Predicting Future Frames using Retrospective Cycle GAN

Yong-Hoon Kwon
Advanced Camera Lab, LG Electronics, Korea
yonghoon.kwon@lge.com

Min-Gyu Park
Korea Electronics Technology Institute
mpark@keti.re.kr

Abstract

Recent advances in deep learning have significantly improved the performance of video prediction, however, top-performing algorithms start to generate blurry predictions as they attempt to predict farther future frames. In this paper, we propose a unified generative adversarial network for predicting accurate and temporally consistent future frames over time, even in a challenging environment. The key idea is to train a single generator that can predict both future and past frames while enforcing the consistency of bi-directional prediction using the retrospective cycle constraints. Moreover, we employ two discriminators not only to identify fake frames but also to distinguish fake contained image sequences from the real sequence. The latter discriminator, the sequence discriminator, plays a crucial role in predicting temporally consistent future frames. We experimentally verify the proposed framework using various real-world videos captured by car-mounted cameras, surveillance cameras, and arbitrary devices with state-of-the-art methods.

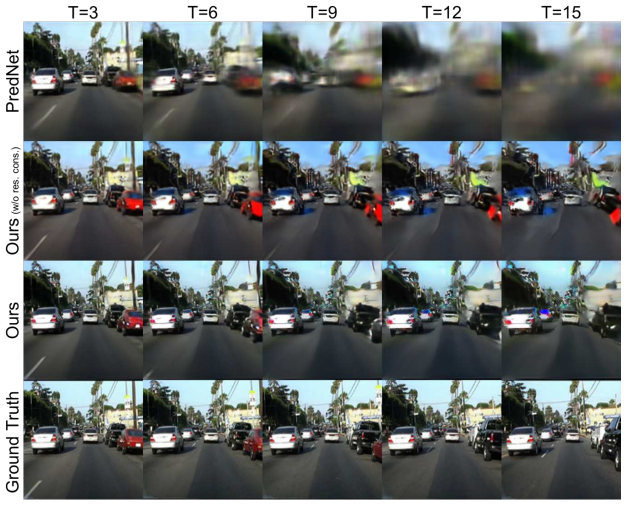

Figure 1. A comparison of predicted frames in a driving environment [5]. The state-of-the-art method, PredNet [17], predicts blurry images as the time step increases, whereas the proposed method shows relatively sharp and accurate images. Here, PredNet uses ten images as input whereas our method takes four images to predict future.

1. Introduction

Video prediction is the problem of generating future frames given a set of consecutive frames, which can be used for abnormal event detection [17], video coding [19], video completion, robotics [6], and autonomous driving. This problem has long been studied, and recently, deep learning has substantially improved the performance of video prediction algorithms, based on the deep architecture models such as convolutional neural networks (CNNs) and generative adversarial networks (GANs).

Conventional video prediction approaches [25] generally compute pixel-wise motion, and then, predict the motion of pixels in the future frame assuming the linearity of motions. A number of deep learning-based methods [16, 29, 31] inherit this idea. They explicitly compute pixel-wise motion through deep networks, *e.g.*, FlowNet [18], and then, the motion information is used to generate future frames together with training images. Although the idea is similar to the conventional approach, deep networks show promising results while handling complex motions in a dynamic scene. One major drawback of this approach is that computing pixel-wise motion is prone to errors owing to illumination change, occlusion, and abrupt camera motion.

A number of studies [2, 11, 13, 19, 24, 32] confirmed that deep networks can predict realistic future images without explicitly computing pixel-wise motion. The majority of them takes CNNs to predict future frames [2, 11, 13, 19], however, CNN-based methods often give blurry predictions because they minimize the loss against all the training images [15]. To avoid the blurry artifact, Byeon *et al*. [2] exploited the convolutional long term short memory (ConvLSTM) to capture both past and spatial contexts, which currently shows the best performance for a few of datasets. On the other hand, GANs have received a considerable attention in predicting future frames [16, 24, 32], which simultaneously train a discriminator network and a generator net-

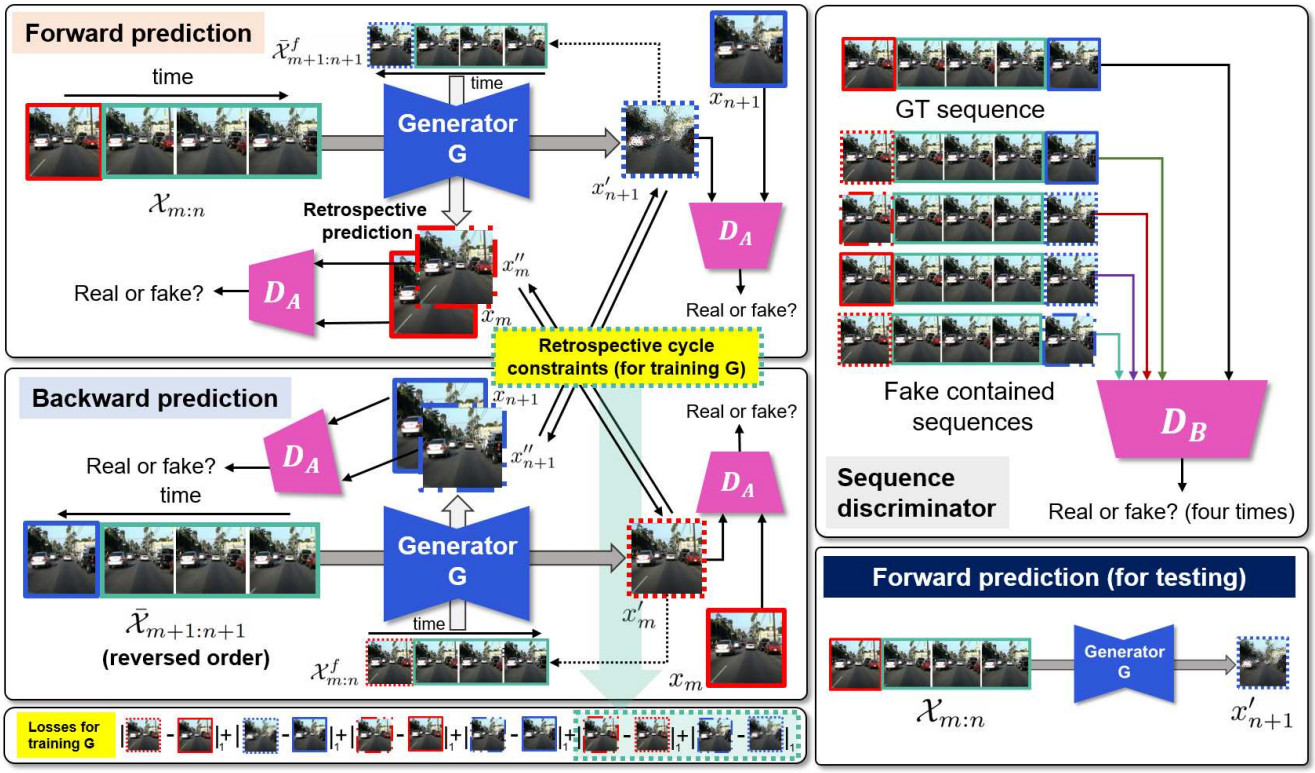

Figure 2. An overview of the proposed method. Our network consists of one generator and two discriminators, frame generator G, frame discriminator D_A, and sequence discriminator D_B. We propose a retrospective prediction scheme which allows the generator to predict a past frame by utilizing the predicted future frame. Furthermore, we train the generator with reversed input sequences and impose retrospective cycle constraints for the generator by minimizing the reconstruction losses between predicted frames, *e.g.* x'_{n+1} and x''_{n+1}. The frame discriminator decides whether the predicted frame is real or fake and the sequence discriminator distinguishes a fake contained image sequence from the real sequence to generate the temporally consistent frames.

work. The discriminator classifies the output image as real or fake whereas the generator predicts an image that fools the discriminator. Liang *et al*. [16] proposed to use dual generators and dual discriminators, to generate both future frames and pixel-wise motion at the same time.

Inspired by the success of deep networks in image generation [34, 35], we propose a deep network architecture for generating future frames having several distinct features as follows. First, we train a generator that is capable of predicting both future and past frames. We experimentally verify that this forward-backward compatible prediction yields better prediction performance. Second, we impose the cycle consistency between predicted frames with the aid of the retrospective prediction scheme, as illustrated in Fig. 2. The underlying idea of retrospective prediction is that if the predicted future frame is realistic, the generator should give a realistic past frame even the predicted future frame is given as input. Above two features significantly improves the future frame prediction performance, especially when predicting multiple frames ahead as shown in Fig. 1. Third, we propose a sequence discriminator that takes fake contained sequences as input, in addition to distinguishing a fake frame. The sequence discriminator is designed to increase the robustness and temporal consistency of predicted frames, which is crucial for video prediction.

2. Related Work

We review relevant studies related to video prediction using deep neural networks.

CNNs and recurrent neural networks (RNNs) have gained huge popularity over the last few years and a number of studies [2, 13, 19, 33] applied CNNs and RNNs to predict future frames from an image sequence. Kalchbrenner *et al*. [13] proposed the video pixel network, a probabilistic inference model consisting of resolution preserving CNN encoders and PixelCNN [30] decoders. They utilized convolutional LSTM to combine the output of the encoders over time and used dilated convolutions to achieve large receptive fields. Several more studies [2, 6, 19] adopted convolutional LSTM to take spatial and temporal contexts into account. Lotter *et al*. [19] introduced a predictive neural network not only to predict the movement of an object but

also to learn internal representation, *e.g*., the pose of an object, based on a series of repeating stacked modules. Byeon *et al*. [2] proposed parallel multi-dimensional LSTM units and blending units to capture past and spatial contexts, respectively. Finn *et al*. [6] proposed the action-conditioned convolutional LSTMs, which can predict different futures of an object conditioned on the action of an agent, *e.g*., a robot which holds the object. Xue *et al*. [33] tried to find an intrinsic representation of intensity changes, *i.e*., the difference image, through the conditional variational autoencoder. They used image-dependent convolution kernels to synthesize a probable future frame from a single image while considering various motions of an object. Villegas *et al*. [31] used two separate encoders for motion and content, but trained both encoders at simultaneously with multi-scale motion-content residual and combination layers. Luo *et al*. [21] proposed an unsupervised approach to predict long-term 3D motions based on the LSTM Encoder-Decoder method for activity recognition.

After the invention of adversarial training [8], many studies applied this scheme to generate images in the context of image-to-image translation [10, 35], super resolution [15], style transfer [12], and video prediction [17, 24]. Mathieu *et al*. [24] employed an image gradient loss in a multi-scale architecture, which significantly reduces blurring artifacts. Liu *et al*. [17] exploited spatial and motion constraints in addition to intensity and gradient losses. They computed optical flow through FlowNet [18] and the flow information is used to predict temporally consistent frames. On the other hand, many researchers tried to advance GANs [1, 22, 34, 35]. For example, WGAN [1] and LSGAN [22] modified a loss function for the discriminator to improve the stability of training. Zhu *et al*. [35] suggested a network having two generators, one takes the source image and the other takes the target image as input to predict each other image, respectively. This scheme enables to train an arbitrary pair of images. Similarly, Yi *et al*. [34] suggested using two discriminators to generate multiple types of outputs. Interestingly, Liang *et al*. [16] employed dual generators and dual discriminators for future frame prediction. Their network predicts pixel-wise motion and a future frame at the same time, but it requires ground truth flow information to train the network.

3. Proposed Method

Our framework consists of one generator and two discriminators, frame and sequence discriminators, as described in Fig. 2. The generator predicts both future and past frames, even if when the input sequence contains a fake frame. Moreover, the frame discriminator distinguishes fake frames individually, whereas the sequence discriminator decides whether the sequence contains fake frames or not.

For the clarification of explanation, we explain the notations used in the rest of the paper. Basically, we denote the generator as G, the frame discriminator as D_{A}, and the sequence discriminator as D_{B}. The input sequence is defined as

$$\mathcal{X}_{m:n} = \{x_m, x_{m+1}, ..., x_{n-1}, x_n\} \quad \text{s.t.} \quad m < n, \quad (1)$$

where $x_i \in \mathbb{R}^2$ is an image, m and n are indices to the first and last frames, the length of the sequence is $n - m + 1$, and the frames are chronologically ordered. Using $\mathcal{X}_{m:n}$ as input, the generator G predicts a future frame x_{n+1}. Here, the predicted frame, *i.e*. the fake frame, is denoted as x'_{n+1} with an apostrophe. Similarly, the reversed input sequence is defined as

$$\bar{\mathcal{X}}_{m:n} = \{x_n, x_{n-1}, ..., x_{m+1}, x_m\} \quad \text{s.t.} \quad m < n. \quad (2)$$

Using $\bar{\mathcal{X}}_{m:n}$, the generator predicts a past frame x_{m-1}. We also denote the sequence containing a fake frame as

$$\mathcal{X}^f_{m:n} = \{x_{m:n-1} \cup x'_n\}, \quad (3)$$

where the last frame is a fake assuming that x'_n is predicted from $x_{m-1:n-1}$. Similarly, its reversed case is defined by

$$\bar{\mathcal{X}}^f_{m:n} = \{\bar{x}_{m+1:n} \cup x'_m\}. \quad (4)$$

When the sequence with fake frames, $\mathcal{X}^f_{m:n}$ or $\bar{\mathcal{X}}^f_{m:n}$, is given as input, we denote predicted frames as x''_{n+1} or x''_{m-1}, to distinguish them from predicted frames x'_{n+1} and x'_{m-1} without fake frames.

3.1. Objective function

For training, we minimize the following objective function,

$$L = L_{\text{image}} + \lambda_1 L_{LoG} + \lambda_2 L^{\text{frame}}_{\text{adv}} + \lambda_3 L^{\text{seq}}_{\text{adv}} \quad (5)$$

which consists of two reconstruction losses and two adversarial losses. λ_1, λ_2, and λ_3 are non-zero weights for balancing four loss functions.

3.1.1 Reconstruction losses

The two reconstruction loss functions are used to train the generator. The first loss function is formulated by

$$L_{\text{image}} = \sum_{(p,q)\in\mathcal{S}^{\text{pair}}_{m,n}} l_1(p, q), \quad (6)$$

where $l_1(\cdot, \cdot)$ stands for L1 error between two images and $\mathcal{S}^{\text{pair}}_{m,n}$ is a set of image pairs defined as

$$\begin{aligned}\mathcal{S}^{\text{pair}}_{m,n} = \{&(x_m, x'_m), (x_m, x''_m), (x'_m, x''_m),\\ &(x_{n+1}, x'_{n+1}), (x_{n+1}, x''_{n+1}), (x'_{n+1}, x''_{n+1})\}.\end{aligned} \quad (7)$$

The first loss function (6) minimizes image reconstruction errors for six different pairs of images. (x_{n+1}, x'_{n+1}) and (x_m, x'_m) are used to minimize prediction errors in forward and backward directions. Therefore, the generator can predict both future and past frames. We define errors computed by (x_{n+1}, x''_{n+1}) and (x_m, x''_m) as retrospective prediction errors because x'_{n+1} is used to predict x''_m and x'_m is used to predict x''_{n+1}. In other words, if the predicted image x'_{n+1} is realistic, the generator can also take x'_{n+1} as one of input frames to look back the past frame. The last two pairs, (x'_m, x''_m) and (x'_{n+1}, x''_{n+1}), take the role of cyclic constraints because x'_m is generated by a forward sequence and x''_m is predicted by a backward sequence and similarly for (x'_{n+1}, x''_{n+1}). We say this loss function is retrospective and cyclic, because it utilizes frames generated through retrospective prediction. These pairs further constrain consistency between fake frames.

Similarly, we define the second reconstruction loss function as

$$L_{LoG} = \sum_{(p,q)\in\mathcal{S}^{\text{pair}}_{m,n}} l_1(LoG(p), LoG(q)). \tag{8}$$

This loss function computes difference between images after applying Laplacian of Gaussian (LoG) [23] operation, to better preserve image edges. In following study [3], they efficiently suppressed low frequency information and high frequency noise using laplacian pyramid for structurally enhanced image generation. We use the LoG operation to focus on the structural similarity that excludes noise.

3.1.2 Adversarial losses

Our proposed method is trained with two adversarial losses as in (9): frame adversarial loss $L^{\text{frame}}_{\text{adv}}$, and sequence adversarial loss $L^{\text{seq}}_{\text{adv}}$. The frame adversarial loss takes the role of classifying a frame as real or fake, Specifically, the frame adversarial loss determines whether four images, $(x'_{n+1}, x''_{n+1}, x'_m, x''_m)$, are real or fake as follows,

$$\begin{aligned} L^{\text{frame}}_{\text{adv}} &= l_{\text{A}}(\mathcal{X}_{m:n}, x_{n+1}) + l_{\text{A}}(\mathcal{X}^f_{m:n}, x_{n+1}) \\ &+ l_{\text{A}}(\bar{\mathcal{X}}_{m+1:n+1}, x_m) + l_{\text{A}}(\bar{\mathcal{X}}^f_{m+1:n+1}, x_m), \end{aligned} \tag{9}$$

where $\mathcal{X}_{m:n}$, $\mathcal{X}^f_{m:n}$, $\bar{\mathcal{X}}_{m+1:n+1}$, and $\bar{\mathcal{X}}^f_{m+1:n+1}$ denote four input sequences for the generator. The loss function, $l_{\text{A}}(p, q)$, is defined as

$$l_{\text{A}}(p, q) = \max_G \min_{D_A} [(D_A(q) - 1)^2 + (D_A(G(p)))^2]. \tag{10}$$

Here, the generator G takes a frame sequence p and predicts the future frame q, and D_{A} aims to distinguish q from $G(p)$. Against an adversary D_{A}, G aims at generating a fake frame in which D_{A} cannot distinguish it from the real frame. This loss function is from the least square GAN [22].

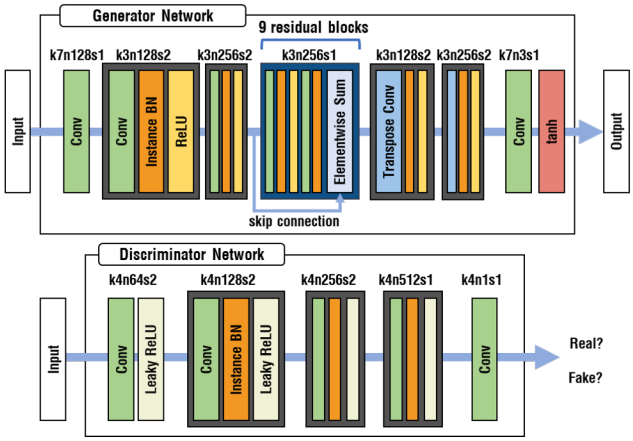

Figure 3. Network architectures of the generator and discriminator networks. Here, k, n, and s denote the kernel size, the number of feature maps, and the stride, respectively. The generator network learns to predict the next frame from the input image sequence, and the discriminator network learns to classify between real or generated frames from the generator network.

Similar to the frame adversarial loss, the sequence adversarial loss takes the role of classifying an input sequence as real or fake,

$$\begin{aligned} L^{\text{seq}}_{adv} &= l_{\text{B}}(\mathcal{X}_{m:n}, \mathcal{X}_{m:n+1}) + l_{\text{B}}(\mathcal{X}^f_{m:n}, \mathcal{X}_{m:n+1}) \\ &+ l_{\text{B}}(\bar{\mathcal{X}}_{m+1:n+1}, \bar{\mathcal{X}}_{m:n+1}) + l_{\text{B}}(\bar{\mathcal{X}}^f_{m+1:n+1}, \bar{\mathcal{X}}_{m:n+1}), \end{aligned} \tag{11}$$

where $l_B(p, r)$ takes two sequences as input,

$$l_{\text{B}}(p, r) = \max_G \min_{D_B} [(D_B(r) - 1)^2 + (D_B(G_c(p)))^2]. \tag{12}$$

Here, the generator G takes p as input to predict a new frame, $G(p)$, then compare it with a real image sequence r after concatenating p and $G(p)$. For the sake of simplicity, we denote the concatenated sequence as $G_c(p) = \{p \cup G(p)\}$, and all the procedures rely on a single generator G. D_{B} decides $G_c(p)$ as fake if at least one of the images is fake. This sequence discriminator encourages temporally consistent and robust prediction because it compares sequences rather than individual frames.

3.2. Network architecture

The generator and discriminator networks are illustrated in Fig. 3, in which we adopt an existing network architecture [12] for the generator network. The difference from [12] is that our generator takes multiple images as input to predict a future frame. The generator network consists of 4 convolution layers, 9 residual blocks [9], and 2 transpose convolution layers. The discriminator network consists of 5 convolution layers with leaky rectified linear units. Moreover, the network structure is the same for both frame and sequence discriminators except the number of input images.

In addition, we use the instance normalization scheme [4] for all layers of the generator and discriminator networks except the input and output layers.

4. Experimental Results

We evaluated the proposed method with three different types of real-world data and compared our results with the state-of-the-art methods. We also performed ablation studies to analyze the importance of each loss term.

4.1. Datasets

Videos captured by car-mounted cameras: We use two popular datasets that were recorded while driving various places using vehicle-mounted cameras: KITTI [7] and Caltech pedestrian [5] datasets. Since it was recorded in a driving car, it involves relatively large motions of pixels compared to other datasets.

Human action videos: The UCF101 [28] dataset consists of 13K video clips that cover 101 classes of human actions, captured with a variety of moving objects in static and dynamic environments.

Surveillance videos: The surveillance videos are captured at a fixed location. Therefore, it usually contains moving objects in a static environment. We used CUHK Avenue [26] and ShanghaiTech Campus [20] datasets to evaluate our method.

4.2. Training details

We set the length of an input sequence N to 4 and normalized intensities to be [-1, 1]. We flipped the input sequence horizontally with a probability of 0.3 for data augmentation. We used the Adam optimizer [14] for mini-batch stochastic gradient descent method with momentum parameters, $\beta_1 = 0.5$ and $\beta_2 = 0.999$, a batch size of 1, and a learning rate 0.0003 with linearly decay per every 100 epochs. For balancing different losses, we set $\lambda_1 = 0.005$, $\lambda_2 = 0.003$ and $\lambda_3 = 0.003$. The negative slope of Leaky ReLU is set to 0.2. To evaluate the Caltech pedestrian dataset, we followed experimental protocols of PredNet [19]. To train the network, we used the KITTI training dataset, that contains 41K images, and adjusted the frame rate of the Caltech dataset to 10 fps. We cropped the input images to 128×160 and resized the resolution of cropped images to 256 ×256. For the UCF 101 dataset, we used 10% of uniformly sampled images as the test set and the others for training as in previous studies [2, 24], for the fair comparison. For the surveillance datasets, we resized images to 256×256. To evaluate the method of Liu *et al*. [17], we calculated errors by using the pre-trained model provided by the authors.

Training took four days to train our network using the KITTI dataset on a single NVIDIA GTX 1080ti GPU. For testing, it took about 23ms to predict a single frame on a single GPU.

Table 1. Quantitative evaluation of video prediction algorithms using various datasets: Caltech pedestrian, UCF 101, and two surveillance datasets. The MSE is multiplied by 1,000 to clearly show the differences among different algorithms. The table compares four and five algorithms for Caltech and UCF101 datasets, respectively. † indicates that the corresponding method explicitly computes pixel-wise motion from images. Numbers are copied from original papers or citing papers. We put a dash if it is not presented in the papers.

Method	Caltech pedestrian			UCF101		
	MSE	PSNR	SSIM	MSE	PSNR	SSIM
Last frame copy	7.95	23.3	0.779	4.09	30.2	0.89
PredNet [19]	2.42	27.6	0.905	-	-	-
DM-GAN† [16]	2.41	-	0.899	-	-	-
BeyondMSE [24]	3.26	-	0.881	-	32	0.92
ContextVP [2]	1.94	28.7	**0.921**	-	34.9	0.92
MCnet+RES† [31]	-	-	-	-	31	0.91
EpicFlow† [27]	-	-	-	-	31.6	0.93
DVF† [36]	-	-	-	-	33.4	**0.94**
Ours	**1.61**	**29.2**	0.919	**1.37**	**35.0**	**0.94**

Dataset	Method	MSE	PSNR	SSIM
CUHK Avenue	Liu *et al*. † [17]	0.51	34.8	**0.98**
	Ours	**0.39**	**35.2**	**0.98**
ShanghaiTech	Liu *et al*. † [17]	0.93	31.4	**0.97**
	Ours	**0.64**	**34.1**	**0.97**

Table 2. Quantitative evaluation of the proposed method according to different lengths of input sequences. We differentiated the length of input from 2 to 10 and computed prediction errors using the Caltech pedestrian dataset trained on the KITTI dataset.

# of images	2	4	6	8	10
PSNR	29.167	**29.222**	29.006	28.940	29.009
SSIM	0.9193	0.9189	**0.9208**	0.9197	0.9189

4.3. Quantitative and qualitative evaluation

For quantitative evaluation, we use three metrics, mean squared error (MSE), structural similarity square error (SSIM), and peak signal to noise ratio (PSNR), that are frequently used for video prediction. Lower is better for MSE and higher is better for PSNR and SSIM.

Table 1 describes the quantitative evaluation result of the state-of-the-art methods and the proposed method, with various datasets. The Caltech dataset is the most challenging dataset due to the fast motion of a camera, therefore, the errors tend to be high compared to other datasets. To deal with abrupt camera motions, PredNet [19] and ContextVP [2] took ten frames as input for this dataset whereas we used four images as input. Nevertheless, our method shows the best results in terms of MSE and PSNR and a couple of predicted images are shown in Fig. 4.

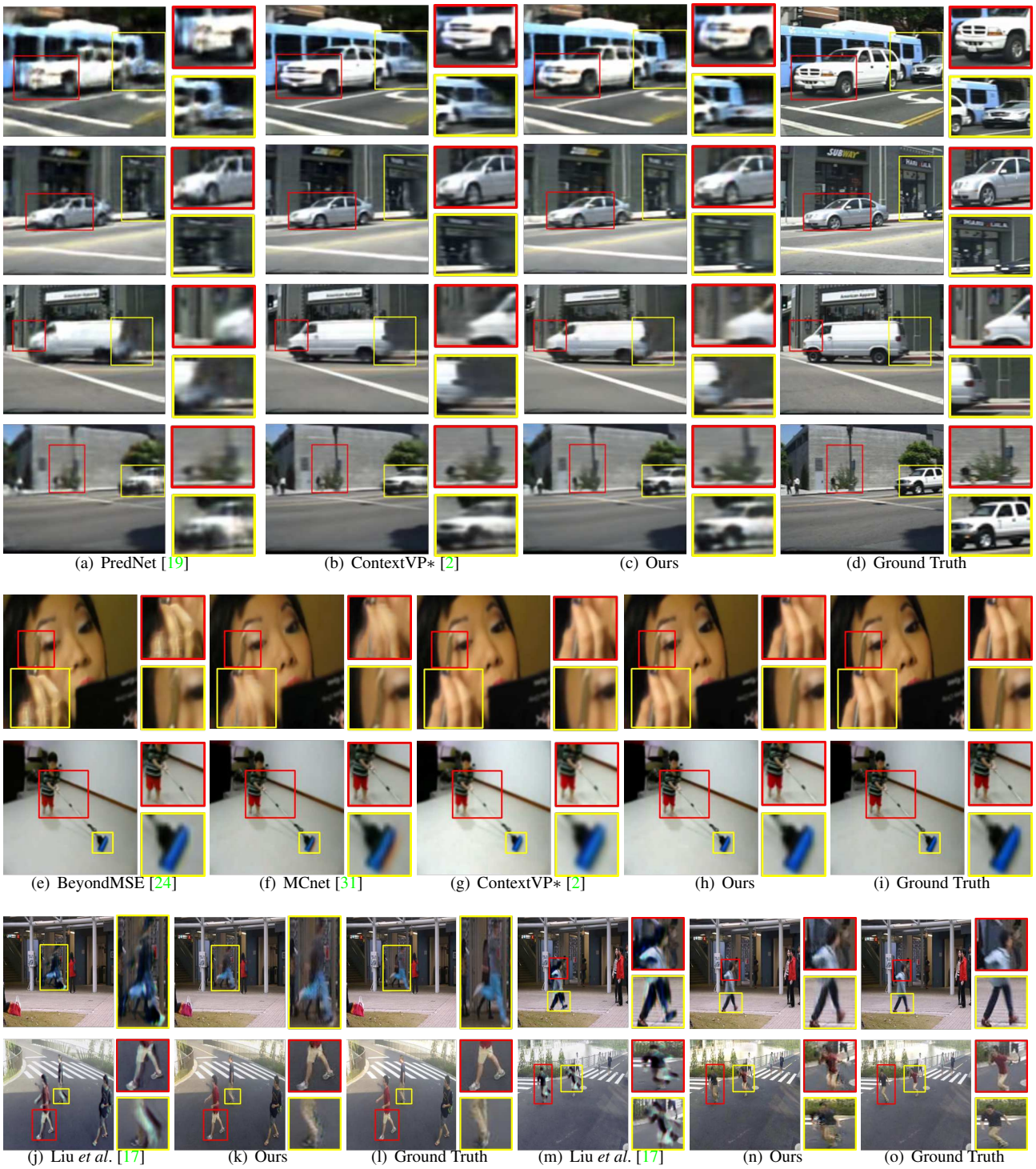

Figure 4. Qualitative comparisons of the predicted frame on the Caltech Pedestrian (a-d), UCF101 test set (e-i), CUHK Avenue test set, and ShanghaiTech test set (j-o). Each row shows the prediction results from consecutive sequence and network trained on the according to the dataset. Our method less artifact and blur around the ambiguity region that occur with fast motion, and denote the remarkable region in color. (∗) This result is provided by ContextVP [2].

For UCF101 dataset, we compare five state-of-the-art methods. As in BeyondMSE [24], we exclude pixels in static regions in computing errors. For this dataset, many papers explicitly compute pixel-wise motion, *i.e*. MCnet [31], EpicFlow [27], and DVF [36]. However, the performance of prediction is lower than ContextVP [2] and the proposed method, that directly generate future frames from an input sequence. For the surveillance datasets, we compare the proposed method with the method of Liu *et al*. [17]. Here, the average accuracy is higher than other datasets, because the surveillance video contains a large number of static regions. Figure 4 compares a few results of Liu *et al*. and ours, where the method of [17] shows unexpected artifacts because of the failure of motion estimation for pixels undergo large motions.

In addition, we also evaluate the sensitivity to the number of input frames. The optimal length of input sequences is four and six in terms of PSNR and SSIM, respectively. There is no big difference according to the number of input images as shown in Table 2; however, it is interesting to see that the use of two images showed better results than using eight or ten images. We presume that the use of two images is adequate for predicting the next frame in most cases, as long as a sufficient amount of training data is used for training. Hence, the larger number of input is desirable for long-term prediction.

4.4. Multi-step prediction evaluation

The multi-step prediction experiment is carried out to see how far the proposed method can predict future frames, *e.g*., fifteen frames later. The procedure of this experiment is as follows. First, we predict the next frame from an input sequence, *i.e*. four consecutive images. Then, we construct a new sequence by concatenating the last three frames of the input sequence and the predicted frame. Then, the new sequence is used to predict the next frame, this procedure is repeated until the designated frame, *e.g*., fifteen frames ahead, is predicted. This experiment was frequently adopted to verify the temporal and spatial consistency of predicted frames [16, 24, 36]. Table 3 shows quantitative evaluation results. Though the errors of predicted images increase as we predict farther future, the proposed method consistently shows better results than PredNet [19], which takes ten images as input. Qualitatively, the proposed method tends to show distorted images as shown in Fig. 5. However, predicted images do not suffer from blurry artifacts while capturing important characteristics of future frames, *e.g*. lanes and cast shadow. These experiments verify that the proposed network architecture is good at predicting far future frames, with the aid of retrospective cycle constraints and multiple discriminators.

Table 3. A quantitative comparison of multi-step prediction results with PredNet [19] and the proposed method. T indicates the time step, *e.g*., if T is 1 then the predicted frame corresponds to the image at 1 time steps ahead. The performance of prediction gradually decreases as T increases.

Method		$T=1$	3	6	9	12	15
PredNet [19]	PSNR	27.6	21.7	20.3	19.1	18.3	17.5
	SSIM	0.90	0.72	0.66	0.61	0.58	0.54
Ours	PSNR	29.2	25.9	22.3	20.5	19.3	18.4
	SSIM	0.91	0.83	0.73	0.67	0.63	0.60

Table 4. An ablation study of the proposed method with various loss configurations. ✔ and ✘ indicate that whether the corresponding part, *e.g*. a discriminator, is used or not for training the network. Forward and Backward with or without the retrospective loss (w/ res. or w/o res.)

Forward (w/o res.)	Forward (w/ res.)	Backward (w/o res.)	Backward (w/ res.)	L_{image}	L_{LoG}	$L_{\text{adv}}^{\text{frame}}$	$L_{\text{adv}}^{\text{seq}}$	PSNR	SSIM
✔	✘	✘	✘	✔	✘	✘	✘	26.3	0.892
✔	✘	✔	✘	✔	✔	✘	✘	26.8	0.899
✔	✔	✘	✘	✔	✔	✘	✘	26.9	0.900
✔	✔	✔	✔	✔	✔	✘	✘	27.5	0.904
✔	✔	✔	✘	✔	✔	✔	✘	28.4	0.912
✔	✔	✔	✔	✔	✔	✔	✔	**29.2**	**0.919**

4.5. Ablation study

We carried out an ablation study under various settings, to see the impact of core ideas such as backward prediction, frame discriminator, and sequence discriminator. Table 4 compares quantitative results with different settings, in the ascending order of PSNR from the top to the bottom row. Overall, the absence of each module degraded the performance of predicting future frames. It is important that the absence of backward prediction implies that all the loss terms related to the backward prediction are eliminated during training; it reduces the number of input images into half for the discriminators. Two different settings, forward prediction with the frame discriminator and bi-directional prediction with the frame discriminator, show near the state-of-the-art performance. The use of all components, the proposed method shows the best performing result, meaning that the combination of all components is crucial for the prediction of future frames.

5. Conclusion

We have proposed an unsupervised framework for predicting future frames, named as Retrospective Cycle GAN, consisting of one generator and two discriminators. The generator takes forward and backward sequences as input during training and the consistency of bi-directional prediction is leveraged through the retrospective cycle constraints. In addition, we exploited two discriminators for adversarial

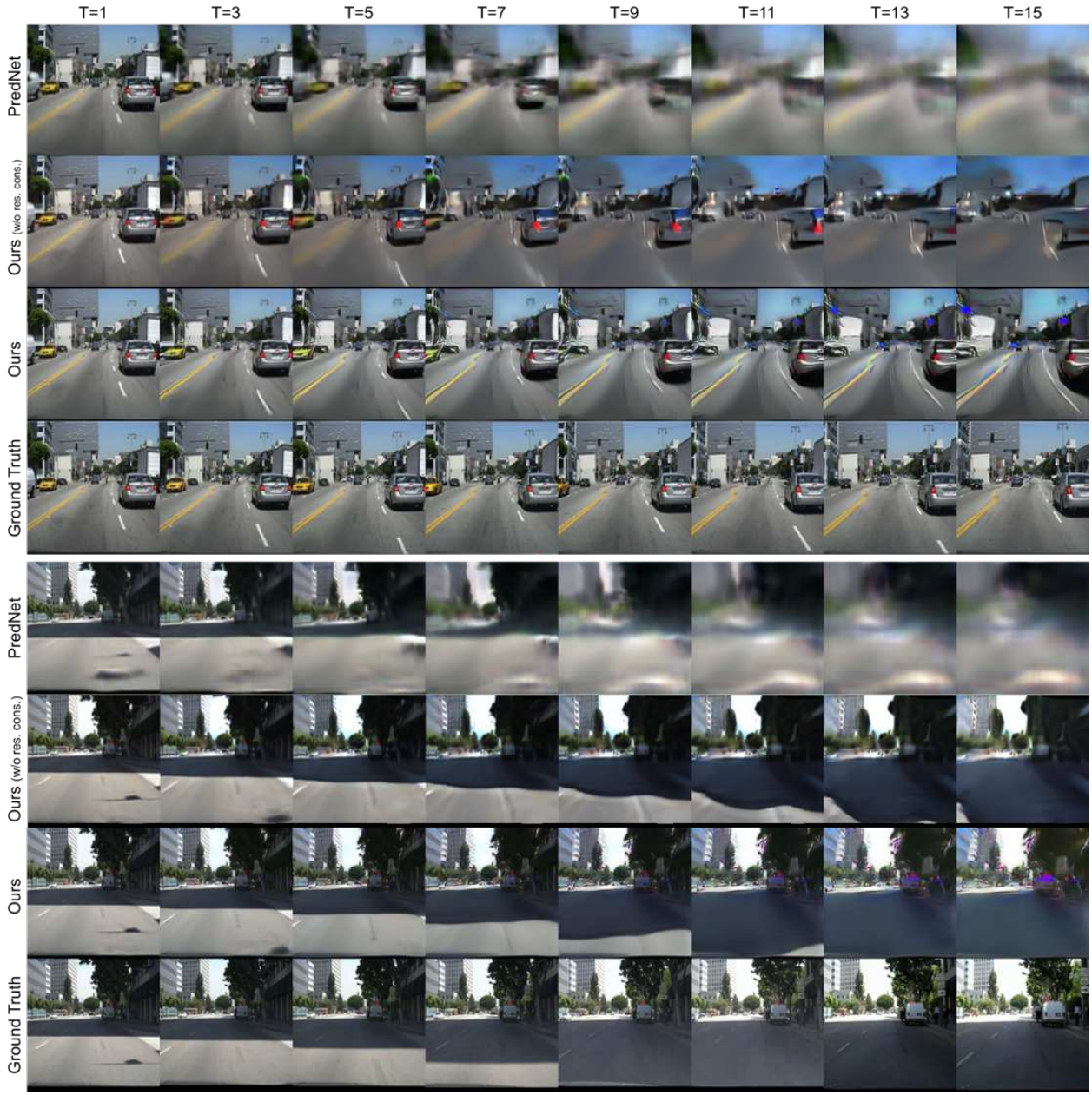

Figure 5. A comparison of multi-step prediction results. The second results of each image (w/o res. cons.) represent without retrospective constraint. The first sequence is captured by a forward moving vehicle while changing the lane and the second sequence contains a cast shadow which is going to dominate the entire road. The proposed method can predict the important characteristics of future frames; for example, the position of cars and lane marking as well as the area of cast shadows. More results can be found in the supplementary material.

training, the frame discriminator is for discriminating fake frames likewise conventional GANs. The sequence discriminator takes fake contained sequences to improve the robustness and accuracy of predicted frames over time under the temporal consistency. We experimentally verified the superiority of the proposed method from various perspectives, showing the state-of-the-art performance in predicting future frames.

Acknowledgement. This work was partially supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No.NRF-2019R1C1C1003676) and by "The Cross-Ministry Giga KOREA Project" grant funded by the Korea government(MSIT) (No.GK19P0200, Development of 4D reconstruction and dynamic deformable action model based hyper-realistic service technology). We would also like to thank Ju Hong Yoon and Yeong Won Kim for many helpful comments and discussions.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 3

[2] Wonmin Byeon, Qin Wang, Rupesh Kumar Srivastava, and Petros Koumoutsakos. Contextvp: Fully context-aware video prediction. In *European Conference on Computer Vision (ECCV)*, 2018. 1, 2, 3, 5, 6, 7

[3] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems (NIPS)*, 2015. 4

[4] Andrea Vedaldi Dmitry Ulyanov and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016. 5

[5] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. In *Pattern Analysis and Machine Intelligence (PAMI)*, 2012. 1, 5

[6] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in neural information processing systems (NIPS)*, 2016. 1, 2, 3

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. In *International Journal of Robotics Research (IJRR)*, 2013. 5

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems (NIPS)*, 2014. 3

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *conference on computer vision and pattern recognition (CVPR)*, 2016. 4

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[11] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 1

[12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, 2016. 3, 4

[13] Nal Kalchbrenner, Aäron Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. In *International Conference on Machine Learning (ICML)*, 2017. 1, 2

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[15] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 3

[16] Xiaodan Liang, Lisa Lee, Wei Dai, and Eric P Xing. Dual motion gan for future-flow embedded video prediction. In *International Conference on Computer Vision (ICCV)*, 2017. 1, 2, 3, 5, 7

[17] W. Liu, D. Lian W. Luo, and S. Gao. Future frame prediction for anomaly detection – a new baseline. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 3, 5, 6, 7

[18] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 3

[19] William Lotter, Gabriel Kreiman, and David Cox. Deep predictive coding networks for video prediction and unsupervised learning. In *International Conference on Learning Representations (ICLR)*, 2017. 1, 2, 5, 6, 7

[20] Weixin Luo, Wen Liu, and Shenghua Gao. A revisit of sparse coding based anomaly detection in stacked rnn framework. In *International Conference on Computer Vision (ICCV)*, 2017. 5

[21] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. 3

[22] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *International Conference on Computer Vision (ICCV)*, 2017. 3, 4

[23] David Marr and Ellen Hildreth. Theory of edge detection. *Proc. R. Soc. Lond. B*, 1980. 4

[24] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *International Conference on Learning Representations (ICLR)*, 2016. 1, 3, 5, 6, 7

[25] Viorica Pătrăucean, Ankur Handa, and Roberto Cipolla. Spatio-temporal video autoencoder with differentiable memory. In *International Conference on Learning Representations (ICLR) Workshop*, 2016. 1

[26] Mahdyar Ravanbakhsh, Moin Nabi, Enver Sangineto, Lucio Marcenaro, Carlo Regazzoni, and Nicu Sebe. Abnormal event detection in videos using generative adversarial nets. In *International Conference on Image Processing (ICIP)*, 2017. 5

[27] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 5, 7

[28] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5

[29] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[30] Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[31] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *International Conference on Learning Representations (ICLR)*, 2017. 1, 3, 5, 6, 7

[32] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems (NIPS)*, 2016. 1

[33] Tianfan Xue, Jiajun Wu, Katherine Bouman, and Bill Freeman. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2, 3

[34] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *International Conference on Computer Vision (ICCV)*, 2017. 2, 3

[35] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *International Conference on Computer Vision (ICCV)*, 2017. 2, 3

[36] Xiaoou Tang Yiming Liu Ziwei Liu, Raymond Yeh and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *International Conference on Computer Vision (ICCV)*, 2017. 5, 7