Action-Conditioned Frame Prediction Without Discriminator

David Valencia[1], Henry Williams[1], Bruce MacDonald[1], and Ting Qiao[1]

Centre for Automation and Robotic Engineering Science, University of Auckland, Auckland, New Zealand
dval@35@aucklanduni.ac.nz, henry.williams@auckland.ac.nz

Abstract. Predicting high-quality images that depend on past images and external events is a challenge in computer vision. Prior proposals have tried to solve this problem; however, their architectures are complex, unstable, or difficult to train. This paper presents an action-conditioned network based upon Introspective Variational Autoencoder(IntroVAE) with a simplistic design to predict high-quality samples. The proposed architecture combines features of Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs) with encoding and decoding layers that can self-evaluate the quality of predicted frames; no extra discriminator network is needed in our framework. Experimental results with two data sets show that the proposed architecture could be applied to small and large images. Our predicted samples are comparable to the state-of-the-art GAN-based networks.

Keywords: Action Conditioned · Deep Learning · Frame Prediction · Generative Models · Variational Autoencoders.

1 Introduction

Humans have the ability to solve problems and understand the surrounding area through visual perception, which allows them to make decisions and predict upcoming events with great precision and speed. Video frame prediction is one way to model this human behaviour from a machine perspective. Frame prediction has been studied for years and applied in areas such as autonomous driving cars[32], robotic manipulation[20,5], trajectory predictions [23], or physical interaction[6]. Predicting upcoming events provides the possibility to plan actions and an understanding of the environment; however, predicting the future is not an easy task since it not only depends on past events but also sometimes on external actions, input controls, or complex high dimensional features.

If we talk about image prediction, we are also talking about image generation, where deep learning approaches have shown great results in recent years. Neural networks can be trained to generate and predict future frames, given the current camera frame and external actions. Much research has been carried out on generative models; the most typical and popular models based their network architectures on Variational Autoencoders (VAEs)[12], and Generative Adversarial

Networks (GANs)[7]. However, these models have complex and unstable architectures with serious limitations that often result in poor predictions. This paper proposes a novel neural network architecture for action-conditioned frame prediction based upon Introspective Variational Autoencoder(IntroVAE)[10] that combines features of VAEs and GANs in a more simplistic design which can self-evaluate the quality of predicted frames. The purpose of this article is to present a network easy to train to produce highly detailed samples with minimal blurriness and increase the network's stability overcoming the VAE and GAN limitations. The proposed model can be applied in both small size images as well as large size images with high details. To the best of our knowledge, it is the first work that applies the introspective manner to action-conditioned frame prediction. This article is organized as follows: In Section 2, a brief background and similar projects are reviewed. In section 3, the overall design of the proposed system is described. In section 4, the datasets, along with the experiments and results, are analyzed. Finally, in section 5, the conclusions and future works are presented.

2 Background

Recent years have seen an increase in studies related to predicting future frames based on neural networks. The current literature suggests that the frame prediction concept could be decomposed into two groups: first, an image generator, by learning a latent distribution of the original samples, new samples can be created, and second, a future frame generator, where the next frames depend not only on previous frames but also on actions or external features. Below is an analysis of related literature within these two groups.

2.1 Image Generator

Generative models have traditionally been used as anomaly detection[26], image completion, super-resolution[14,9], or as a way to learn representations of images or videos. There have been a number of promising approaches for image generation developed previous to this paper; for example, VAE is a well-known algorithm that provides an attractive solution to the image generation problem by learning a latent representation of the data. VAE architecture is comprised of two networks, an encoder and a decoder. The encoder translates from the input x (e.g., an image) to a low-dimensional representation vector z called the latent representation. The decoder takes as input a latent sample z randomly sampled from a prior distribution (e.g., normal distribution) and produces samples in the domain of the input x. VAE generally applies Kullback–Leibler divergence [16] and pixel-wise error as loss functions during the training process. VAE is stable, easy to train, and computationally inexpensive; however, it has a strong drawback; the generated images lack details and tend to have high blurriness levels. GANs comes as a viable solution to the low-quality output from VAEs. GANs

consist of two networks, a generator G and a discriminator D. The generator receives as input a latent sample z and produces a sample $G(z)$. The discriminator takes as input both the generated image $G(z)$ and the input image x and tries to differentiate real data from generated samples; meanwhile, the generator tries to produce better images to fool the discriminator playing a mix-max game based upon the principle of game theory. Although GANs have notoriously bettered other alternatives in producing sharp images, training GANs is not always easy, especially when handling high-resolution images. GANs may face challenges in training stability and sampling diversity[2,26,11]. Also balancing the convergence of the discriminator and the generator could be a difficult task even with several tricks applied[2,21,25].

There are also some hybrid models such as [24,13,2,33,11] that try to reduce the instability of GANs and improving the blurring typical of VAEs. These approaches generally use an extra discriminator in their architecture to add an adversarial constraint and improve the generated images' quality. However, the majority of the current literature notes that adding a discriminator to the network may result in some challenges, such as increasing the network's complexity, high probabilities of mode collapse (i.e. produces limited varieties of samples in the output[33]), or high sensitivity to the hyper-parameters.

2.2 Frame Prediction

Trying to predict future video frames is one of the areas that has had the most interest in recent years, both recurrent and feedforward models are widely used in this area(see [3,19], and [22] for survey). Several works have been presented trying to produce a sequence of future frames, for example,[8] and [17] present a combination of VAE and recurrent neural network for frames prediction from an initial seed image. However, these approaches are implemented using small-sized images (around 64 x 64 pixels) where their predictions are not of the best quality as they lack detail. A Stochastic Adversarial Video Prediction (SAVP) is present in [15] which improves the prediction quality by introducing adversarial terms to the loss and models pixel motion. A Convolutional Dynamic Neural Advection (CDNA) model that can predict various futures of an object conditioned on the action of an agent is presented by Finn et al., in [6]. Walker et al.,[27] combine the advantages of VAE with those of GANs for video frame forecasting of human pose, while Wang et al., present in [28] an actioned conditioned frame prediction in Atari video games where four consecutive frames are concatenated with the actions and passed as inputs to a convolutional autoencoder. In [18], a similar strategy is presented, but a recurrent encoding network is also tested as a second architecture. These proposals' predictions are accurate, but it must be considered that Atari environments are fixed tasks; in other words, the environments in Atari are the same in each game, and the size of each frame is relatively small; therefore, these methods may not be applied with more complex data. For these reasons, our main contribution is a novel model that can handle different sizes of images to predict high-quality samples without using an external discriminator that performs better than the state-of-the-art image-predictor methods.

3 Methodology

3.1 We do not Need a Discriminator

To predicted clear and high-quality images, we propose a model architecture as shown in Fig. 1. Our design consists of two components: an encoder E and a Generator G. No extra discriminator is needed in our proposal since the encoder here also plays the role of a discriminator. Not having an extra discriminator makes our network considerably more stable and easier to train compared to GAN or Hybrid-GAN architectures. This idea of *re-use* the encoder as a discriminator was initially proposed by Huang et al., and called IntroVAE[10]; its encoder and generator are trained in an introspective way. Inspired by this idea, we modified IntroVAE at several points. First, IntroVAE was originally conceived as an image generator/reconstructor but not as an image-predictor; consequently, the loss functions must be redefined. Second, our model is an action-conditioned image predictor where future frames also depend on external actions; therefore, an action vector needs to be added as a second input. Finally, our architecture is considerably simpler; we follow the same principle of operation of a convolutional variational autoencoder. See Fig. 1.

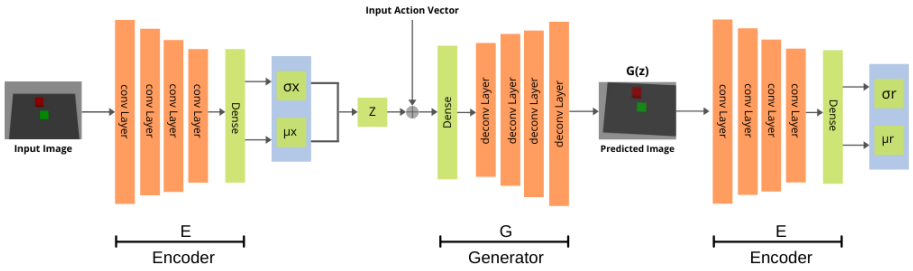


Fig. 1: Proposed Network Architecture.

As mentioned beforehand, our framework consists of an encoder network and a generator network (analogous to an encoder and a decoder in a VAEs). Our aim is to learn a function f capable of predicts the next frame x_{t+1}, receiving as input a previous frame x_t and an action a_t.

$$f : x_t, a_t \longmapsto x_{t+1} \tag{1}$$

Our encoder network E plays two roles: as the encoder of VAEs for real samples and as the discriminator of GANs for generated samples. On the other hand, our generator network G works the same role as a generator of GANs. We train our model following the same adversarial game idea of GANs, but in this case, the encoder learns how to distinguish between the real data from the generated samples, while the generator tries its best to produce more realistic samples to fool the encoder [9].

The encoder E takes an input image and encodes it into a smaller hidden representation, then outputs two individual vectors, one representing the mean values μ, and one denoting the standard deviations σ. Motivated by energy-based GANs [29], the encoder is trained to perform two tasks simultaneously; first, to minimize the prior regularization term $D_{KL}(q_\phi(z|x)||p(z))$ - where D_{KL} denotes Kullback-Leibler divergence - to encourage the posterior $q_\phi(z|x)$ to match the prior $p(z)$, and second, to maximize the prior regularization term to encourage the posterior $q_\phi(z|G(z))$ of the generated samples $G(z)$ to deviate from the prior $p(z)$. On the other hand, the generator G is trained to produce samples that have a small D_{KL}, such that the generated samples' posterior distribution matches the prior distribution $p(z)$ [31,10]. The input z of the generator G is generally sampled from $N(\mu, \sigma)$.

Therefore, given a real data sample x, an action vector a, the losses to train the encoder E and the generator G are designed as:

$$L_E(x, z, a) = D_{KL}(q_\phi(z|x)||p(z)) + [m - D_{KL}(q_\phi(z|G(z, a))||p(z))]^+ \tag{2}$$

$$L_G(z, a) = D_{KL}(q_\phi(z|G(z, a))||p(z)) \tag{3}$$

where m is a positive constant, $[.]^+ = max(0, .)$ and the prior probability is described following the original VAE notation where $p(z)$ is sampled from a know distribution, for instance $N(0, 1)$. The equation (2) and (3) form a min-max game between the encoder network E and the generator network G aligning the generated and true distributions producing sharp samples. However, training the model in this adversarial manner is the main cause of the difficulties related to GANs, such as training instability, non-convergence, or mode collapse. As mentioned in [10], to solve these obstacles, the simpler but efficient way is to build a bridge between the encoder E and generator G, adding the reconstruction error L_{AE} to equations (2) and (3) as follow:

$$L_E(x, z, a) = D_{KL}(q_\phi(z|x)||p(z)) + [m - D_{KL}(q_\phi(z|G(z, a))||p(z))]^+ + L_{AE}(x) \tag{4}$$

$$L_G(z, a) = D_{KL}(q_\phi(z|G(z, a))||p(z)) + L_{AE}(x) \tag{5}$$

The input z of the generator network G is sampled from $N(\mu, \sigma)$ using the method of reparameterization trick, where μ and σ are the outputs of the encoder network E. Therefore, the posterior probability could be denoted by $q_\phi(z|x) = N(z; \mu, \sigma)$. Then under these parameters and given N data samples, the $D_{KL}(q_\phi(z|x)||p(z))$ - denoted as L_{REG} for simplicity of notation - can be computed as follows:

$$L_{REG}(z; \mu, \sigma) = -\frac{1}{2}\sum_{i=1}^{N}(1 + log(\sigma_i^2) - \mu_i^2 - \sigma_i^2) \tag{6}$$

The reconstruction error L_{AE}, which measures the difference between the target image (denoted by y) and the predicted image (denoted by x_r), is expressed by the pixel-wise mean squared error (MSE) function. Note that we measure MSE with respect to the next ground-truth image(target image) instead of the input image x since we are not reconstructing the same image; this is the main difference from the original IntroVAE framework. The reconstruction error can be computed as below:

$$L_{AE}(x_r, y) = \frac{1}{2}\sum_{i=1}^{N} ||x_{ri} - y_i||^2 \tag{7}$$

As proposed in VAE/GAN[13], the use of two types of fake samples, passed as input to the discrimination (in our case, the Encoder network E), helps produce better images and learn more expressive latent features[10,31,30]. These two types of samples are the predicted sample x_r from the posterior $q_\phi(z|x)$ and the generated sample from the prior $p(z)$ denoted by x_p. The complete architecture of the proposed model is presented in Fig 2

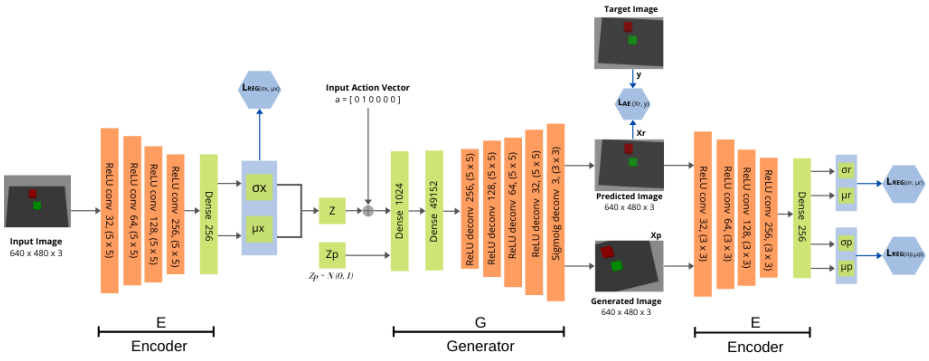
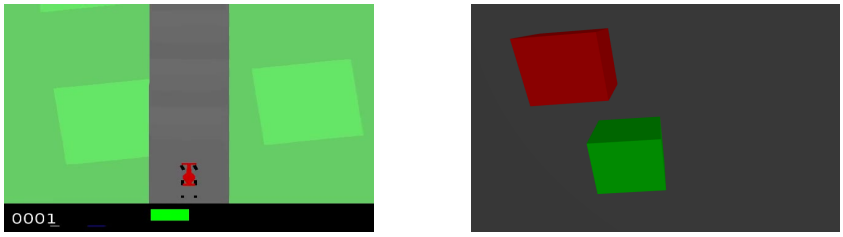


Fig. 2: Complete architecture of the proposed action-conditioned frame prediction model.

To resume, our model acts as a standard VAE for real samples and acts like a GAN when handling generated/predicted samples distinguishing the real sample x and generated samples x_r and x_p. Therefore the total loss functions for the Encoder network E and the Generator network G are redefined as:

$$L_E = L_{REG}(E(x)) + \alpha[m - L_{REG}(E(x_r))]^+ + \alpha[m - L_{REG}(E(x_p))]^+ + L_{AE}(x_r, y) \tag{8}$$

$$L_G = \alpha L_{REG}(E(x_r)) + \alpha L_{REG}(E(x_p)) + L_{AE}(x_r, y) \tag{9}$$



(a) Car-Racing Dataset(64x64) **(b)** Two-Cubes Dataset(640x480)

Fig. 3: Examples of frames from the two used datasets.

α is weighting parameter used to balance the importance of each item. Keep in mind that the encoder network E has two output variables, therefore $E(x) = (\mu_x, \sigma_x)$, $E(x_r) = (\mu_{x_r}, \sigma_{x_r})$, and finally $E(x_p) = (\mu_{x_p}, \sigma_{x_p})$. The equations (8) and (9) form a min-max game between the encoder and the generator when $L_{REG}(E(x_r)), L_{REG}(E(x_p)) \leq m$

3.2 Network Architecture

Our goal is to have a system easy to train without any special configurations that may also be applied to different image sizes. The encoder network consists of four convolutional layers with 32, 64, 128, 256 filters with a kernel size of 5x5, respectively. All of these convolutional layers use a stride of 2 and a *ReLU* as activation function[1]. The output of the convolutional layer is flattened and routed to a fully connected layer, which is then connected via two fully connected layers that each output the vectors μ and σ. After encoding, the resulting hidden representation is flattened into a vector and concatenated with the one-hot encoded action vector. The generator network consists of two fully connected layers, followed by five deconvolutional layers. The first four layers mirror the encoder configuration with 256, 128, 64, and 32 filters with a stride of 2, kernel size of 5x5 and *ReLU*. The last deconvolutional layer employs 3 filters of size 3×3 kernel with a stride of 1 and *Sigmoid* as an activation function. See Fig 2. The pseudocode of training this network is presented in Algorithm 1.

4 Implementations

4.1 Data-sets and Experiments

To examine the performance of our proposal, we applied our network architecture to two datasets. These datasets differ in size and features, see Fig 3. We collected and standardized each of the images that compose these datasets.

[1] We have to mention that other activation functions were also tested, specifically LeakyReLU and Tanh (for the last layer of the generator). However, the results did not improve, and the computational load increased significantly

Algorithm 1 Action-Conditioned Frame Prediction

Require: $\phi_{Enc}, \theta_{Gen}, \leftarrow$ Initialize network parameters
while not converged **do**
  $x, a \leftarrow$ random mini-batch of images and actions from training dataset
  $y \leftarrow$ random mini-batch of target images from training dataset
  $Z \leftarrow Enc(x)$
  $Z_p \leftarrow$ sample from $N(0, I)$
  $a_p \leftarrow$ vector of zeros of size of a
  $x_r \leftarrow Gen(Z, a)$
  $x_p \leftarrow Gen(Z_p, a_p)$
  $L_{AE} \leftarrow L_{AE}(x_r - y)$ ▷ error w.r.t. target image
  $Z_r \leftarrow Enc(ng(x_r))$ ▷ $ng(.)$ back prop. of the gradient is stopped
  $Z_{pp} \leftarrow Enc(ng(x_p))$
  $L^E_{adv} \leftarrow [m - L_{reg}(Z_r)]^+ + [m - L_{reg}(Z_{pp})]^+$
  $\phi_{Enc} \leftarrow \phi_{Enc} - \eta \nabla_{\phi_{Enc}} (L_{REG}(Z) + \alpha L^E_{adv} + L_{AE})$ ▷ Update params. for Enc.
  $Z_r \leftarrow Enc(x_r)$
  $Z_{pp} \leftarrow Enc(x_p)$
  $L^G_{adv} \leftarrow L_{reg}(Z_r) + L_{reg}(Z_{pp})$
  $\theta_{Gen} \leftarrow \theta_{Gen} - \eta \nabla_{\theta_{Gen}} (\alpha L^G_{adv} + L_{AE})$ ▷ Update params. for Gen.
end while

Car-Racing Dataset The frames for this dataset were collected using the Car-Racing environment of OpenAI Gym [1]. The agent acts randomly throughout the environment during multi-role times. Each of the random actions, along with the corresponding resulting observations of the environment, were stored. Each image of this dataset is composed of 64×64 pixels, with 3 channels (RGB image). In total, 30,000 images with their respective random action compose the training dataset. The validation and testing set both consist of 3,000 samples each. An example of a single frame is given in Fig 3a. This database is a good starting point since the image size is small; Additionally, this is a 2D environment where the objects (road, car, grass, road marking) features and colours in the images are clearly defined. We train the model for 5,000 iterations with a random sample batch of size 32, using the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with a learning rate of 0.00004 and a decay rate value of 1e-8. The latent dimension for this dataset is $z = 32$, the parameters $m = 2.0$, and $\alpha = 0.25$. These hyper-parameters were determined empirically through the use of a small portion of the training dataset.

Two-Cubes Dataset To examine the proposed model's performance on larger dimensional and more complex data, we use our second dataset. Each image of this dataset is composed of 640×480 pixels, with 3 channels. Each image includes two cubes with a side length of 8 cm, a red and a green one, are placed on a grey pad. An example of an image is given in Fig 3b. The state-space is relatively simple, with only two cubes; however, it should be considered this is in a 3D environment, which increases the complexity of the prediction task. This

dataset was collected in a simulated environment using Gazebo. A UR5 robot arm placed on a horizontal plane performs six possible moves with a camera mounted on its arm's gripper. After each action (randomly selected), the robot arm takes a picture of the cubes and stored it along with the respective action; The process is repeated until complete one episode. Each episode started in an initial state and consisted of 100 steps with one action per step. In total, 20,000 images, with large variations in poses, features, and angles, compose the training data, while the validation and testing set both consist of 2,000 samples each. Furthermore, the task initially seems simplistic in its design; however, the actions space with the six possible actions that move the arm in a 3D space is vastly more complex than the Car-Racing dataset, which was limited to a 2D environment. For this dataset, the latent dimension is 64, $m = 12$ and $\alpha = 0.25$. We train the model for 10,000 iterations with a random sample batch of size 32, using the Adam optimizer with a learning rate of 0.00002 and a decay rate value of 1e-8. The source code, as well as the images dataset, could be found at https://github.com/dvalenciar/Action-Frame-Prediction

4.2 Experiments and Results

In order to evaluate the predictions' quality of our proposal after the training process, we carry out two experiments, a single-step prediction, and a sequence-prediction. The single-step experiment consists of the prediction of one target image using one input image and one input action. On the other hand, the sequence-prediction experiment consists of predicting n sequence target images using one input image with n input actions, i.e., the model will use its own predictions as inputs during the following steps since only one input image from the dataset is available; this experiment is a complex challenge for the neural network since the predictions' quality dependent on the quality of all previous predictions. Additionally, we compare our model results against a well known convolutional VAE adapted for action-conditioned frame prediction. The results from experiment one show our system can generate and predict realistic frames. The images' quality matches the expectations in both datasets, which proves that our design allows an easy scale up the resolution of input images; in other words, the proposed architecture can be applied to small and large images. The predicted samples have enough details, the blurriness is almost imperceptible, and the predictions locate the image's objects in the right positions, see some samples results in Fig 4. Additionally, a quantitative analysis using MSE (Mean Square Error), Peak-to-Noise Ratio (PSNR) and SSIM (Structural Similarity Index) is presented in Table 1, where the results obtained with the two data sets show our proposal is significantly superior to VAE.

Regarding the second experiment; since the model reintroduces its own predictions as inputs, high-quality in the predictions is crucial. We carry out this experiment for 100 consecutive predictions; we found that get a good performance out of the model after 30 consecutive predictions is challenging, since the model starts to generate samples where the objects (cubes or car) are located in the wrong location. However, the first 30 consecutive predictions almost exactly

Table 1: Numerical comparison between our model and VAE for the two implemented datasets. The prediction accuracy is quantified by computing the average MSE, SSIM and PSNR among all the images that compose the testing dataset.

	Car-Racing Dataset							
	Our Model				**VAE**			
	Avg	Std	Max	Min	Avg	Std	Max	Min
MSE	0.000662	0.00065	0.00745	0.00044	0.001098	0.00987	0.00197	0.00066
SSIM	0.972032	0.00976	0.98334	0.67765	0.900213	0.01323	0.99343	0.41654
PSNR	34.53924	0.91034	38.0232	32.3838	29.31243	1.89040	30.0012	26.5656

	Two-Cubes Dataset							
	Our Model				**VAE**			
	Avg	Std	Max	Min	Avg	Std	Max	Min
MSE	0.000120	0.00046	0.00117	0.00002	0.003402	0.01246	0.00763	0.00011
SSIM	0.980012	0.00132	0.98995	0.00987	0.916295	0.05321	0.93222	0.68030
PSNR	35.12535	0.45464	37.3736	31.0456	25.74028	3.00333	26.0056	21.5626

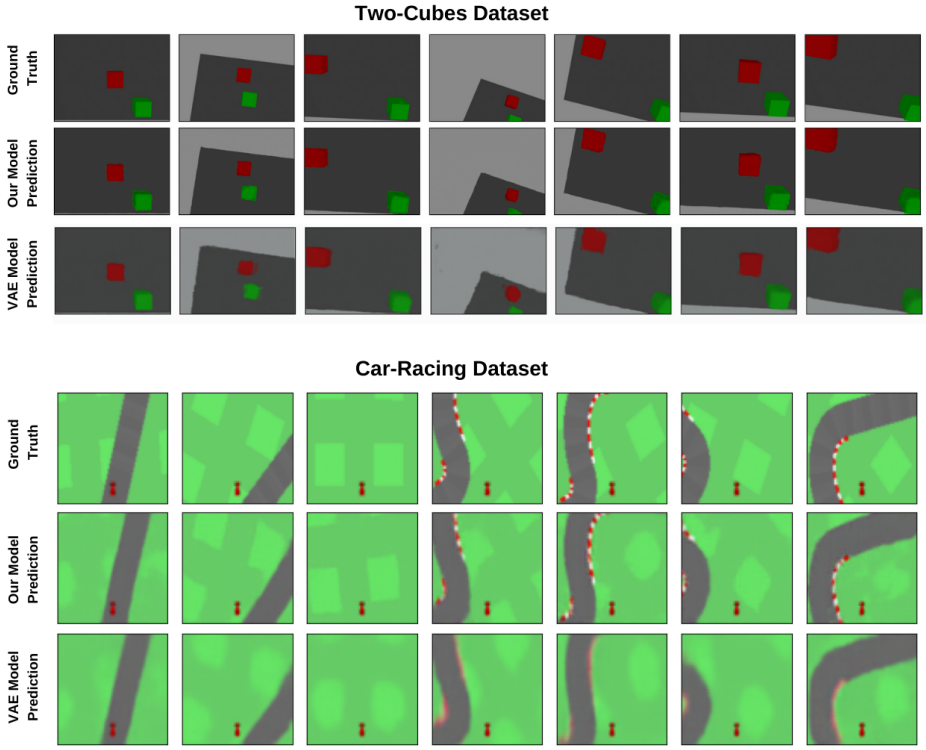


Fig. 4: Results from single-step prediction experiment. Rows one and four show the target images of the Two-cube and the Car-Racing datasets, respectively. The predicted images obtained with our model are presented in rows two and five for each data set. Rows three and six show the prediction using VAE.

match the target images. From step 21 to step 30, the predicted samples show blurriness on the objects' edges; however, objects are placed in the correct position. Considering the outputs of the system are re-inserted as inputs, the results meet the expectations. A sample result with the first 15 steps of the sequence of the two data sets is presented in Fig 5. From prediction 31, the frames get worse with each step until they reach the point where the image's objects begin to lose their geometric shape or are placed in the incorrect positions. The MSE and SSIM achieved throughout the sequence prediction for the Two-Cubes dataset can be seen in Fig 6. It shows that prediction errors accumulate gradually, staying below a value of 0.04, while the SSIM value in each prediction decreases slowly, reaching a minimum value of 0.71 during all 100 steps. Similar results are obtained with the Car-Racing dataset and can be analyzed in Fig 7. The MSE rise steadily, reaching a peak of 0.023, while the SSIM value in each prediction decreases progressively, reaching a minimum value of 0.876 through all 100 predictions.

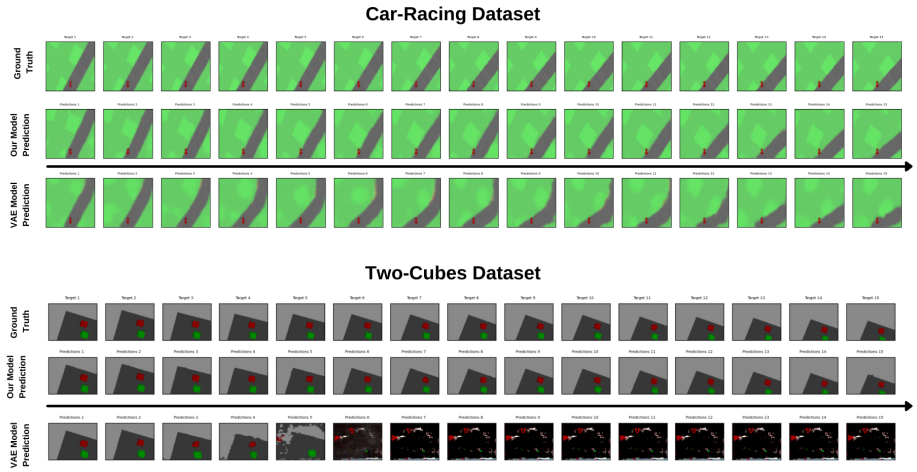
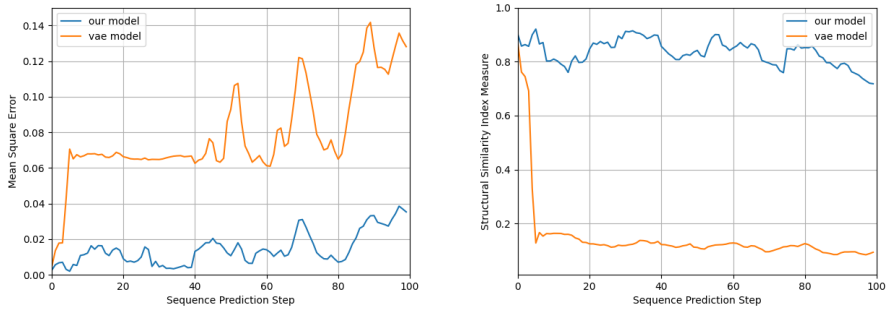


Fig. 5: Results from sequence-predictions experiment. Each prediction is re-inserted as input for this experiment. The predicted images achieved with our model are shown in rows two and five for each dataset.

5 Conclusions

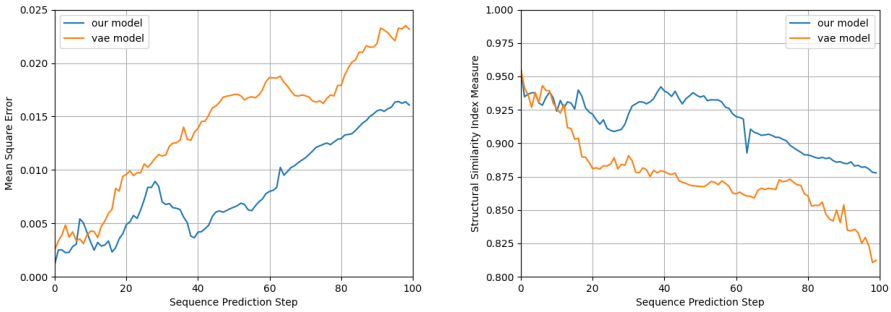
In this paper, we propose an action-conditioned model for frame prediction. A model consisting of two parts is trained introspectively to predict sharp, clear, and diverse images without using an extra discriminator. The proposed architecture overcomes the limitation of VAE and GAN, especially in the stability and training process; This model can be trained easier than state-of-the-art frame

prediction networks while producing equivalent results. Moreover, two action-conditioned datasets were created to test the performance of the system with different sizes of images.



(a) MSE of the sequence prediction **(b)** SSIM of the sequence prediction

Fig. 6: Accuracy of sequence prediction using MSE and SSIM of Two Cubes Dataset



(a) MSE of the sequence prediction **(b)** SSIM of the sequence prediction

Fig. 7: Accuracy of sequence prediction using MSE and SSIM of Car-Racing DataSet

We considerer is essential to mention that directly comparing the performance of similar proposals such as soft-introVAE[4] or SRVAE[9] against our model would be unbalanced because those methods do not include external actions, and they were designed as image generators but not as image predictors; therefore, we have to modify their loss functions and part of their original archi-

tectures. That is why we compared our proposal against VAE, which is the most common and accepted algorithm for image prediction that has been previously tested with external actions.

Nevertheless, our proposed architecture still leaves much to be accomplished; even when the results were as expected, we believe that it is necessary to look for additional machine learning techniques that speed up the training process since, at the moment, our system needs to be trained for long periods to achieve acceptable results. Our future work will attempt to link this work with reinforcement learning(RL), specifically model-based RL where an accurate prediction of future events could help to learn better models of the system. Finally, we believe the model proposed in this article could bring prominent benefits in real-robot practical applications such as object detection in mobile robots, self-driven cars, or self-generation trajectories for robot arms.

References

1. AI, O.: Gym toolkit, https://gym.openai.com/envs/CarRacing-v0.html
2. Berthelot, D., Schumm, T., Metz, L.: Began: Boundary equilibrium generative adversarial networks (2017)
3. Castelló, J.S.: A comprehensive survey on deep future frame video prediction (2018)
4. Daniel, T., Tamar, A.: Soft-introvae: Analyzing and improving the introspective variational autoencoder (2021)
5. Ebert, F., Finn, C., Dasari, S., Xie, A., Lee, A., Levine, S.: Visual foresight: Model-based deep reinforcement learning for vision-based robotic control (2018)
6. Finn, C., Goodfellow, I., Levine, S.: Unsupervised learning for physical interaction through video prediction (2016)
7. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks (2014)
8. Ha, D., Schmidhuber, J.: World models. arXiv preprint arXiv:1803.10122 (2018)
9. Heydari, A.A., Mehmood, A.: Srvae: super resolution using variational autoencoders. In: Pattern Recognition and Tracking XXXI. vol. 11400, p. 114000U. International Society for Optics and Photonics (2020)
10. Huang, H., Li, Z., He, R., Sun, Z., Tan, T.: Introvae: Introspective variational autoencoders for photographic image synthesis (2018)
11. Khan, S.H., Hayat, M., Barnes, N.: Adversarial training of variational auto-encoders for high fidelity image generation (2018)
12. Kingma, D.P., Welling, M.: Auto-encoding variational bayes (2014)
13. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric (2016)
14. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network (2017)
15. Lee, A.X., Zhang, R., Ebert, F., Abbeel, P., Finn, C., Levine, S.: Stochastic adversarial video prediction (2018)
16. M., J.J.: Kullback-Leibler Divergence, pp. 720–722. Springer Berlin Heidelberg, Berlin, Heidelberg (2011), https://doi.org/10.1007/978-3-642-04898-2_327
17. Malik, A., Troute, M., Capoor, B.: Deepgifs: Using deep learning to understand and synthesize motion

18. Oh, J., Guo, X., Lee, H., Lewis, R., Singh, S.: Action-conditional video prediction using deep networks in atari games (2015)
19. Oprea, S., Martinez-Gonzalez, P., Garcia-Garcia, A., Castro-Vargas, J.A., Orts-Escolano, S., Garcia-Rodriguez, J., Argyros, A.: A review on deep learning techniques for video prediction (2020)
20. Paxton, C., Barnoy, Y., Katyal, K., Arora, R., Hager, G.D.: Visual robot task planning (2018)
21. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks (2016)
22. Rasouli, A.: Deep learning for vision-based prediction: A survey (2020)
23. Rhinehart, N., McAllister, R., Kitani, K., Levine, S.: Precog: Prediction conditioned on goals in visual multi-agent settings (2019)
24. Sainburg, T., Thielk, M., Theilman, B., Migliori, B., Gentner, T.: Generative adversarial interpolative autoencoding: adversarial training on latent space interpolations encourage convex latent distributions (2019)
25. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., Chen, X.: Improved techniques for training gans. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 29, pp. 2234–2242. Curran Associates, Inc. (2016), https://proceedings.neurips.cc/paper/2016/file/8a3363abe792db2d8761d6403605aeb7-Paper.pdf
26. Vu, H.S., Ueta, D., Hashimoto, K., Maeno, K., Pranata, S., Shen, S.M.: Anomaly detection with adversarial dual autoencoders (2019)
27. Walker, J., Marino, K., Gupta, A., Hebert, M.: The pose knows: Video forecasting by generating pose futures (2017)
28. Wang, E., Kosson, A., Mu, T.: Deep action conditional neural network for frame prediction in atari games. Tech. rep., Technical Report, Stanford University (2017)
29. Zhao, J., Mathieu, M., LeCun, Y.: Energy-based generative adversarial network (2017)
30. Zhao, S., Song, J., Ermon, S.: Infovae: Information maximizing variational autoencoders (2018)
31. Zheng, K., Cheng, Y., Kang, X., Yao, H., Tian, T.: Conditional introspective variational autoencoder for image synthesis. IEEE Access **8**, 153905–153913 (2020), https://doi.org/10.1109/ACCESS.2020.3018228
32. Zhu, D., Chen, H., Yao, H., Nosrati, M., Yadmellat, P., Zhang, Y.: Practical issues of action-conditioned next image prediction (2018)
33. Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation (2018)