Deep Reinforcement Learning method for Humanoid Kick Motion

Luckeciano C. Melo

Advisor: Prof. Dr. Adilson Marques da Cunha

Co-advisor: Prof. Dr. Marcos R. O. A. Máximo

Summary

- Introduction
- Background
- Deep Learning
- Reinforcement Learning
- Methodology
- Results
- Conclusions and Future Work

Introduction

Examples of Reinforcement Learning

AlphaGo seals 4-1 victory over Go grandmaster Lee Sedol

DeepMind's artificial intelligence astonishes fans to defeat human opponent and offers evidence computer software has mastered a major challenge

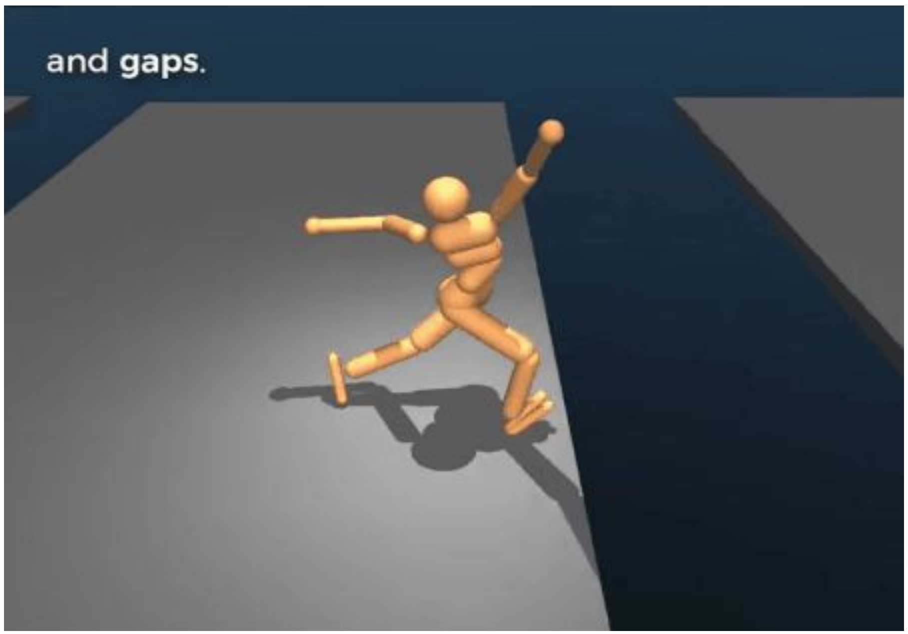
Google DeepMind's AlphaGo program triumphed in its final game against South Korean Go grandmaster Lee Sedol to win the series 4-1, providing further evidence of the landmark achievement for an artificial intelligence program.

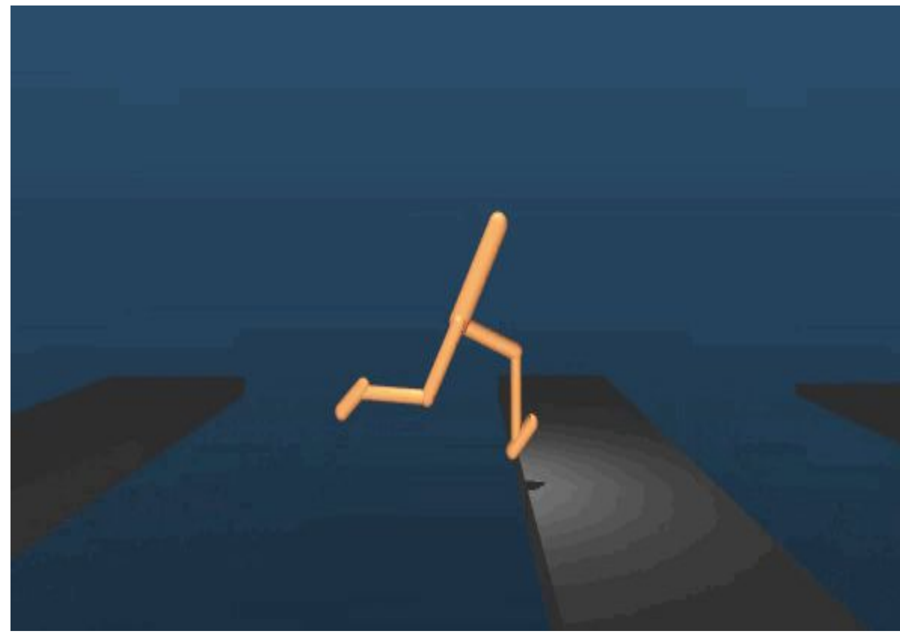
Play Go very well and without human knowledge

SILVER *et al*. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. CoRR, abs/1712.01815, 2017.

Introduction

Examples of Reinforcement Learning





Humanoid Walk (and Parkour)

HEESS *et al.* Emergence of locomotion behaviours in rich environments. CoRR, abs/1707.02286, 2017.

Introduction

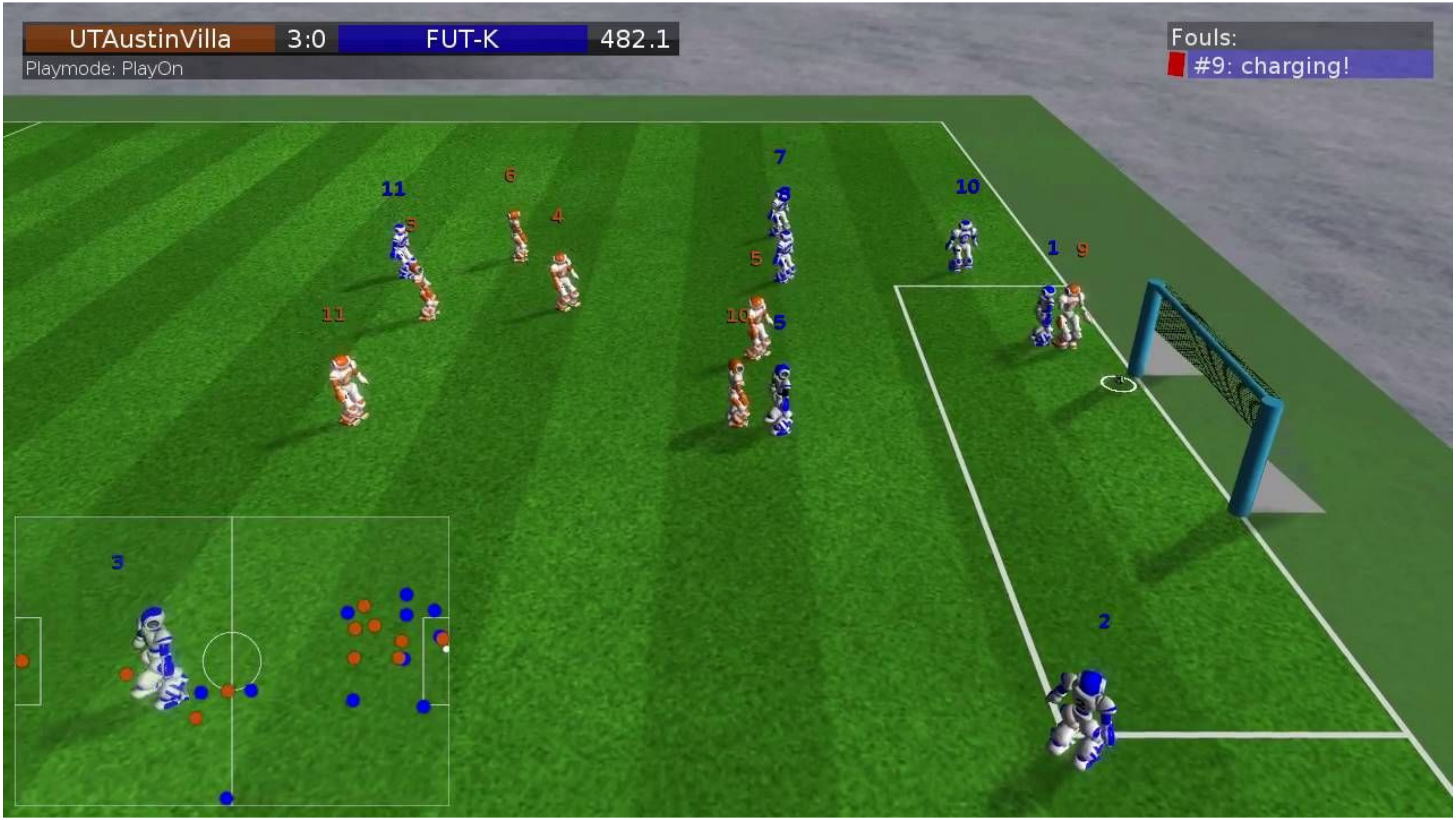
Examples of Reinforcement Learning

OpenAI Five

https://blog.openai.com/openai-five/

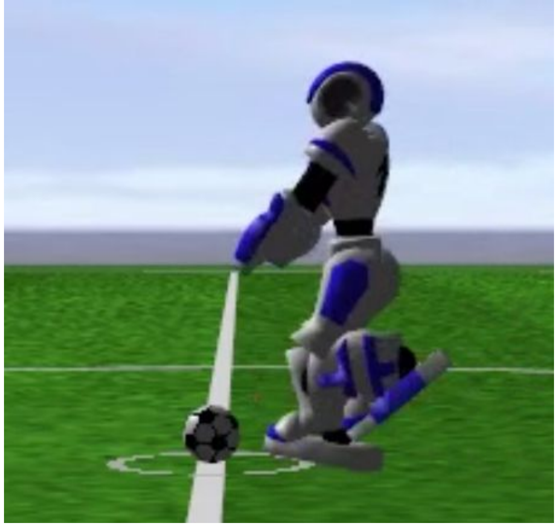
Introduction

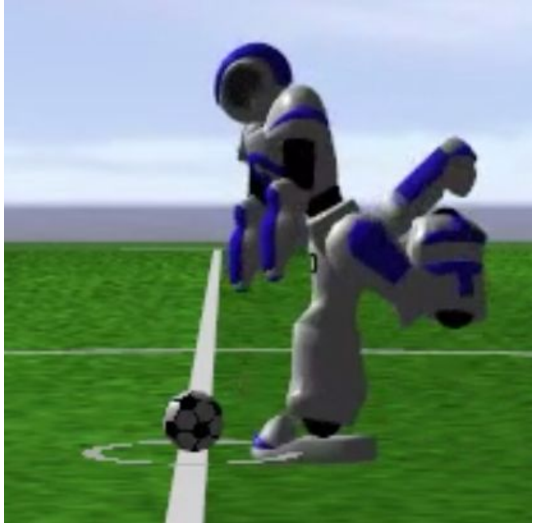
Domain Description



Introduction

Kick - Keyframe



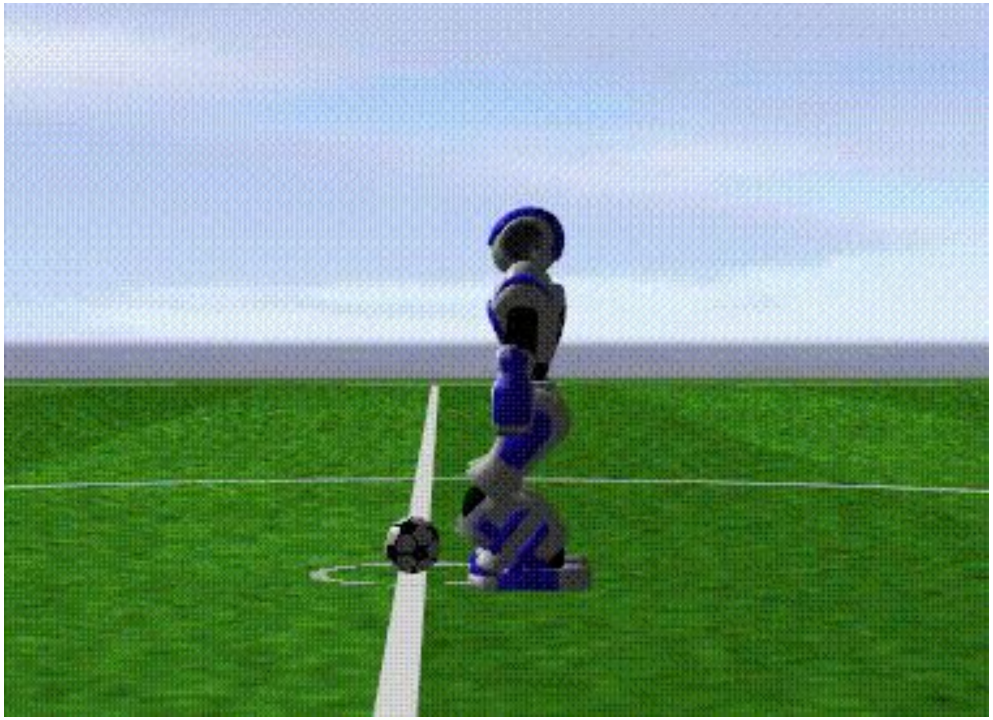


T1 T2 T3

Introdução

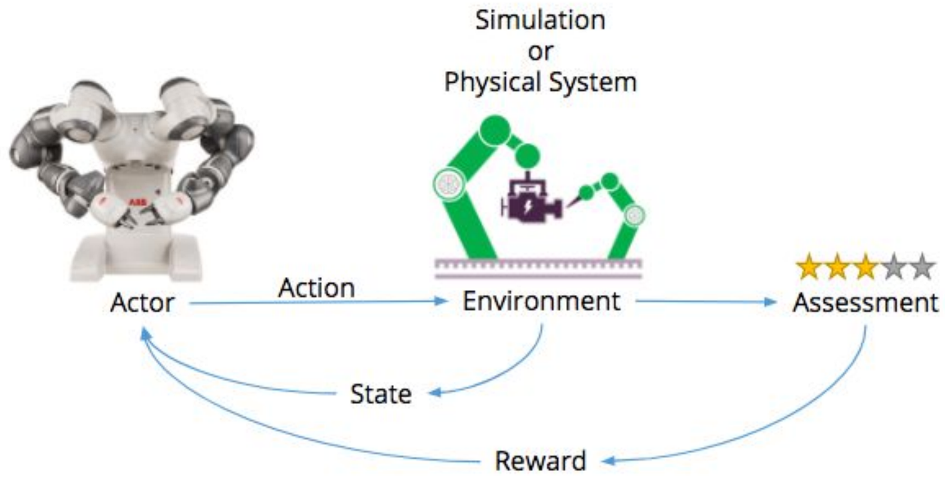
Objective

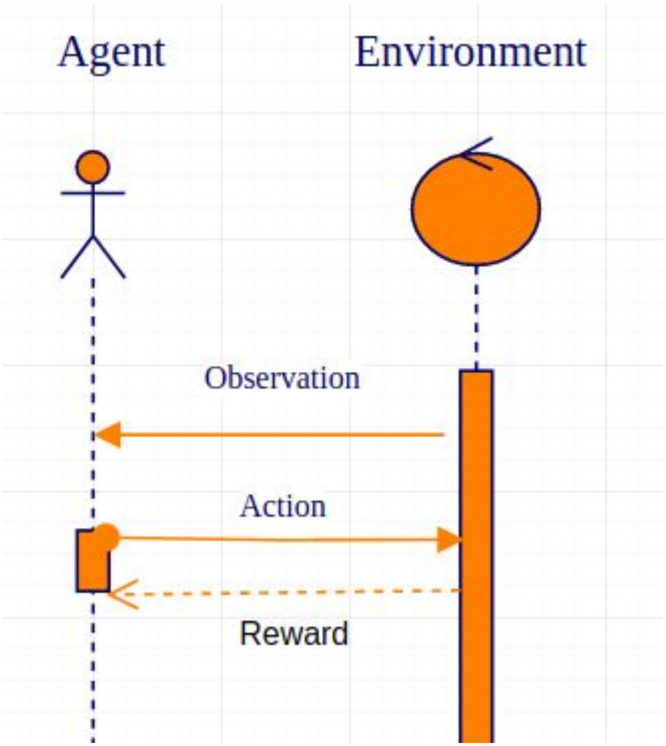
Find optimal policies for humanoid robot kick motion through Deep Reinforcement Learning

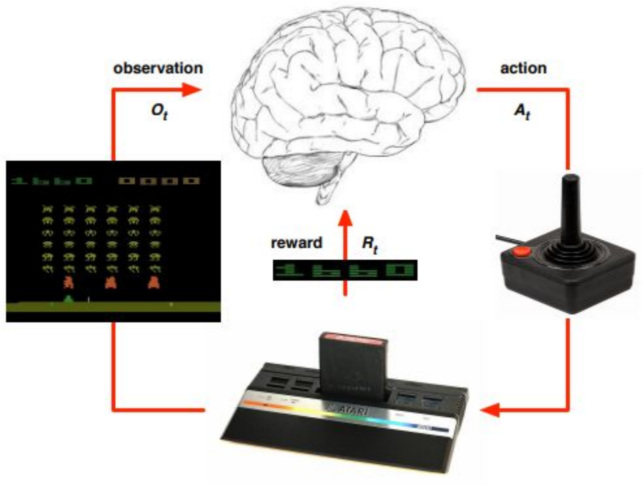


Background

Reinforcement Learning System

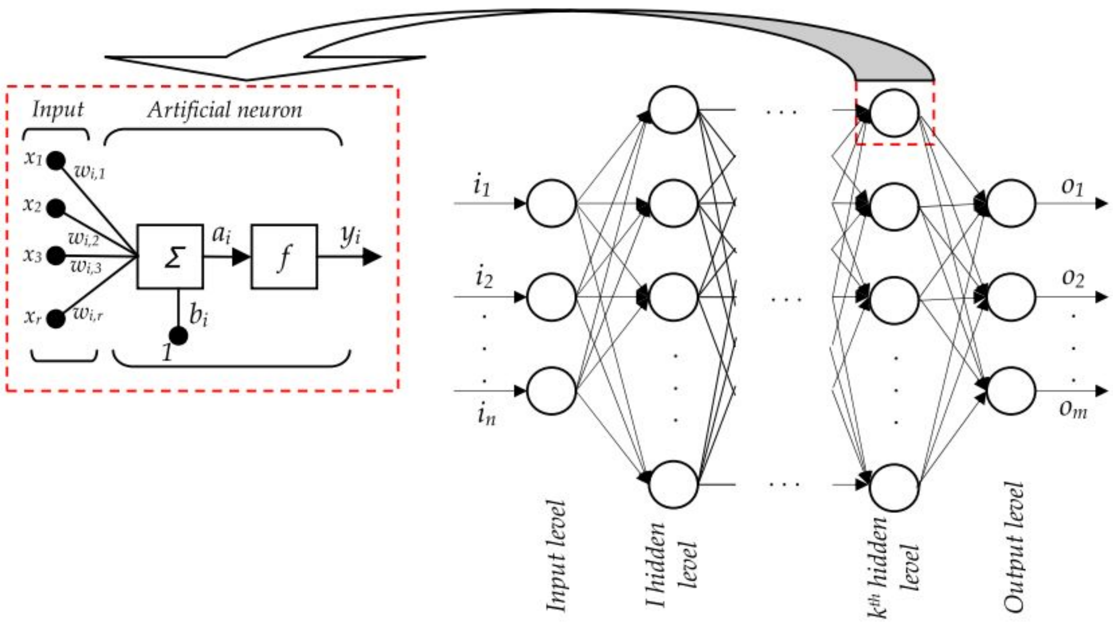






Deep Learning

Neural Networks



$$J(\boldsymbol{\theta}) = -\mathbb{E}_{\mathbf{x},\mathbf{y}\sim\hat{p}_{data}} \log p_{model}(\mathbf{y}|\mathbf{x})$$

$$\nabla_{\mathbf{X}} z = \sum_j (\nabla_{\mathbf{X}} Y_j) \frac{\partial z}{\partial Y_j}$$

Reinforcement Learning

Markov Decision Process

A Markov Decision Process, is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$*, where:*

- $\mathcal{S}$ *is a set of states;*
- $\mathcal{A}$ *is a set of actions;*
- $\mathcal{P}$ *is the state transition probability matrix;*
- $\mathcal{R}$ *is a reward function, i.e,* $\mathcal{R}_s = \mathbb{E}[R_{t+1} | S_t = s]$*; and*
- γ *is a discount factor, where* $\gamma \in [0, 1]$.

Reinforcement Learning

Value Function

- **Return**

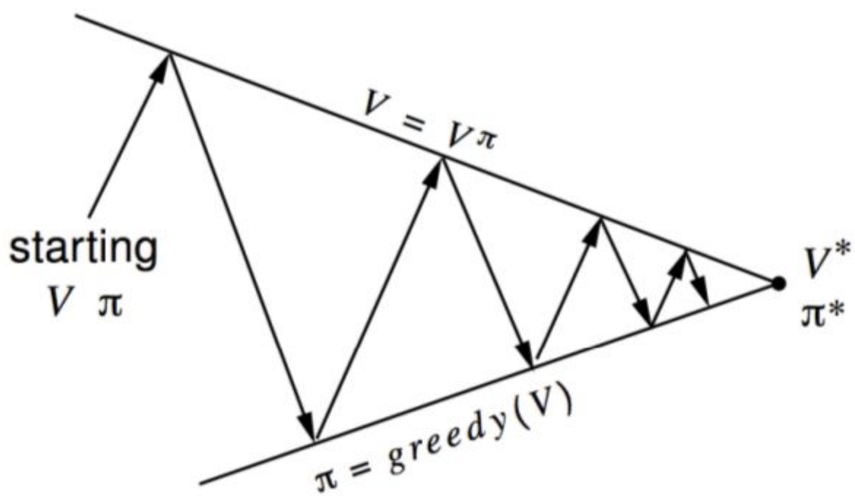
$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- **Value Function**

$$v_\pi(s) = \mathbb{E}_\pi[G_t \big| S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \middle| S_t = s\right]$$

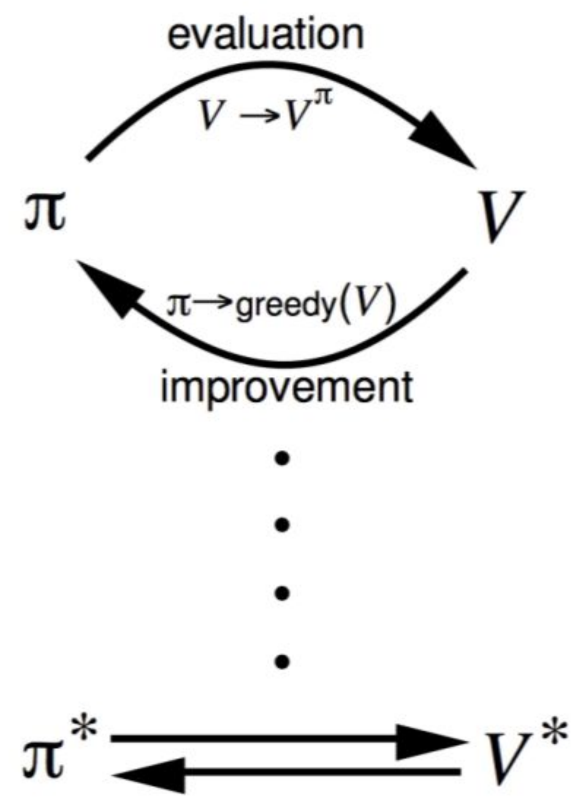
Reinforcement Learning

Generalized Policy Iteration - Control



Policy evaluation Estimate v_π
e.g. Iterative policy evaluation

Policy improvement Generate $\pi' \geq \pi$
e.g. Greedy policy improvement



Available in: http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/DP.pdf

Algorithm: Proximal Policy Optimization

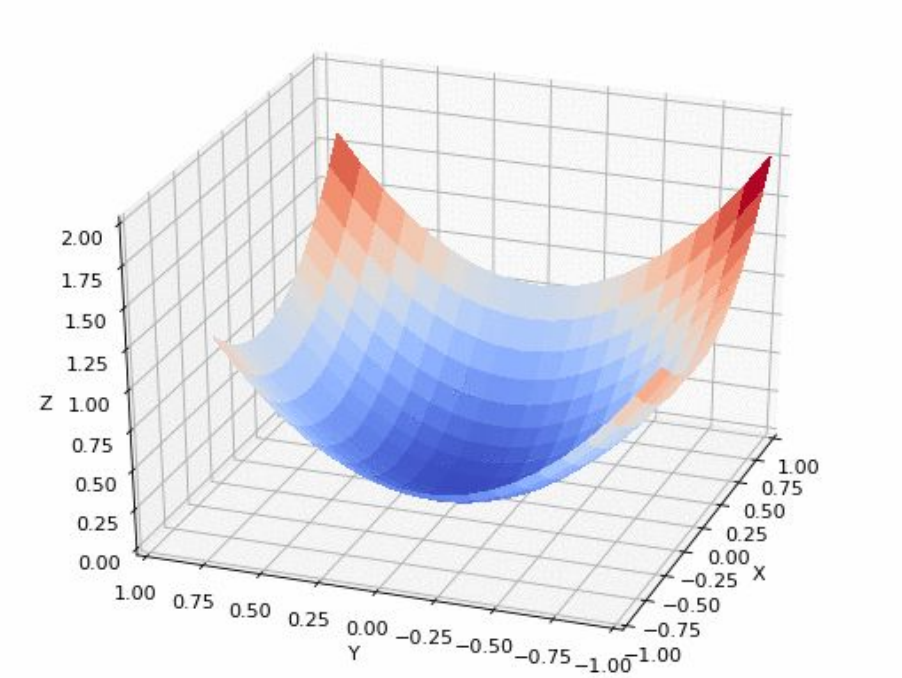
Algorithm 1 PPO, Actor-Critic Style

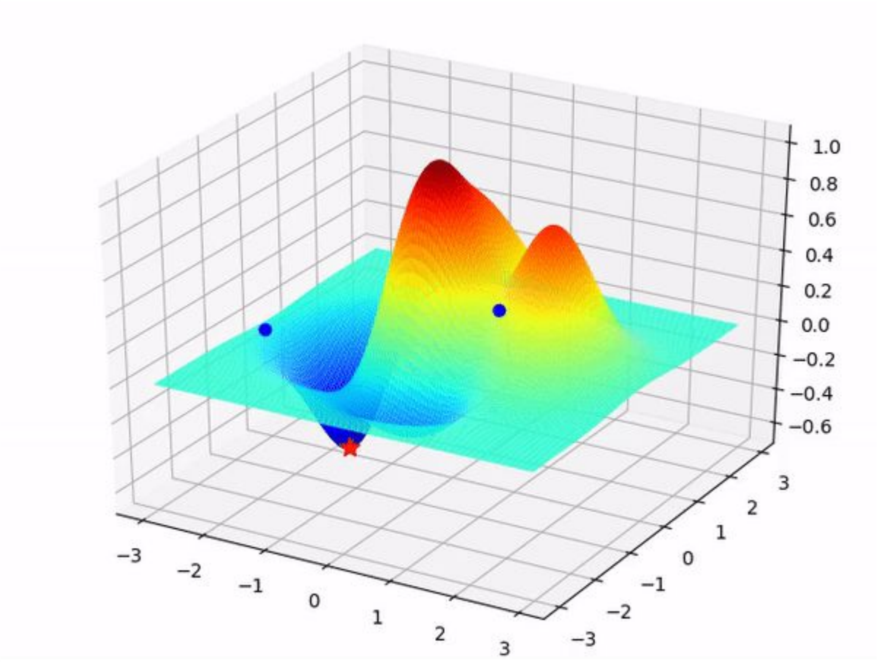
for iteration=$1, 2, \ldots$ **do**
  for actor=$1, 2, \ldots, N$ **do**
    Run policy $\pi_{\theta_{\text{old}}}$ in environment for T timesteps
    Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
  end for
  Optimize surrogate L wrt θ, with K epochs and minibatch size $M \leq NT$
  $\theta_{\text{old}} \leftarrow \theta$
end for

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t \left[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t) \right]$$

SCHULMAN, J.; WOLSKI, F.; DHARIWAL, P.; RADFORD, A.; KLIMOV, O.Proximal policy optimization algorithms. CoRR, abs/1707.06347, 2017. Dispon ́ıvel em: <http://arxiv.org/abs/1707.06347>.

Learning → Optimization Problem



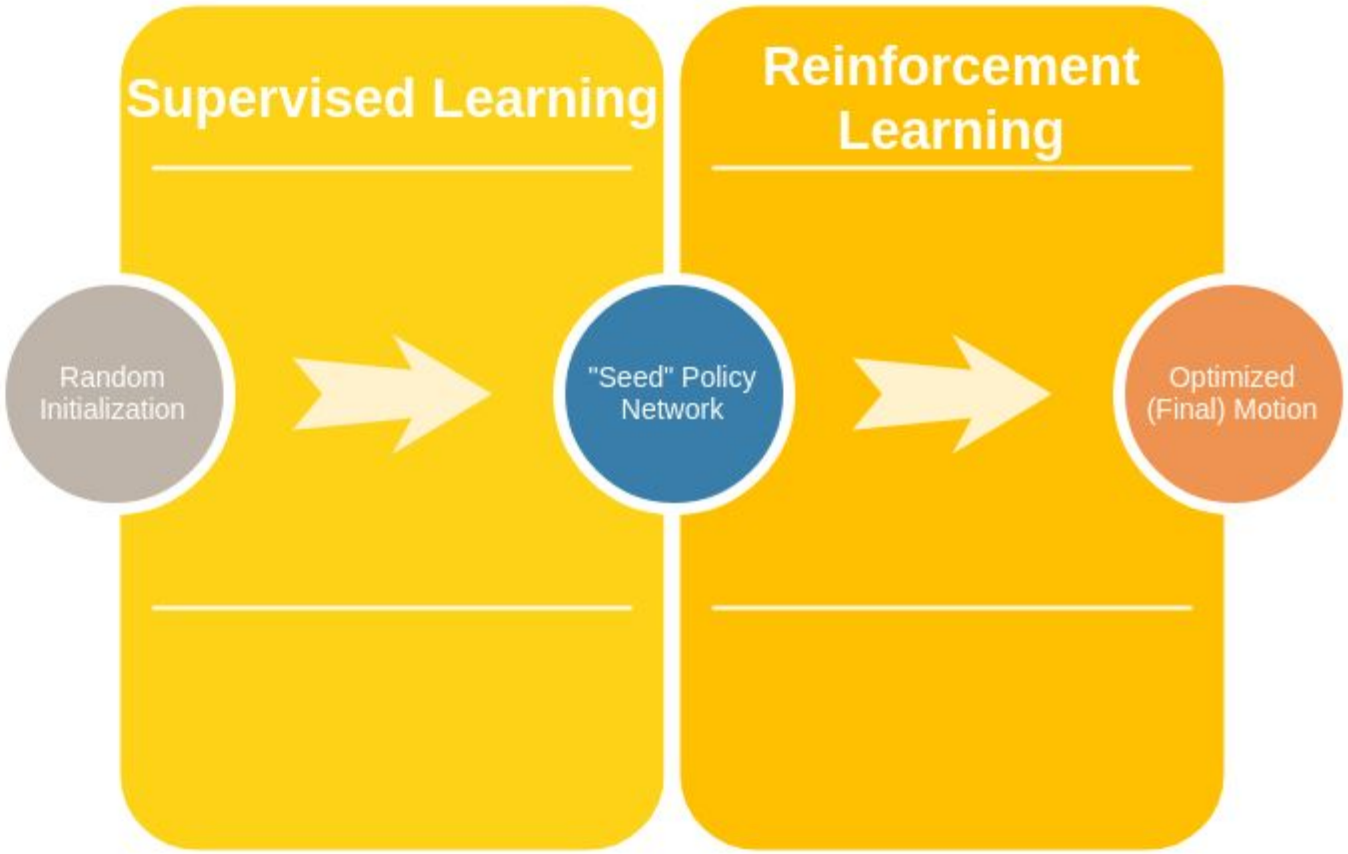


Hypothesis

- **Let suppose a policy represented by a neural network with thousands of parameters, where:**
    - **There is a first training phase, supervised, that copies the keyframe motion to this neural network; and**
    - **There is a second training phase, using reinforcement learning, which optimizes the neural network motion**
- **Therefore, we will have a better policy than that based on keyframe representation.**

Methodology

Approach - Hybrid Learning Model



Methodology

Approach - Reinforcement Learning

- **Reinforcement for "Naive" Reward - RNR**

$$R(s) = u^T v$$

- **Reinforcement for Reference Motion - RRR**

$$R(s) = w_{ref}^T(\pi - r)$$

- **Reinforcement for Initial State Distribution - RISD**

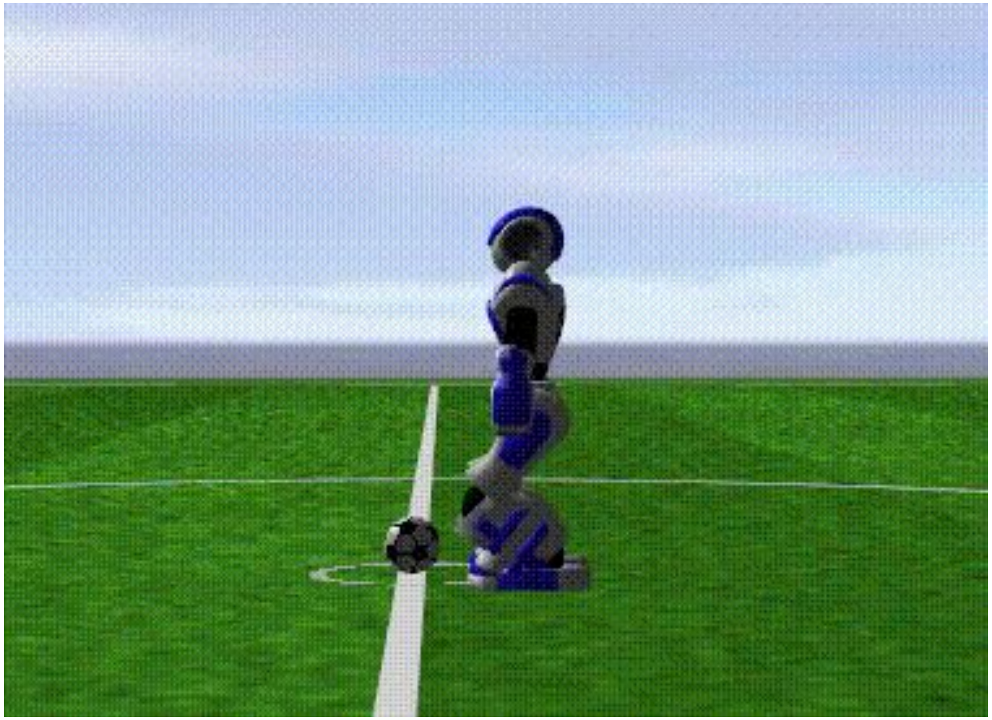
- **Reinforcement for Early Termination - RET**

Methodology

Supervised Learning - Overview

Methodology

Supervised Learning - Dataset



Methodology

Supervised Learning - Architecture

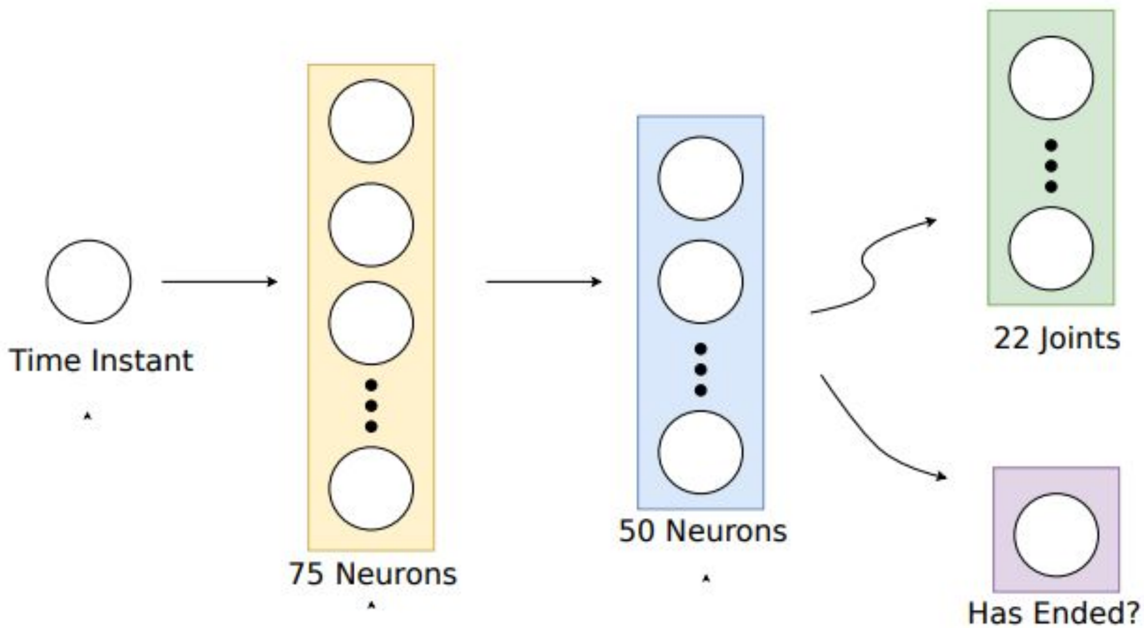


TABLE 4.1 – The Network Summary

Layer	Neurons	Activation	Parameters
Dense	75	LeakyReLU	130
Dense	50	LeakyReLU	3800
Dense	23	Linear	1173

Total Parameters	5123

Methodology
“Pure” Reinforcement Learning - Architecture

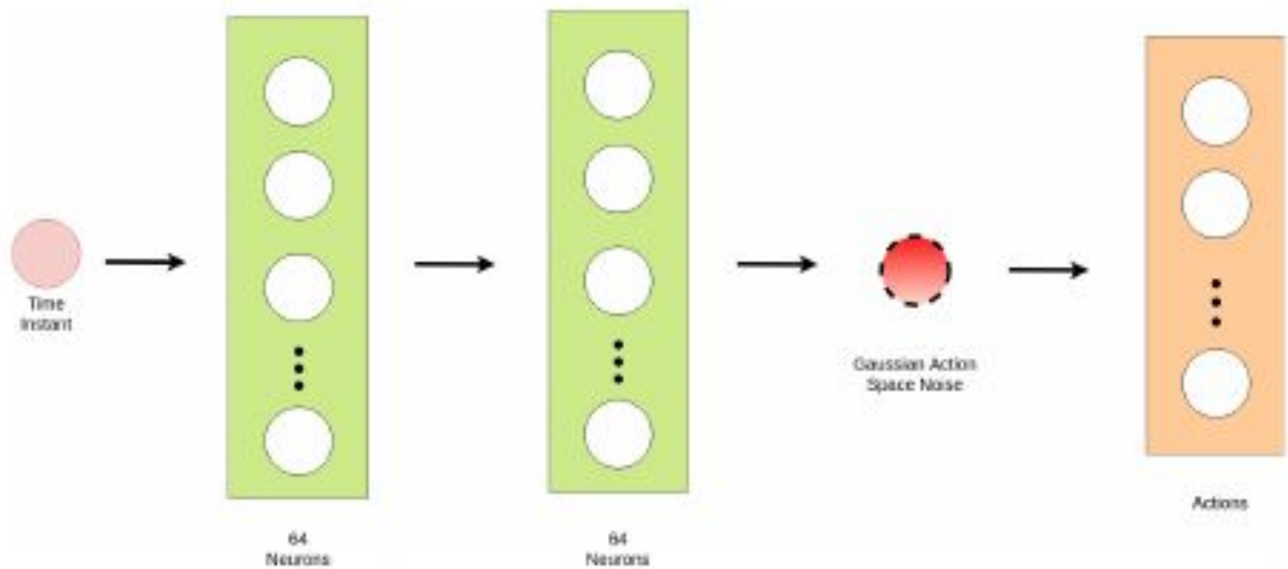


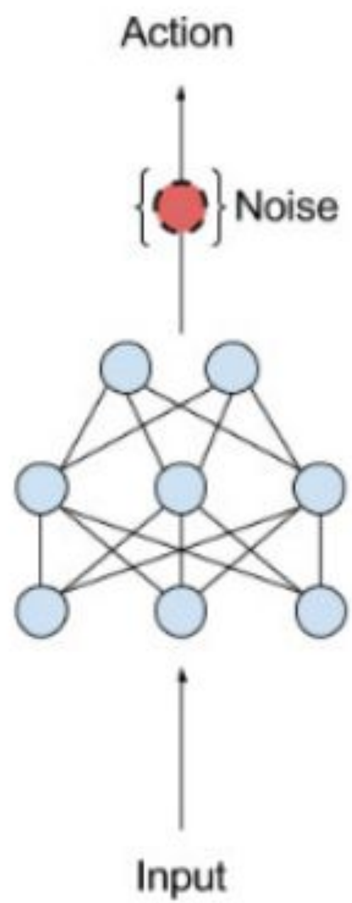
TABLE 5.2 – The Reinforcement Learning Network Summary.

Layer	**Neurons**	**Activation**	**Parameters**
Dense	64	*tanh*	128
Dense	64	*tanh*	4160
Output	23	Linear	1495
Noise	23	Linear	23

Total Parameters	5806

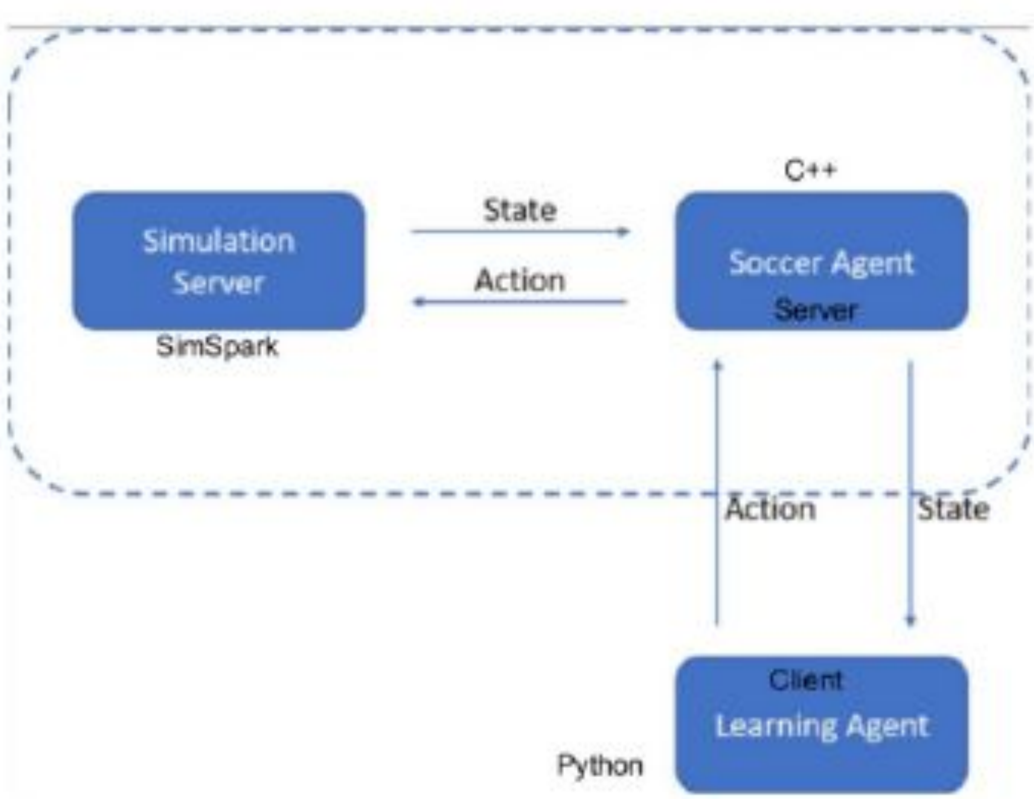
Methodology

Exploration: Gaussian Noise



Methodology

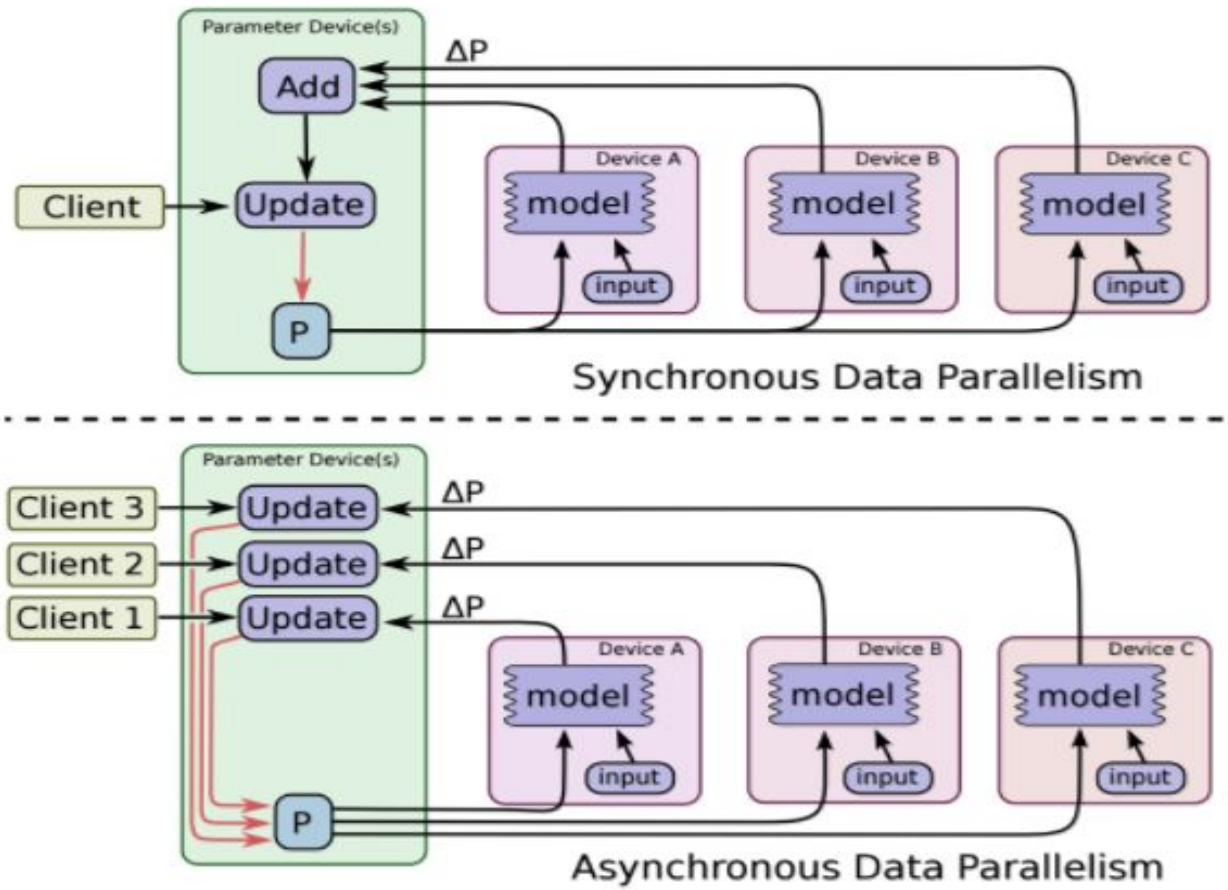
Infrastructure



Methodology

Infrastructure

- **Distributed Training**



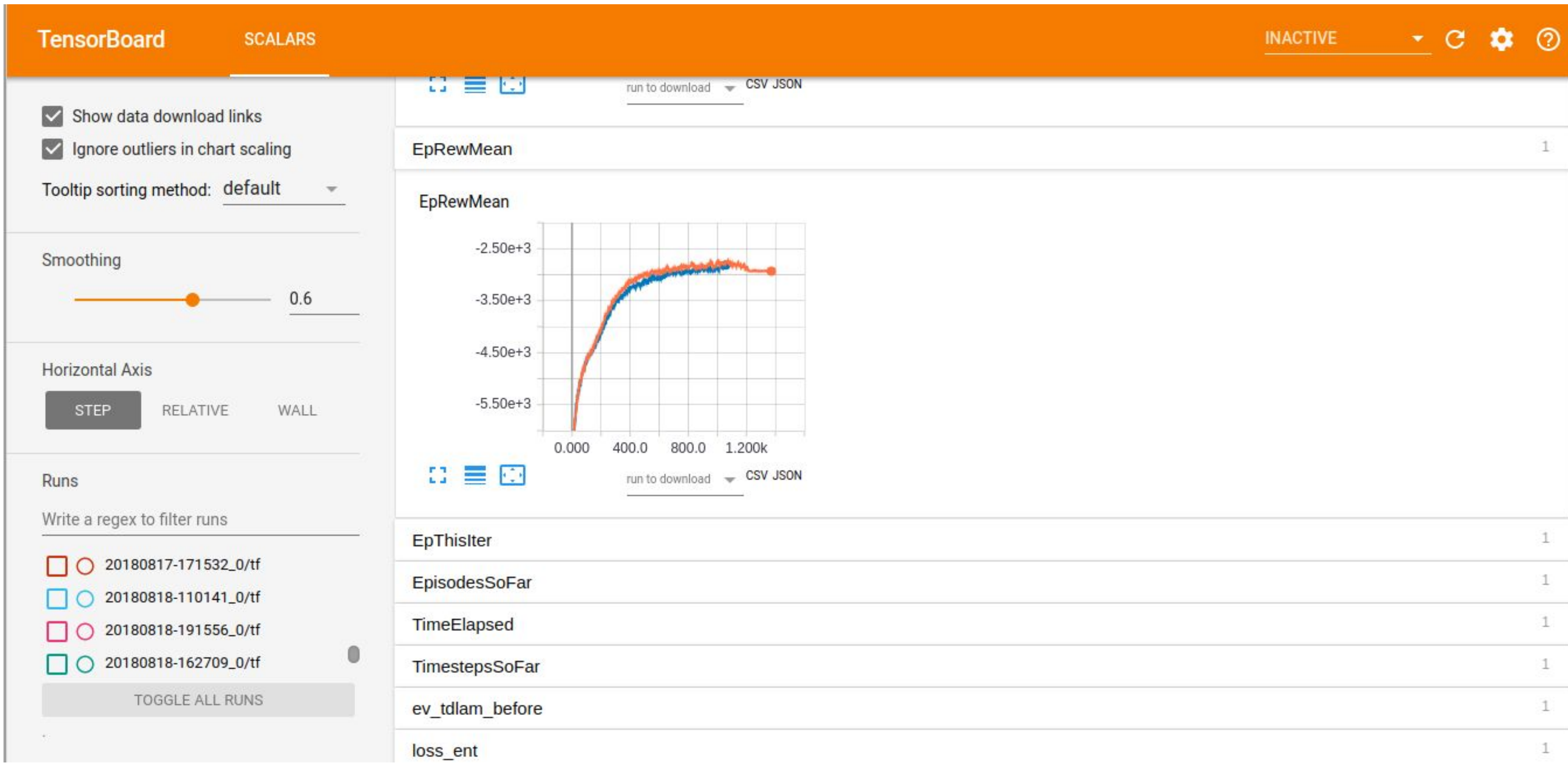
Methodology
Infrastructure

- **Distributed Training**

Computação em nuvem gratuita está disponível para os membros da Intel® AI Academy. Use o Intel® AI DevCloud equipado com processadores escalonáveis Intel® Xeon® para treinamento de aprendizado de máquina e aprendizagem profunda e necessidades de computação de inferência.

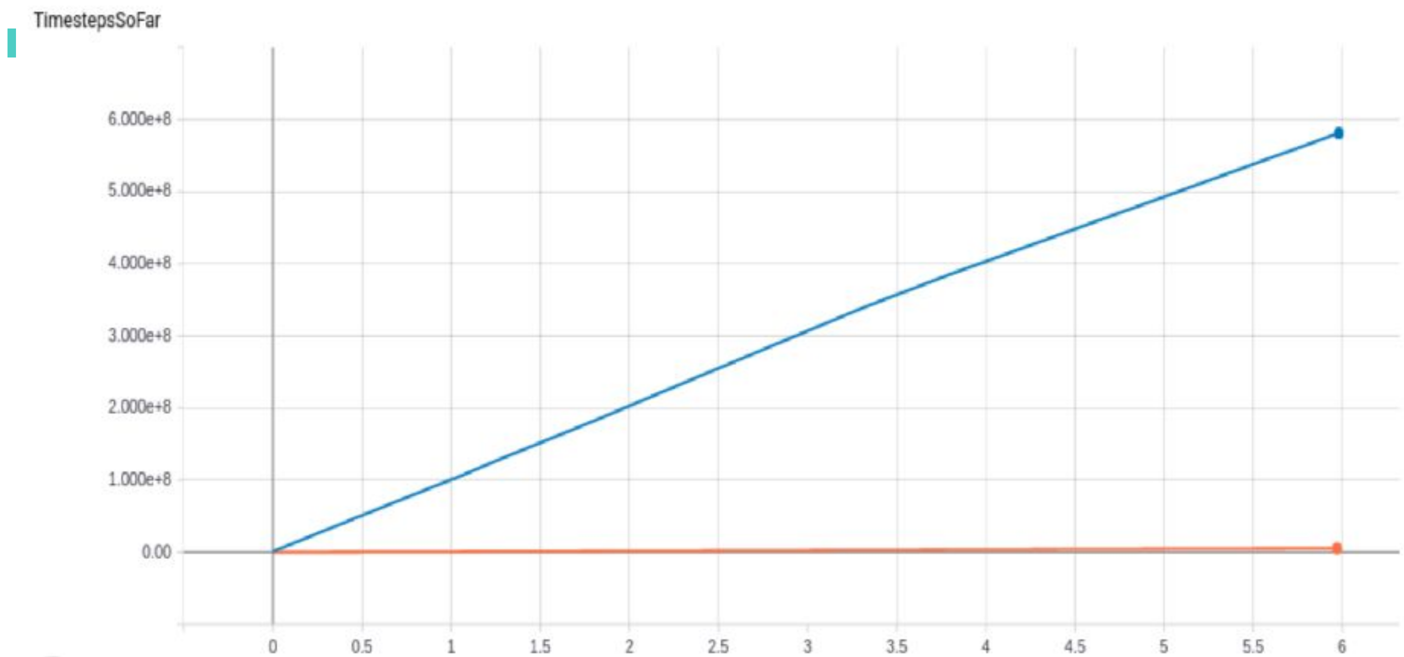
Methodology

Monitoring by Tensorboard



Results

Distributed Training

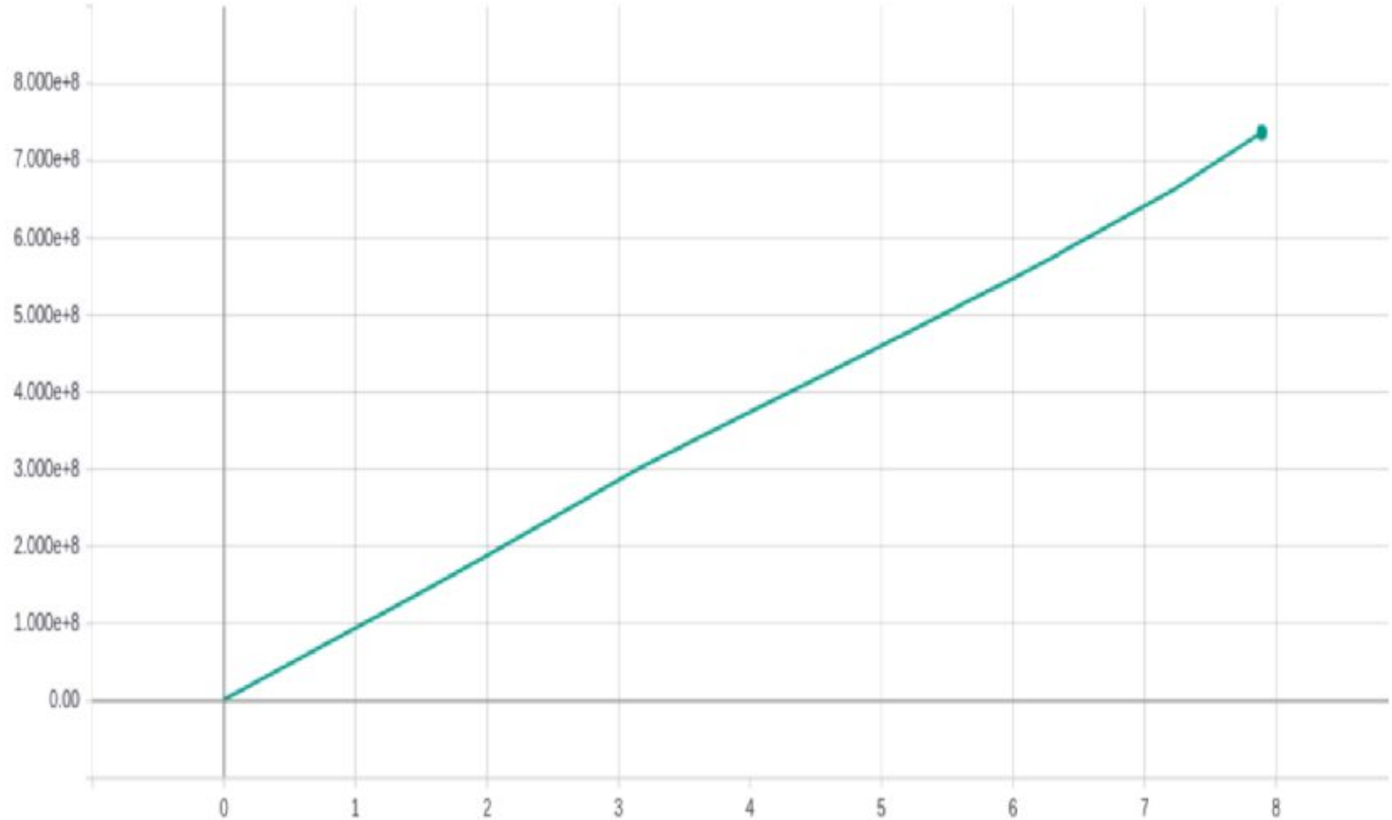


Value	Step	Time	Relative
5.8126e+8	1.314k	Thu Oct 25, 15:39:12	5h 58m 47s
5.2756e+6	1.288k	Thu Oct 25, 07:00:38	5h 58m 9s

$$SpeedUp \approx \frac{5.81 * 10^8}{5.27 * 10^6} \approx \mathbf{110}.$$

Results

Distributed Training



Value	Step	Time	Relative
7.3786e+8	1.668k	Sun Aug 26, 00:33:37	7h 54m 14s

Results

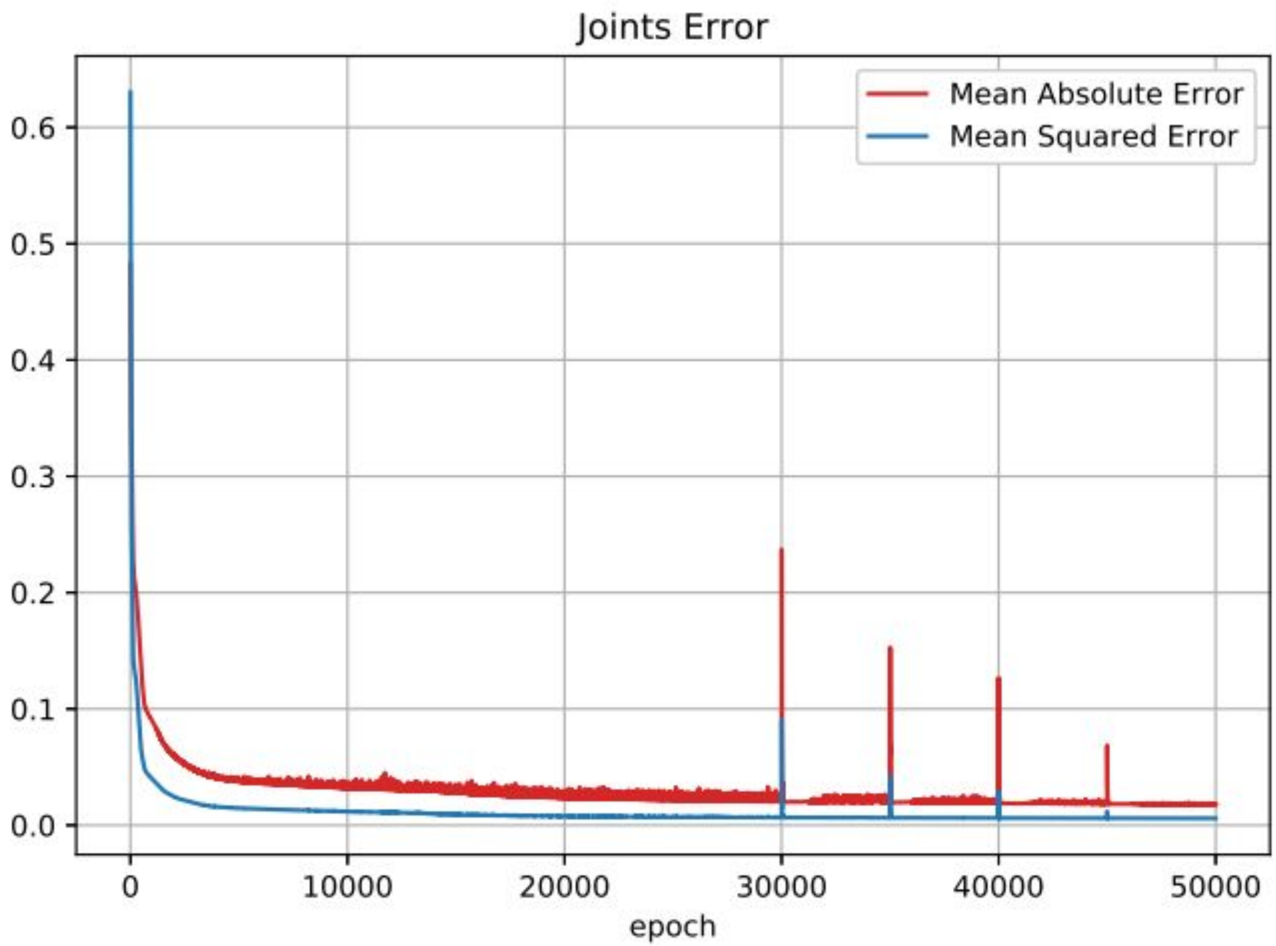
Distributed Training

- **740 millions of samples within 8 training hours**
  - 92.5 millions of samples per hour
  - 21,4 days of training in real-time per hour of simulation

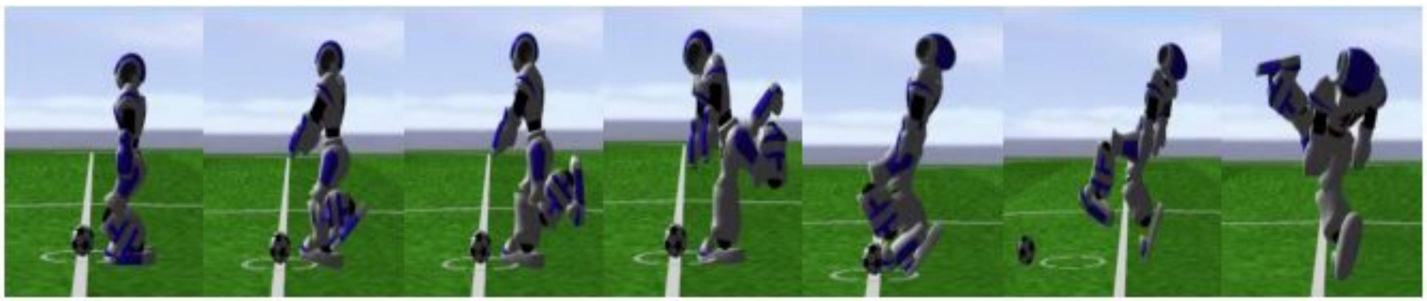
- **171 days of uninterrupted training in real-time.**

Results

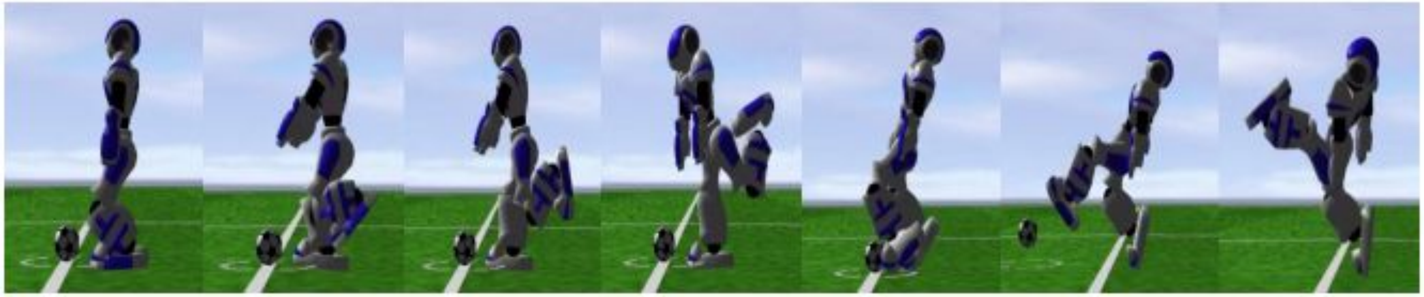
Results - Supervised Training



Results
Learned Kick

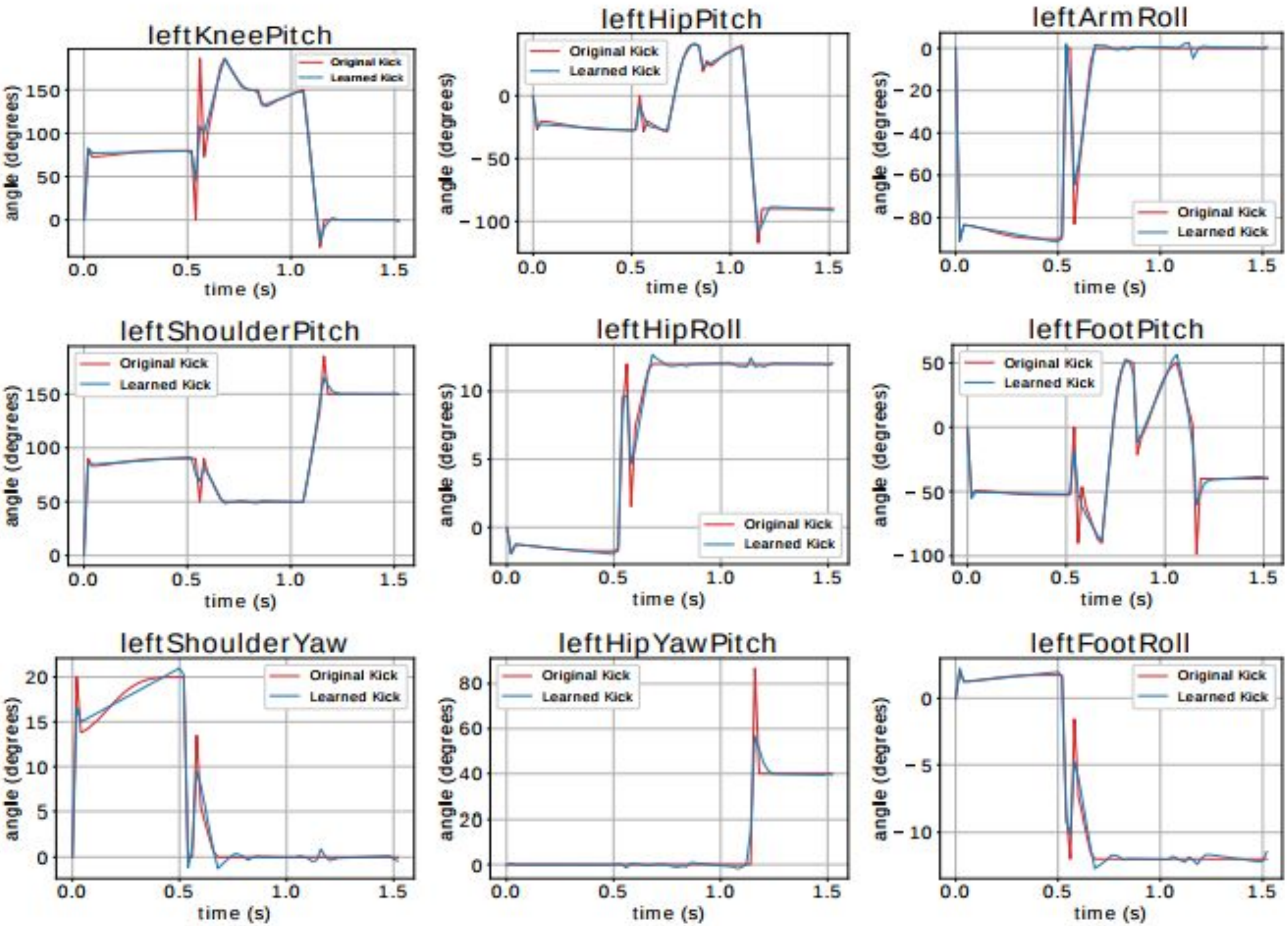


Keyframe



Neural Network

Results



leftKneePitch
leftHipPitch
leftArmRoll
leftShoulderPitch
leftHipRoll
leftFootPitch
leftShoulderYaw
leftHipYawPitch
leftFootRoll
Original Kick
Learned Kick
angle (degrees)
time (s)

Results

TABLE 5.1 – The Kick Comparison

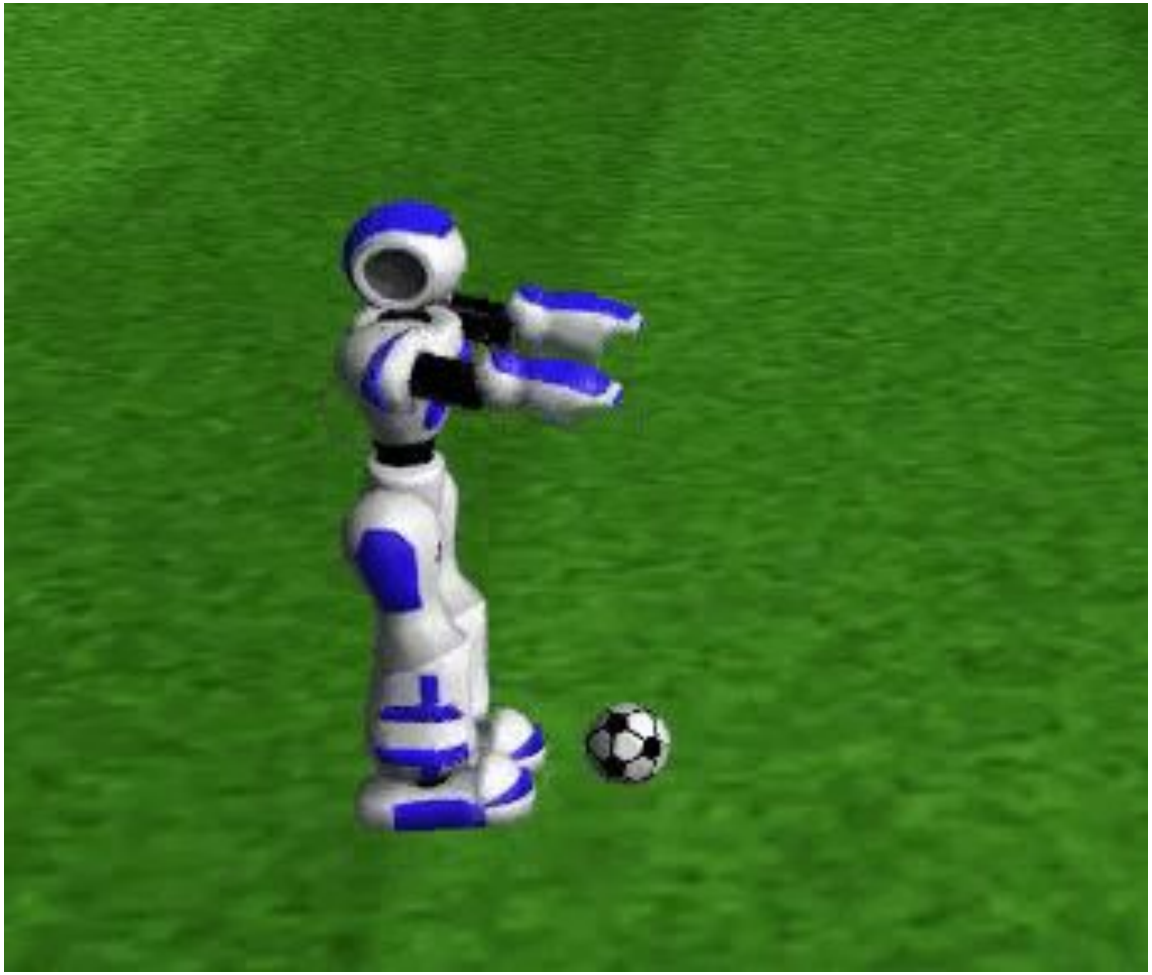
Kick Type	Statistics		
	Accuracy (%)	**Distance (m)**	
		Mean	***Std***
Original Kick	64.5	8.92	3.82
Neural Kick	52.6	7.16	4.06

- **Bonus: It is possible to mimic motion from opponent teams!**

Results

Results

Random policy



Results

RNR

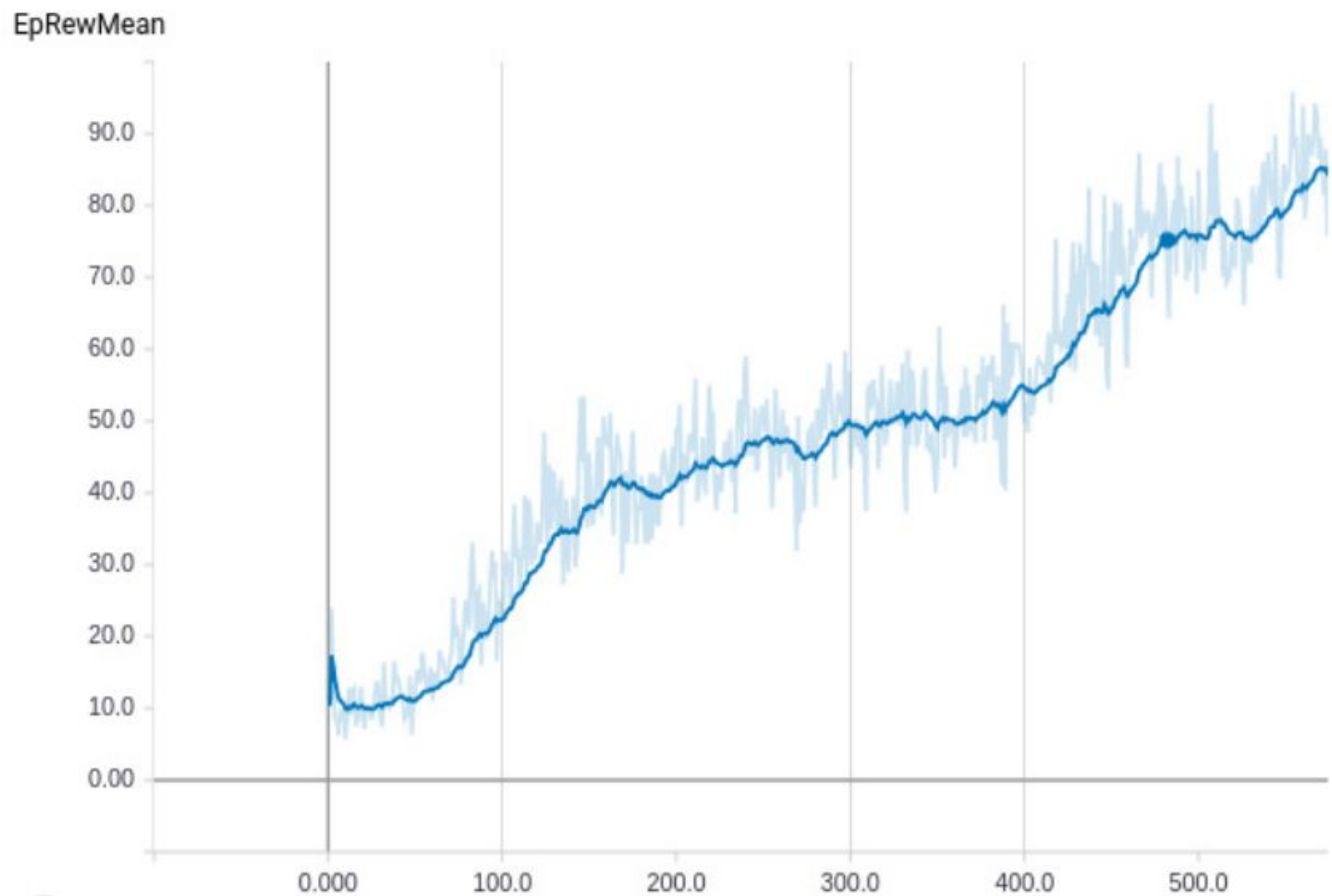


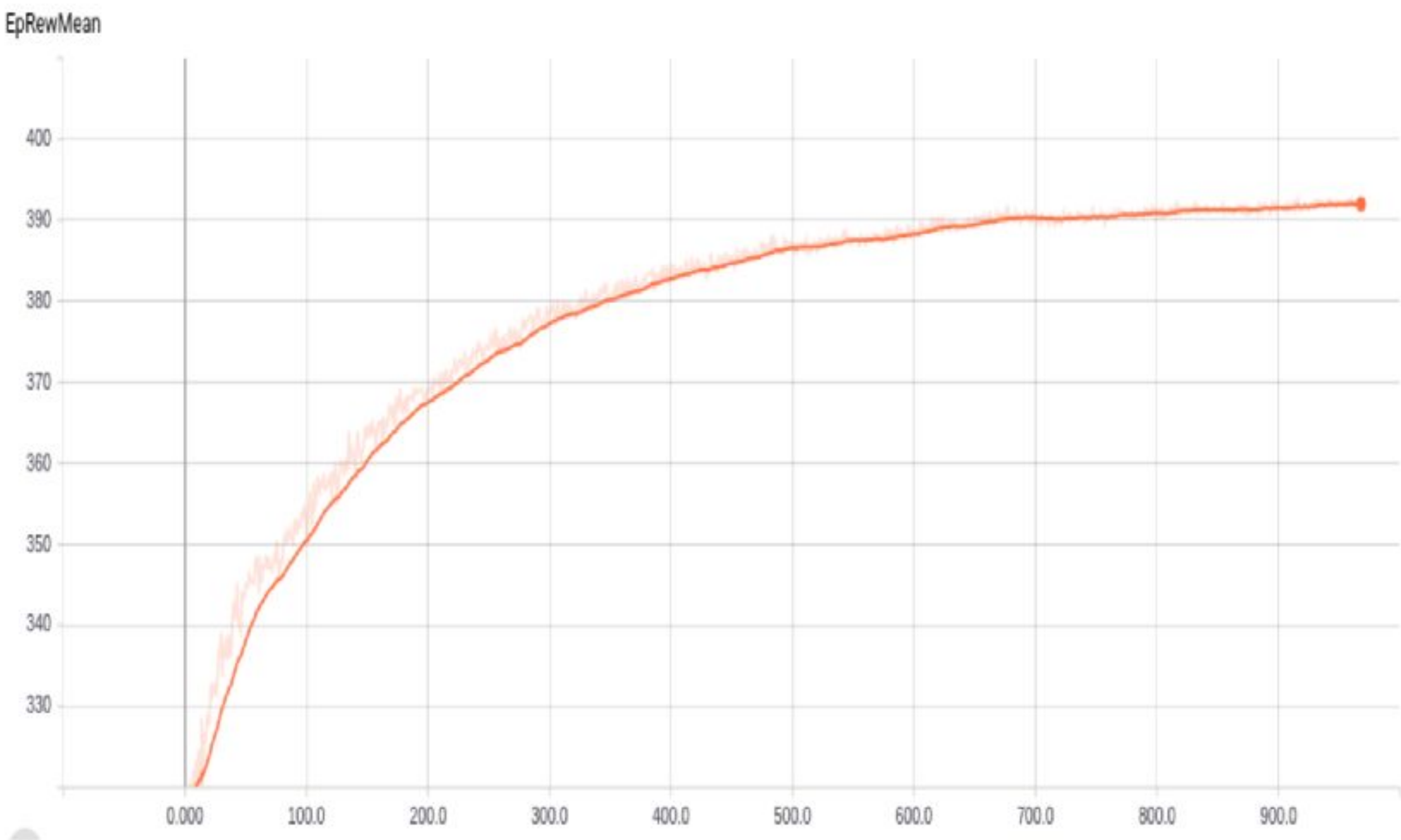
FIGURE 6.3 – RNR Reward Curve by learning update

Results

RNR

Results

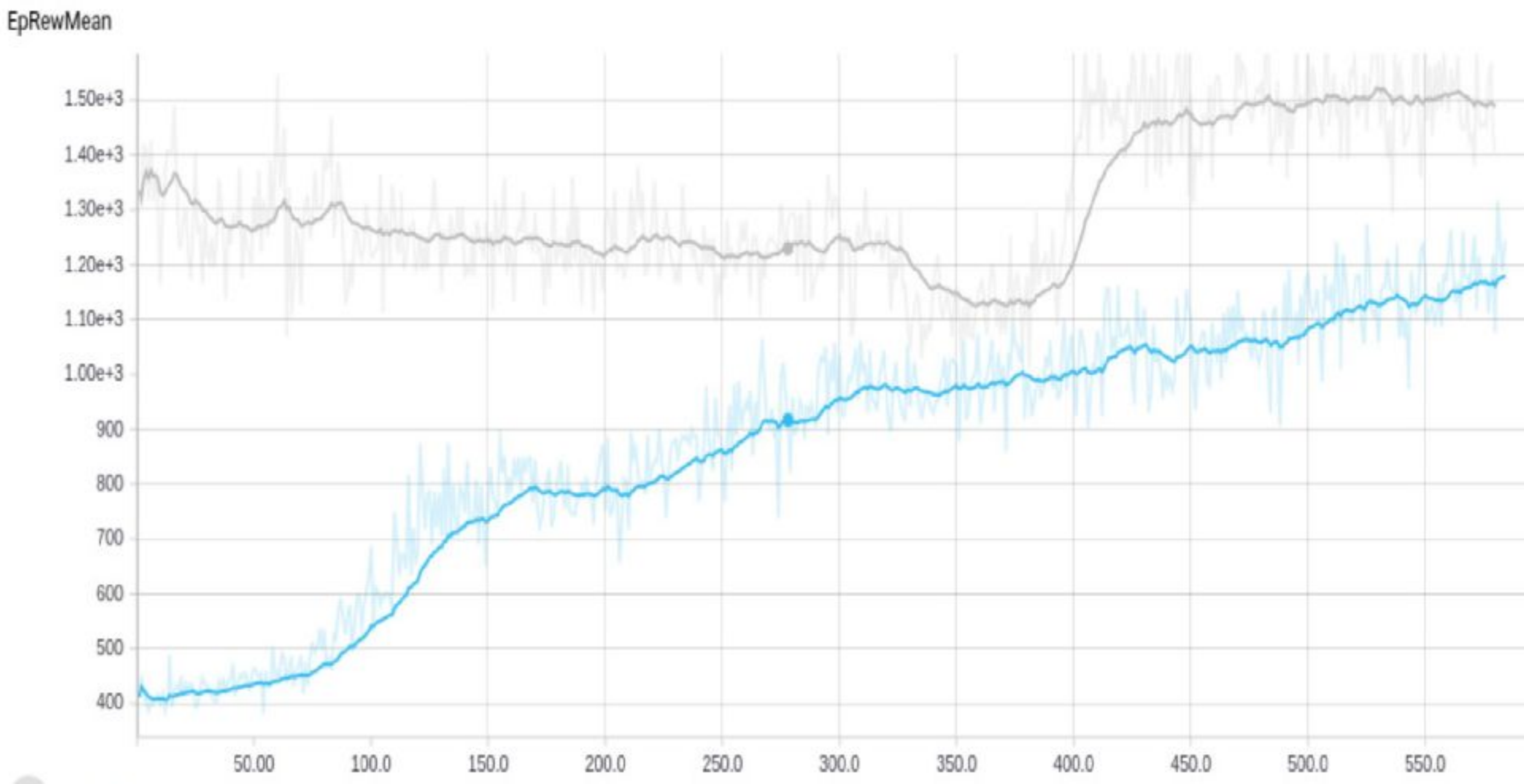
RRR



EpRewMean
400
390
380
370
360
350
340
330
0.000
100.0
200.0
300.0
400.0
500.0
600.0
700.0
800.0
900.0

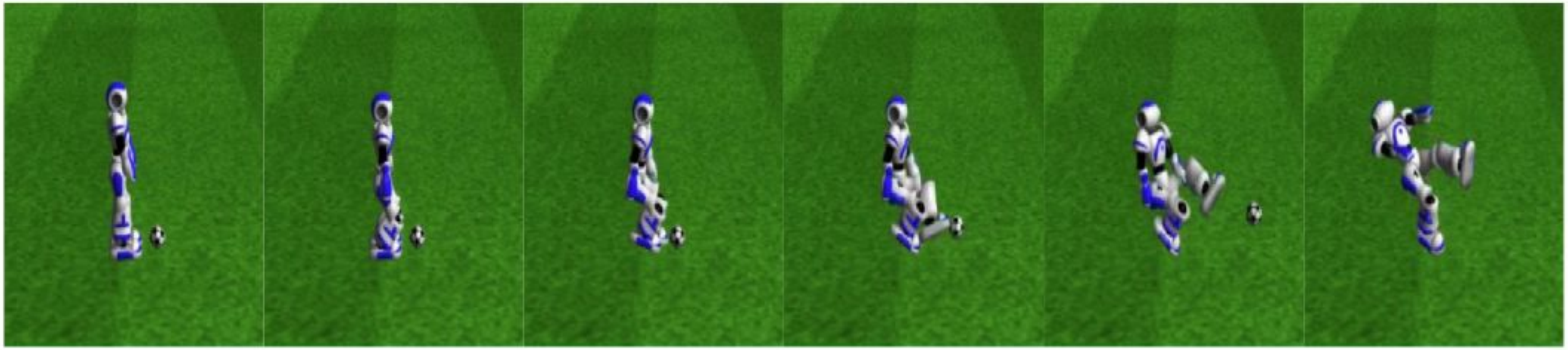
Results

RNR + RRR



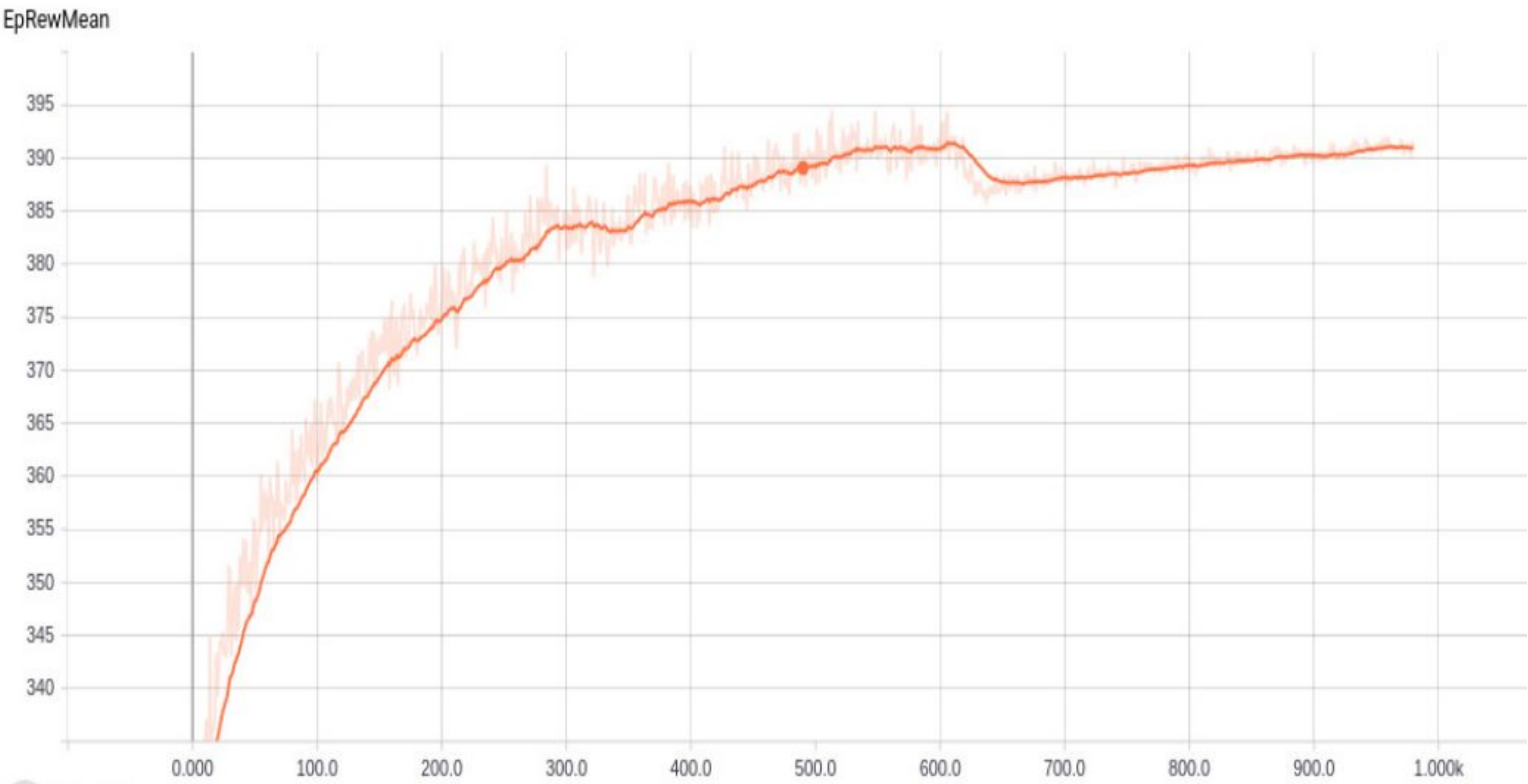
Results

RNR



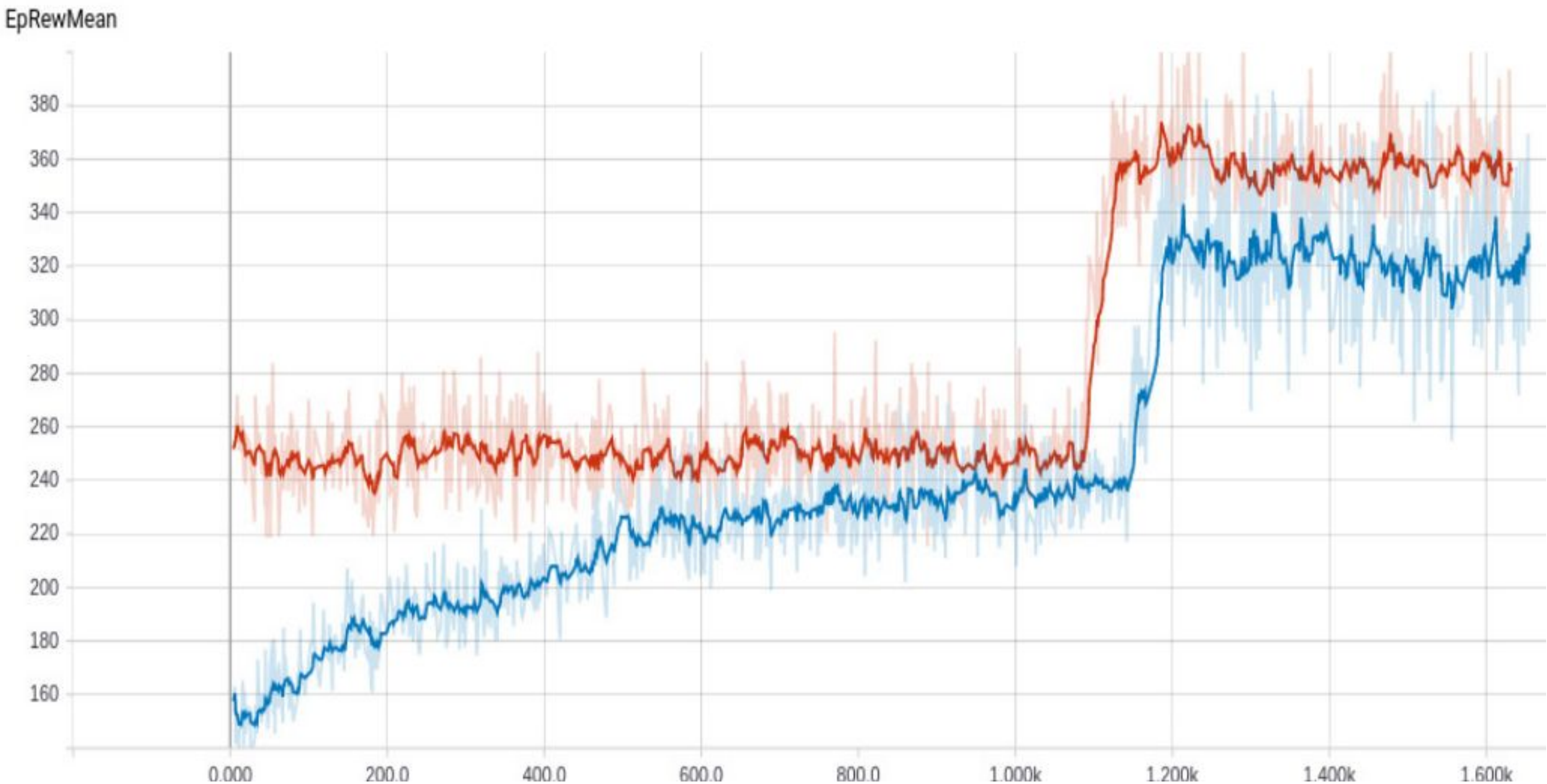
Results

RNR + RRR - Unbalanced



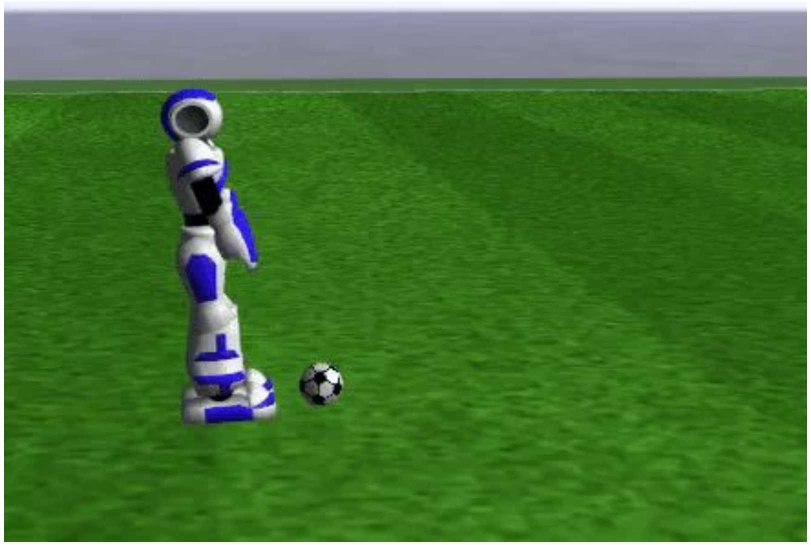
Results

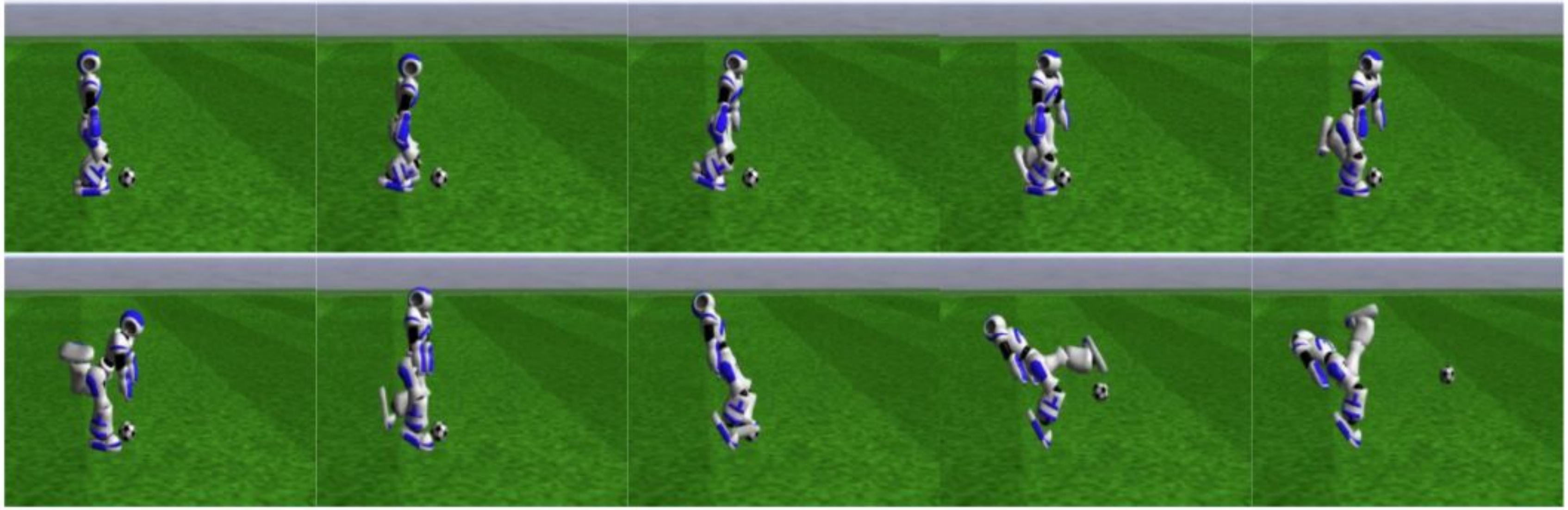
RNR + RRR + RISD



Results

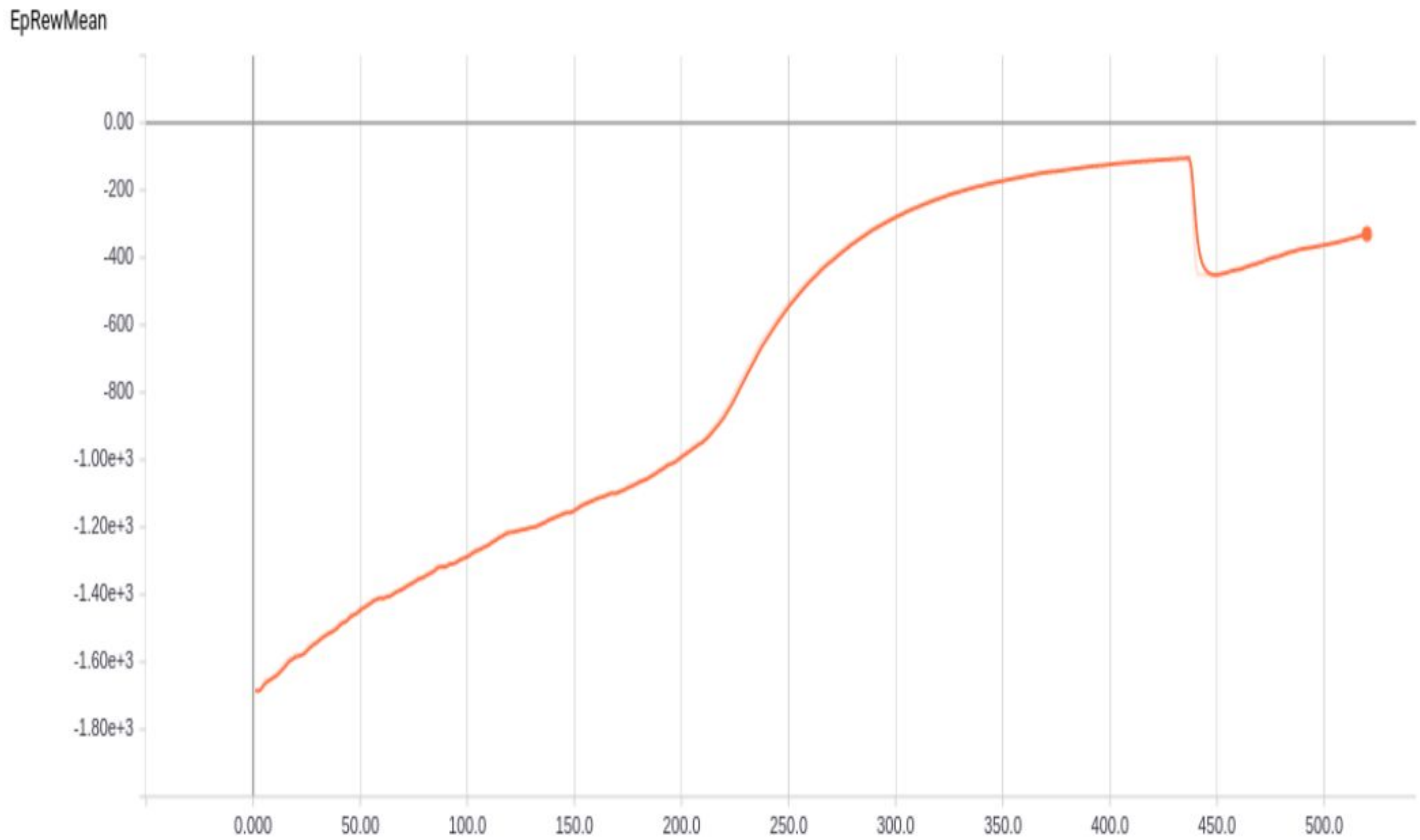
RNR + RRR + RISD





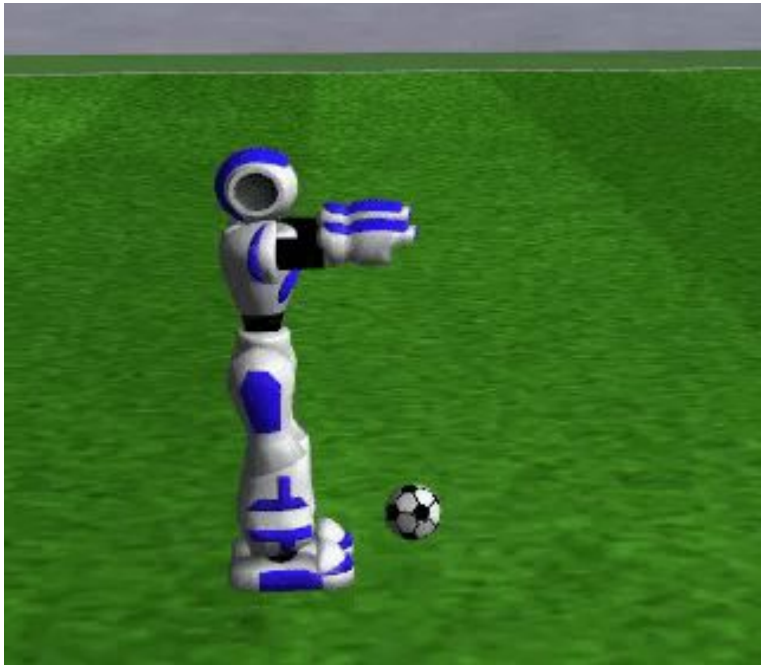
Resultados

“Supervised” Reinforcement



Results

Results - “Supervised” Reinforcement

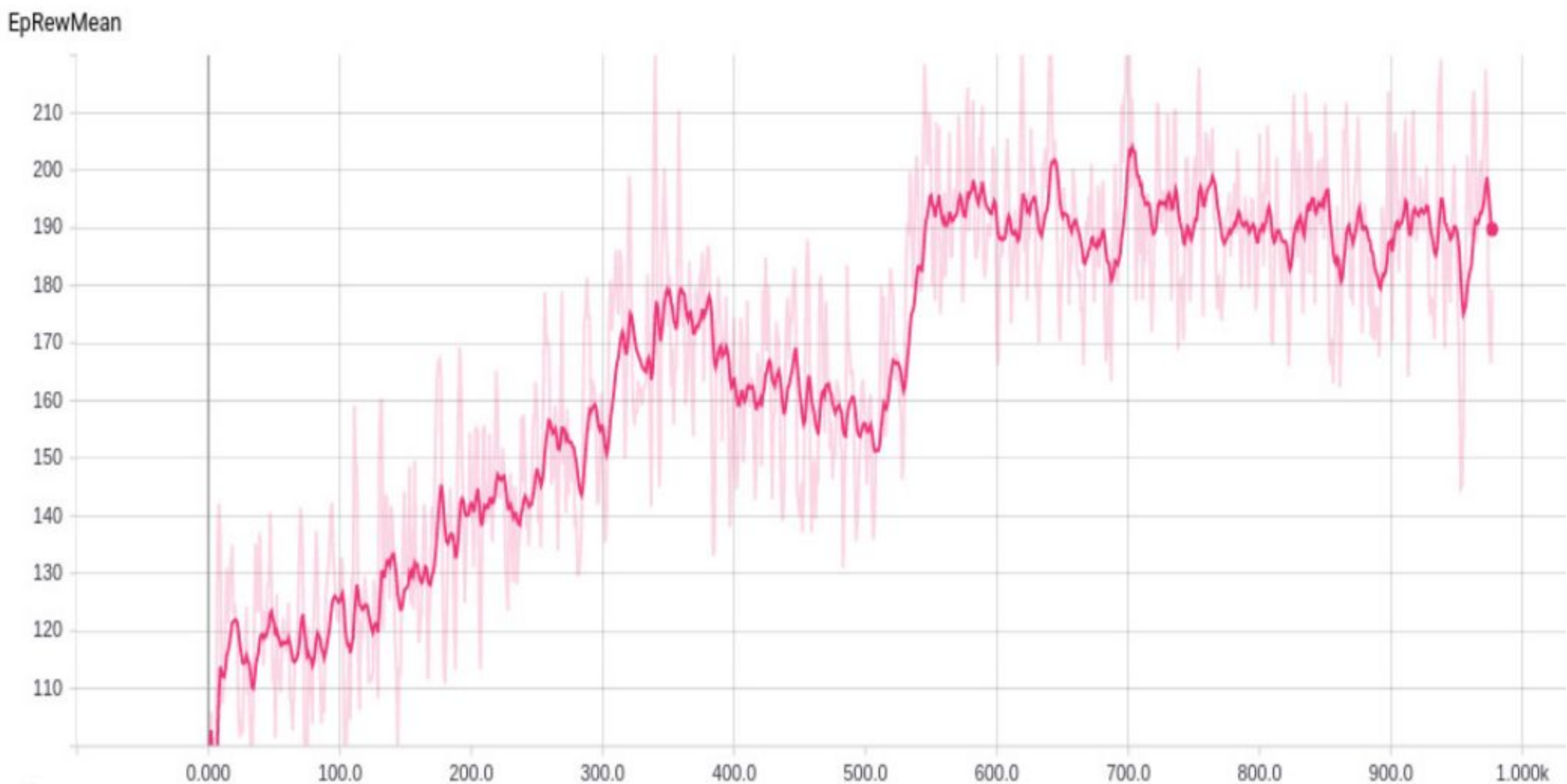


Other ideas for pure RL

- Consider the center of pressure from support foot on the ground; the more centralized, the more stable the kick should become;
- Consider the curve that the kick foot does in relation to the torso during the reference motion; and
- Consider torso’s coordinates from the reference motion.

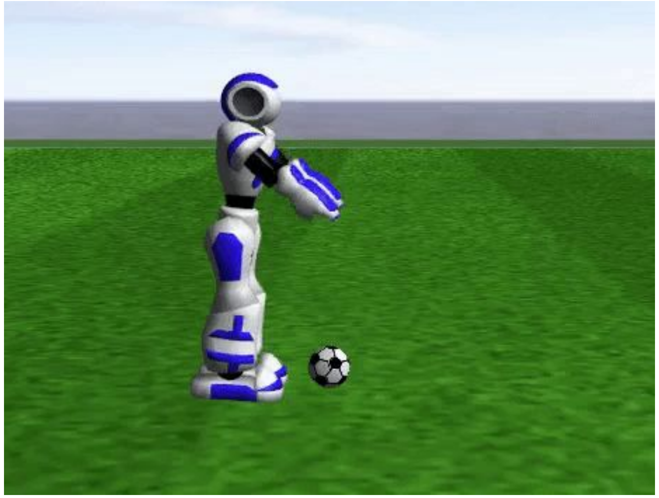
Results

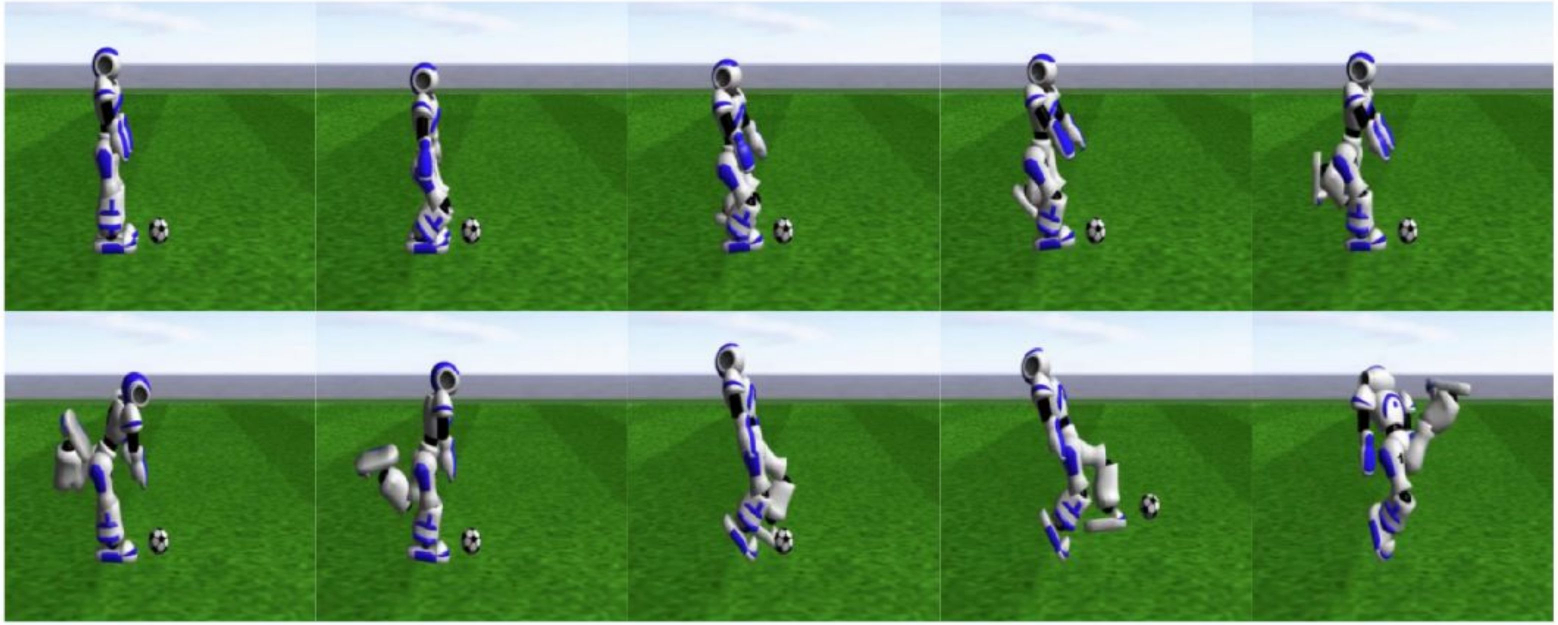
HLM + RNR



Results

HLM + RNR



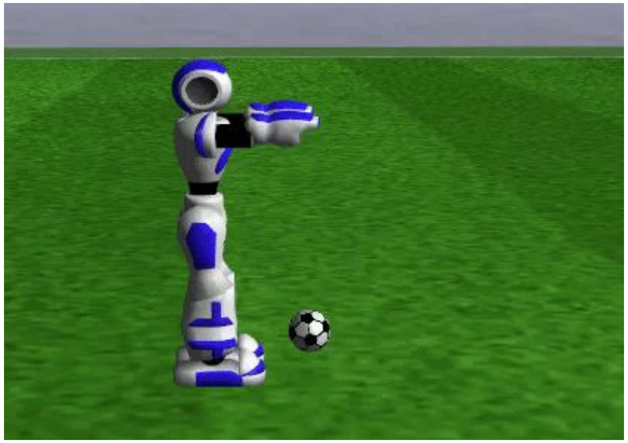


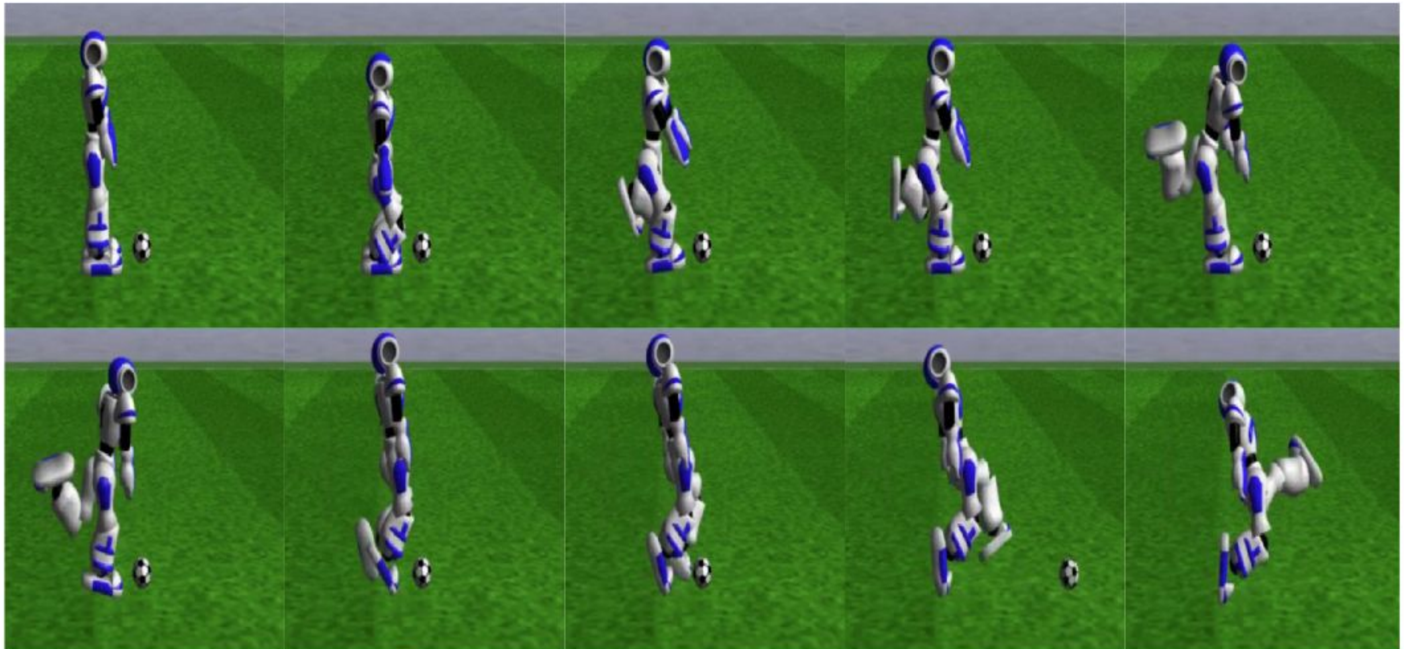
Resultados

HLM + RNR + RET - Session 1

Results

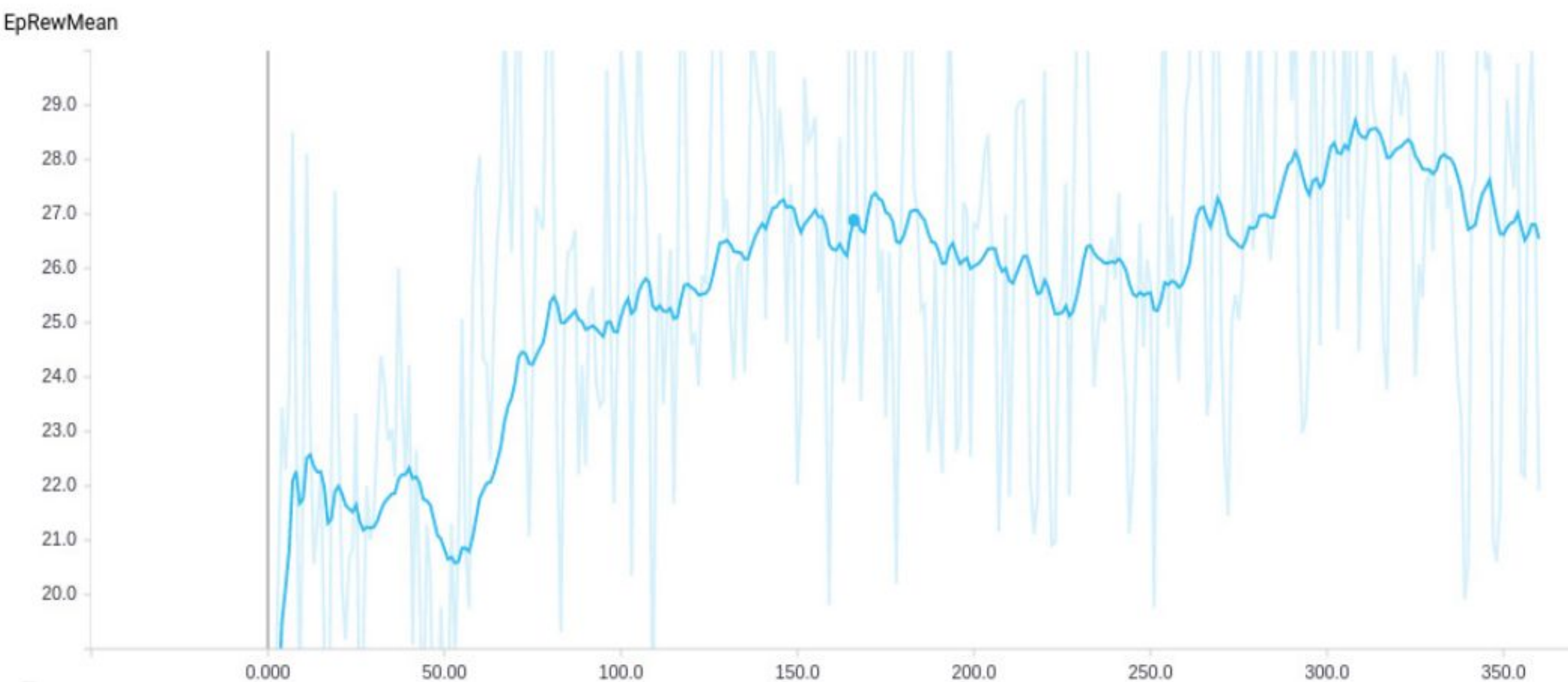
HLM + RNR + RET - Session 1





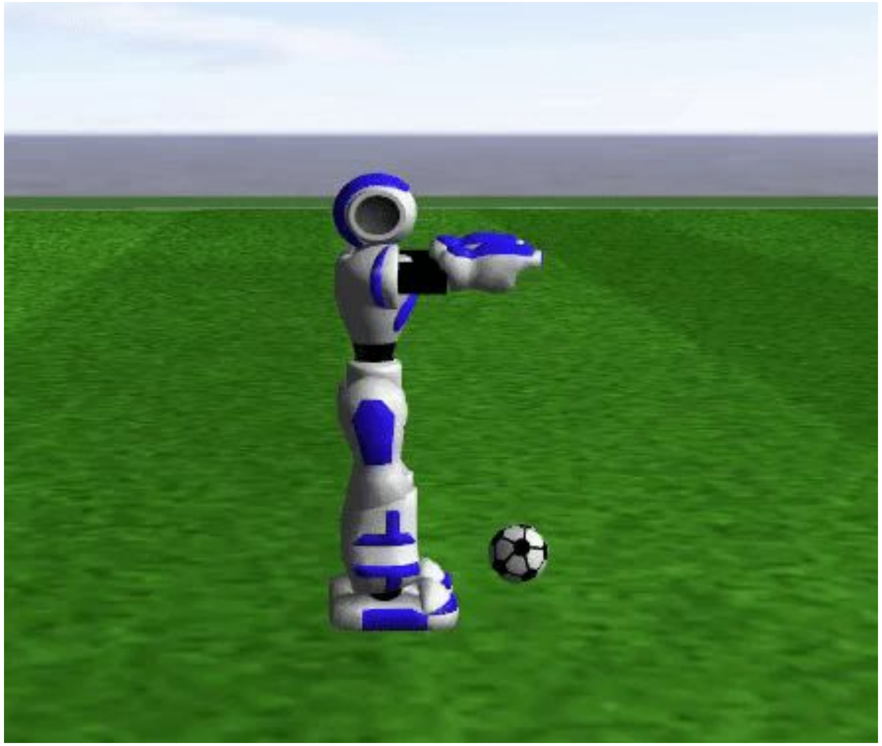
Results

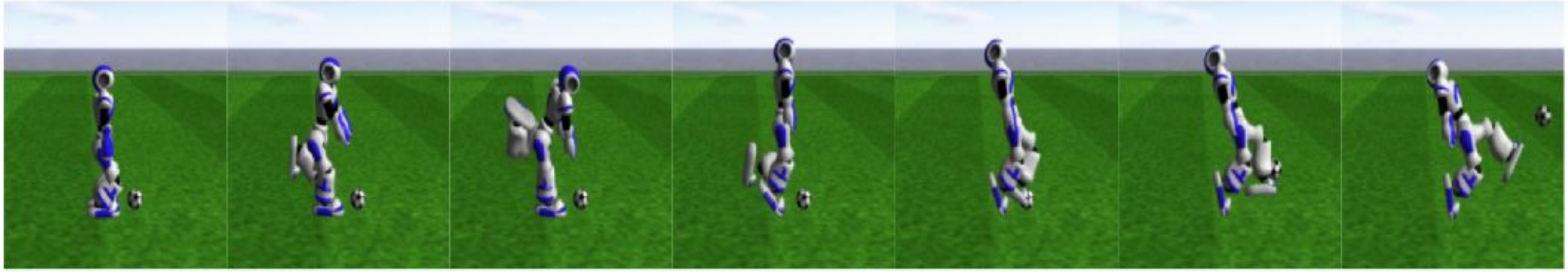
HLM + RNR + RET - Session 2



Results

HLM + RNR + RET - Session 2





Results

Numeric Results

TABLE 6.3 – Kick Comparison - General Evaluation

Kick Type	Statistics				
	Accuracy (%)	Distance X(m)		Distance Z (m)	
		Mean	*Std*	*Mean*	*Std*
Original Kick	69.0	6.27	5.03	0.16	0.41
Learned Kick	63.0	3.06	4.22	0.09	0.17
HLM+RNR Kick	92.0	6.52	3.89	0.33	0.55
HLM+RNR+RET Kick	**92.0**	**7.60**	**3.71**	**0.45**	**0.49**

TABLE 6.4 – Kick Comparison - Effective Evaluation

Kick Type	Statistics			
	Distance X(m)		Distance Z (m)	
	Mean	*Std*	*Mean*	*Std*
Original Kick	**9.05**	**3.44**	0.21	0.49
Learned Kick	4.82	4.46	0.12	0.21
HLM+RNR Kick	7.07	3.55	0.36	0.57
HLM+RNR+RET Kick	**8.26**	**3.09**	**0.48**	**0.49**

Results

Bonus: Nao with Toe

TABLE 6.4 – Kick Comparison - General Evaluation

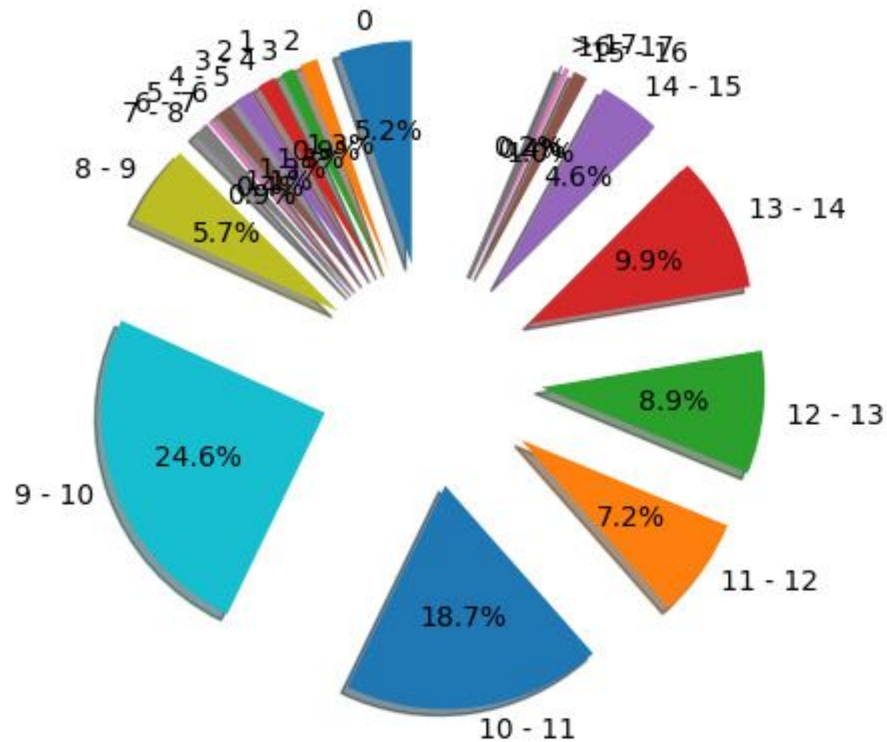
Kick Type	**Statistics**				
	Accuracy (%)	**Distance X(m)**		**Distance Z (m)**	
		Mean	***Std***	***Mean***	***Std***
Original Kick	69.0	6.27	5.03	0.16	0.41
Learned Kick	63.0	3.06	4.22	0.09	0.17
HLM+RNR Kick	92.0	6.52	3.89	0.33	0.55
HLM+RNR+RET Kick	92.0	7.60	3.71	0.45	0.49
Best Kick Toe	**95.0**	**9.47**	**3.43**	**0.66**	**0.63**

TABLE 6.4 – Kick Comparison - Effective Evaluation

Kick Type	**Statistics**			
	Distance X(m)		**Distance Z (m)**	
	Mean	***Std***	***Mean***	***Std***
Original Kick	**9.05**	**3.44**	0.21	0.49
Learned Kick	4.82	4.46	0.12	0.21
HLM+RNR Kick	7.07	3.55	0.36	0.57
HLM+RNR+RET Kick	8.26	3.09	0.48	0.49
Best Kick Toe	**9.96**	**2.75**	**0.69**	**0.63**

Results

Bônus: Nao with Toe



Conclusions

- It is possible to transfer the knowledge from a keyframe motion to a neural network with a minor residual error;
- It is possible to optimize this neural network to perform better a objective (in this case, humanoid kick motion); and
- Pure RL technique lead to suboptimal policies.

Future Work

- Replicate the methodology from this work in other types of keyframe motion;
- Apply this learning framework in humanoid robot walk;
- Policy Optimization through reference motion improvement;
- Derive theoretically the relation between RL, SL and “Supervised” Reinforcement;
- Explore Intel DevCloud hardware; and
- Development of techniques that improve data efficiency and hyperparameter tuning.