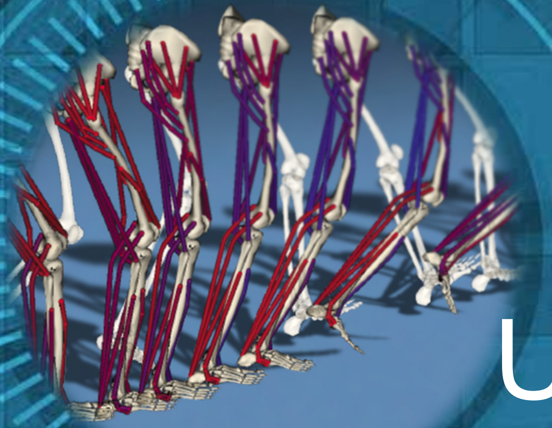


Understanding motion kinematics using prosthetic devices after lower limb amputation through AI-engineered model

Catalina Botia
Isabella Ramos
Daniela Tamayo

Universidad de los Andes
Colombia

Contents

Introduction

Bipedal locomotion

6500 steps

1.3m/s

Introduction

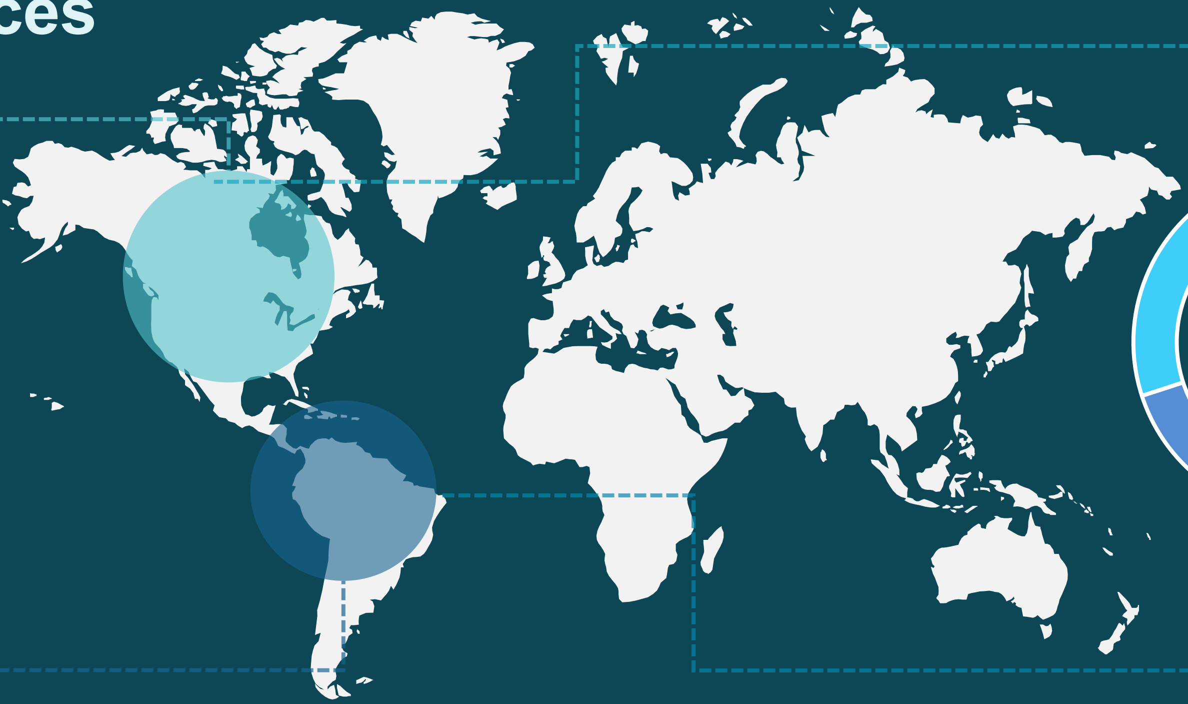
Amputation incidences

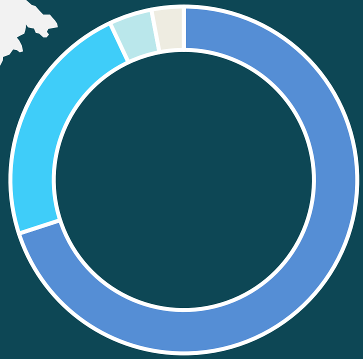
United states

2007: 1.7 million people

Colombia

2006: 200-300 persons for every 100,000





Circulatory disease 70%
Trauma 23%
Tumor 4%
Congenital conditions 3%

90% **Lower limb amputations**

$ **Costs**

Introduction

Artificial Intelligence (AI)

Healthcare system has been benefited by technological advances built through machine learning.

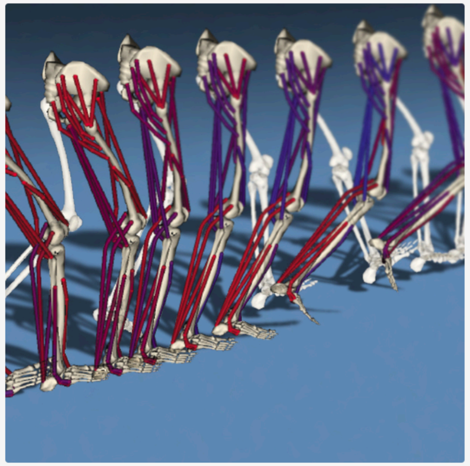
Healthcare system

Automated medical diagnosis, image classification for pathogen identification, recognition of ailments and personalized treatment strategies, health data control, management of patient information, etc.

Reinforcement Learning (RL)

Aims to find the best possible behavior or path to maximize a reward in a specific situation or environment.

Related Work



NeurIPS 2018: AI for Prosthetics Challenge

Reinforcement learning with musculoskeletal models

Stanford Neuromuscular Biomechanics Laboratory | crowdAI | EPFL Digital Epidemiology Lab

Completed

73717 Views | 477 Participants | 4575 Submissions

173 | FOLLOW

CONTINUOUS CONTROL WITH DEEP REINFORCEMENT LEARNING

Timothy P. Lillicrap*, Jonathan J. Hunt*, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver & Daan Wierstra
Google Deepmind
London, UK

[1]

DDPG

+

ACE

Table 2: Performance of ACE

Experiment	# Test	# Actor	# Critic	Average reward	Max reward	# Fall off
A1C0	100	1	0	32.0789	41.4203	25
A10C1	100	10	1	37.7578	41.4445	7
A10C10	100	10	10	39.2579	41.9507	4

[2]

Improved performance, reduced dooming actions

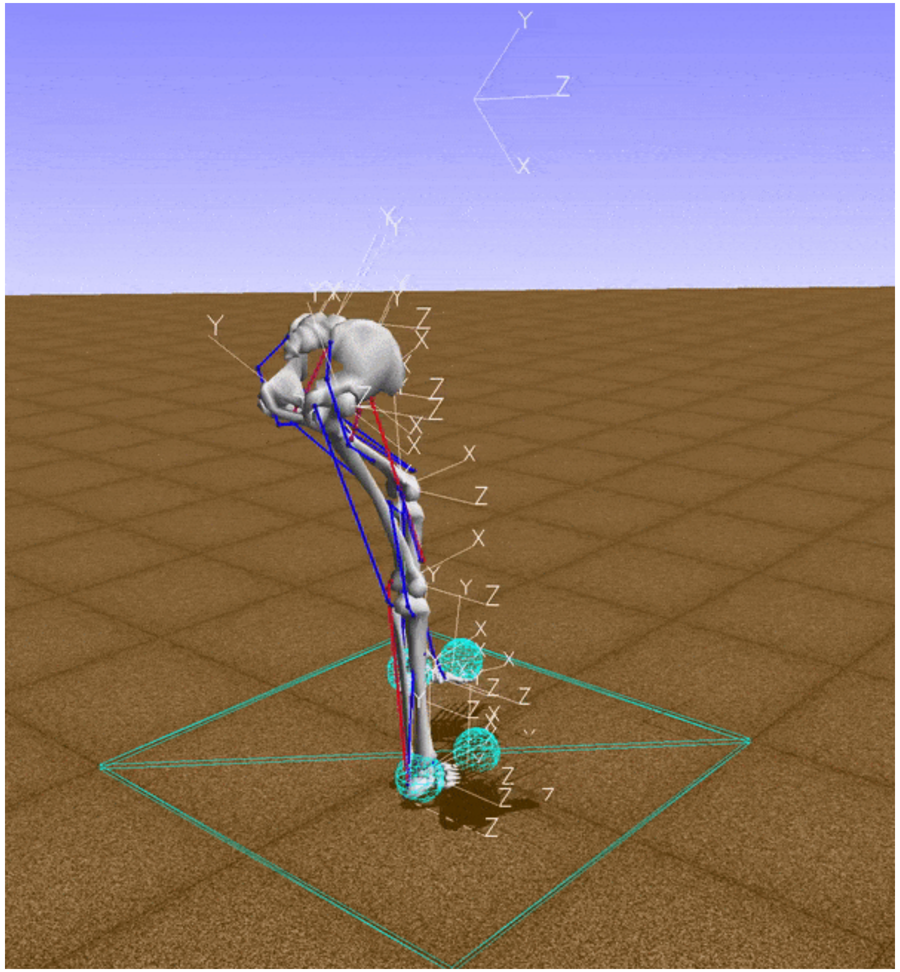
1. T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous con- trol with deep reinforcement learning. arXiv preprint arXiv:1509.02971, 2015.
2. Ł. Kidzinski, S. P. Mohanty, C. F. Ong, Z. Huang, S. Zhou, A. Pechenko, A. Stelmaszczyk, P. Jarosik, M. Pavlov, S. Kolesnikov, et al. Learning to run challenge solutions: Adapting reinforcement learning methods for neuromus- culoskeletal environments. In The NIPS'17 Competition: Building Intelligent Systems, pages 121–153. Springer, 2018.

OpenSim Prosthetics Environment

Dictionary describing the current state observation

19-dimensional vector of muscle excitation and **14** degrees of freedom

A **1000** iterations correspond to **10** seconds in the virtual environment



Reward function:
*R = 9 * s - p * p*

Stop criteria:

1) The pelvis of the model falls below 0.6 meters,
2) 1000 iterations are reached

In the *AI for Prosthetics challenge* all competitors were ranked according to their maximum reward.

Approach

01 ACTOR & CRITIC NETWORKS

02 TARGET NETWORKS

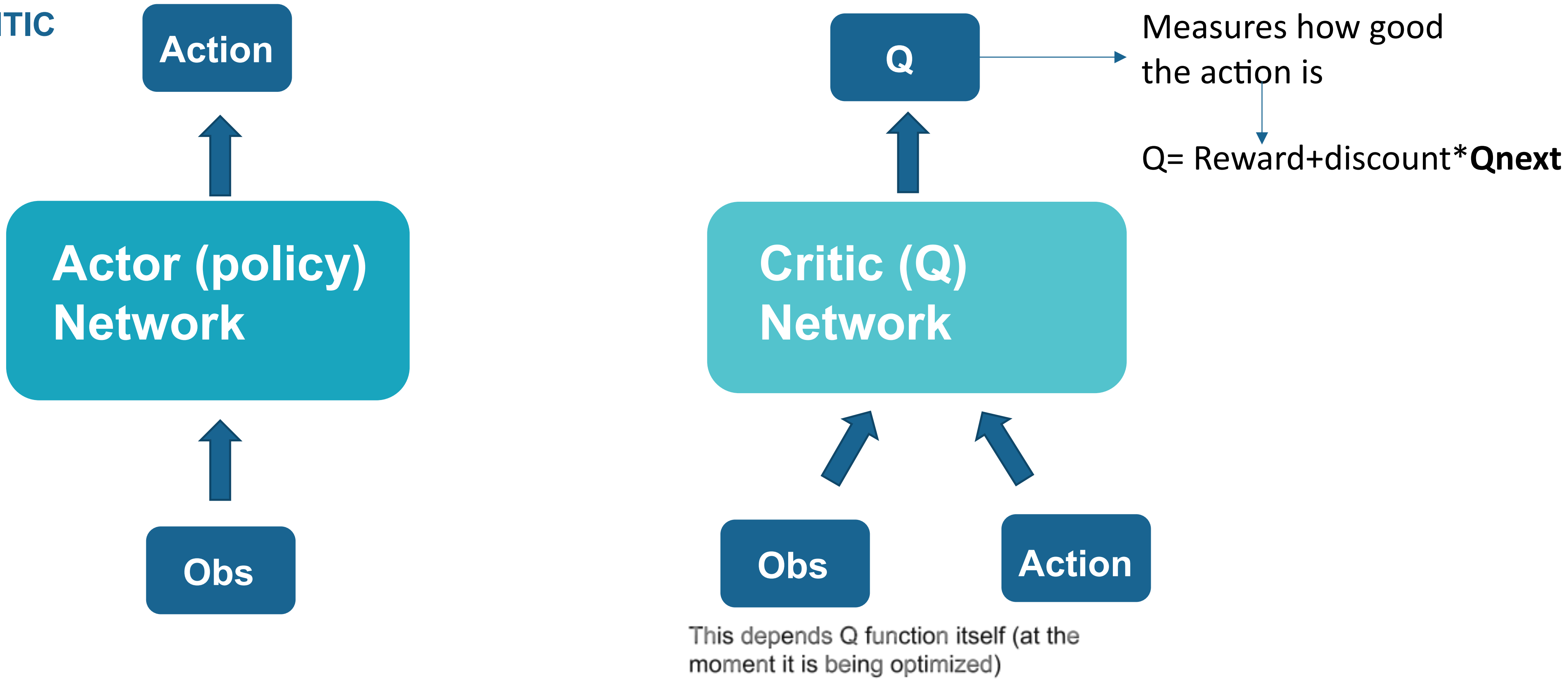
03 EXPERIENCE BUFFER

04 EXPLORATION

Deep Deterministic Policy Gradient

Approach

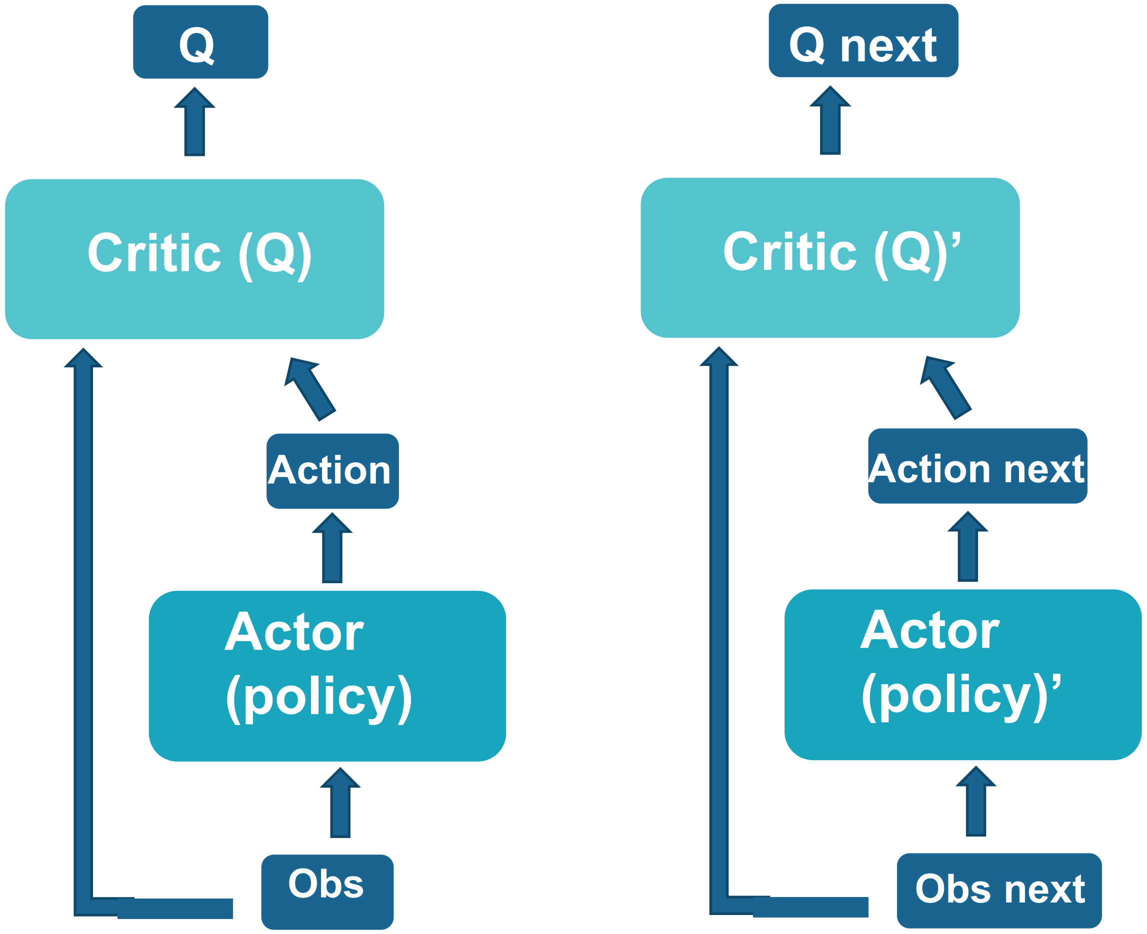
01 ACTOR & CRITIC NETWORKS



$$Q(s,a) \leftarrow Q(s,a) + \alpha[R(s,a) + \gamma \overbrace{max Q(s',a')} - Q(s,a)]$$

Approach

02 TARGET NETWORKS



Time delayed copies of original networks

Next-state Q values are calculated with the target networks

Minimize the loss between the updated Q value and the original Q value

Improve stability

Approach

03 EXPERIENCE BUFFER

Sample experience

Save all the experience tuples (state, action, reward, next_state)

Finite-sized memory

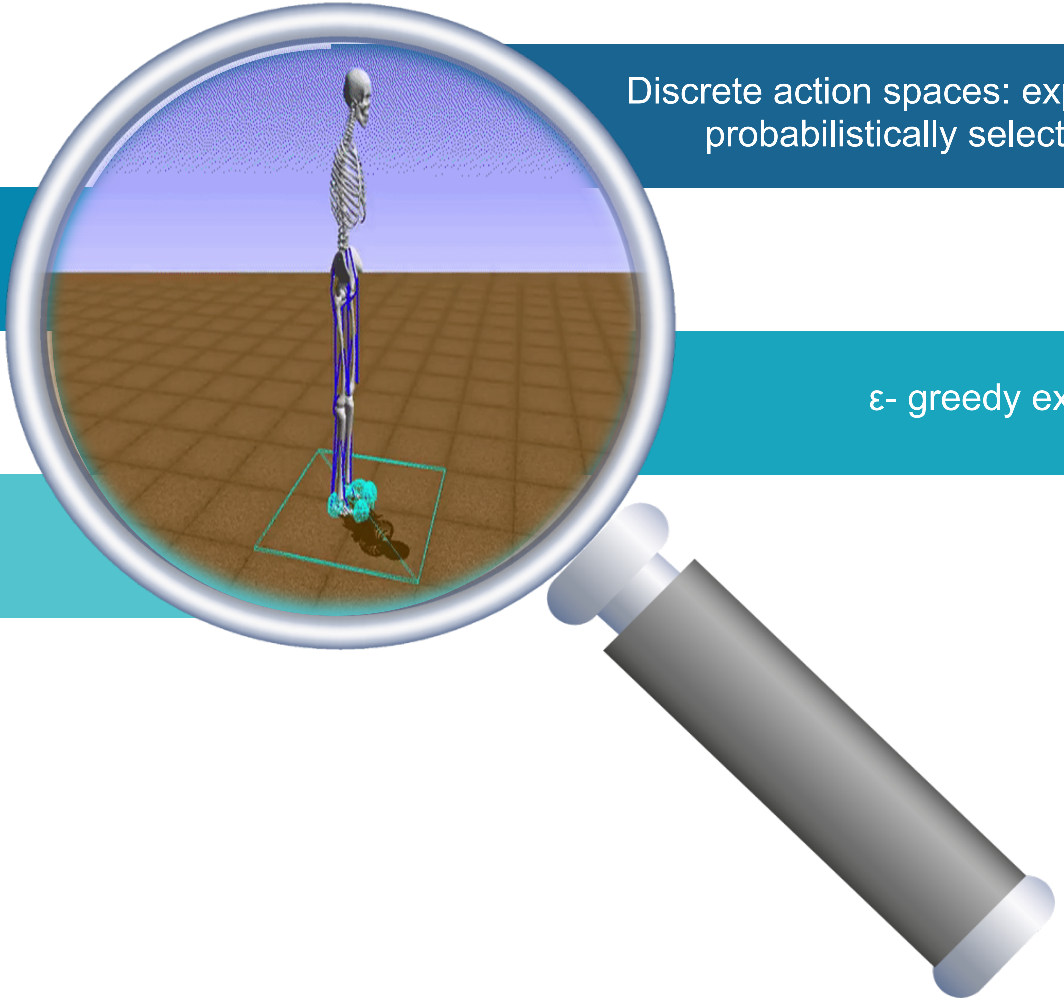
Mini-batches from replay buffer to update

Approach

04 EXPLORATION

Continuous action spaces: exploration is done via adding noise to the action itself

Ornstein-Uhlenbeck Process (OU)



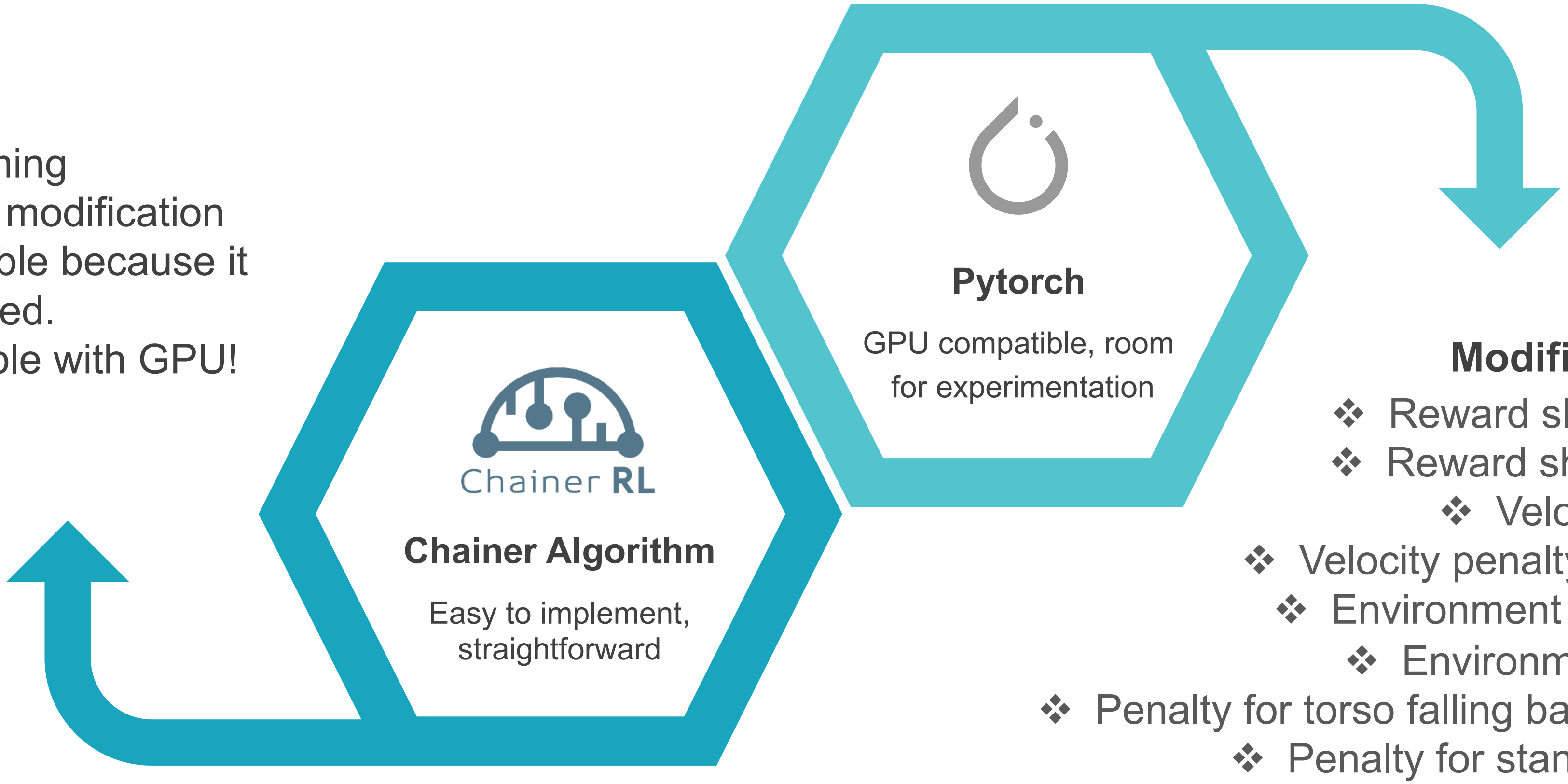
Discrete action spaces: exploration is done via probabilistically selecting a random action

ε- greedy exploration

DDPG Baseline

Problems

❖Time consuming
❖A-C network modification was not posible because it was predefined.
❖Not compatible with GPU!



Modifications

❖ Reward shaping A
❖ Reward shaping B
❖ Velocity loss
❖ Velocity penalty/reward
❖ Environment wrapper
❖ Environment step
❖ Penalty for torso falling backwards
❖ Penalty for standing still

Chainer Experiments

Reward Shaping Function

Penalties for falling and not following the expected velocity.

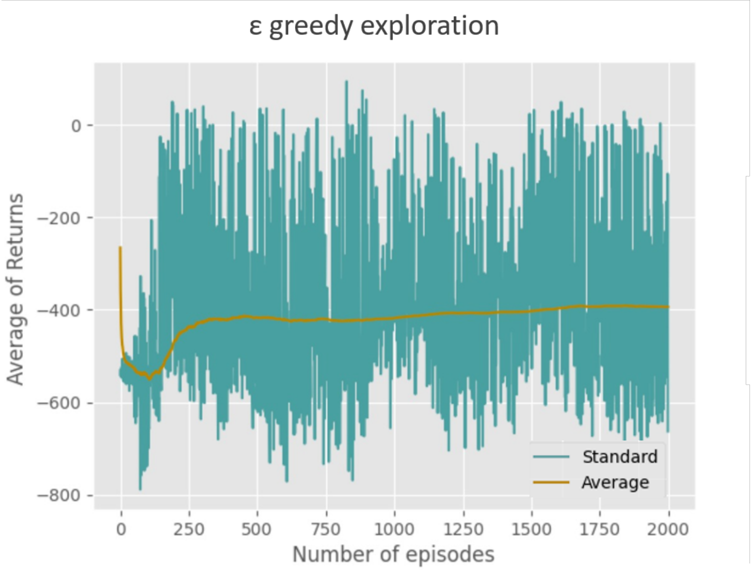
ε-Greedy exploration & Ornstein-Uhlenbeck Process

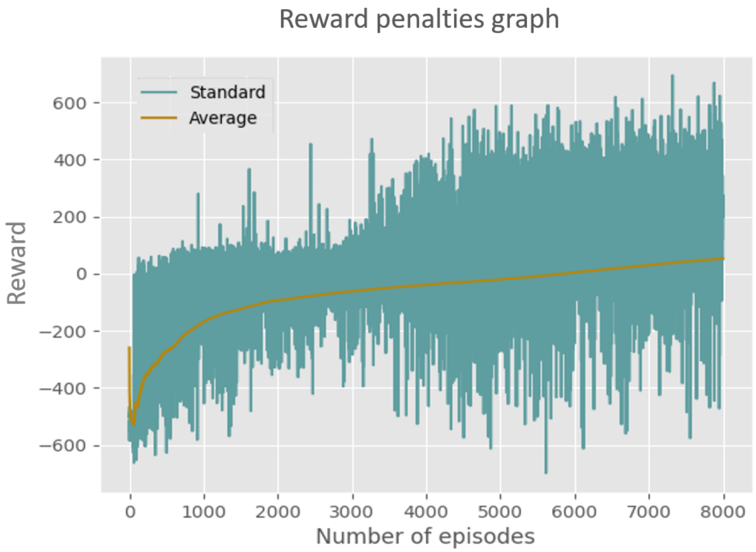
Compare differences using each process.

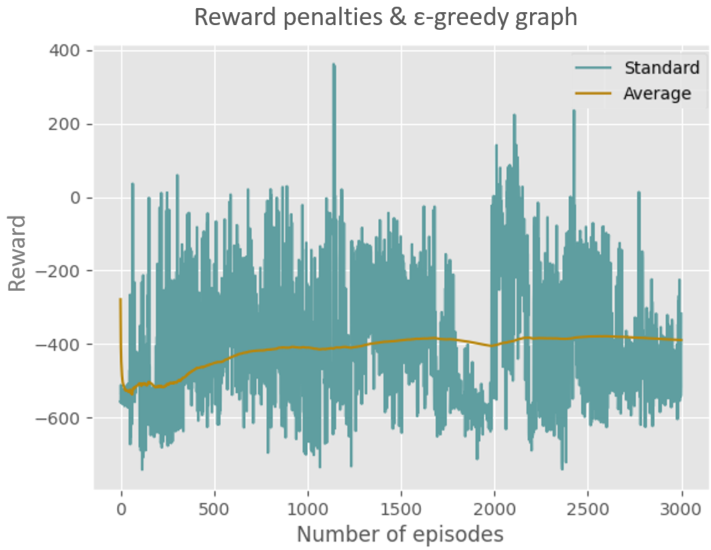
Chainer RL

Chainer Results

Experiment	Episodes	Step size	Max memory	Test best reward	Max reward	Av. reward	Time (hours)
ε greedy	2000	500	50000	-513.23	101.53	-401.20	16
Reward Shaping + OU	8000	500	50000	-426.71	693.65	52.51	65.5
Reward shaping + ε greedy	3000	500	50000	-453.23	30.78	-362.24	24.5

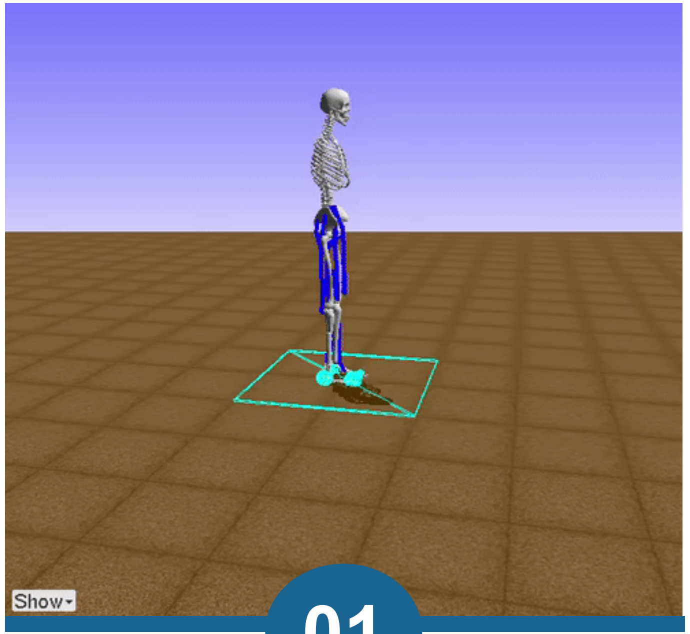






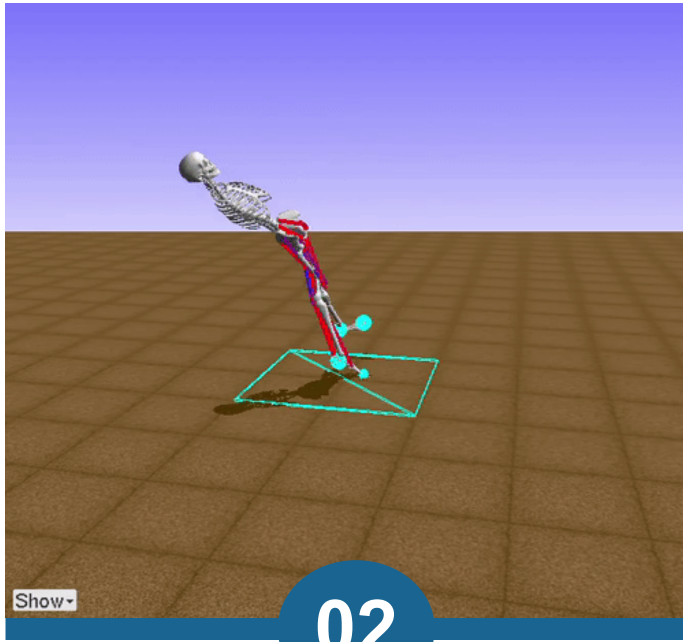
Chainer RL

Visualization



Reward Shaping + OU

8000 episodes



Epsilon explorer

3000 episodes

Pytorch Experiments

Reward Shaping Function

Two types of reward function. (A) Penalties for falling and not following the expected velocity. (B) Penalties for crossing legs, deviation from walking forward, taking steps to the sides and bonus for bending knees.

Extra Penalties

Penalty for torso falling backward and penalty for standing still.

Target Velocity

Bonus/Penalty for reaching target velocity and velocity loss incorporation to DDPG.

Environment Wrapper

Two types of environment wrapper used to endow the observation state with more information.

Pytorch Results

	ID	Episodes	Batch size	Step size	Target vel.	Max memory	Max Test R	Max R	Average R	Time (h)
Reward Shaping	6	9500	600	600	None	50000	-26.17	54.75	-524.35	59
	7	8000	600	600	None	50000	18.82	63.89	-287.95	61
Target Loss	8	10000	400	500	3	50000	-267.09	30.78	-557.02	57
	9	8500	300	500	6	50000	-87.53	36.50	-383.25	52
	10	7500	400	500	3	50000	-403.45	-187.39	-742.22	35
	11	8500	300	1000	None	50000	-44.65	23.48	-58.04	46
	12	6841	50	1000	None	50000	**36.76**	60.60	-115.41	33.2
Env W + Penalties	13	12715	200	1000	None	50000	-202.63	77.39	-467.16	61.6
	14	6498	200	8000	None	50000	-412.78	73.02	-469.20	35.3
	15	10252	200	1000	None	500000	-426.65	78.49	-459.15	70.6
	16	5200	200	8000	None	500000	-470.56	75.23	-470.41	31.7
	17	16207	500	1000	None	50000	4.20	68.36	-534.30	93.58
	18	16207	50	1000	None	50000	-502.27	55.92	-522.74	71
Penalties	19	4341	50	1000	None	50000	-202.63	-635.86	-1740.88	26.25
	20	11919	300	1000	None	50000	-412.78	-415.83	-1686.20	60.77
	21	8072	300	1000	None	50000	**43.39**	-190.18	-813.58	57.4

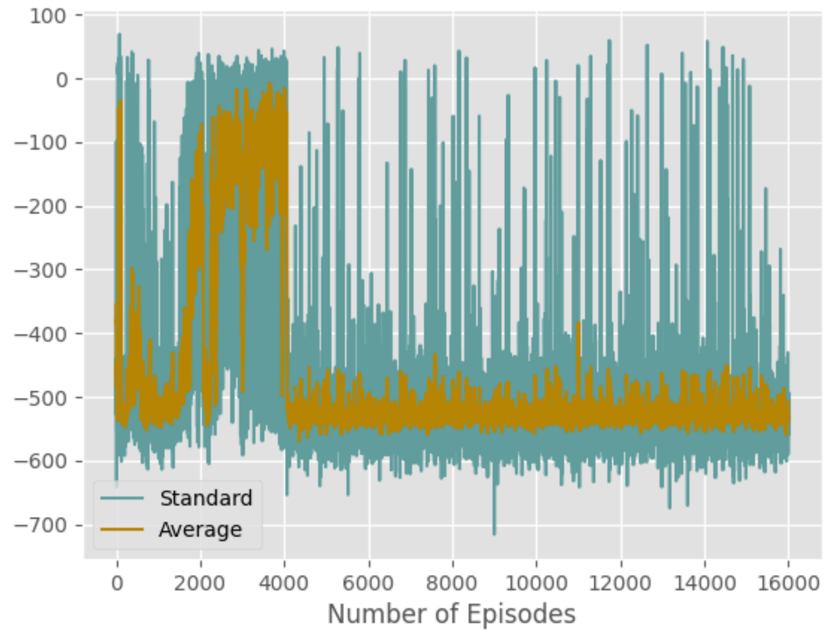
Pytorch Experiments

Reward Shaping Function

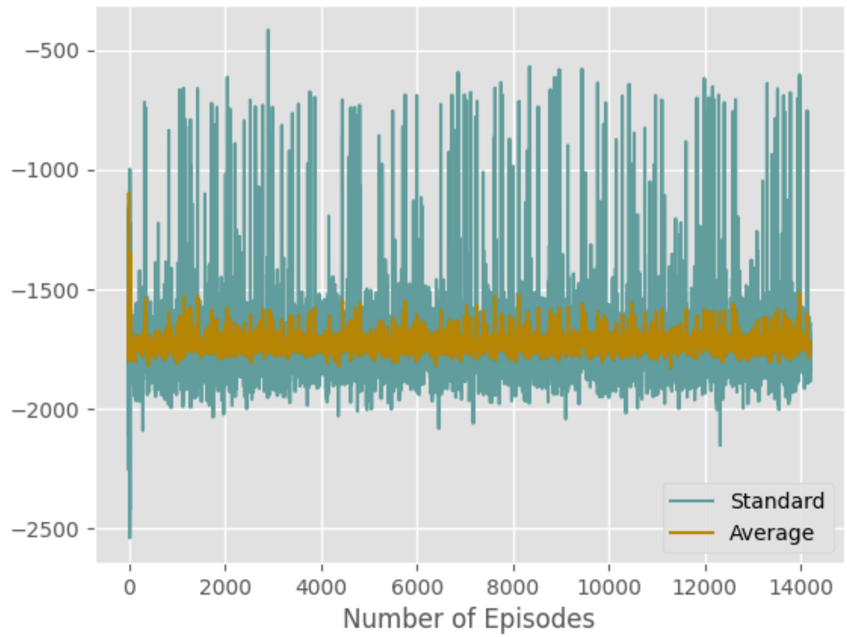
Pytorch Experiments

Penalties

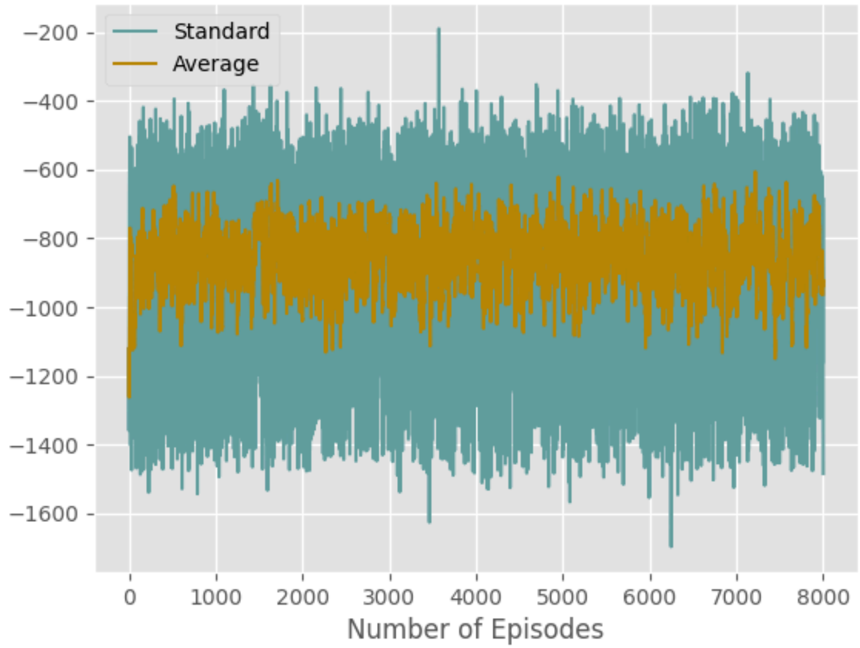
Standing penalty



Falling penalty

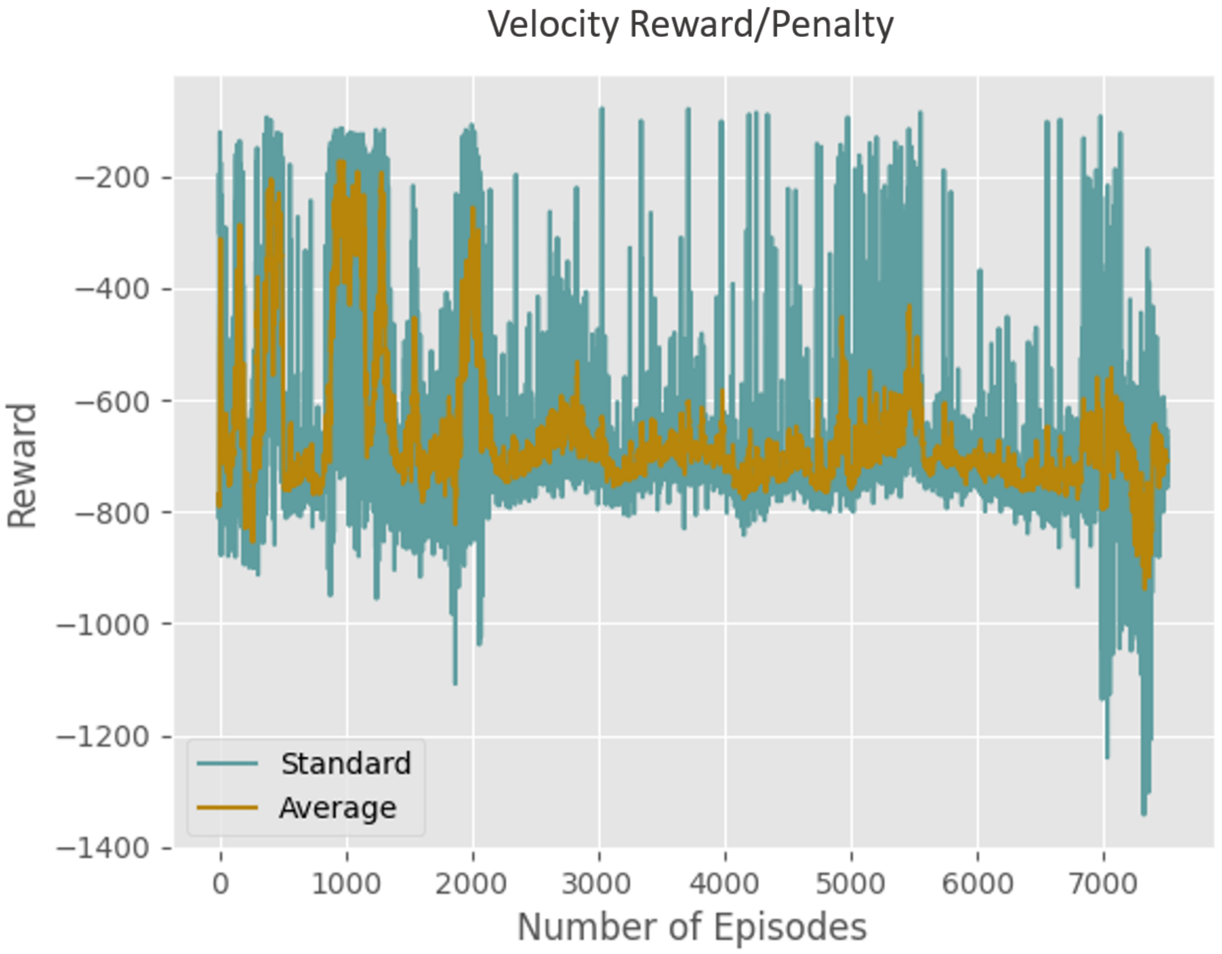


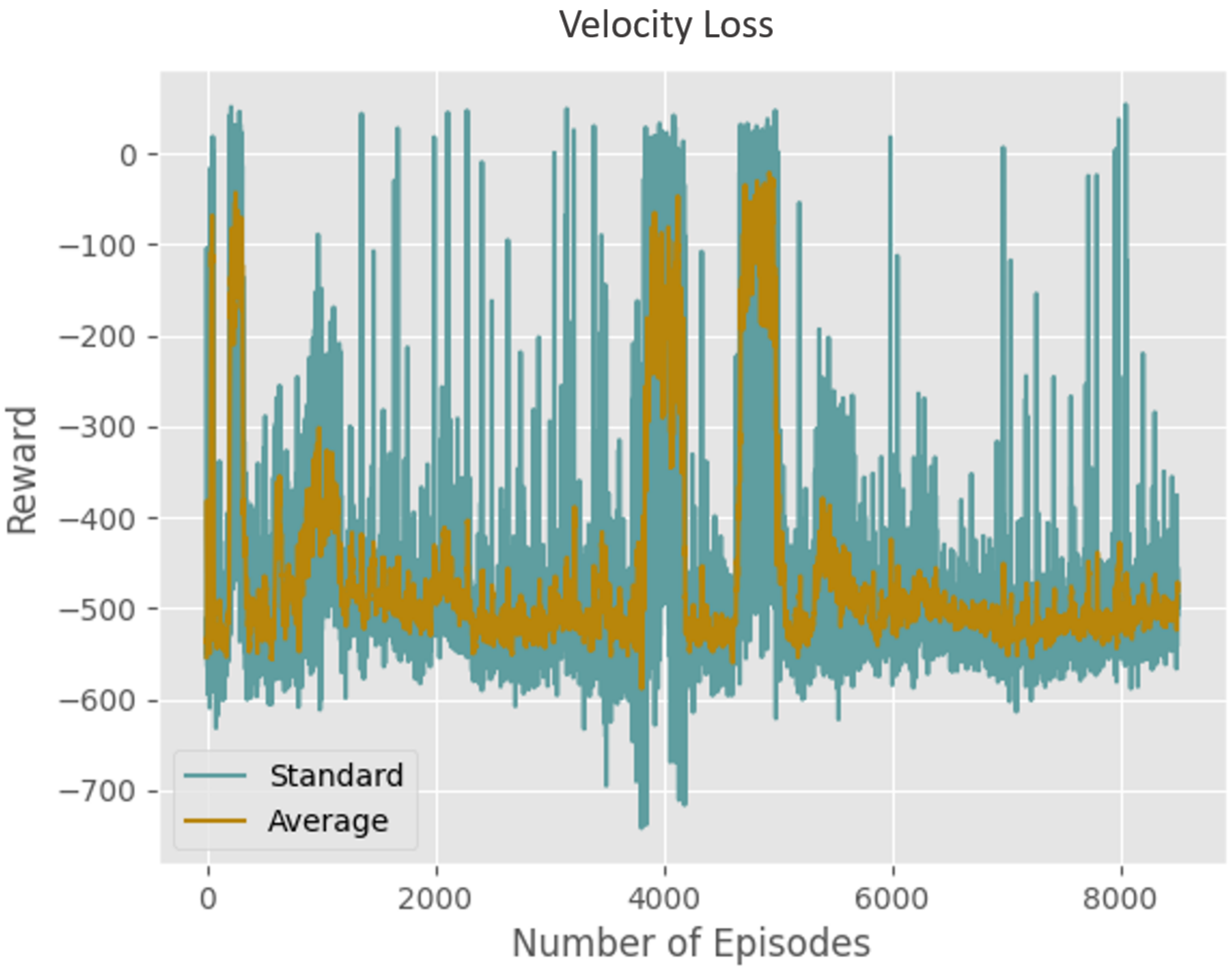
Falling and Standing penalty



Pytorch Experiments

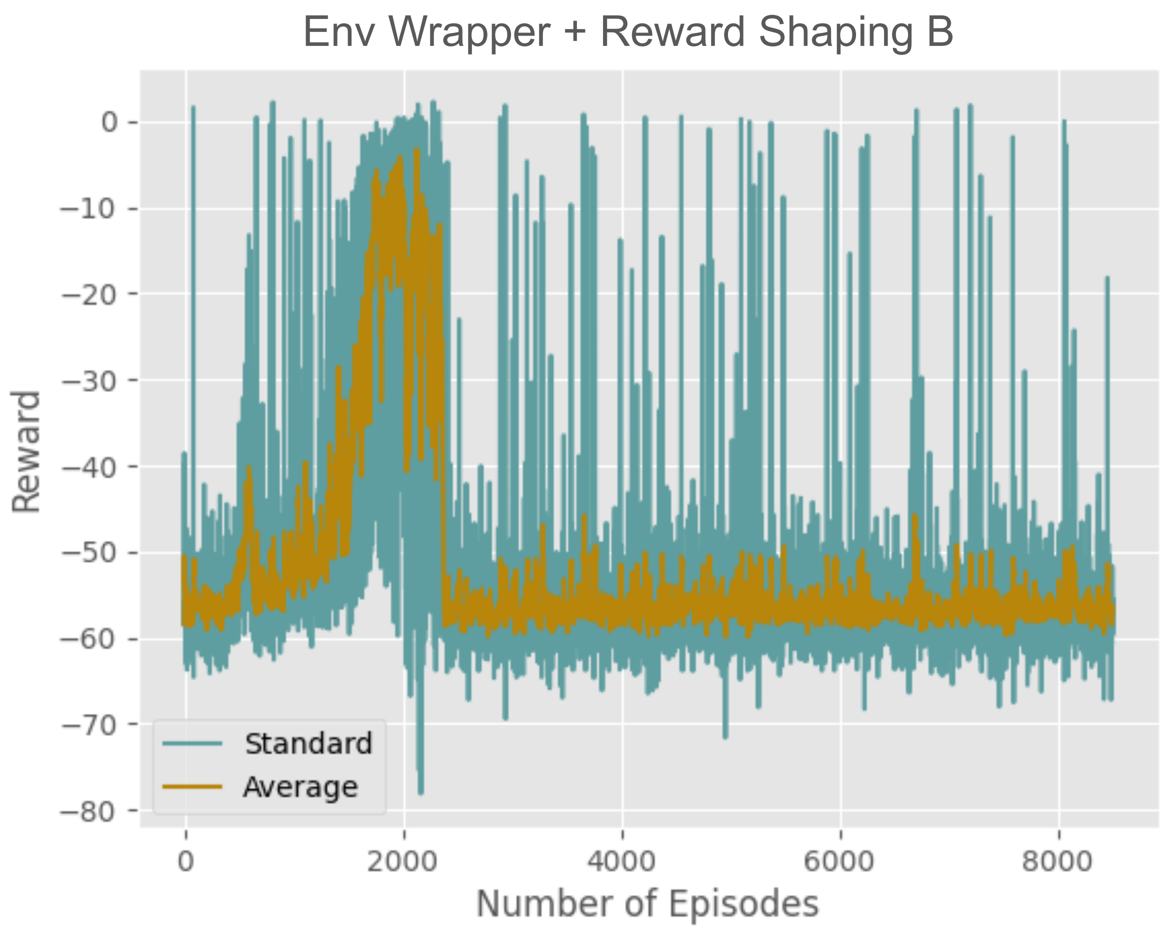
Target Velocity



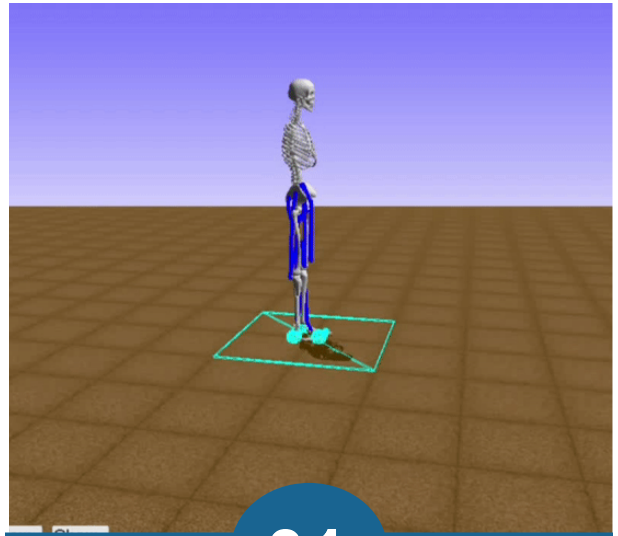


Pytorch Experiments

Environment Wrapper



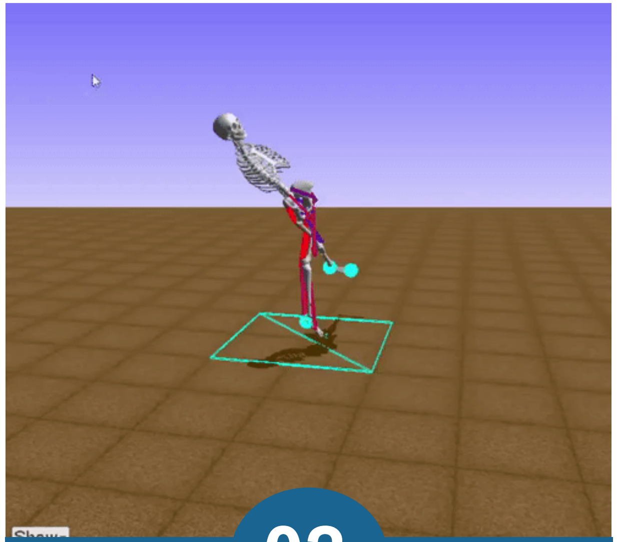
Visualization



01

Reward Shaping B

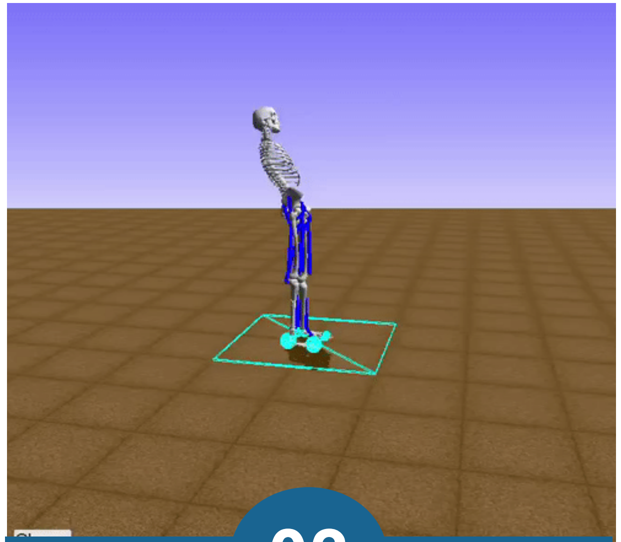
8000 episodes



02

Environment Wrapper + Reward Shaping A

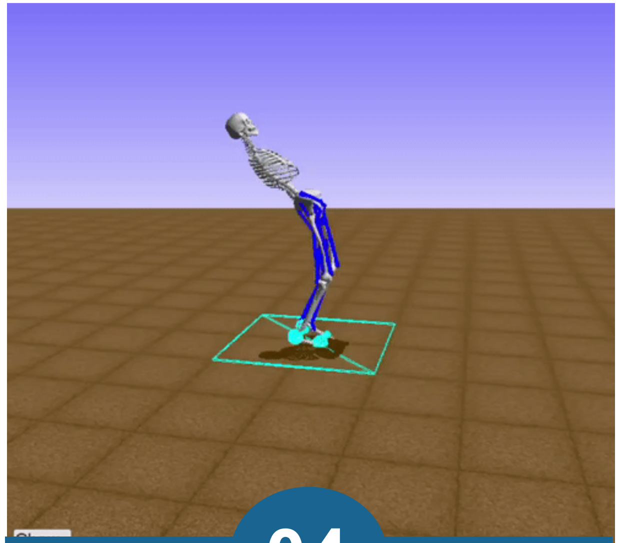
5200 episodes



03

Penalty for standing still + torso falling backward

8072 episodes



04

Velocity Loss

10000 episodes

Limitations

Computational requirement

- In terms of CPU
- Replay Memory grows continuously

Time requirement

- At least 100 hours are necessary
- Batch size
- Full hyperparameter search was not feasible.

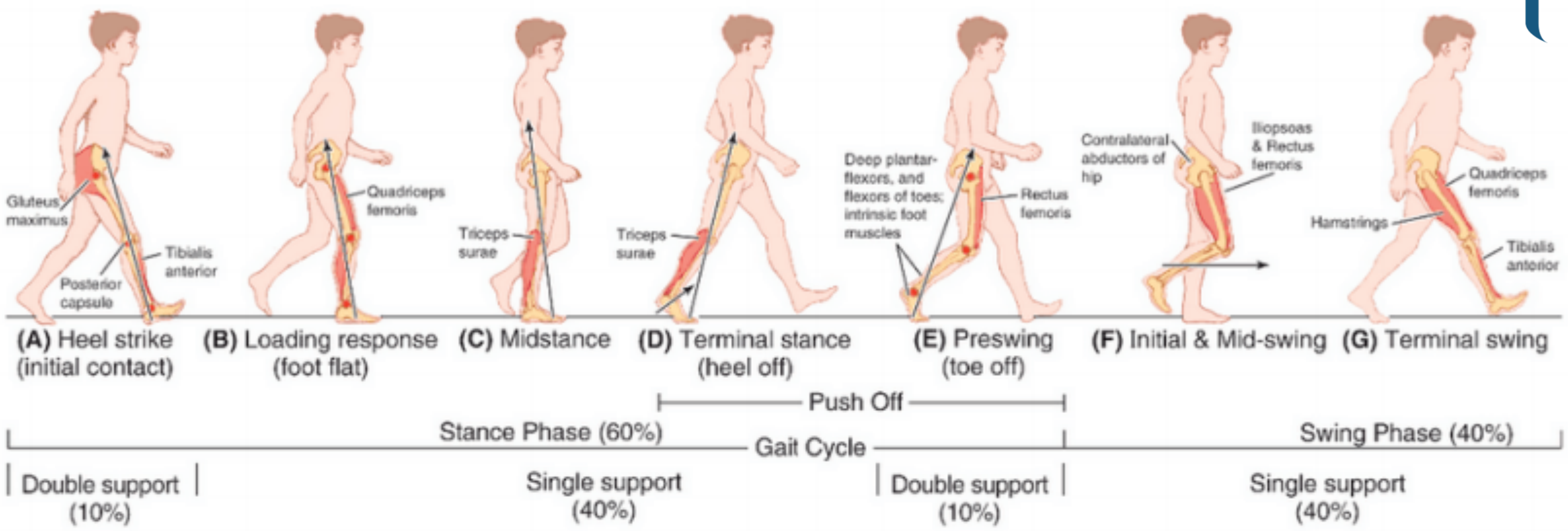
DDPG

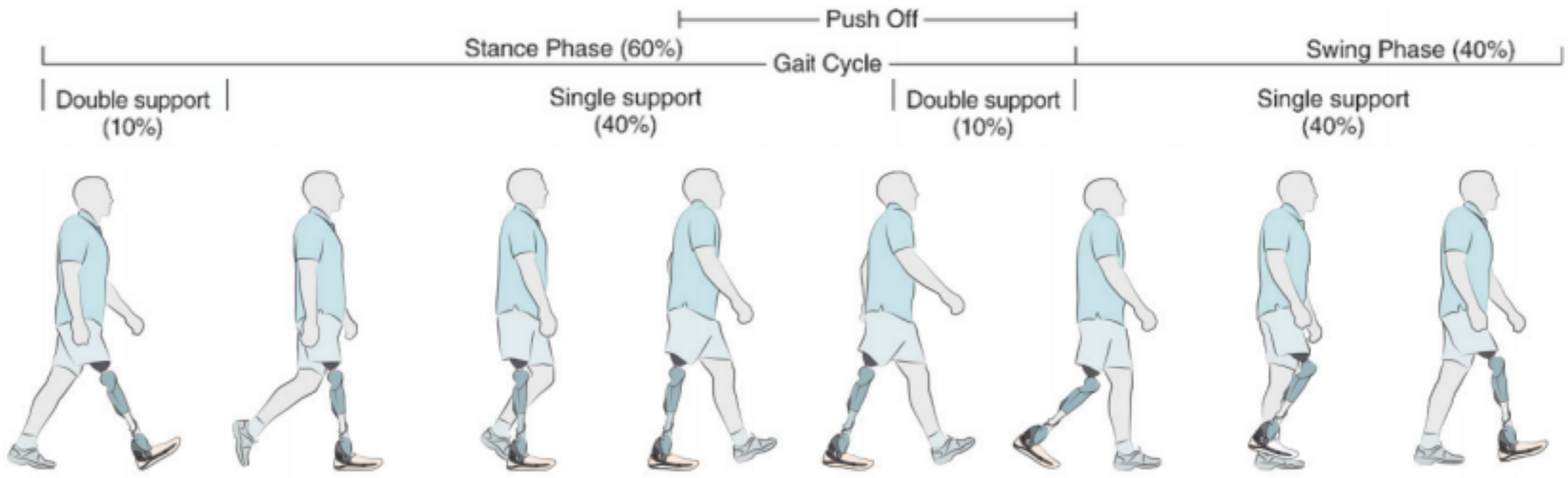
- Multiple actor and critic pairs
- Environment wrapper and reward shaping function

Reinforcement Learning

- Reproducing results for state-of-the-art RL methods is seldom straightforward.

Biomechanical analysis





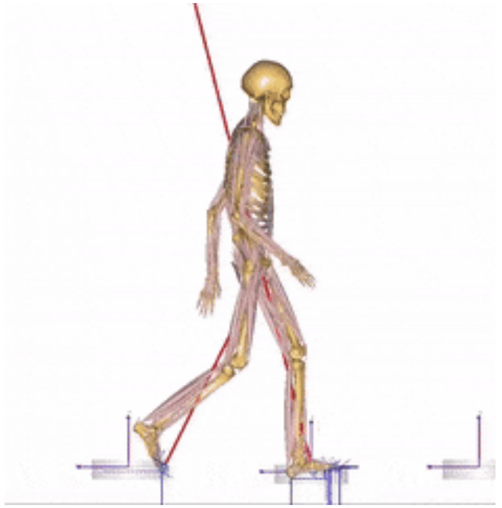
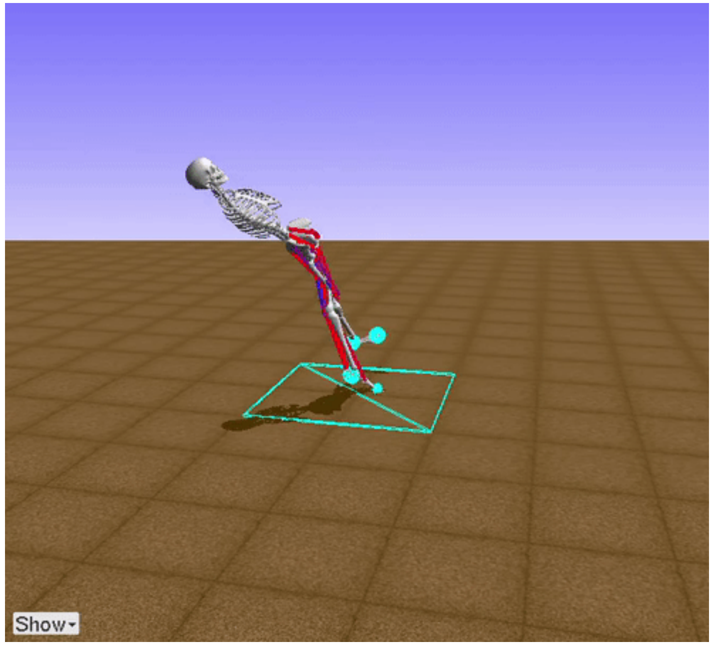


Fig. 2 Normal gait cycle vs. gait cycle with prosthesis

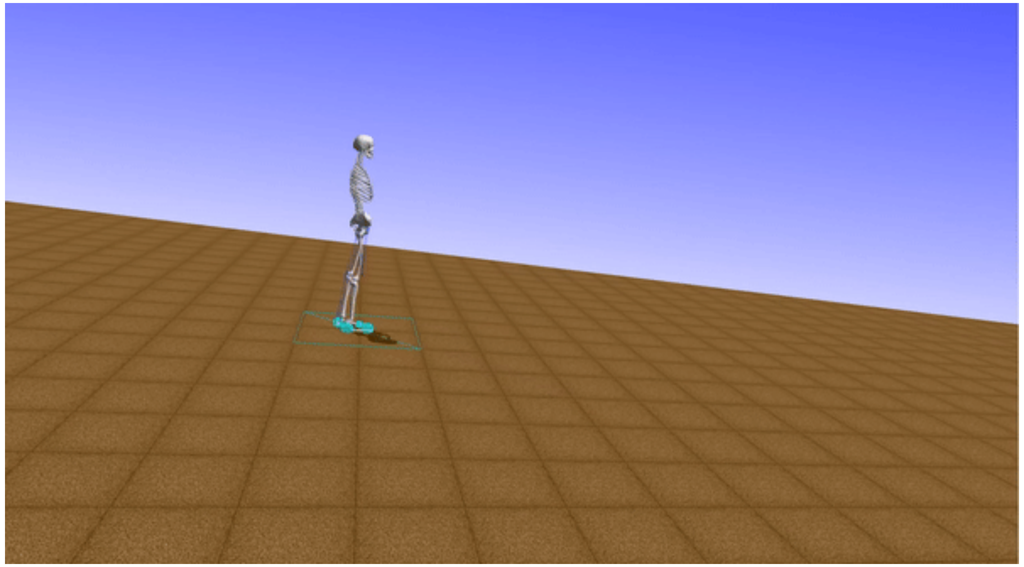
Discussion

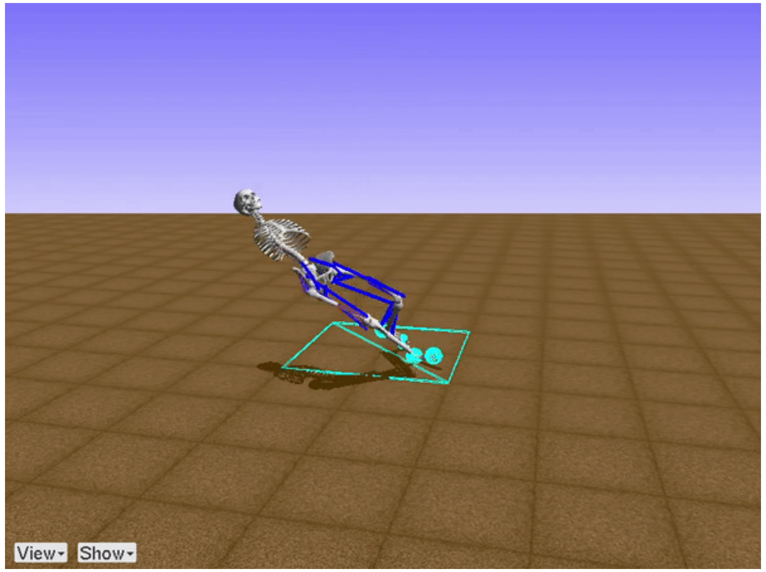


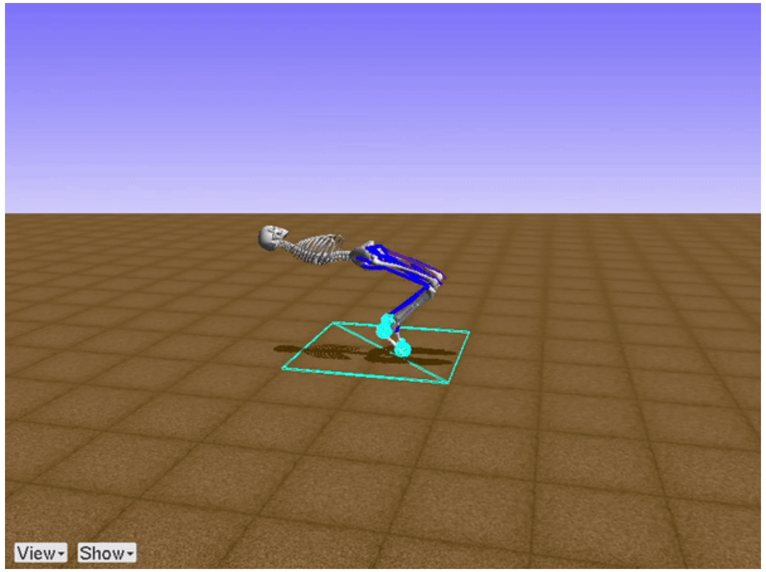
- Load is moved to the healthy limb
- Stabilization depends on the residual muscles
- Minimize the energy cost and gait asymmetry

- Transfer of load and first contact are very important→hardest part
- Overloading of certain groups of muscles
- Increased energy cost

Discussion





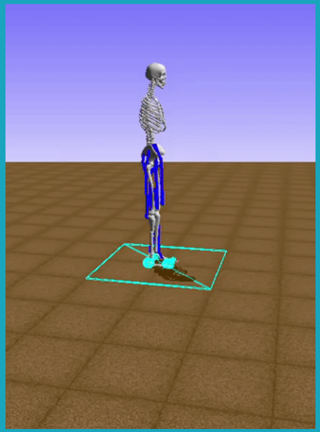


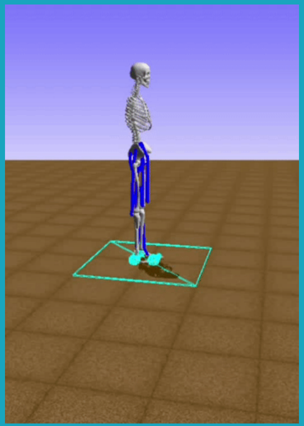
- ✓ Problems with the support phase
  →impossible swing phase
- ✓ It learned to activate less muscles and try to transfer the load (60%)

Comparison with other methods

Challenge solutions

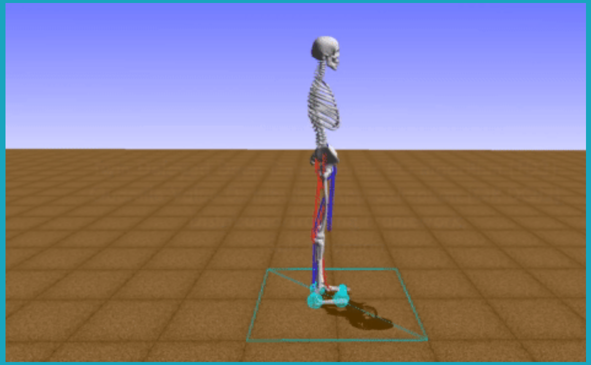
- ❖ Greater cumulative reward
- ❖ Eventually falls if not going fast enough.
- ❖ Movements are usually not human-like.
- ❖ Greater computational capacities and time.





Our result

- ❖ Lacks walking motion but shows intent to move both legs in a step-like fashion.
- ❖ Did not have 500 CPUs or 100 hours per experiment.
- ❖ Negative rewards due to penalization mainly.

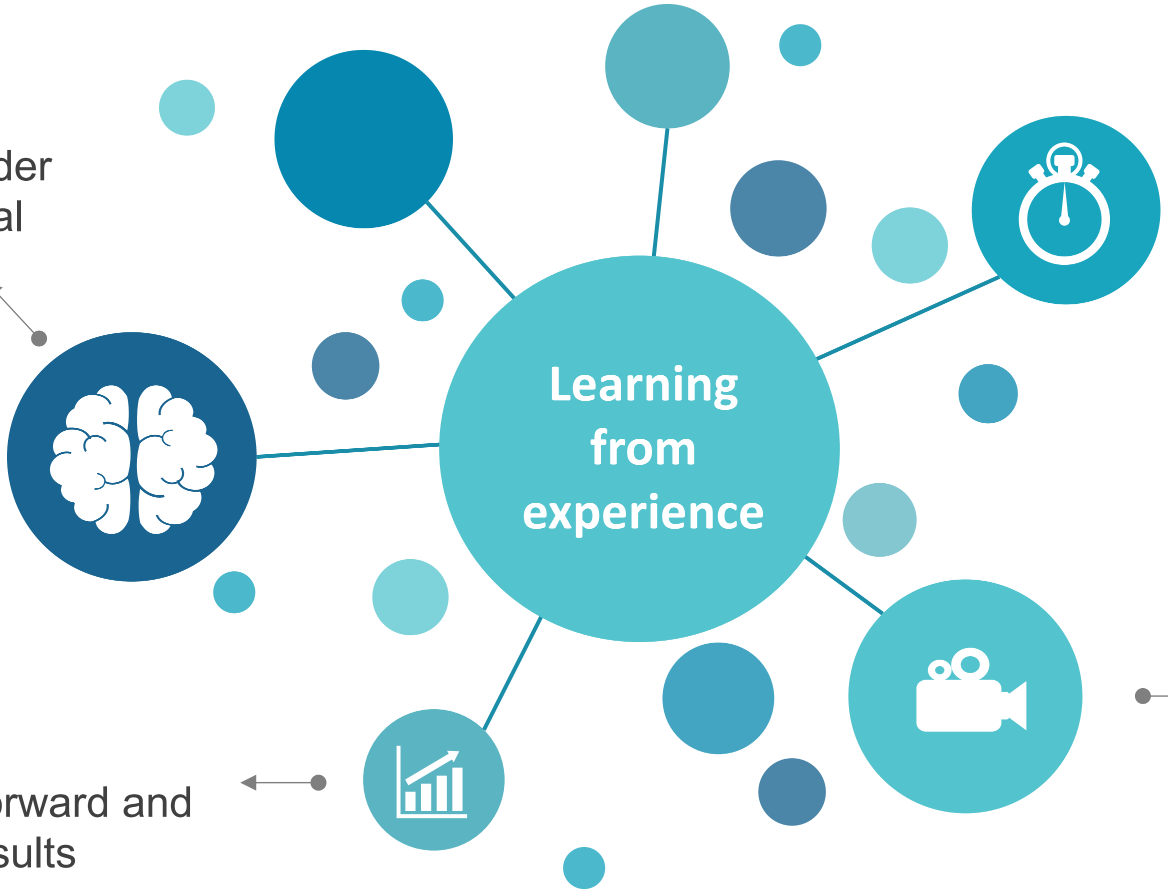


Δ	#	Participant	Media	Cumulative Reward	Mean Episode Time
•	01.	Firework		9980.46	325.066
•	02.	NNAISENSE		9949.93	1716.815
•	03.	Jolly R...		9947.096	297.983

Final insights

Understand results

Extrapolate simulation results with real life in order to understand the medical problem



Time management

As in all RL problems, the limiting factor is mostly time

Further work

Manage to leap forward and repeat positive results

Visualization

Possibility of having remarkable numerical results but questionable movements

REFERENCES

[1] Neurips 2018: Ai for prosthetics challenge. https://www.crowdai.org/challenges/ neurips-2018-ai-for-prosthetics-challenge, 2018. (Accessed on 08/24/2020).
[2] B. Bakker. Reinforcement learning with long short-term memory. Advances in neural information processing sys- tems, 14:1475–1482, 2001.
[3] C. Crerand and L. Magee. Amputations and prosthetic de- vices. In Encyclopedia of body image and human appear- ance, pages 1–7. Elsevier, 2012.
[4] B. Davies and D. Datta. Mobility outcome following unilat- eral lower limb amputation. Prosthetics and orthotics inter- national, 27(3):186–190, 2003.
[5] El Tiempo. S´ı hay salida para los amputados - eltiempo.com. https://www.eltiempo.com/ archivo/documento/MAM- 1992341, April 2006. (Accessed on 11/26/2020).
[6] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger. Deep reinforcement learning that matters. arXiv preprint arXiv:1709.06560, 2017.
[7] Ł. Kidzin´ski, S. P. Mohanty, C. F. Ong, J. L. Hicks, S. F. Carroll, S. Levine, M. Salathe´, and S. L. Delp. Learning to run challenge: Synthesizing physiologically accurate motion using deep reinforcement learning. In The NIPS'17 Competi- tion: Building Intelligent Systems, pages 101–120. Springer, 2018.
[8] Ł. Kidzin´ski, S. P. Mohanty, C. F. Ong, Z. Huang, S. Zhou, A. Pechenko, A. Stelmaszczyk, P. Jarosik, M. Pavlov, S. Kolesnikov, et al. Learning to run challenge solutions: Adapting reinforcement learning methods for neuromus- culoskeletal environments. In The NIPS'17 Competition: Building Intelligent Systems, pages 121–153. Springer, 2018.
[9] Ł. Kidzinski, C. Ong, S. P. Mohanty, J. Hicks, S. Carroll, B. Zhou, H. Zeng, F. Wang, R. Lian, H. Tian, et al. Arti- ficial intelligence for prosthetics: Challenge solutions. The NeurIPS'18 Competition: From Machine Learning to Intel- ligent Conversations, page 69, 2019.
[10] S. Kolesnikov and O. Hrinchuk. Catalyst.rl: A distributed framework for reproducible rl research. arXiv preprint arXiv:1903.00027, 2019.
[11] A. Kumar, N. Paul, and S. Omkar. Bipedal walking robot using deep deterministic policy gradient. arXiv preprint arXiv:1807.05924, 2018.
[12] S. Li, Y. Wu, X. Cui, H. Dong, F. Fang, and S. Rus- sell. Robust multi-agent reinforcement learning via mini- max deep deterministic policy gradient. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 4213–4220, 2019.
[13] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous con- trol with deep reinforcement learning. arXiv preprint arXiv:1509.02971, 2015.
[14] M. Mohammedalamen. Reinforcement learning for pros- thetics. https://github.com/montaserFath/Reinforcement-Learning-for-Prosthetics, 2019.

REFERENCES

[15] C. F. Ong, T. Geijtenbeek, J. L. Hicks, and S. L. Delp. Pre-
dictive simulations of human walking produce realistic cost of transport at a range of speeds. In *Proceedings of the 16th International Symposium on Computer Simulation in Biome- chanics*, pages 19–20, 2017.
[16] PaddlePaddle. The winning solution for the neurips 2018: Ai for prosthetics challenge. https://github.com/ PaddlePaddle/PARL/tree/develop/examples/ NeurIPS2018-AI-for-Prosthetics-Challenge, 2018.
[17] M.R. Pitkin. *Biomechanics of lower limb prosthetics*. Springer, 2009
[18] V.Rajtukova, M. Michalikova, L Bednarcikova, A. Balogova & J. Zivcak. Biomechanics of lower limb prostheses. *Procedia engineering*, 96:382–391, 2014.
[19] A.M.Scha ̈fer.Reinforcementlearningwithrecurrentneural networks. 2008.
[20] J.Schulman,S.Levine,P.Abbeel,M.Jordan,andP.Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.
[21] G. L. Team. Use of reinforcement learning in healthcare. *Great Learning Blog, Power ahead*, 2020.
[22] S. Tokui, R. Okuta, T. Akiba, Y. Niitani, T. Ogawa, S. Saito, S. Suzuki, K. Uenishi, B. Vogel, and H. Yamazaki Vincent. Chainer: A deep learning framework for accelerating the research cycle. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2002–2011, 2019.
[23] R. Wightman. Pytorch reinforcement learning for opensim environments. https://github.com/rwightman/ pytorch-opensim-rl, 2018.
[24] M. Windrich, M. Grimmer, O. Christ, S. Rinderknecht, and P. Beckerle. Active lower limb prosthetics: a systematic re- view of design issues and solutions. *Biomedical engineering online*, 15(3):140, 2016.
[25] C. Yoon. Rlcycle. https://github.com/ cyoon1729/RLcycle, 2020.
[26] C. Yu, J. Liu, and S. Nemati. Reinforcement learning in healthcare: A survey. *arXiv preprint arXiv:1908.08796*, 2019.