CS-GY-6613 – Artificial Intelligence – Project 2
World Models - Can agents learn inside of their own dreams?

Task 1

INTRODUCTION

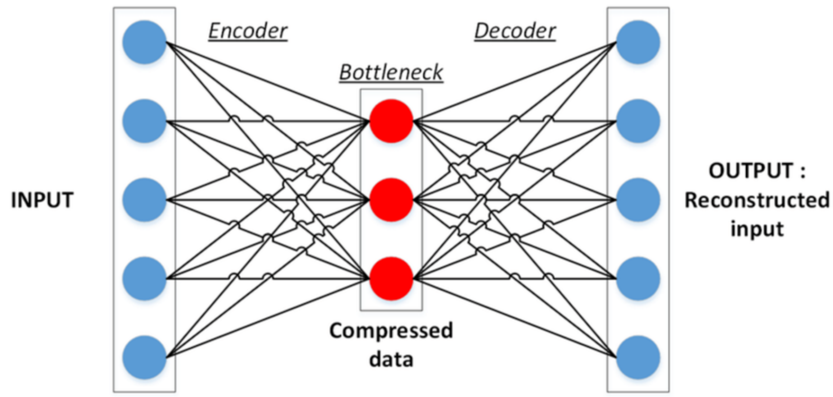
The paper investigates the creation of generative neural network models of typical reinforcement learning environments. Unsupervised training allows the world model to master a compact spatial and temporal description of the universe quickly. We can train a very compact and straightforward policy that can solve the necessary task using features derived from the world model as inputs to an agent. According to the paper, to deal with the massive volume of information that regularly runs through our lives, our brain learns an abstract representation of this information's spatial and temporal dimensions. We may witness a scene and recall an abstract explanation of it. Evidence also shows that our brain's prediction of the future based on the internal model governs what we experience at any given time. The researchers also demonstrate that, once the environment and predictive models have been trained, the controller model can be trained using data provided by the sensory and predictive models, allowing the intelligent agent to be trained in its dreams while still performing well in the intended environment. The aim is to create a reinforcement learning algorithm (an "agent") that improves driving a car around a 2D racetrack. This environment (Car Racing) is accessible through the OpenAI Gym. In conclusion, the main idea behind this paper is that since training agents in the real world are more expensive, world models that are incrementally conditioned to replicate reality can help move policies back to the real world.

ALGORITHMS

VAE (V) Model:

The Variational Auto Encoder which was referred to in the paper as V was trained first. At each time point, the environment provides our agent with a high-dimensional input observation. This input is usually a 2D picture frame from a video series. The V model's function is to learn an abstract, compact representation of each input frame that is observed. In other words, a Variational Auto-Encoder reduces the CNN outputs to a small number of features, modifies them slightly, and then extends them back to their original size. This causes the network to enter a dream-like state. In certain ways, this makes the picture shown by the MDN-RNN even more hazy and vague. Although it does increase

training time and money, it has a significant impact on the overall process by making the whole network even more robust.
Because many complex environments are stochastic in nature, we train our RNN to output a probability density function p(z)instead of a deterministic prediction of z.

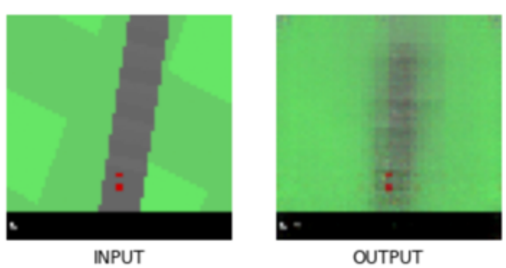


In [8]:
```
### output from the full_model
DIR_NAME = './data/rollout/'
file = os.listdir(DIR_NAME)[179]
obs_data = np.load(DIR_NAME + file)['obs']

obs = obs_data[50]
reconstruction = vae.full_model.predict(np.array([obs]))[0]

ax1 = plt.subplot(121)
plt.imshow( obs)
ax1.axis('off')
ax1.text(0.5,-0.1, "INPUT", size=12, ha="center",
         transform=ax1.transAxes)

ax2 = plt.subplot(122)
plt.imshow( reconstruction)
ax2.axis('off')
ax2.text(0.5,-0.1, "OUTPUT", size=12, ha="center",
         transform=ax2.transAxes);
```



MDD-RNN (M) Model:

The V model's job is to compress what the agent sees at and time frame, but the authors also want to compress what happens over time. The M model's job in this context is to forecast the future. The M model is a predictive model of the z vectors that V is predicted to generate in the future. The MDN-RNN enables the network to make more informed decisions. MDNs are an excellent way to model data, especially if the data being modeled has multiple states or is inherently a random variable that cannot be predicted with absolute certainty.

RESULT/PROCESS

1. Clone the repository from https://github.com/pantelis-classes/world-models-latest.git.
2. Build the Docker Container by running the following command.

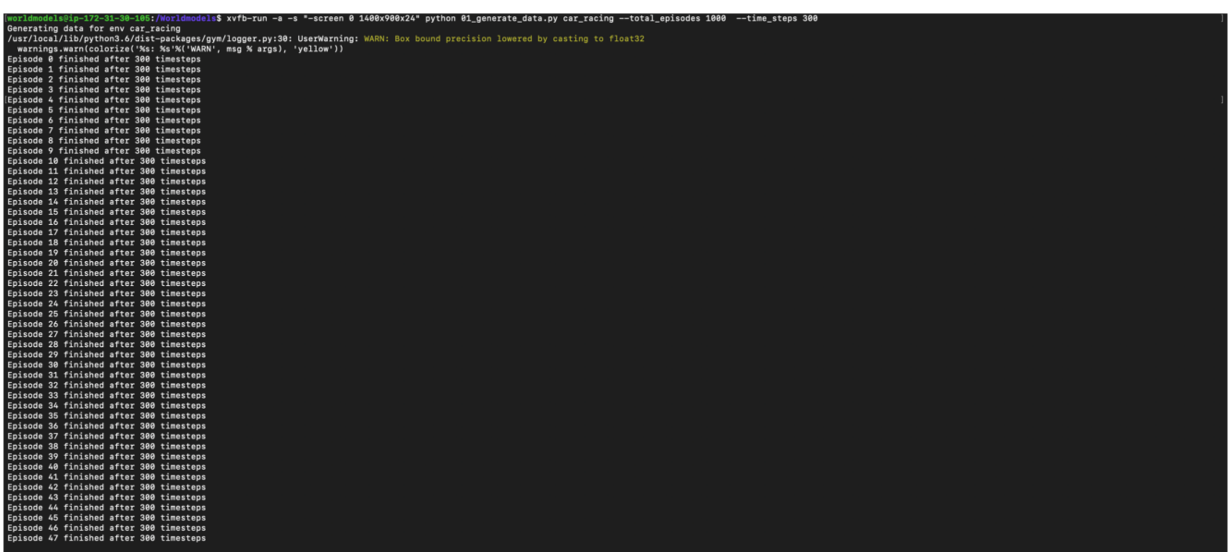
`ln -sf Dockerfile.cpu Dockerfile && docker build --network=host -t worldmodels-image-cpu .`

3. Run the docker container

`./launch-docker-cpu.sh $(pwd)`

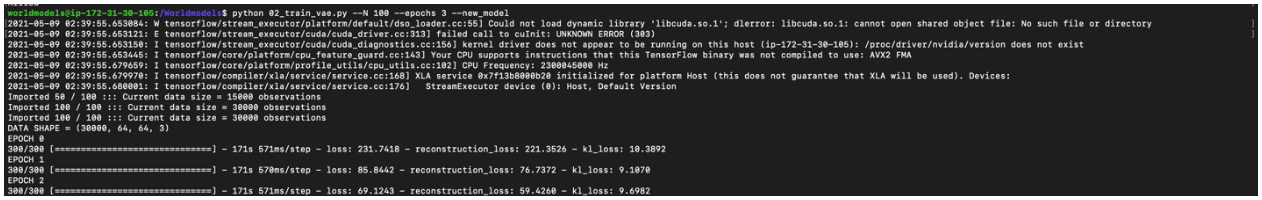
4. Generate the Random rollouts

`xvfb-run -a -s "-screen 0 1400x900x24" python 01_generate_data.py car_racing --total_episodes 2000 --start_batch 0 --time_steps 300`



5. We must train the VAE

python 02_train_vae.py --new_model



```
worldmodels@ip-172-31-30-105:/Worldmodels$ python 02_train_vae.py --N 100 --epochs 3 --new_model
2021-05-09 02:39:55.653084: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory
2021-05-09 02:39:55.653121: E tensorflow/stream_executor/cuda/cuda_driver.cc:313] failed call to cuInit: UNKNOWN ERROR (303)
2021-05-09 02:39:55.653150: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (ip-172-31-30-105): /proc/driver/nvidia/version does not exist
2021-05-09 02:39:55.653445: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
2021-05-09 02:39:55.679659: I tensorflow/core/platform/profile_utils/cpu_utils.cc:102] CPU Frequency: 2300045000 Hz
2021-05-09 02:39:55.679970: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7f13b8000b20 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2021-05-09 02:39:55.680001: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
Imported 50 / 100 ::: Current data size = 15000 observations
Imported 100 / 100 ::: Current data size = 30000 observations
Imported 100 / 100 ::: Current data size = 30000 observations
DATA SHAPE = (30000, 64, 64, 3)
EPOCH 0
300/300 [==============================] - 171s 571ms/step - loss: 231.7418 - reconstruction_loss: 221.3526 - kl_loss: 10.3892
EPOCH 1
300/300 [==============================] - 171s 570ms/step - loss: 85.8442 - reconstruction_loss: 76.7372 - kl_loss: 9.1070
EPOCH 2
300/300 [==============================] - 171s 571ms/step - loss: 69.1243 - reconstruction_loss: 59.4260 - kl_loss: 9.6982
```

6. Using the trained VAE, we must use it to generate the RNN Data

python 03_generate_rnn_data.py

```
worldmodels@ip-172-31-30-105:/Worldmodels$ python 03_generate_rnn_data.py
2021-05-09 03:31:48.436966: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'libcuda.so.1'; dlerror: libcud
a.so.1: cannot open shared object file: No such file or directory
2021-05-09 03:31:48.437002: E tensorflow/stream_executor/cuda/cuda_driver.cc:313] failed call to cuInit: UNKNOWN ERROR (303)
2021-05-09 03:31:48.437021: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (ip-172-31
-30-105): /proc/driver/nvidia/version does not exist
2021-05-09 03:31:48.438280: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary was not compil
ed to use: AVX2 FMA
2021-05-09 03:31:48.445015: I tensorflow/core/platform/profile_utils/cpu_utils.cc:102] CPU Frequency: 2300045000 Hz
2021-05-09 03:31:48.445341: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7f749c000b20 initialized for platform Host (this does not guar
antee that XLA will be used). Devices:
2021-05-09 03:31:48.445375: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
Encoded 50 / 5000 episodes
Encoded 100 / 5000 episodes
Encoded 150 / 5000 episodes
Encoded 200 / 5000 episodes
Encoded 250 / 5000 episodes
Encoded 300 / 5000 episodes
Encoded 350 / 5000 episodes
Encoded 400 / 5000 episodes
Encoded 450 / 5000 episodes
Encoded 500 / 5000 episodes
Encoded 550 / 5000 episodes
Encoded 600 / 5000 episodes
Encoded 650 / 5000 episodes
Encoded 700 / 5000 episodes
Encoded 750 / 5000 episodes
Encoded 800 / 5000 episodes
Encoded 850 / 5000 episodes
Encoded 900 / 5000 episodes
Encoded 950 / 5000 episodes
Encoded 1000 / 5000 episodes
Encoded 1050 / 5000 episodes
Encoded 1100 / 5000 episodes
Encoded 1150 / 5000 episodes
Encoded 1200 / 5000 episodes
Encoded 1250 / 5000 episodes
Encoded 1300 / 5000 episodes
Encoded 1350 / 5000 episodes
Encoded 1400 / 5000 episodes
Encoded 1450 / 5000 episodes
Encoded 1500 / 5000 episodes
Encoded 1550 / 5000 episodes
Encoded 1600 / 5000 episodes
Encoded 1650 / 5000 episodes
Encoded 1700 / 5000 episodes
Encoded 1750 / 5000 episodes
Encoded 1800 / 5000 episodes
Encoded 1850 / 5000 episodes
Encoded 1900 / 5000 episodes
Encoded 1950 / 5000 episodes
Encoded 2000 / 5000 episodes
```

7. Train the RNN data

python 04_train_rnn.py --new_model

```
worldmodels@ip-172-31-30-105:/Worldmodels$ python 04_train_rnn.py --new_model --batch_size 100
2021-05-09 05:49:47.692086: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory
2021-05-09 05:49:47.692137: E tensorflow/stream_executor/cuda/cuda_driver.cc:313] failed call to cuInit: UNKNOWN ERROR (303)
2021-05-09 05:49:47.692157: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (ip-172-31-30-105): /proc/driver/nvidia/version does not exist
2021-05-09 05:49:47.692388: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
2021-05-09 05:49:47.699447: I tensorflow/core/platform/profile_utils/cpu_utils.cc:102] CPU Frequency: 2300045000 Hz
2021-05-09 05:49:47.699771: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7f3174000b20 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2021-05-09 05:49:47.699802: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
STEP 0
1/1 [==============================] - 0s 504us/step - loss: 2.1876 - rnn_z_loss: 1.4467 - rnn_rew_loss: 0.7409
STEP 1
1/1 [==============================] - 0s 498us/step - loss: 2.0014 - rnn_z_loss: 1.4295 - rnn_rew_loss: 0.5718
STEP 2
1/1 [==============================] - 0s 505us/step - loss: 1.8639 - rnn_z_loss: 1.4227 - rnn_rew_loss: 0.4413
STEP 3
1/1 [==============================] - 0s 488us/step - loss: 1.7420 - rnn_z_loss: 1.4103 - rnn_rew_loss: 0.3317
STEP 4
1/1 [==============================] - 0s 489us/step - loss: 1.6285 - rnn_z_loss: 1.3915 - rnn_rew_loss: 0.2369
STEP 5
1/1 [==============================] - 0s 493us/step - loss: 1.5586 - rnn_z_loss: 1.3745 - rnn_rew_loss: 0.1840
STEP 6
1/1 [==============================] - 0s 489us/step - loss: 1.5469 - rnn_z_loss: 1.3658 - rnn_rew_loss: 0.1812
STEP 7
1/1 [==============================] - 0s 511us/step - loss: 1.5211 - rnn_z_loss: 1.3525 - rnn_rew_loss: 0.1686
STEP 8
1/1 [==============================] - 0s 533us/step - loss: 1.5170 - rnn_z_loss: 1.3359 - rnn_rew_loss: 0.1811
STEP 9
1/1 [==============================] - 0s 520us/step - loss: 1.4941 - rnn_z_loss: 1.3161 - rnn_rew_loss: 0.1780
STEP 10
1/1 [==============================] - 0s 484us/step - loss: 1.4763 - rnn_z_loss: 1.3020 - rnn_rew_loss: 0.1743
STEP 11
1/1 [==============================] - 0s 489us/step - loss: 1.4574 - rnn_z_loss: 1.2842 - rnn_rew_loss: 0.1732
STEP 12
1/1 [==============================] - 0s 475us/step - loss: 1.4484 - rnn_z_loss: 1.2773 - rnn_rew_loss: 0.1711
STEP 13
1/1 [==============================] - 0s 479us/step - loss: 1.4264 - rnn_z_loss: 1.2668 - rnn_rew_loss: 0.1596
STEP 14
1/1 [==============================] - 0s 504us/step - loss: 1.4170 - rnn_z_loss: 1.2556 - rnn_rew_loss: 0.1615
STEP 15
1/1 [==============================] - 0s 516us/step - loss: 1.4066 - rnn_z_loss: 1.2461 - rnn_rew_loss: 0.1605
STEP 16
```

```
1/1 [==============================] - 0s 487us/step - loss: 1.1839 - rnn_z_loss: 1.1525 - rnn_rew_loss: 0.0313
STEP 3969
1/1 [==============================] - 0s 504us/step - loss: 1.1778 - rnn_z_loss: 1.1526 - rnn_rew_loss: 0.0252
STEP 3970
1/1 [==============================] - 0s 503us/step - loss: 1.1810 - rnn_z_loss: 1.1522 - rnn_rew_loss: 0.0289
STEP 3971
1/1 [==============================] - 0s 504us/step - loss: 1.1802 - rnn_z_loss: 1.1535 - rnn_rew_loss: 0.0267
STEP 3972
1/1 [==============================] - 0s 492us/step - loss: 1.1829 - rnn_z_loss: 1.1506 - rnn_rew_loss: 0.0323
STEP 3973
1/1 [==============================] - 0s 482us/step - loss: 1.1809 - rnn_z_loss: 1.1511 - rnn_rew_loss: 0.0298
STEP 3974
1/1 [==============================] - 0s 483us/step - loss: 1.1847 - rnn_z_loss: 1.1517 - rnn_rew_loss: 0.0330
STEP 3975
1/1 [==============================] - 0s 487us/step - loss: 1.1787 - rnn_z_loss: 1.1526 - rnn_rew_loss: 0.0261
STEP 3976
1/1 [==============================] - 0s 508us/step - loss: 1.1861 - rnn_z_loss: 1.1520 - rnn_rew_loss: 0.0341
STEP 3977
1/1 [==============================] - 0s 496us/step - loss: 1.1786 - rnn_z_loss: 1.1520 - rnn_rew_loss: 0.0266
STEP 3978
1/1 [==============================] - 0s 515us/step - loss: 1.1845 - rnn_z_loss: 1.1516 - rnn_rew_loss: 0.0329
STEP 3979
1/1 [==============================] - 0s 476us/step - loss: 1.1841 - rnn_z_loss: 1.1489 - rnn_rew_loss: 0.0351
STEP 3980
1/1 [==============================] - 0s 484us/step - loss: 1.1774 - rnn_z_loss: 1.1511 - rnn_rew_loss: 0.0263
STEP 3981
1/1 [==============================] - 0s 505us/step - loss: 1.1829 - rnn_z_loss: 1.1531 - rnn_rew_loss: 0.0298
STEP 3982
1/1 [==============================] - 0s 482us/step - loss: 1.1785 - rnn_z_loss: 1.1506 - rnn_rew_loss: 0.0280
STEP 3983
1/1 [==============================] - 0s 491us/step - loss: 1.1782 - rnn_z_loss: 1.1504 - rnn_rew_loss: 0.0278
STEP 3984
1/1 [==============================] - 0s 497us/step - loss: 1.1745 - rnn_z_loss: 1.1520 - rnn_rew_loss: 0.0225
STEP 3985
1/1 [==============================] - 0s 502us/step - loss: 1.1790 - rnn_z_loss: 1.1492 - rnn_rew_loss: 0.0298
STEP 3986
1/1 [==============================] - 0s 500us/step - loss: 1.1809 - rnn_z_loss: 1.1526 - rnn_rew_loss: 0.0283
STEP 3987
1/1 [==============================] - 0s 472us/step - loss: 1.1803 - rnn_z_loss: 1.1526 - rnn_rew_loss: 0.0278
STEP 3988
1/1 [==============================] - 0s 506us/step - loss: 1.1775 - rnn_z_loss: 1.1522 - rnn_rew_loss: 0.0253
STEP 3989
1/1 [==============================] - 0s 512us/step - loss: 1.1797 - rnn_z_loss: 1.1516 - rnn_rew_loss: 0.0281
STEP 3990
1/1 [==============================] - 0s 530us/step - loss: 1.1783 - rnn_z_loss: 1.1508 - rnn_rew_loss: 0.0276
STEP 3991
1/1 [==============================] - 0s 510us/step - loss: 1.1764 - rnn_z_loss: 1.1512 - rnn_rew_loss: 0.0252
STEP 3992
1/1 [==============================] - 0s 527us/step - loss: 1.1807 - rnn_z_loss: 1.1515 - rnn_rew_loss: 0.0292
STEP 3993
1/1 [==============================] - 0s 490us/step - loss: 1.1762 - rnn_z_loss: 1.1514 - rnn_rew_loss: 0.0249
STEP 3994
1/1 [==============================] - 0s 518us/step - loss: 1.1832 - rnn_z_loss: 1.1505 - rnn_rew_loss: 0.0326
STEP 3995
1/1 [==============================] - 0s 498us/step - loss: 1.1771 - rnn_z_loss: 1.1510 - rnn_rew_loss: 0.0261
STEP 3996
1/1 [==============================] - 0s 485us/step - loss: 1.1855 - rnn_z_loss: 1.1513 - rnn_rew_loss: 0.0341
STEP 3997
1/1 [==============================] - 0s 508us/step - loss: 1.1829 - rnn_z_loss: 1.1513 - rnn_rew_loss: 0.0317
STEP 3998
1/1 [==============================] - 0s 479us/step - loss: 1.1769 - rnn_z_loss: 1.1517 - rnn_rew_loss: 0.0252
STEP 3999
1/1 [==============================] - 0s 492us/step - loss: 1.1785 - rnn_z_loss: 1.1511 - rnn_rew_loss: 0.0274
```

8. Train the Controller

```
xvfb-run -s "-screen 0 1400x900x24" python 05_train_controller.py car_racing --num_worker 16 --num_worker_trial 2 --num_episode 4 --max_length 1000 --eval_steps 25
```

```
worldmodels@ip-172-31-30-105:/Worldmodels$ xvfb-run -a -s "-screen 0 1400x900x24" python 05_train_controller.py car_racing -n 4 -t 1 -e 1 --max_length 1000
['mpirun', '-np', '5', '/usr/local/bin/python', '05_train_controller.py', 'car_racing', '-n', '4', '-t', '1', '-e', '1', '--max_length', '1000']
assigning the rank and nworkers 5 1
2021-05-09 15:33:04.970952: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory
2021-05-09 15:33:04.971009: E tensorflow/stream_executor/cuda/cuda_driver.cc:313] failed call to cuInit: UNKNOWN ERROR (303)
2021-05-09 15:33:04.971044: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (ip-172-31-30-105): /proc/driver/nvidia/version does not exist
2021-05-09 15:33:04.971293: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
2021-05-09 15:33:04.991909: I tensorflow/core/platform/profile_utils/cpu_utils.cc:102] CPU Frequency: 2300045000 Hz
2021-05-09 15:33:04.999886: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7f1728000b20 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2021-05-09 15:33:04.999980: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
assigning the rank and nworkers 5 3
assigning the rank and nworkers 5 0
2021-05-09 15:33:05.060094: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory
2021-05-09 15:33:05.060910: E tensorflow/stream_executor/cuda/cuda_driver.cc:313] failed call to cuInit: UNKNOWN ERROR (303)
2021-05-09 15:33:05.061496: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (ip-172-31-30-105): /proc/driver/nvidia/version does not exist
2021-05-09 15:33:05.062351: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
2021-05-09 15:33:05.063180: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory
2021-05-09 15:33:05.063267: E tensorflow/stream_executor/cuda/cuda_driver.cc:313] failed call to cuInit: UNKNOWN ERROR (303)
2021-05-09 15:33:05.063358: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (ip-172-31-30-105): /proc/driver/nvidia/version does not exist
2021-05-09 15:33:05.063733: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
2021-05-09 15:33:05.072453: I tensorflow/core/platform/profile_utils/cpu_utils.cc:102] CPU Frequency: 2300045000 Hz
2021-05-09 15:33:05.072824: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7f4538000b20 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2021-05-09 15:33:05.072848: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
2021-05-09 15:33:05.084340: I tensorflow/core/platform/profile_utils/cpu_utils.cc:102] CPU Frequency: 2300045000 Hz
2021-05-09 15:33:05.085347: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7fe198000b20 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2021-05-09 15:33:05.085381: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
assigning the rank and nworkers 5 4
2021-05-09 15:33:05.176699: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory
2021-05-09 15:33:05.176803: E tensorflow/stream_executor/cuda/cuda_driver.cc:313] failed call to cuInit: UNKNOWN ERROR (303)
2021-05-09 15:33:05.176853: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (ip-172-31-30-105): /proc/driver/nvidia/version does not exist
2021-05-09 15:33:05.177180: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
2021-05-09 15:33:05.195643: I tensorflow/core/platform/profile_utils/cpu_utils.cc:102] CPU Frequency: 2300045000 Hz
2021-05-09 15:33:05.196736: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7fde74000b20 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2021-05-09 15:33:05.196774: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
assigning the rank and nworkers 5 2
2021-05-09 15:33:05.234836: W tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load dynamic library 'libcuda.so.1'; dlerror: libcuda.so.1: cannot open shared object file: No such file or directory
2021-05-09 15:33:05.234942: E tensorflow/stream_executor/cuda/cuda_driver.cc:313] failed call to cuInit: UNKNOWN ERROR (303)
2021-05-09 15:33:05.234994: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:156] kernel driver does not appear to be running on this host (ip-172-31-30-105): /proc/driver/nvidia/version does not exist
2021-05-09 15:33:05.235261: I tensorflow/core/platform/cpu_feature_guard.cc:143] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
2021-05-09 15:33:05.242842: I tensorflow/core/platform/profile_utils/cpu_utils.cc:102] CPU Frequency: 2300045000 Hz
2021-05-09 15:33:05.243233: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x7f034c000b20 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2021-05-09 15:33:05.243265: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): Host, Default Version
(2_w,4mirr1)-aCMA-ES (mu_w=1.5,w_1=80%) in dimension 867 (seed=609479, Sun May  9 15:33:06 2021)
('process', 4, 'out of total ', 5, 'started')
(2_w,4mirr1)-aCMA-ES (mu_w=1.5,w_1=80%) in dimension 867 (seed=656724, Sun May  9 15:33:06 2021)
('process', 1, 'out of total ', 5, 'started')
(2_w,4mirr1)-aCMA-ES (mu_w=1.5,w_1=80%) in dimension 867 (seed=695096, Sun May  9 15:33:06 2021)
('process', 3, 'out of total ', 5, 'started')
(2_w,4mirr1)-aCMA-ES (mu_w=1.5,w_1=80%) in dimension 867 (seed=689820, Sun May  9 15:33:06 2021)
('process', 0, 'out of total ', 5, 'started')
('training', 'car_racing')
('population', 4)
('num_worker', 4)
('num_worker_trial', 1)
('num_episode', 1)
('max_length', 1000)
(2_w,4mirr1)-aCMA-ES (mu_w=1.5,w_1=80%) in dimension 867 (seed=640537, Sun May  9 15:33:06 2021)
('process', 2, 'out of total ', 5, 'started')
/usr/local/lib/python3.6/dist-packages/gym/logger.py:30: UserWarning: WARN: Box bound precision lowered by casting to float32
  warnings.warn(colorize('%s: %s'%('WARN', msg % args), 'yellow'))
/usr/local/lib/python3.6/dist-packages/gym/logger.py:30: UserWarning: WARN: Box bound precision lowered by casting to float32
  warnings.warn(colorize('%s: %s'%('WARN', msg % args), 'yellow'))
/usr/local/lib/python3.6/dist-packages/gym/logger.py:30: UserWarning: WARN: Box bound precision lowered by casting to float32
  warnings.warn(colorize('%s: %s'%('WARN', msg % args), 'yellow'))
/usr/local/lib/python3.6/dist-packages/gym/logger.py:30: UserWarning: WARN: Box bound precision lowered by casting to float32
  warnings.warn(colorize('%s: %s'%('WARN', msg % args), 'yellow'))
('car_racing', (1, 118, -23.35, -31.14, -13.9, 7.87, 0.09946, 1000.0, 1000))
/usr/local/lib/python3.6/dist-packages/gym/logger.py:30: UserWarning: WARN: Box bound precision lowered by casting to float32
  warnings.warn(colorize('%s: %s'%('WARN', msg % args), 'yellow'))
/usr/local/lib/python3.6/dist-packages/gym/logger.py:30: UserWarning: WARN: Box bound precision lowered by casting to float32
  warnings.warn(colorize('%s: %s'%('WARN', msg % args), 'yellow'))
/usr/local/lib/python3.6/dist-packages/gym/logger.py:30: UserWarning: WARN: Box bound precision lowered by casting to float32
  warnings.warn(colorize('%s: %s'%('WARN', msg % args), 'yellow'))
/usr/local/lib/python3.6/dist-packages/gym/logger.py:30: UserWarning: WARN: Box bound precision lowered by casting to float32
```

```
('car_racing', (21, 2455, -55.62, -67.15, -45.25, 7.86, 0.09122, 1000.0, 1000))
('car_racing', (22, 2574, -59.59, -70.0, -53.33, 6.24, 0.09088, 1000.0, 1000))
('car_racing', (23, 2691, -57.55, -62.14, -52.56, 3.5, 0.09055, 1000.0, 1000))
('car_racing', (24, 2807, -53.69, -57.69, -49.82, 2.78, 0.09023, 1000.0, 1000))
('car_racing', (25, 2925, -51.77, -60.28, -39.39, 7.91, 0.08991, 1000.0, 1000))
[-53.0201 -60.1449 -42.8571 -51.3108]
[-51.833225]
('improvement', 25, -28.483224999999997, 'curr', -51.833225, 'prev', -23.35, 'best', -51.833225)
('car_racing', (26, 3158, -36.68, -53.9, -19.41, 15.54, 0.0896, 1000.0, 1000))
('car_racing', (27, 3277, -56.11, -78.19, -37.69, 15.34, 0.08929, 1000.0, 1000))
('car_racing', (28, 3392, -44.35, -51.66, -36.32, 5.46, 0.08899, 1000.0, 1000))
('car_racing', (29, 3511, -35.46, -38.56, -34.06, 1.84, 0.0887, 1000.0, 1000))
('car_racing', (30, 3629, -28.64, -58.46, -10.54, 18.11, 0.0884, 1000.0, 1000))
('car_racing', (31, 3746, -53.61, -76.66, -30.0, 19.31, 0.08812, 1000.0, 1000))
('car_racing', (32, 3865, -33.81, -62.83, -8.78, 19.56, 0.08784, 1000.0, 1000))
('car_racing', (33, 3982, -46.33, -58.21, -41.17, 6.97, 0.08756, 1000.0, 1000))
('car_racing', (34, 4100, -44.61, -63.08, -29.32, 12.92, 0.08729, 1000.0, 1000))
('car_racing', (35, 4217, -51.59, -61.16, -37.5, 8.83, 0.08702, 1000.0, 1000))
('car_racing', (36, 4333, -53.09, -77.69, -28.81, 21.18, 0.08675, 1000.0, 1000))
('car_racing', (37, 4453, -54.81, -77.41, -32.92, 15.86, 0.08649, 1000.0, 1000))
('car_racing', (38, 4569, -39.33, -50.76, -32.2, 7.02, 0.08623, 1000.0, 1000))
('car_racing', (39, 4688, -33.49, -42.96, -20.15, 9.0, 0.08597, 1000.0, 1000))
('car_racing', (40, 4806, -28.89, -42.12, -19.23, 9.83, 0.08572, 1000.0, 1000))
('car_racing', (41, 4921, -17.22, -29.05, 4.72, 13.19, 0.08547, 1000.0, 1000))
('car_racing', (42, 5040, -24.62, -44.8, -4.76, 15.48, 0.08523, 1000.0, 1000))
('car_racing', (43, 5157, -15.11, -23.32, -5.36, 6.43, 0.08499, 1000.0, 1000))
('car_racing', (44, 5277, -8.2, -20.15, 6.46, 9.59, 0.08475, 1000.0, 1000))
('car_racing', (45, 5394, -27.03, -45.01, -17.52, 10.88, 0.08452, 1000.0, 1000))
('car_racing', (46, 5511, -8.65, -35.81, 21.21, 21.99, 0.0843, 1000.0, 1000))
('car_racing', (47, 5628, -12.21, -22.55, 15.26, 15.93, 0.08408, 1000.0, 1000))
('car_racing', (48, 5743, 14.3, -19.41, 57.5, 27.64, 0.08385, 1000.0, 1000))
('car_racing', (49, 5861, -3.57, -81.87, 59.85, 51.24, 0.08364, 1000.0, 1000))
('car_racing', (50, 5980, 38.42, -52.54, 114.59, 74.99, 0.08342, 1000.0, 1000))
[ 67.7852  48.5507 -60.7142 -58.8014]
[-0.7949249999999974]
('improvement', 50, 51.0383, 'curr', -0.7949249999999974, 'prev', -51.833225, 'best', -0.7949249999999974)
('car_racing', (51, 6216, 8.37, -75.52, 57.34, 54.59, 0.08321, 1000.0, 1000))
('car_racing', (52, 6333, 26.77, -31.74, 64.7, 39.25, 0.083, 1000.0, 1000))
('car_racing', (53, 6453, -12.96, -42.12, 15.75, 24.92, 0.08279, 1000.0, 1000))
('car_racing', (54, 6571, 26.43, 24.99, 29.77, 1.96, 0.08259, 1000.0, 1000))
('car_racing', (55, 6688, -25.56, -67.97, 63.76, 52.47, 0.08238, 1000.0, 1000))
('car_racing', (56, 6806, -21.94, -55.78, 56.46, 45.51, 0.08218, 1000.0, 1000))
('car_racing', (57, 6924, 31.19, -10.95, 63.26, 29.55, 0.08198, 1000.0, 1000))
('car_racing', (58, 7042, 11.27, -21.98, 52.54, 27.24, 0.08179, 1000.0, 1000))
('car_racing', (59, 7163, 0.53, -27.2, 49.42, 29.99, 0.08159, 1000.0, 1000))
('car_racing', (60, 7279, 54.88, 20.56, 77.3, 22.19, 0.0814, 1000.0, 1000))
('car_racing', (61, 7397, 13.17, -11.18, 57.89, 26.87, 0.08121, 1000.0, 1000))
('car_racing', (62, 7515, 8.54, -11.03, 56.58, 27.85, 0.08102, 1000.0, 1000))
('car_racing', (63, 7633, 34.72, 20.37, 67.21, 18.94, 0.08083, 1000.0, 1000))
('car_racing', (64, 7753, -20.38, -35.81, 6.66, 16.22, 0.08065, 1000.0, 1000))
('car_racing', (65, 7868, -56.24, -75.86, -16.05, 24.17, 0.08046, 1000.0, 1000))
('car_racing', (66, 7988, -3.44, -25.8, 16.73, 17.31, 0.08027, 1000.0, 1000))
('car_racing', (67, 8105, 14.86, -19.29, 52.72, 32.66, 0.08009, 1000.0, 1000))
('car_racing', (68, 8220, -10.31, -41.78, 6.16, 19.55, 0.07991, 1000.0, 1000))
('car_racing', (69, 8340, -15.78, -28.79, 7.63, 14.94, 0.07973, 1000.0, 1000))
('car_racing', (70, 8459, -11.89, -22.82, 2.23, 11.16, 0.07956, 1000.0, 1000))
('car_racing', (71, 8580, 4.96, -2.43, 14.69, 6.16, 0.07939, 1000.0, 1000))
('car_racing', (72, 8697, -6.17, -17.49, 13.13, 12.27, 0.07922, 1000.0, 1000))
('car_racing', (73, 8813, -3.5, -36.39, 18.95, 21.41, 0.07905, 1000.0, 1000))
('car_racing', (74, 8933, -15.85, -56.14, 8.01, 24.15, 0.07888, 1000.0, 1000))
('car_racing', (75, 9049, 4.18, -10.39, 14.86, 9.19, 0.07872, 1000.0, 1000))
[-16.1073  -2.1739  28.5714  23.5955]
[8.471425]
('improvement', 75, 9.266349999999997, 'curr', 8.471425, 'prev', -0.7949249999999974, 'best', 8.471425)
('car_racing', (76, 9284, 2.26, -40.52, 27.79, 26.7, 0.07856, 1000.0, 1000))
('car_racing', (77, 9401, 31.71, 13.47, 48.69, 13.16, 0.0784, 1000.0, 1000))
('car_racing', (78, 9522, -10.58, -52.68, 24.99, 28.81, 0.07824, 1000.0, 1000))
('car_racing', (79, 9641, -9.91, -18.84, 5.08, 9.84, 0.07809, 1000.0, 1000))
('car_racing', (80, 9761, 11.18, -15.45, 30.84, 17.41, 0.07793, 1000.0, 1000))
('car_racing', (81, 9879, 16.19, -16.96, 36.69, 20.0, 0.07778, 1000.0, 1000))
('car_racing', (82, 9999, -4.75, -24.91, 21.49, 19.29, 0.07763, 1000.0, 1000))
('car_racing', (83, 10120, -4.78, -40.34, 41.34, 31.48, 0.07748, 1000.0, 1000))
('car_racing', (84, 10236, 15.76, -0.68, 25.44, 10.02, 0.07734, 1000.0, 1000))
('car_racing', (85, 10355, -15.9, -35.69, 7.14, 18.16, 0.07719, 1000.0, 1000))
```

9. Finally, Visualising the agent

```
python model.py car_racing --filename ./controller/car_racing.cma.4.32.best.json --
render_mode –record_video
```

Task 3

Based on the results, we can come to the following conclusion. In principle, by combining a variational autoencoder and a generative adversarial network, the VAE reconstruction objective can be based on learned feature representations in the GAN method. As a result, we will replace element-wise errors with feature-wise errors to better capture the data distribution while providing invariance. We apply our approach to world models in terms of visual fidelity and show that it outperforms VAEs with element-wise similarity tests. Furthermore, we will demonstrate that the method learns an embedding in which high-level abstract visual features can be modified using simple arithmetic. Also, we must keep in mind that the GAN method needs more training for a better recreation of images.