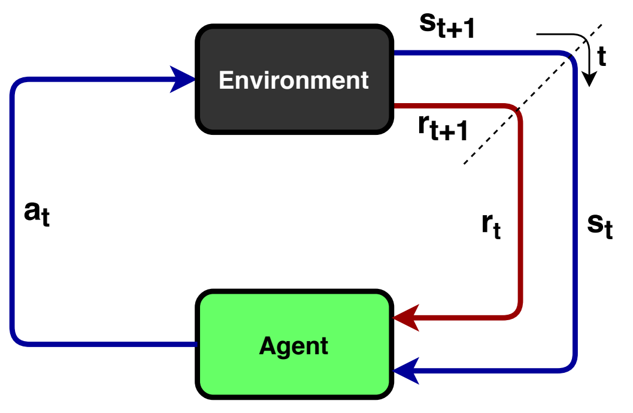
Combining primitive DQNs for improved reinforcement learning in Minecraft

Matthew Reynard*, Herman Kamper*, Benjamin Rosman[†], Herman A. Engelbrecht*

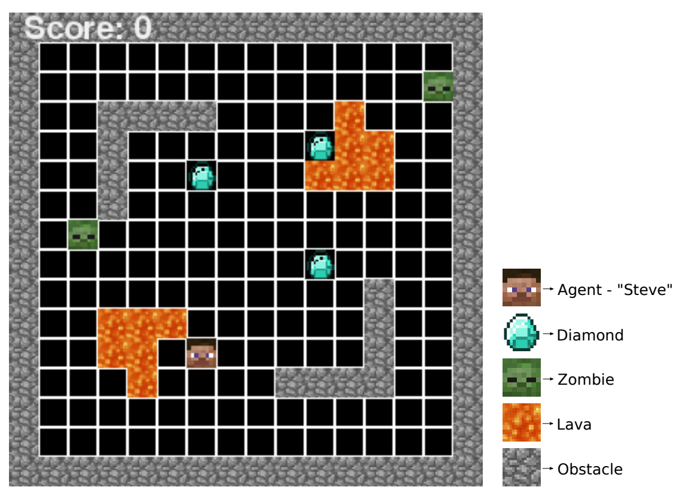
*Stellenbosch University, [†]University of the Witwatersrand

Background

- Minecraft is a popular 3D open world sandbox game, with a procedurally generated environment
- In Minecraft, mobs roam the environment at night and it's the players job to gather resources and survive
- Having an agent perform well in a challenging environment using reinforcement learning is a long standing goal for researchers
- For training optimization, a Python version of Minecraft was created
- The same environment which is trained using Python, is run using Project Malmo
- Project Malmo is a machine learning platform developed by Microsoft to test RL algorithms in Minecraft
- We use the method of Q-learning in our experiments, a value-based algorithm



RL iterative interaction cycle



Minecraft - Python, PyGame

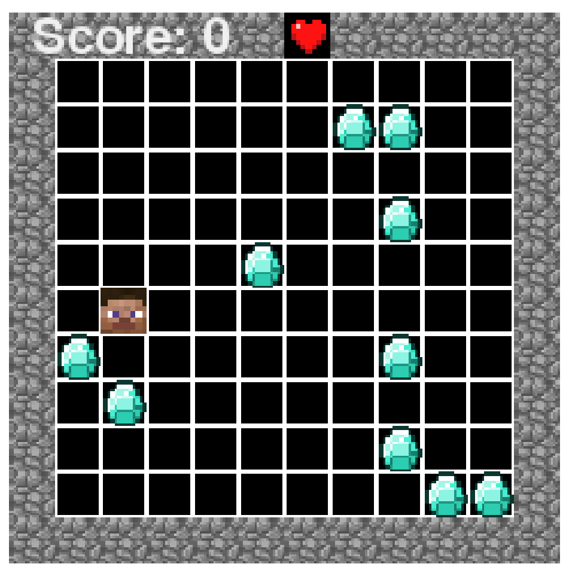
Minecraft - Project Malmo

Goals

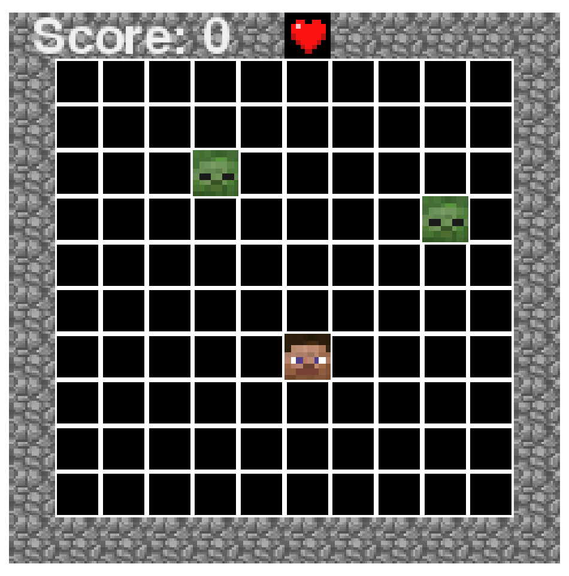
- **Goal of this paper:** To compare our new network architecture where an agent learns more complex actions in simpler environments to the current standard of RL
- **Overall Goal:** To have an agent survive the night in Minecraft using RL

Dojos

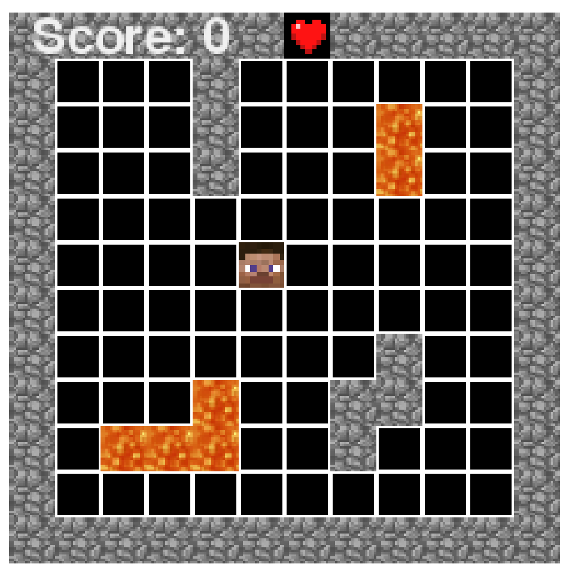
- The premise of these independent and isolated training environments, referred to as dojos, stems from humans learning in classrooms
- The idea is to have an agent learn a particular skill in each dojo
- A model is used to decide which dojo skill is necessary in the complex environment



Diamond Dojo



Zombie Dojo



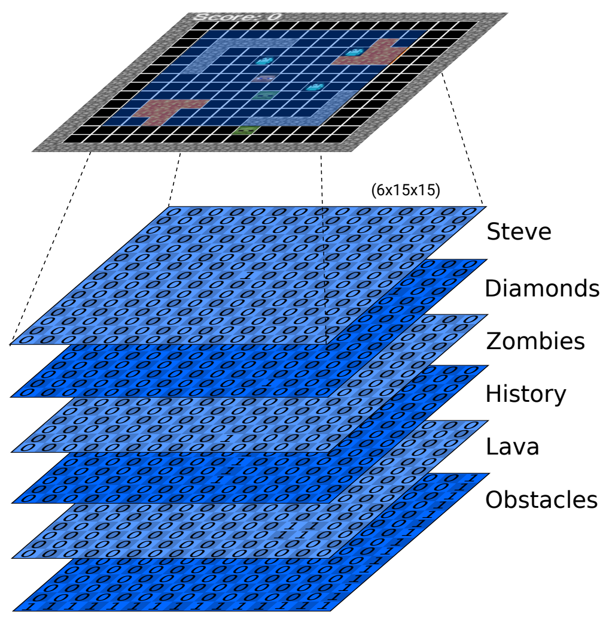
Explore Dojo

Approach

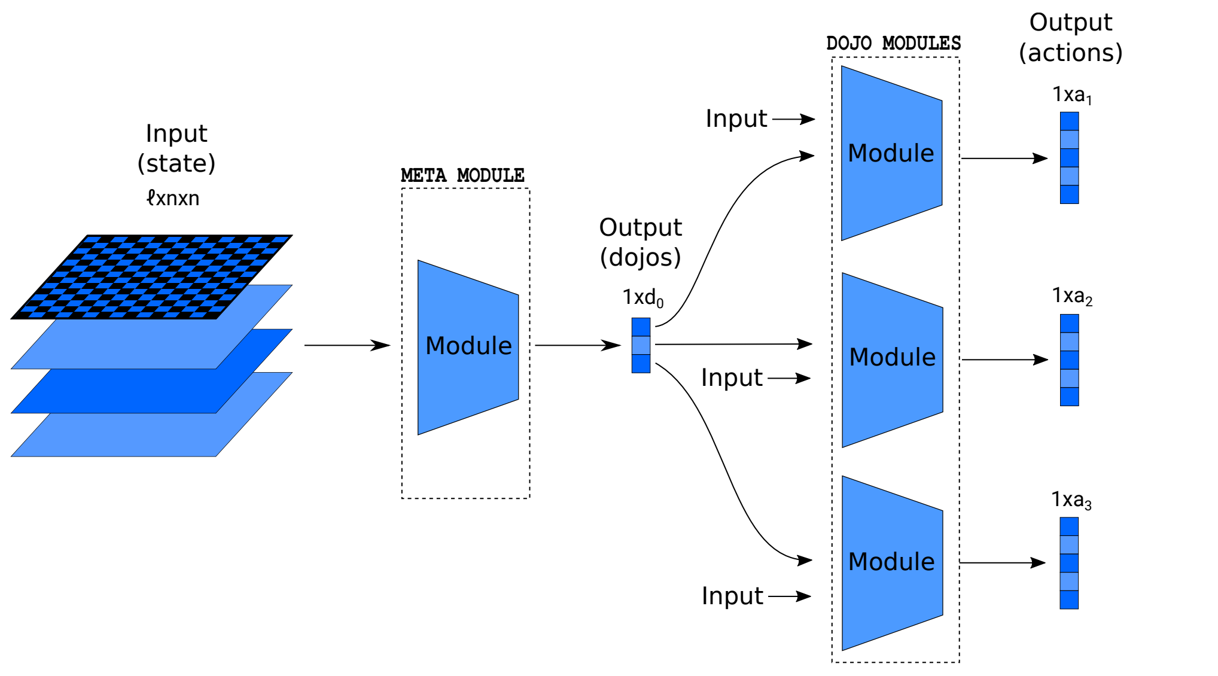
- We setup a large, complex environment in Minecraft
- Appropriate dojos were chosen for the agent to learn specific skills which are needed in the larger environment
- Each DOJO MODULE is trained separately and integrated into the larger model with the META MODULE trained last

Experiments

- The input to the model is manually feature extracted as opposed to raw pixel data
- The MODULES all had the same network architecture for simplicity
- Our approach and architecture is compared to a standard model using Q-learning with the same input



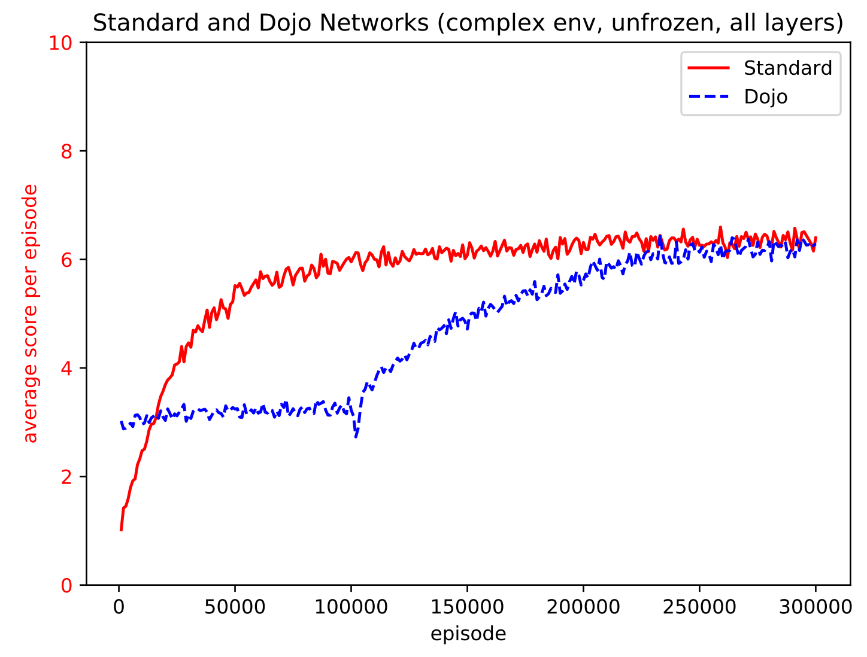
Input layers



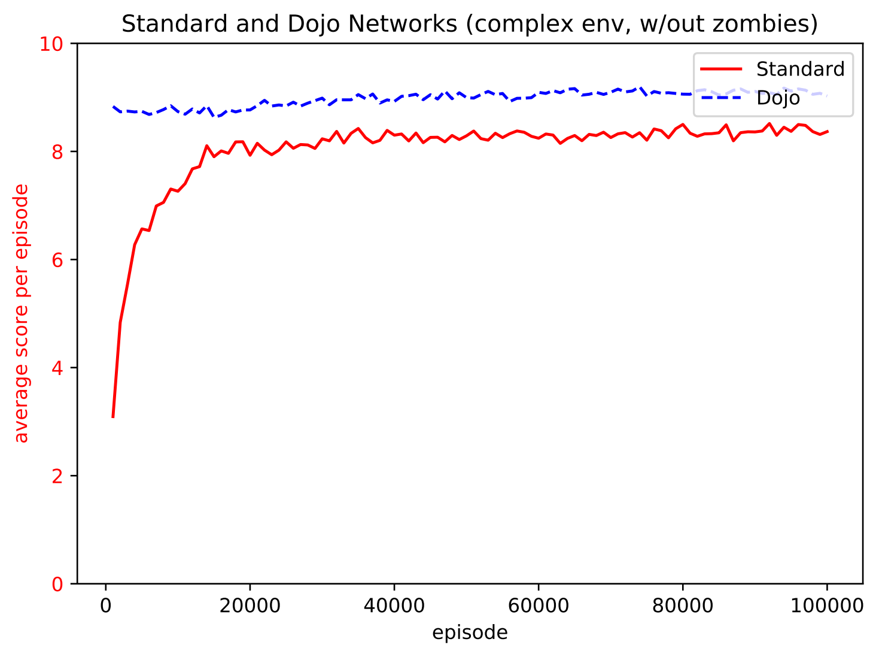
Overall network architecture

Results

- The STANDARD network (red) outperformed our approach (blue) until the dojos were allowed to further train
- Our DOJO network started well and ended on par with the STANDARD network (training graph on left)
- With no zombies in the environment, our DOJO network outperformed the STANDARD (training graph on right)



STANDARD and DOJO networks, unfrozen at 100k, $\epsilon = 0.1$, complex environment



No zombies, complex environment

Conclusion

Conclusion:

- Our DOJO network works well in certain environments and not in others
- The agent is being limited by the chosen dojo modules and when exposed to the complex larger environment, it performs in a sub-optimal manner

Future work:

- An additional DOJO MODULE for a new complex action
- Increase complexity in the network architecture
- Investigate which environments work for this type of model

Related work

- Options Framework, stems from SMDPs, which has a combination of primitive actions which have an extended duration
- Curriculum learning, learning one simple task and gradually increasing the complexity

SAUPEC/RobMech/PRASA, Cape Town, South Africa, 2020 • https://github.com/matthew-reynard/ • https://www.matthewreynard.com/