RoboSkate

Final Project Presentation

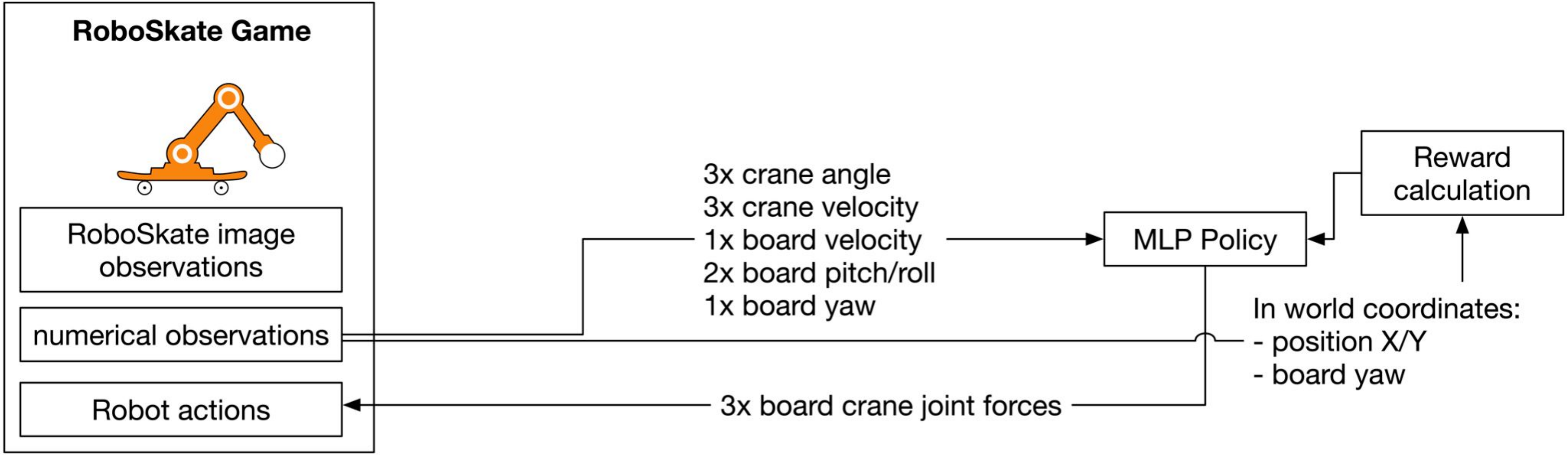
Michelle Bettendorf
Gintautas Palinauskas
Meriç Sakarya
Batuhan Yumurtacı
Finn Süberkrüb

Cloud-Based Machine Learning in Robotics
Summer Semester 2021

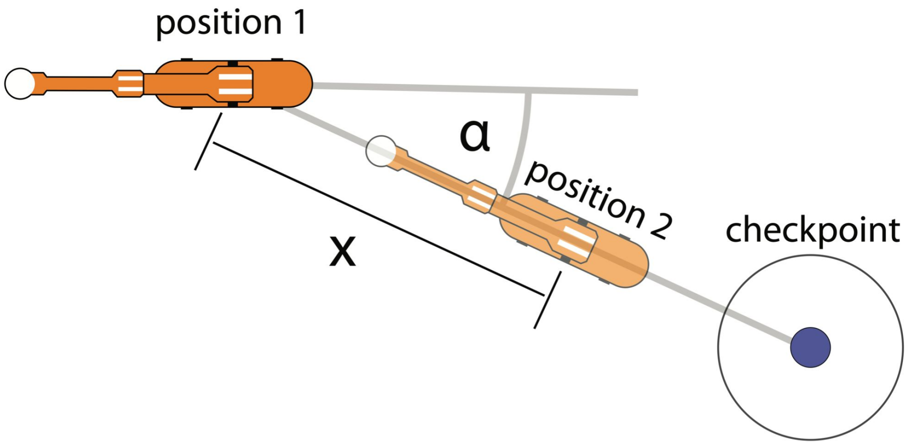
Agenda

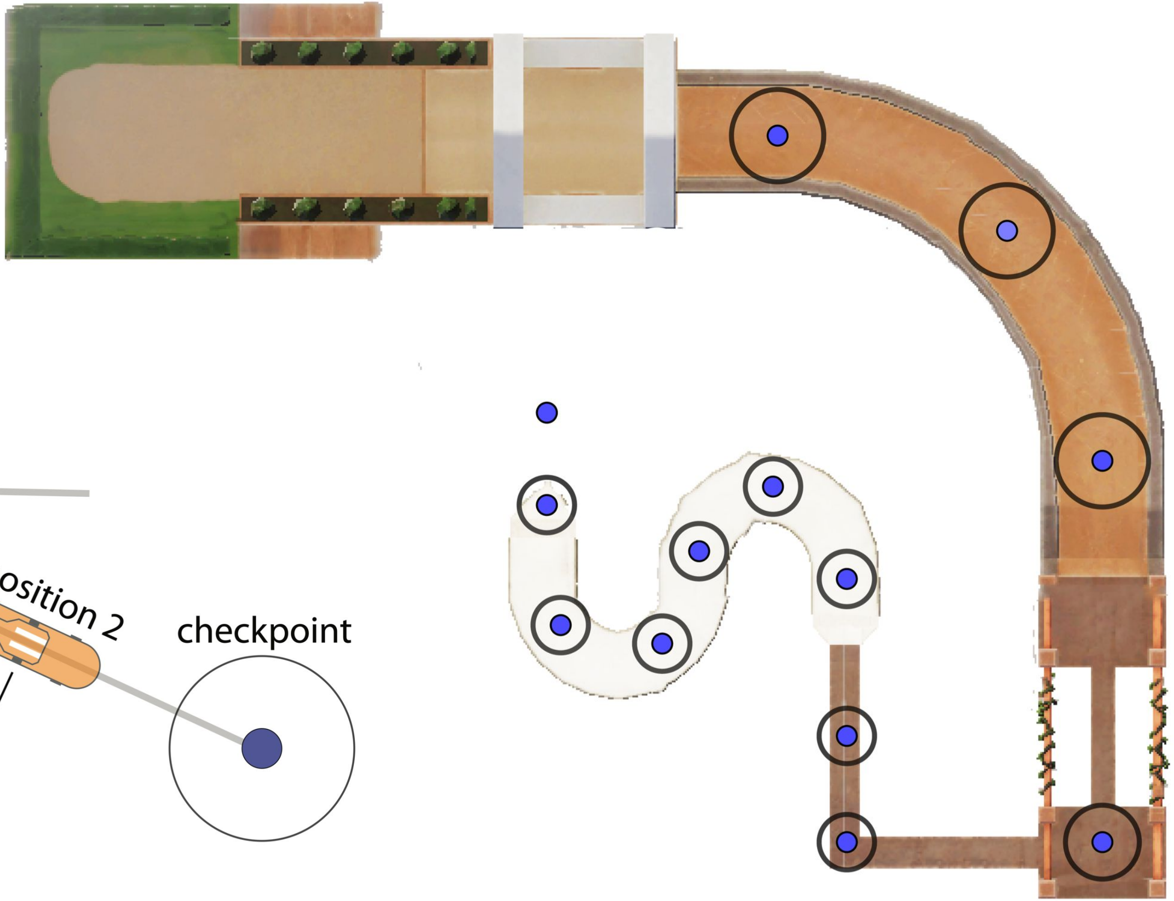
1. **RL algorithm structure**
2. **Results (numerical)**
3. **Image feature extractor**
   a. **Image preprocessing**
   b. **VAE**
   c. **Image segmentation**
4. **Behavioral cloning**
5. **NRP**
6. **Suggestions for RoboSkate and NRP**

RL Algorithm structure

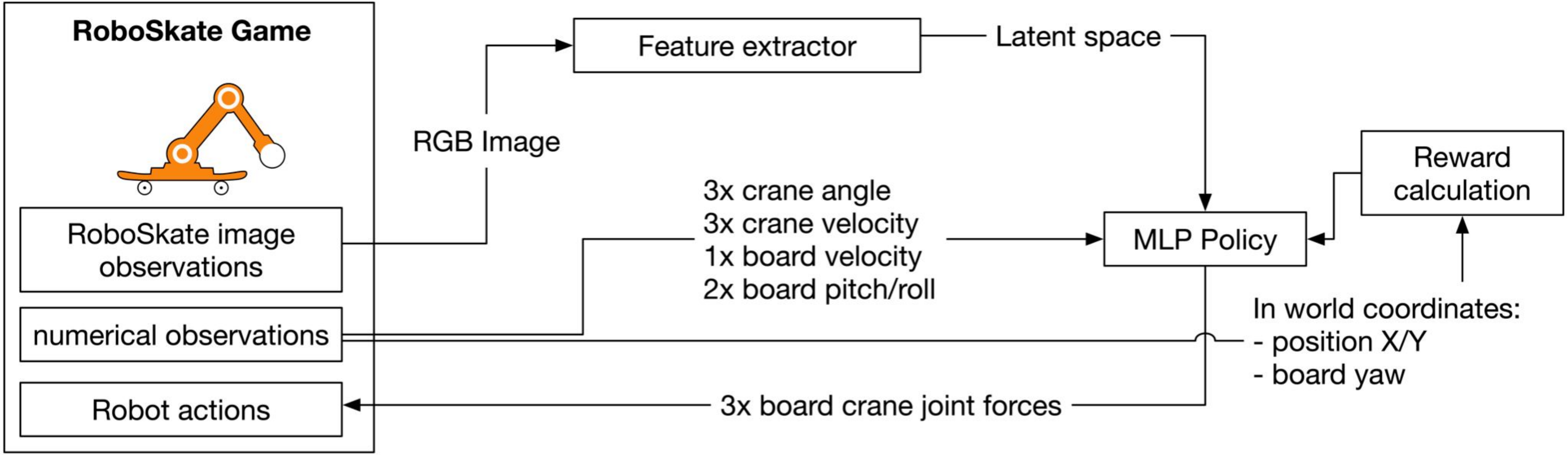


Checkpoints





RL Algorithm structure

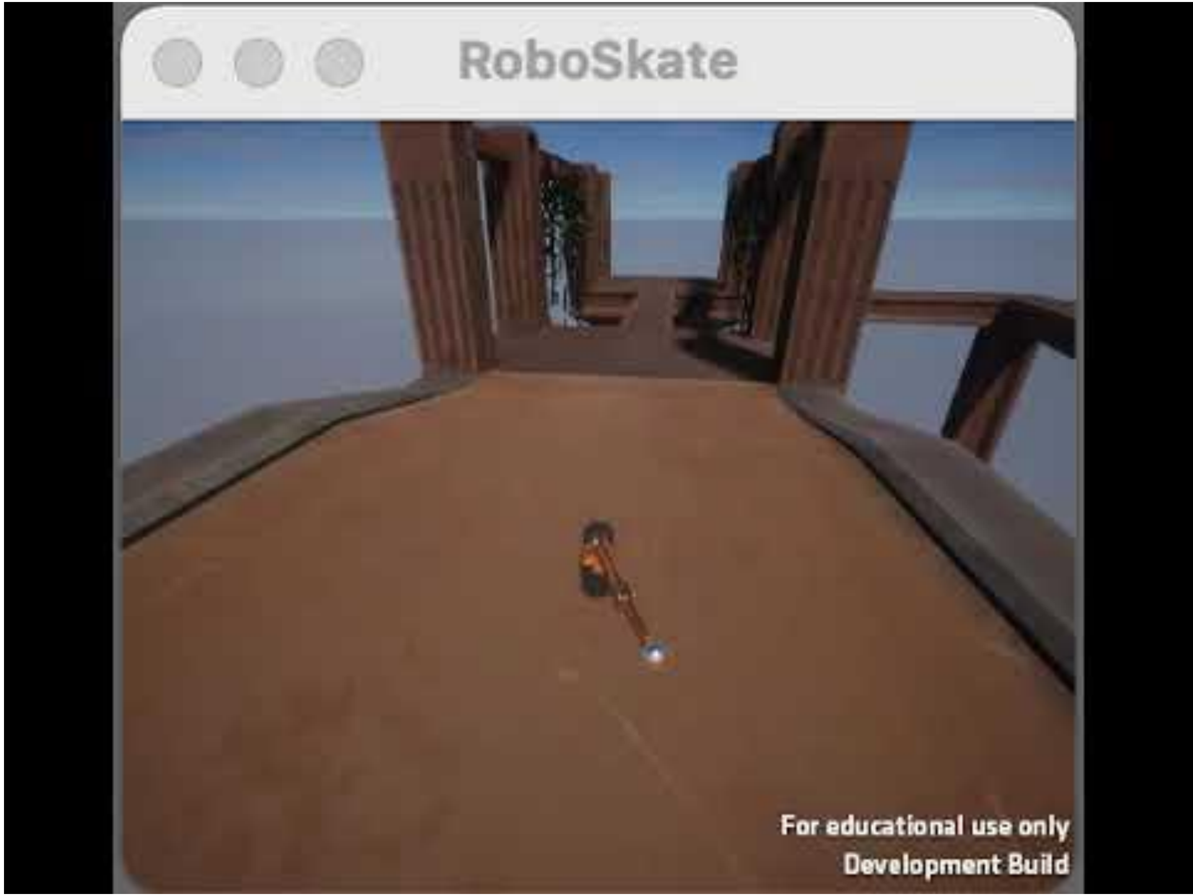


Results

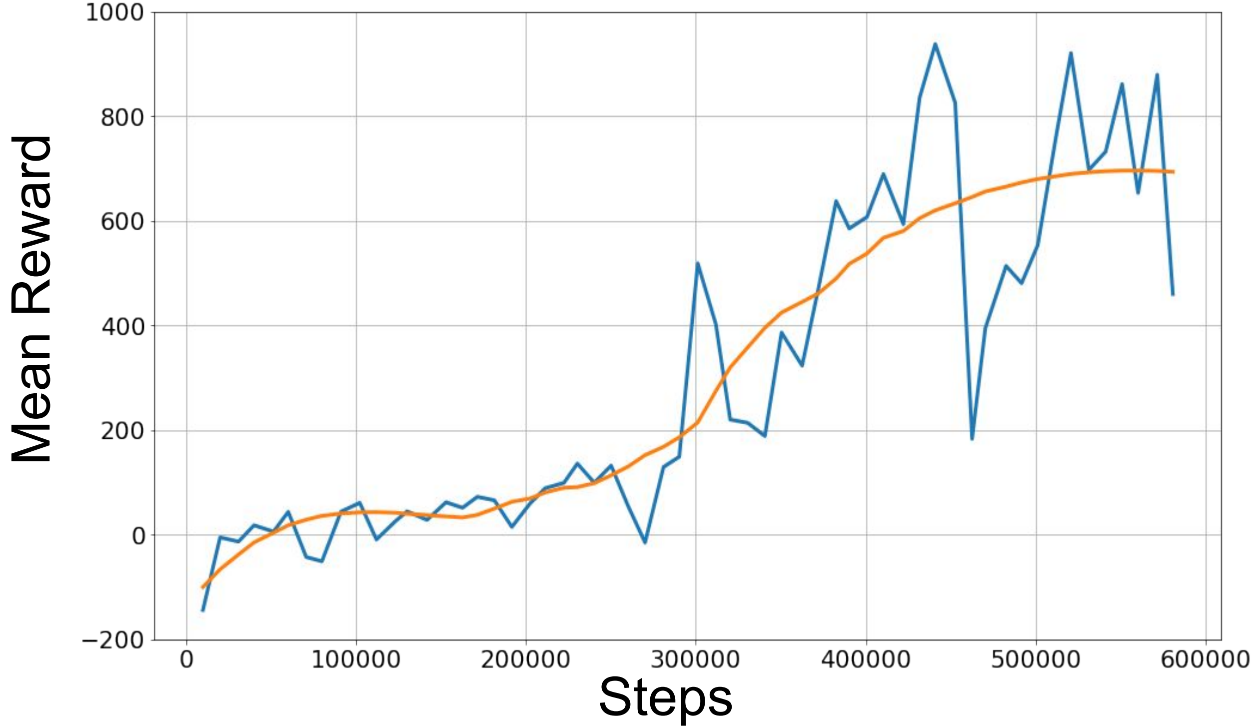
RoboSkate Numerical Agent

Time to finish level 1-3:

0:50

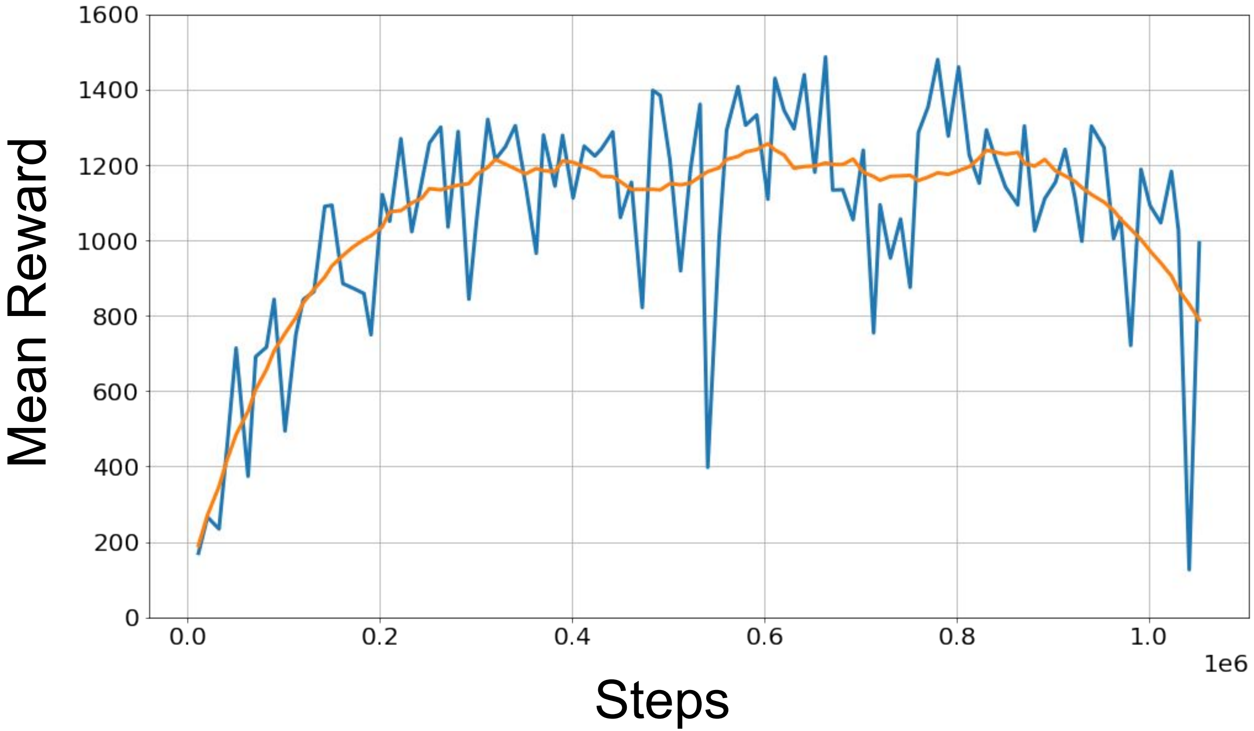


Learning graphs from tensorboard SAC Agent v1



Training Time
10h 36m 6s

Learning graphs from tensorboard SAC Agent v2



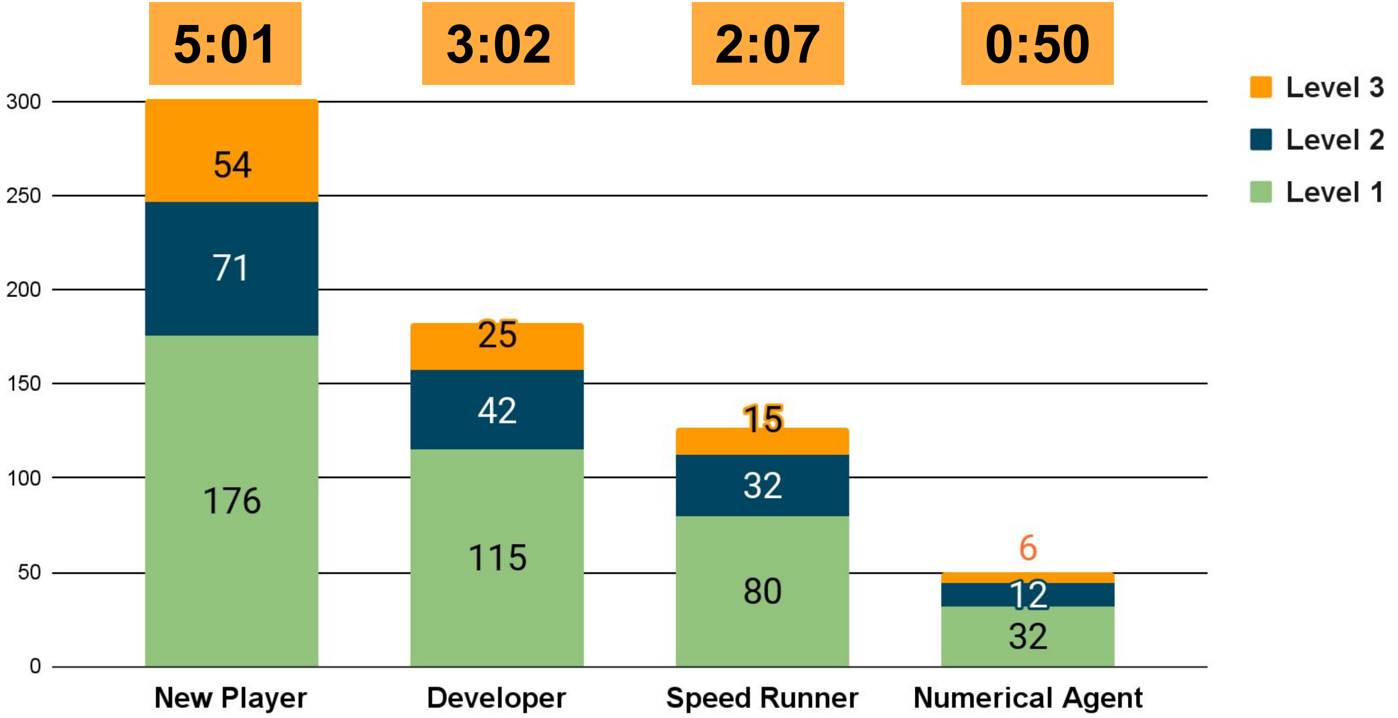
Uses v1 as pre-trained agent

Training Time 21h 9m 45s

Agent Training

New Player

Comparison between human and robot



RoboSkate Developer Playthrough by Matas Sakalauskas https://www.youtube.com/watch?v=_BWT11ttgJ4
Youtube Speedrunner https://www.youtube.com/watch?v=xL9v7Ss90LU

Image feature extractors

Image Preprocessing

Images: 100 x 100 pixels

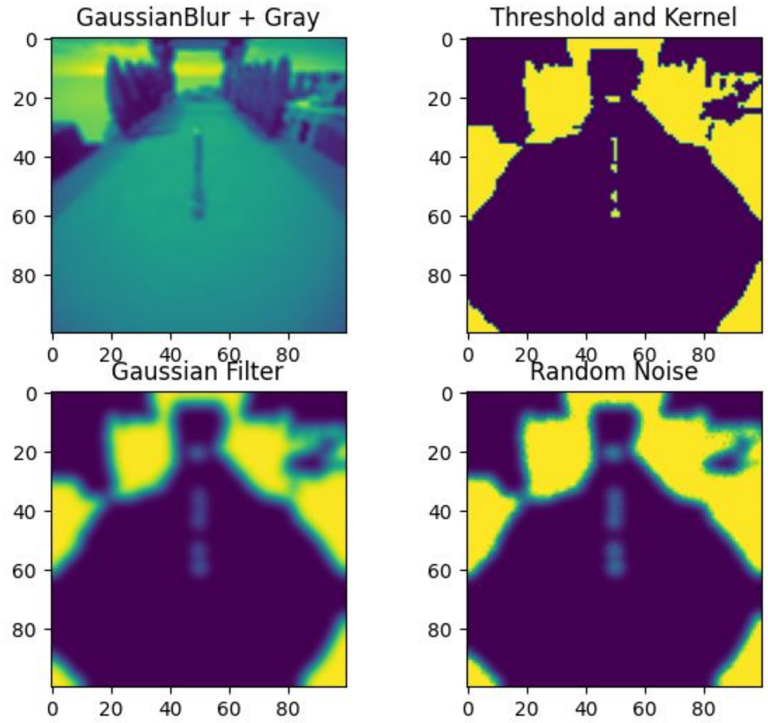


Image Preprocessing: Detecting the obstacles

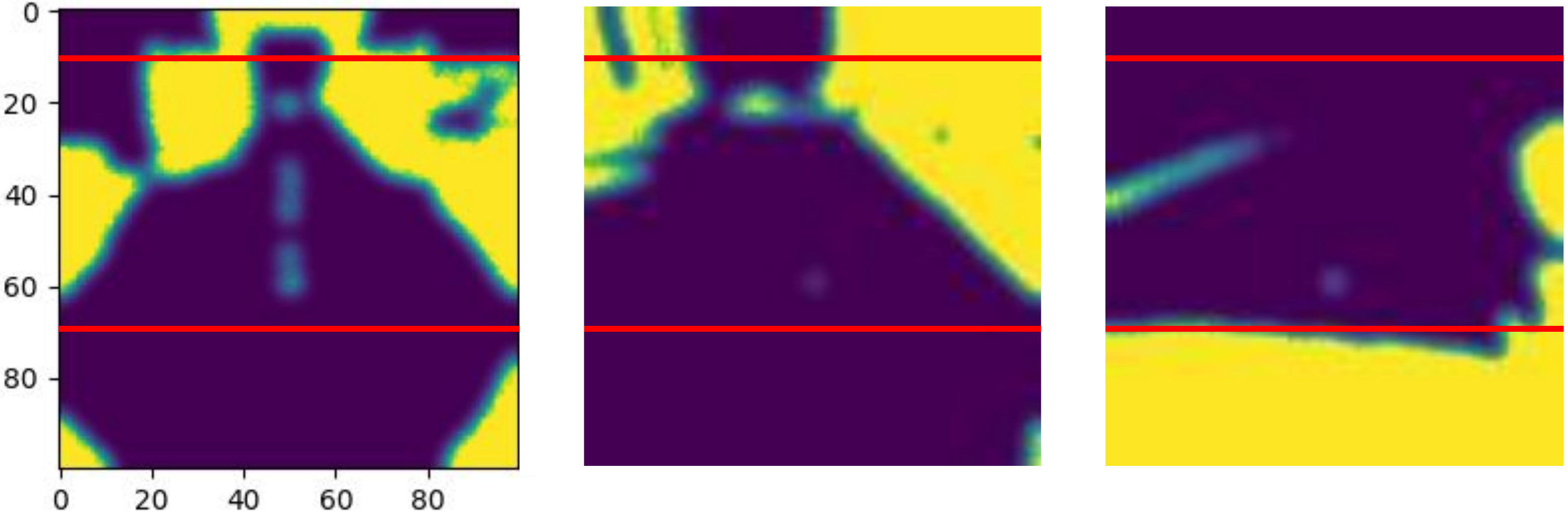
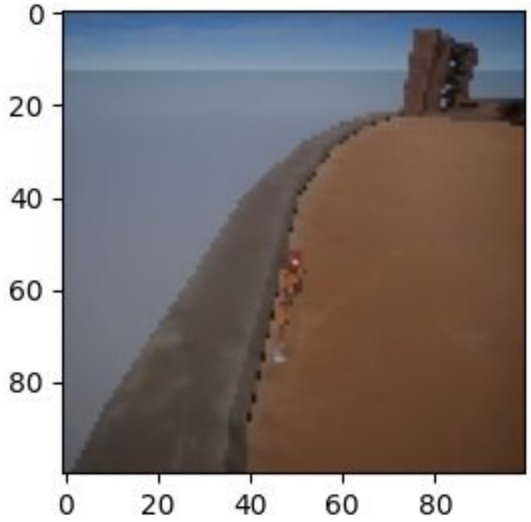


Image Preprocessing: Problems



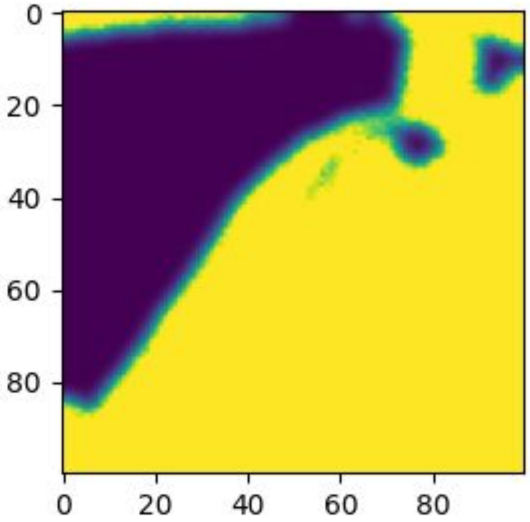
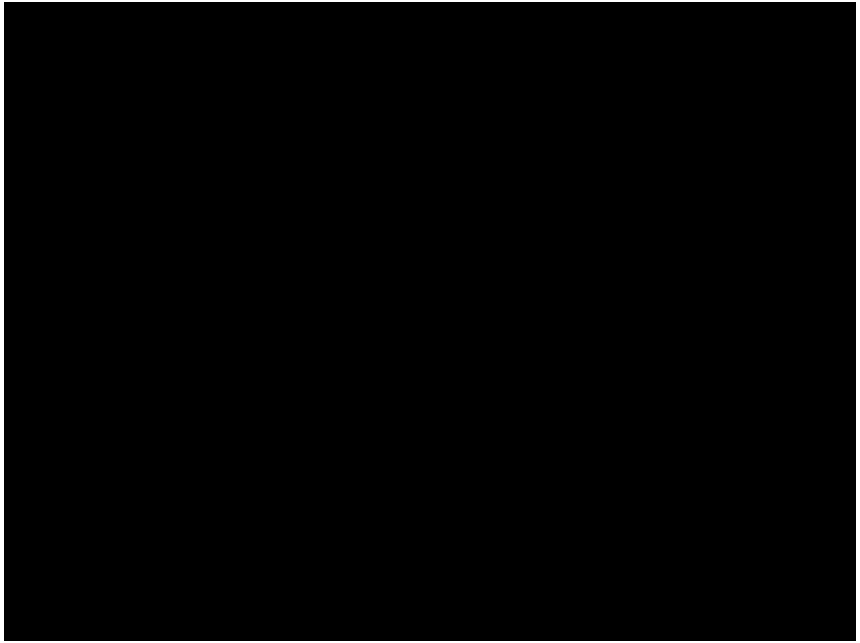


Image Preprocessing: Results



Level 1: average of 4:30 min
around 1800 pictures needed

Level 2: too much speed for the algorithm

VAE - image to image

- Dimensionality reduction
- Observing the latent space
- Faster training due to smaller observation shape
- Avoids overfitting

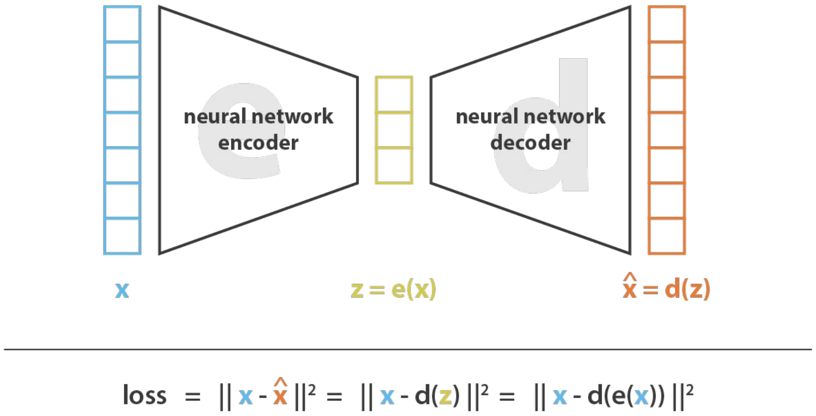


Illustration of an autoencoder with its loss function. [1]

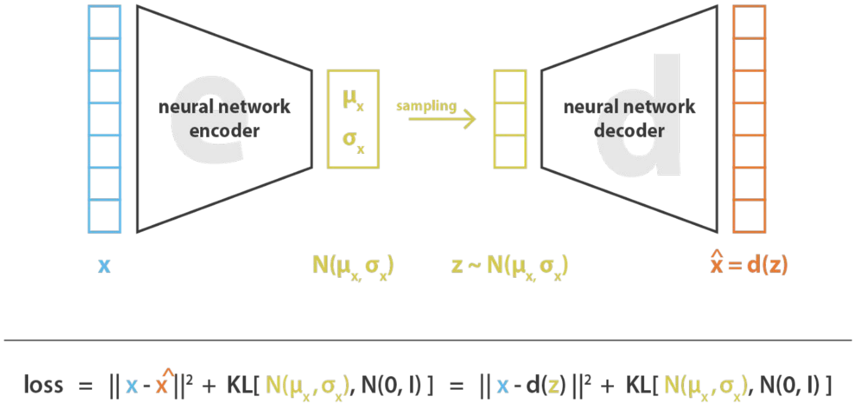
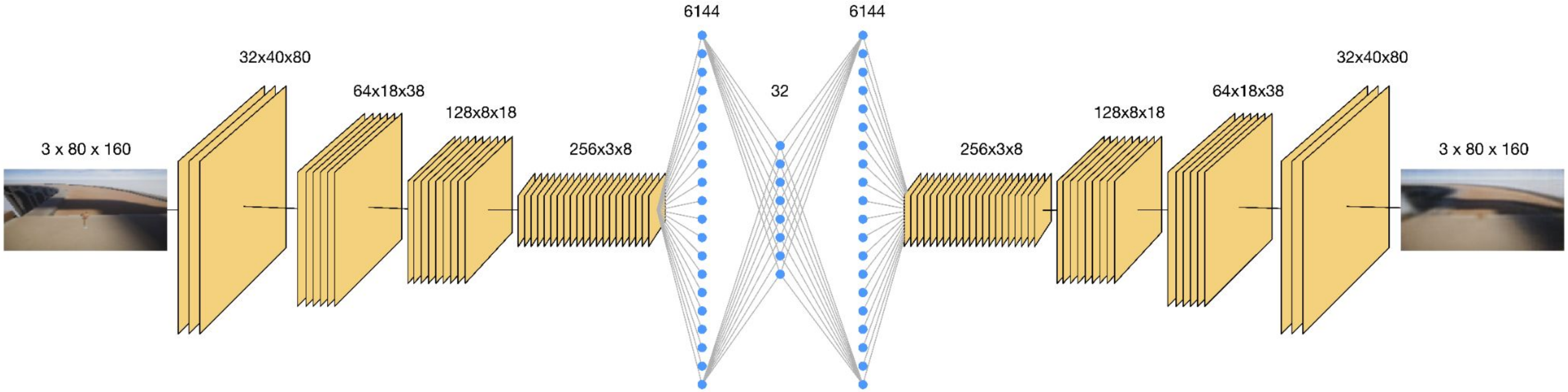


Illustration of a variational autoencoder. [1]

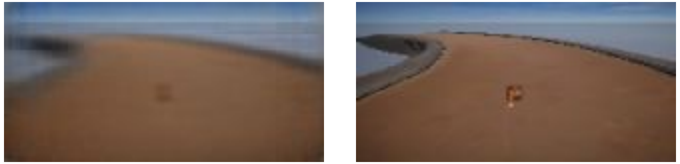
VAE - image to image



VAE - image to image

reconstructed original

Epoch 0

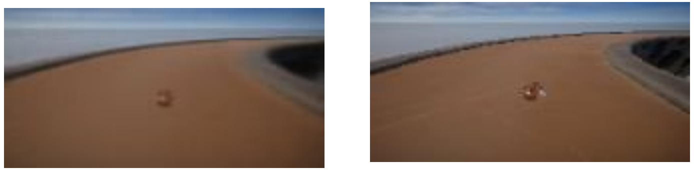


Epoch 10

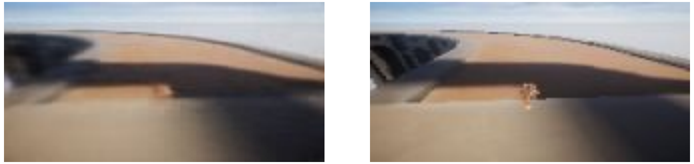
Epoch 50

reconstructed original

Epoch 100

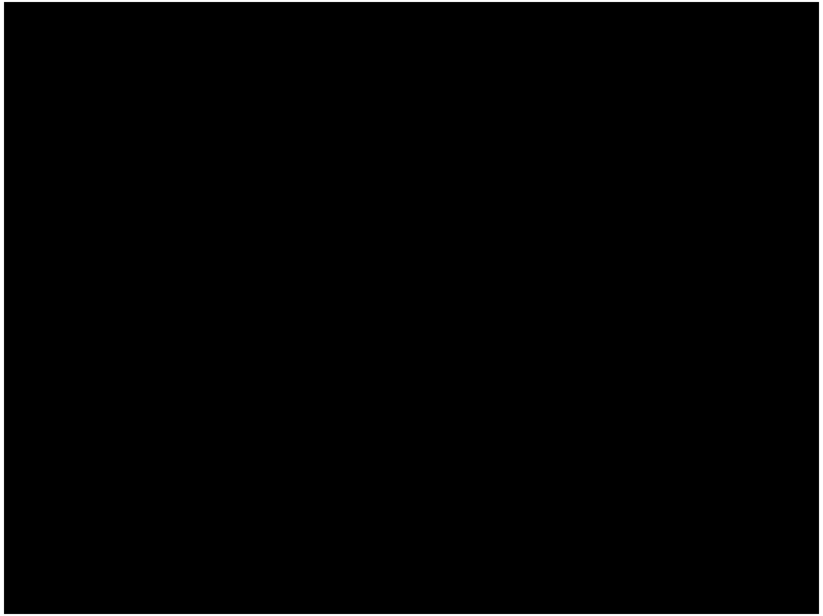


Epoch 150

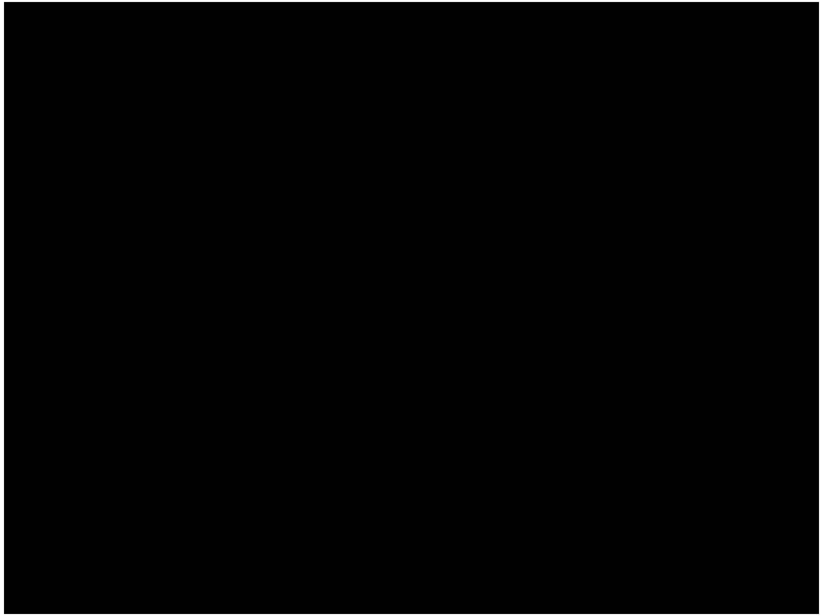


Epoch 200

VAE - image to image

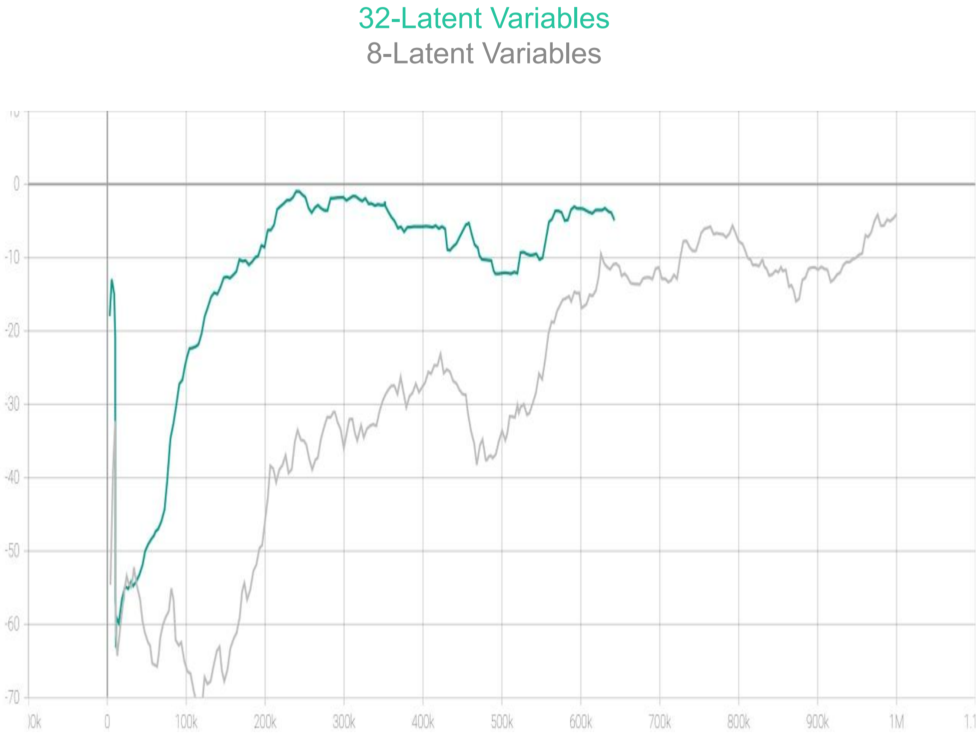


Latent Space: 32 variables



Latent Space: 8 variables

VAE - image to image



VAE - image segmentation



Input

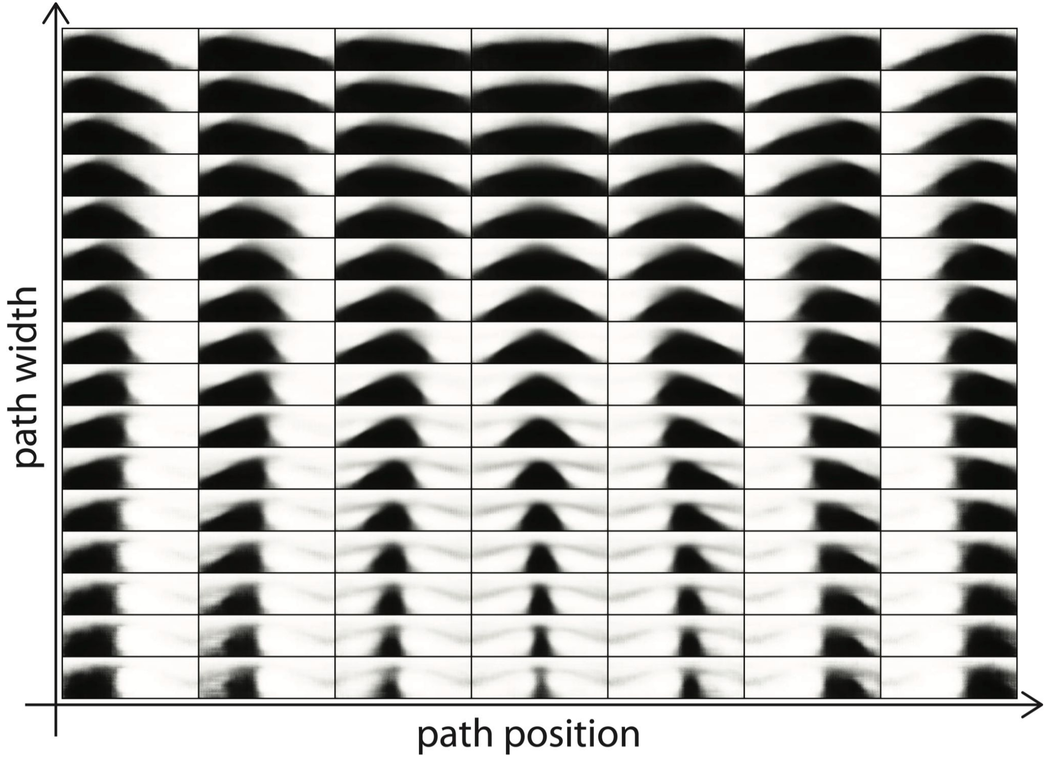
Output

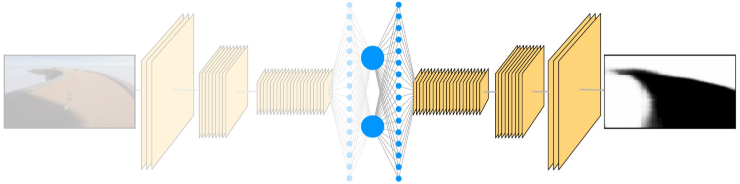
≈

Label

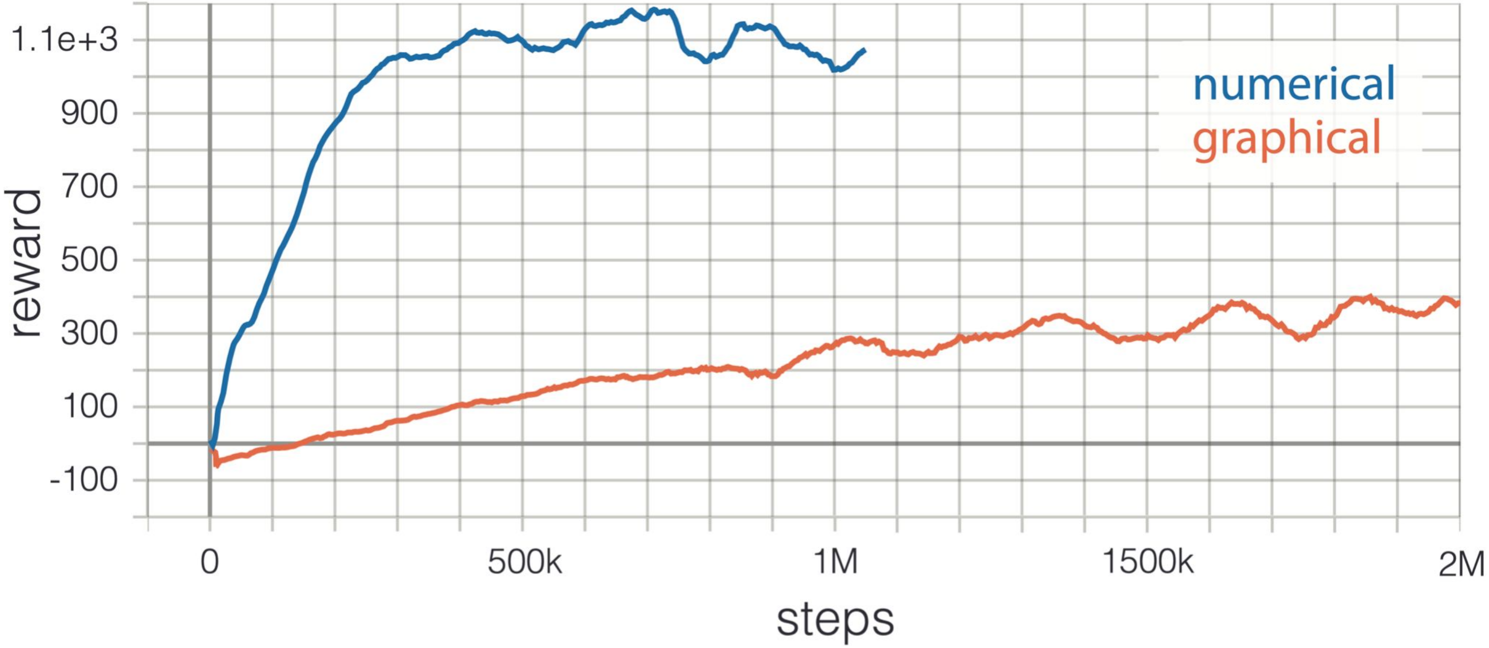


VAE - image segmentation - feature space





VAE - image segmentation - learning curve



VAE - image segmentation - agent level 1 & 3

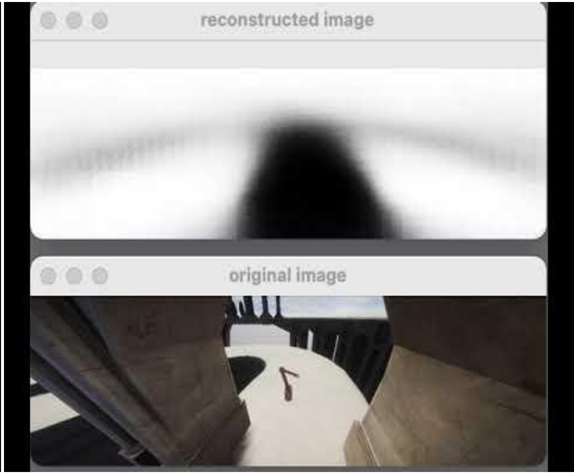
Level 1

Latent Space: 2 variables

Level 2

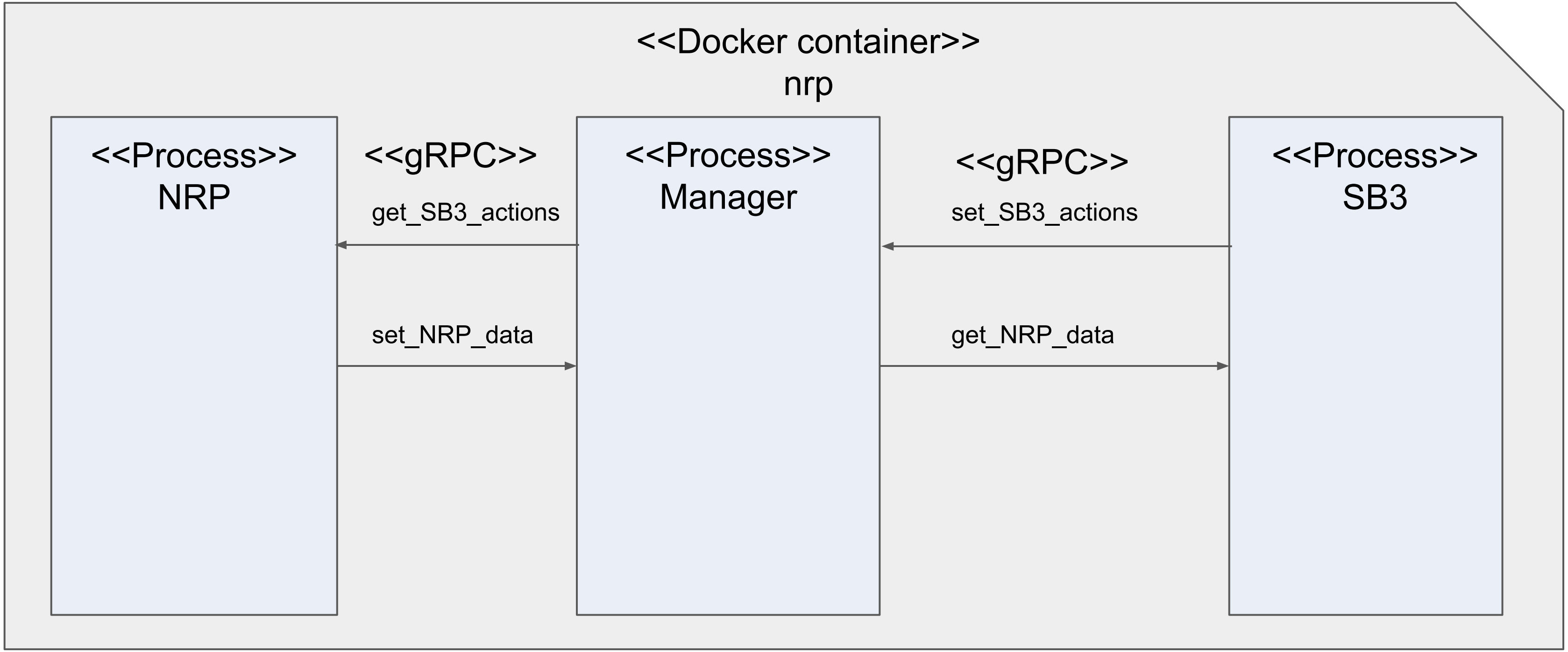
Latent Space: 8 variables

Level 3



Latent Space: 2 variables

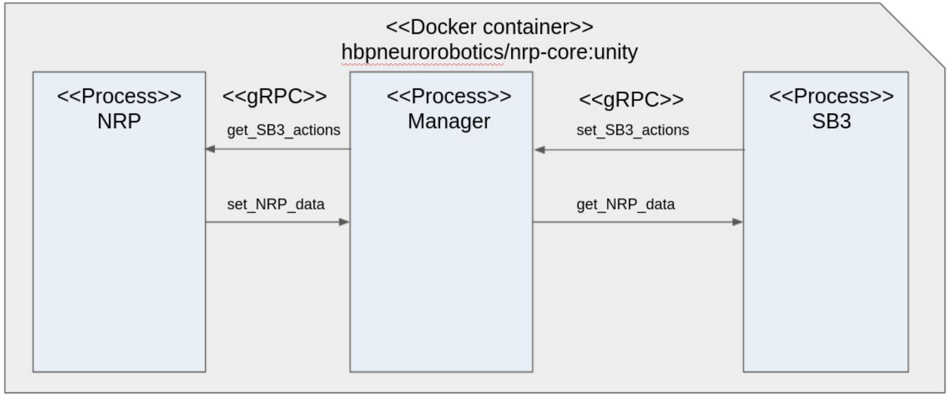
Stable Baseline 3 integration in the NRP (Dockerfiles)



Suggestions

NRP issues

1. Trained agent’s behaviour changes.
2. Synchronization between NRP and SB3.
3. There is no shutdown() or restart() method for NRP, which are needed to train agents with SB3.
4. Only importing locally libraries worked.



Suggestions

1. Unity control methods like restart(), shutdown() are useful.
2. Simpler way of retrieving data from unity.

RoboSkate for reinforcement learning issues

1. The game is GPU intensive.
2. Running without rendering (headless mode) gives out errors that don't disrupt the simulation.
3. NVIDIA docker images don't fully support Vulkan rendering engine.

Suggestions

1. Useful to control the rendering quality of the game.
2. Having a unity physics engine running without rendering.
3. Adjusting game running speed.
4. Changing Vulkan rendering engine to another.
5. Disable user visual output, but keep the image rendering.

Recap of the achievements

- Running the games locally
- Implementing custom gym env.
- Training with SAC, A2C, PPO on SB3
- Training in the cloud
- First successful numerical DB agent
- Successful image based DummyBall agent
- RS first real push forward
- Running standard env. with RL_zoo
- Teleoperation v1.0, v2.0 implemented
- Finishing level 1, 2, 3 with numerical RS
- Implemented a VAE
- Successfully ran SB3 with NRP
- Finishing level 1 with VAE segmentation RS
- Using IL on standard env., generating expert data for custom env.

Future plans

- Get training result of latent space size 2 VAE with RS
- Investigate the issues with VAE training
- Merge with master branch after clean up and thorough testing
- Detailed documentation

Lessons learned

- R&D in RL
- Different image segmentation methods
- Git, Docker, CLI, cloud ML
- Teamwork, Leadership, Coordination

RS - RoboSkate | DB - DummyBall | IL - Imitation learning

Responsibilities

	Preliminary Tasks
Michelle Bettendorf	Image preprocessing
Gintautas Palinauskas	Imitation learning, SB3 in NRP
Meriç Sakarya	VAE image to image
Batuhan Yumurtacı	Human-Agent benchmarking, evaluation of results
Finn Süberkrüb	Learning from numerical data, from segmentation and combined policy, VAE - with labeled images

Questions?

Michelle Bettendorf
Gintautas Palinauskas
Meriç Sakarya
Batuhan Yumurtacı
Finn Süberkrüb

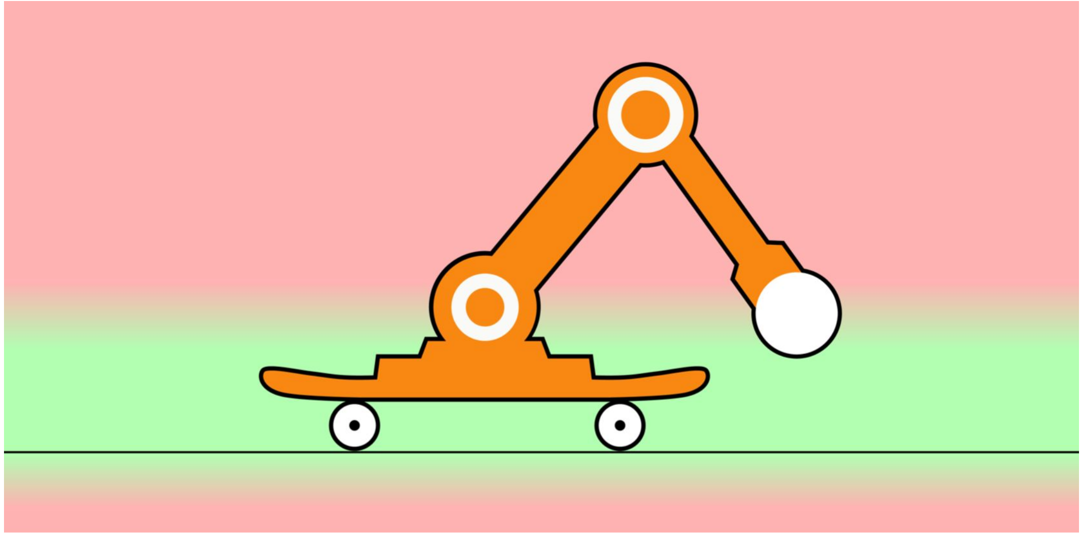
Cloud-Based Machine Learning in Robotics
Summer Semester 2021

Appendix

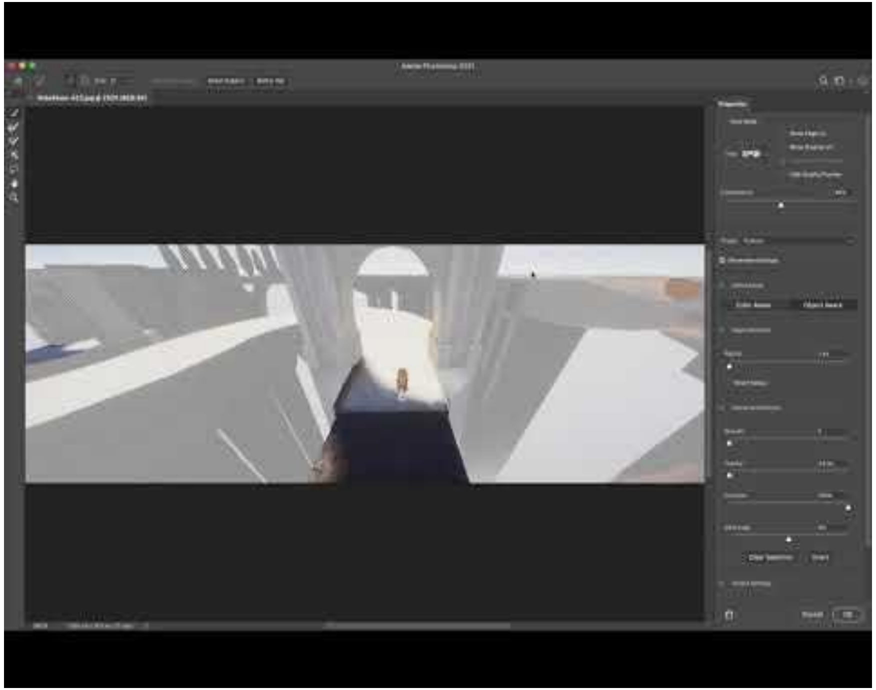
Resources summary

RoboSkate: https://store.steampowered.com/app/1404530/RoboSkate/
Unity headless: https://stackoverflow.com/questions/52316136/how-to-run-unity-in-headless-mode-on-linux
Unity without monitor: https://towardsdatascience.com/how-to-run-unity-on-amazon-cloud-or-without-monitor-3c10ce022639
Nvidia gpu container runtime: https://github.com/NVIDIA/nvidia-container-runtime
Nvidia Vulkan: https://hub.docker.com/r/nvidia/vulkan
SB3: https://stable-baselines3.readthedocs.io/en/master/
RL_zoo: https://github.com/DLR-RM/rl-baselines3-zoo
Imitation: https://github.com/HumanCompatibleAI/imitation

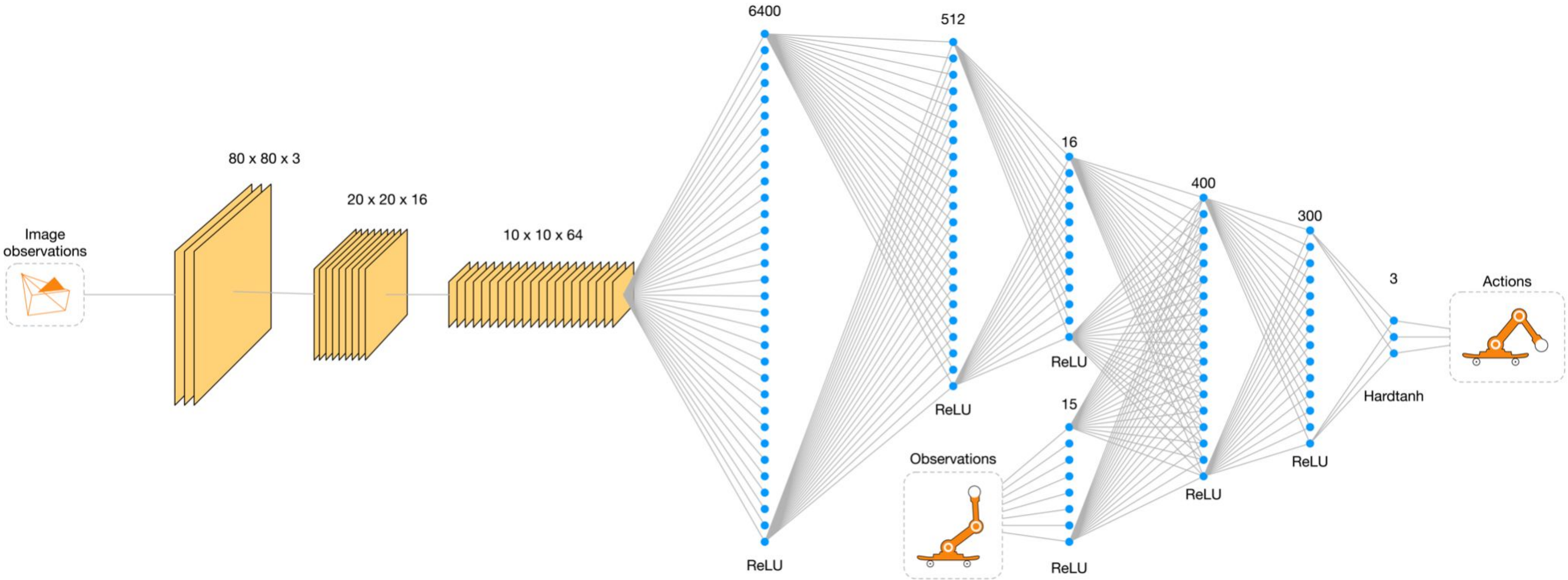
Accelerated learning through new starting position



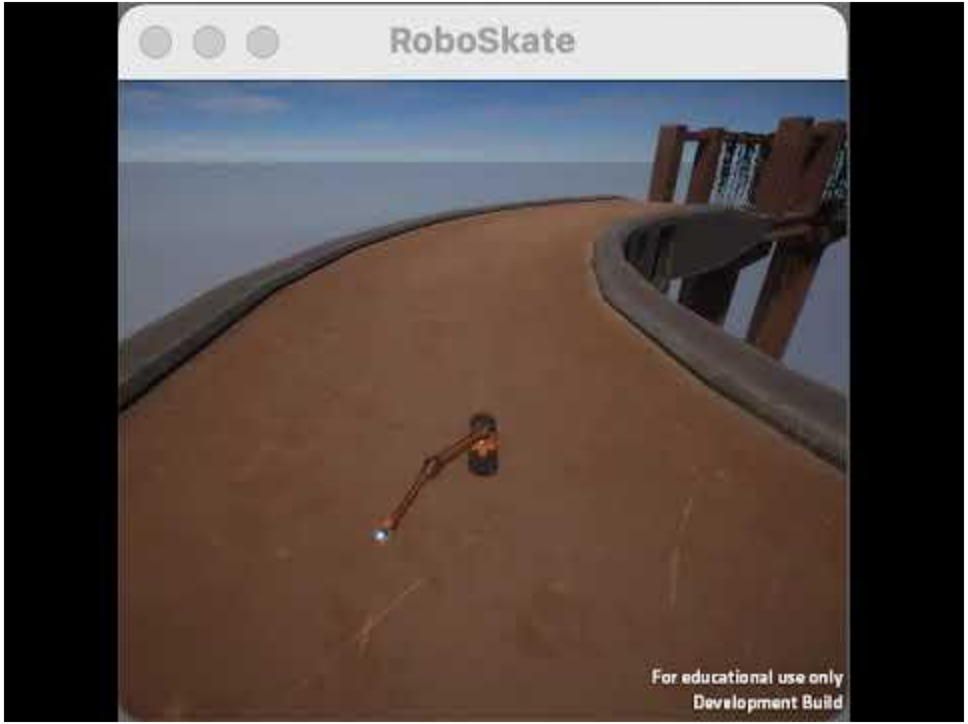
RoboSkate image labelling



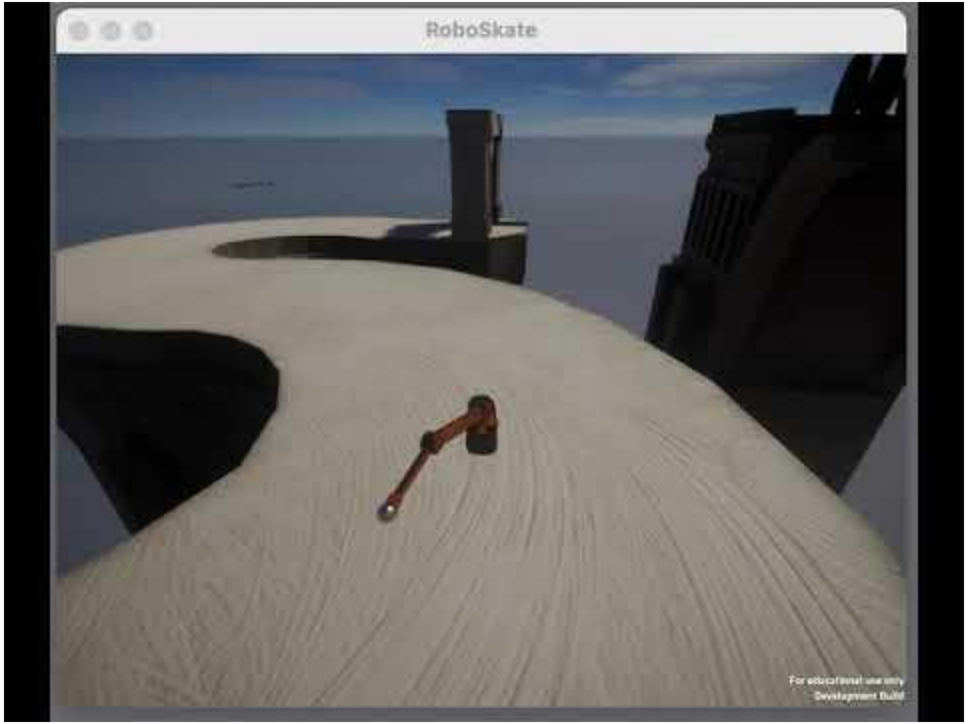
Multiple Input Observation



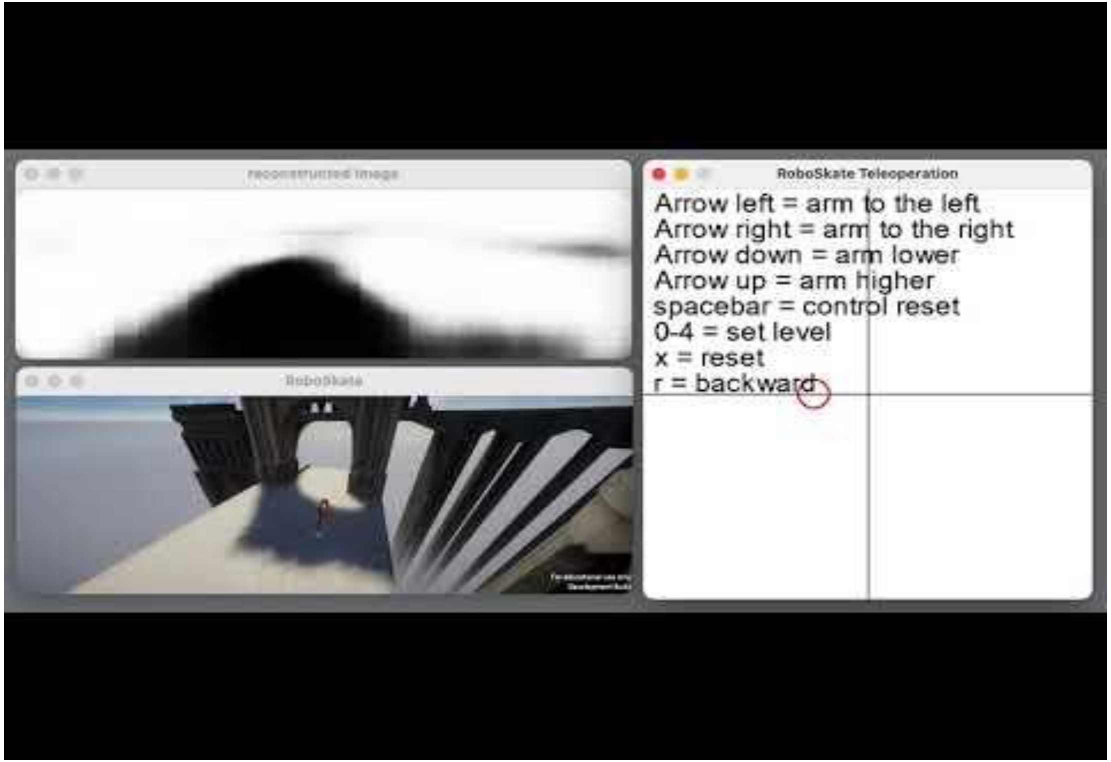
Agent without limitations learn to pull forward



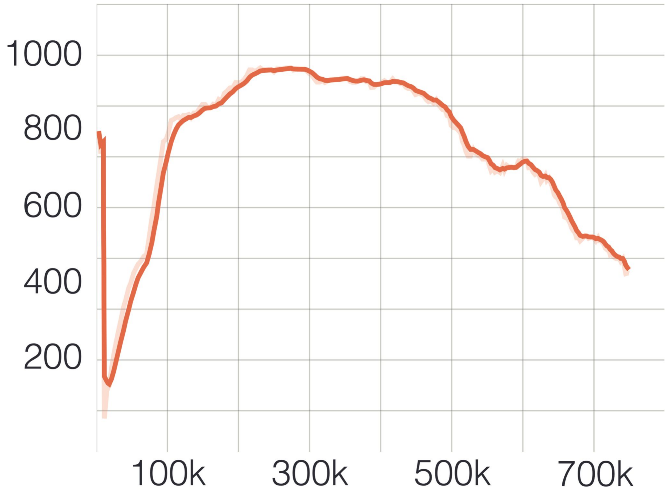
Model trained on level 2 finishes level 3

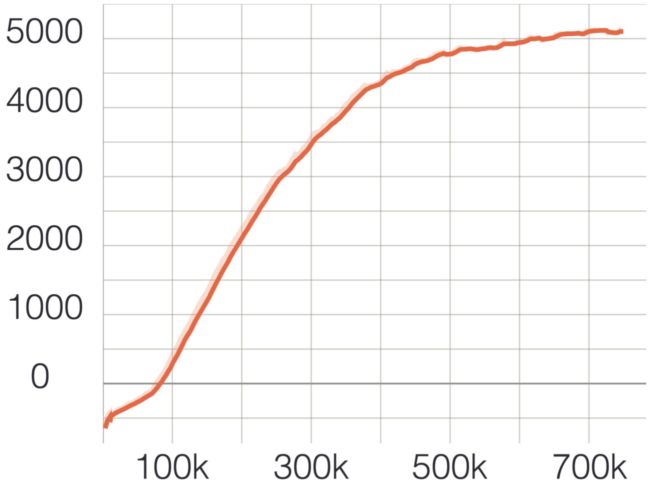


VAE - image segmentation - problems



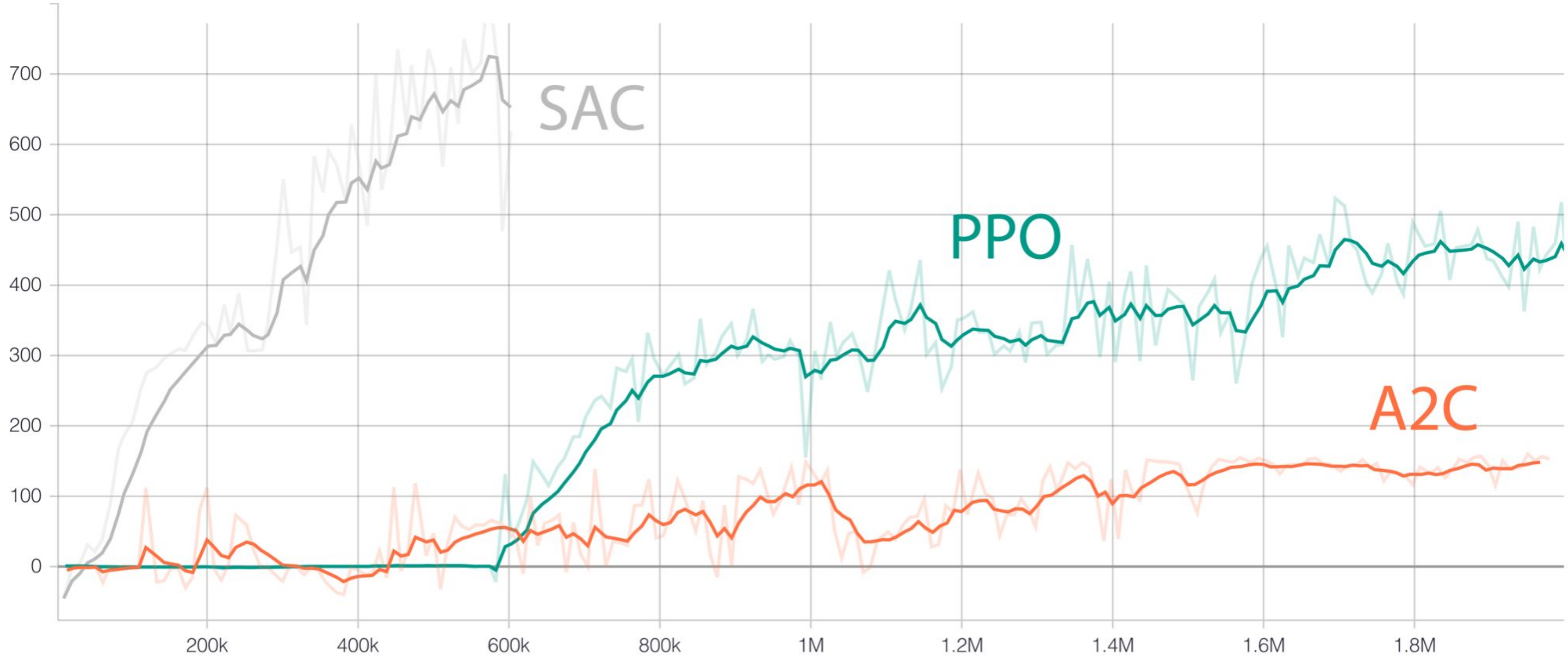
RoboSkate numerical agent level 1



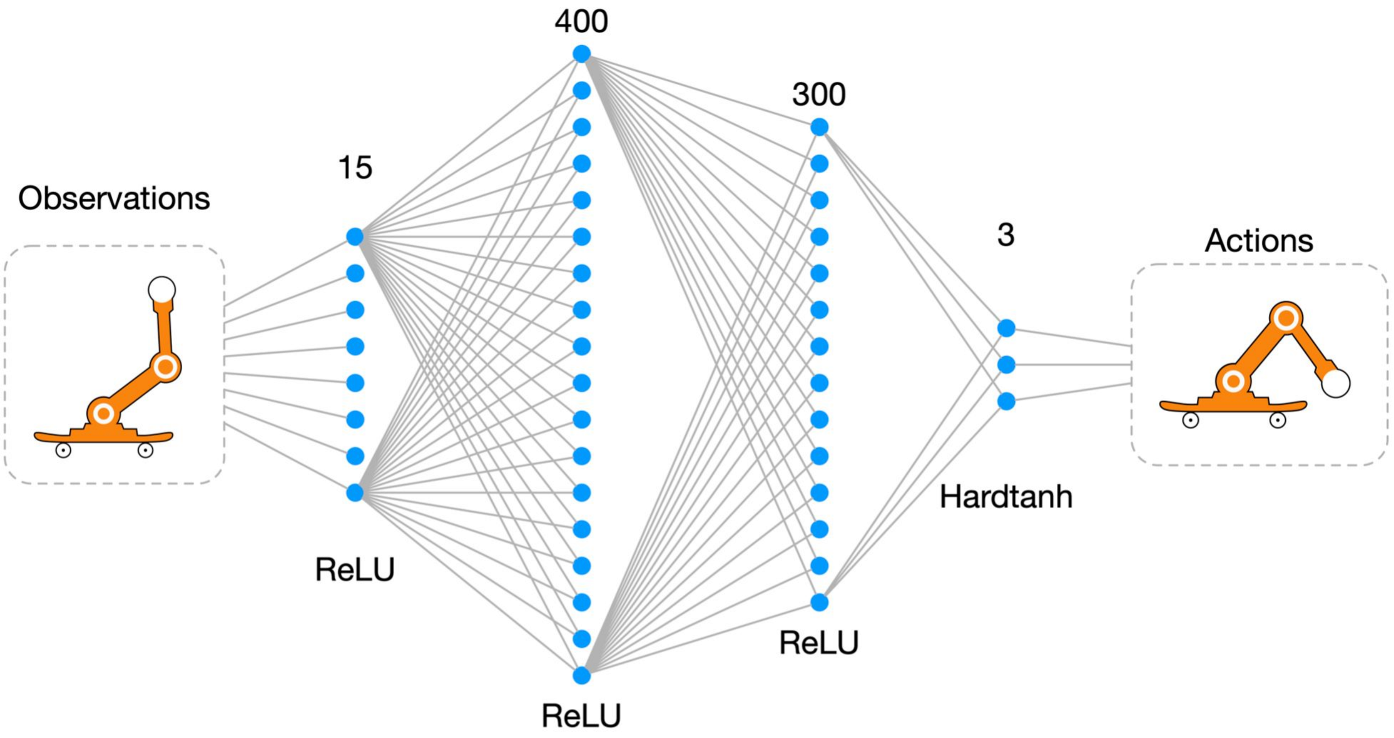


Learns first to finish the level and then optimise the movements.

Algorithms comparison (numerical agent level 1)

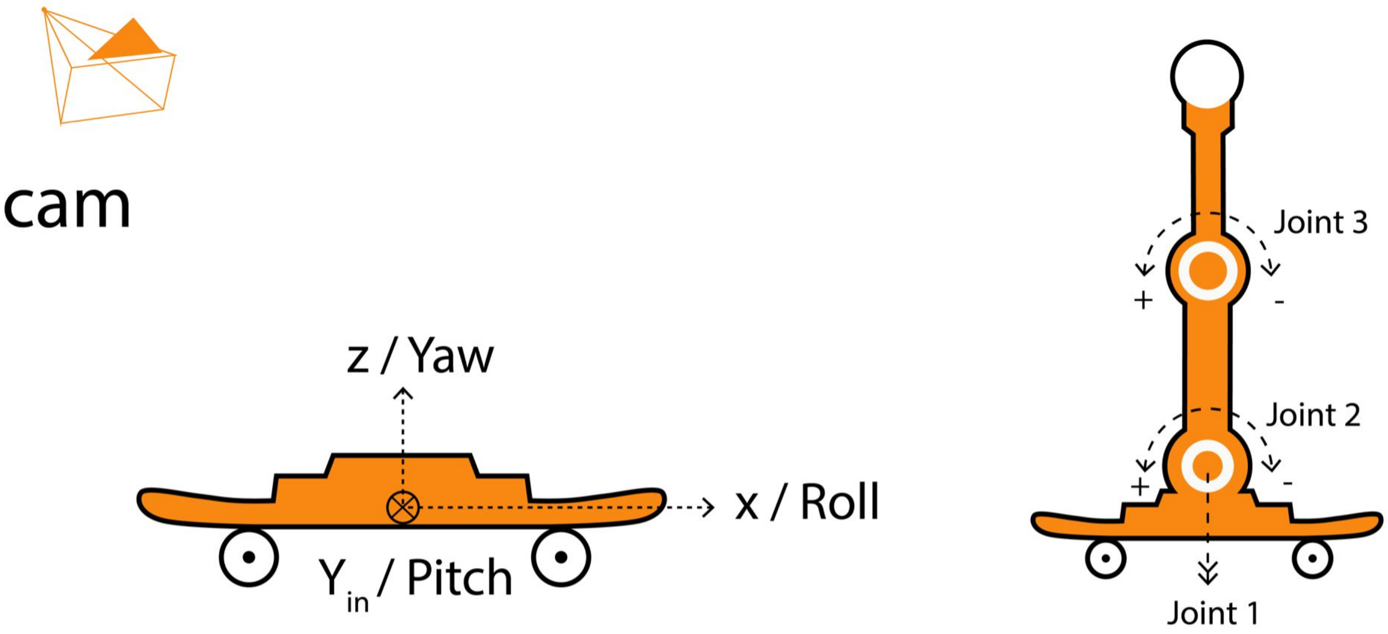


MLP Policy

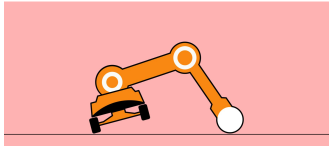


Teleoperation

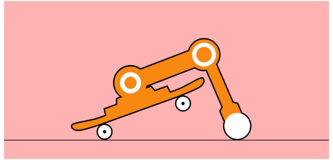
Agent



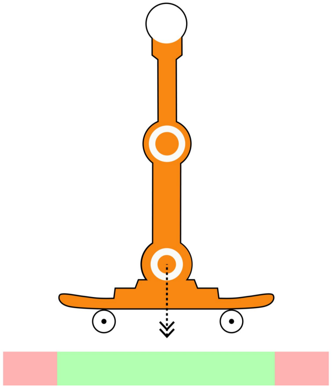
Termination conditions



Roll angle too high



Pitch angle too high

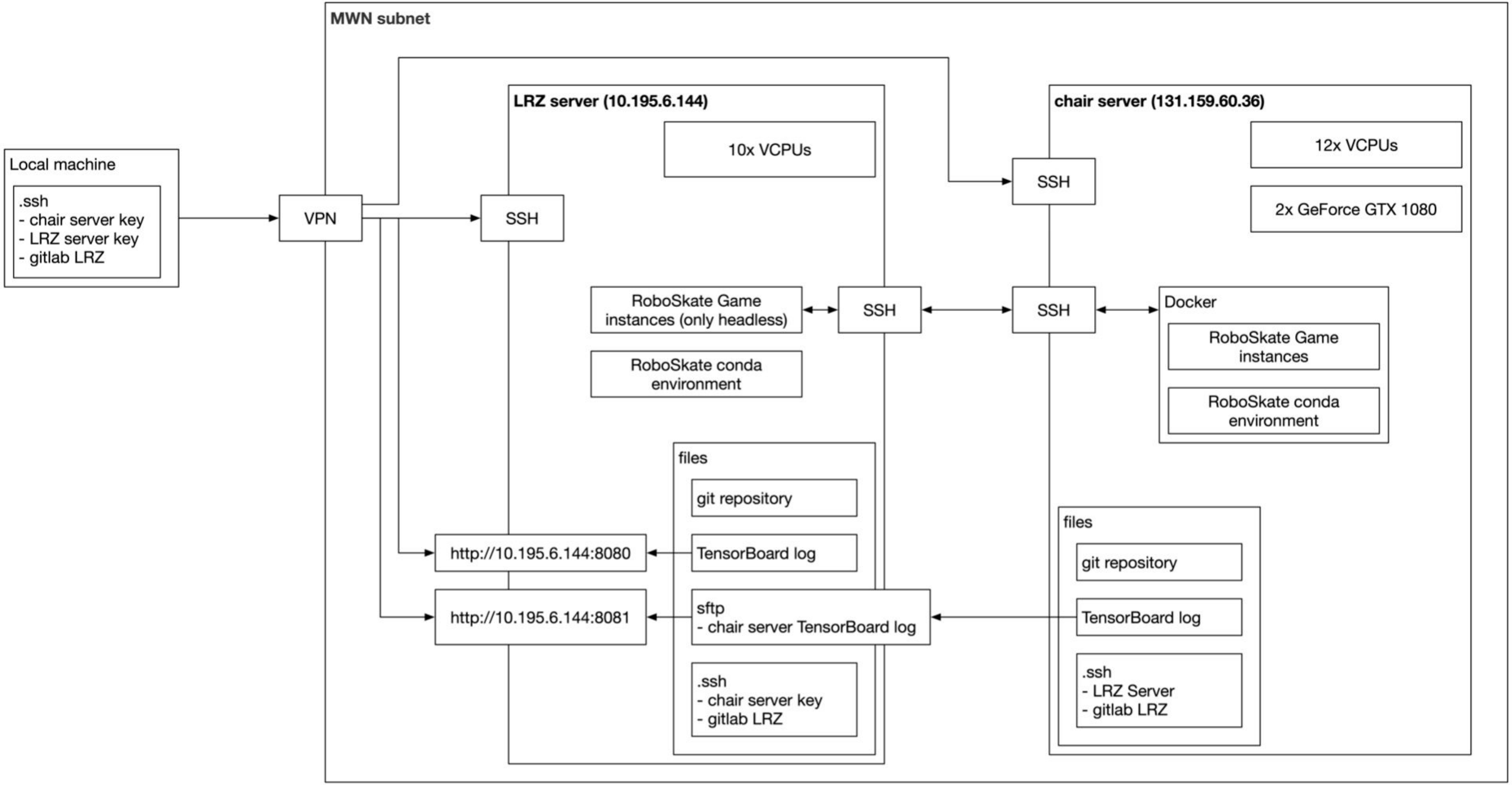


Joint velocity 1 too high

Imitation learning issues with HumanCompatibleAI/imitation

BC - behaviour cloning
GAIL
AIRL

- Not runnable on Windows
- Able to generate expert trajectories.
- At the moment, GAIL and BC don't interoperate well with SB3 in environments with image-based observation spaces
- The policy created by BC is not compatible with all SB3 algorithms
- Errors running AIRL with Cnn Policy


MWN subnet
Local machine
.ssh
- chair server key
- LRZ server key
- gitlab LRZ
VPN
LRZ server (10.195.6.144)
10x VCPUs
SSH
RoboSkate Game instances (only headless)
SSH
RoboSkate conda environment
files
git repository
TensorBoard log
sftp
- chair server TensorBoard log
.ssh
- chair server key
- gitlab LRZ
http://10.195.6.144:8080
http://10.195.6.144:8081
chair server (131.159.60.36)
12x VCPUs
2x GeForce GTX 1080
SSH
SSH
Docker
RoboSkate Game instances
RoboSkate conda environment
files
git repository
TensorBoard log
.ssh
- LRZ Server
- gitlab LRZ