Nintendo Super Smash Bros. Melee: An "Untouchable" Agent

Deepak Dilipkumar (ddilipku), Yuan Liu (yuanl4), Ben Parr (bparr)

Deep Reinforcement Learning and Control, Carnegie Mellon University

Introduction

Objective: To create Super Smash Bros. Melee agents that avoid getting hit.

- **Melee**: A multiplayer video game for the Nintendo GameCube console where each character tries to knock the other off the stage.
- Melee is the longest-played game on ESPN's 2016 Top 10 Esports Draft.
- **Dolphin Emulator**: Software capable of playing Melee on modern computers instead of the original GameCube hardware.
- **Models Trained**: Deep Q-Network, Double Q-Network, and Dueling Deep Q-Network

Game Emulation on Google Cloud

- Each model was trained on a *full month* of gameplay data generated in *8 hours*.
- The central manager updates the global model as N=50 workers upload gameplay data.
- Each Google Cloud g1-small worker plays Melee.

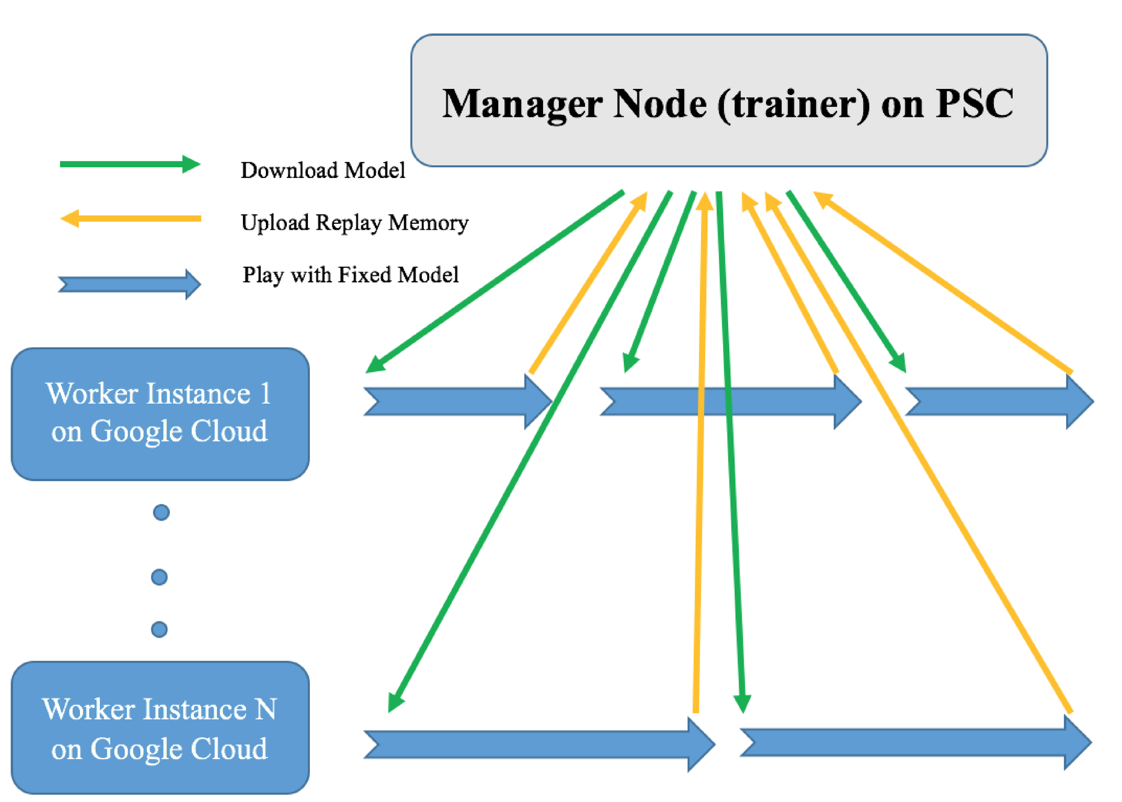


Figure 1: Infrastructure outline where a single Pittsburgh Supercomputing Center (PSC) job trains on gameplay generated on N Google Cloud workers

Melee Screenshot

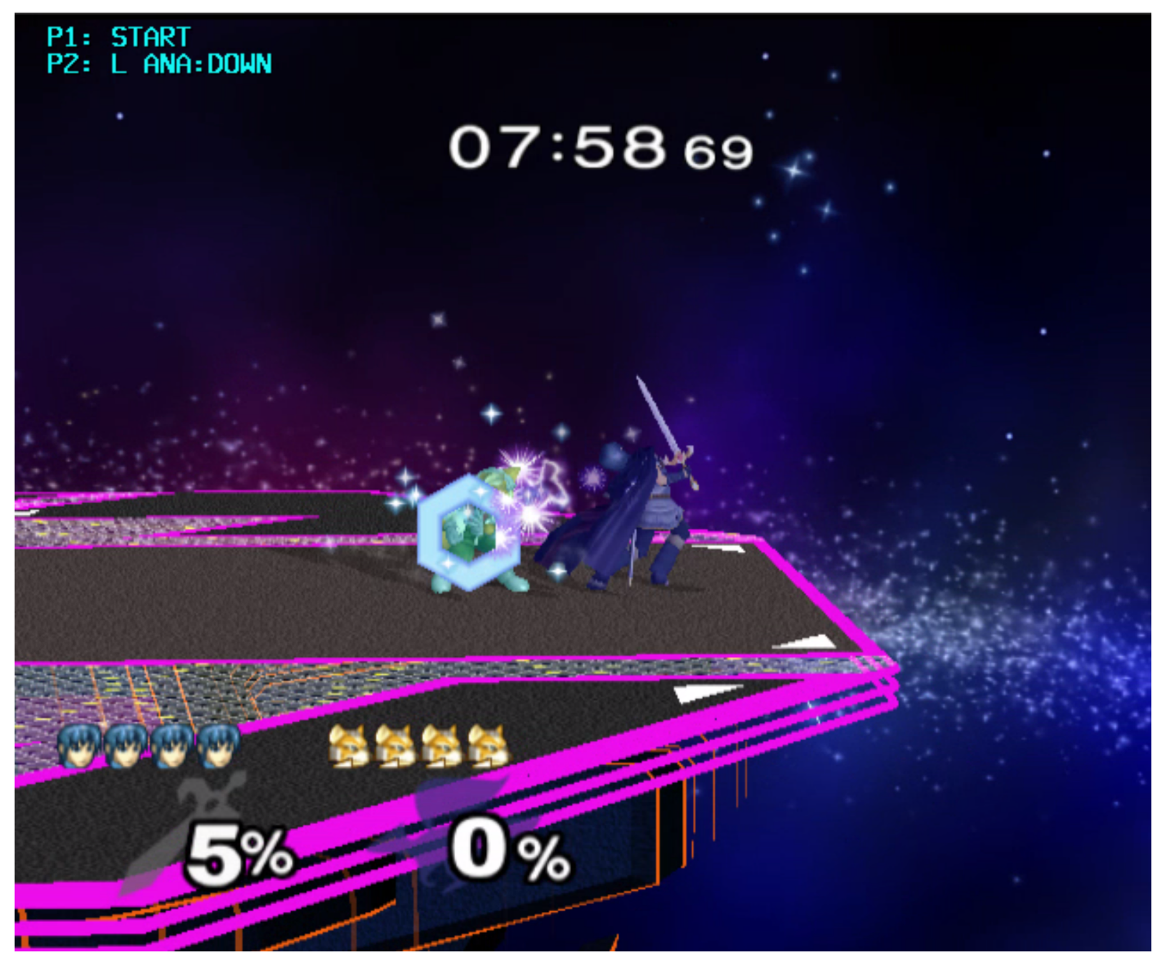


Figure 2: A screenshot of our agent (Fox) using a special attack on the sword-wielding opponent (Marth) on the entirely flat Final Destination stage

Environment Description

Melee includes 9 in-game AI's with increasing difficulty. We train against all 9, but evaluate only on the hardest difficulty. All our agents play the Fox character versus the Marth in-game AI on Final Destination. The Marth opponent and occasional frame-skipping make the environment stochastic.

- **State**: Internal memory values we selected, including character positions, speeds, damage percentages and action states.
- **Action**: Five actions: Do nothing, left dodge, right dodge, standing dodge, and Fox's shine.
- **Reward**: The agent gets a reward of 1/60 for every frame that it is not moving, and has zero damage. As soon as it gets hit it receives a terminal flag and the game is reset. This is in line with our aim of creating an "untouchable" AI.

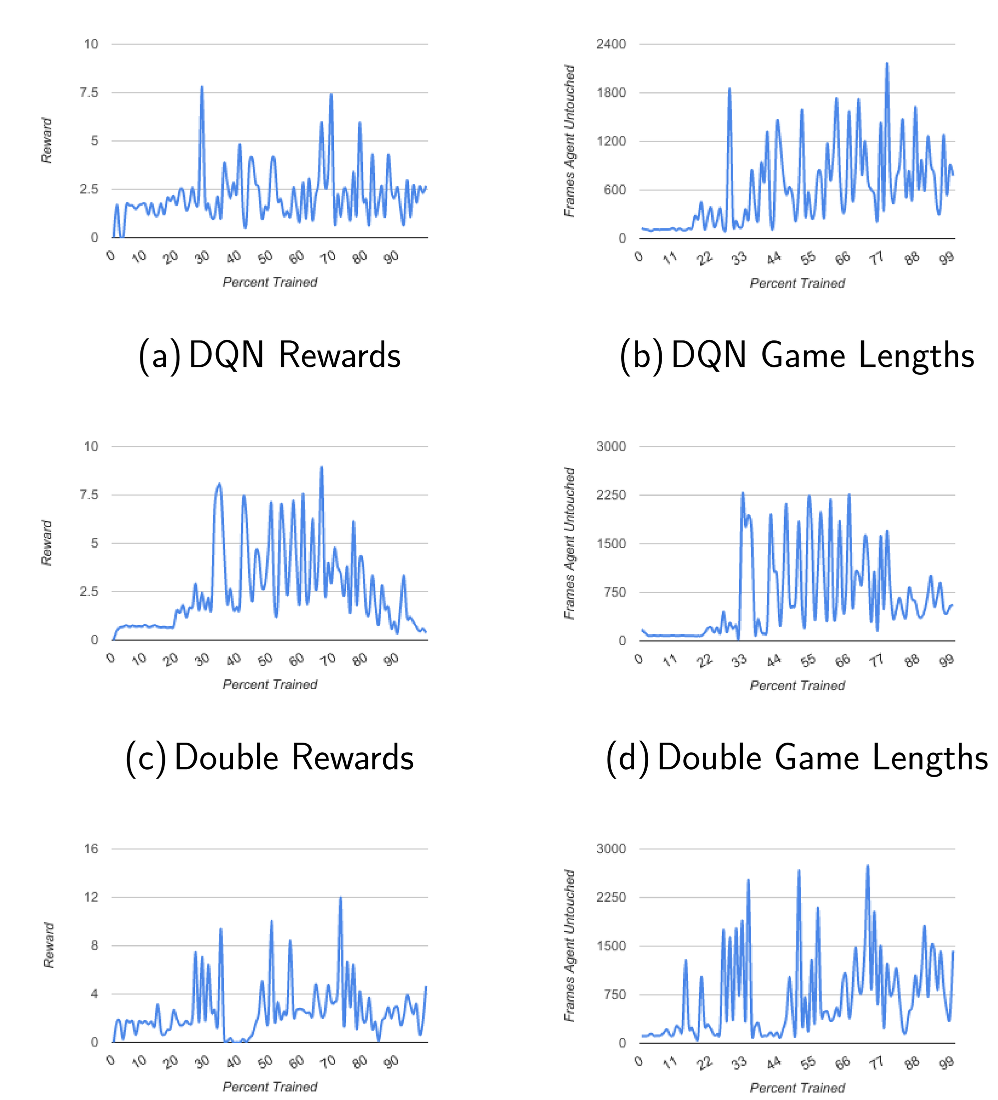
Methods

We implement Q-learning to train our agent. Q-learning can be seem as a pseudo stochastic gradient descent step on:

$$l(w) = E_{s,a,r,s'}\left(r + \gamma \max_{a' \in A} Q_{w^-}(s', a') - Q_w(s,a)\right)^2$$

- Each Google Cloud instance uses a fixed target model Q_{w^-}, within one match, to generate replay memory tuples (s, a, r, s'). At the end of each match, it uploads the replay memory to the Manager on PSC and downloads the newest Q_{w^-}.
- We train the on-line model Q_w using target update frequency 10,000 on the Manager.
- We train our agent through three different deep Q-learning networks: Deep Q-Network, Double Q-Network, and Dueling Deep Q-Network. Finally, we compare the performance of these three networks.

Results



(a) DQN Rewards
(b) DQN Game Lengths
(c) Double Rewards
(d) Double Game Lengths
(e) Dueling Rewards
(f) Dueling Game Lengths

Figure 3: The Dueling DQN had the highest peaks and the Double DQN had the most consistently good results

Figure 4: Mean Max Q over training

Discussion

- **ReLU on the Final Layer**: This worked well on Atari, but it seems to completely block learning in this setting, where initially zero Q-values never change over training.
- **Overfitting the Opponent**: The agent mainly trained on the hardest in-game AI, and so the agent "overfits" to the in-game AI's strategy. With more resources, we could instead train directly against humans.
- **Erratic Q-values**: For dueling, the Q-values appear to be very large, and are quite erratic as compared to the other two architectures.

Conclusion

After 8 hours of parallelized training using the Dueling Network, our best agent is able to avoid getting hit by the highest level AI in the game for a full minute, roughly 74.6% of the time.

Our Code: https://github.com/bparr/melee-ai.

Finally, thank you to Vlad Firoiu, without whom this project would not have been possible.