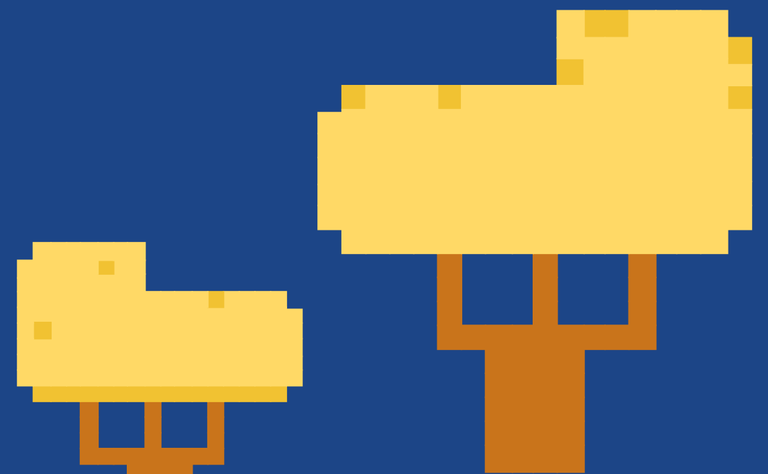
REINFORCEMENT LEARNING: PONG AND MORE

Mateo Jure, Aran Oliveras, Juan Sebastián Mañosas

TABLE OF CONTENTS



01

PONG ENVIRONMENT

Preprocessing and Wrappers

Atari Preprocessing

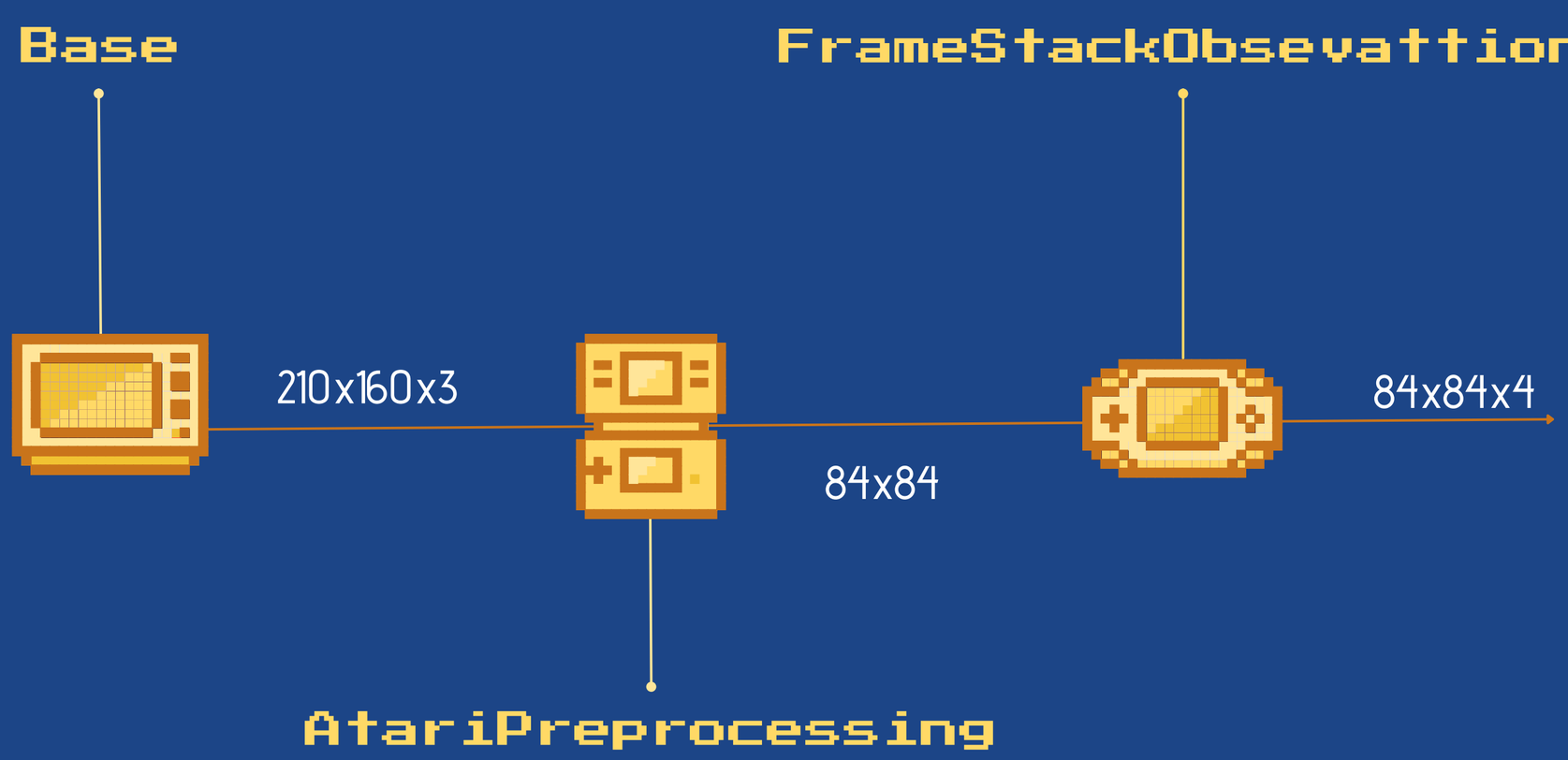
AtariPreprocessing implements the most common wrappers used for atari environments:

- Resizing to 84x84
- Grayscaling
- Observation scaling to [0,1]

Frame Stack

FrameStackObservation creates a markovian state that allows the agent to perceive motion

Output Shape



Selected Agents and Models

Deep Q-Network DQN

- Off-Policy
- Replay buffer
- Target Network

Proximal Policy Optimization PPO

- On-Policy
- Actor-Critic
- Clipped Objective Function

Architecture of the PPO Agent

Feature Extractor

- Conv2D (4, 32, kernel=8, stride=4) + ReLU
- Conv2D (32, 64, kernel=4, stride=2) + ReLU
- Conv2D (64, 64, kernel=3, stride=1) + ReLU

Heads

The output is flattened and passed to a linear layer which branches into:

→ Actor

→ Critic

Training and Fine-Tuning

Hyperparameter Sweep Ranges

Learning Rate

[1e - 5, 1e- 3]

Optimal: **~9e - 4**

Entropy Coefficient

[0.0, 0.02]

Optimal: **~0.016**

Clip Coefficient

[0.1, 03]

Optimal: **~0.18**

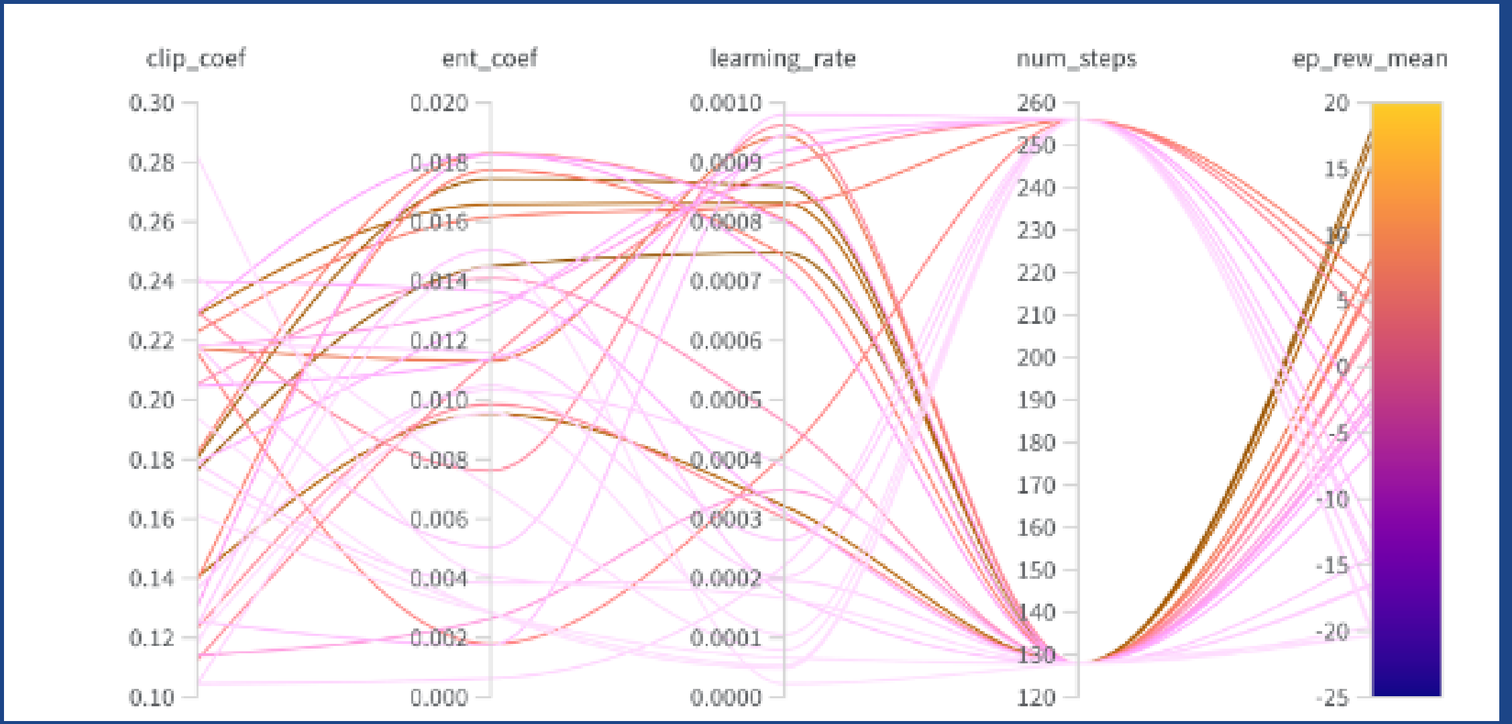
Number of Steps

{128, 256}

Optimal: **128**

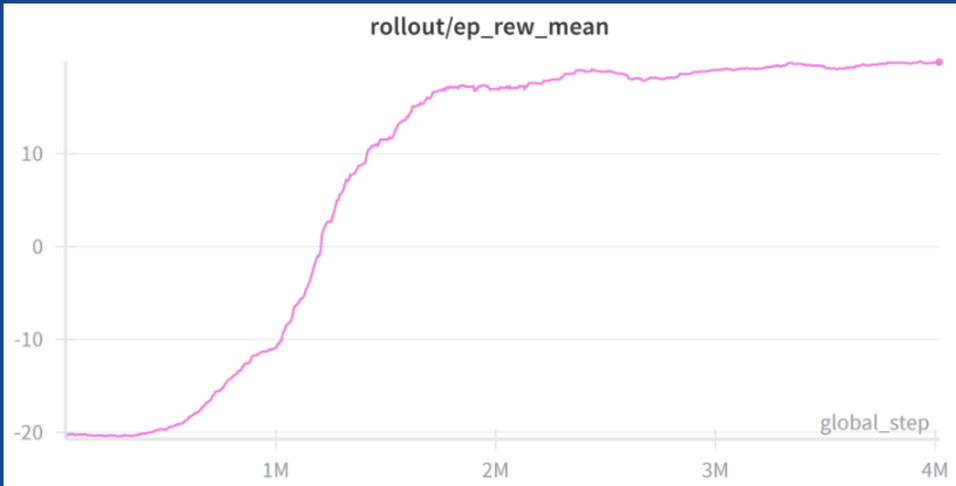
Training and Fine-Tuning

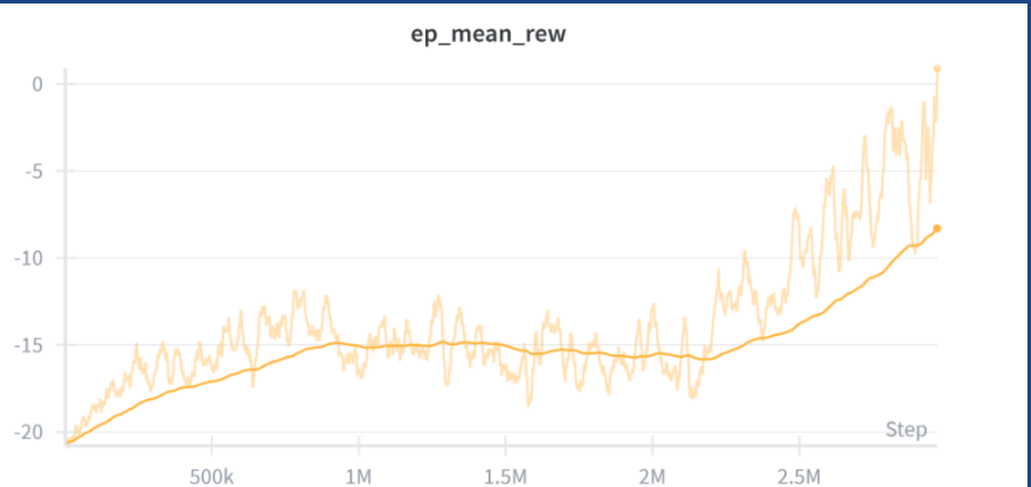
Hyperparameter Sweep



Visualizations and Learning Curves

Comparison of Average Episodic Reward





PPO Agent

DQN Agent

Evaluations and Success Rate

PPO Agent

Avg Reward (0 - 21)
20.44

Win Rate (%)
100%

DQN Agent

Avg Reward (0 - 21)
19.86

Win Rate (%)
100%

Results Analysis and Justification

PPO was selected as the superior model

WHY?

Time: Faster convergence due to parallel data collecting

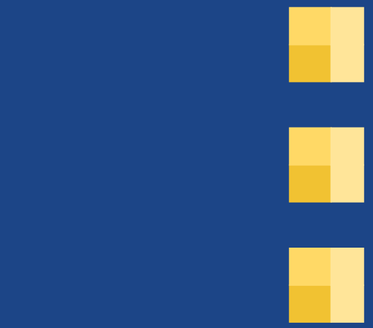
Stability: Less sensitive to initialization seeds

nuestro agente

un crack basicamente

02

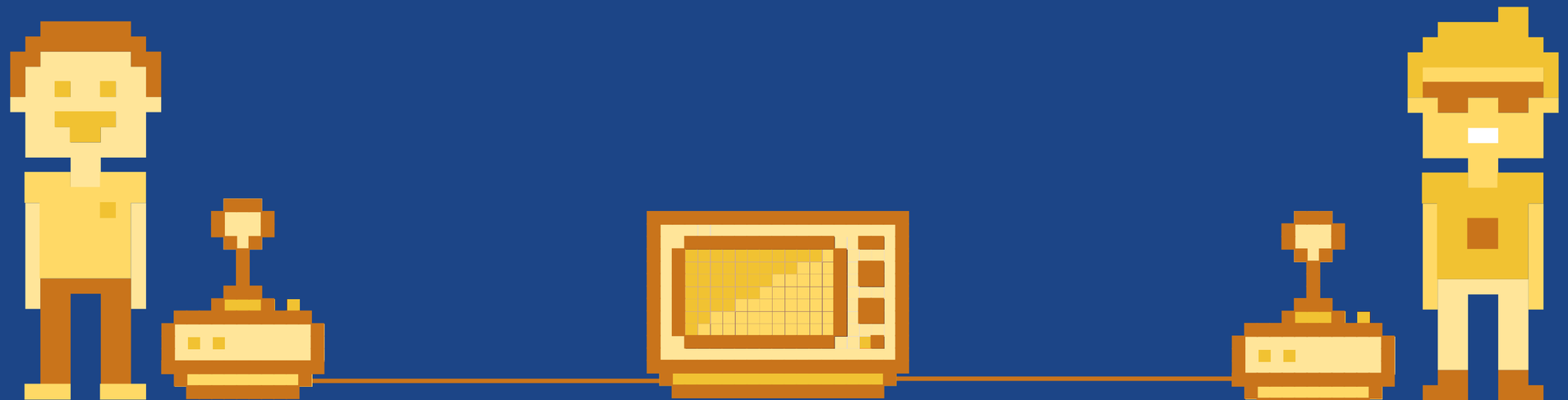
PONG TOURNAMENT



Our Philosophy

To win
Pong, we must play Pong; but
to win a tournament, we must
play a tournament

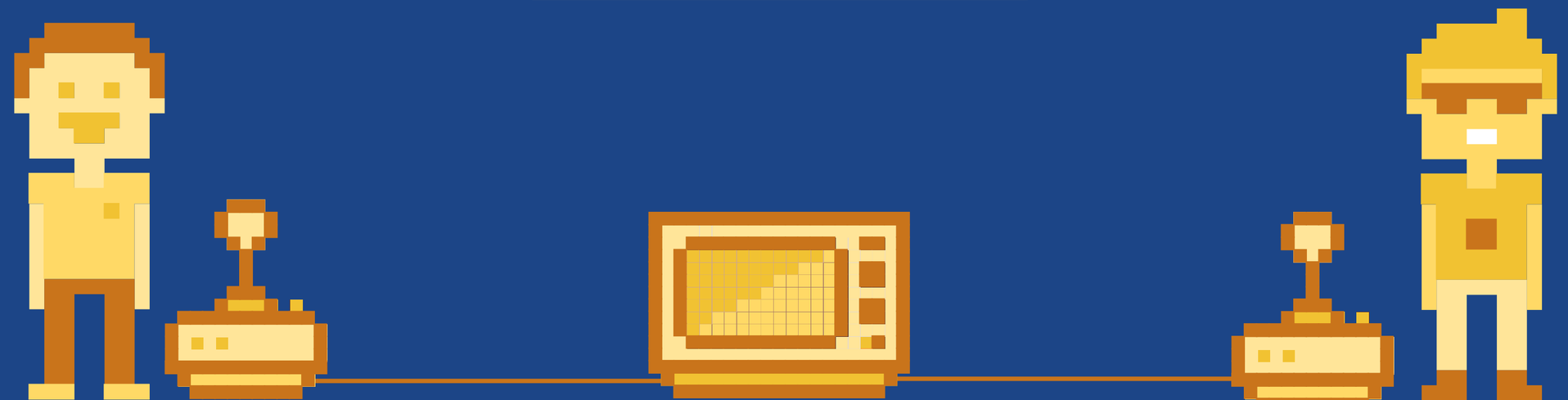
And so we did



Our Philosophy

To win
Pong, we must play Pong; but
to win a tournament, we must
play a tournament

try
And so we did



Model Selection

Model

PPG improves PPO by decoupling policy optimization from value learning

Architecture

ImpalaCNN was used to allow for much deeper networks to match the dynamics complexity

Customization

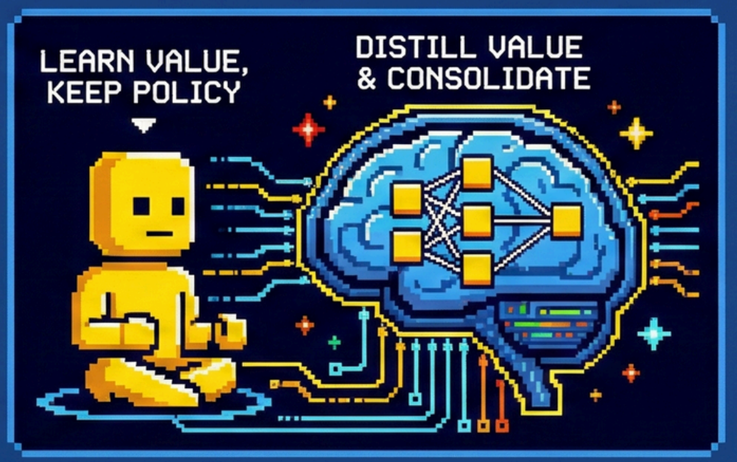
Everything was built from scratch using PyTorch

PPG

PHASIC POLICY GRADIENT

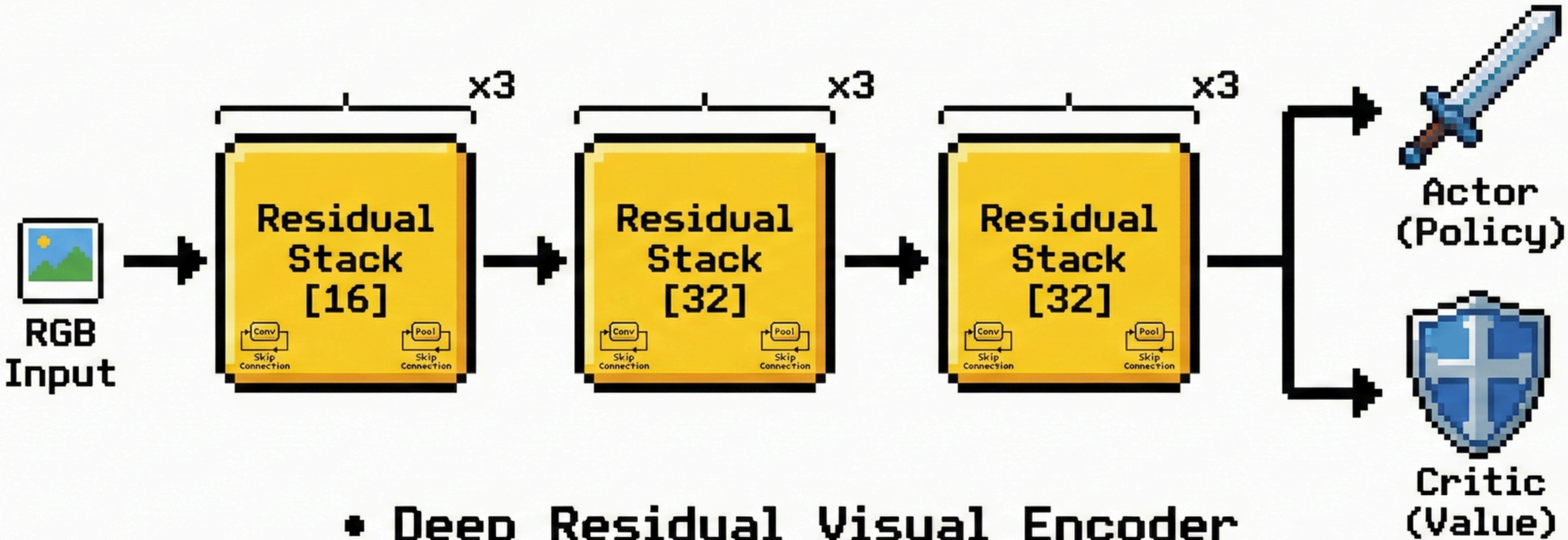
POLICY PHASE

AUXILIARY PHASE



- SOLVES INTERFERENCE IN SHARED NETWORKS
- TWO SEPARATE PHASES FOR EFFICIENT LEARNING
- REUSES DATA FOR BETTER SAMPLE EFFICIENCY

ImpalaCNN



- Deep Residual Visual Encoder
- LSTM Memory for Context
- Actor-Critic Output Heads

The Journey

Initial roster:

We wish it had worked!

It was too complex to manage

← Us

The Journey Pt.2

Reduced roster:

PPO Impala | PPG Nature | PPG Impala | DQN Impala

Alpha/Beta

Training:

Pre-Training (Gymnasium) » The Arena (PettingZoo) » Cluster Bottleneck Constraint

Hyperparameter Tuning

No Sweep This Time

Learning Rate (PPG)

5e - 4

Standard PPG Literature

Learning Rate (DQN)

1e - 4

Stability for Impala Deep Net

Gamma

0.99

Standard for Pong

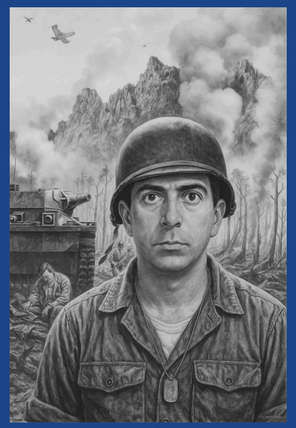
Parallel Environments

16

Max supported by memory

The Journey Pt.3

Reduced roster:



Alpha/Beta

Training:

Agent vs Agent

Training and Single-Player Results

PPG Impala

Reached parity with the computer without overfitting

DQN (Impala)

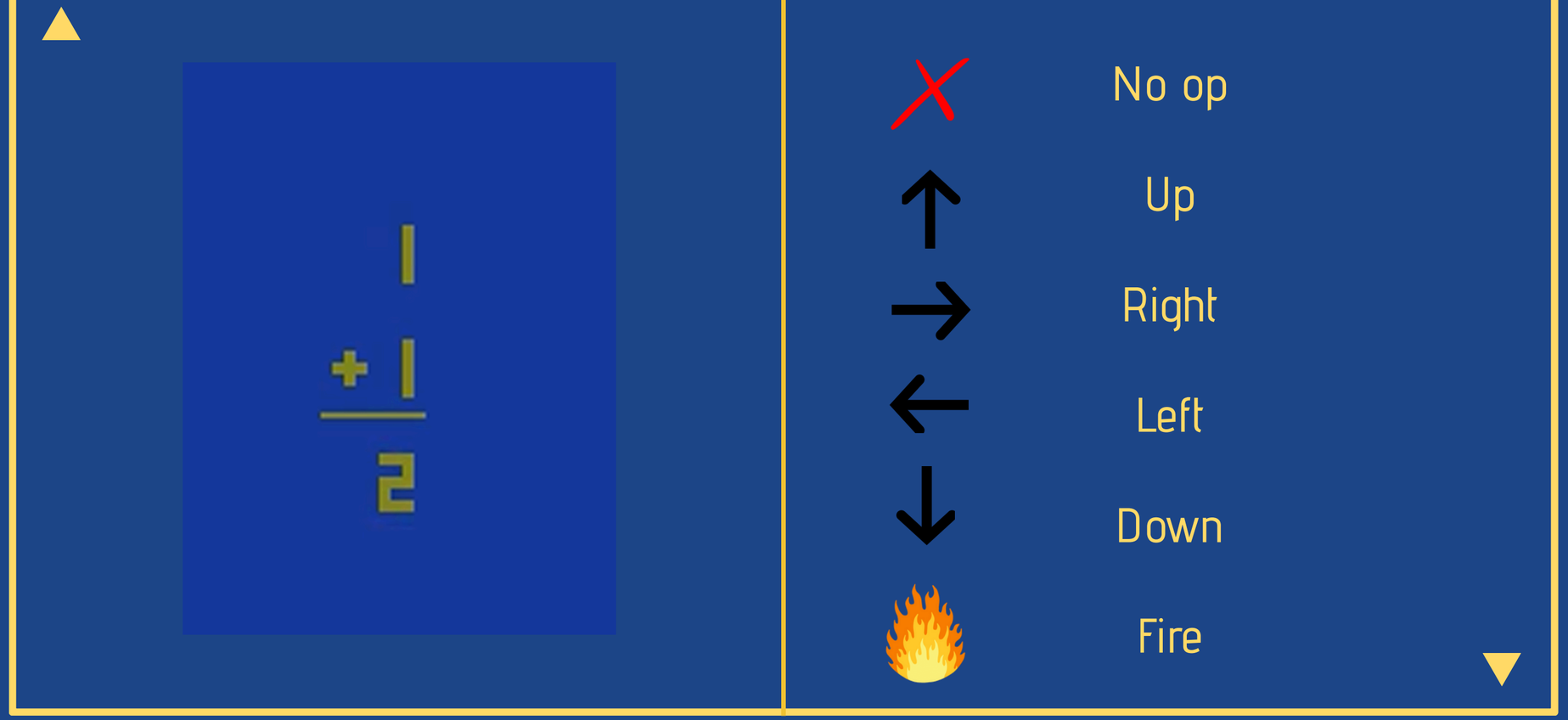
Converged slower but showed interesting behaviors

Performance in Multi-Agent Environment

03

OUR ALE ENVIRONMENT

Environment Description



Preprocessing and Wrappers

STANDARD PREPROCESSING

AtariPreprocessing and Framestack

CROP WRAPPER

Crops irrelevant border noise

BINARY WRAPPER

To assist CNN in feature extraction

TIME PENALTY

Encourages speed

ACTIVITY REWARD

Encourages movement

Agent Architecture and Algorithms

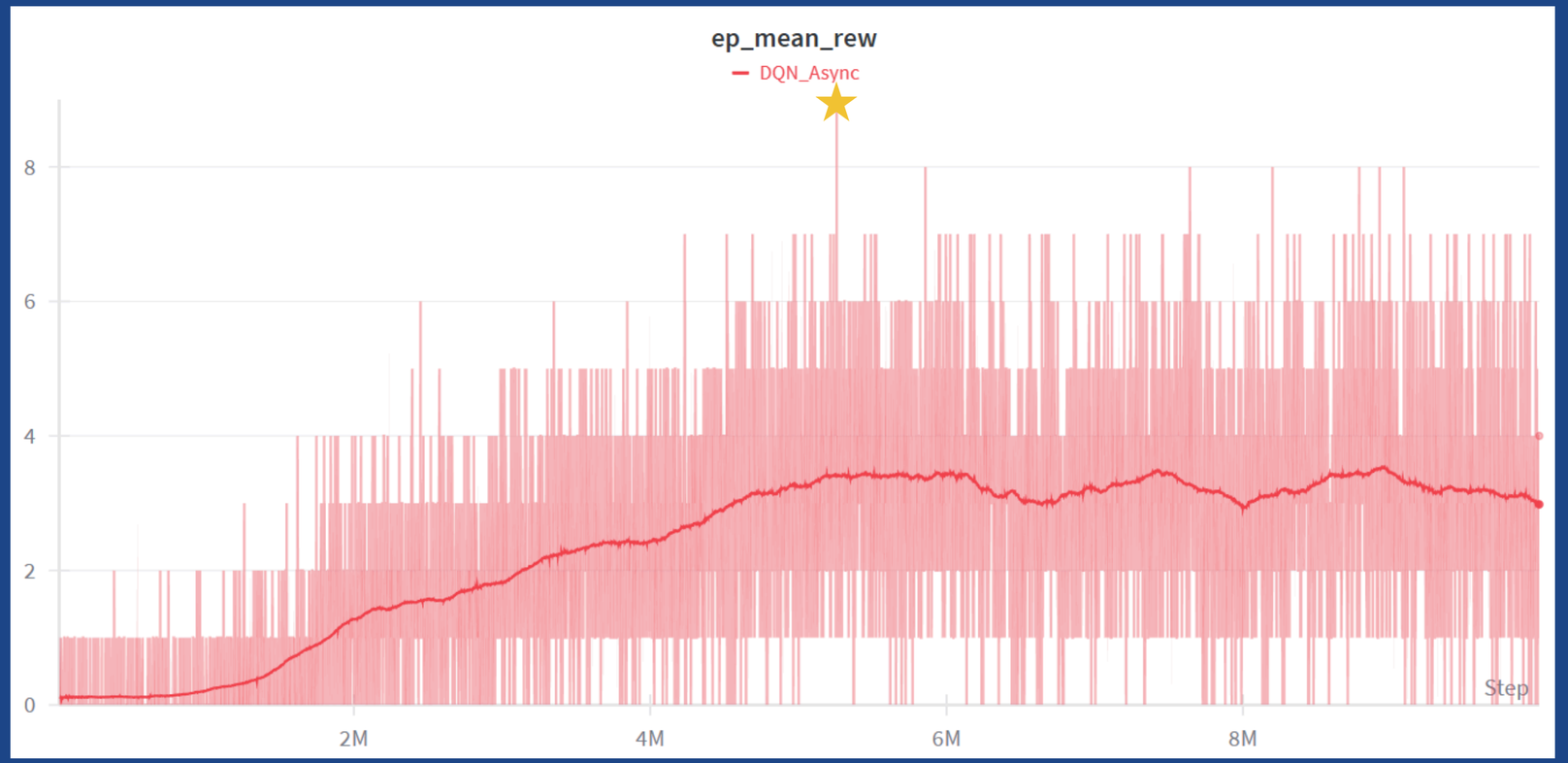
IMPALA ASYNCHRONOUS DQN

- Runs 16 parallel processes
- Maximizes sample efficiency
- Reduces temporal correlation

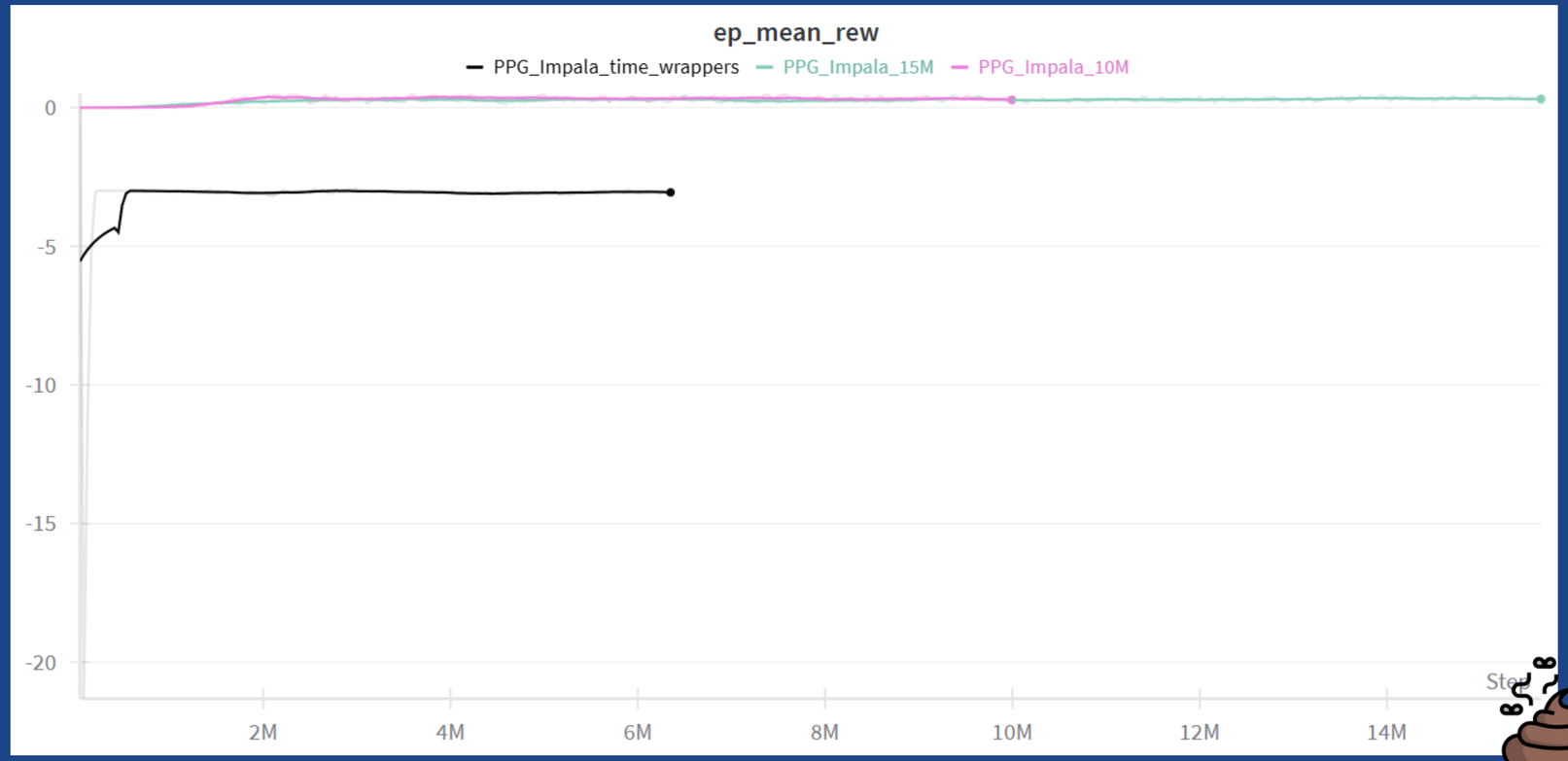
IMPALA PPG

- Policy optimization
- Every 16 actualizations predicts the value
- KL loss to not change drastically

Visualizations and Learning Curves



Visualizations and Learning Curves



Evaluation Metrics

METRICS	DQN	PPG
Max Reward	9.0	4.0
Avg Reward	3.5	0.25

Final Analysis and Justification

DQN was selected as the superior model

WHY?

Experience Replay: DQN reuses rare succeses | PPG discards them

Architecture Overhead: DQN learns < 4M steps | PPG needs > 50M

A VIDEO IS WORTH A THOUSAND... NUMBERS

THANKS!

Do you have any questions?
youremail@freepik.com
+91 620 421 838
yourcompany.com