples of possible future distribution. The stochastic video prediction methods are a promising solution to obtain realistic and sharp future frames, e.g., SAVP [40] and SVG [41].

In our work, we observe that existing video prediction methods often execute in a single recursive manner, and the prediction error would accumulate over time. Therefore, we introduce a retrospection process to look back what has been learned and rectify the prediction deficiencies. CycleGAN [33] brings the translated images back to the original image. While the high-level idea is similar, we explore the retrospection idea in video prediction, and a novel formulation has been developed.

III. Method

Figure 1 illustrates our long-term video prediction model. Our model includes two direction mappings: prediction process and retrospection process. The prediction process aims to predict the subsequent frames while the retrospection process aims to reconstruct the observed frames in a reverse chronological order. Let $\mathbf{x}_t \in \mathrm{R}^{w \times h \times c}$ denote the t-th frame of an input video, where w, h, and c denote the width, height, and number of channels, respectively. Assuming that we have T observed frames $\{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$, the prediction frame at $(T+1)$ time index is produced by a video prediction model $\hat{\mathbf{x}}^G_{T+1} = G(\mathbf{x}_{1:T})$. Then the subsequent frames is generated by observing the predicted frames recursively (see blue lines). That is to say, at $(T+k)$ time index, the prediction $\hat{\mathbf{x}}_{T+k}$ is achieved by observing both the groun850d-truth frames $\{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$ and predicted frames $\left\{\hat{\mathbf{x}}^G_{T+1}, \cdots, \hat{\mathbf{x}}^G_{T+k-1}\right\}$. We express the formula as:

$$\hat{\mathbf{x}}^G_{T+k} = \begin{cases} G(\mathbf{x}_{1:T}), & k = 1 \\ G([\mathbf{x}_{1:T}, \hat{\mathbf{x}}^G_{T+1:T+k-1}]), & 1 < k \leqslant K. \end{cases} \quad (1)$$

As a consequence, the quality of subsequent predictions relies on the accuracy of the previous predictions. Most of the time, the prediction deviation would accumulate and the quality of prediction frames would decrease dramatically. In our method, we introduce a retrospection process (e.g., green arrows in Figure 1) to further alleviate prediction deviation. In the retrospection process, the predicted frames are required to reconstruct the input frame backward. The retrospection model F is constructed to estimate the reconstruction frames $\tilde{\mathbf{x}}^F_T = F(\hat{\mathbf{x}}^G_{T+K:T+1})$ given the observed predicted frames $\left\{\hat{\mathbf{x}}_{T+K}, \cdots, \hat{\mathbf{x}}_{T+1}\right\}$ in a reverse chronology order. The preceding frames are generated recursively by the retrospection network:

$$\tilde{\mathbf{x}}^F_t = \begin{cases} F\left(\hat{\mathbf{x}}^G_{T+K:T+1}\right), & t = T \\ F\left(\left[\hat{\mathbf{x}}^G_{T+K:T+1}, \tilde{\mathbf{x}}^F_{T:t+1}\right]\right), & 1 \leqslant t < T, \end{cases} \quad (2)$$

where the $\left[\hat{\mathbf{x}}^G_{T+K:T+1}, \tilde{\mathbf{x}}^F_{T:t+1}\right]$ indicates the concatenation of the predicted frames $\hat{\mathbf{x}}^G_{T+K:T+1}$ and retrospections $\tilde{\mathbf{x}}^F_{T:t+1}$ in the channel of time.

The rest of this section organized as follow. We first briefly review the previous methods for predicting future frames. Then we introduce the proposed retrospection process in detail. Lastly, we summarize the full objective function of our model.

A. Preliminaries: Prediction Process

We first briefly review the previous video prediction model which only contains the prediction process. Normally, the video prediction model G is trained to predict K subsequent frames given T observed frames. Let the training data set denote as $D = \{\mathbf{x}^{(i)}_{1,\cdots,T+K}\}_{i=1}^N$ where i indicates the index of videos. A prediction loss of $\mathcal{L}_{img}$ is usually adopted to minimize the distance between predicted frames and ground-truth. The function can be expressed as follow:

$$\mathcal{L}_{img}(G) = \mathcal{L}_p\left(\mathbf{x}_{T+k}, \hat{\mathbf{x}}^G_{T+k}\right) + \mathcal{L}_{gdl}\left(\mathbf{x}_{T+k}, \hat{\mathbf{x}}^G_{T+k}\right), \quad (3)$$

where $\mathcal{L}_P$ guides the model to match the average pixel values, and $\mathcal{L}_{gdl}$ guides the model to match the gradients of pixel values between ground-truth and predicted frames:

$$\mathcal{L}_p(\mathbf{y}, \mathbf{z}) = \sum_{k=1}^{K} ||\mathbf{y} - \mathbf{z}||_p^p, \quad (4)$$

$$\mathcal{L}_{gdl}(\mathbf{y}, \mathbf{z}) = \sum_{i,j}^{h,w} |\left(|\mathbf{y}_{i,j} - \mathbf{y}_{i-1,j}| - |\mathbf{z}_{i,j} - \mathbf{z}_{i-1,j}|\right)|^{\lambda} + |\left(|\mathbf{y}_{i,j-1} - \mathbf{y}_{i,j}| - |\mathbf{z}_{i,j-1} - \mathbf{z}_{i,j}|\right)|^{\lambda}. \quad (5)$$

The $\mathbf{x}_{T+k}$ and $\hat{\mathbf{x}}_{T+k}$ are the ground-truth and predicted frames at time $T+k$ respectively. Since training to generate average sequences would result in blurry generations [42], the additional adversarial loss $\mathcal{L}_{GAN}$ is used to produce realistic visual outputs [2]:

$$\mathcal{L}_{GAN}(G) = -\log D_g\left(\left[\mathbf{x}_{1:T}, \hat{\mathbf{x}}^G_{T+1:T+K}\right]\right), \quad (6)$$

where $\mathbf{x}_{1:T}$ is the concatenation of the input images, $\hat{\mathbf{x}}_{T+1:T+K^G}$ is the concatenation of all predicted images along the time dimension, and $D_g(\cdot)$ is the discriminator in adversarial training. The discriminative loss for discriminator D_g in

TABLE I
SUMMARY OF NOTATION

Notation	Description
$\mathbf{x}_t$	ground-truth frame at time t
$\hat{\mathbf{x}}_t^G$	frame predicted by network G given the true past frames
$\tilde{\mathbf{x}}_t^G$	frame predicted by network G given the generated past frames
$\hat{\mathbf{x}}_t^F$	frame generated by network F given the true past frames
$\tilde{\mathbf{x}}_t^F$	frame generated by network F given the predictions
$\mathbf{x}_{t_1:t_2}$	concatenation of frames $\mathbf{x}_t$ from t_1 to t_2
$[\mathbf{x}_{t_1}, \mathbf{x}_{t_2}]$	concatenation of frame $\mathbf{x}_{t_1}$ and frame $\mathbf{x}_{t_2}$
$[\mathbf{x}_{t_1:t_2}, \mathbf{x}_{t_3:t_4}]$	concatenation of frames from t_1 to t_2 and t_3 to t_4

adversarial training is defined by minimizing:

$$\mathcal{L}_{GAN}\left(D_g\right) = -\log D_g\left(\left[\mathbf{x}_{1:T}, \mathbf{x}_{T+1:T+K}\right]\right) - \log\left(1 - D_g\left(\left[\mathbf{x}_{1:T}, \hat{\mathbf{x}}^G_{T+1:T+K}\right]\right)\right), \quad (7)$$

where $\mathbf{x}_{T+1:T+K}$ is the concatenation of the ground-truth future images. $\mathcal{L}_{GAN}$ and $\mathcal{L}_{img}$ allow the model to criticize the quality of the prediction sequence, by comparing the predicted frames with the ground-truth.

B. Retrospection Process

As shown in left-bottom of Fig 2, our model consists of two routes: route $G \rightarrow F$ and route $F \rightarrow G$. The route $G \rightarrow F$ is designed to train the prediction network, and the auxiliary retrospection network is adopted to reconstruct the prediction to the past. The other route $F \rightarrow G$, on the other hand, is built to train the retrospection network. In the route $F \rightarrow G$, the retrospection network first generates a few past frames in backward, then pauses to check if the generated frames could be used to predict the future accurately by prediction network.

1) Route $G \rightarrow F$: In the route $G \rightarrow F$, our proposed model consists of two generative networks: a forward prediction network G and a retrospection network F. The forward network G is to perform predictions, which is similar to the previous work [2]; the retrospection network F is introduced to retrospect the past by observing current predictions in Equation 2. The retrospection loss $\mathcal{L}_{ret1}$ is introduced to minimize the distance between the retrospections $\tilde{\mathbf{x}}^F_{1:T}$ and the original frames $\mathbf{x}_{1:T}$:

$$\mathcal{L}_{ret1}\left(F, G\right) = \mathcal{L}_p\left(\mathbf{x}_t, \tilde{\mathbf{x}}_t^F\right) + \mathcal{L}_{gdl}\left(\mathbf{x}_t, \tilde{\mathbf{x}}_t^F\right), \quad t \in \{1:T\} \quad (8)$$

Notice that only one route $G \rightarrow F$ cannot guarantee to alleviate the deviation of predicted frames. The retrospection network F can hardly be trained to reconstruct the exact original frames from noisy frames. As a result, another route $F \rightarrow G$ is necessary to train the retrospection network by observing the ground-truth frames.

2) Route $F \rightarrow G$: In order to train the retrospection network F, the inverse route $F \rightarrow G$ is built. In route $F \rightarrow G$, we feed the inverse of frame sequence $\{\mathbf{x}_{T+K}, \cdots, \mathbf{x}_T\}$ into the network, and encourage the network to generate preceding frames recursively:

$$\hat{\mathbf{x}}_t^F = \begin{cases} F\left(\mathbf{x}_{T+K:T+1}\right), & t = T \\ F\left(\left[\mathbf{x}_{T+K:T+1}, \hat{\mathbf{x}}^F_{T:t+1}\right]\right), & 1 \leqslant t < T, \end{cases} \quad (9)$$

where $\hat{\mathbf{x}}^F_{T:t+1}$ indicates the retrospection frames of network F. A reconstruction loss is defined similar to Equation 3 to minimize the distance between retrospections $\hat{\mathbf{x}}_t^F$ and ground-truth frames $\mathbf{x}_t$:

$$\mathcal{L}_{img}(F) = \mathcal{L}_p\left(\mathbf{x}_t, \hat{\mathbf{x}}_t^F\right) + \mathcal{L}_{gdl}\left(\mathbf{x}_t, \hat{\mathbf{x}}_t^F\right), \quad t \in \{1:T\}. \quad (10)$$

Similar to route $G \rightarrow F$, the retropsections $\hat{\mathbf{x}}^F_{1:T}$ are required to capable of predicting the frames $\tilde{\mathbf{x}}^G_{T+1:T+K}$ by prediction network recursively:

$$\tilde{\mathbf{x}}^G_{T+k} = \begin{cases} G(\hat{\mathbf{x}}^F_{1:T}), & k = 1 \\ G(\left[\hat{\mathbf{x}}^F_{1:T}, \tilde{\mathbf{x}}^G_{T+1:T+k-1}\right]), & 1 < k \leqslant K. \end{cases} \quad (11)$$

The retrospection loss $\mathcal{L}_{ret2}$ is introduced to rectify the error between the predicted frames and original input frames:

$$\mathcal{L}_{ret2}\left(F, G\right) = \mathcal{L}_p\left(\mathbf{x}_{T+k}, \tilde{\mathbf{x}}^G_{T+k}\right) + \mathcal{L}_{gdl}\left(\mathbf{x}_{T+k}, \tilde{\mathbf{x}}^G_{T+k}\right), \quad k \in \{1:K\}, \quad (12)$$

where $\tilde{\mathbf{x}}^G_{T+k}$ indicates the reconstructed frames of network G.

In our model, the retrospection loss $\mathcal{L}_{ret1}$ in Equation 8 is used to alleviate the accumulation error produced from forward video prediction model G, meanwhile $\mathcal{L}_{ret2}$ in Equation 12 is used to alleviate the accumulation error produced from retrospection model F. We analyzed the function of the two-route model in Section V-C. In order to train the retrospection network, we introduce the adversarial loss to cheat the discriminator D_f to classify the retrospection samples as real samples:

$$\mathcal{L}_{GAN}(F) = -log D_f(\left[\mathbf{x}_{T+K:T+1}, \hat{\mathbf{x}}^F_{T:1}\right]). \quad (13)$$

The discriminator aims to distinguish between the real frames and the retrospected frames in a reverse chronological order. The adversarial loss of the discriminative network is defined by minimizing the function:

$$\mathcal{L}_{\text{GAN}}\left(D_f\right) = -\log D_f\left(\left[\mathbf{x}_{T+K:T+1}, \mathbf{x}_{T:1}\right]\right) - \log\left(1 - D_f\left(\left[\mathbf{x}_{T+K:T+1}, \hat{\mathbf{x}}^F_{T:1}\right]\right)\right), \quad (14)$$

where $\mathbf{x}_{T+K:T+1}$ indicates reversed ground-truth video sequence, while $\hat{\mathbf{x}}^F_{T+K:T+1}$ denotes reversed generated frames of retrospection network F. Here we adopt a conditional GAN [43] that conditioned on the reversed ground-truth frames $\mathbf{x}_{T+K:T+1}$.

C. Full Objective

Our full objective for generative models of two process can be summarized as:

$$\begin{aligned}\mathcal{L}(G, F) = {} & \alpha_1\mathcal{L}_{img}(G) + \alpha_2\mathcal{L}_{img}(F) \\ & + \beta_1\mathcal{L}_{GAN}(G) + \beta_2\mathcal{L}_{GAN}(F) \\ & + \gamma_1\mathcal{L}_{ret1}(F, G) + \gamma_2\mathcal{L}_{ret2}(F, G) \end{aligned} \quad (15)$$

where α_1, α_2, β_2, β_1, β_2, γ_1 and γ_2 control the relative importance. In our model, the predictions of network G are required not only to minimize the discrepancy with the ground-truth frames, but also to be able to retrospect to the original

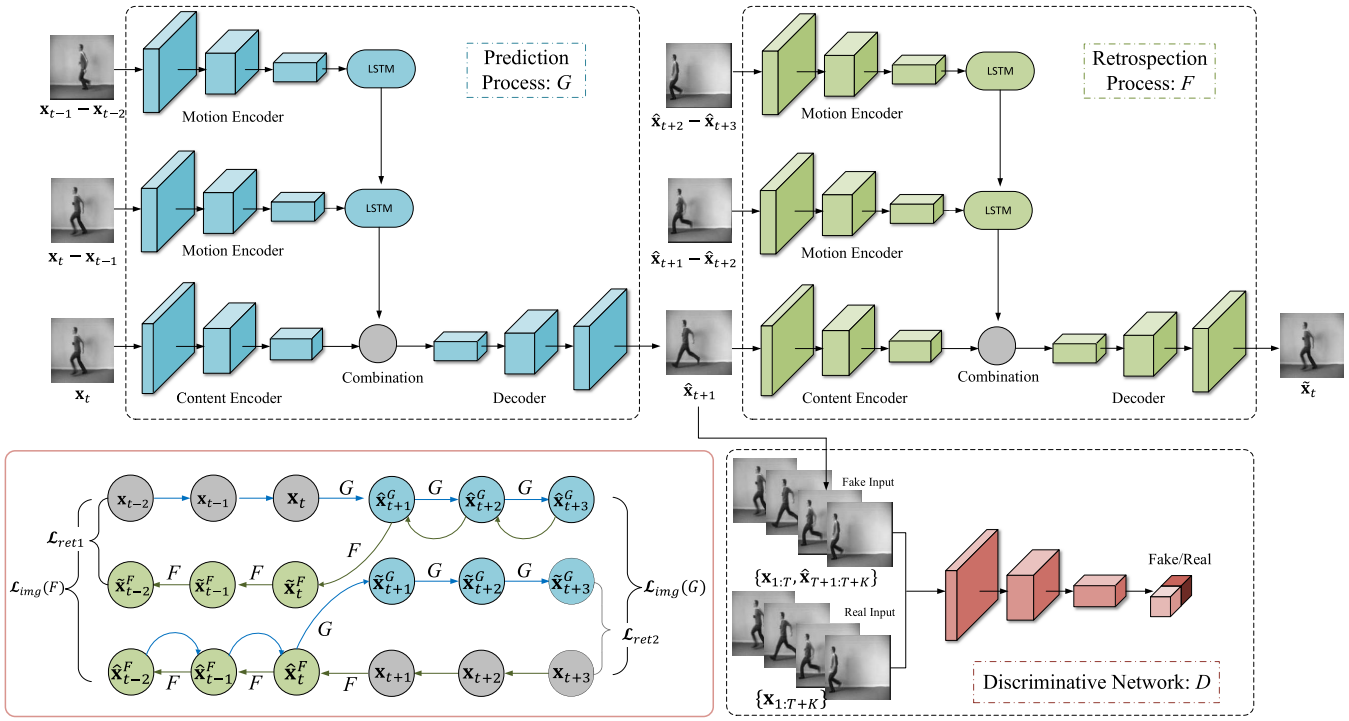

Fig. 2. The overall architecture of the proposed network. Left bottom: our model contains two routes: one is $G \rightarrow F$, and the other is $F \rightarrow G$. Right top: illustration of the route $G \rightarrow F$ in detail. In each route, our model consists of two processes. The prediction process executes recursively by taking the observations to generate subsequent frames, while the retrospection process synthesizes frames by observing the predicted frames in a backward manner. The discriminative network uses an adversarial loss to distinguish between predicted and real frames.

frames. As a result, the objective of the prediction network G is to solve:

$$arg \min_{G} \left[\mathcal{L}_{img}(G) + \beta_1 \mathcal{L}_{GAN}(G) + \gamma_1 \mathcal{L}_{ret1}(F, G)\right]. \quad (16)$$

The generative network with the image loss and adversarial loss would lead to the preliminary generation results. The introduced retrospection loss provides an auxiliary constraint to further enhance the quality of predicted frames. In addition, given a certain capacity of retrospective network, compared with a noisy input, high-quality predicted frames for the retrospection module tends to have a big chance to better reconstruct the past frames. That is to say, by minimizing the retrospection loss, the predicted frames would be further optimized to achieve higher quality. Meanwhile, the retrospection network is updated by observing the ground-truth frames in the route $F \rightarrow G$. The objective of retrospection network F is to solve:

$$arg \min_{F} \left[\mathcal{L}_{img}(F) + \beta_2 \mathcal{L}_{GAN}(F) + \gamma_2 \mathcal{L}_{ret2}(F, G)\right]. \quad (17)$$

While the discriminative network is trained to minimize the loss function Equation 7 and Equation 14 in order to distinguish between the generated frames and real frames:

$$D_g^* = arg \min_{D_g} \mathcal{L}_{\text{GAN}}(D_g), \quad (18)$$

$$D_f^* = arg \min_{D_f} \mathcal{L}_{\text{GAN}}(D_f). \quad (19)$$

The detailed training procedure is shown in Algorithm 1.

IV. Implementation

A. Network Configuration

We design the architecture for our prediction network G and retrospection networks F according to MCNET [2]. The prediction network and retrospection network are designed to share the same architecture. As shown in Figure 2, each network contains a motion encoder, a content encoder, an LSTM module, a combination layer and a decoder. The motion encoder and content encoder contain a series of convolutional layers and max pooling layers. The decoder consists of a series of deconvolution layers and unpooling layers. The unpooling layer [44] is an up-sampling operation which can be viewed as an inverse of max pooling. We adopt *tanh* function in the output layer of decoder, and the ReLU function in the rest of layers.

In addition, a discriminative network is built to facilitate more realistic outputs. The forward and backward discriminative network D_g and D_f are designed to adopt the same architecture, which consists of a series of convolutional layers. The Batch Normalization [45] and leaky-Relu [46] are used after convolutional layers. At the last layer, the sigmoid function is used. Details are listed in Table II.

1) Motion Encoder: The motion encoder of the prediction network captures the temporal dynamics of the scenes' components by recurrently observing the difference between frames $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$. Correspondingly, in the retrospection network, the difference between the $\mathbf{x}_{t+1}$ and $\mathbf{x}_{t+2}$ is fed into its motion encoder. The outputs of the motion features are filtered by a

Algorithm 1 Adversarial Training of Our Proposed Model

Require: the set of training data $\{\mathbf{x}^{(i)}_{1,\cdots,T+K}\}_{i=1}^{N}$, margin m.

for number of training iterations **do**
  Sample minibatch of sequence frames $\mathbf{x}_{1:T+K}$.
  Predict the subsequent frames $\hat{\mathbf{x}}_{T+1:T+K}$ by given the observed frames $\mathbf{x}_{1:T}$ in the prediction network $G : \mathbf{x}_{1:T} \rightarrow \hat{\mathbf{x}}^{G}_{T+1:T+K}$ in Equation 1.
  Retrospect the preceding frames $\mathbf{x}_{1:T}$ by given the predicted frames $\hat{\mathbf{x}}^{G}_{T+1:T+K}$ in the retrospection network $F : \hat{\mathbf{x}}^{G}_{T+1:T+K} \rightarrow \tilde{\mathbf{x}}^{F}_{1:T}$ in Equation 2.
  Generate the preceding frames $\tilde{\mathbf{x}}^{F}_{1:T}$ by given the ground-truth frames $\mathbf{x}_{T+K:T+1}$ in the retrospection network $F : \mathbf{x}_{T+K:T+1} \rightarrow \hat{\mathbf{x}}^{F}_{1:T}$ in Equation 9.
  Generate the subsequent frames $\tilde{\mathbf{x}}^{G}_{T+1:T+K}$ by given the generated frames $\hat{\mathbf{x}}^{F}_{1:T}$ in the network $G : \hat{\mathbf{x}}^{F}_{1:T} \rightarrow \tilde{\mathbf{x}}^{G}_{T+1:T+K}$ in Equation 11.
  if $\log(1 - D_g([\mathbf{x}_{1:T}, \hat{\mathbf{x}}^{G}_{T+1:T+K}])) > m$ and $\log(D_g(\mathbf{x}_{1:T+K})) > m$ **then**:
    Update discriminator D_g using Equation 18:
$$\Delta_{\theta_{D_g}} \leftarrow \nabla_{\theta_{D_g}} \mathcal{L}_{\text{GAN}}(D_g).$$
  end if
  if $\log(1 - D_g([\mathbf{x}_{1:T}, \hat{\mathbf{x}}^{G}_{T+1:T+K}])) < 1 - m$ and $\log(D_g(\mathbf{x}_{1:T+K})) < 1 - m$ **then**
    Update prediction network G using Equation 16:
$$\Delta_{\theta_G} \leftarrow \nabla_{\theta_G}(\alpha_1 \mathcal{L}_{img} + \beta_1 \mathcal{L}_{GAN} + \gamma_1 \mathcal{L}_{ret1}).$$
  end if
  if $\log(1 - D_f([\mathbf{x}_{T+K:T+1}, \hat{\mathbf{x}}^{F}_{T:1},])) > m$ or $\log(D_f(\mathbf{x}_{T+K:1})) > m$ **then**
    Update discriminator D_f using Equation 19:
$$\Delta_{\theta_{D_f}} \leftarrow \nabla_{\theta_{D_f}} \mathcal{L}_{\text{GAN}}(D_f).$$
  end if
  if $\log(1 - D_f([\mathbf{x}_{T+K:T+1}, \hat{\mathbf{x}}^{F}_{T:1}]))) < 1 - m$ or $\log(D_f(\mathbf{x}_{T+K:1})) < 1 - m$ **then**
    Update retrospection network F using Equation 17:
$$\Delta_{\theta_F} \leftarrow \nabla_{\theta_F}(\alpha_2 \mathcal{L}_{img} + \beta_2 \mathcal{L}_{GAN} + \gamma_2 \mathcal{L}_{ret2}).$$
  end if
end for

TABLE II
ARCHITECTURE OF DISCRIMINATIVE NETWORK

	Operation	Kernel size	Stride	Padding	Feature maps	Normalization	Nonlinearity
Discriminative Network	Convolution	5	2	3	64	-	LeakyReLU
	Convolution	5	2	3	128	Batch Normalization	LeakyReLU
	Convolution	5	2	3	256	Batch Normalization	LeakyReLU
	Convolution	5	2	3	512	Batch Normalization	LeakyReLU
	Convolution	5	1	2	1	-	Sigmoid
LeakyReLU: Slope 0.2							

series of convolutional layers. The architecture is similar to VGG16 [47] up to the third pooling layer, except that the consecutive 3×3 convolutions are replaced by single 5×5, 5×5, and 7×7 convolutions in each layer.

2) Content Encoder: The content encoder extracts important spatial features from a single frame, such as the spatial layout of the scene and salient objects in a video. In our experiment, it takes the last observed frame $\mathbf{x}_t$ as input in the prediction process, while it takes the first predicted frame $\hat{\mathbf{x}}_{t+1}$ as input in the retrospection process. The content encoder is also built with the same architecture as VGG16 [47] up to the third pooling layer.

3) Decoder Module: The decoder is used to generate a pixel-level prediction of the next frames in the prediction process or the past frames in the retrospection process. To reduce the information loss caused by pooling at the encoding phase, a residual connection [28] is adopted by taking the computed features from both the motion and content encoders. We employ deconvolution network [48] for the decoder module. The architecture of the decoder is the mirror of the content encoder. The output layer is passed through a $\tanh(\cdot)$ activation function.

4) Discriminative Network: The discriminative network consists of 4 convolutional layers with kernel size 5×5 and stride 2, whose output channels are 64, 128, 256, 512 respectively. From the second layer, each layer is followed by a batch normalization layer [49]. After that, each layer is followed by a leaky ReLU layer [50]. The final layer is a fully-connected layer with 1 hidden unit followed by a sigmoid function, so as to predict the possibility of input's

TABLE III
ARCHITECTURE OF THE PREDICTION AND RETROSPECTION NETWORK

Module	Operation	Kernel	Stride	Padding	Features
Motion encoder	Convolution	5	1	2	64
	MaxPooling	2	2	-	64
	Convolution	5	1	2	128
	MaxPooling	2	2	-	128
	Convolution	7	1	3	256
	MaxPooling	2	2	-	256
Content Encoder	Convolution	3	1	1	64
	Convolution	3	1	1	64
	MaxPooling	2	2	-	64
	Convolution	3	1	1	128
	Convolution	3	1	1	128
	MaxPooling	2	2	-	128
	Convolution	3	1	1	256
	Convolution	3	1	1	256
	Convolution	3	1	1	256
	MaxPooling	2	2	-	256
Decoder	Unpooling	2	2	-	256
	Deconvolution	3	1	1	256
	Deconvolution	3	1	1	256
	Deconvolution	3	1	1	128
	Unpooling	2	2	-	128
	Deconvolution	3	1	1	128
	Deconvolution	3	1	1	128
	Deconvolution	3	1	1	64
	Unpooling	2	2	-	64
	Deconvolution	3	1	1	64
	Deconvolution	3	1	1	1

label. Note that we concatenate the frames of both observation and prediction along the channel of time as the inputs.

The LSTM module predicts or retrospects the dynamics of frames in a chronological or reversed order recursively. We adopt a convolutional LSTM [21] as our LSTM module. The combination module is used to fuse the content representation and motion representation, which consists of 3 consecutive 3×3 convolutions (256, 128, and 256 channels in each layer).

B. Training Strategy

All networks are trained by the Adam optimization [51] for 100,000 iterations with the learning rate of 0.0001, the exponential decay rate for the first moment of 0.5 and the exponential decay rate for the second moment estimates of 0.999. To stabilize training, we set a margin to balance the adversarial training of generative network and discriminative model. Following MCNET [2], the loss margin m is set to be 0.3.

Algorithm 1 shows the detailed training procedure. In the training procedure, the generative network G and retrospective network F are trained alternatively. The route $G \rightarrow F$ is used to optimize the generative network G, and route $F \rightarrow G$ is used to optimize the retrospective network. During training each network, we assume that the other network has been optimized. For instance, the retrospective network F is well optimized to generated realistic and sharp frames by the $\mathcal{L}_{GAN}(F)$. As a result, as long as the retrospection is pixel-wise consistent to the input frames, the error of predicted frames could be rectified. A similar idea has been applied in CycleGAN [33]. In CycleGAN [33], they train two generative networks in two directions, and the cycle-consistent loss is adopted in each direction.

V. EXPERIMENTS

In this section, we present experiments using our model for long-term video prediction. We first evaluate our model on the KTH [19] and Weizmann datasets [11]. We then proceed to evaluate on a more challenging dataset, UCF-101 [12]. We compare our model against MCNET [2], which achieves state-of-the-art performance on the KTH [19], Weizmann [11] and UCF101 datasets [12]. For all our experiments, we use $\lambda = 1$, and $p = 2$ in the loss function. We train our network by observing 10 frames and predicting 10 subsequent frames. For fairness, the prediction network of our model is adopted the same architecture of MCNET [2]. As the retrospection process would be discarded in the test phase, it costs the same computation complexity in the test phase. We demonstrate the quantitative comparisons in terms of PSNR, SSIM and LPIPS [52]. PSNR and SSIM are commonly used in previous works [2]. However, they are shallow functions, and not necessarily coincide with human perception. Recently, perceptual metrics are proposed by adopting the deep features in neural networks, such as LPIPS (Learned Perceptual Image Patch Similarity) [52] and FVD [53]. To partially mitigate the limitations of these metrics, we evaluate on LPIPS as well. In addition, we used a person detection evaluation to test the quality of generation frames on person action dataset, with the idea that acceptable predictions should contain a recognizable person. Meanwhile, we take two variants of our method for ablation study, details are presented in Section V-C.

A. KTH and Weizmann Action Datasets

1) Experimental Setting: The KTH human action dataset [19] contains 6 categories of periodic motions on a simple background: running, jogging, walking, boxing, hand-clapping, and hand-waving. Following MCNET [2], we used person 1-16 for training and 17-25 for testing, and also resize frames to 128×128 pixels. During the evaluation, to demonstrate the effectiveness of our proposed long-term model, we predicted 100 subsequent frames given the previous 10 frames as input. We also selected the walking, running, one-hand waving, and two-hands waving sequences from the Weizmann action dataset [11] to verify the networks' generalization. Since most of the videos in the Weizmann dataset only contain around 70-80 frames, we test our model to predict 70 future frames on the Weizmann dataset.

2) Results Analysis: Figure 5 presents qualitative results of long-term prediction by our model, MCNET and SVG [41] on the KTH dataset. Compared to MCNET, prediction results by our full objective preserve a better quality with time passing. Our method could achieve more satisfying results at 100-time steps, while MCNET's results tend to collapse to meaningless noise (see the fourth column in each case at $t = 100$). In action of boxing, we notice that at $t = 40$, the output of MCNET appears some noisy pattern besides the person's feet. With the increase of time step, the region of noise becomes larger and eventually dominates the whole

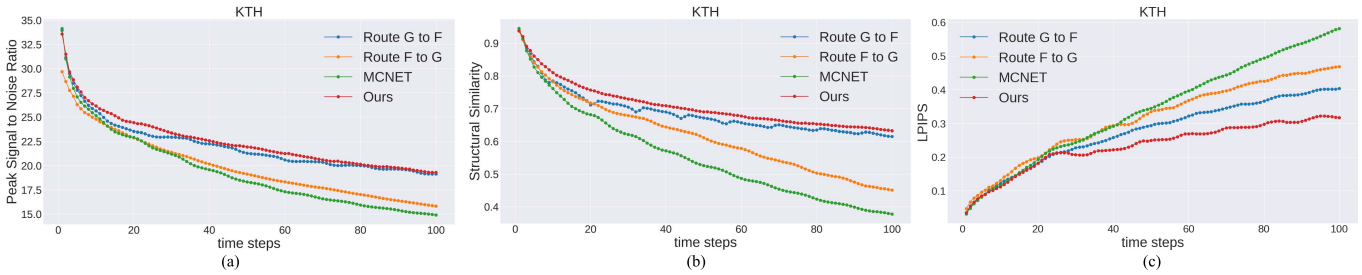

Fig. 3. Quantitative comparisons between different variants of our method and MCNET baseline in terms of PSNR, SSIM and LPIPS on the KTH dataset. "Ours" denotes our method (MCNET+Retrospection) with full objective. "Route F to G" represents Route $F \rightarrow G$ alone (Equation 20). "Route G to F" indicates Route $G \rightarrow F$ alone (Equation 21). Given 10 input frames, the models predict 100 frames recursively. For PSNR and SSIM, higher is better. For LPIPS, lower is better.

images. In testing of hand-waving action, we find that MCNET is able to achieve a competitive result in the first several time steps. However, with the time passing, the prediction deviation is accumulated whose images appear noisier. Compared to SVG [41], though SVG could output unblurred and realistic frames for a long-term horizon, we notice that after a few times step, the motion information in the SVG might be lost or incorrect. For instance, the generated person keeps a fixed gesture after t=30. Our method with retrospection loss is able to output consistent high-quality results for a long time.

Figure 3 summarizes the quantitative comparisons of our methods, MCNET, and two variants of our model. In the KTH test set (Figure 3), our models and variants outperform the MCNET baseline in long-term video prediction. Although the four methods achieve comparable LPIPS scores for the first 20 future frames, with the increase of time step, the margin between our method (blue line) and MCNET (red line) becomes more significant. One reason for this result is that the baseline method only considers the prediction recursively in the forward time steps, which is easy to accumulate the prediction error and leads to dramatic degradation of prediction performance. Also, we tested on unseen data of Weizmann dataset [11] by the pre-trained model. Figure 4 reveals that our method outperforms MCNET, especially on the long-term horizon.

3) Person Detection Evaluation: We used the person detection evaluation with the idea that acceptable predictions should contain a recognizable person. Similar to [28], we used the pre-trained Mask-RCNN[1] as the person detector, and calculated the percentage of frames that a recognizable person is in the image. We record the person recognition rate of Mask-RCNN in Figure 6. The proposed method stays relatively constant over prediction time-steps. For longer-term predictions, e.g., at $t = 100$, more than 80% of our generated frames could be recognized as a person, while the performance of MCNET drops to only 65% of predictions can be recognized. The evaluation shows that the proposed method is better than the baselines on the long-term horizon.

B. UCF-101 Dataset

This section presents results on more challenging real-world videos dataset, the UCF-101 dataset [12]. Collected from

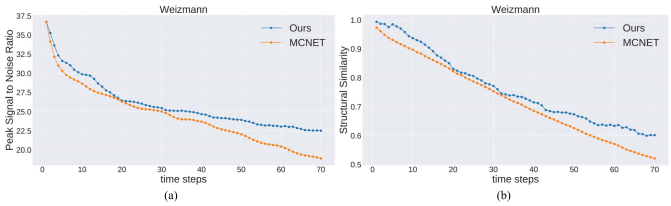

Fig. 4. Quantitative comparisons between our method and MCNET in terms of PSNR, SSIM on the Weizmann dataset.

YouTube, the dataset contains 101 realistic human actions taken in a wild and exhibits various challenges, such as background clutter, occlusion, and complicated motion. The videos in 101 action categories are grouped into 25 groups, where each group consists of 4-7 videos of an action. We use group 1-7 for testing and group 8-25 for training. We employed the same network architecture as in the KTH dataset but resized frames to 240 $\times$ 320 pixels.

1) Results Analysis: Figure 9 shows the quantitative comparisons between our model and MCNET. We test the official-released pre-trained model [2](denoted as "MCNET"), which is trained to predict one future frame by observing four frames. We found that only predicting one future frame is easily to miss the motion pattern of the video and leads to stationary output. As a result, the performance in terms of PSRN, SSIM and LPIPS drops with the increase of time (see blue line).

For fairness, we also compare with the MCNET [2] which is trained to predict 10 frames by given 10 input frames (denoted as "MCNET-T10" in orange line). We observe that our model still outperforms the compared method and the gap becomes more significant with the increase of predicted time. Figure 8 presents qualitative comparisons between frames generated by our method, MCNET and MCNET-10T. We observe that the results of MCNET is becoming blur while ours are still recognizable (see details in the zoomed region). In addition, the quality of MCNET's results falls dramatically with time passing, e.g., at $t = 100$ in the second row. Results of MCNET-T10 is better compared with MCNET, while we still could find some green noisy patches. Our method, on the other hand, could maintain structural outputs with reasonable dynamic.

[1]https://github.com/matterport/Mask_RCNN

[2]https://github.com/rubenvillegas/iclr2017mcnet

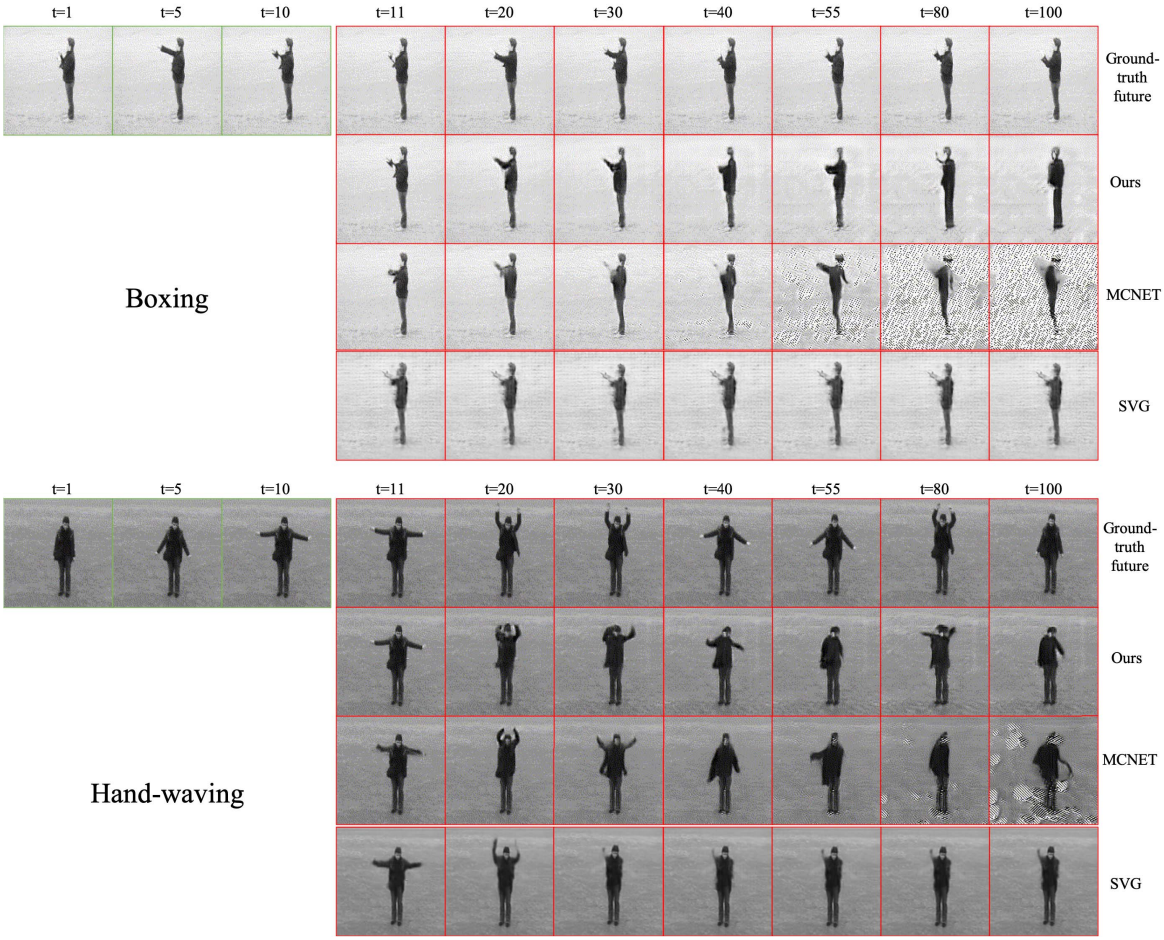

Fig. 5. Qualitative results of our method, MCNET, and SVG on the KTH dataset. The top case corresponds to the action of boxing, and the lower case corresponds to the action of hand-waving.

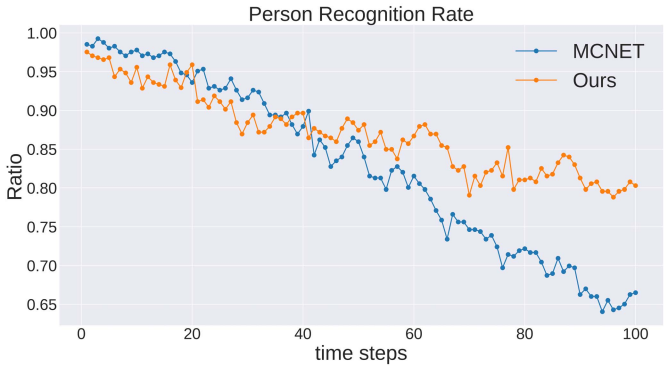

Fig. 6. Comparison with MCNET [2] in terms of the recognition rate. The recognition rate of the person detector that a person is recognized in the predicted frame.

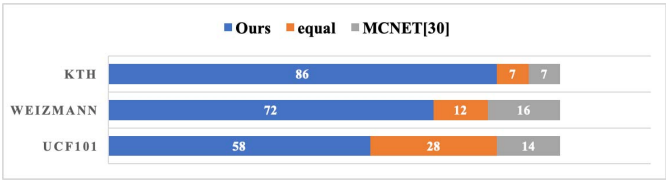

Fig. 7. The stacked bar chart of participants preferences for our method compared to MCNET [2]. The blue bar indicates the number of videos that more participants prefer our results. The gray bar indicates the number of videos that more participants prefer MCNET's results. The orange bar indicates the number of videos where two methods get a equal number of votes.

2) Human Perceptual Study: We further evaluate our algorithm via human study. We perform pairwise A/B tests deployed on a service similar to Mechanical Turk. We follow the experiment procedure in [3]. The participants are asked to select the more realistic video generated from our method and MCNET. Each pair contains two video predictions observed by the same input frames generated. We use the data sets of KTH, Weizmann and UCF101 [12]. In each data set, we collected comparisons of 100 generated videos, and presented each video to 10 human raters. The table in Figure 7 shows the video number of the most users' preference. It demonstrates that most users prefer our results, which indicates that the qualitative performance of our method is better than MCNET.

C. Model Analysis

1) Ablation Study: We take two variants of our method for ablation study. The first variant is denoted as "Route $F \rightarrow G$". In this variant, we first train a retrospection network and

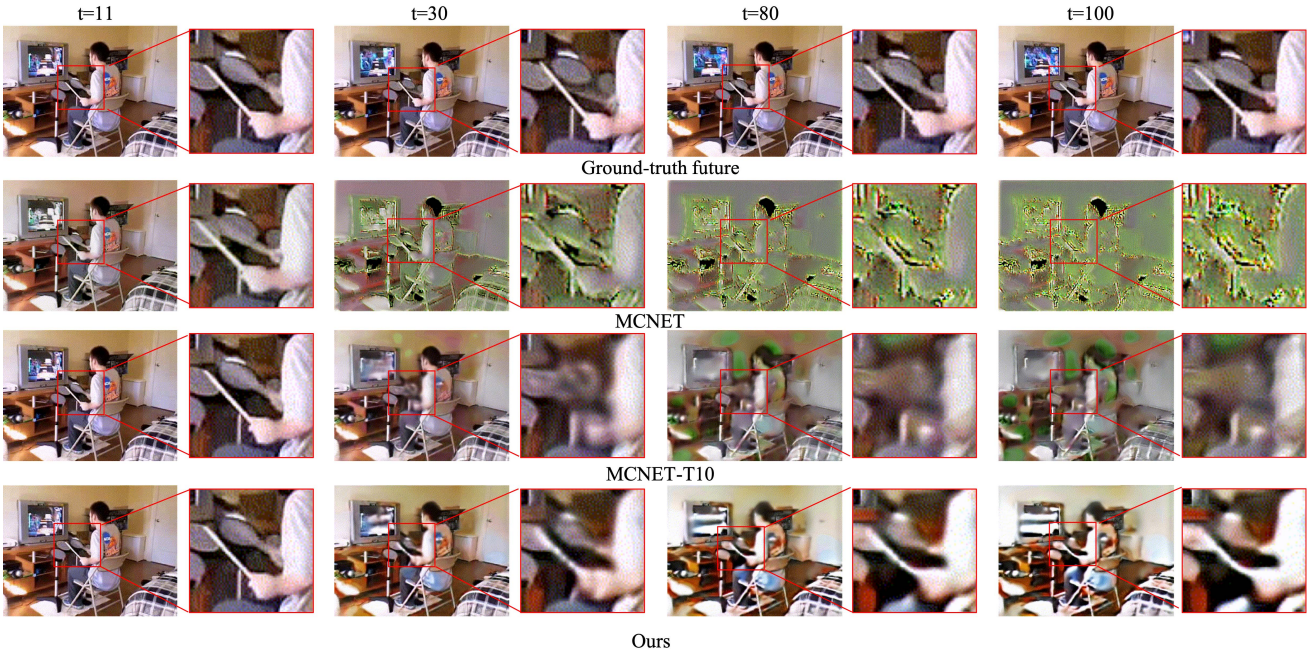

Fig. 8. Qualitative comparison on the UCF-101 dataset. MCNET [2] is trained to optimize one future frame. "MCNET-T10" is trained to optimize 10 future frames. Our method less artifact and blur around the ambiguity region. The remarkable region is denoted in color and scaled.

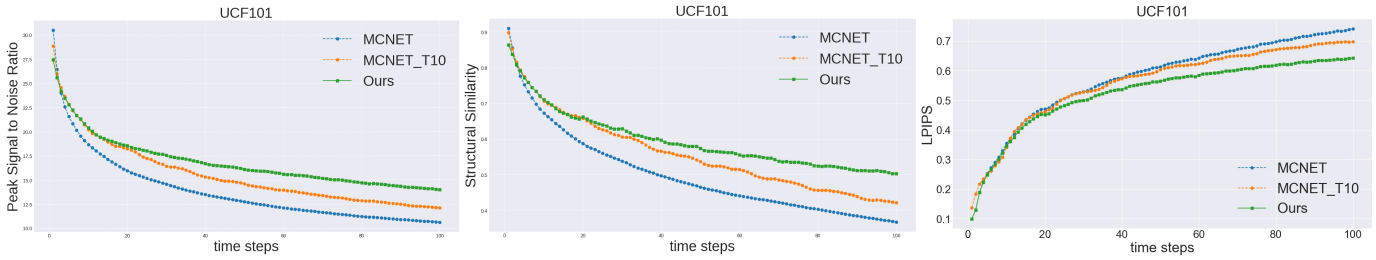

Fig. 9. Quantitative comparisons of our model, MCNET [2] and MCNET trained by 10 input frame and 10 output frames (indicated by "MCNET-T10"). In the test phase, the models predict 100 frames recursively given 10 input frames.

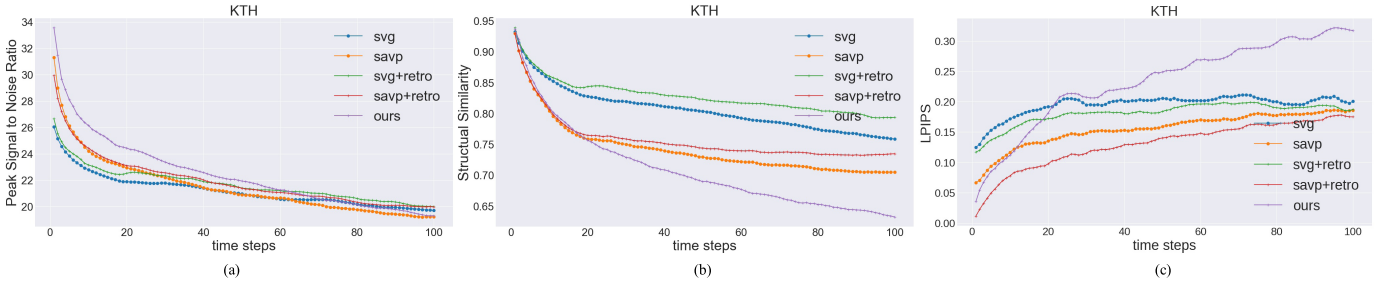

Fig. 10. Quantitative comparisons of our method (MCNET+retro), SAVG, SVG w/o the retrospection process in terms of PSNR, SSIM and LPIPS.

prediction network by solving:

$$\arg\min_{G,F}\left[\alpha_2\mathcal{L}_{img}(F)+\beta_2\mathcal{L}_{GAN}(F)+\gamma_2\mathcal{L}_{ret2}(F,G)\right]. \quad (20)$$

The prediction is obtained by feeding the test frames to the trained prediction network.

The second variant is denoted as "route $G \rightarrow F$". In this variant, we simultaneously train the prediction and retrospection network by removing the auxiliary route $F \rightarrow G$:

$$\arg\min_{G,F}\left[\alpha_1\mathcal{L}_{img}(G)+\beta_1\mathcal{L}_{GAN}(G)+\gamma_1\mathcal{L}_{ret1}(F,G)\right], \quad (21)$$

where α_1, β_2 and γ_1 are set to be the same as the full objective function. Table IV presents results of ablation study on the KTH dataset. We observe that our model and its variants with retrospection process outperform the MCNET baseline. In "route $G \rightarrow F$", the generative network is optimized by the prediction loss $\mathcal{L}_{img}(G)$, pixel adversarial loss $\mathcal{L}_{GAN}(G)$, and rectified by the retrospection loss $\mathcal{L}_{ret1}(F,G)$. The retrospection loss enforces the predictions to reconstruct the original input. As a result, the performance is close to the two routes version. In our full model, an auxiliary route "route $F \rightarrow G$", is designed to reduce the accumulation error produced from F. In Figure 3, we observe that our method still outperforms the "route $G \rightarrow F$", which indicates a better retrospective network F is able to improve the accuracy of prediction. In "route $F \rightarrow G$", the generative network G only optimizes

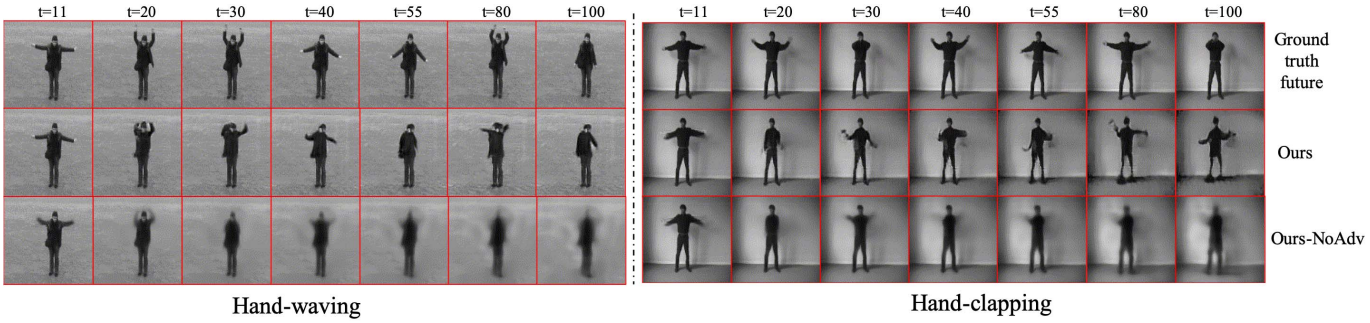

Fig. 11. Qualitative analysis for the function of the adversarial loss. First Row: ground-truth frames; Second Row: our results with full objective; Third Row: results without adversarial loss.

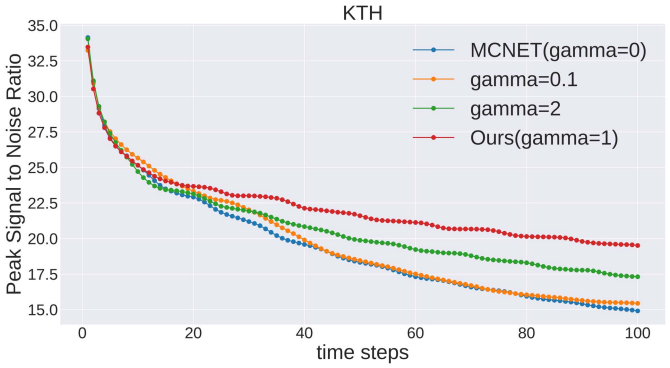

Fig. 12. Quantitative comparisons of the retrospection loss with different parameter γ.

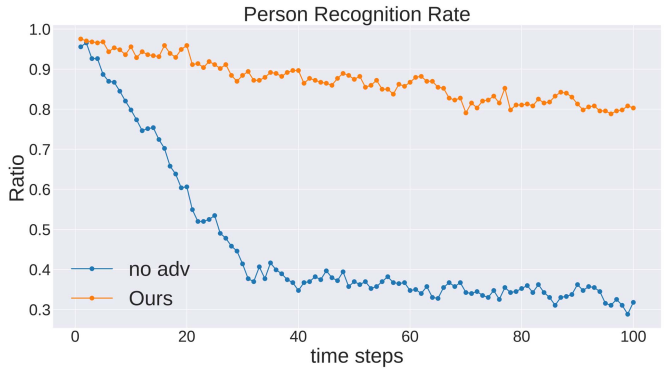

Fig. 13. Ablation study for the function of the adversarial loss in terms of the person recognition rate. The person recognition rate denotes the ratios of the predicted frames that can be recognized as a person.

the reconstruction loss: $\arg\min_G \mathcal{L}_{ret2}(F, G)$, whose aim is to generate the future frames $\{\tilde{x}^G_{T+1}, \cdots, \tilde{x}^G_{T+K}\}$ from the previous frames $\{\hat{x}^F_1, \cdots, \hat{x}^F_T\}$. That is to say, the generative network G in this route does not get the benefit from the retrospective process. As the results shown in Figure 3, the performance of "route $F \rightarrow G$" is close to the baseline (i.e., MCNET). Overall, the proposed training strategy of two routes achieves the best result.

2) Parameter Selection: We discuss our parameter selection strategy in this subsection. Considering Route $G \rightarrow F$ and Route $F \rightarrow G$ are symmetrical, we set $\alpha_1 = \alpha_2$, $\beta_1 = \beta_2$, and

TABLE IV
ABLATION STUDY OF THE PROPOSED MODEL

Model	PSNR	SSIM	LPIPS
MCNET (Only G)	19.42	0.5541	0.3410
Route $F \rightarrow G$	19.92	0.6186	0.3136
Route $G \rightarrow F$	22.05	0.6881	0.3749
Two routes (Ours)	**22.48**	**0.7092**	**0.2336**

$\gamma_1 = \gamma_2$. This selection strategy is similar to CycleGAN [33], which uses the same parameters for the same functionality loss (e.g., cycle-consistent loss) in the two translation directions. In our model, α_1 (α_2) and β_1 (β_2) balance the importance of image reconstruction loss and adversarial loss. We follow MCNET [2] to set these hyper-parameters as $\alpha_1 = \alpha_2 = 1$ in all experiments, $\beta_1 = \beta_2 = 0.02$ on KTH dataset, and $\beta_1 = \beta_2 = 0.001$ on UCF101 dataset.

We analyze the function of the newly introduced parameter of retrospection loss γ_1 and γ_2. When $\gamma_1(\gamma_2) = 0$, our model are equivalent to MCNET. Figure 12 shows results of $\gamma_1(\gamma_2) = \{0.1, 1, 2\}$. We found that $\gamma_1(\gamma_2) = 0.1$ is too small and the performance drops to be similar to MCNET. $\gamma_1(\gamma_2) = 2$ is too large that the model emphasizes too much on the auxiliary network and suppresses the prediction accuracy. In all, $\gamma_1(\gamma_2) = 1$ achieves the best performance, so we set $\gamma_1(\gamma_2) = 1$ in our model.

3) Adversarial Loss: We explore the contribution of the adversarial loss in our model. We analyze the results of the function without the adversarial loss. The function without the adversarial loss can be expressed as follow:

$$\mathcal{L}_{no\_adv}(G, F) = \alpha_1 \mathcal{L}_{img}(G) + \alpha_2 \mathcal{L}_{img}(F) + \gamma_1 \mathcal{L}_{ret1}(F, G) + \gamma_2 \mathcal{L}_{ret2}(F, G). \quad (22)$$

For a fair comparison, we set $\alpha_1 = \alpha_2 = 1$ and $\gamma_1 = \gamma_2 = 1$, that are the same with our model. Results are demonstrated in Figure 11. Results of Equation 22 are denoted as "Ours-No Adv". We observed that the prediction frames are blurry and hard to be recognized. We also test the person recognition rate. As shown in Figure 13, the model without adversarial loss results in a dramatic decrease in terms of person recognition rate. At t=[50,100], it turns out that only 34.5% generated frames could be recognized as a person. In comparison, our method with adversarial losses produces 83.0% predictions

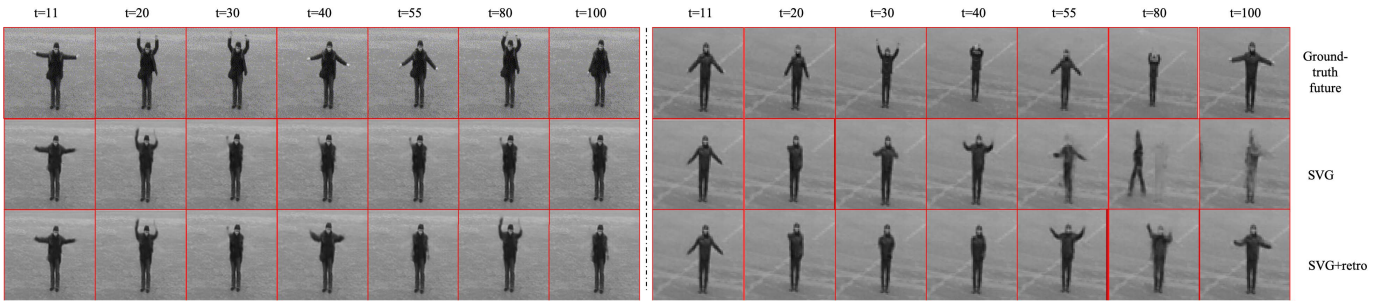

Fig. 14. Qualitative comparison between SVG and "SVG+retro" on the action of hand-waving. "SVG+retro" indicates the method of SVG incorporating the retrospection process.

with recognizable persons. The reason is that $\mathcal{L}_{img}$ guides the model to match the average pixel of all possible future frames. The adversarial loss, in contrast, allows our model to predict realistic looking frames.

VI. Extension to Other Video Prediction Methods

In this section, we discuss the generation of the proposed retrospection process for other video prediction methods. In addition to applying the retrospective training procedure to MCNET, we compare our model to SAVP [40] and SVG [41], and extend those methods by incorporating the proposed retrospective training procedure. While MCNET is a simple framework that outputs deterministic future frames, SAVP [40] and SVG [41] are stochastic video prediction methods that output samples from the possible future distribution. The stochastic video prediction methods are considered as a promising solution to generate realistic and sharp future frames.

For a fair comparison, we train the SVG, SAVP model by using the same dataset and image size to our model. Specifically, we used person 1-16 for training and 17-25 for testing, and the image size is 128×128. The model is trained by observing 10 frames and predicting 10 subsequent frames. The code is from authors' homepages[3] .[4] In the test phase, we predicted 100 subsequent frames given the previous 10 frames as input. By incorporating the retrospection processing, we train a retrospective generator F to generate preceding frames by feeding the true future frames. Then we retrain the prediction network G by adding the retrospection loss.

Figure 14 demonstrates the qualitative comparison between SVG and SVG by incorporating the retrospection process (denoted as "SVG+retro"). We notice that after a few times step, the motion information in the SVG might be lost or incorrect. In constrast, results of SVG with the retrospection process shows that the motion information can be preserved. For instance, in the right cases of Figure 14, the generated person of "SVG+retro" keeps hand-waving all the time, while the person of SVG becomes the action of walking after $t = 80$. The reason is that the introduced retrospection process tends to keep the action of prediction consistent with that in the past frames.

[3]https://github.com/alexlee-gk/video_prediction
[4]https://github.com/edenton/svg

Results in Figure 10 shows the quantitative comparisons between our method (i.e., MCNET+retro), SVG, SAVP, "SVG+retro" and "SAVP+retro". We observe that the performance of "SVG+retro" and "SAVP+retro" outperforms the "SVG" and "SAVP+retro", which demonstrates that our method can improve the stochastic based video prediction method as well. We also notice that our method (MCNET+retro) outperforms the compared method in terms of PSNR We think the reason is that MCNET is based on the deterministic prediction so that our results can get close to the ground truth future frames. However, our method (i.e., MCNET+retro) gets a lower score in terms of perceptual metrics (e.g., SSIM and LPIPS), which indicates that the stochastic-based methods achieve more realistic video frames in terms of human perception.

VII. Conclusion

We have proposed a long-term video prediction model via criticization and retrospection. Our model consists of a prediction network and a retrospection network. The prediction network is used to generate future frames given a set of consecutive frames. The retrospection network aims to look back to rectify the prediction deficiencies. In addition to considering the discrepancy between the predicted frames and ground truth frames, we feed the predicted frames to the retrospection network to minimize the discrepancy between the retrospective frames and the observed frames. To optimize the prediction network and the retrospection network, an auxiliary route is built by reversing the flow of time and executing a similar way. Our method is able to alleviate the accumulation deviation during the recursive prediction process. Extensive experiments demonstrate the effectiveness of the proposed model on long-term video prediction.

References

[1] X. Liang, L. Lee, W. Dai, and E. P. Xing, "Dual motion GAN for future-flow embedded video prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 1762–1770, doi: 10.1109/ICCV.2017.194.

[2] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, Apr. 2017, pp. 1–22.

[3] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," in *Proc. 34th Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, Aug. 2017, pp. 3560–3569. [Online]. Available: http://proceedings.mlr.press/v70/villegas17a.html

[4] J. Xu, B. Ni, Z. Li, S. Cheng, and X. Yang, "Structure preserving video prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1460–1469.

[5] C. Finn, I. J. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 64–72. [Online]. Available: http://papers.nips.cc/paper/6161-unsupervised-learning-for-physical-interaction-through-video-prediction

[6] S. Chiappa, S. Racaniere, D. Wierstra, and S. Mohamed, "Recurrent environment simulators," 2017, *arXiv:1704.02254*. [Online]. Available: http://arxiv.org/abs/1704.02254

[7] J. Oh, X. Guo, H. Lee, R. L. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in atari games," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2863–2871.

[8] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets

[9] E. L. Denton and V. Birodkar, "Unsupervised learning of disentangled representations from video," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 4417–4426.

[10] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, Cambridge, U.K., Aug. 2004, pp. 32–36, doi: 10.1109/ICPR.2004.1334462.

[11] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007, doi: 10.1109/TPAMI.2007.70711.

[12] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, pp. 1–7, Nov. 2012. [Online]. Available: http://arxiv.org/abs/1212.0402

[13] W. Byeon, Q. Wang, R. Kumar Srivastava, and P. Koumoutsakos, "Contextvp: Fully context-aware video prediction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 753–769.

[14] N. Kalchbrenner *et al.*, "Video pixel networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1771–1779.

[15] W. Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," 2016, *arXiv:1605.08104*. [Online]. Available: http://arxiv.org/abs/1605.08104

[16] T. Xue, J. Wu, K. Bouman, and B. Freeman, "Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 91–99.

[17] I. Sutskever, G. E. Hinton, and G. W. Taylor, "The recurrent temporal restricted Boltzmann machine," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2008, pp. 1601–1608.

[18] R. Mittelman, B. Kuipers, S. Savarese, and H. Lee, "Structured recurrent temporal restricted Boltzmann machines," in *Proc. 31th Int. Conf. Mach. Learn. (ICML)*, Beijing, China, Jun. 2014, pp. 1647–1655.

[19] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using LSTMS," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, Lille, France, Jul. 2015, pp. 843–852. [Online]. Available: http://jmlr.org/proceedings/papers/v37/srivastava15.html

[20] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra, "Video (language) modeling: A baseline for generative models of natural videos," 2014, *arXiv:1412.6604*. [Online]. Available: http://arxiv.org/abs/1412.6604

[21] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst*, Montreal, QC, Canada, 2015, pp. 802–810.

[22] N. Ballas, L. Yao, C. Pal, and A. C. Courville, "Delving deeper into convolutional networks for learning video representations," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, San Juan, Puerto Rico, May 2016, pp. 1–11.

[23] V. Patraucean, A. Handa, and R. Cipolla, "Spatio-temporal video autoencoder with differentiable memory," *CoRR*, vol. abs/1511.06309, pp. 1–13, Nov. 2015. [Online]. Available: http://arxiv.org/abs/1511.06309

[24] K. Zhang, W. Luo, Y. Zhong, L. Ma, W. Liu, and H. Li, "Adversarial spatio-temporal learning for video deblurring," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 291–301, Jan. 2019.

[25] T. Marwah, G. Mittal, and V. N. Balasubramanian, "Attentive semantic video generation using captions," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 1435–1443.

[26] J. S. Yoon, F. Rameau, J. Kim, S. Lee, S. Shin, and I. S. Kweon, "Pixel-level matching for video object segmentation using convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2186–2195.

[27] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, "Stochastic variational video prediction," 2017, *arXiv:1710.11252*. [Online]. Available: http://arxiv.org/abs/1710.11252

[28] N. Wichers, R. Villegas, D. Erhan, and H. Lee, "Hierarchical long-term video prediction without supervision," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, Stockholm, Sweden, Jul. 2018, pp. 6033–6041.

[29] R. Mahjourian, M. Wicke, and A. Angelova, "Geometry-based next frame prediction from monocular video," in *Proc. IEEE Intell. Vehicles Symp.*, Los Angeles, CA, USA, Jun. 2017, pp. 1700–1707.

[30] M. Liu, X. He, and M. Salzmann, "Geometry-aware deep network for single-image novel view synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 4616–4624, doi: 10.1109/CVPR.2018.00485.

[31] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434

[32] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," 2018, *arXiv:1809.11096*. [Online]. Available: http://arxiv.org/abs/1809.11096

[33] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[34] Y. Li, S. Tang, R. Zhang, Y. Zhang, J. Li, and S. Yan, "Asymmetric GAN for unpaired image-to-image translation," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5881–5896, Dec. 2019.

[35] X. Chen, C. Xu, X. Yang, L. Song, and D. Tao, "Gated-GAN: Adversarial gated networks for multi-collection style transfer," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 546–560, Feb. 2019.

[36] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," 2016, *arXiv:1605.05396*. [Online]. Available: http://arxiv.org/abs/1605.05396

[37] M. Mathieu, C. Couprie, and Y. Lecun, "Deep multi-scale video prediction beyond mean square error," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, San Juan, Puerto Rico, May 2016, pp. 1–14.

[38] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection–A new baseline," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6536–6545.

[39] A. Dosovitskiy *et al.*, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[40] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic adversarial video prediction," 2018, *arXiv:1804.01523*. [Online]. Available: http://arxiv.org/abs/1804.01523

[41] E. Denton and R. Fergus, "Stochastic video generation with a learned prior," 2018, *arXiv:1802.07687*. [Online]. Available: http://arxiv.org/abs/1802.07687

[42] W. Lotter, G. Kreiman, and D. D. Cox, "Unsupervised learning of visual structure using predictive generative networks," *CoRR*, vol. abs/1511.06380, pp. 1–12, Nov. 2015. [Online]. Available: http://arxiv.org/abs/1511.06380

[43] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*. [Online]. Available: http://arxiv.org/abs/1411.1784

[44] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Zürich, Switzerland: Springer, 2014, pp. 818–833.

[45] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[46] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, 2013, vol. 30, no. 1, p. 3.

[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–14.

[48] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proc. Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 2018–2025, doi: 10.1109/ICCV.2011.6126474.

[49] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, Lille, France, Jul. 2015, pp. 448–456.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1026–1034.

[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–15.

[52] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[53] C. Chan, S. Ginosar, T. Zhou, and A. Efros, "Everybody dance now," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5933–5942.

Xinyuan Chen received the B.S. degree from Xidian University, China, in 2014. She is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, Shanghai Jiao Tong University, China. In 2017, she joined Center for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia, for a dual Ph.D. Program. Her research interests include computer vision, machine learning, image processing, and video processing.

Chang Xu (Member, IEEE) received the Ph.D. degree from Peking University, China. He is currently a Lecturer and an ARC DECRA Fellow with the School of Computer Science, The University of Sydney. He has published over 80 papers in prestigious journals and top-tier conferences, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, ICML, NIPS, IJCAI, and AAAI. His research interests include machine learning algorithms and related applications in computer vision. He has received several best paper awards, including the Distinguished Paper Award at IJCAI 2018. He regularly serves as the (senior) PC for many conferences, e.g., NIPS, ICML, CVPR, ICCV, IJCAI, and AAAI. He has been recognized as a Top Ten Distinguished Senior PC in IJCAI 2017.

Xiaokang Yang (Fellow, IEEE) received the B.S. degree from Xiamen University, Xiamen, China, in 1994, the M.S. degree from the Chinese Academy of Sciences, Shanghai, China, in 1997, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, in 2000. From September 2000 to March 2002, he worked as a Research Fellow with Centre for Signal Processing, Nanyang Technological University, Singapore. From April 2002 to October 2004, he was a Research Scientist with Institute for Infocomm Research (I2R), Singapore. From August 2007 to July 2008, he visited the Institute for Computer Science, University of Freiburg, Germany, as an Alexander von Humboldt Research Fellow. He is currently a Distinguished Professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University. He has published over 200 refereed articles and has filed 60 patents. His current research interests include image processing and communication, computer vision, and machine learning. He is an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA and a Senior Associate Editor of the IEEE SIGNAL PROCESSING LETTERS.

Dacheng Tao (Fellow, IEEE) is Professor of computer science and an ARC Laureate Fellow with the School of Computer Science and the Faculty of Engineering and the Inaugural Director of the UBTECH Sydney Artificial Intelligence Centre, The University of Sydney. His research results in artificial intelligence have expounded in one monograph and 200+ publications at prestigious journals and prominent conferences, such as the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IJCV, JMLR, AIJ, AAAI, IJCAI, NeurIPS, ICML, CVPR, ICCV, ECCV, ICDM, and KDD, with several best paper awards. He is a Fellow of the AAAS, ACM and the Australian Academy of Science. He received the 2018 IEEE ICDM Research Contributions Award and the 2015 Australian Museum Scopus-Eureka Prize.