SSA-GAN: Cloud Video Generation from a Single Image with Spatial Self-Attention Generative Adversarial Networks

Daichi Horita[1,a)] Keiji Yanai[1,b)]

Abstract

We usually predict how objects will move in the near future in our daily lives. However, how do we predict? In this paper, to address this problem, we propose a GAN-based network to predict the near future for fluid object domains such as a cloud scene. Our model takes one frame and is able to predict future frames. Inspired by the self-attention mechanism [15], we propose introducing the spatial self-attention mechanism into the model. The self-attention mechanism calculates the reaction at a certain position as a weighted sum of the features at all positions, which enables us to learn the model efficiently in one-stage learning. In the experiment, both quantitative and qualitative evaluation show that our model is comparable compared with the state-of-the-art method which performs two-stage learning.

1. Introduction

We propose the Spatial Self-Attention Generative Adversarial Network (SSA-GAN) for future frame prediction. Our model consists of a generator and a discriminator. The generator has not a simple encoder-decoder architecture but the architecture like 3D U-Net [3] to avoid generating blurred images caused by losing content details. In addition, the generator has spatial self-attention layers based on [15] after each 3D convolutions and deconvolutions to preserve the spatial physical structure. Given a stationary input frame, the generator predicts future video frames which indicate how it will move in the future. In this way, our model keeps content details and predict as realistic dynamic scene transition as possible. We present a few example frames which are generated by our method and existing method. As shown in Fig. 1, the image frames generated by our model is realistic because it keeps the edge in the final frame and moves the cloud more.

Major contributions of this paper can be summarized as follows:

(1) We propose the Spatial Self-Attention Generative Adversarial Networks(SSA-GAN) for video prediction.
(2) We propose the spatial self-attention framework based on a self-attention mechanism [15], which enables to learn a model in one-stage while emphasizing spatial correlation between time series.
(3) We demonstrate that our model achieves comparable results with the state-of-the-art method.

1 The University of Electro-Communications, Japan
a) horita-d@mm.inf.uec.ac.jp
b) yanai@cs.uec.ac.jp

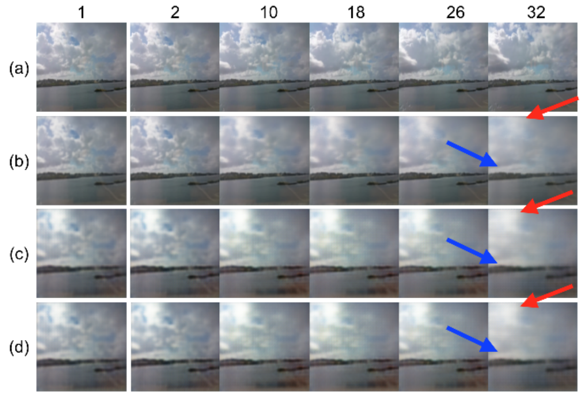

Fig. 1 Some example results of cloud motions generated by our proposed model trained with the cloud time-lapse dataset. The first column shows input images, and the next five columns show the predicted frames. From top to bottom: (a) the ground truth, (b) our model, (c) first stage of MD-GAN [18], and (d) second stage of MD-GAN, respectively.

2. Related Work

2.1 Generative Adversarial Networks

Generative adversarial networks (GANs) [1], [4] have achieved impressive results in image generation [9], [13] and image-to-image translation [8], [21]. GANs consists of a generator and a discriminator. The discriminator learns to distinguish the produced fake samples from the real ones, while the generator learns to generate fake samples which are not distinguishable from the real ones. In this paper, we also leverage an adversarial loss to learn the mapping to generate future frames as realistic as possible.

2.2 Video generation

There are two main approaches to the field of video generation using GAN. One of them is to produce plausible videos by limiting video datasets to specific areas such as human faces and poses [2], [19]. The other is a study to deal without such constraints [14], [16]. MoCoGAN [14] generates videos efficiently by decomposing the latent space into content and motion subspaces. In this paper, our study is close to the latter because our model generates video frames with free movement without such constraints.

2.3 Video prediction

Video prediction has tasks different from the video generation and it is one of the major problems in the field of computer vision. In particular, the method of modeling the domain of videos is not unified, but in the existing research, the next frame is inferred using the recurrent neural networks like LSTM. In addition, a well-known approach is to

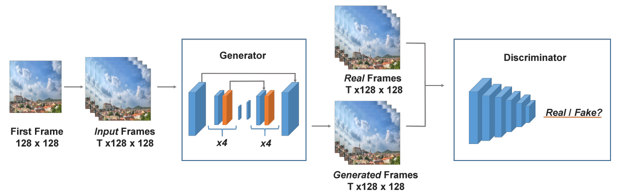

Fig. 2 The overview architecture of our SSA-GAN. Blue layers indicate 3D convolutional layers and 3D deconvolutional layers, and orange layers indicate the spatial self-attention layers. The generator consists of an architecture like 3D U-Net, preventing skip connection from missing content. The input image is duplicated T times from the first frame of the ground truth.

estimate intermediate features of dynamic motion using optical flow [10], [11]. However, our model architecture is different from other methods because our model does not use optical flow and the recurrent neural network. The cutting-edge study is MD-GAN [18], which predicts future frames from a stationary image. However, there is a big difference between our model and MD-GAN. The first is that [18] learns in two-stage, but our model learns in one-stage. The second is that MD-GAN [18] leverage the Gram matrix to explicitly model dynamic motion, while our model leverages the spatial self-attention to model by the spatial average weight.

2.4 Self Attention Mechanism

Recently, there are many works which produced remarkable results using the self-attention mechanism [15], [17]. The self-attention module [15] calculates the response at the position in the feature map by paying attention not only to the surroundings of an attending point but also to all the positions in the image and taking a weighted average of them. Non-local Neural Networks [17] proposes a non-local operator which handles global information in spatial and temporal directions using the self-attention method [15]. Similarly, our spatial self-attention is likewise based on a self-attention mechanism. However, in the case of frame prediction and generation, because all the frames are equally important, only the spatial direction is used without considering the time direction.

3. Our Approach

3.1 Spatial Self-Attention GAN

As shown in Fig. 2, SSA-GAN is a generative adversarial network composed of the generator G and the discriminator D. G consists of 3D U-Net [3] with the skip connection which prevents future frames from blurring and losing content information. More information on the generator is shown in Table 1. In addition, G has the spatial self-attention module followed by each convolutional and deconvolutional layers. By using the module, it enables G to efficiently learn spatial features. D consists of the same structure as the encoder part of the generator and has a sigmoid function in the final layer except for the spatial self-attention layers. D takes real and fake videos as input and tries to distinguish them. Our model learns to take a single RGB frame and predict next T future frames as realistic as possible by one-stage learning.

3.2 Spatial Self-Attention Module

We propose to introduce a spatial self-attention module to learn the long-range dependence within a frame, which allows the network to first rely on the cues in only neighboring pixels and then gradually learn to assign more weight to areas outside the neighborhood. In other words, it enables

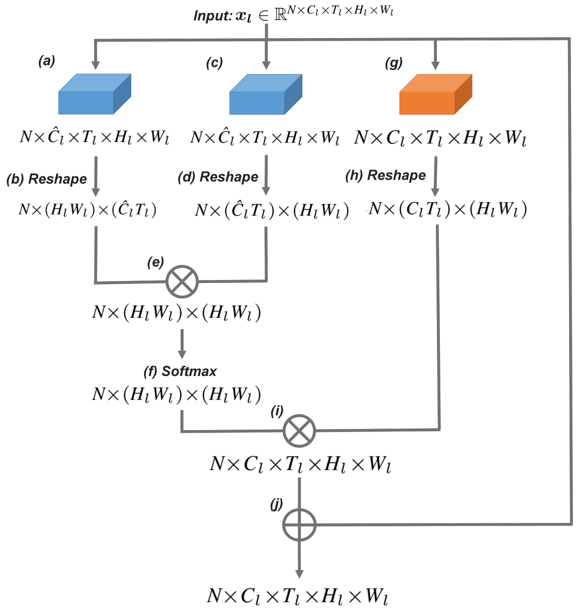

Fig. 3 The overview architecture of our spatial self-attention mechanism. The feature maps are shown as the shape of their tensors. "$\otimes$" denotes a matrix multiplication, and "$\oplus$" denotes element-wise sum. The softmax operations are calculated in each column. The blue box changes the matrix of channel size C_l to $\hat{C}_l$ and outputs it, but the orange box outputs a matrix with the channel size C_l.

the network to learn simple tasks firstly and to gradually increase the complexity of the task to get better features. Each l-th layer of the convolution and deconvolution output is $\boldsymbol{x_l} \in \mathbb{R}^{N\times C_l\times T_l\times H_l\times W_l}$, where (N, C_l, T_l, H_l, W_l) are the batch size, the number of channels, length of the time axis, the height and the width of the feature maps, respectively. As shown in Fig. 3, (a) the spatial self-attention layer firstly applies the 3D convolution to the input feature $\boldsymbol{x_l}$ and obtains $\boldsymbol{x_{l_1}} \in \mathbb{R}^{N\times\hat{C}_l\times T_l\times H_l\times W_l}$ and (b) resizes to $\hat{\boldsymbol{x}}_{\boldsymbol{l_1}} \in \mathbb{R}^{N\times(H_lW_l)\times(\hat{C}_lT_l)}$. Next, (c) the layer gets $\boldsymbol{x_{l_2}} \in \mathbb{R}^{N\times\hat{C}_l\times T_l\times H_l\times W_l}$ by (a) the same operation and (d) resizes to $\hat{\boldsymbol{x}}_{\boldsymbol{l_2}} \in \mathbb{R}^{N\times(\hat{C}_lT_l)\times(H_lW_l)}$. Furthermore, (e) after calculating the matrix multiplication of $\hat{\boldsymbol{x}}_{\boldsymbol{l_1}}$ and $\hat{\boldsymbol{x}}_{\boldsymbol{l_2}}$, (f) softmax calculate to obtain the attention $\hat{\boldsymbol{X}}_{\boldsymbol{l}} \in \mathbb{R}^{N\times(H_lW_l)\times(H_lW_l)}$, defined as

$$\hat{\boldsymbol{X}}_{\boldsymbol{l}} = \frac{\exp(\boldsymbol{X_l})}{\sum\exp(\boldsymbol{X_l})}, \text{where } \boldsymbol{X_l} = \boldsymbol{x_{l_1}} \otimes \boldsymbol{x_{l_2}}. \tag{1}$$

This represents the weighted average inside the feature map. Following, (g) the layer applies the 3D convolution to the input feature $\boldsymbol{x_l}$ and obtains $\boldsymbol{x_{l_3}} \in \mathbb{R}^{N\times C_l\times T_l\times H_l\times W_l}$ and (h) resizes to $\hat{\boldsymbol{x}}_{\boldsymbol{l_3}} \in \mathbb{R}^{N\times(C_lT_l)\times(H_lW_l)}$. Then, (i) the resized output of the layer is $\boldsymbol{o} \in \mathbb{R}^{N\times C_l\times T_l\times H_l\times W_l}$, defined as

$$\boldsymbol{o_l} = \hat{\boldsymbol{X}}_{\boldsymbol{l}} \otimes \boldsymbol{x_{l_3}}. \tag{2}$$

Finally, (j) the layer multiplies the output o_l scale parameter γ and calculates the sum of it with the input feature map x_l. Therefore, the final output is y_l, defined as

$$\boldsymbol{y_l} = \gamma\boldsymbol{o_l} + \boldsymbol{x_l}, \tag{3}$$

where γ is a parameter initialized with 0. We leverage all 3D convolution of kernel 1 in the spatial self-attention layer and $\hat{C}_l = C_l$ for all experiments.

Table 1 The architecture of the generator.

Layer	Filters	Kernel	Stride	Padding
conv1	32	(3, 4, 4)	(1, 2, 2)	(1, 1, 1)
conv2	64	(4, 4, 4)	(2, 2, 2)	(1, 1, 1)
conv3	128	(4, 4, 4)	(2, 2, 2)	(1, 1, 1)
conv4	256	(4, 4, 4)	(2, 2, 2)	(1, 1, 1)
conv5	512	(4, 4, 4)	(2, 2, 2)	(1, 1, 1)
conv6	512	(2, 4, 4)	(1, 1, 1)	(0, 0, 0)
deconv1	512	(2, 4, 4)	(1, 1, 1)	(0, 0, 0)
deconv2	256	(4, 4, 4)	(2, 2, 2)	(1, 1, 1)
deconv3	128	(4, 4, 4)	(2, 2, 2)	(1, 1, 1)
deconv4	64	(4, 4, 4)	(2, 2, 2)	(1, 1, 1)
deconv5	32	(4, 4, 4)	(2, 2, 2)	(1, 1, 1)
deconv6	3	(3, 4, 4)	(1, 2, 2)	(1, 1, 1)

3.3 Spatial Self-Attention GAN Objectives

Our goal is to predict future frames from the stationary image in the one-stage learning model and to predict it as realistically as possible.

Adversarial Loss. In order to make the generated future frame more realistic, we adopt an adversarial loss

$$\mathcal{L}_{adv} = \min_G \max_D \mathop{\mathbb{E}}_{Y \sim \mathbb{P}_r} [\log D(Y)] + \mathop{\mathbb{E}}_{\bar{X} \sim \mathbb{P}_g} [\log (1 - D(\bar{X}))], \tag{4}$$

where Y is sampled from the data distribution $\mathbb{P}_r$ and $\bar{X}$ is sampled from the model distribution $\mathbb{P}_g$ implicitly defined by $\bar{X} = G(X), X \sim \mathbb{P}_r$. The generator predicts a future frame $\bar{X}$ from a stationary video X to fool the discriminator, while the discriminator tries to distinguish between real and fake frames.

Content Loss. Previous approach [12] indicates that it is more beneficial to combine traditional loss like $L1$ norm and $L2$ norm with the adversarial loss. Although the role of the discriminator remains unchanged, the role of the generator play a role not only to fool the discriminator but also to generate the fake images closer to the real ones. In addition, pix2pix [8] shows that the output images become less blurred at $L1$ norm than $L2$ norm. To ensure that the content of the generated frames is a pattern similar to the content of the real video, the content objective is defined as complementing the adversarial objective,

$$\mathcal{L}_{con} = \mathop{\mathbb{E}}_{Y \sim \mathbb{P}_r, \bar{X} \sim \mathbb{P}_g} [\|Y - \bar{X}\|], \tag{5}$$

where the generator tries to generate a frame similar to Y at the pixel level.

Full Objective. Finally, the loss objectives which optimize the generator and the discriminator are defined as

$$\mathcal{L}_D = -\mathcal{L}_{adv}, \tag{6}$$

$$\mathcal{L}_G = \mathcal{L}_{adv} + \lambda_{con} \mathcal{L}_{con}, \tag{7}$$

where λ_{con} is a hyperparameter that controls the relative importance of content loss compared to the adversarial loss. We leverage $\lambda_{con} = 1$ for all experiments.

4. Implementation Details

As shown in Fig. 2, SSA-GAN is composed of the generator of the 3D U-Net architecture [3] with the skip-connection and the discriminator.The skip connection is useful as identity mapping [6]. The generator network consists of a six convolution layer, six transposed convolutions, and skip connection. In addition, the generator has the spatial self-attention module following each convolutional and deconvolutional layers. We apply Batch Normalization [7] to all 3D convolutional layers except the first and last layers, followed by Leaky ReLU and ReLU. The output layer exploits

Table 2 Experiment results on the cloud dataset by MSE, PSNR, and SSIM. Ours (a) and Ours (b) are models which proposed layers are added to the first and second stages of MD-GAN [18], respectively.

Method	MSE↓	PSNR↑	SSIM↑
[18] *I*	0.0280	23.14	0.5997
[18] *II*	0.0245	23.8529	0.6327
Ours (a)	**0.0238**	**24.3512**	**0.6991**
Ours (b)	0.0259	23.5224	0.6460

Table 3 Quantitative comparison results on the cloud time-lapse dataset. The value range of POS is $[0, 1000]$.

"Which is more realistic?"	POS
Prefer Ours over [18] Stage *I*	871
Prefer Ours over [18] Stage *II*	526
Prefer [18] Stage *I* over Real	286
Prefer [18] Stage *II* over Real	322
Prefer Ours over Real	334

Tanh as an activation function for the generator. We adopt Adam as the optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.9$. The learning rate is fixed at 0.0002 during learning. We perform one generator update after five discriminator updates as in [5]. We set the batch size to 16 for all experiments. We use the same architecture as [18] regarding the architecture of the generator network.

5. Experiments

5.1 Datasets

To evaluate the robustness and effectiveness of our approach, we compare our model with other approaches using two datasets, which are the cloud time-lapse dataset [18] and the beach dataset [16].

Cloud Time-Lapse Dataset. We leverage the time lapse video dataset *1 gathered from the Internet [18] for evaluation. The dataset consists of 35,392 training video clips and 2,815 testing video clips each containing 32 frames. However, the original size of each frame is $3 \times 640 \times 360$, and we resize it into a square image size $3 \times 128 \times 128$. We duplicate the first frame of the input video 32 times to make it a static input video. We normalized the inputs by converting the color value to $[-1, 1]$.

Beach Dataset. We leverage the unlabeled video dataset which is released by [16] *2, which do not contain any time-lapse video. We divide the dataset of 10% into training data and 90% into evaluation data.

5.2 Experiments on the Cloud Time-Lapse dataset

In this section, we evaluate the performance of SSA-GAN for both quantitative and qualitative evaluation. As a baseline model, we adopt MD-GAN, which is the method of performing the highest accuracy using the cloud time-lapse dataset. In addition, we also experiment with our model (a) to learn Stage *I* and our model (b) to learn Stage *II* that introduced our proposed layer at each stage of MD-GAN.

Quantitative Results. To evaluate whether the predicted future frames is more natural, we compare these models in each pair in the same way as [18]. We prepare 100 pairs of videos according to the five cases shown in Table 3, which is selected randomly from the evaluation dataset. We show ten subjects the pairs of generated video and ask them "which is more realistic?". Then, we count the answers of their evaluation, which means Preference Opinion Score (POS). The results generated from our model randomly appear in either left or right side in the test to

*1 https://sites.google.com/site/whluoimperial/mdgan
*2 http://www.cs.columbia.edu/ vondrick/tinyvideo/

Table 4 Experiment results on the beach dataset by MSE, PSNR, and SSIM.

Method	MSE↓	PSNR↑	SSIM↑
RNN-GAN	0.1849	7.7988	0.5143
VGAN	0.0958	11.5586	0.6035
MD-GAN Stage *II*	0.0422	16.1951	**0.8019**
Ours (a)	0.0379	23.6601	0.7320
Ours (b)	**0.0374**	**25.6432**	0.7346

get a more reliable evaluation. As shown in Table 3, our model achieved the better results than other models. We demonstrate that the spatial self-attention module generates dynamic cloud motion prediction from all spatial relationships in the image. Finally, for each approach, we calculate the Mean Squared Error (MSE), Peak Signal to Noise Ratio (PSNR), and Structural Similarity Index (SSIM) between the full of evaluation datasets. As shown in Table 4, our model (a) shows better performance than other methods.

Qualitative Results. Fig. 1 shows the output of each model. We compare (b) our output video, (c) the video generated by stage one of MD-GAN and (d) the video generated by stage two of MD-GAN. The red arrow is used to indicate the locations and areas where obvious movement occurs between adjacent frames. The result shows that the clouds of (c) and (d) move hardly, but the clouds of (b) move. As shown by the blue arrow, (b) does not blur objects other than clouds. The difference in model structure between (b) and (c) is only the spatial self-attention. Thus, the results demonstrate that the spatial self-attention module generates dynamic cloud motion prediction from the spatial relationships in the image.

5.3 Experiments on the Beach Dataset

In this section, we compare our model with MD-GAN, VGAN, and RNN-GAN [20] using the beach dataset in a quantitative evaluation. All models generate 32 future frames and are trained using the adversarial loss. VGAN and RNN-GAN take an image of 64×64 resolution and predict future frames of 64×64 resolution. In addition, MD-GAN takes also the same resolution image to satisfy these conditions. Therefore, for a fair comparison, our model is also adjusted to learning with a 64×64 resolution image. To learn this model, our model was removed the first convolutional and deconvolutional layer so that model can predict future frames of resolution 64×64. All models calculate MSE, PSNR, and SSIM using randomly sampled 1000 videos from the evaluation dataset. As shown in Table. 4, our model showed the better scores than the other models regarding PSNR an MSE, although the MD-GAN Stage *II* achieved the best score in SSIM.

6. Conclusion

We propose SSA-GAN with the spatial self-attention mechanism based on the self-attention [15]. The spatial self-attention mechanism calculates the reaction at a certain position as a weighted sum of the features at all positions. In addition, the mechanism makes it possible to learn models efficiently in the one-stage of end-to-end learning. We hope our work is to enable users to develop how we predict future movement.

More experimental results including generated videos can be seen at https://luv2019ssagan.github.io/.

7. Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 15H05915, 17H01745, 17H06100 and 19H04929.

References

[1] Arjovsky, M., Chintala, S. and Bottou, L.: Wasserstein Generative Adversarial Networks, *International Conference on Machine Learning(ICML)* (2017).

[2] Cai, H., Bai, C., Tai, Y. and Tang, C.: Deep Video Generation, Prediction and Completion of Human Action Sequences, *Proc.of European Conference on Computer Vision(ECCV)* (2018).

[3] Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T. and Ronneberger, O.: 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation, *International Conference on Medical Image Computing Computer Assisted Intervention(MICCAI)* (2016).

[4] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y.: Generative Adversarial Nets, *Proc.of Neural Information Processing Systems* (2014).

[5] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. and Courville, A. C.: Improved Training of Wasserstein GANs, *Proc.of Neural Information Processing Systems* (2018).

[6] He, K., Zhang, X., Ren, S. and Sun, J.: Identity Mappings in Deep Residual Networks, *Proc.of European Conference on Computer Vision(ECCV)* (2016).

[7] Ioffe, S. and Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, *International Conference on Machine Learning(ICML)* (2015).

[8] Isola, P., Zhu, J.-Y., Zhou, T. and Efros, A. A.: Image-to-Image Translation with Conditional Adversarial Networks, *Proc.of IEEE Computer Vision and Pattern Recognition(CVPR)* (2017).

[9] Karras, T., Aila, T., Laine, S. and Lehtinen, J.: Progressive Growing of GANs for Improved Quality, Stability, and Variation, *"Proc. of the International Conference on Learning Representation(ICLR)"* (2017).

[10] Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X. and Yang, M.: Flow-Grounded Spatial-Temporal Video Prediction from Still Images, *Proc.of European Conference on Computer Vision(ECCV)* (2018).

[11] Ohnishi, K., Yamamoto, S., Ushiku, Y. and Harada, T.: Hierarchical Video Generation from Orthogonal Information: Optical Flow and Texture, *Proc.of AAAI Conference on Artificial Intelligence(AAAI)* (2018).

[12] Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T. and Efros, A. A.: Context Encoders: Feature Learning by Inpainting, *Proc.of IEEE Computer Vision and Pattern Recognition(CVPR)* (2016).

[13] Radford, A., Metz, L. and Chintala, S.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, *Proc. of the International Conference on Learning Representation(ICLR)* (2016).

[14] Tulyakov, S., Liu, M., Yang, X. and Kautz, J.: MoCoGAN: Decomposing Motion and Content for Video Generation, *Proc.of IEEE Computer Vision and Pattern Recognition(CVPR)* (2018).

[15] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I.: Attention Is All You Need, *Proc.of Neural Information Processing Systems* (2017).

[16] Vondrick, C., Pirsiavash, H. and Torralba, A.: Generating Videos with Scene Dynamics, *Proc.of Neural Information Processing Systems* (2016).

[17] Wang, X., Girshick, R. B., Gupta, A. and He, K.: Non-local Neural Networks, *Proc.of IEEE Computer Vision and Pattern Recognition(CVPR)* (2018).

[18] Xiong, W., Luo, W., Ma, L., Liu, W. and Luo, J.: Learning to Generate Time-Lapse Videos Using Multi-Stage Dynamic Generative Adversarial Networks, *Proc.of IEEE Computer Vision and Pattern Recognition(CVPR)* (2018).

[19] Yang, C., Wang, Z., Zhu, X., Huang, C., Shi, J. and Lin, D.: Pose Guided Human Video Generation, *Proc.of European Conference on Computer Vision(ECCV)* (2018).

[20] Zhou, Y. and Berg, T. L.: Learning Temporal Transformations From Time-Lapse Videos, *Proc.of European Conference on Computer Vision(ECCV)* (2016).

[21] Zhu, J.-Y., Park, T., Isola, P. and Efros, A. A.: Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networkss, *Proc.of IEEE International Conference on Computer Vision(ICCV)* (2017).