Non-Adversarial Video Synthesis with Generative Latent Nearest Neighbors

Daniel Afrimi
Department of Computer Science
Hebrew University Of Jerusalem
daniel.afrimi@mail.huji.ac.il

Abstract

Generative video models can impact many applications (e.g., future prediction) in video understanding and simulation. We propose a generative non-adversarial network for video based on GLANN [1] model, with spatio-temporal convolutional architecture that untangles the scene's foreground from the background which is based on MIT [2] model. The experiments showed that the model provides good but unsatisfactory results.

1 Introduction

The extension of image generation to video generation turns out to be a very difficult task, since the temporal dimension of videos introduces an extra challenge during the generation process, in addition to the visual presentation of objects.. Besides, due to the limitation of memory and training stability, the generation becomes increasingly challenging with the increase of the resolution/duration of videos.

Learning to generate future frames of a video sequence is a challenging research problem with great relevance to reinforcement learning, planning and robotics. Although impressive generative models of still images have been demonstrated, these techniques do not extend to video sequences. The main issue is the inherent uncertainty in the dynamics of the world.

2 Previous Work

Generative Modeling: Generative modeling of videos is a long-standing problem of wide applicability.

Adversarial Generative Models: Generative Adversarial Networks (GANs) were first introduced by Goodfellow[3]. GANs have shown a remarkable capability for image generation but steal an open topic in video generation - GANs suffer from difficult training and mode dropping.

Non-Adversarial Methods: The disadvantages of GANs motivated research into GAN alternatives. GLO, a recently introduced encoder-less generative model which uses a non-adversarial loss function, gets better results than VAEs. Due to the lack of a good sampling procedure, it does not outperform GANs. IMLE, a method related to ICP was also introduced for training unconditional generative models, however due to computational challenges and the choice of metric, it also does not outperform GANs.

Preprint. Under review.

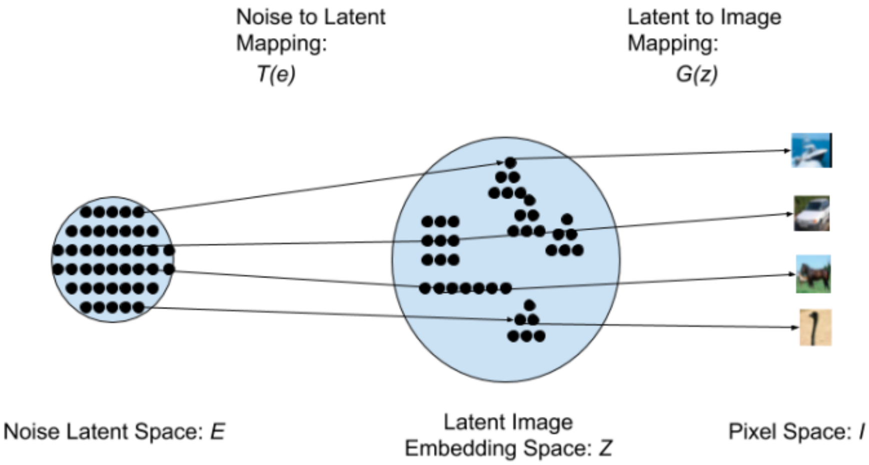

Figure 1: An illustration of GLANN architecture: a random noise vector e is sampled and mapped to the latent space to yield latent code $z = T(e)$. The latent code is projected by the generator to yield image $I = G(z)$. © GLANN paper.

2.1 MIT Model - VGAN

Previous work has focused mostly on small patches, and evaluated it for video clustering. In the paper of MIT they develop a generative video model for natural scenes using state-of-the-art adversarial learning methods. In their model propuse they leverage methods for recognizing actions in video with deep networks, and apply them for video generation instead. Using spatio-temporal 3D convolutions to model videos, but they use fractionally strided convolutions for generation. They also use two-streams to model video, but apply them for video generation instead of action recognition.

Their experiments showed several examples of the videos generated from their model (Figure 3). their observation were that the generated scenes tend to be fairly sharp and that the motion patterns are generally correct for their respective scene.

2.2 GLANN

Our model based on GLANN - A model for synthesizing high-quality images without using GANs. which contains two different models - GLO and IMLE.

Generative Latent Optimization (GLO):

GLO model proposed in the paper Optimizing the Latent Space of Generative Networks [4]. GLO is a generative model which enjoys many of the desirable properties of GANs including modeling data distributions, generating realistic samples, interpretable latent space, but more importantly, it doesn't suffer from unstable adversarial training dynamics.

The GLO optimization objective is written in Eq. 1:

$$(1)\ argmin_{G,\{z_i\}} \sum_i \ell(G(z_i), x_i) s.t. \|z_i\| = 1$$

In addition GLO enforcing all latent vectors to be on a unit sphere or a unit ball, replacing the linear matrix W, by a deep CNN generator $G()$ which is more suitable for modeling images and using a Laplacian pyramid loss function (in GLANN paper they found that VGG perceptual loss works better)

Implicit Maximum Likelihood Estimation (IMLE):

Implicit probabilistic models are models defined naturally in terms of a sampling procedure and often induces a likelihood function that cannot be expressed explicitly[5]. Li and Malik developed a simple method for estimating parameters in implicit models that does not require knowledge of the form of

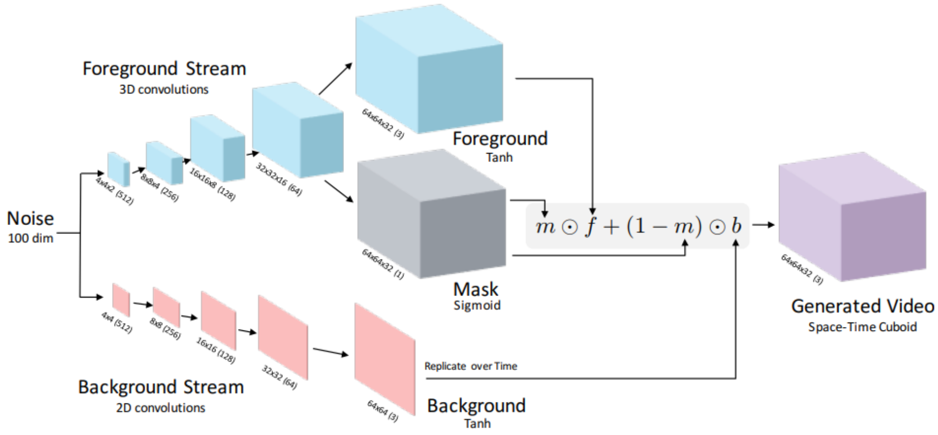

Figure 2: **Video Generator Network:** Two Stream Architecture. The input is 100 dimensional (Gaussian noise). There are two independent streams: a moving foreground pathway of fractionally-strided spatio-temporal convolutions, and a static background pathway of fractionally-strided spatial convolutions, both of which up-sample. © MIT paper.

the likelihood function or any derived quantities, but can be shown to be equivalent to maximizing likelihood under some conditions.

IMLE was proposed for training generative models by sampling a large number of latent codes from an arbitrary distribution, mapping each to the image domain using a trained generator and ensuring that for every training image there exists a generated image which is near to it. IMLE is trivial to sample from and does not suffer from mode-dropping. Like other nearest neighbor methods, IMLE is sensitive to the exact metric used, particularly given that the training set is finite.

2.3 Limitations of GLO and IMLE

The main problem with generating images with GLO is that the generator is not trained to sample from any known distribution (i.e sampling latent variables from a normal distribution, generations that are of much lower quality than GANs are usually obtained).

However, sampling from an IMLE trained generator is trivial, the training is not, a good metric might not be known

GLANN Combines the good properties of both models and generate qualitative images (see figure 1).

3 Methods

The methods that were used in our model rely mainly on GLANN.

3.1 GLO

As mentioned before GLO try to minimize the formula in eq 1. Unlike GLANN, the network is not a regular DNN, but a network based on the MIT work (see 2.3 figure 2), which use two-streams to model video (foreground and background). While keeping the same net as MIT Suggest, we also tried to expand the network for learning more parameters and to get better results.

3.1.1 Two-streams Architecture

This architecture enforces a static background and moving foreground. using a two-stream architecture where the generator is determine by the combination:

$$G_2(z) = m(z) \odot f(z) + (1 - m(z)) \odot b(z)$$

$m(z)$ can be viewed as a spatio-temporal mask that selects either the foreground $f(z)$ model or the background model $b(z)$ for each pixel location and timestep. By summing the foreground model with the background model, the final generation can happend (there is an option to add to the objective a small sparsity prior on the mask for encourage the network to use the background stream).

Moreover, there are fractionally strided convolutional networks for $m(z)$, $f(z)$, and $b(z)$. and the generator produces 64 x 64 videos for 32 frames.

3.2 IMLE

Instead of using IMLE model for map noise vectors to a latent space, we tried to use another models for this mapping (i.e.VAE [7] and Gaussian Mixture Model (GMM)).

3.3 Perceptual losses

In instances where we want to know if two images look like each-other, we could use a mathematical equation to compare the images but this is unlikely to produce good results. Two images can look the same to humans but be very different mathematically (i.e. if there is a picture of a man vs the same picture of the man but the man is shifted one pixel to the left). Using a perceptual loss function solves this issue by taking a neural network that recognizes features of the image; these can include autoencoders, image classifiers, etc.

As mentioned in the paper of GLANN, they used the VGG perceptual loss for minimize the error in GLO model, while all parameters are optimized directly by SGD. The VGG loss function is a pretrained network for image classification[8]. However, our model supposed to generate video, so other perceptual losses will fit better.

The perceptual losses that we used are:

1. Resnext-101: 3D model for action recognition[9] (pretrained model on the Kinetics dataset). `https://github.com/kenshohara/3D-ResNets-PyTorch`).

2. Resnet-50: 3D model for action recognition[10] (pretrained model on the Kinetics dataset. `https://github.com/kenshohara/3D-ResNets-PyTorch`).

3. VGG16: calculate the loss on each frame on the generated video and we sum it up for one error (like glann, but instead of calculating the loss on the entire video we took each frame and calculate the error).

4. Laplacian loss: this loss is not a perceptual one, but we tried to use it like in GLO model. we calculate the loss on each frame on the generated video and we sum it up for one error .

5. L1 loss: simple L1 loss, this loss is not a perceptual one, but we tried to use it.

4 Experiments

We experiment with the generative non-adversarial network for video (VGLANN) on video generation task.

We used the “golf” dataset (50K videos - from examination the dataset contains random videos) that MIT published in their github repository (`https://github.com/cvondrick/videogan`). The datasets downloaded from Flickr by querying for popular Flickr tags as well as querying for common English words.

Stabilization: Because we are interested in object movements and not camera moves, the dataset that MIT released has been stabilized by extracting SIFT keypoints, use RANSAC to estimate a homography between adjacent frames, and warp frames to minimize background motion.We use 32-frame videos of spatial resolution 64 x 64.

4.1 Learning and Implementation

We trained the generator (GLO) and IMLE with stochastic gradient descent (all networks are trained from scratch). We used the Adam optimizer and a learning rate of 0.01. The latent code has 100

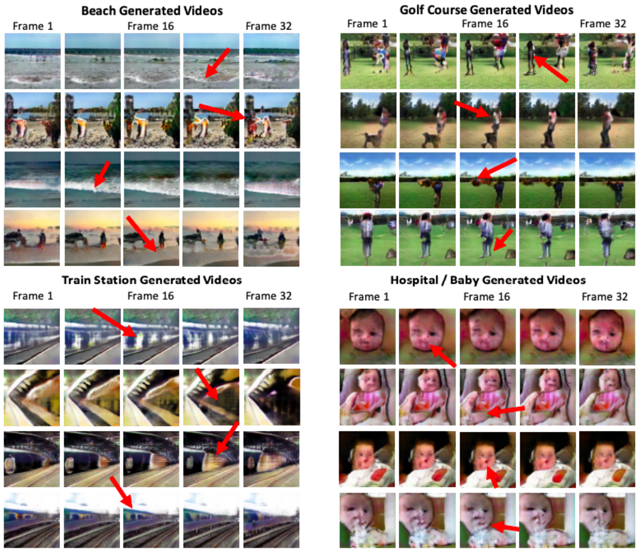

Figure 3: **Video Generations:**some generations from the two-stream model of MIT. The red arrows highlight motions. © MIT paper.

Table 1: FVD Scores - Mapping models from noise space to a latent space (with Original model)

Model	FVD Score - **Laplacian loss**	FVD Score - **Resnext-101 perceptual loss**
IMLE	±2651.40	±2305.44
VAE	±2891.24	±2421.96
GMM	±2700.96	±2587.32

dimensions, which we sampled from a normal distribution. We used a batch size of 15 and 200 epochs for GLO model. We use batch normalization followed by the ReLU activation functions after every layer in the generator, except the output layers, which uses tanh. For VAE, IMLE model we used 200 and 100 epochs respectively.

We used Frechet Video Distance (FVD) [6] mertic to evaluate our generated videos. Other tasks of genereating videos like MIT used the metric of showing a worker two random videos and ask them "Which video is more realistic?" (collected over 13, 000 opinions across 150 unique workers). this kind of metric is something that can be used in our model too.

4.2 Frechet Video Distance

FVD builds on the principles underlying Frechet Inception Distance (FID), which is a good metric for image generation. FVD is a different feature representation that captures the temporal coherence of a video, in addition to the quality of each frame. Unlike popular metrics such as the Peak Signal to Noise Ratio (PSNR) or the Structural Similarity (SSIM) index, FVD considers a distribution over entire videos, thereby avoiding the drawbacks of frame-level metrics

4.3 Results

We experiment the result with the original video generator as mention in MIT work and with different losses - Laplacian Loss and Resnext-101.

4.4 Original Model

see results in the next sections figures.

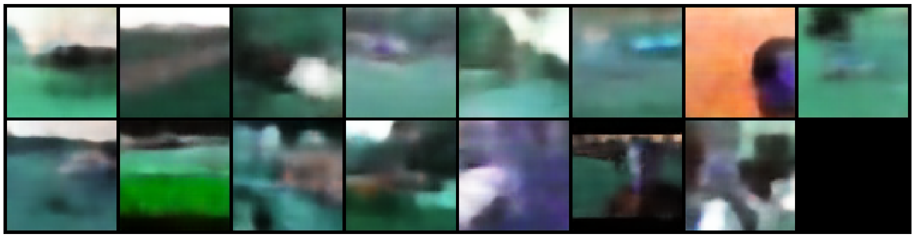

Figure 4: Reconstruction of GLO model (Laplacian loss) - one frame out of 32 frames for each video

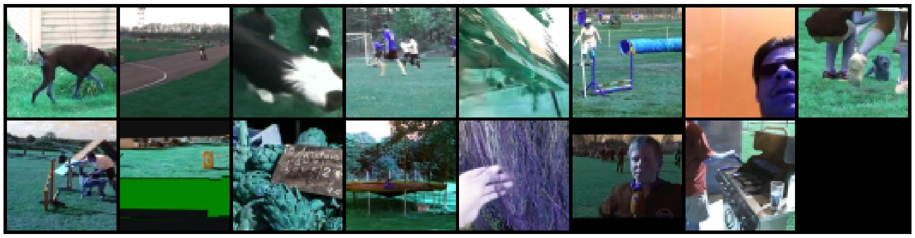

Figure 5: Original videos - one frame out of 32 frames for each video

4.4.1 Laplacian Loss in GLO model

see results in figures 4,5 and 8 (error was 2.500929, 200 epochs). The training was 4 days on 3 GPU's.

4.4.2 Resnext-101 Perceptual Loss in GLO model

see results in figures 6, 7 and 9 (error was 0.170761, 184 epochs, ended after a week because TIMEOUT on cluster school). The training was on 3 GPU's.

4.5 Expanded Model

NOT DONE YET - still running on school cluster

5 Discussion

Loss Function: In this work, we replaced the standard adversarial loss function by a perceptual loss for video. Instead of using the ImageNet-trained VGG features, we used resnet-50 and resnext-101 - a 3D model for action recognition.

Mapping between spaces: We examine few mapping models from the noise space to a latent space. In this project we tried IMLE, VAE and GMM as a mapping functions. For future work we can try another models like invertible flow models (like Invertible ResNets) and autoregressive models (like pixelcnn++).

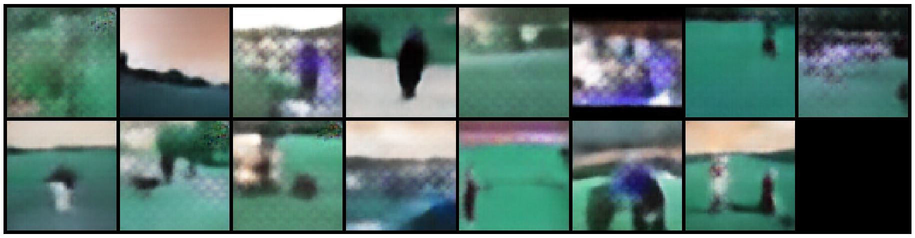

Figure 6: Reconstruction of GLO model (Resnext-101 loss) - one frame out of 32 frames for each video

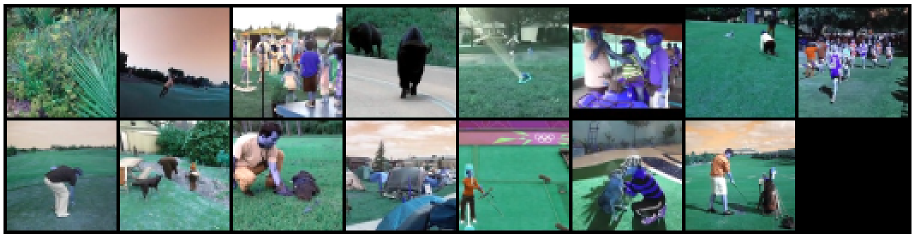

Figure 7: Original videos - one frame out of 32 frames for each video

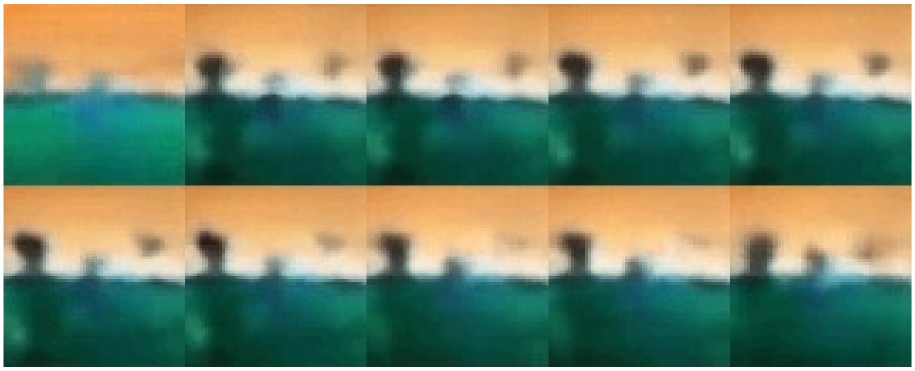

Figure 8: Generated video with VAE model (Laplacian Loss on GLO)

6 Conclusions

In this paper we introduced a non-adversarial model for generating videos. For the proposed model, it looks like that the adversarial model of MIT works better. From testing it seems that the main problem is in GLO - the reconstructions of the videos not done perfectly (unlike in reconstructions images in the original paper) - i.e Even after an optimization has occurred for each video in data set and for each latent code the reconstruction not done well. In my opinion, if the training was done better - the generation of video by mapping noise vector to a latent space (IMLE) and than mapping it to the video space (GLO) - the results were more satisfying than MIT's model (our model does no suffer from mode dropping and difficult in training, especially they used more data than us).

Possibly that even if the reconstructions of GLO were done perfectly, the mapping with IMLE/VAE/GMM model Would encounter with difficulties. if the noise vector mapping does not work perfectly we could use another model for mapping like we mention before.

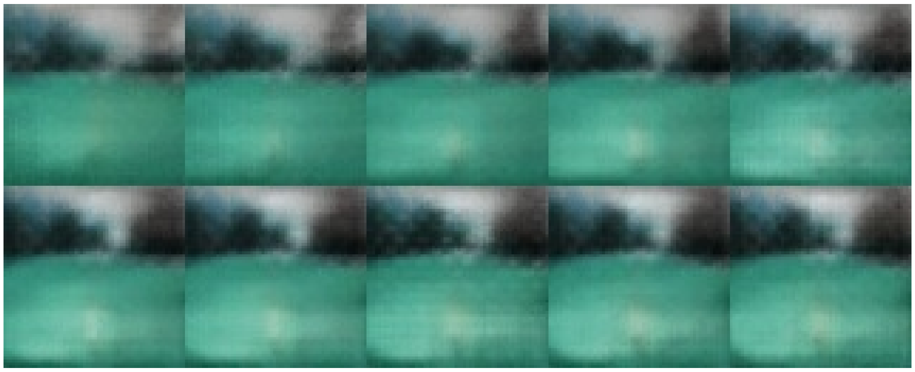

Figure 9: Generated video with VAE model (Resnext-101 on GLO)

References

[1] Yedid hoshen. & Jitendra MalikṄon-Adversarial Image Synthesis with Generative Latent Nearest Neighbors.

[2] Carl Vondrick & Hamed Pirsiavash & Antonio Torralba. MIT. Generating Videos with Scene Dynamics.

[3]I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, pages 2672–2680, 2014.

[4] P. Bojanowski, A. Joulin, D. Lopez-Paz, and A. Szlam. Optimizing the latent space of generative networks. In ICML, 2018.

[5] K. Li and J. Malik. Implicit maximum likelihood estimation. arXiv preprint arXiv:1809.09087, 2018

[6] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, Sylvain Gelly. Towards Accurate Generative Models of Video: A New Metric & Challenges.

[7] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In ICLR, 2014.

[8] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. arXiv preprint arXiv:1801.03924, 2018

[9] Kensho Hara, Hirokatsu Kataoka, Yutaka Satoh National Institute of Advanced Industrial Science and Technology (AIST). Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?

[10] Kensho Hara, Hirokatsu Kataoka, Yutaka Satoh National Institute of Advanced Industrial Science and Technology (AIST). Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?