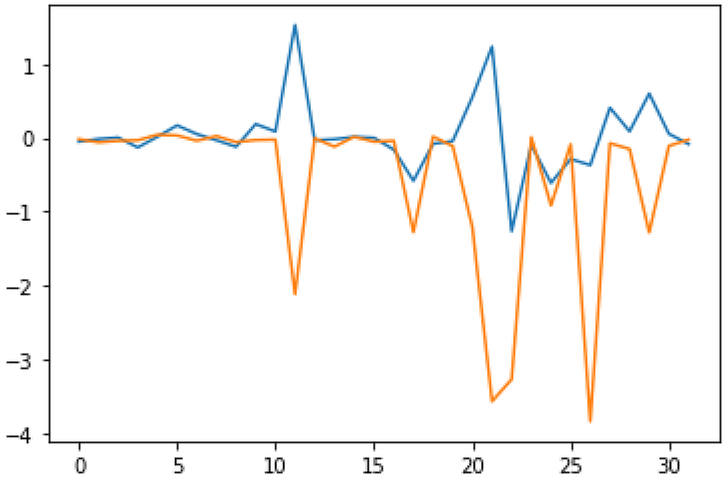
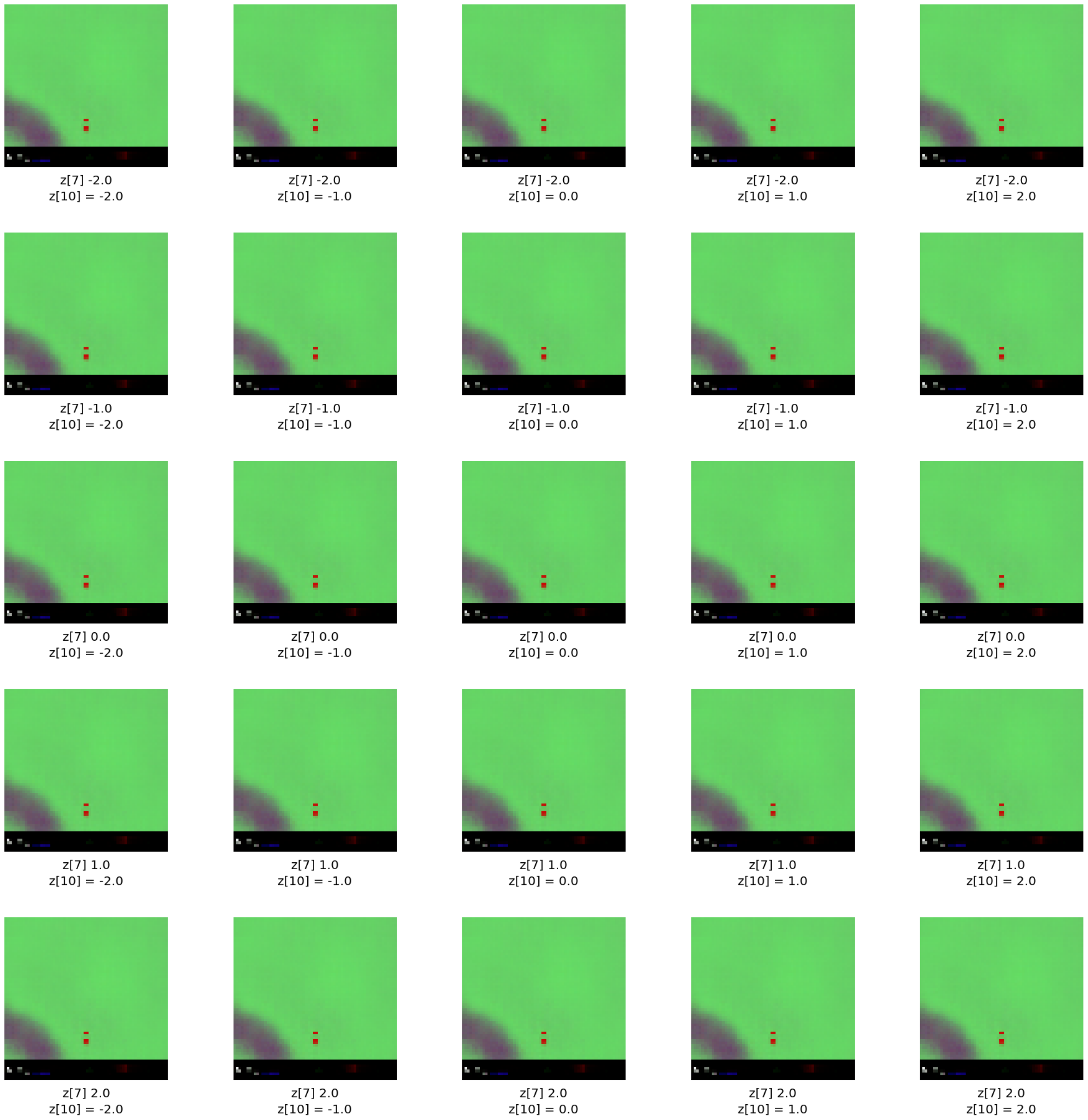
Task2

During this project, we have learnt some precise experience. The first strait for us are the limitation of memory size(i.e. we can’t put too much data into memory), because when fit the model, it will continuously store more information into memory then finally too much data will break through the memory. To overcome this dilemma, we tried to adjust the batch size, decrease the number of epochs and even generate less data for training but still maintain a relatively good accuracy. We spent a lot of time finding a balance between performance and cost. We also tried to parallelly process rnn training data generation and rnn training after the vae trained well.

For each phase we finished, we will run the check script to validate our result. As an example, below is the mean and log variance we get from the trained vae.



And some observations processed through encoding and decoding of vae as below. There are obviously some fuzzes around the road and that means some information is cut by the vae.



After we trained the controller, we want to see how exactly our controller is driving the car.We find a method to view the car driving process visually by using `xvfb-run -a -s "-screen 0 1400x900x24" jupyter notebook` to launch a jupyter server and visit it from our local machine by using http and a unique token produced, then we can see outputs on the screen of our local machine. The output gif is in the “result” folder.

The “result” folder also contains the results from the original world model and the results from our model with vae-gan.