

# Training Mario Experiments Journal

## Playing Atari with Deep Reinforcement Learning

**Key contributions:** Introduced deep Q-networks (DQN) that combine reinforcement learning with deep neural networks, enabling agents to learn directly from high-dimensional sensory input and achieve human-level performance on Atari games using experience replay and target networks.

### Additional tricks

**Training instability:** I've taken two additional steps to stabilize training: \* L1+MSE loss function \* Gradient clipping

**Previous action as input:** The agent quickly learns that run and jump buttons together provide a lot of reward, but Mario only jumps if you've not pressed that button before. This in turns makes Mario continuously run into the mushroom enemies. To distibguish the states where pressing jump will result in a jump vs. continue because we have already pressed I've added the previous action as input. It does improve this behavior slightly.

**Softmax action selection:** Choosing highest Q-value action at every step results in Mario getting stuck in parts where the agent is pressing against an object. To perform more randomized actions I've implemented softmax action selection, which gretly improves agent performance.

### Summary

Feature	Paper	My Mario
Target Network	✓	✓
Experience Replay	✓	✓
Loss Function	MSE	MSE/L1
Gradient Clipping	×	✓
Action Repetition (Frame Skip)	✓	✓
Input Preprocessing	✓	✓
$\epsilon$ -greedy Action Selection	✓	✓
Softmax Action Selection	×	✓
Network Architecture	✓	✓
Reward Shaping	✓	✓
Reward Clipping	✓	×
Prioritized Replay	×	✓
Checkpointing	×	✓
TensorBoard Logging	×	✓
Previous Action as Input	×	✓

**Comparison with random Mario?**

TODO