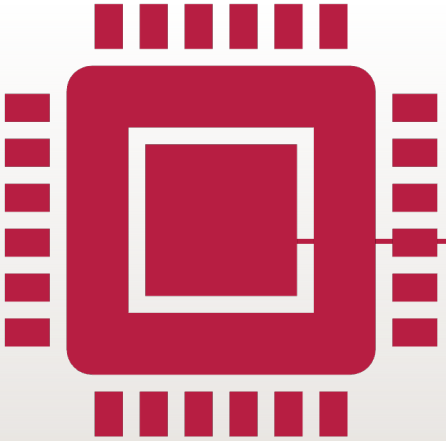


REINFORCEMENT LEARNING WITH DQN & DOUBLE DQN ON ATARI

MARIOBROS-V5 & PONG-V5

BY: FIYORI DEMEWEZ & TIMOTHEE TRAN

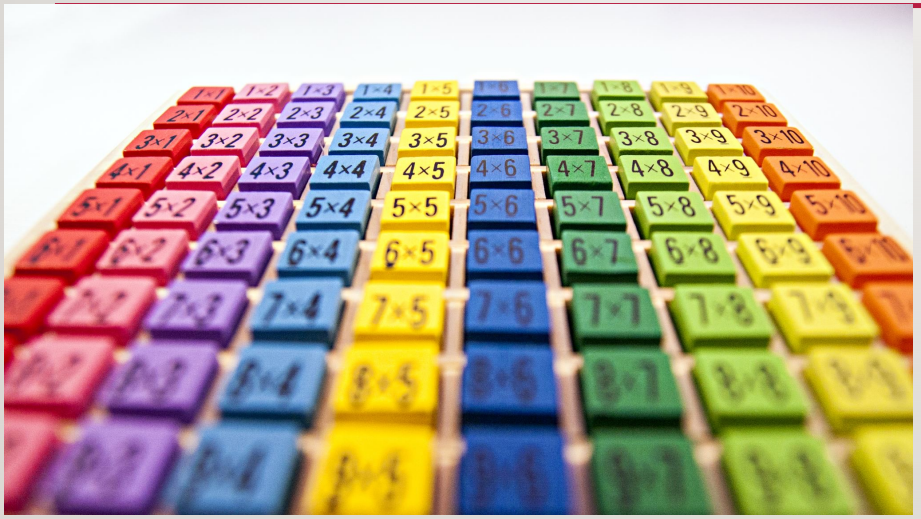
PROJECT INTRODUCTION

Implemented **Baseline DQN** and **Double DQN (DDQN)**	Evaluated on **two Atari environments**:	ALE/MarioBros-v5	ALE/Pong-v5
Goals:	Compare performance	Analyze stability and learning quality	Understand behavior differences between DQN variants

PROJECT COMPONENTS

- Baseline DQN for both games
- Double DQN with extended training:
- Mario: **205k & 305k frames**
- Pong: **100k & 50k frames**
- Comparison:
- Learning curves
- Behavior (gameplay videos)
- Quantitative metrics

ENVIRONMENT: MARIOBROS-V5



- Observation: 84×84 grayscale (4-frame stack)
- Action space: Atari joystick
- Reward structure: **sparse**
  - survival time
  - enemy stomps
  - platform progress
- Key difficulty: delayed rewards + precise timing

ENVIRONMENT: PONG-V5

- Observation: 84×84 grayscale, 4-frame stack
- Action space: up / down / no-op
- Reward structure: +1 / −1 / 0
- Difficulty: long-term paddle strategy, delayed scoring

BASELINE DQN

CNN → Fully connected → Actions

Experience replay

Target network

Epsilon-greedy exploration

Adam optimizer

Issue: overestimates Q-values → poor stability

DOUBLE DQN (DDQN)

Online network selects action

Target network evaluates action

Fixes overestimation problem

Leads to **more stable, accurate learning**

MARIO HYPERPARAMETERS

- Learning rate: **1e-4**
- Replay buffer: 100k
- Batch size: 32
- Gamma: 0.99
- Target sync:
  - DQN: 10k
  - DDQN: 5k

PONG HYPERPARAMETERS

Learning rate: **1e-4** (also tested 0.01)

Replay: 50k

Batch size: 32

Target sync: 1,000 frames

Epsilon decay:

100k frames

50k frames (experiment)

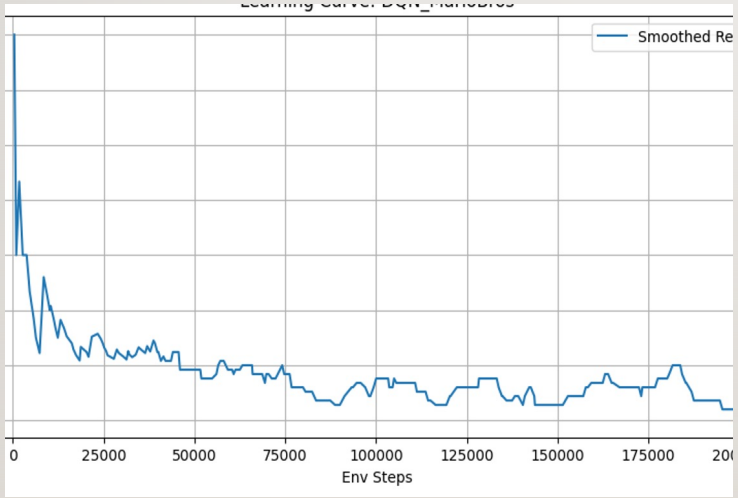
	Mario-v5	Pong-v5
Learning rate:	adam	1e-4
Optimzer:	Gamma 0.99	adam
Replay size:	100k	50,000
Batch size:	32	32
Target sync:	10k(DQN), 5k(DDQN)	Every 1,000 frames
Frame Stack:	4	4
Epsilon Schedule	1.0->0.1 over-training	Linear decay from 1.0 to 0.01 over 100,000 frames

EXPERIMENT LOG

- Increased warmup steps (10k → 50k)
- Reduced DDQN target sync
- Extended DDQN Mario training (205k → 305k)
- Adjusted epsilon decay rate
- Changed replay buffer sizes
- Pong: faster epsilon decay + learning rate experiments

Change number	What we try	Why
1	Increased warmup steps 10k-50k	To stabilize DQN replay Samples early on
2	Reduced target sync for DDQN	Better stability for a large training window
3	Extended DDQN run from 205k-305k	To check whether performance recovers from local minima
4	Changed epsilon decay speed	To balance exploration vs. exploitation
5	Adjusted replay buffer size	To reduce correlation in samples

MARIO DQN LEARNING CURVE

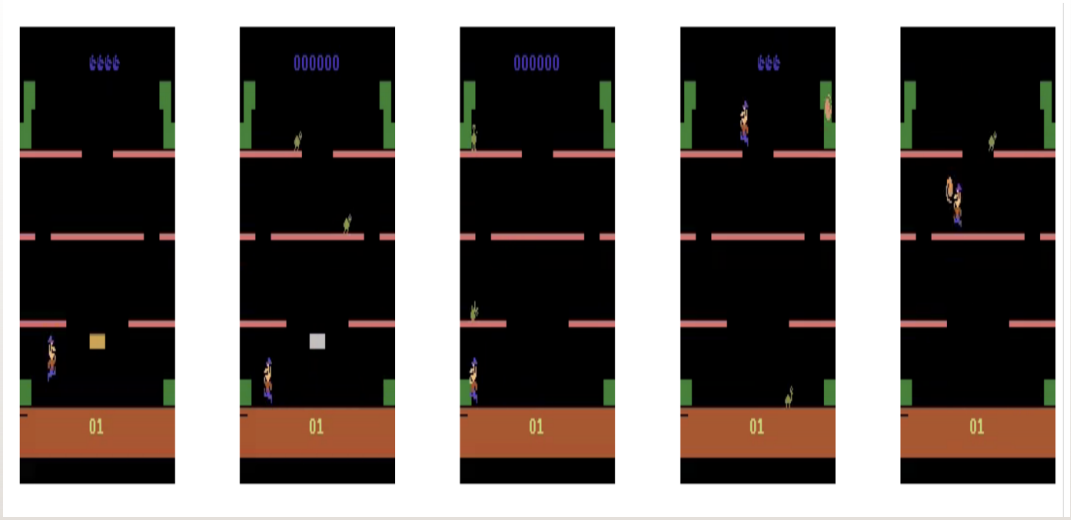


- Early reward spike from random actions
- Performance collapses mid-training
- Returns settle at **250–350**
- Highly unstable

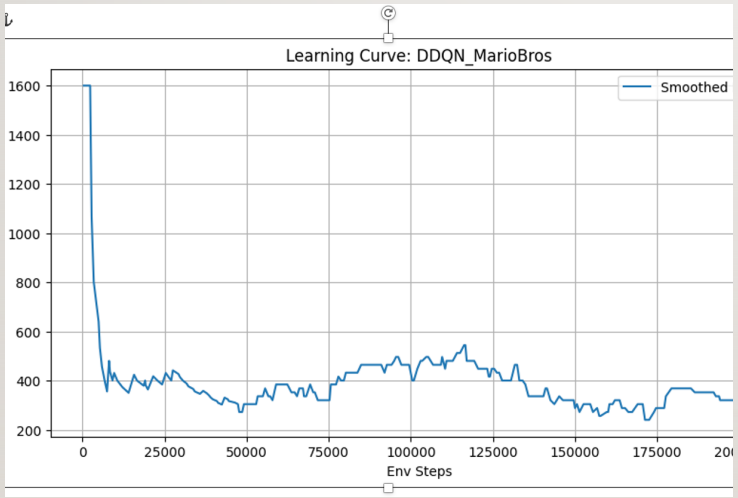
MARIO DQN GAMEPLAY BEHAVIOR

- **early episode:**
- Hesitant movement
- Poor enemy avoidance
- Backward steps
- **Later episode:**
- Slightly smoother
- Still inconsistent
- Weak survival

DQN Early vs. Learned



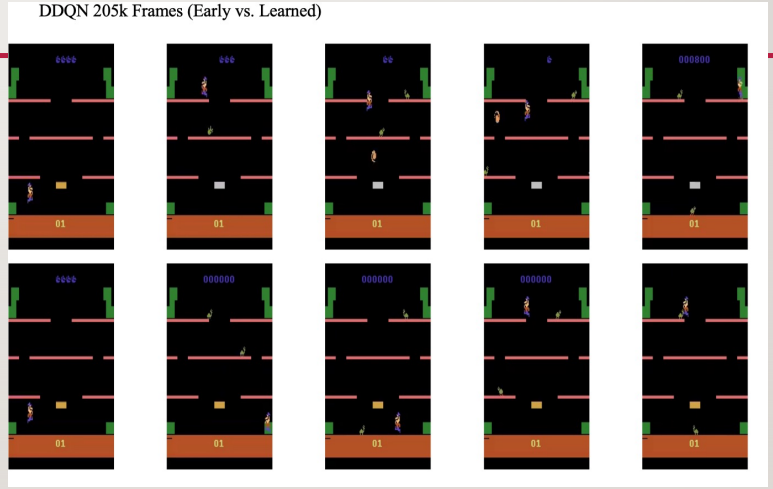
MARIO DDQN (205K FRAMES)



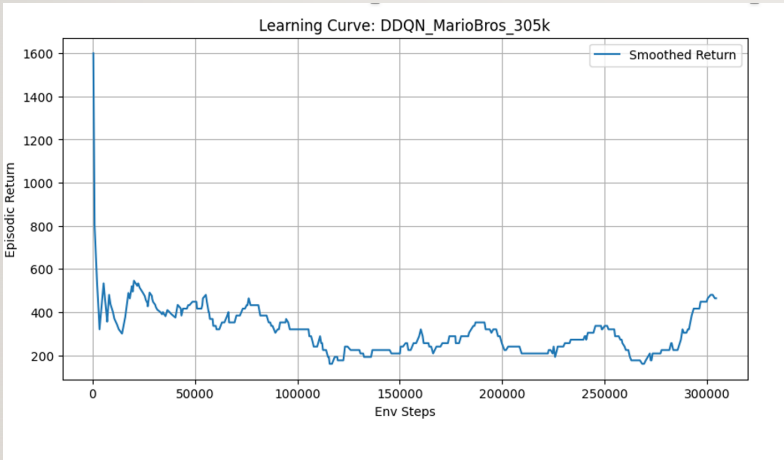
- More stable than DQN
- Mid-training oscillations
- Moderate improvement
- Not fully stable

MARIO DDQN 205K GAMEPLAY

- **Early episode:**
- Still hesitates
- Mistimed jumps
- **Learned episode:**
- Smoother movement
- Lasts longer than DQN
-



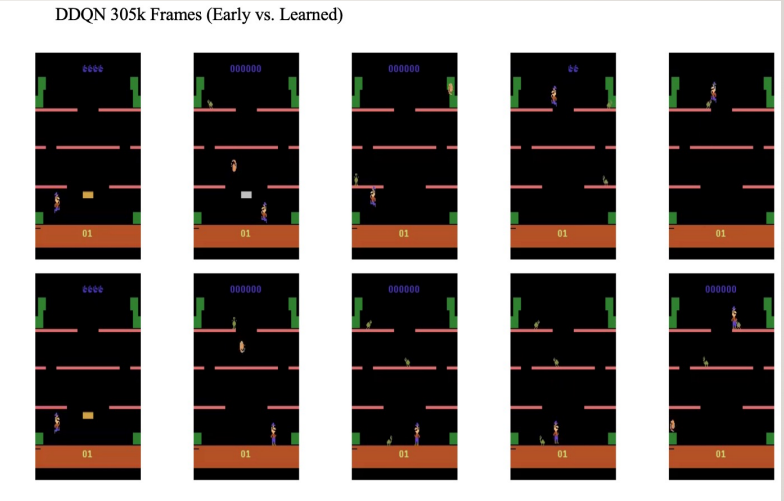
MARIO DDQN (305K FRAMES) — BEST MODEL



- Recovers from local minimum
- Achieves **400–450 average return**
- Most stable learning curve
- Smooth movement + confident jumps
- Longest survival

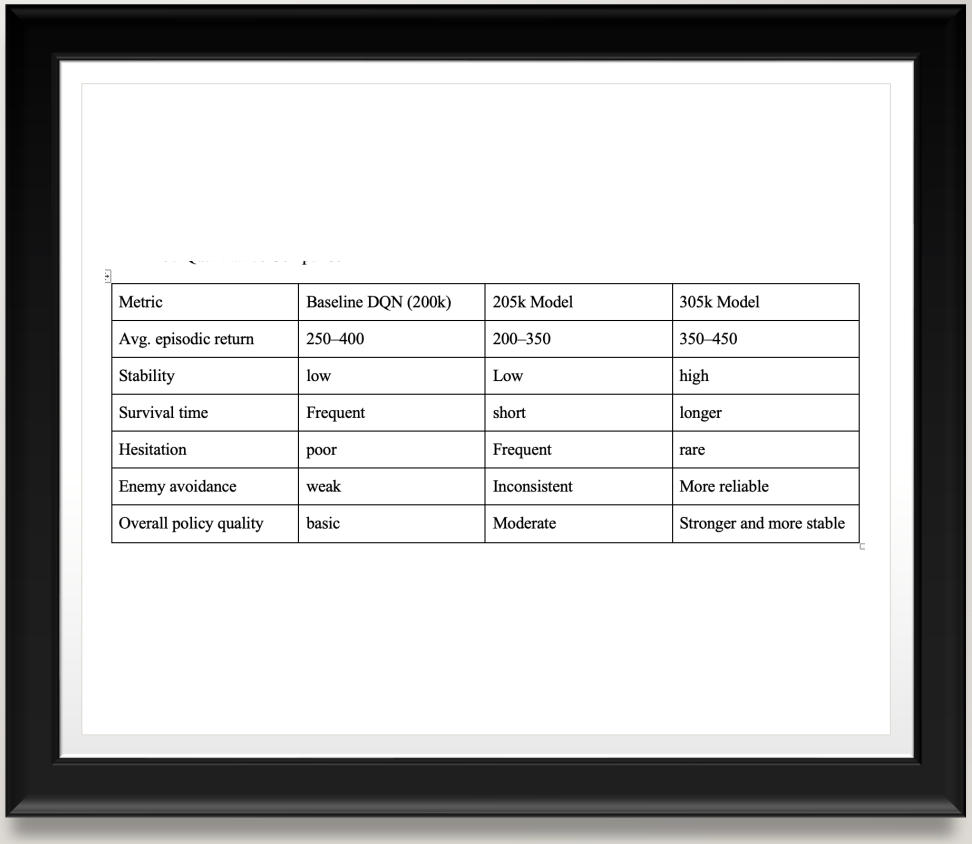
MARIO DDQN 305K GAMEPLAY

- Most stable agent
- Almost no hesitation
- Accurate platform movement
- Significantly improved survival



MARIO QUANTITATIVE TABLE

- **Metric | DQN (200k) | DDQN (205k) | DDQN (305k)**
  Avg Return | 250–400 | 200–350 | **350–450**
  Stability | Low | Low–Medium | **High**
  Survival | Short | Medium | **Longest**
  Hesitation | High | Medium | **Low**
  Enemy Avoidance | Weak | Inconsistent | **Reliable**
  Policy Quality | Basic | Moderate | **Strongest**



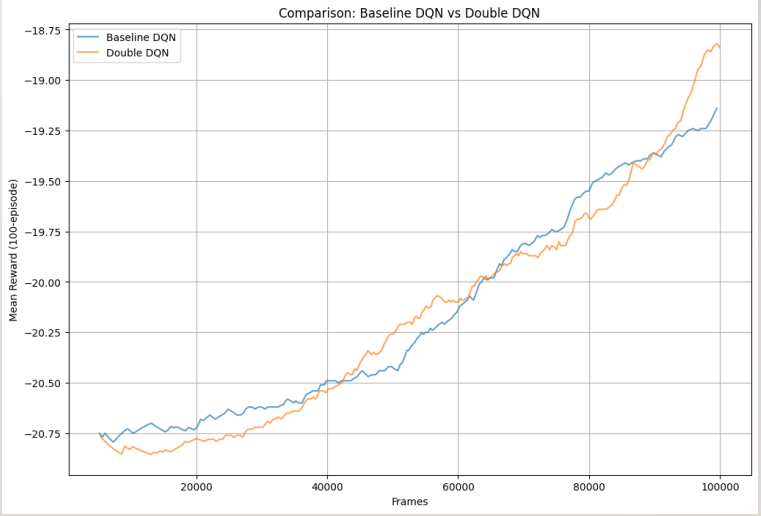
Metric	Baseline DQN (200k)	205k Model	305k Model
Avg. episodic return	250–400	200–350	350–450
Stability	low	Low	high
Survival time	Frequent	short	longer
Hesitation	poor	Frequent	rare
Enemy avoidance	weak	Inconsistent	More reliable
Overall policy quality	basic	Moderate	Stronger and more stable

PONG 100K LEARNING CURVE

DDQN begins outperforming DQN after ~90k frames

Strong long-term performance trend

DQN improves but plateaus lower



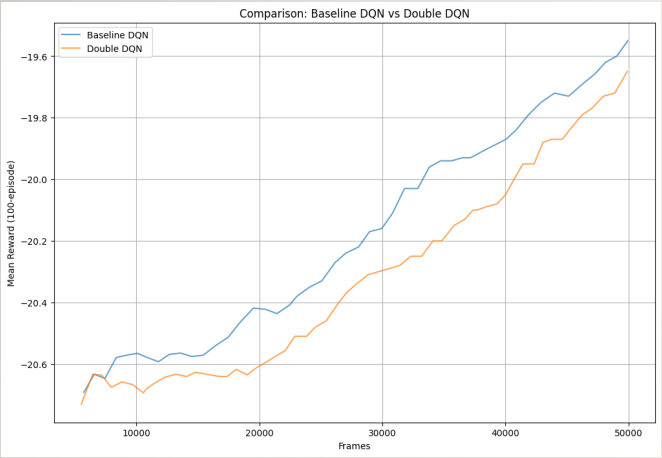
PONG 50K FRAME EXPERIMENT

Faster epsilon decay → 30 min training (vs 175 min)

Slight drop in final score

DDQN still trending upward

Good exploration/exploitation trade-off



PONG LEARNING RATE EXPERIMENT

- **LR = 1e-4, 50k Frame**
- Stable
- Score is steadily in an uptrend
- Getting better over time

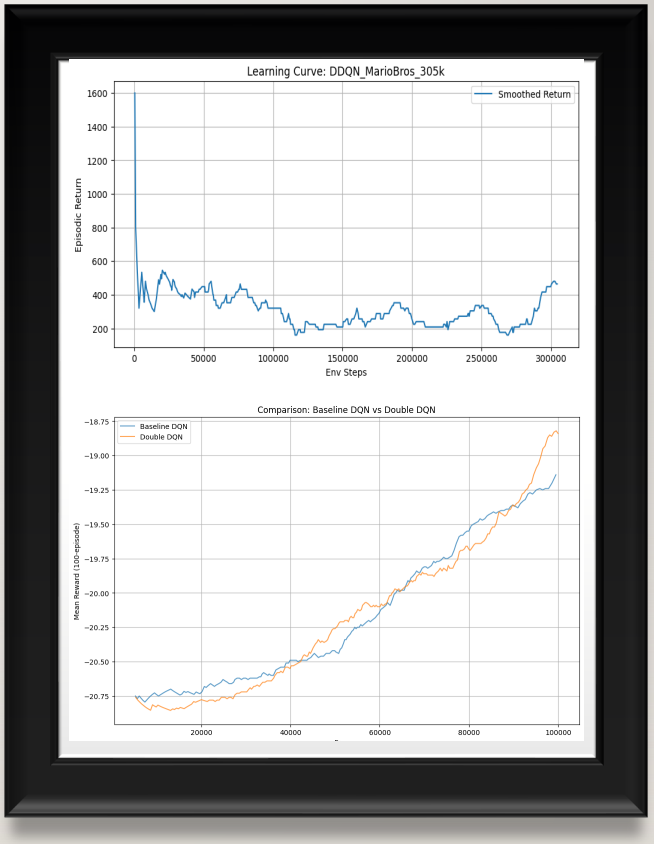
- **LR = 0.01, 50k Frame**
- Unstable
- Score is on a downtrend
- Overshooting → failure to learn

PONG SUMMARY

- Faster epsilon decay significantly reduces training time
- DDQN continues to show long-term advantage
- Learning rate is critical
- Pong is easier → more noticeable algorithm differences

FINAL COMPARATIVE INSIGHTS

- **Mario:**
- Hard environment
- Sparse rewards
- DDQN helps, but improvement limited
- **Pong:**
- Easier and more structured
- Clearly shows DDQN benefits
- Stronger stability + earlier improvement



REFLECTION

MARIO IS VERY CHALLENGING FOR BASIC DQN/DDQN

DDQN REDUCES Q-VALUE OVERESTIMATION

LONGER TRAINING (1–3 MILLION FRAMES) NEEDED FOR STRONG MARIO AGENT

PONG DEMONSTRATES DDQN'S THEORETICAL ADVANTAGES

HYPERPARAMETER TUNING (EPSILON SCHEDULE, REPLAY, LR) GREATLY AFFECTS RESULTS

FUTURE WORK


N-step returns


Dueling networks


Prioritized replay


Sticky actions

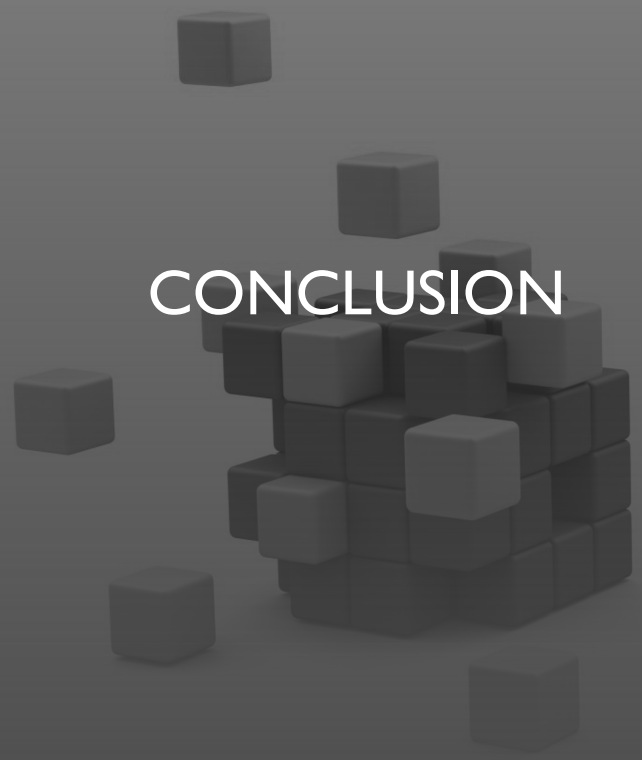

Slower epsilon schedules


GPU training for speed


Extended training duration

LINK

HTTPS://GITHUB.COM/TIMMYT110/DEEP-Q-LEARNING-ON-ATARI-FINAL-PROJECT



CONCLUSION

- **Double DQN consistently outperforms DQN** across both games
- Biggest improvements seen in **Pong**
- Mario requires deeper architectures and more frames
- Clear relationship between environment difficulty & RL performance
- Strong evidence that DDQN reduces overestimation and improves stability