SAVGAN: SELF-ATTENTION BASED GENERATION OF TUMOUR ON CHIP VIDEOS

Sandeep Manandhar[1]*, Irina Veith*[2,3]*, Maria Carla Parrini*[2,3]*, and Auguste Genovesio*[1*]

[1]IBENS, ENS, [2]Institut-Curie, CNRS, [3]INSERM,PSL Univ, 75005, Paris, France

ABSTRACT

Generation of videomicroscopy sequences will become increasingly important in order to train and evaluate dynamic image analysis methods. The latter are crucial to the study of biological dynamic processes such as tumour-immune cell interactions. However, current generative models developed in the context of natural image sequences employ either a single 3D (2D+time) convolutional neural network (CNN) based generator, which fails to capture long range interactions, or two separate (spatial and temporal) generators, which are unable to faithfully reproduce the morphology of moving objects. Here, we propose a self-attention based generative model for videomicroscopy sequences that aims to take into account for the full range of interactions within a spatio-temporal volume of 32 frames. To reduce the computational burden of such a strategy, we consider the Nyström approximation of the attention matrix. This approach leads to significant improvements in reproducing the structures and the proper motion of videomicroscopy sequences as assessed by a range of existing and proposed quantitative metrics.

***Index Terms*—** video generation, self-attention, Nyström approximation, tumour-on-chip

1. INTRODUCTION

Generation of video sequences is becoming increasingly important. Many tasks useful to biological research, such as videomicroscopy denoising, spatio-temporal super resolution, future frame prediction or video-to-video translation leverage advances in video generation. It will also become critical to create realistic synthetic videos to evaluate quantitative videomicroscopy analysis methods, or to augment experimental dynamic datasets that are often relatively small, in order to efficiently train deep network models.

Generative Adversarial Networks (GANs) have performed remarkably well in producing realistic looking high resolution images, but also text, audio and even genetic sequences. As we will see in further detail below, interesting research studies have followed in the field of unconditional video generation of natural images [1, 2, 3, 4, 5, 6]. However, generation of videos with realistic objects consistent along the sequence remains an unsolved problem . Possible difficulties may be that 1) databases of natural scene videos may be too sparse to allow the full capture of precise spatio-temporal features; 2) the information content of videos is distributed differently in spatial and temporal dimensions, but is tightly related and should then be learnt jointly; 3) learning motion of rigid and non-rigid objects requires capturing long range relationships in space and time; 4) videos are large data. Therefore, with constant memory size, increasing the considered number of consecutive frames for training, necessitates a decrease in the batch size and reversely.

In this work, we attempt to provide solutions to the aforementioned issues. 1) We restrict training to a narrow domain such as videomicroscopy sequences of tumor-on-chip [7]. These sequences were acquired in a fixed setup in the context of an experiment producing a dataset made of standardised videos containing similar objects with repetitive behaviours. This setup mitigates difficulties such as those encountered in large and sparse natural scene videos. Indeed, we believe that videomicroscopy often offers such a restricted context that it makes realistic video generation possible. 2) Our approach learns spatio-temporal features simultaneously from batches of spatio-temporal volumes. 3) We introduce a spatio-temporal self-attention mechanism in order to properly capture long range relationships in space and time. 4) We consider the Nyström approximation of the full spatio-temporal attention matrix to reduce the computational load. This savings in memory and computation allow us to consider a larger batch size of 2D+time volumes during training and thus helps to reach convergence.

In the following sections, we briefly describe state-of-the-art work in video generation, present our method, and report evaluation in comparison to two other methods following existing and proposed quantitative metrics.

2. RELATED WORK

An unconditional generative model takes as input a vector sampled from a reference distribution, and learns to map it to a data output. Early work in video GAN, such as VGAN [1], used two streams of 3D CNN generators to map a vector to a video clip. One generator synthesises a dynamic foreground, while the other one creates a static background. However, this approach led to rather poor results. Following this, a body of

*Correspondence: auguste.genovesio@ens.psl.eu

work proposed to split the generation of image from the generation of sequence. In TGAN, a sequence of vectors is generated by a temporal generator [2]. Each vector is then input to a sequential image generator to finally produce a video clip. In MocoGAN, a random sample vector is dedicated to the content, and a sequence of random sample vectors is dedicated to the motion [3]. A generator is then employed in a sequential manner to generate a video. Recently, TSGAN proposed yet another 2D generator for video frames complemented by a Gated Recurrent Unit (GRU) to encode the temporal variation [5]. In TGANv2, a memory efficient network was proposed where a LSTM based generator first produces a sequence of feature maps from a single latent vector. They are then transformed into a sequence of frames [6]. Most of these methods splitting spatial and dynamic domains still suffer from mode-collapse, require large memory, and produce inconsistent and low quality videos.

To deal with these inefficiencies, our work on video generation is inspired by SAGAN [8] that introduced a self-attention (SA) module for image generation. The SA mechanism introduces non-local behaviour by computing the similarities between all points in a given feature map. In contrast with convolution filters, which only consider local affinities, this module computes affinities between all feature points in a given feature map. Furthermore, unlike fully connected networks, this mechanism is only dependent on the input feature map, thus no further change in the loss function is required. For instance the authors of [9] inspired from work in [10] introduced the idea of SA in the context of video-classification. However, such an approach is heavy to compute for large feature maps such as spatio-temporal 3D volumes. For this reason, an approach such as SSA-GAN [11] combined 3D UNet with a spatially restricted SA module for future frame prediction.

A similar issue has been identified in natural language processing (NLP) research [12, 13, 14], where computing the full attention matrix is at $O(n^2)$ complexity cost. The authors of [15] recently proposed to compute affinities between a given set of feature points and an external learnable memory map which is much smaller than the given set. In our case, it was also crutial to keep a large video batch size in order to maintain quality. To this aim, we followed the work in [14], and proposed to approximate the spatio-temporal SA with the Nyström method which reduces the complexity to $O(n)$.

3. SELF-ATTENTION BASED VIDEO GENERATION

Our method (SAVGAN) consists of a 3D (video) generator and a 3D (video) discriminator that are both equipped with SA modules, and an additional 2D (image) discriminator that is fully convolutional. Moreover, we employ a Wasserstein loss with gradient penalty for training. The 3 networks and associated losses are described in the following sections.

3.1. Generator

We employ a single 3D (video) generator $\mathcal{G}_V$ that takes as input a d-dimensional vector $\mathbf{z}$ randomly drawn from a uniform distribution and outputs a video clip with size $T \times W \times H$, where T is the number of frames and W and H are the width and the height of the video, respectively. VGAN comprises 3D extension of DCGAN [1, 16]. However, a simple replacement of 2D kernels with 3D ones is unable to learn long range interactions of dynamic elements, leading to unrealistic videos where shapes remain barely consistent over time. To tackle this issue, we introduce a spatio-temporal attention module after a few layers of 3D convolutions. We design this architecture in order for the model to learn the association between distanced spatio-temporal points. $\mathcal{G}_V$ employs a series of 3D up-convolution layers, each followed by an instance normalization layer and a leaky rectified linear unit. We preferred up-convolution to upsampling layer in order to avoid interpolation artefacts. Approximated 3D SA layers stand at the end of the network, where the feature map size is close to the video output. The computation of the approximated SA of such a large feature map is described in section 3.3. We provide the details of $\mathcal{G}_V$ in the supplementary material.

3.2. Discriminator(s)

As in [17, 3, 18], we use two discriminators. A 3D discriminator $\mathcal{D}_V$ assesses the spatio-temporal consistency of the videos, and a 2D discriminator $\mathcal{D}_I$ assesses the spatial consistency of the generated frames. The architecture of $\mathcal{D}_V$ consists of a series of 3D convolution blocks, each followed by an instance normalization (IN) layer and a rectified linear unit (ReLU). A full SA module is placed towards the end of the network where the number of feature points is lower and does not necessitate computing an approximation. As in [8], the convolutional kernel weights are spectrally normalized to facilitate a stable training. $\mathcal{D}_V$ assesses the overall flow of the generated videos. On the other hand, $\mathcal{D}_I$ makes use of a series of 2D convolution blocks, each followed by an IN layer and a ReLU with no SA layer. $\mathcal{D}_I$ exclusively monitors the spatial consistency of each frame and thus improves the overall image quality. At each iteration, $\mathcal{D}_I$ is trained using a single frame randomly sampled from each video of the training batch. We provide the details of $\mathcal{D}_V$ and $\mathcal{D}_I$ in the supplementary material.

3.3. Self-Attention

Since it was first published in the context of NLP [10], many studies have benefited from the notion of SA [11, 9]. However, a SA matrix for a large set of feature points is expensive to compute. Furthermore, in the context of generative models, this memory usage plays against the need for a large batch size at training time in order to stabilise convergence.

Therefore, instead of computing the full SA matrix, we approximate its low-rank version. The low-rankness of a SA matrix has been been studied recently [19, 12, 14] and we have opted for the Nyström's approximation method to lower the computational burden [14].

Let $\mathbf{x}$ be k-dimensional spatio-temporal feature computed at the hidden layer l which precedes the SA layer. It is linearly projected to two C-dimensional feature spaces using functions $\mathbf{f}$ and $\mathbf{g}$ such that $\mathbf{f}(\mathbf{x}) = \mathbf{W}_f\mathbf{x}$ and $\mathbf{g}(\mathbf{x}) = \mathbf{W}_g\mathbf{x}$. Here, $\mathbf{W}_f \in \mathbb{R}^{C\times k}$ and $\mathbf{W}_g \in \mathbb{R}^{C\times k}$ are parameterised by $1 \times 1 \times 1$ learnable convolution kernels. Similarity between these projected feature points is obtained by the matrix product $\mathbf{A} = \mathbf{f}(\mathbf{x})^T\mathbf{g}(\mathbf{x})$. Then each element α_{ji} of the matrix $\mathbf{A}$ associates vectors $\mathbf{f}(\mathbf{x}_i)$ and $\mathbf{g}(\mathbf{x}_j)$ regardless of their position in the feature maps. A complete association between these vectors yields an attention matrix computed as:

$$\mathbf{S} = \sigma\left(\frac{\mathbf{A}}{\sqrt{C}})\right), \tag{1}$$

where σ is the row-wise softmax function. A full SA feature map $\mathbf{O}$ is computed by $\mathbf{O} = \mathbf{Sh}(\mathbf{x})$ where $\mathbf{h}(\mathbf{x}) = \mathbf{W}_h\mathbf{x}$ with $\mathbf{W}_h \in \mathbb{R}^{C\times k}$ that is implemented by $1 \times 1 \times 1$ learnable convolution kernels. Thanks to $\mathbf{A}$, non-local interactions in the feature map $\mathbf{h}(\mathbf{x})$ are considered while learning its weight $\mathbf{W}_h$. The final output of this module is given as $\mathbf{y} = \gamma\mathbf{O} + \mathbf{x}$, where γ is a learnable scalar value initialised as 0 in order to encourage learning from a local neighbourhood in the beginning [8].

At this point, $\mathbf{y}$ represents a spatio-temporal feature map that takes into account the whole range of interactions. However, the quadratic complexity of Eq. (1) is computationally strenuous when a large set of vectors is involved. To this end, we rely on the work of [14], where $\mathbf{A}$ is approximated with $\mathcal{O}(n)$ complexity in the context of NLP. The idea of this work is to select a few landmark $\mathbf{f}(\tilde{\mathbf{x}})$ and $\mathbf{g}(\tilde{\mathbf{x}})$, and compute only the similarities between these points and all the other non-landmark points. The remaining similarities are approximated via Nyström's method. Using this approach, Eq. (1) can be approximated as:

$$\tilde{\mathbf{S}} = \sigma\left(\frac{\mathbf{f}(\mathbf{x})^T\mathbf{g}(\tilde{\mathbf{x}})}{\sqrt{C}}\right)\sigma\left(\frac{\mathbf{f}(\tilde{\mathbf{x}})^T\mathbf{g}(\tilde{\mathbf{x}})}{\sqrt{C}}\right)^+\sigma\left(\frac{\mathbf{f}(\tilde{\mathbf{x}})^T\mathbf{g}(\mathbf{x})}{\sqrt{C}}\right), \tag{2}$$

where $(\mathbf{X})^+$ is the Moore-Penrose inverse of $\mathbf{X}$. The landmark points are sampled randomly at each iteration during the training (see supplementary material).

3.4. Training Loss

In order to stabilise the training, we used a Wasserstein loss with gradient penalty as proposed in [20] to enforce 1-Lipschtiz constraint. Let $\mathbb{P}_r$ and $\mathbb{P}_g$ be the probability distributions of the real and the generated video samples. Similarly, let $\mathbb{Q}_r$ and $\mathbb{Q}_g$ be the probability distributions of the real and the generated image frames. Then the losses to be minimized, with respect to $\mathcal{D}_V$ and $\mathcal{D}_I$ are given as:

$$L_V = \mathop{\mathbb{E}}_{\tilde{p}\sim\mathbb{P}_g}[\mathcal{D}_V(\tilde{p})] - \mathop{\mathbb{E}}_{p\sim\mathbb{P}_r}[\mathcal{D}_V(p)] + \lambda \mathop{\mathbb{E}}_{\hat{p}\sim\mathbb{P}_{\hat{p}}}[(\|\nabla_{\hat{p}}\mathcal{D}_V(\hat{p})\|_2 - 1)^2], \tag{3}$$

$$L_I = \mathop{\mathbb{E}}_{\tilde{q}\sim\mathbb{Q}_g}[\mathcal{D}_I(\tilde{q})] - \mathop{\mathbb{E}}_{q\sim\mathbb{Q}_r}[\mathcal{D}_I(q)] + \lambda \mathop{\mathbb{E}}_{\hat{q}\sim\mathbb{Q}_{\hat{q}}}[(\|\nabla_{\hat{q}}\mathcal{D}_I(\hat{q})\|_2 - 1)^2], \tag{4}$$

where, λ is the weight of the gradient penalty term empirically set to 10, and $\mathbb{P}_{\hat{p}}$ and $\mathbb{Q}_{\hat{q}}$ are distributions formed by uniform sampling of lines between pairs of points between $\mathbb{P}_r$ and $\mathbb{P}_g$, and $\mathbb{Q}_r$ and $\mathbb{Q}_g$, respectively. Finally, the training of $\mathcal{G}_V$ proceeds by maximizing the following equation:

$$L_G = \mathop{\mathbb{E}}_{\tilde{p}\sim\mathbb{P}_g}[\mathcal{D}_V(\tilde{p})] + \lambda_i \mathop{\mathbb{E}}_{\tilde{q}\sim\mathbb{Q}_g}[\mathcal{D}_I(\tilde{q})], \tag{5}$$

where λ_i is the weight given to the image discriminator, and empirically set to 50 in all our experiments.

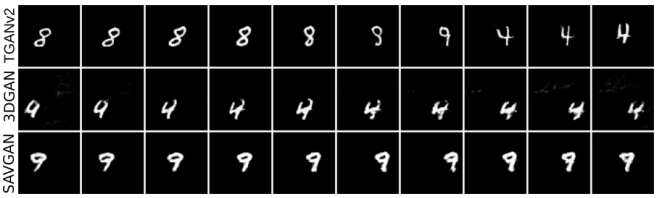

Fig. 1. From top to bottom row: frames generated by TGANv2, 3DGAN and SAVGAN. Note the spatial inconsistency in the sample generated by TGANv2 and in a lesser extend by 3DGAN.

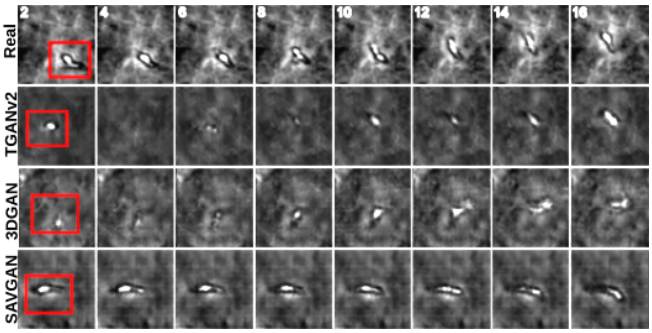

Fig. 2. Frames sampled from TOC datast. From top to bottom row: Real, generated by TGANv2, 3DGAN and SAVGAN. The red rectangles highlight the moving cell. Frame number is labelled on top.

4. EXPERIMENT

4.1. Datasets

Moving MNIST videos: this dataset consists of synthetic sequences where a digit moves rigidly in a linear horizontal trajectory while bouncing at the frame border. The training set is 2225 sequences each consisting of 10 frames of size 64×64.

Dataset	TOC			MMNIST		
Method	**TGANv2**	**3DGAN**	**SAVGAN**	**TGANv2**	**3DGAN**	**SAVGAN**
FVD	738.75	729.25	**389.3**	84.97	**80.4**	81.29
AACD	0.739	2.26	**0.477**	1.218	1.040	**0.469**
Flips/seq	-	-	-	3.42	3.1	**1.76**
Res18-cosTS	-	-	-	0.63	0.69	**0.76**

Table 1. Benchmark scores on generated sequences trained on TOC and MMNIST dataset by all three methods

Tumour-On-Chip (TOC) videos: TOC set consists of 3D co-cultures of lung cancer cells (A549 cell line) and cytotoxic T lymphocytes (H5B) at 1:1 ratio, embedded in collagen gel [21]. Phase-contrast image datasets were acquired using an automated video-microscope (Leica DMi8), every 10 seconds for 1 hour, using a 5x objective. From the acquired 4 large videos, we cropped square video patches of size $32 \times 64 \times 64$ around moving lymphocytes detected by thresholding the motion magnitude computed using [22]. The training set consists of 1224 videos after further removal of out-of-focus cells.

4.2. Training

We trained our model using the Adam optimizer with parameters $(\beta_1, \beta_2) = (0, 0.9)$. We used 0.0001 and 0.0004 as the learning rates respectively for $\mathcal{G}_V$ and both discriminators. We updated the training gradients of ($\mathcal{D}_V$, $\mathcal{D}_I$) and $\mathcal{G}_V$ at an iteration ratio of $5 : 1$ as suggested in [23]. We compared SAVGAN with TGANv2 [6], and as an ablation study, with a version of SAVGAN without the SA layers, which we called 3DGAN. The training is done for 300K iteration steps for SAVGAN and 3DGAN, and 200K iteration steps for TGANv2. For both datasets MNIST and TOC, we used training batch sizes of 32 for TGANv2, and 8 for 3DGAN and SAVGAN.

4.3. Evaluation metrics

Frechet video distance (FVD): the metric proposed in [24] computes the Frechet Distance between two batches of 256 real and generated samples.
Average ACD: this metric inspired from [3] computes the average over all frames of the L2 distances between the average intensities of any two consecutive frames.
Flips/seq: we propose this metric that assesses the consistency of the sequence semantic overtime. It is only applicable to the MMNIST dataset. It counts the number of times the rendered digit changes throughout a generated MMNIST video as assessed by a pretrained MMNIST frame classifier. The score is computed over samples containing 10 frames, 0 being the best possible score.
Res-18 cosTS: we propose this metric to assess the temporal continuity between pairs of consecutive frames. It is also only suited to the MMNIST dataset. It computes the mean cosine similarity in a 512 dimensional representation (obtained by the last layer of a Resnet-18 pretrained to classify the digits between $0 - 9$ with 95% accuracy) between a batch of 256 images and their consecutive frames.

4.4. Results

FVD results in Table 1 indicate a clear improvement on the TOC dataset by SAVGAN over the two other methods. However, FVD was not conclusive when applied to the MMNIST dataset (Figure 1). The reason could be that FVD is trained on natural images which might not be well suited to evaluate results on MMNIST. On the other hand, AACD, a simpler metric not depending on any training data, was conclusive for both datasets and demonstrated stability of SAVGAN over the two other approaches.

Furthermore, by eyeballing the results in the TOC dataset (Figure 2), we also noticed that one of the improvements brought by the use of SA layers in SAVGAN over 3DGAN, was that it produces a better consistency of the video over time. To evaluate this hypothesis, we proposed Flips/seq and Res-18 cosTS, two additional metrics based on MMNIST to quantify how semantics of objects (digits in this case) are preserved over time and how continuous the generated sequence of images is. Both metrics demonstrated a clear improvement of SAVGAN over the other two approaches (Table 1 and supplementary material for more examples). Finally, the GPU memory consumption of full vs approximated SA layers were respectively 28 GB and 2.4 GB for a training batch size of 4.

5. CONCLUSION

In this work, we proposed SAVGAN, a self-attention based video generative adversarial network. The architecture is made up of a mix of convolutional and self-attention modules to better capture spatio-temporal interactions. Furthermore, in order to reduce the computational and memory load, we implemented a Nyström approximation of the full self-attention matrix. Quantitative results demonstrated a better preservation of structures and semantic over time for rigid objects. The distribution of generated movie was found closer to the real distribution for biological video as assessed by FVD. We believe SAVGAN provides a robust method to generate consistent videos and should benefit other tasks of interest such as domain transfer or segmentation in biological imaging analysis, and possibly in other fields.

6. ACKNOWLEDGMENTS

This work was supported by INSERM ITMO N°19CR046-00, ANR–10–LABX–54 MEMOLIFE and ANR–10–IDEX –0001–02 PSL* Université Paris and was granted access to the HPC resources of IDRIS under the allocation 2020-AD011011495 made by GENCI.

7. REFERENCES

[1] C. Vondrick et al., "Generating videos with scene dynamics," in *NeurIPS*, 2016.

[2] M. Saito et al., "Temporal generative adversarial nets with singular value clipping," in *ICCV*, 2017.

[3] S. Tulyakov et al., "MoCoGAN: Decomposing motion and content for video generation," in *CVPR*, 2018.

[4] A. Clark et al., "Efficient video generation on complex datasets," *ArXiv*, vol. abs/1907.06571, 2019.

[5] A. Munoz et al., "Temporal shift GAN for large scale video generation," in *WACV*, 2021.

[6] M. Saito et al., "Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal GAN," in *IJCV*, 2020.

[7] M. Nguyen et al., "Dissecting effects of anti-cancer drugs and cancer-associated fibroblasts by on-chip reconstitution of immunocompetent tumor microenvironments," *Cell Reports*, vol. 25, no. 13, pp. 3884–3893.e3, 2018.

[8] H .Zhang et al., "Self-attention generative adversarial networks," in *ICML*, 2019.

[9] X. Wang et al., "Non-local neural networks," in *CVPR*, 2018.

[10] A. Vaswani et al., "Attention is all you need," in *NeurIPS*, 2017.

[11] H. Daichi and Y. Keiji, "SSA-GAN: End-to-end time-lapse video generation with spatial self-attention," in *ACPR*, 2019.

[12] S. Wang et al., "Linformer: Self-attention with linear complexity," 2020.

[13] N. Kitaev et al., "Reformer: The efficient transformer," in *ICLR*, 2020.

[14] Y. Xiong et al., "Nyströmformer: A nyström-based algorithm for approximating self-attention," in *AAAI*, 2021.

[15] M. Guo et al., "Beyond self-attention: External attention using two linear layers for visual tasks," *CoRR*, vol. abs/2105.02358, 2021.

[16] A. Radford et al., "Unsupervised representation learning with deep convolutional generative adversarial networks," in *ICLR*, 2016.

[17] I. Durugkar et al., "Generative multiadversarial networks," in *ICLR*, 2017.

[18] T. Wang et al., "Video-to-video synthesis," in *NeurIPS*, 2018.

[19] Y. Dong et al., "Attention is not all you need, pure attention loses rank doubly exponentially with depth," *CoRR*, vol. abs/2103.03404, 2021.

[20] I. Gulrajani et al., "Improved training of wasserstein GANs," in *NeurIPS*, 2017.

[21] I. Veith et al., "Apoptosis mapping in space and time of 3d tumor ecosystems reveals transmissibility of cytotoxic cancer death," *PLoS Comput Biol.*, vol. 17(3), 2021.

[22] C. Zach et al., "A duality based approach for realtime tv-l1 optical flow," in *Pattern Recognition*, Berlin, Heidelberg, 2007, pp. 214–223.

[23] T. Miyato et al., "Spectral normalization for generative adversarial networks," in *ICLR*, 2018.

[24] T. Unterthiner et al., "Towards accurate generative models of video: A new metric & challenges," *CoRR*, vol. abs/1812.01717, 2018.