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

DAICHI HORITA\textsuperscript{1,a)} KEIJI YANAI\textsuperscript{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 \cite{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 \cite{3} to avoid generating blurred images caused by losing content details. In addition, the generator has spatial self-attention layers based on \cite{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 \cite{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.

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 \cite{18}, and (d) second stage of MD-GAN, respectively.

2. Related Work

2.1 Generative Adversarial Networks

Generative adversarial networks (GANs) \cite{1}, \cite{4} have achieved impressive results in image generation \cite{9}, \cite{13} and image-to-image translation \cite{8}, \cite{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 \cite{2}, \cite{19}. The other is a study to deal without such constraints \cite{14}, \cite{16}. MoCoGAN \cite{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

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

N.S. outside the neighborhood. In other words, it enables pixels and then gradually learn to assign more weight to learn the long-range dependence within a frame, which is 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 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 $x_l \in \mathbb{R}^{N \times C_l \times T_l \times H_l \times W_l}$, 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 $x_l$ and obtains $x_{l1} \in \mathbb{R}^{N \times C_l \times T_l \times H_l \times W_l}$ and (b) resizes to $\tilde{x}_{l1} \in \mathbb{R}^{N \times (H_l W_l) \times (C_l T_l)}$. Next, (c) the layer gets $x_{l2} \in \mathbb{R}^{N \times C_l \times T_l \times H_l \times W_l}$ by (a) the same operation and (d) resizes to $\tilde{x}_{l2} \in \mathbb{R}^{N \times (C_l T_l) \times (H_l W_l)}$. Furthermore, (e) after calculating the matrix multiplication of $\tilde{x}_{l1}$ and $\tilde{x}_{l2}$, (f) softmax calculate to obtain the attention $\hat{x}_{l1} \in \mathbb{R}^{N \times (H_l W_l) \times (H_l W_l)}$, defined as

$$
\hat{x}_{l1} = \frac{\exp(x_{l1})}{\sum \exp(x_{l1})}, \text{where } x_{l1} = x_{l1} \odot x_{l2}. \tag{1}
$$

This represents the weighted average inside the feature map. Following, (g) the layer applies the 3D convolution to the input feature $x_l$ and obtains $x_{l3} \in \mathbb{R}^{N \times C_l \times T_l \times H_l \times W_l}$ and (h) resizes to $\tilde{x}_{l3} \in \mathbb{R}^{N \times (C_l T_l) \times (H_l W_l)}$. Then, (i) the resized output of the layer is $o \in \mathbb{R}^{N \times C_l \times T_l \times H_l \times W_l}$, defined as

$$
o_l = \hat{x}_{l1} \odot x_{l3}. \tag{2}
$$

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

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

where $\gamma$ is a parameter initialized with 0. We leverage all 3D convolution of kernel 1 in the spatial self-attention layer and $\tilde{C}_l = C_l$ for all experiments.
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 are 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/
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://luv2019sasagan.github.io/.

### 7. Acknowledgements

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

### References


