HW1

Kin Chang (013783848)

<2023-03-02 Thu>

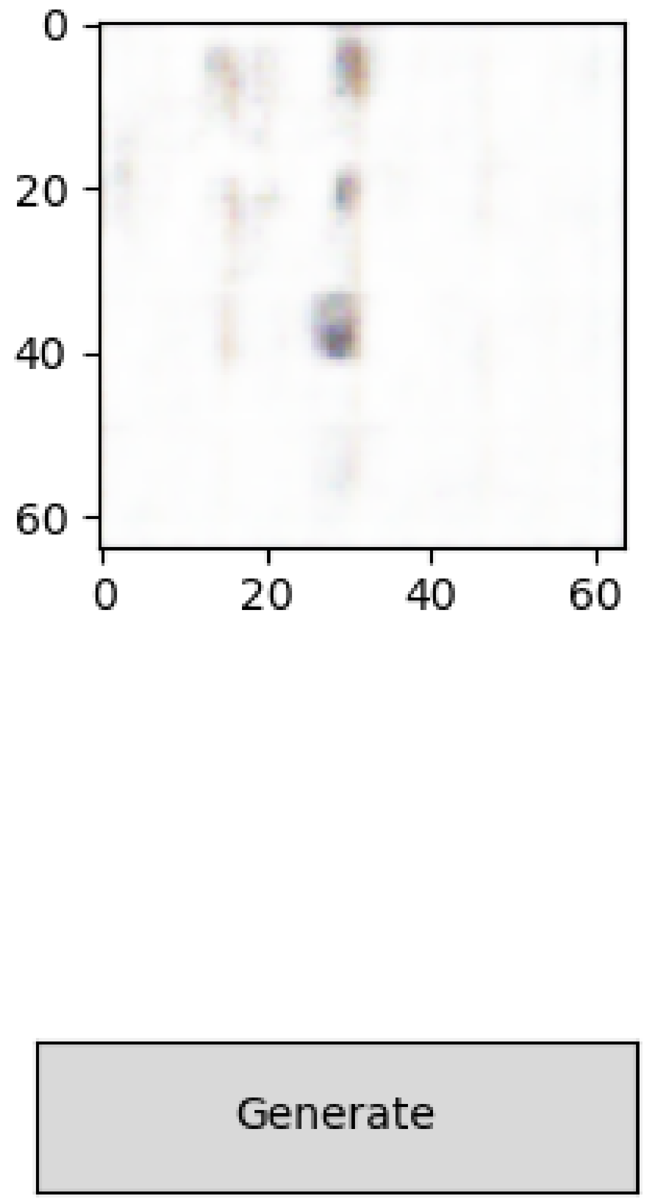
HW1

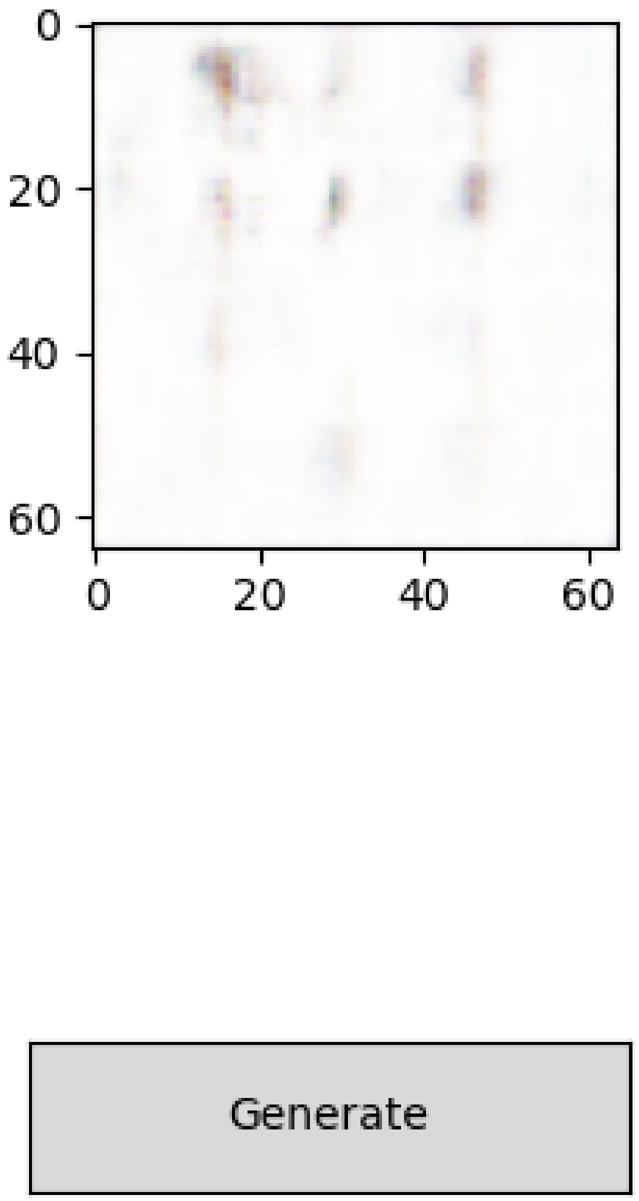
Q1

- Model forward pass: NN of hidden dim [32, 64, 128, 256, 512] -> representation layer of 10 latent features

- In this AE, sampling between encoder and decoder is done by using the `self.z_simple=nn.Linear(...)`, i.e. a straight forward linear transformation.

- Data sampling is done from taking a random actions in gym env, and collect images after each action


0
20
40
60
0
20
40
60
Generate

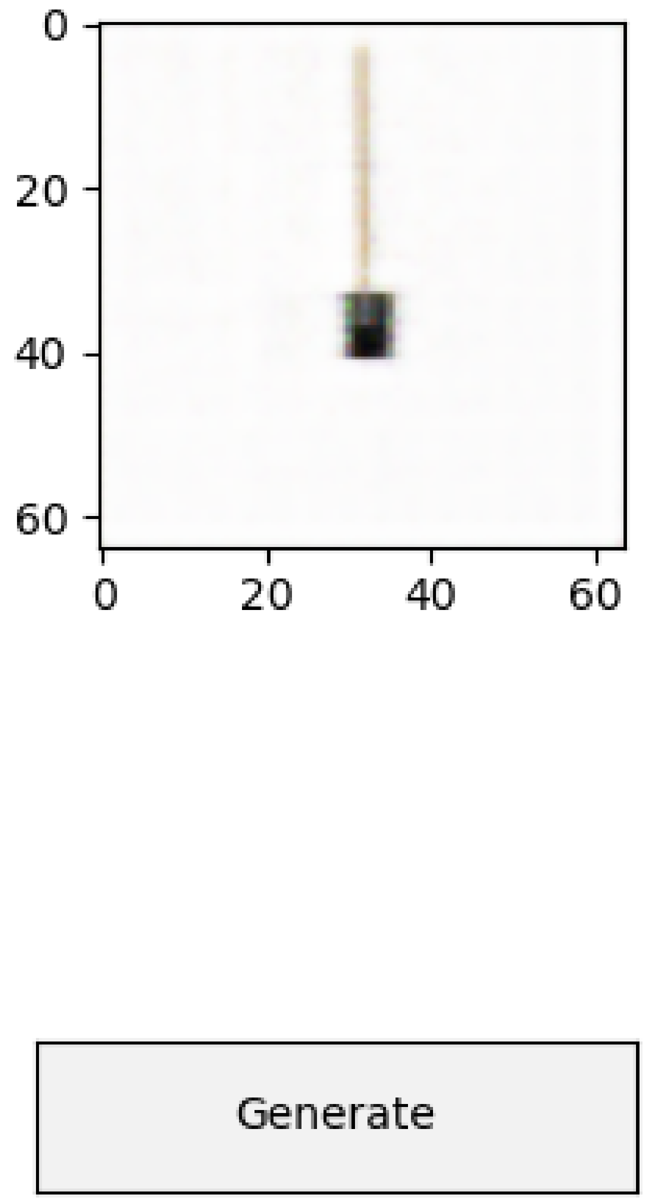

0
20
40
60
0
20
40
60
Generate

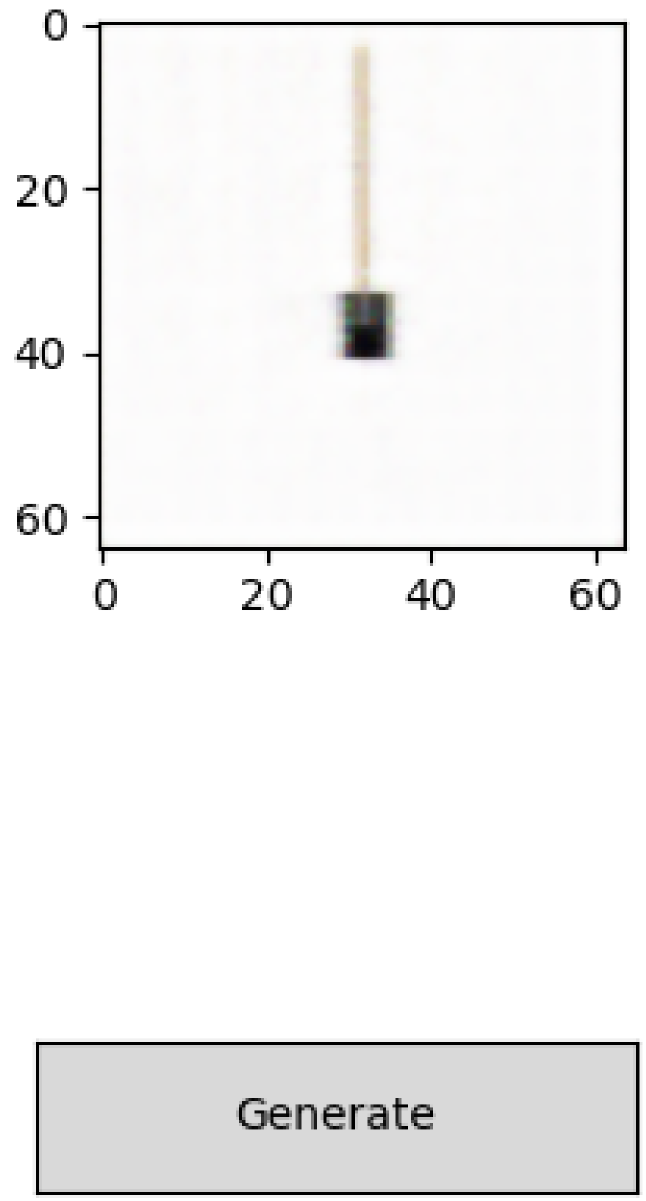
Q2

- The image generated by ae is hit or miss. It could generate an image with cart and pole, but most of the time it generates image that barely resembles anything. On the other hand, vae seems to capture the essence of cart and pole, and generate image that clearly resembles cart and balanced pole consistently.

- They are similar in the sense that they have similar structure, i.e. encoder and decoder. VAE differs in that the encoding layer outputs the mean and standard deviation of each latent variable, instead of the latent vector. Sampling of the latent vector is then done using the mean and std, and is subsequently feed into the decoder.

- The reparametrization trick, i.e. sampling from the mean and std of each latent variable, allow us to backpropagate and estimate the gradient of the Evidence Lower Bound (EVLO). Hence, making stochastic gradient descent possible to minimize loss in VAE.

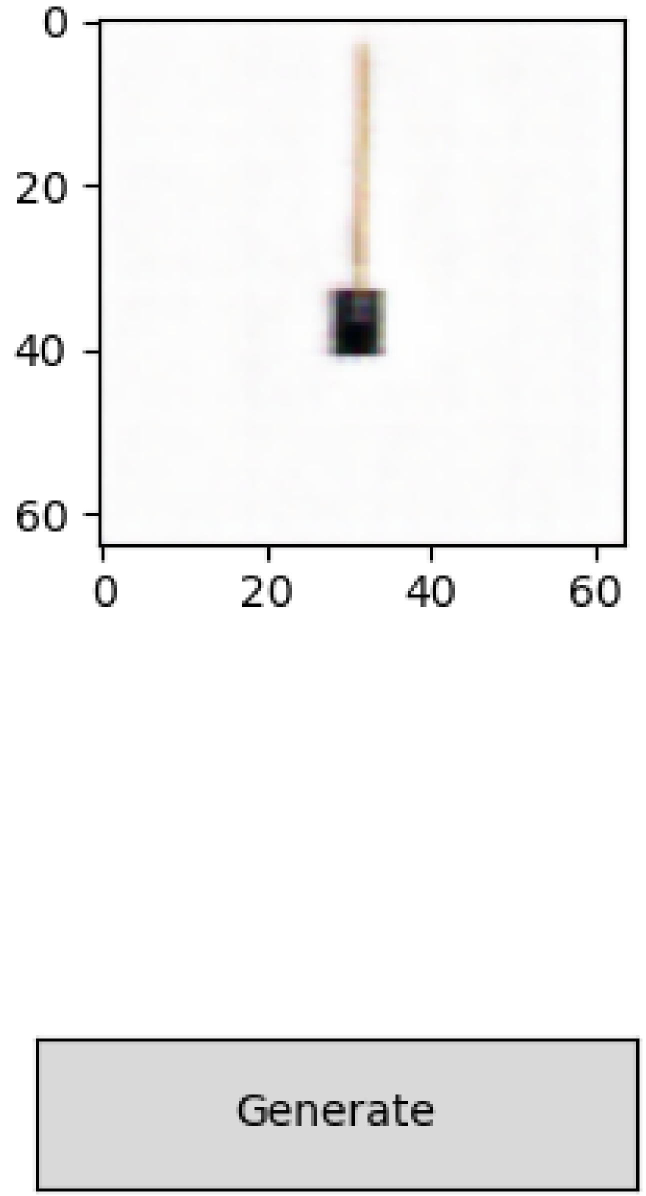

0
20
40
60
0
20
40
60
Generate

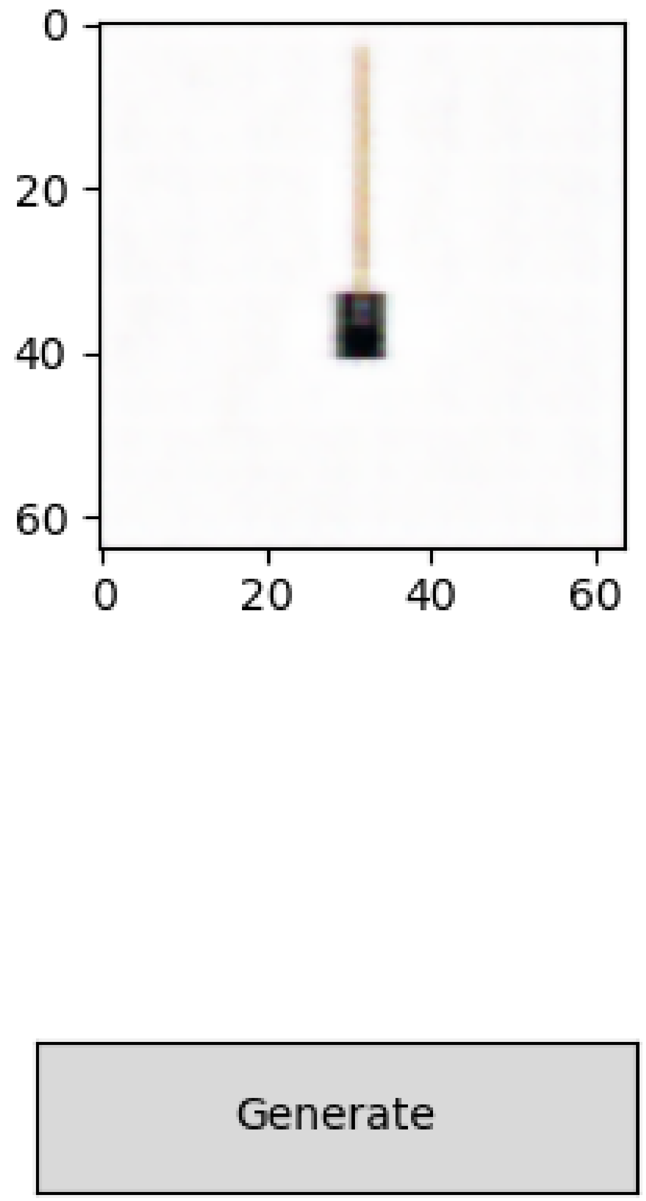

0
20
40
60
0
20
40
60
Generate

Q3

- The environment returns done = True if the following happens:
  - Pole Angle is greater than ±12°
  - Cart Position is greater than ±2.4 (center of the cart reaches the edge of the display)
  - Truncation: Episode length is greater than 500 (200 for v0)
- Code:

```
if done or frame_idx >= 20:
    obs = env.reset()
    frame_idx = 0
```


0
20
40
60
0
20
40
60
Generate

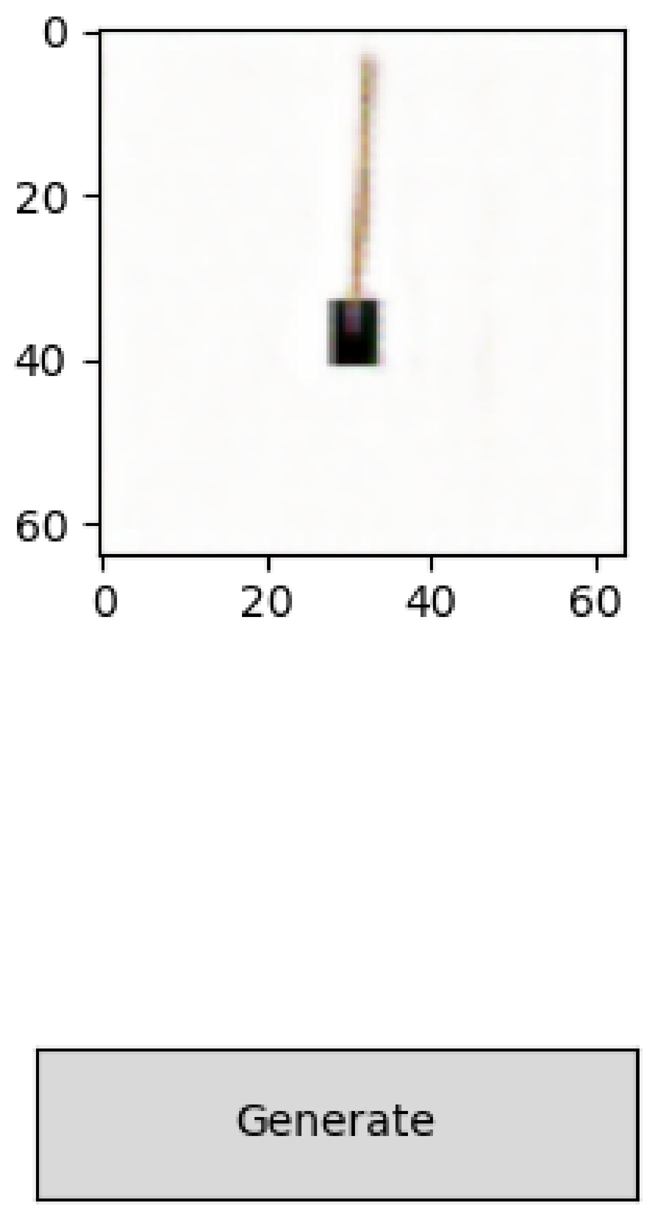

0
20
40
60
0
20
40
60
Generate

Q4

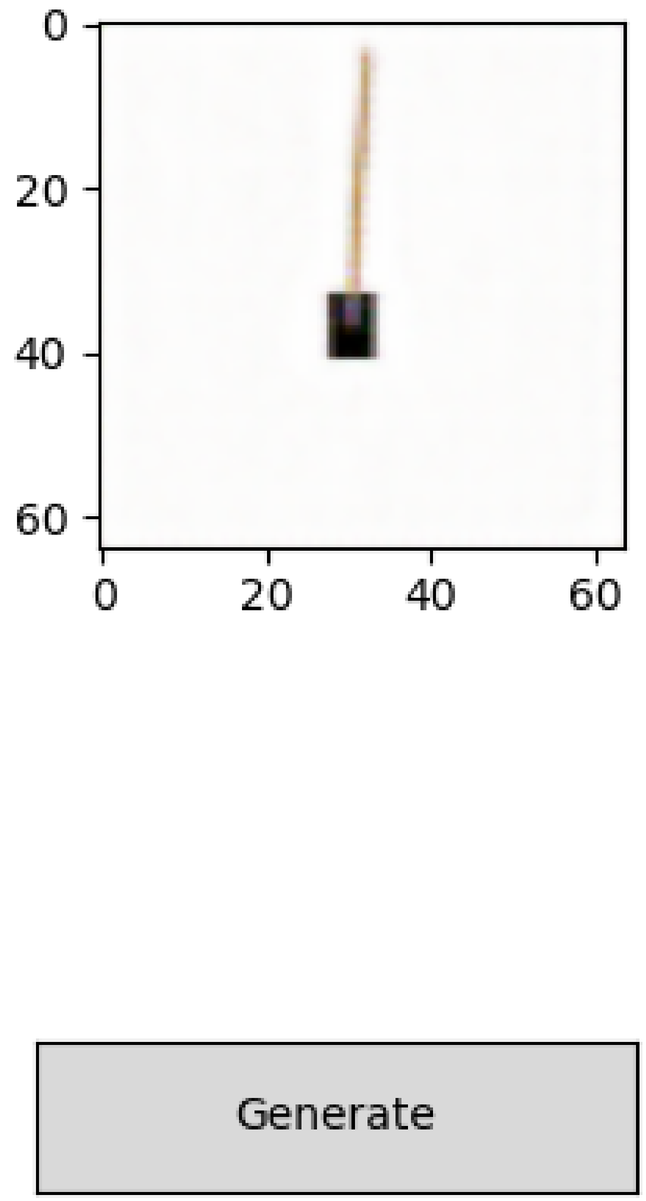
- Range chosen: 0.1-0.2, so that all the images will have the pole on the right side

```
# The pole angle can be observed between (-.418, .418) radians (or ±24°),
# but the episode terminates if the pole angle is not in the range (-.2095, .2095) (or ±12°)
minAngle, maxAngle = 0.1, 0.2
if minAngle <= obs[2] <= maxAngle:
    imgs[i] = img
    i+=1 # also changed the for loop into a while loop
```

- The model does generate images that have the pole on the right side, as expected!



0
20
40
60
0
20
40
60
Generate

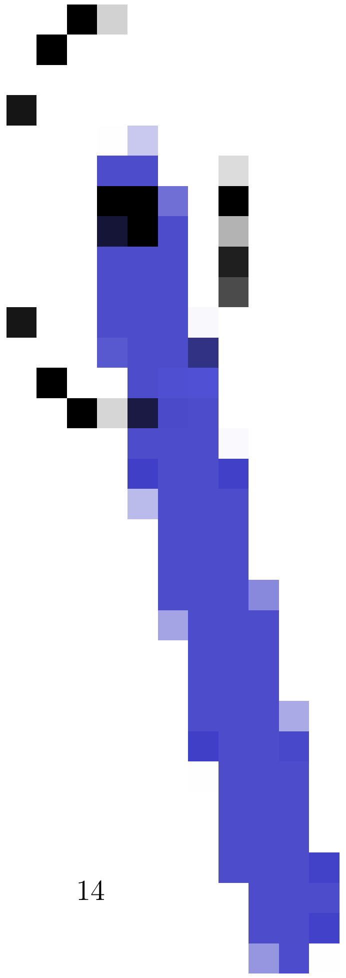

0
20
40
60
0
20
40
60
Generate

Q5

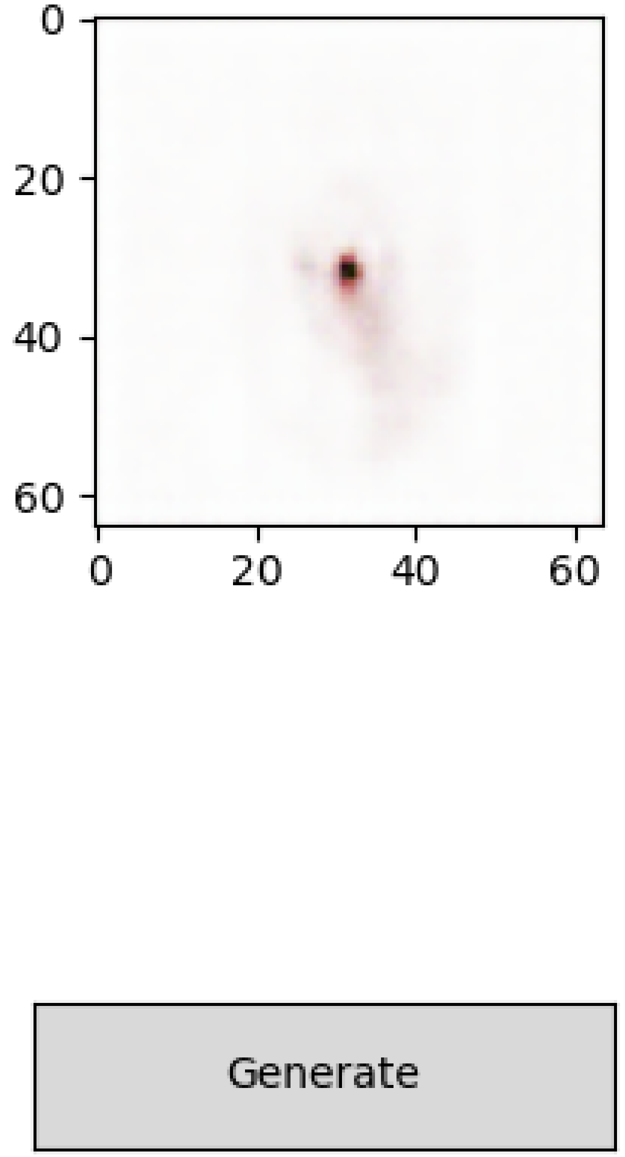
- Chose pendulum environment, and adjusted crop proportion to fit the entire pendulum in the image

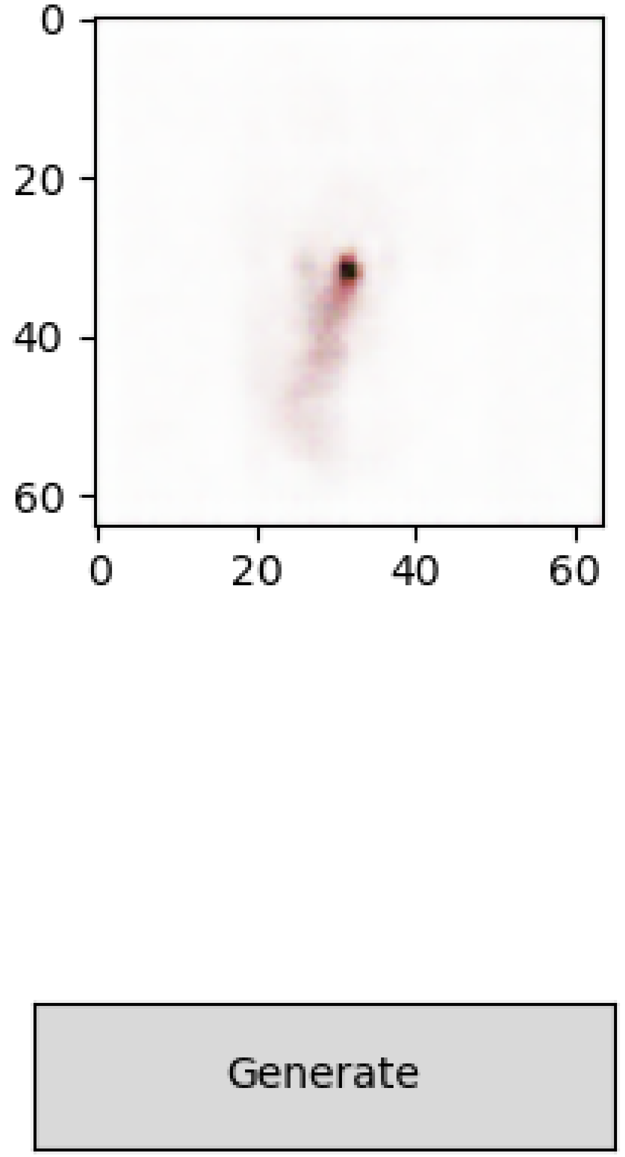
```
crop_proportions = (0.2, 0.0, 0.8, 1.0) # cut top and bottom
...
env = gym.make("Pendulum-v1", render_mode='rgb_array')
```

- A sample training image:



- Everything else in the VAE stays the same, the result somewhat ensembles a pendulum, but isn't crisp or clear at all. The loss is around 0.0070 when the training finished; hence, I believe that futher tuning of the model will yield a better result.


0
20
40
60
0
20
40
60
Generate


0
20
40
60
0
20
40
60
Generate