SUPER MARIO BROS

INTELLIGENT AGENT

TRAINED BY REINFORCEMENT LEARNING

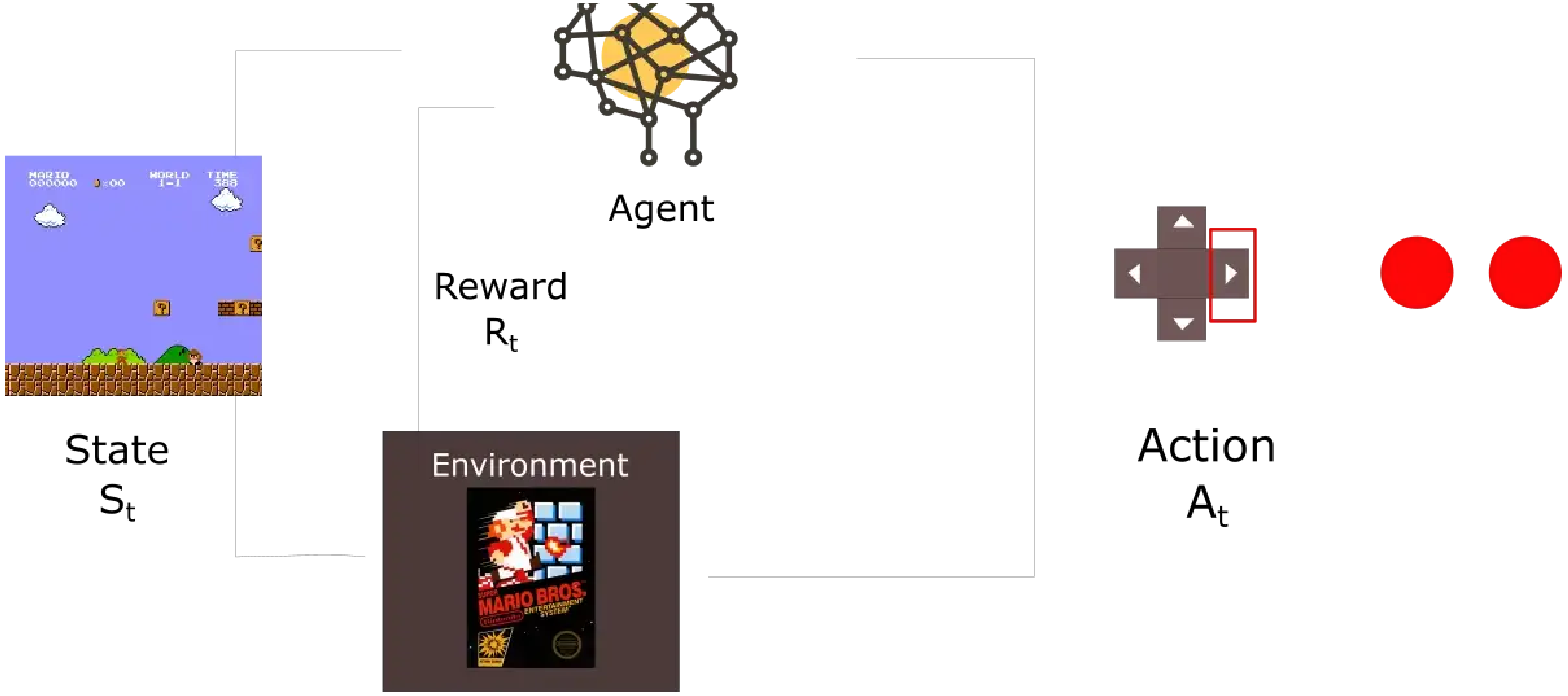
NOURIN AHMED
KALYAN VENKATESH POLUDASU
SIDDHARTHA PITCHIKA

PROBLEM STATEMENT

- Reinforcement Algorithms have been used to solve Atari games.
- There are not much work done on solving NES(Nintendo Platform games) games with RL.
- Our approach is to implement different RL algorithms to train an intelligent Super Mario Agent.

WHAT IS REINFORCEMENT LEARNING?

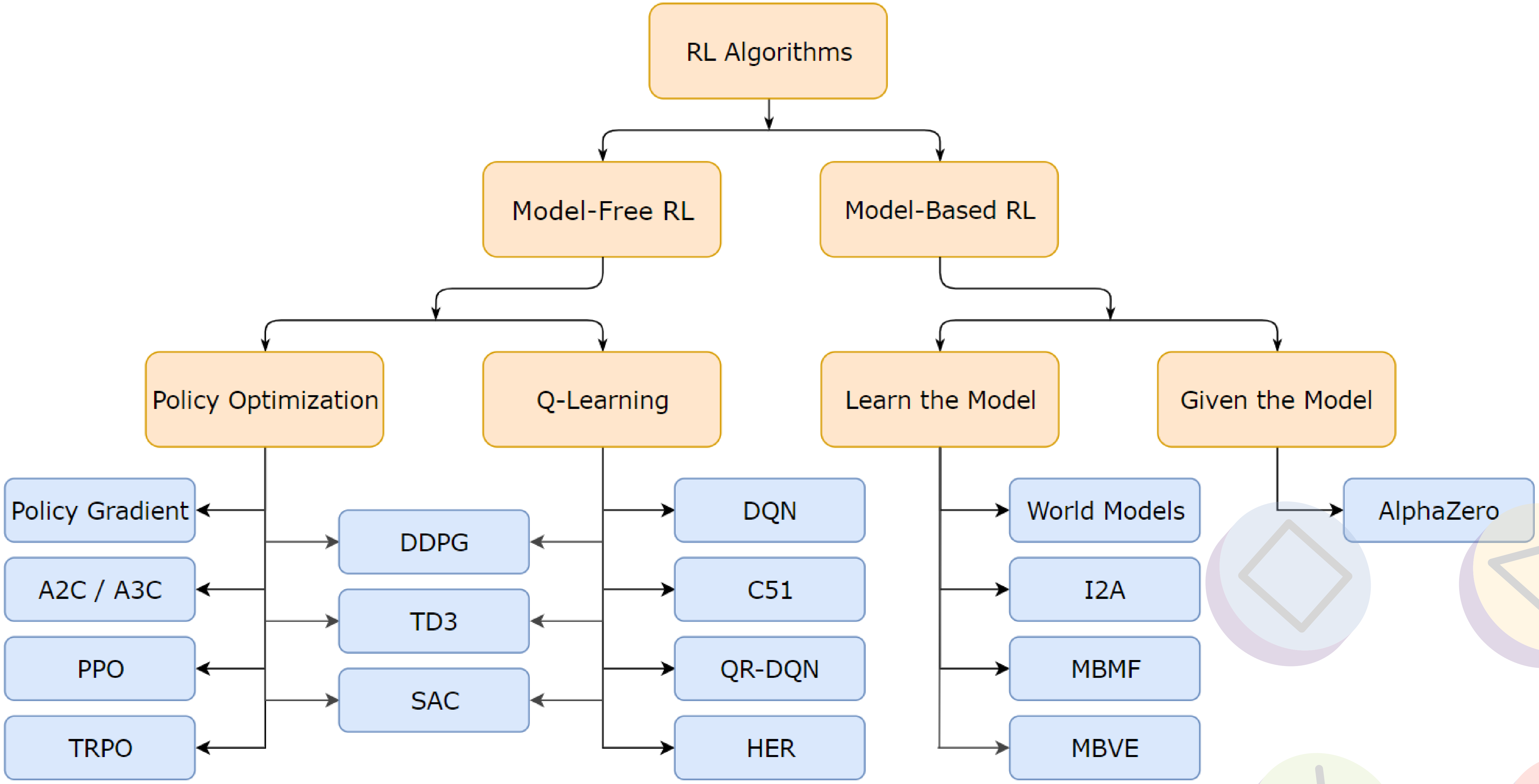
- Agent
- Environment
- State
- Action
- Reward



MODEL-FREE
VS
MODEL-BASED

- Model-Based: Agent can think ahead, and explicitly choose option by evaluating the possible choices.
- Model-Free: Easier to implement and tune.
  - Policy Optimization
  - Q-Learning

TYPES OF REINFORCEMENT LEARNING ALGORITHMS



MACHINE LEARNING TOOLS USED

python™

Keras

TensorFlow 2.0

EXPERIMENT SETUP

1. Initialize Environment from OpenAI GYM
2. Preprocess :
   GreyScaleObservation
   ResizeObservation
   SkipFrame
   FrameStack
3. Agent: Act, Remember, Learn
4. Train
5. Result

POLICIES

- Agent's Brain
- Deterministic or Stochastic
- Goal: Maximize Reward

$$a_t = \mu_\theta(s_t)$$

$$a_t \sim \pi_\theta(\cdot|s_t).$$

Parameterized Equation of Policies

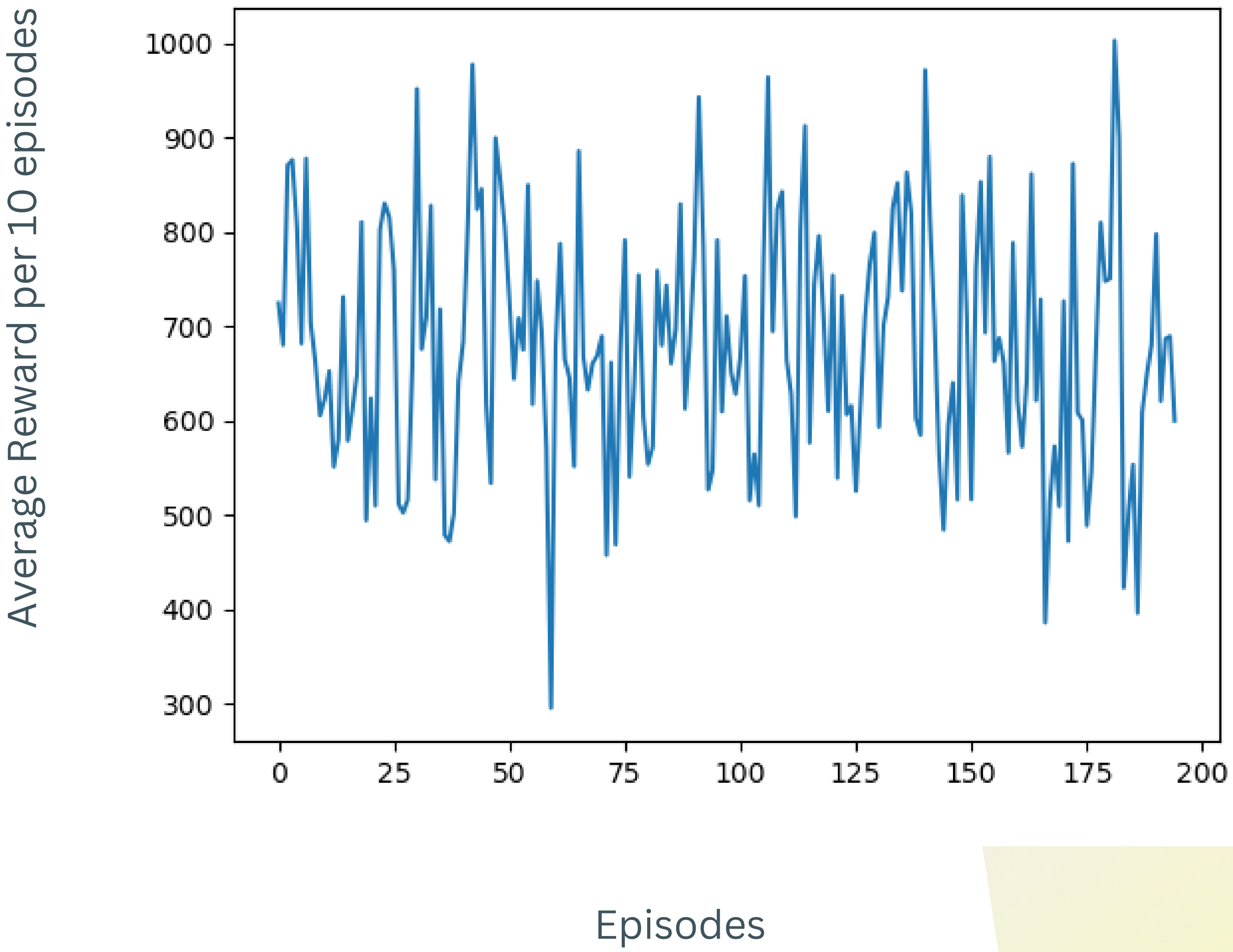
POLICY GRADIENT

- Optimizes the parameters of policy directly
- Gradient based approach
- Two main steps: Policy Evaluation, Policy Improvement
- Example application domain: Control, Navigation, Robotics.

$$a_t = \mu_\theta(s_t)$$

$$a_t \sim \pi_\theta(\cdot | s_t).$$

POLICY GRADIENT - RESULT

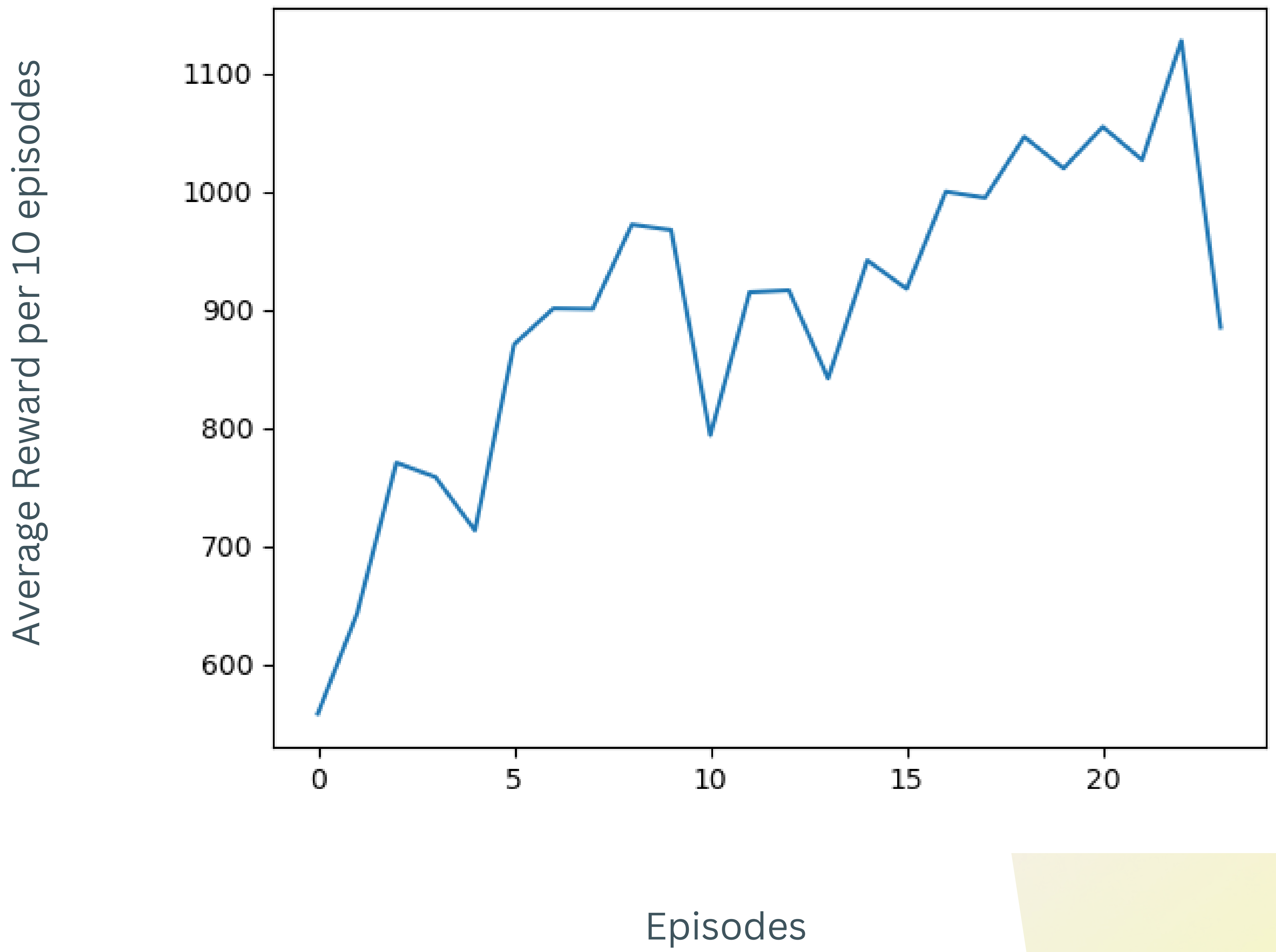


POLICY GRADIENT - DEMO

DEEP DOUBLE Q-LEARNING

- Combination of: Q-learning with Deep learning.
- Used two separate neural network.
- Replay buffer: Opportunity to learn from past.

DDQN - RESULT



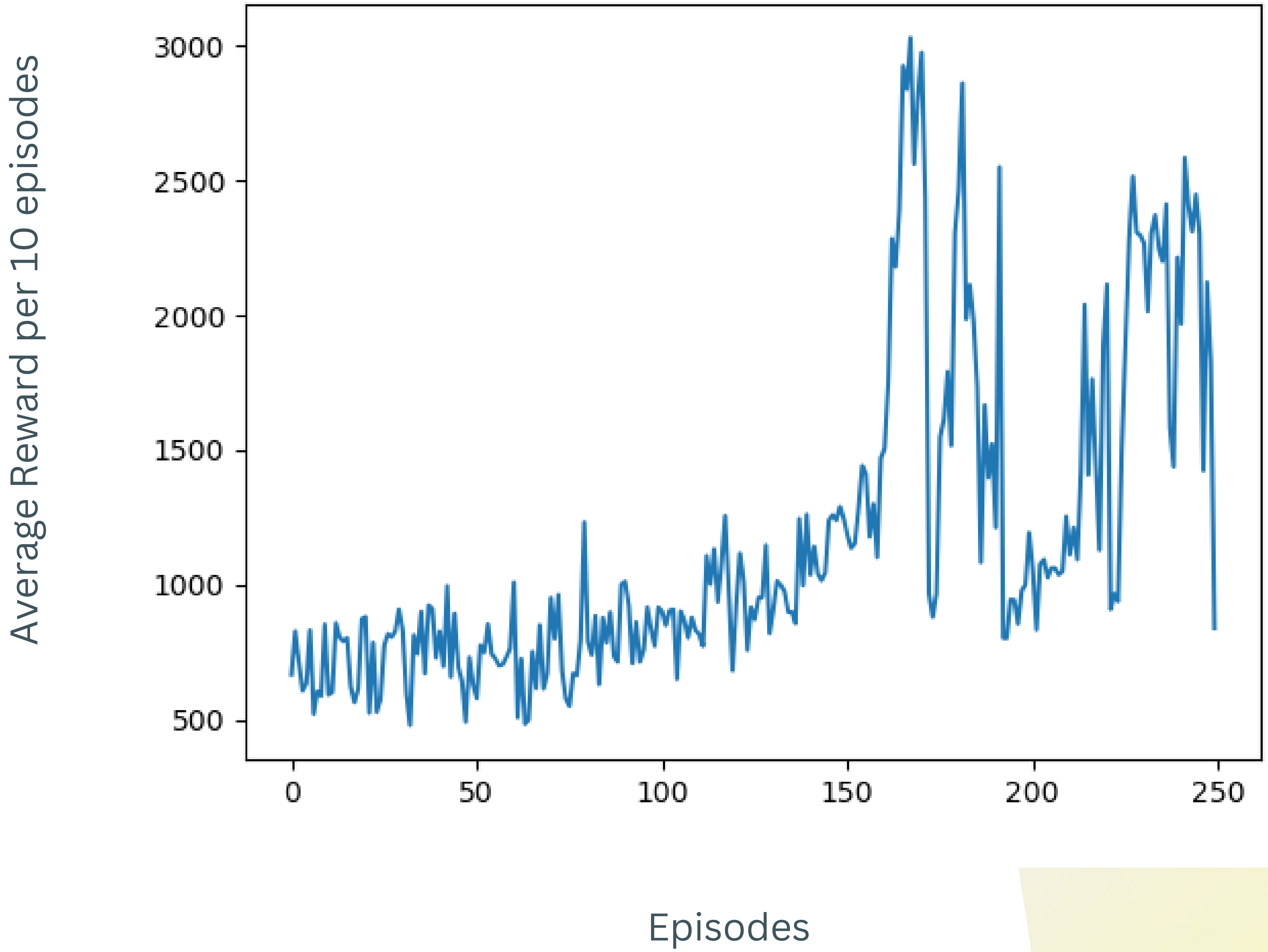
1100
1000
900
800
700
600
0
5
10
15
20
Average Reward per 10 episodes
Episodes

DDQN - DEMO

PROXIMAL POLICY OPTIMIZATION (PPO)

- Aims to maximize policy performance.
- Clipped objective function: Prevent too much change at each iteration.
- Adaptive learning rate.
- Trust region constraint: Controls the step size of policy update.

PPO - RESULT


Average Reward per 10 episodes
3000
2500
2000
1500
1000
500
0
50
100
150
200
250
Episodes

PPO - DEMO

LIMITATION & FUTURE WORK

- Loss plot, time plot.
- Other RL Algorithms.
- Experiment with learning rate, optimizer neural network layers.

THANK YOU