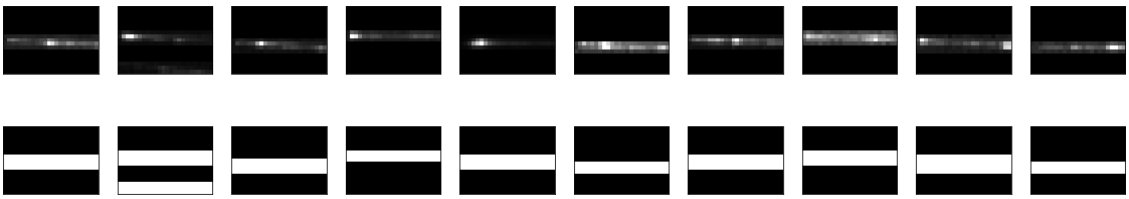
APPENDIX III: LESS SUCCESSFUL GENERATIVE MODELS

Autoencoders



A Variational Autoencoder with Convolutional Architecture:

- $n_{latent} = 100$
- Batch size $= 64$
- Optimizer: Adam, Learning rate $\eta = 10^{-4}$
- Epochs $= 1\ 000\ 000$
- Sample size at each epoch $= 64$

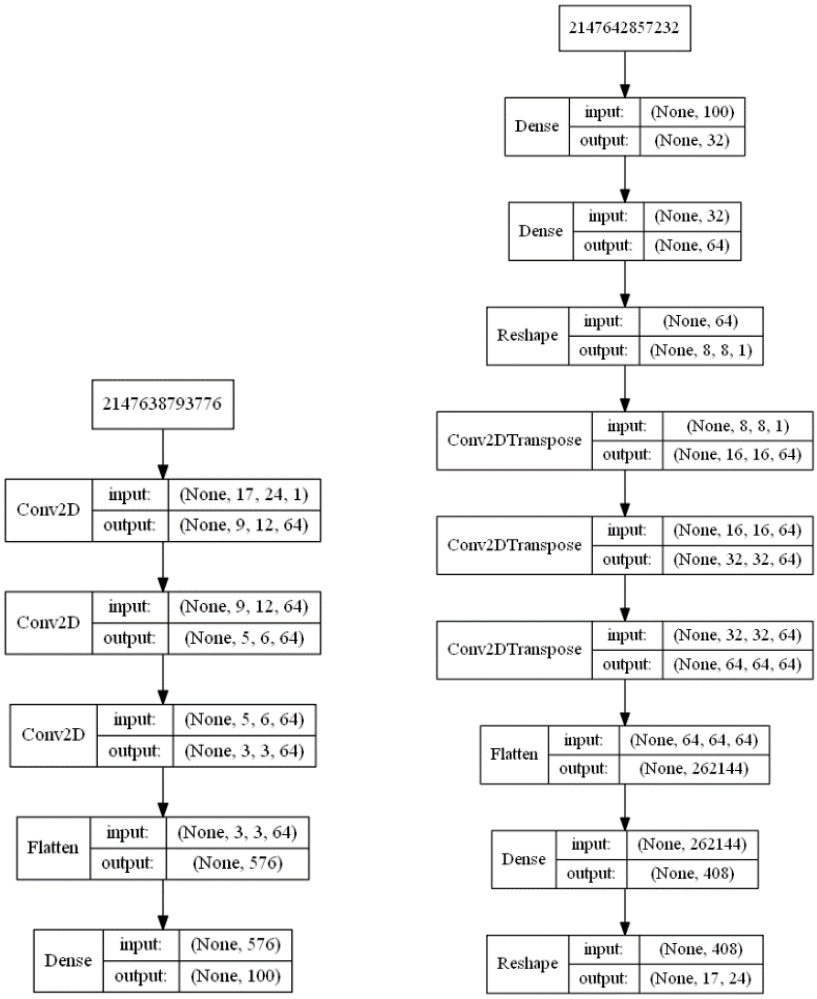


Figure 1: VAE Encoder (left) and Decoder (right)

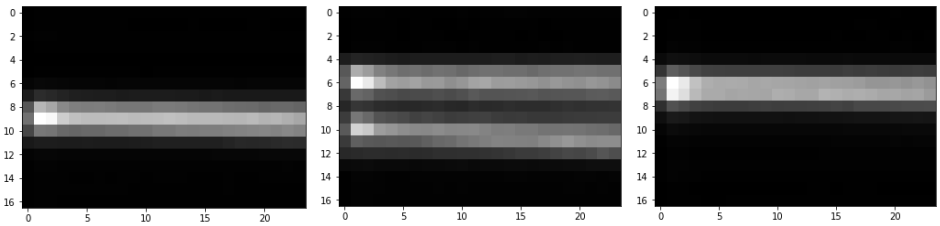
Here, the encoder returns the μ for each of the 100 latent dimensions, $\sum$, which is calculated as $\mu \times 0.5$; and z, which is the result of multiplying a random normal vector ε with e^{Σ} and adding μ, i.e.

$$z = \left(\varepsilon \times e^{\Sigma}\right) + \mu$$

Equation 1

The input to the decoder is a sampled z vector as defined above.

Below are six examples of simulated tracklet image data, produced by the VAE as explained above.



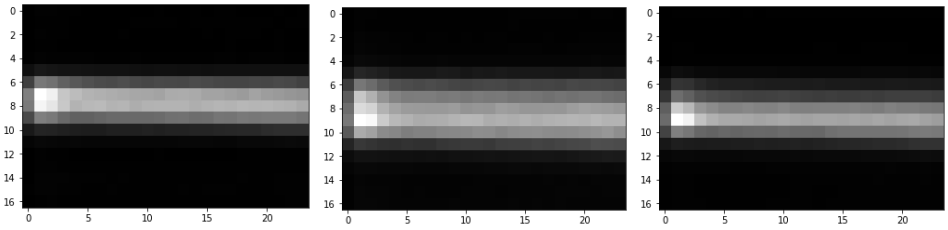


Figure 2: Six examples of simulated data created using a Variational Autoencoder

Boundary-Seeking Generative Adversarial Network

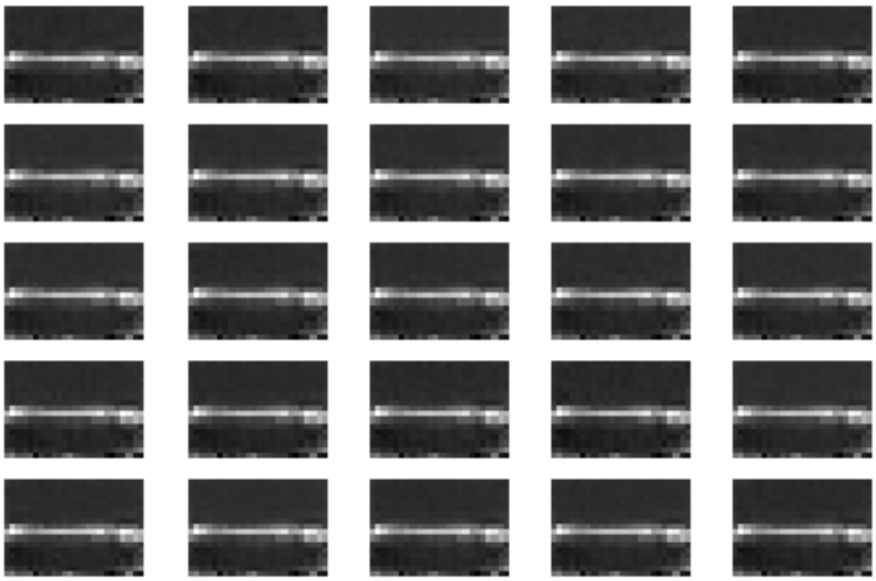
100 Latent dimensions

Adam optimizer with learning rate = 0.000001 and a batch size of 32

Generator with 8 hidden layers with 128, 256, 256, 256, 512, 512, 512 and 1024 nodes, using leaky ReLU activation and an output layer using tanh activation

Convolutional discriminator using two convolutional layers, max-pooling and 5 hidden layers with 1024, 512, 256, 128 and 64 nodes and a single node output layer with sigmoid activation.

Example output after 29000 epochs:



1.1.1.1 Adversarial Autoencoder

1.1.1.1.1 Version 1

10 Latent dimensions

Adam optimizer with learning rate = 0.002 and beta1 = 0.5

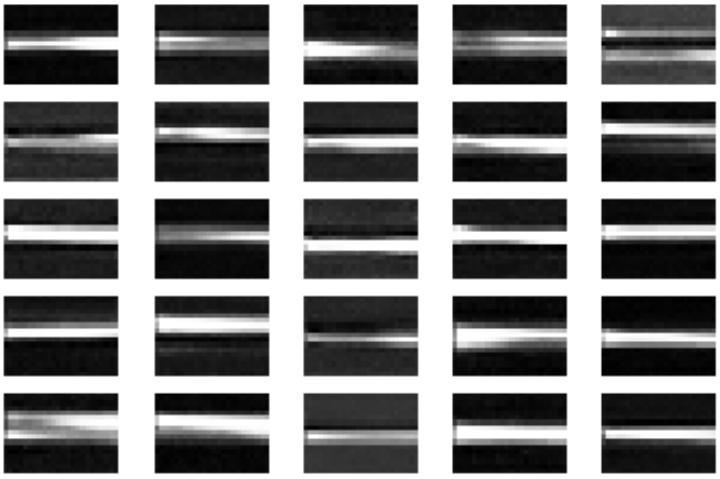
Batch size = 32

Encoder with 2 hidden layers with 512 nodes each, using leaky ReLU activation

Decoder with 2 hidden layers with 512 nodes each, using leaky ReLU activation and an output layer with tanh activation

Discriminator with two hidden layers, with 512 and 256 nodes respectively and sigmoid activation in the single-node output layer

Sample result after 19800 epochs:



1.1.1.1.2 Version 2

100 Latent dimensions

Adam optimizer with learning rate = 0.00002 and beta1 = 0.5

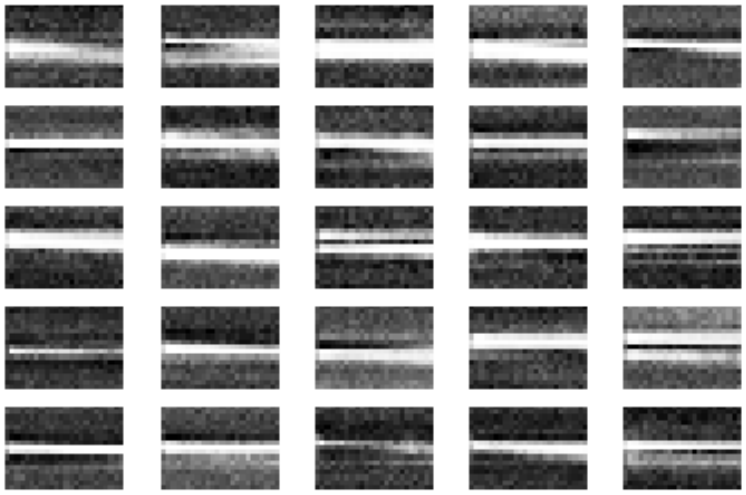
Batch size = 32

Encoder with 3 hidden layers with 512 nodes each, using leaky ReLU activation

Decoder with 3 hidden layers with 512 nodes each, using leaky ReLU activation and an output layer with tanh activation

Discriminator with two hidden layers, with 512 and 256 nodes respectively and sigmoid activation in the single-node output layer

Sample result after 30800 epochs:



1.1.1.1.3 Version 3

8 Latent dimensions

Adam optimizer with learning rate = 0.000002 and beta1 = 0.5

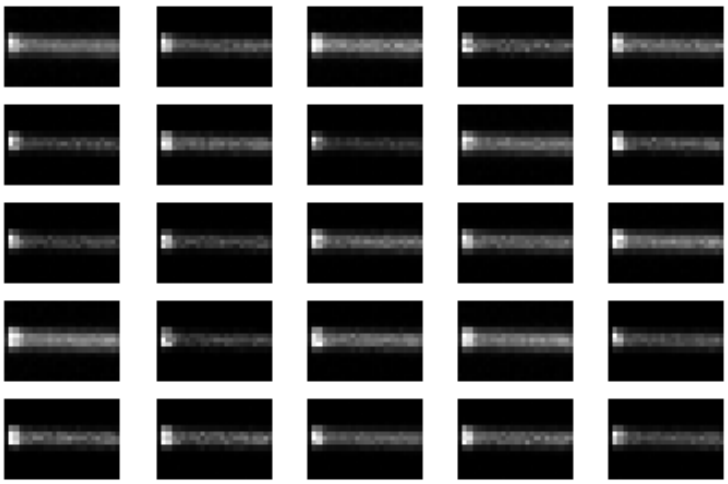
Batch size = 32

Encoder with 7 hidden layers with 1024, 512, 256, 128, 64, 32 and 16 nodes respectively, using leaky ReLU activation

Decoder with 4 hidden layers with 128, 256, 512 and 1024 nodes respectively, using leaky ReLU activation and an output layer with tanh activation

Discriminator with 4 hidden layers, with 1024, 512, 256 and 128 nodes respectively and sigmoid activation in the single-node output layer

Sample result after 23600 epochs:



1.1.1.1.4 Version 4

12 Latent dimensions

Adam optimizer with learning rate = 0.000002 and beta1 = 0.5

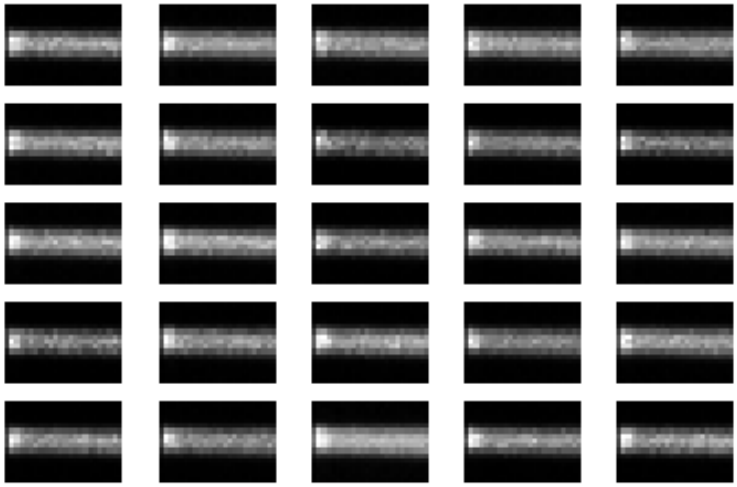
Batch size = 32

Encoder with 5 hidden layers with 1024, 512, 512, 128 and 128 nodes respectively, using leaky ReLU activation

Decoder with 4 hidden layers with 256, 256, 512 and 1024 nodes respectively, using leaky ReLU activation and an output layer with tanh activation

Discriminator with 8 hidden layers, with 512 nodes each and sigmoid activation in the single-node output layer

Sample result after 73200 epochs:



1.1.1.2 Bidirectional Generative Adversarial Network

100 Latent dimensions

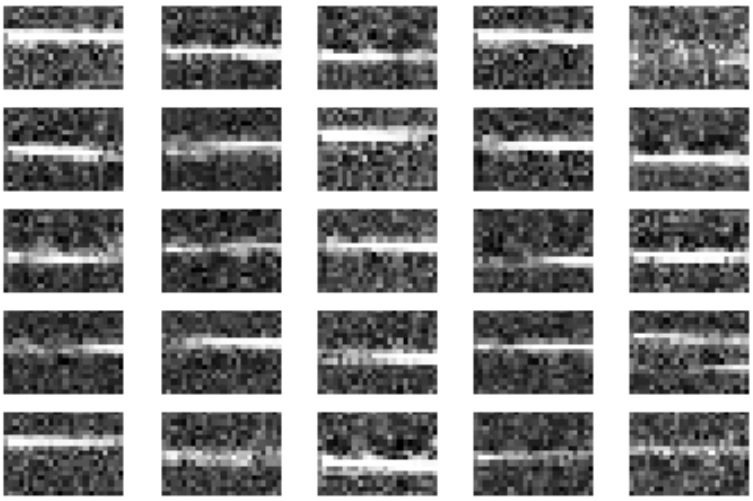
Encoder with 2 hidden layers with 512 units each, using Leaky ReLU activation and employing Batch Normalization with momentum = 0.8 after each

Generator with the same architecture as the encoder, with an additional dense layer of the same dimensions as the number of pixels in an image, with tanh activation

Discriminator with 3 hidden layers with 1024 nodes each using leaky relu activation and a single node output layer with sigmoid activation function

Trained using Adam Optimizer with Learning Rate = 0.0002 and Beta1=0.9 and a batch size of 32

Example output after 39600 epochs:



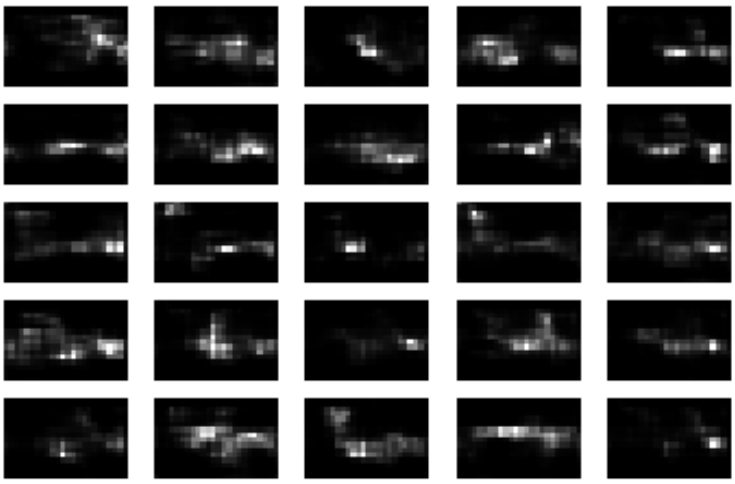
Deep Convolutional Generative Adversarial Network

100 Latent dimensions

Generator with Dense layer with 3072 nodes, reshaped to $4 \times 6 \times 128$ *with 2D Upsampling, followed by 3 2D convolutional layers, with the first two followed by Batch normalization and 2D Upsampling, using ReLU activation in the first 4 layers and tanh in the final output layer*

Discriminator with 4 2D convolutional layers, with Batch Normalization applied after the second, third and fourth layer and zero-padding after the second layer, using leaky ReLU activation in the hidden layer and sigmoid in the output node.

Example output after 57500 epochs:



Generative Adversarial Network

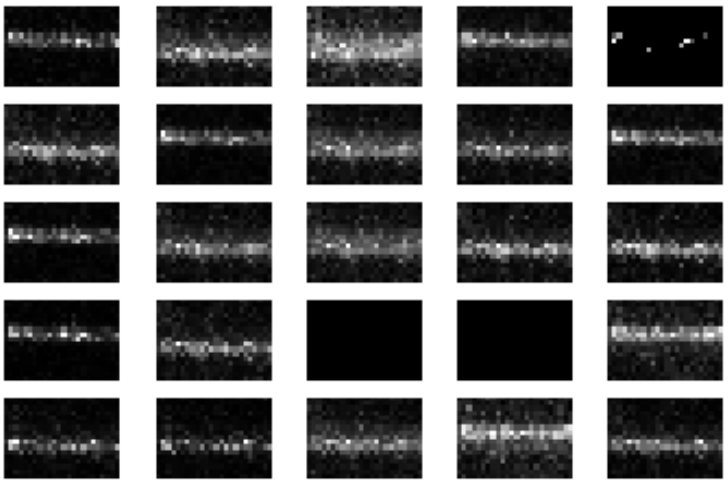
100 Latent dimensions

Two separate Adam Optimizers used for Generator and Discriminator, i.e. with learning rate = 0.00002 and beta1=0.5 for Discriminator and learning rate = 0.00001 and beta1=0.5 for Generator

Generator with 12 hidden layer with 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100,1200 hidden units, using leaky ReLU activation and a dense output layer of the desired output image dimensions with tanh activation, including Batch Normalization with momentum = 0.8 after each hidden layer

Discriminator with 7 hidden layers with 1024, 1024, 512, 512, 256, 256 and 128 units, respectively, using leaky ReLU activation and a single node output layer using sigmoid activation.

Example Output after 66400 epochs:



Least Squares Generative Adversarial Network

100 Latent dimensions

Adam Optimizer with learning rate = 0.00002 and beta1 = 0.5 using batch size = 32

Generator with 6 hidden dense layers with 256, 256, 512, 512, 1024 and 1024 hidden layers using leaky ReLU activation and an output layer with tanh activation, using Batch Normalization with momentum = 0.8 after each hidden layer

Discriminator with 8 hidden layers with 1024, 1024, 512, 512, 256, 256, 128 and 128 hidden layer and a single node output layer with no activation function

Example output after 64600 epochs: