Final Presentation of Bachelor Project

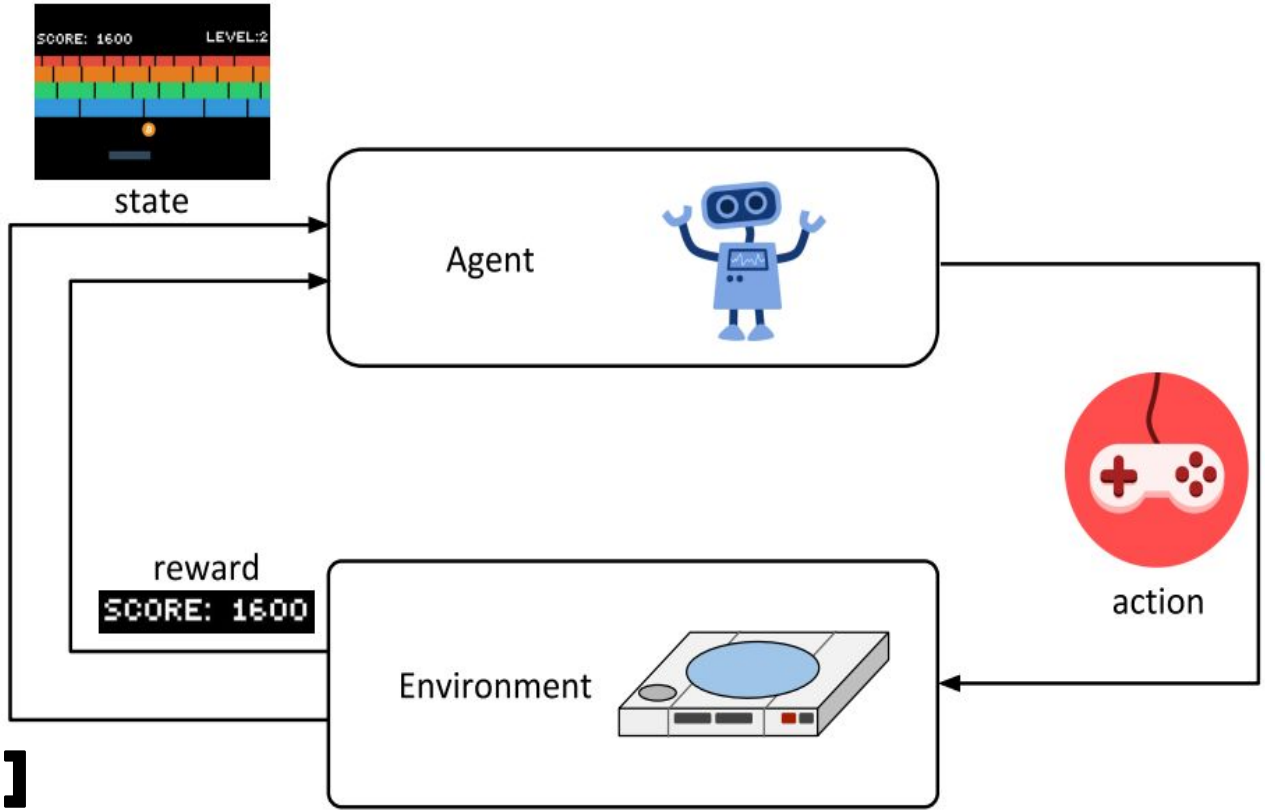
Sources of Randomness in Deep Reinforcement Learning

universität innsbruck

Elizaveta Terente
Supervised by : Jakob Hollenstein and Samuele Tosatto

Introduction

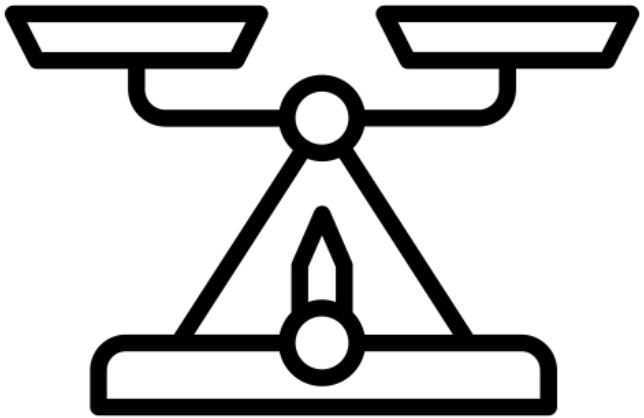
Reinforcement Learning



[1]

Randomness in RL

Exploration + Exploitation



Randomness in RL

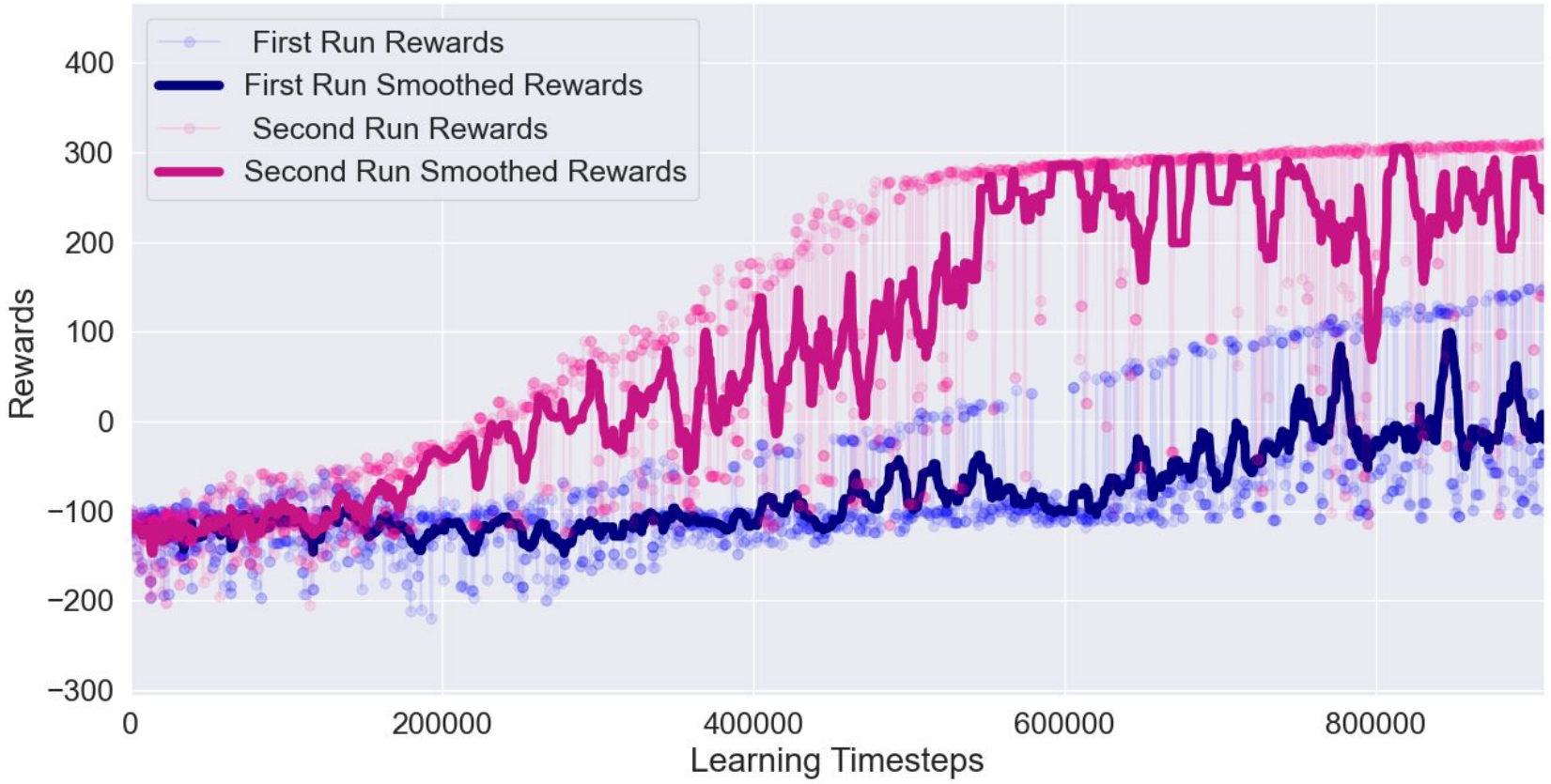
+ :

Exploring more
=>
finding **optimal strategy**

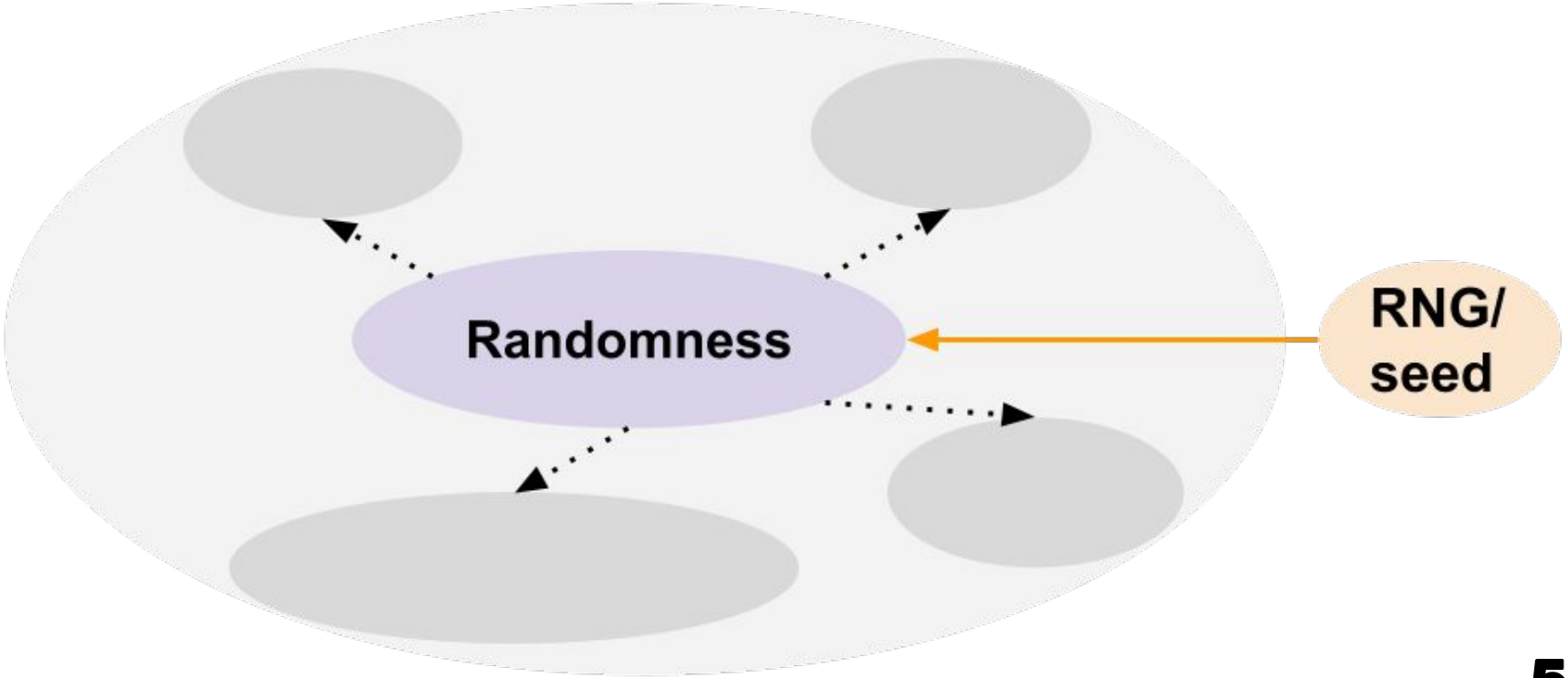
- :

different performance with **identical** training **conditions**

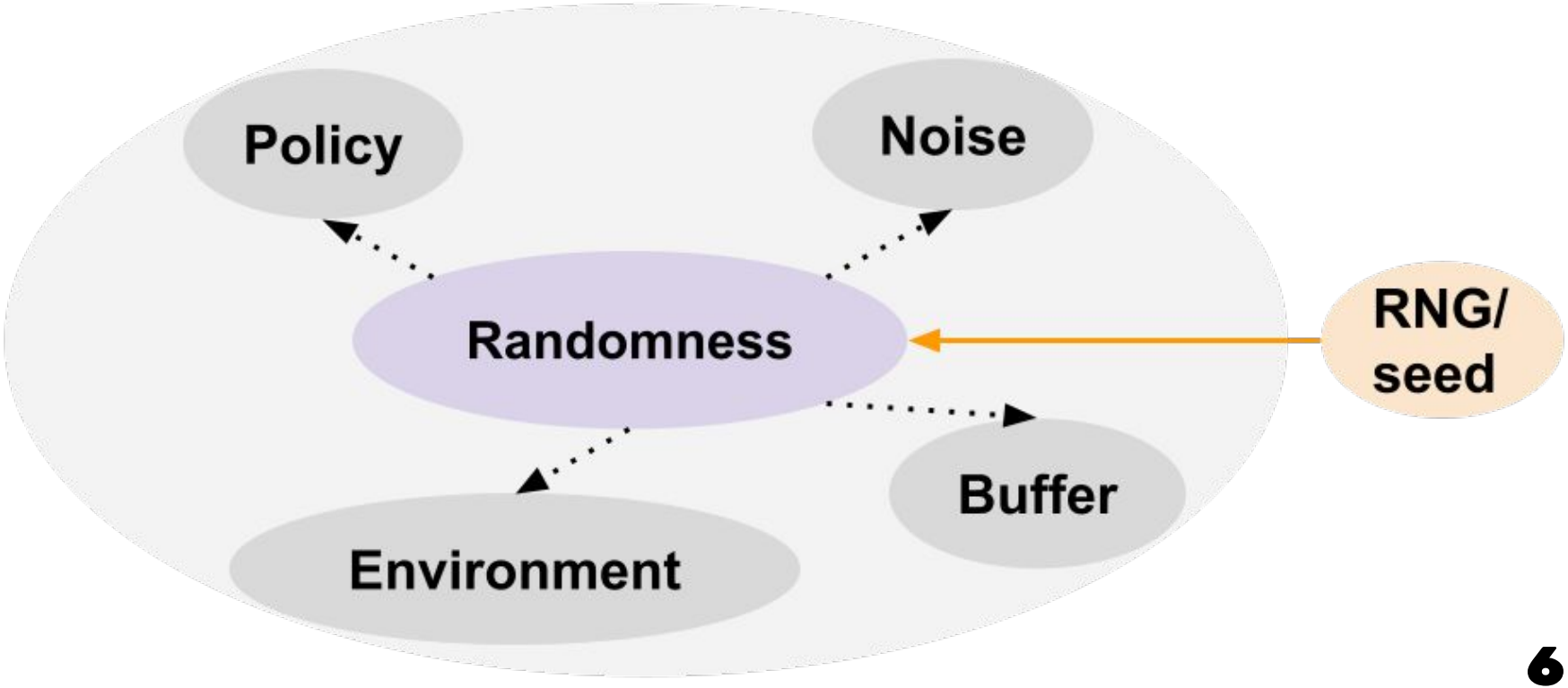
2 runs with identical conditions



The Seed



Sources of randomness



Goals

Investigate each source of randomness isolated to be able to answer following questions:

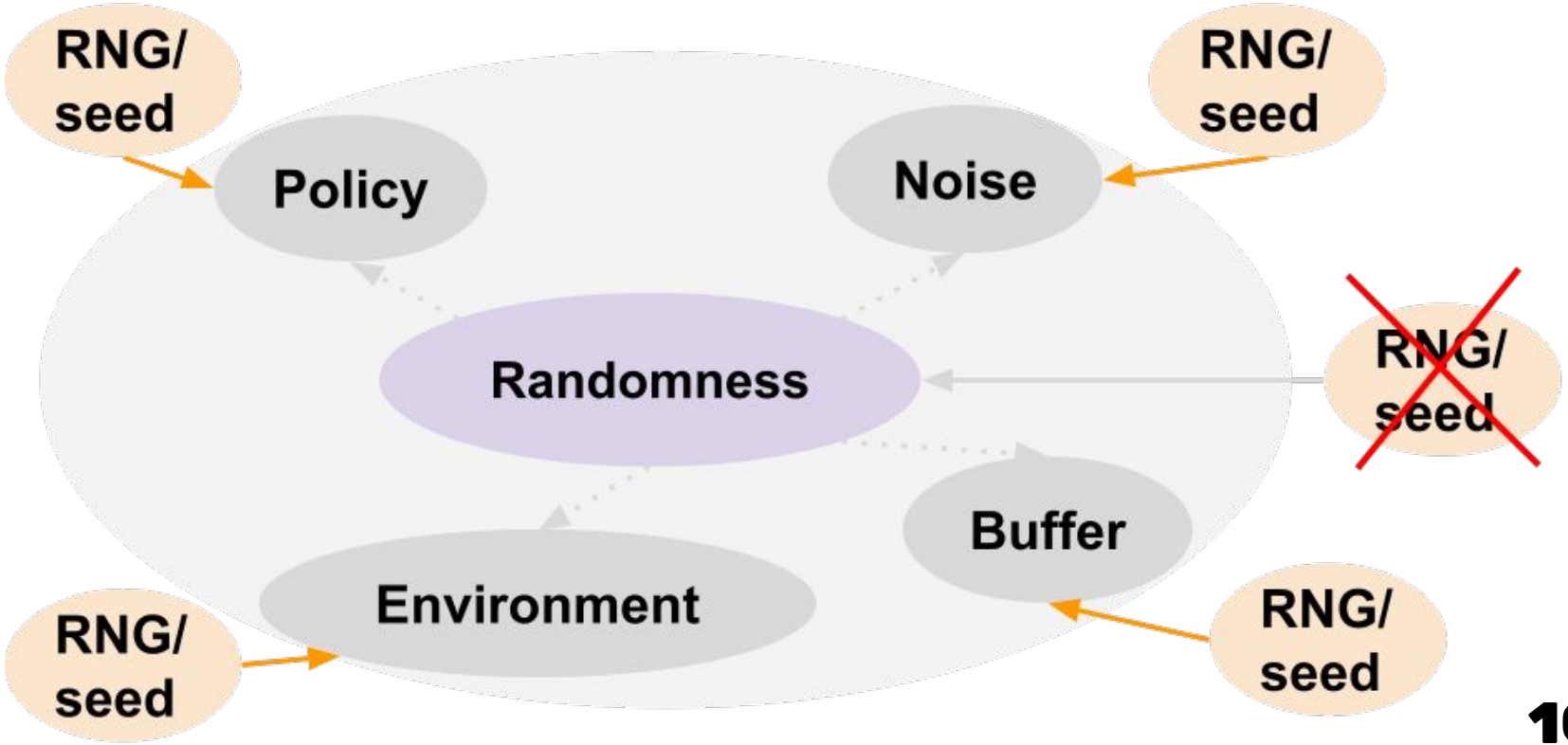
- How much do sources of randomness affect agent performance?
- Which sources of randomness have a greater impact, and which are less influential?

Methodology

Algorithms and Environments used

- **Soft Actor-Critic (SAC)** and **Proximal Policy Optimization (PPO)**
- reliable **implementations** of these algorithms taken from **Stable Baselines 3** [2]
- **16** environments provided by **Gymnasium** [3] and **MuJoCo** [4].

Set separate seeds



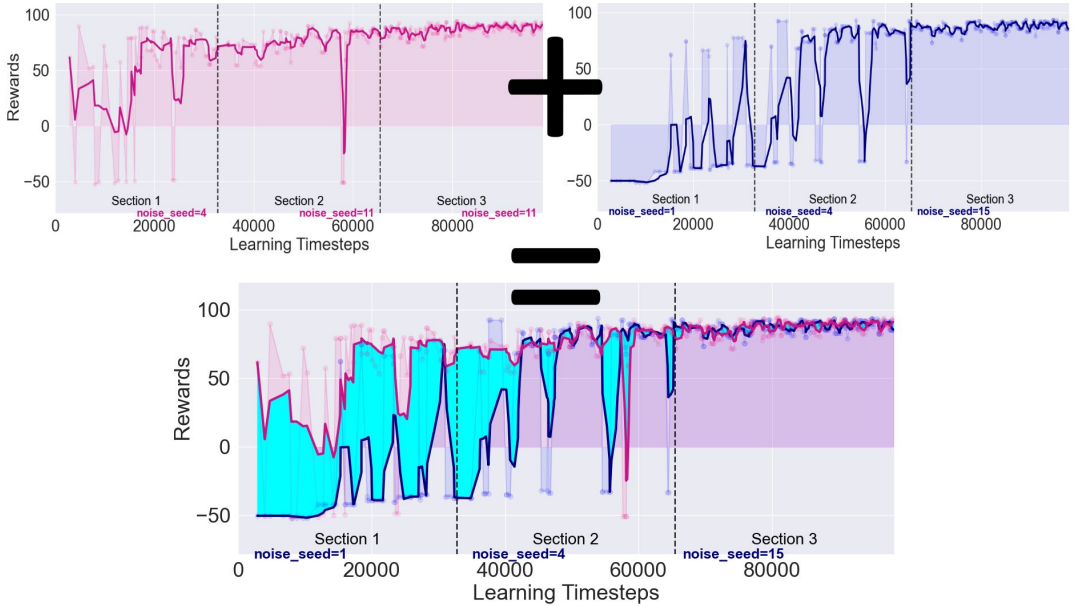
Abstract of the Central Concept

- **Source of randomness to investigate = environment**
- **Fixed seeds = 5(buffer), 6(noise), 7(policy)**
- **Seeds for environment = [1 ... 4]**

name	buffer_seed	noise_seed	policy_seed	env_seed	performance
model1	5	6	7	1	*0.85*
model2	5	6	7	2	*0.83*
model3	5	6	7	3	*0.87*
model4	5	6	7	4	*0.82*

Impact estimation

name	buffer_seed	noise_seed	policy_seed	env_seed	performance
model1	5	6	7	1	0.85
model2	5	6	7	2	0.83
model3	5	6	7	3	0.87
model4	5	6	7	4	0.82

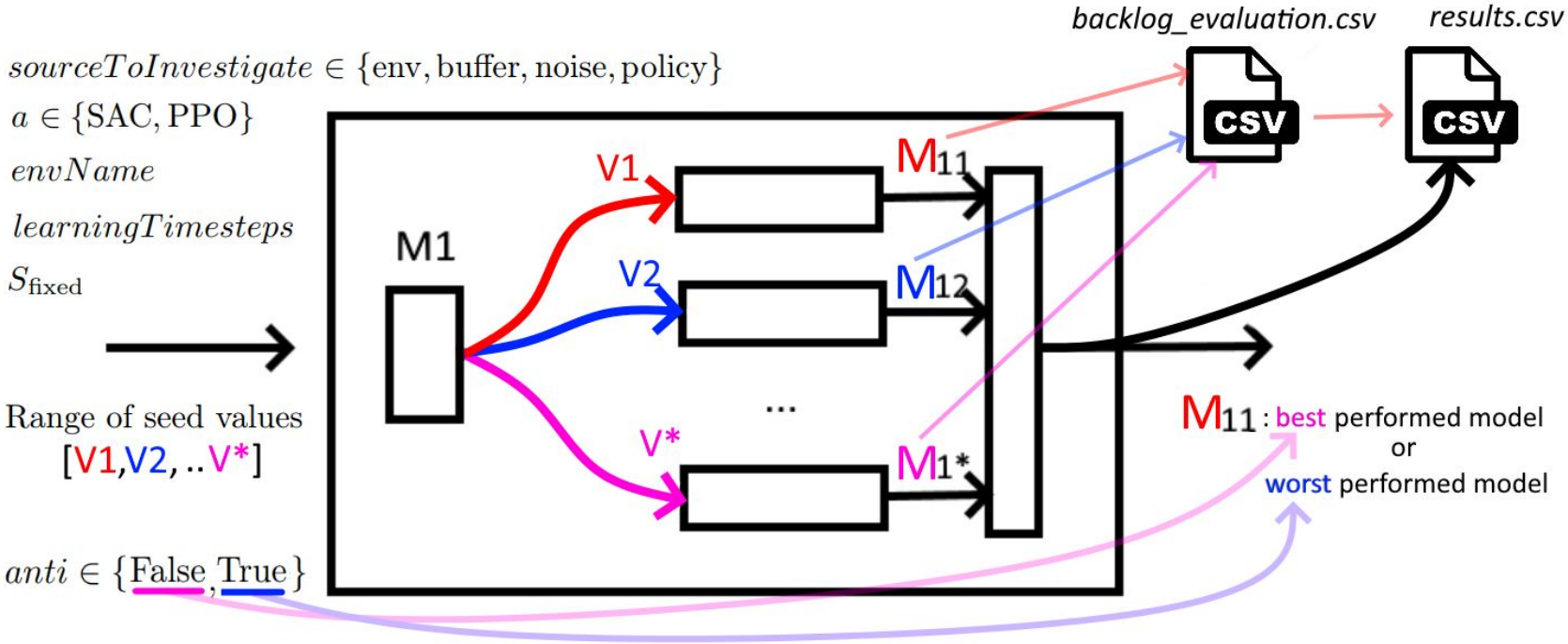


Difference between best-performed and worst-performed models

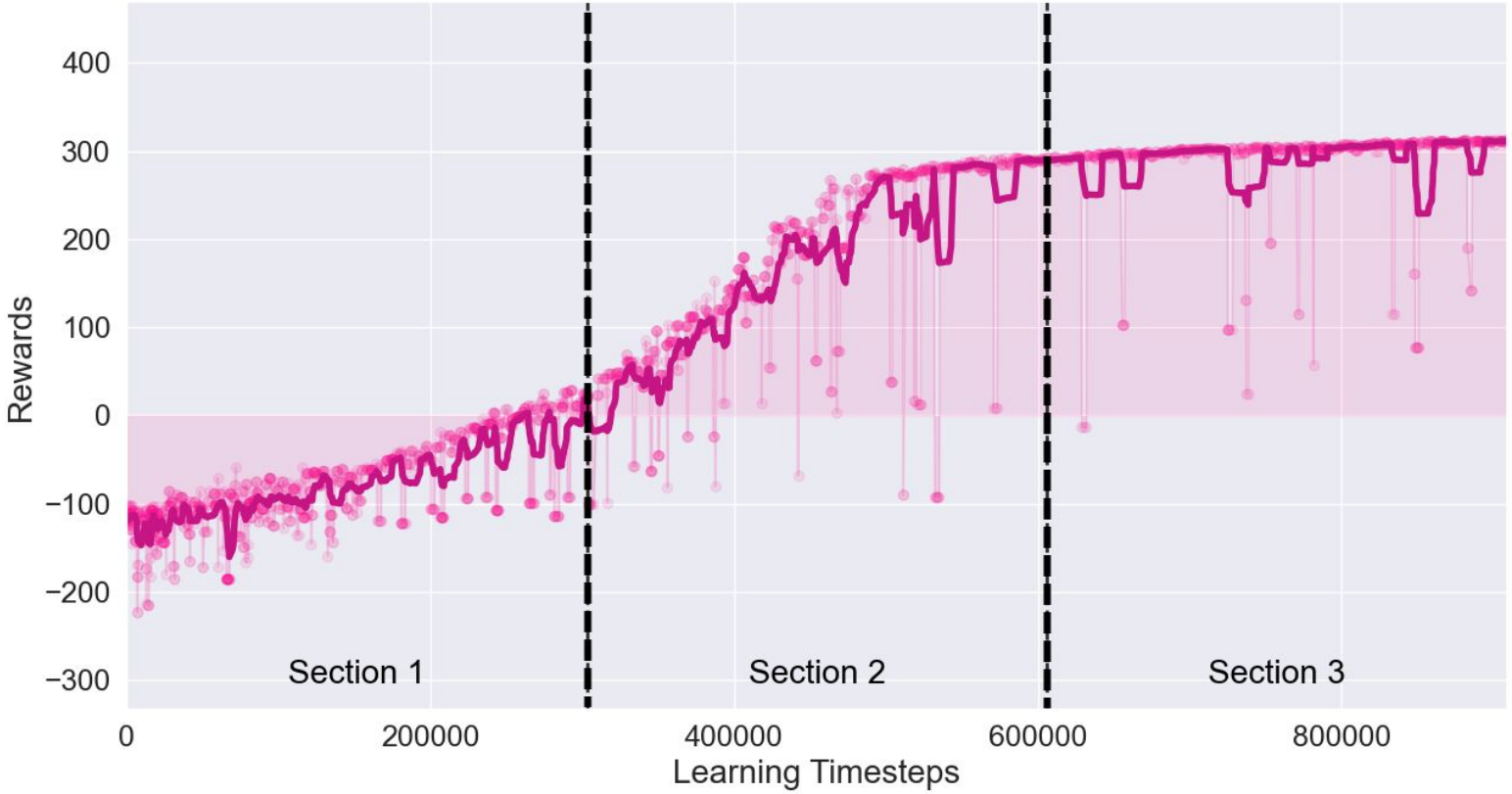
=

Impact range

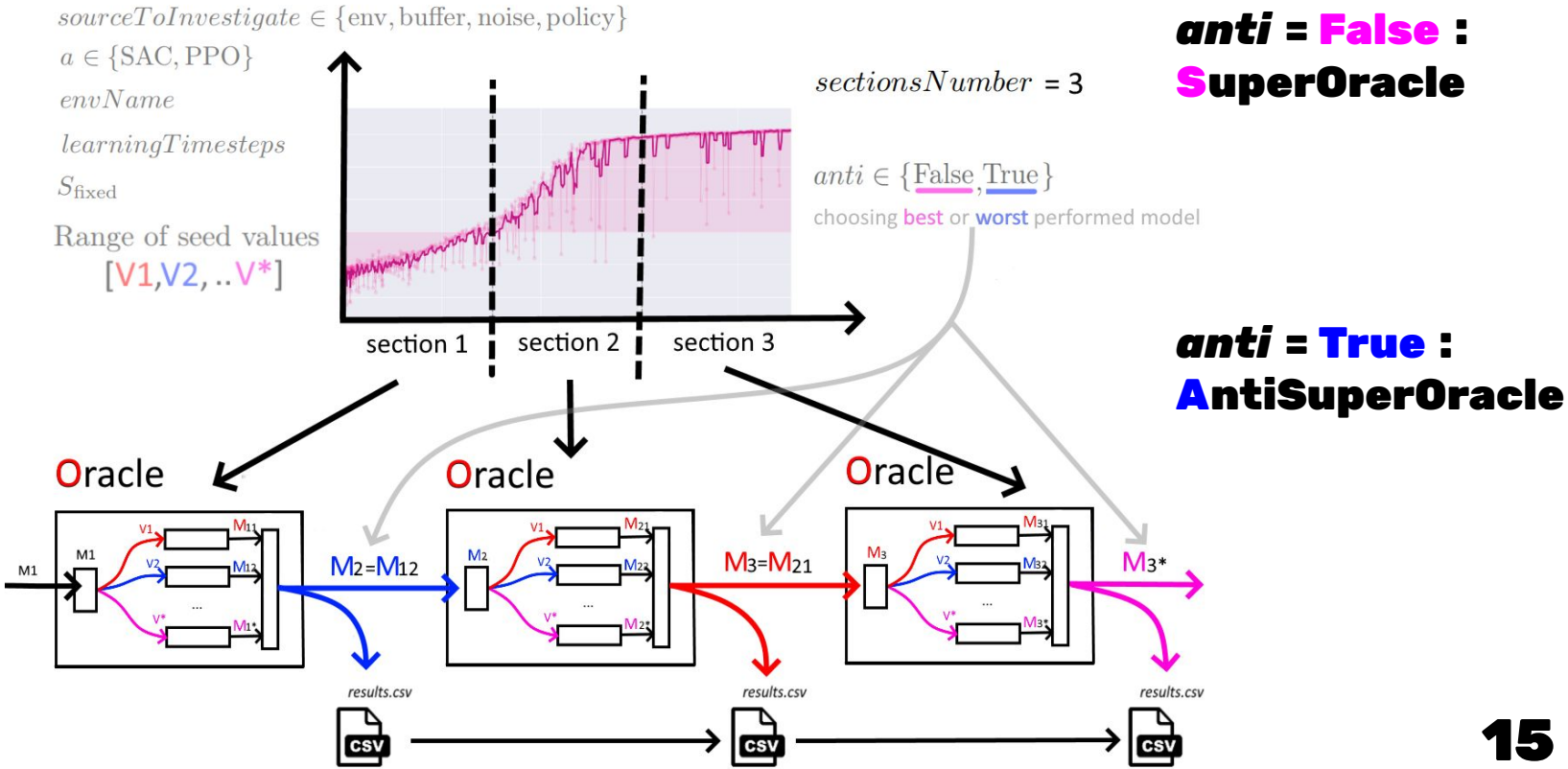
Oracle


backlog_evaluation.csv
results.csv
$sourceToInvestigate \in \{\text{env}, \text{buffer}, \text{noise}, \text{policy}\}$
$a \in \{\text{SAC}, \text{PPO}\}$
$envName$
$learningTimesteps$
S_{fixed}
Range of seed values
[V1,V2, ..V*]
$anti \in \{\text{False}, \text{True}\}$
M1
V1
V2
V*
...
M11
M12
M1*
CSV
CSV
M11 : best performed model
or
worst performed model

Different stages of learning



SuperOracle

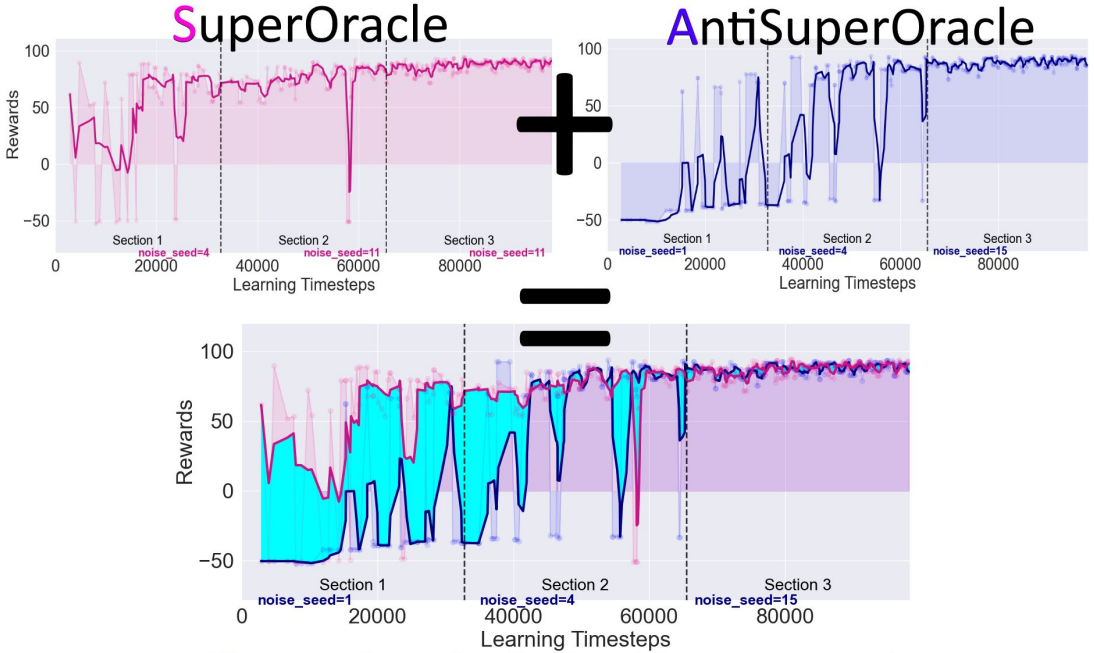


SuperOracle & AntiSuperOracle

Run script twice to obtain:

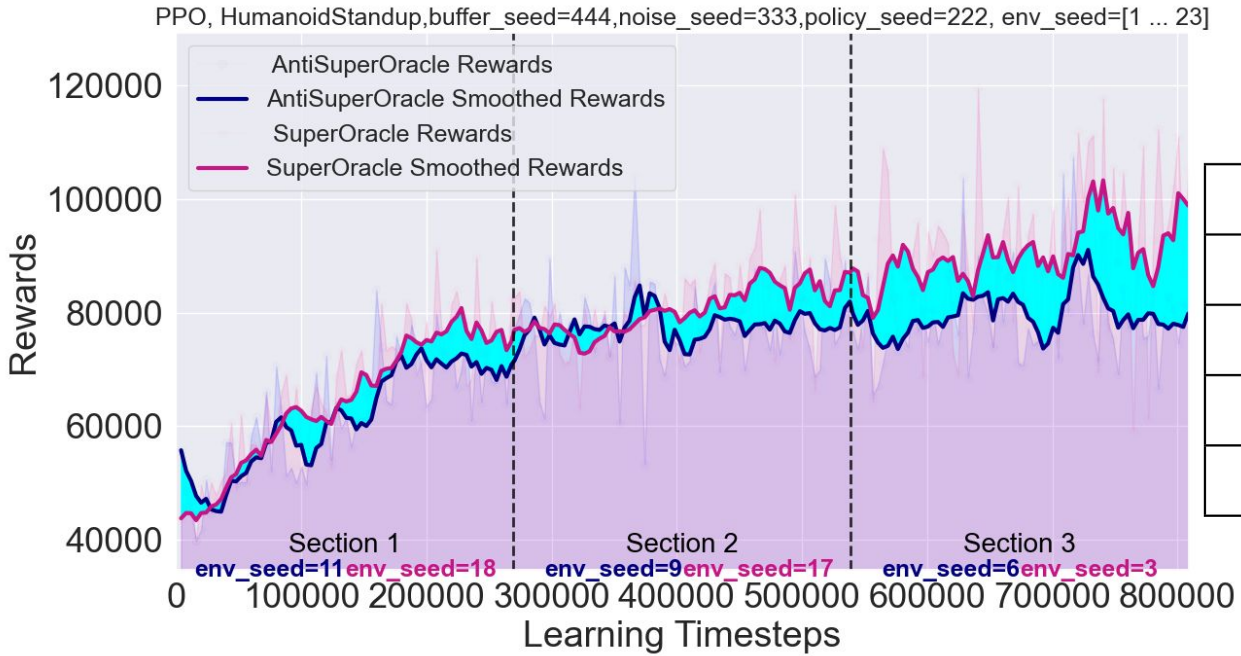
Upper performance bound = SuperOracle

Lower performance bound = AntiSuperOracle



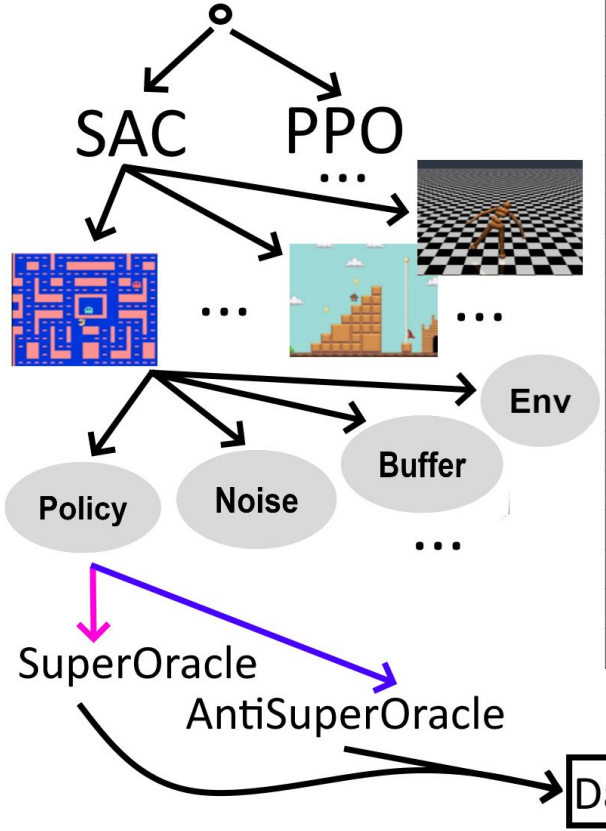
Impact quantification : SMAPE

source =env



Section	env
1	5.1%
2	4.9%
3	8.6%
all	6.2%

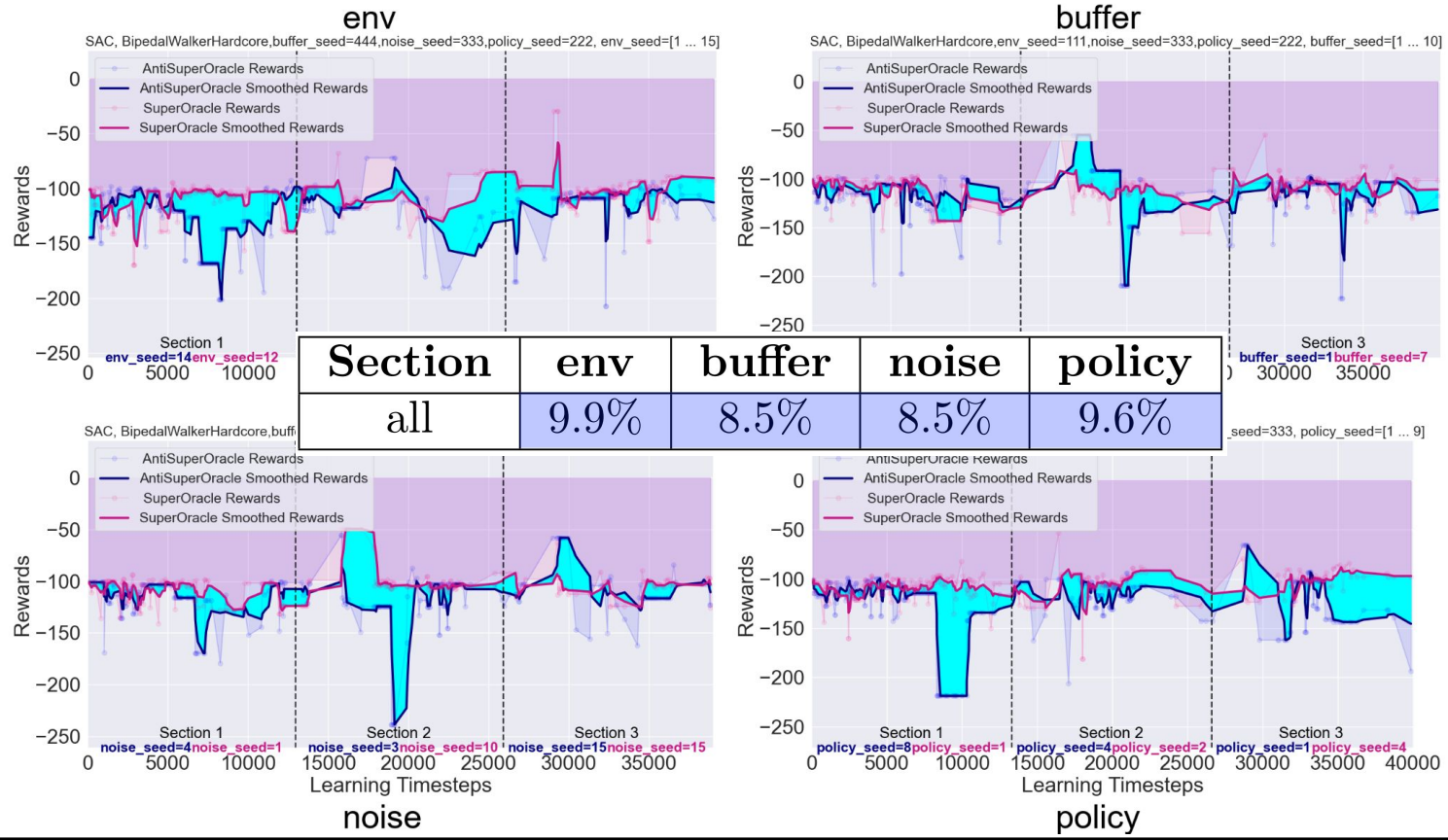
All Experiments



algorithm	environment	source	section	SMAPE (%)
SAC	Humanoid-v4	env	1	11.1
SAC	Humanoid-v4	env	2	10.2
SAC	Humanoid-v4	env	3	11.4
SAC	Humanoid-v4	env	all	10.9
SAC	Humanoid-v4	buffer	1	10.5
SAC	Humanoid-v4	buffer	2	10.8
SAC	Humanoid-v4	buffer	3	12.9
SAC	Humanoid-v4	buffer	all	11.1
SAC	Humanoid-v4	noise	1	9.9
SAC	Humanoid-v4	noise	2	10.0
SAC	Humanoid-v4	noise	3	12.3
SAC	Humanoid-v4	noise	all	10.4
SAC	Humanoid-v4	policy	1	10.7
SAC	Humanoid-v4	policy	2	12.1
SAC	Humanoid-v4	policy	3	13.1
SAC	Humanoid-v4	policy	all	11.6

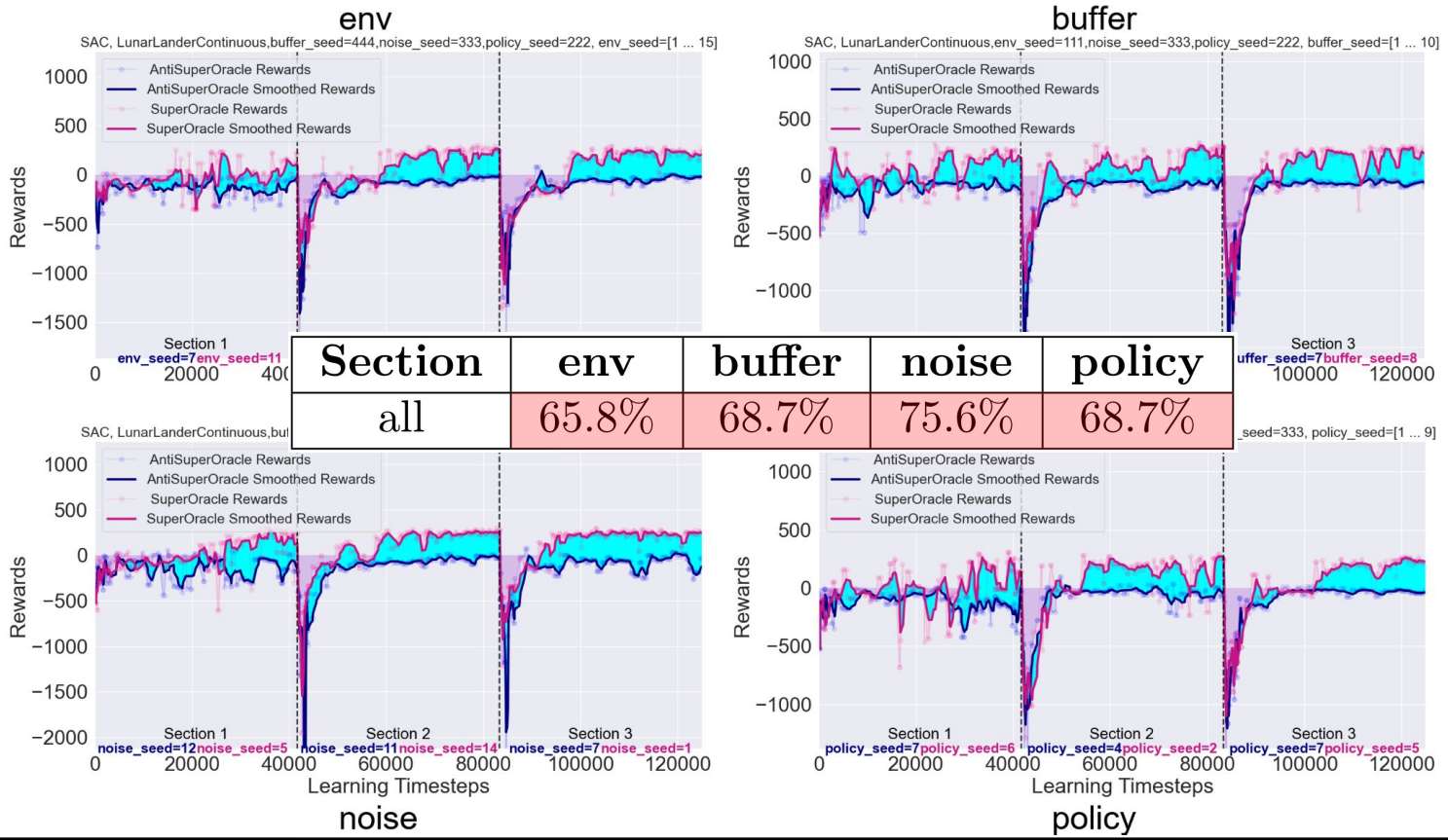
Experimental results and conclusions

the impact is consistently small



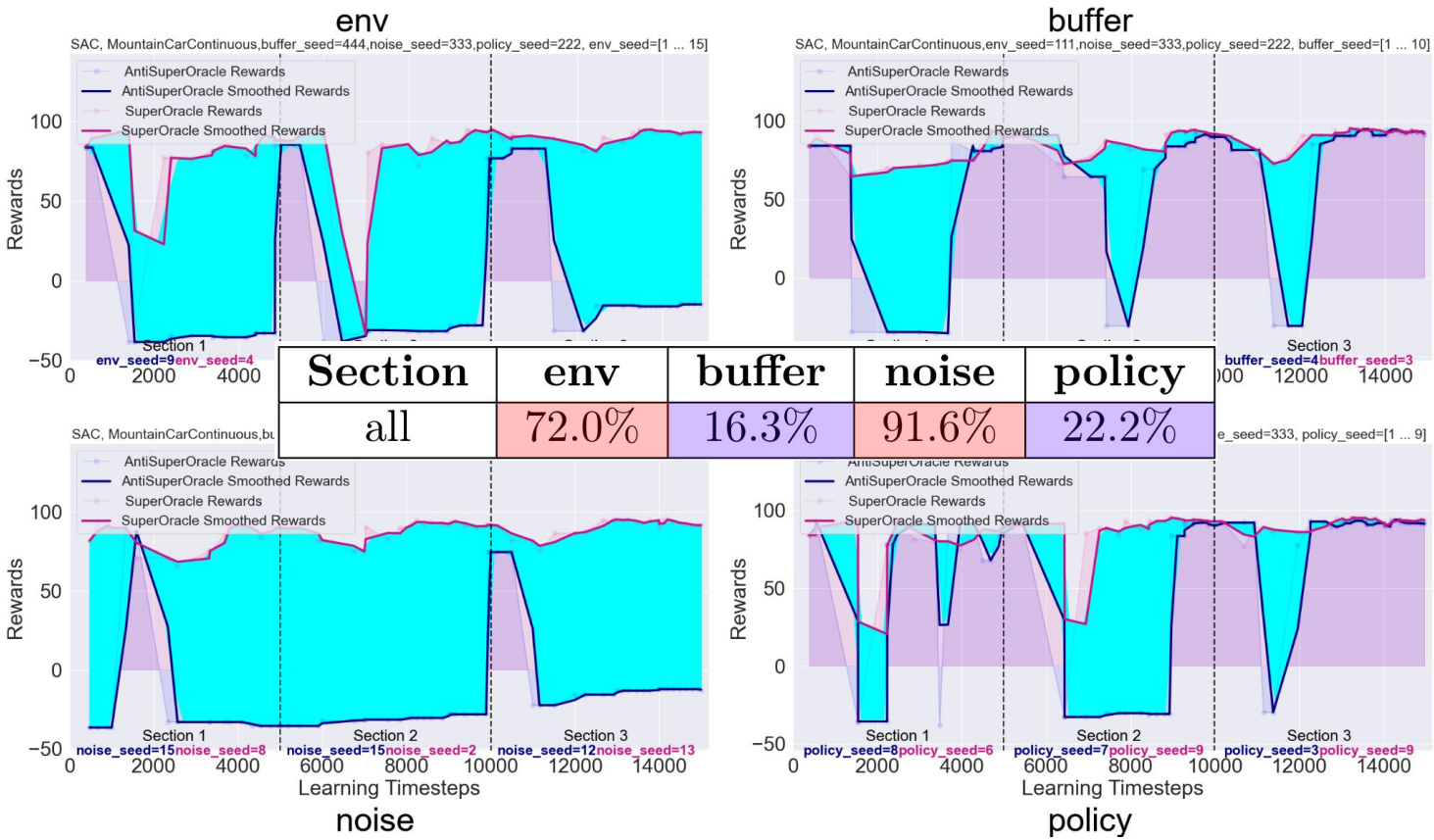
Section	env	buffer	noise	policy
all	9.9%	8.5%	8.5%	9.6%

the impact is consistently large

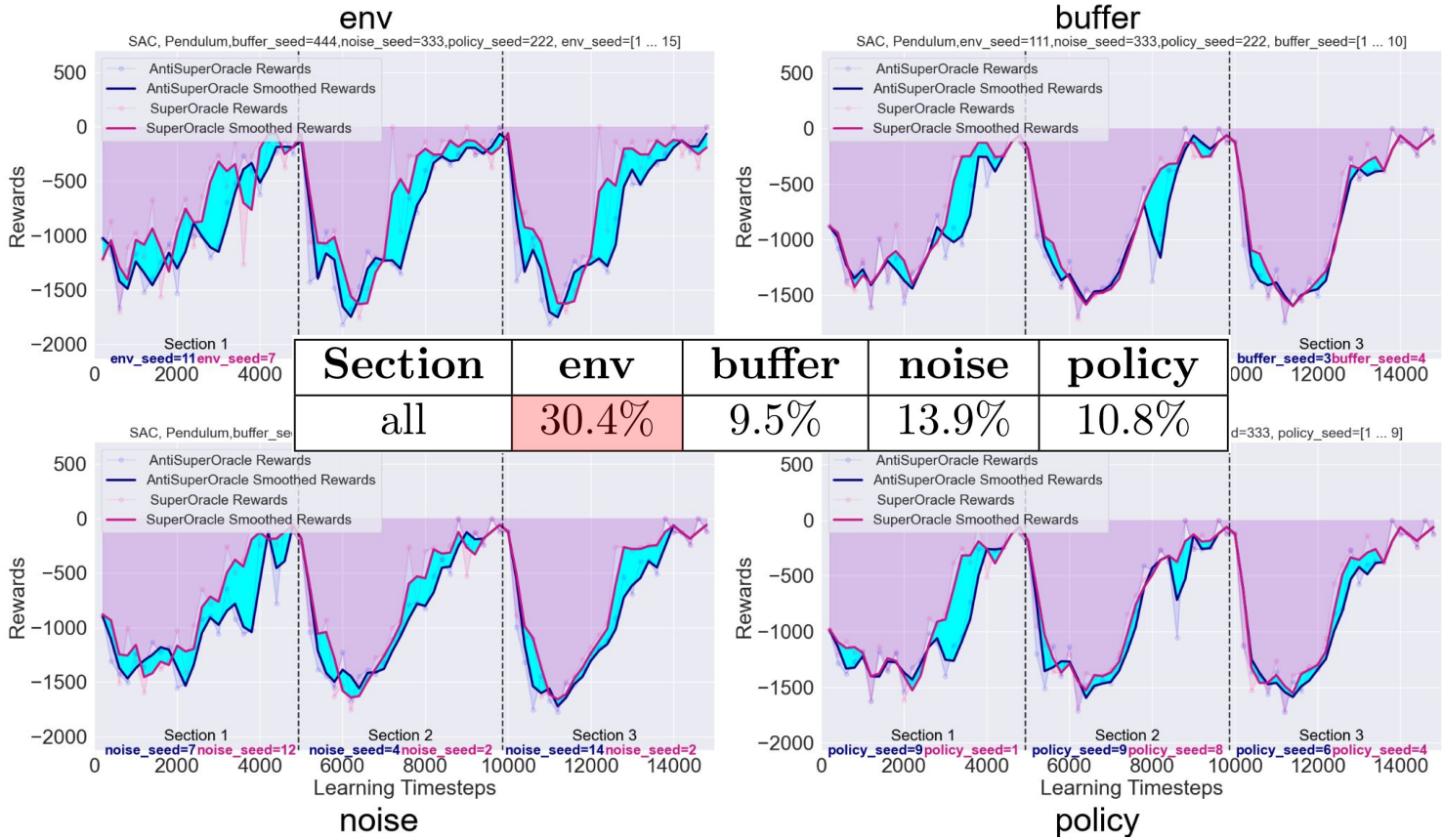


Section	env	buffer	noise	policy
all	65.8%	68.7%	75.6%	68.7%

impacts are significantly different

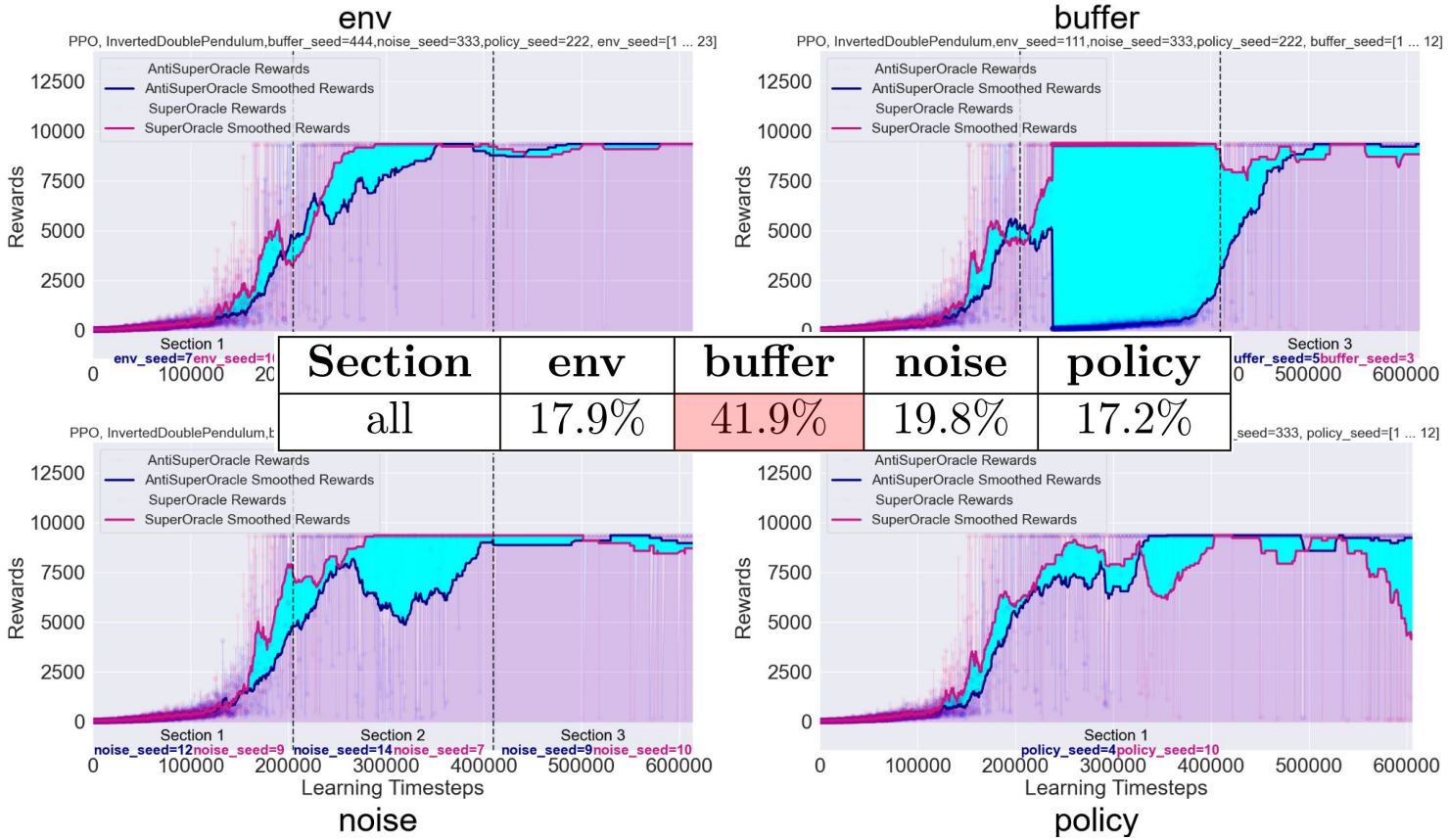


Leading source : environment



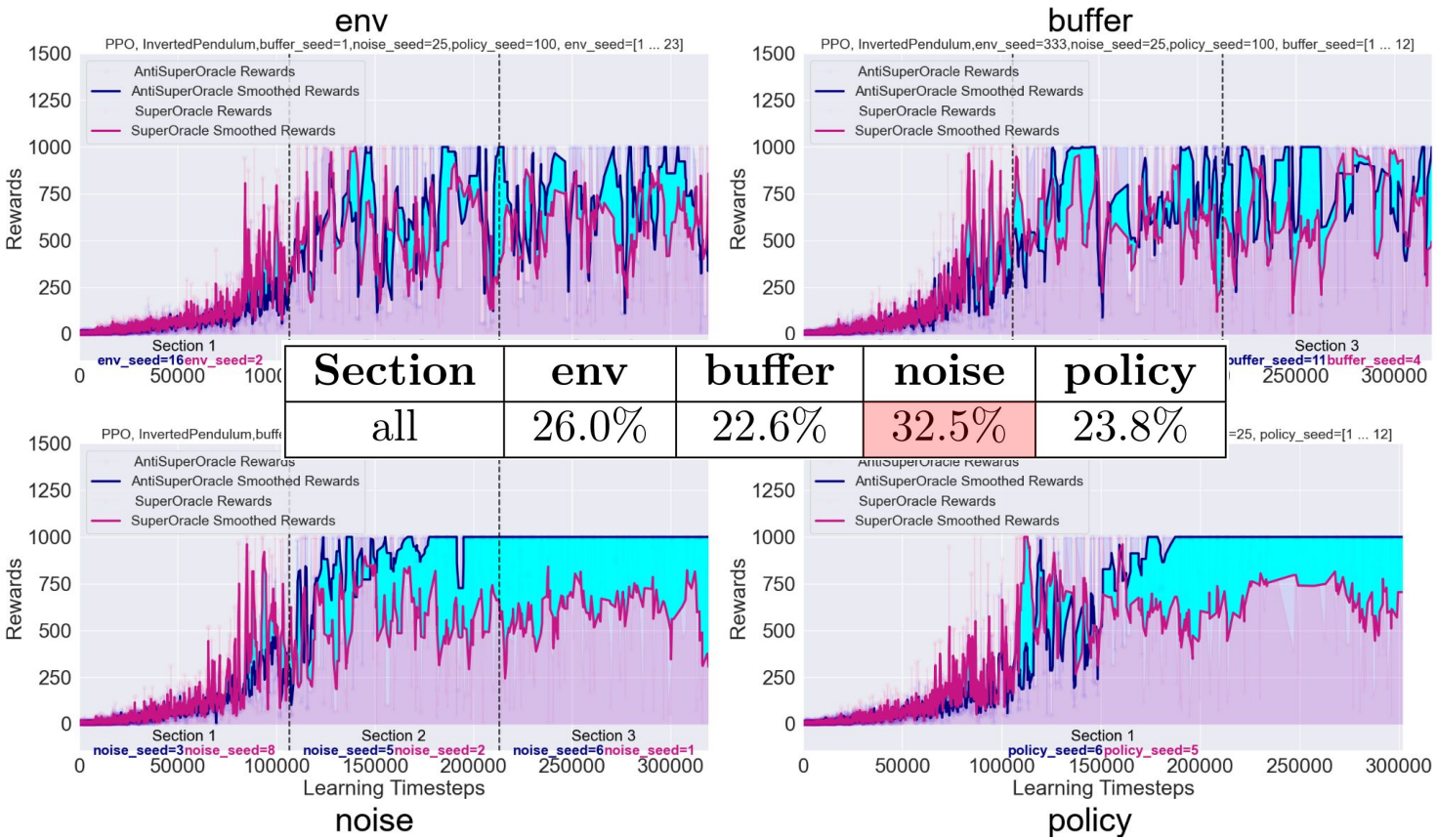
Section	env	buffer	noise	policy
all	30.4%	9.5%	13.9%	10.8%

Leading source : buffer



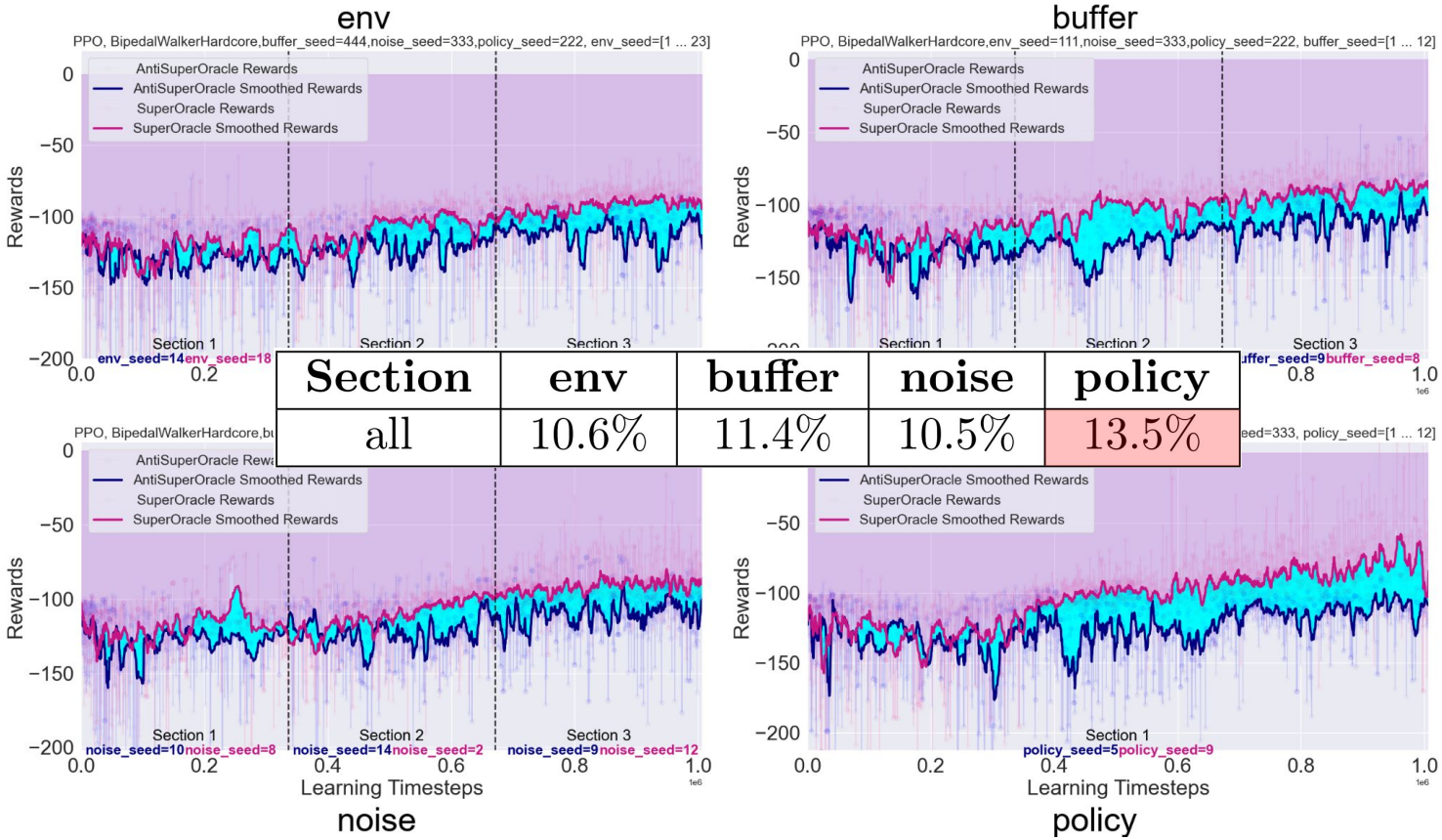
Section	env	buffer	noise	policy
all	17.9%	41.9%	19.8%	17.2%

Leading source : noise



Section	env	buffer	noise	policy
all	26.0%	22.6%	32.5%	23.8%

Leading source : policy



Section	env	buffer	noise	policy
all	10.6%	11.4%	10.5%	13.5%

Equality of Source Impacts

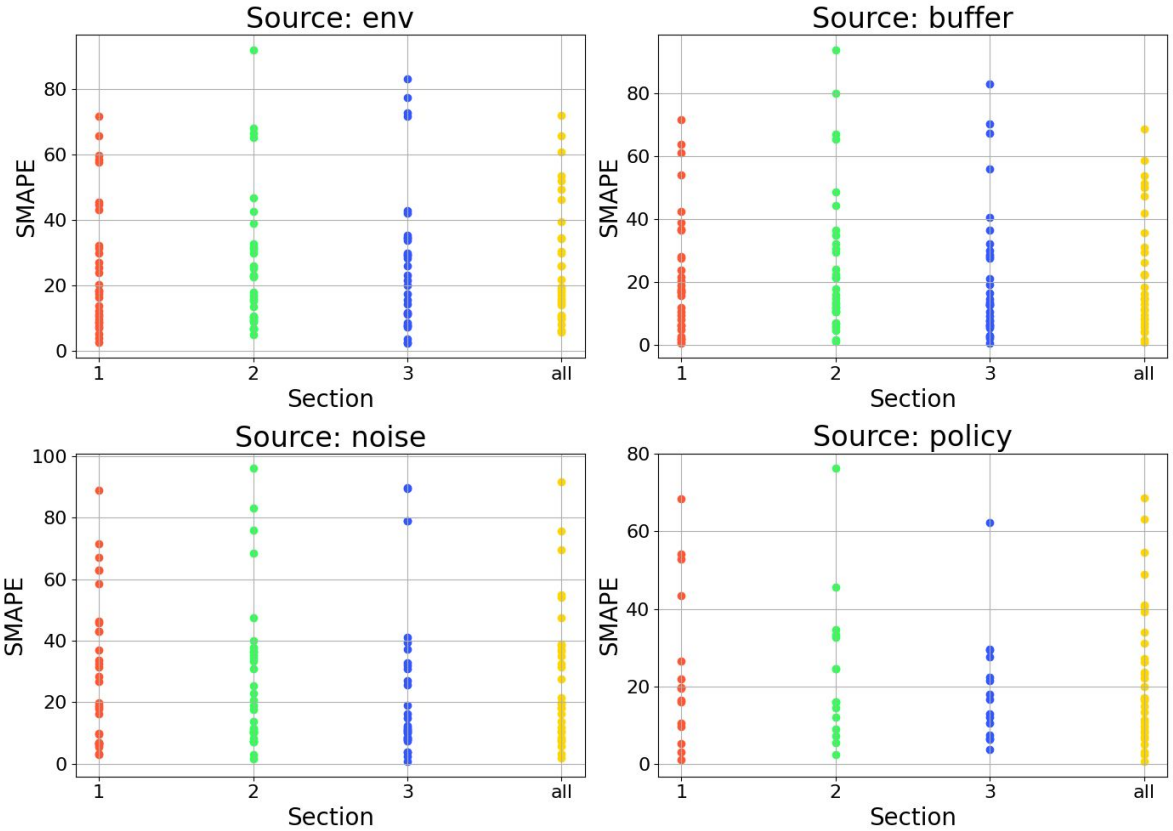
Section	Environment	SAC	PPO
all	Humanoid-v4	Equal	Equal
all	HumanoidStandup-v4	Equal	Equal
all	Walker2d-v4	Different	Equal
all	Ant-v4	Different	Equal
all	BipedalWalker-v3	Equal	Different
all	BipedalWalkerHardcore-v3	Equal	Equal
all	CarRacing-v2	-	Equal
all	HalfCheetah-v4	Equal	Different
all	Hopper-v4	Equal	Equal
all	InvertedDoublePendulum-v4	Equal	Different
all	InvertedPendulum-v4	Equal	Equal
all	LunarLanderContinuous-v2	Equal	Equal
all	MountainCarContinuous-v0	Different	Different
all	Pendulum-v1	Different	Different
all	Pusher-v4	Equal	Equal
all	Reacher-v4	Different	Different

65% of the time, the **impact of the sources** within environment **is nearly equal** (within a 10% SMAPE range)

Border values

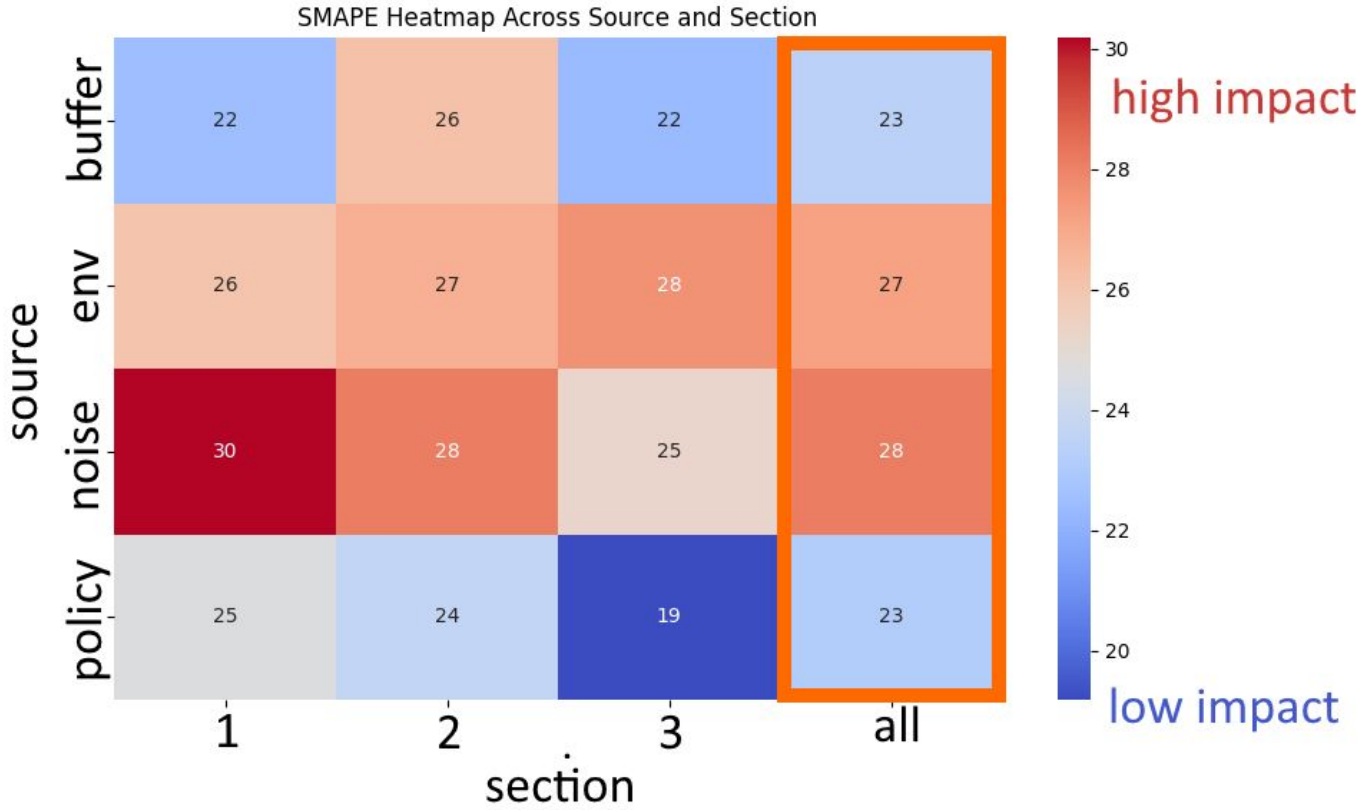
- The highest recorded impact reached 92% (SMAPE) by noise
- The lowest recorded impact accounted for only 0.6% (SMAPE) by policy

75th percentiles

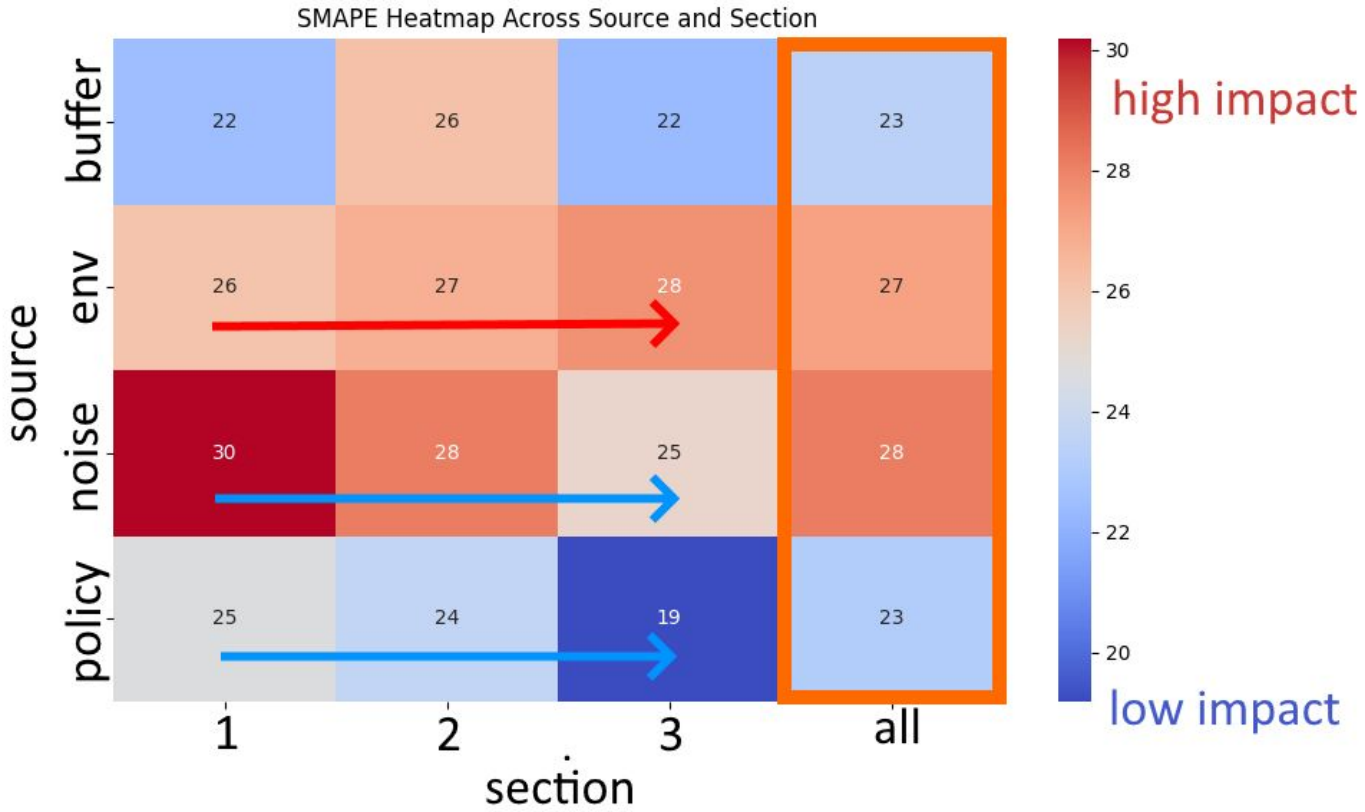


75% of the time, the **impact of any source** of randomness **falls below 45%** (SMAPE)

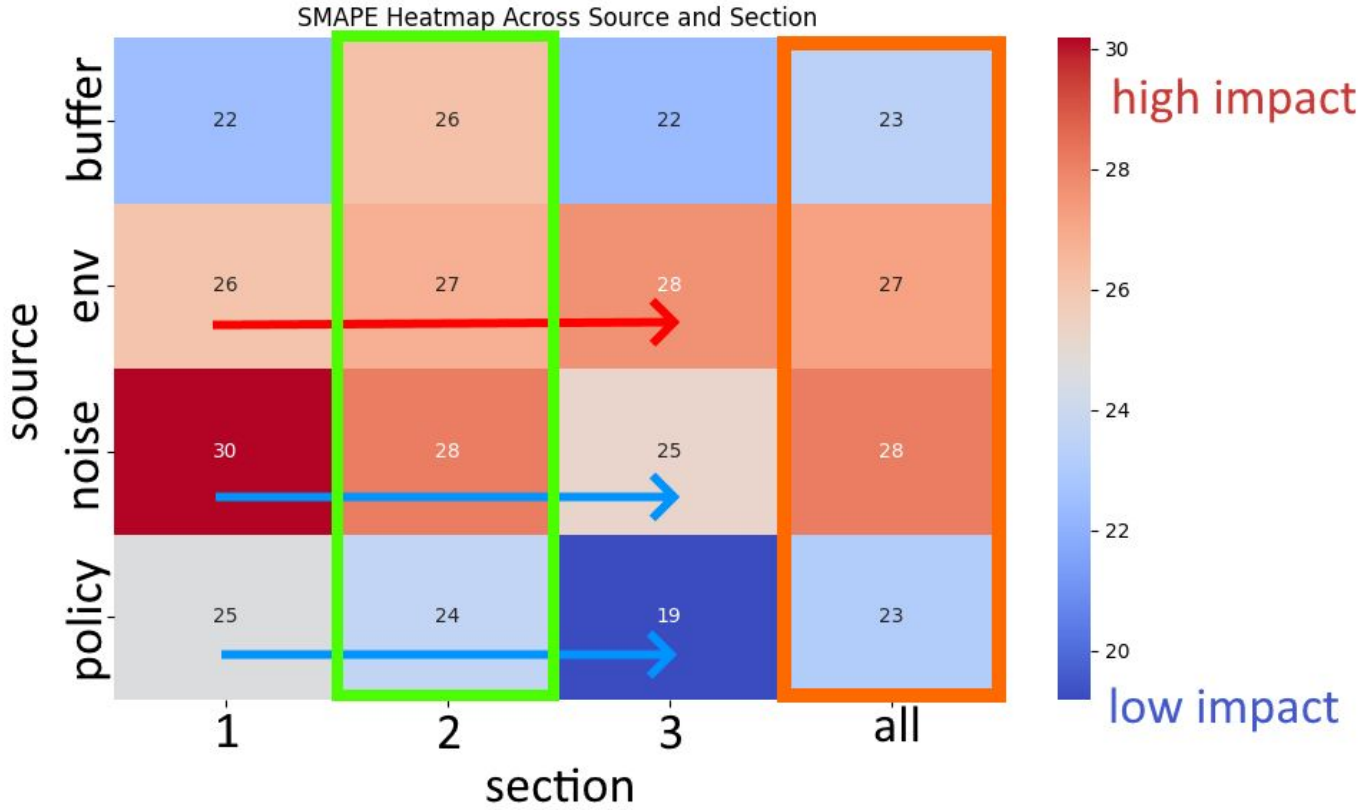
SMAPE Heatmap



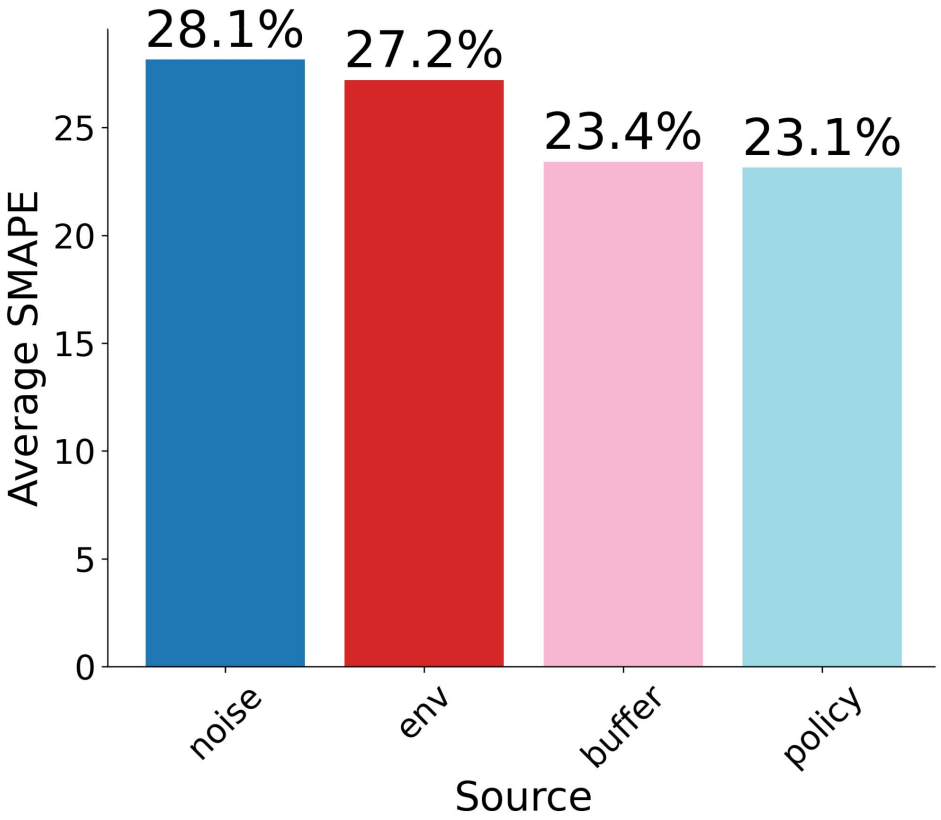
SMAPE Heatmap



SMAPE Heatmap



Conclusion

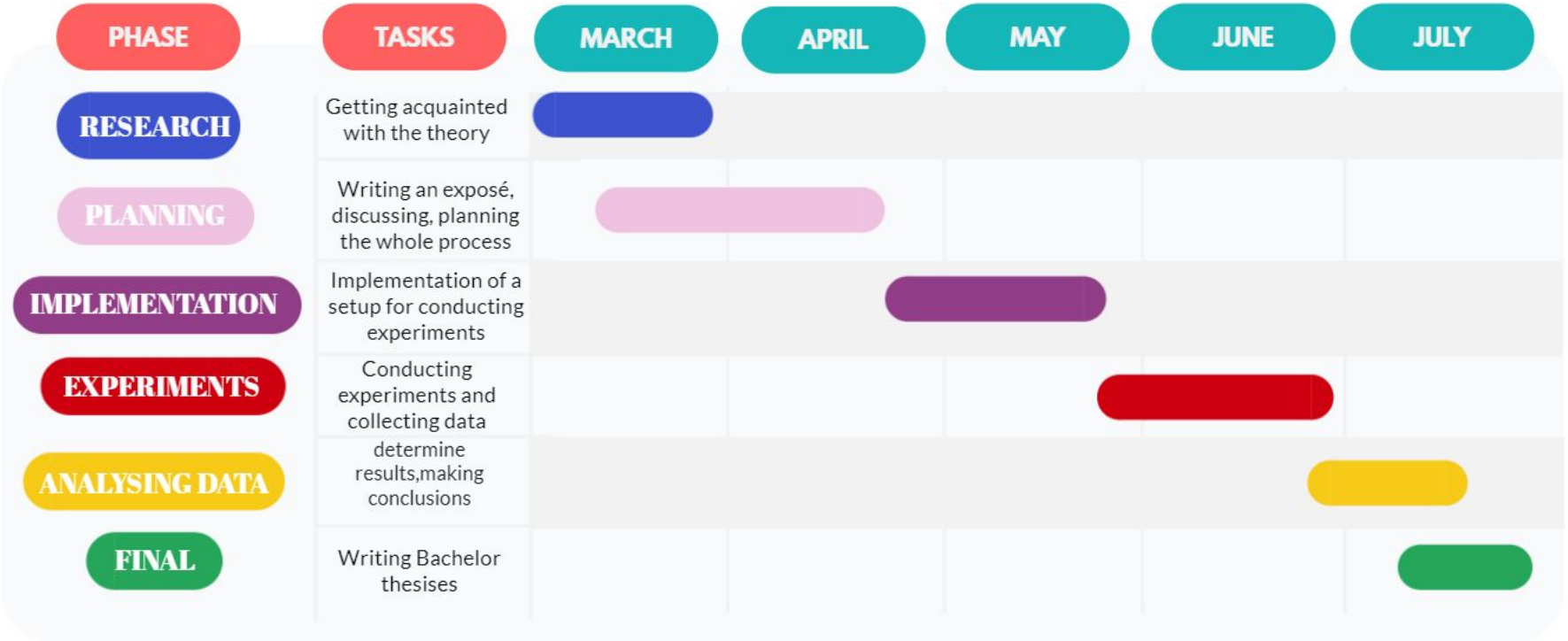


- **Noise > Env > Buffer > Policy**
- **In average all sources are nearly equal in their impact**

Questions?

(better not)

Timeline established in initial presentation



Future works

- investigate the **combined effects** on performance of **multiple sources of randomness**
- there is potential to **further subdivide the sources of randomness**
- experiments with different sets of **fixed seeds**
- **expand the range of seeds tested** for each source of randomness
- custom **division into sections** for different environments

Bibliography

[1] Saveriano, M. Reinforcement Learning (Lecture slides). Machine Learning.

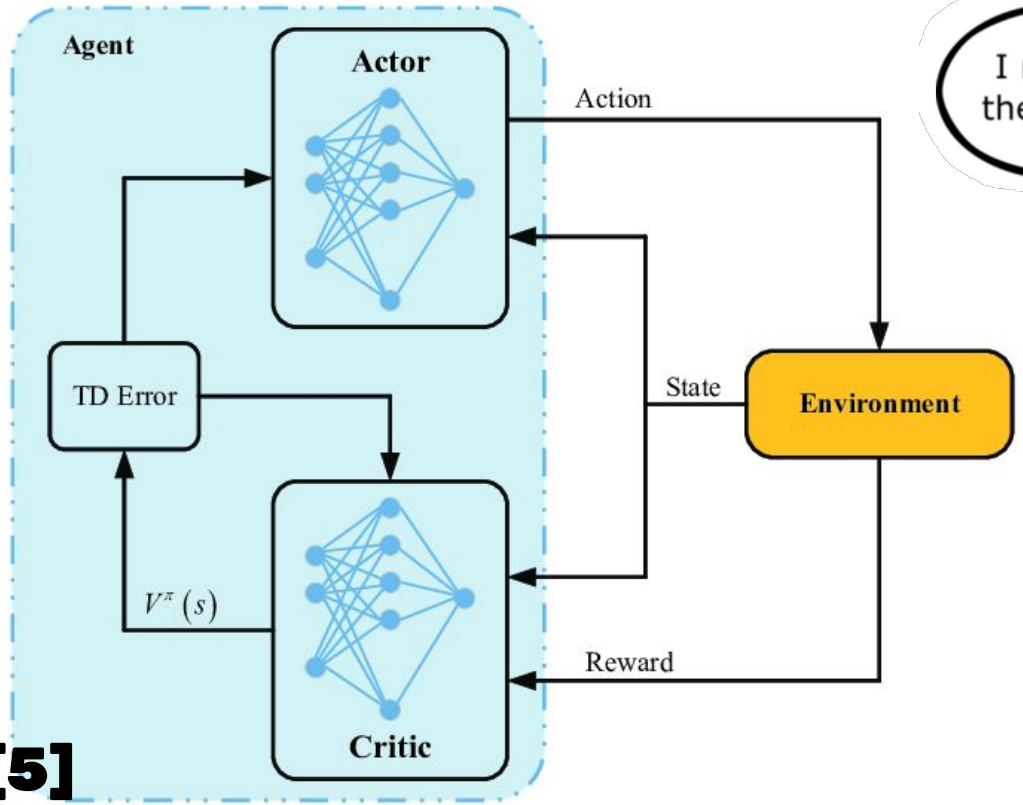
[2] Stable-Baselines3. (2024). Stable-Baselines3 Docs - Reliable Reinforcement Learning Implementations. Retrieved from : https://stable-baselines3.readthedocs.io/en/master/

Bibliography

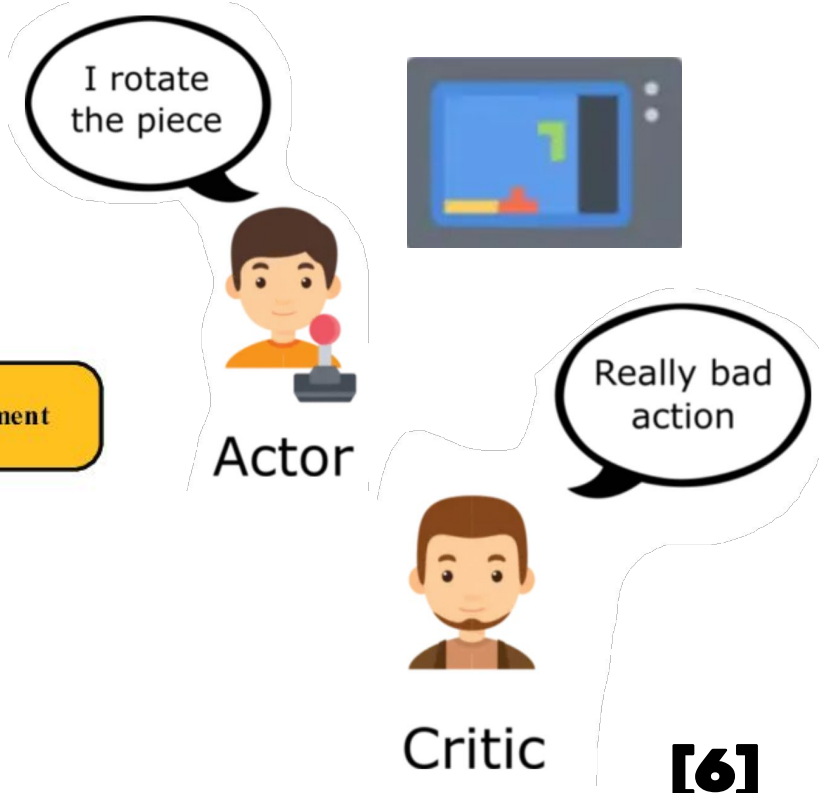
[3] Farama Foundation. Gymnasium: An api standard for reinforcement learning with a diverse collection of reference environments. Retrieved from https://gymnasium.farama.org/index.html, 2023.

[4] Gym Library. Mujoco environments. Retrieved from https://www.gymlibrary.dev/environments/mujoco/index.html, 2022.

Actor-Critic algorithm?

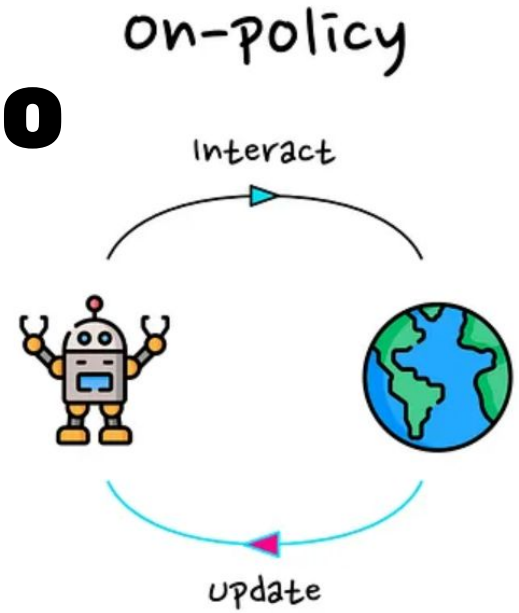


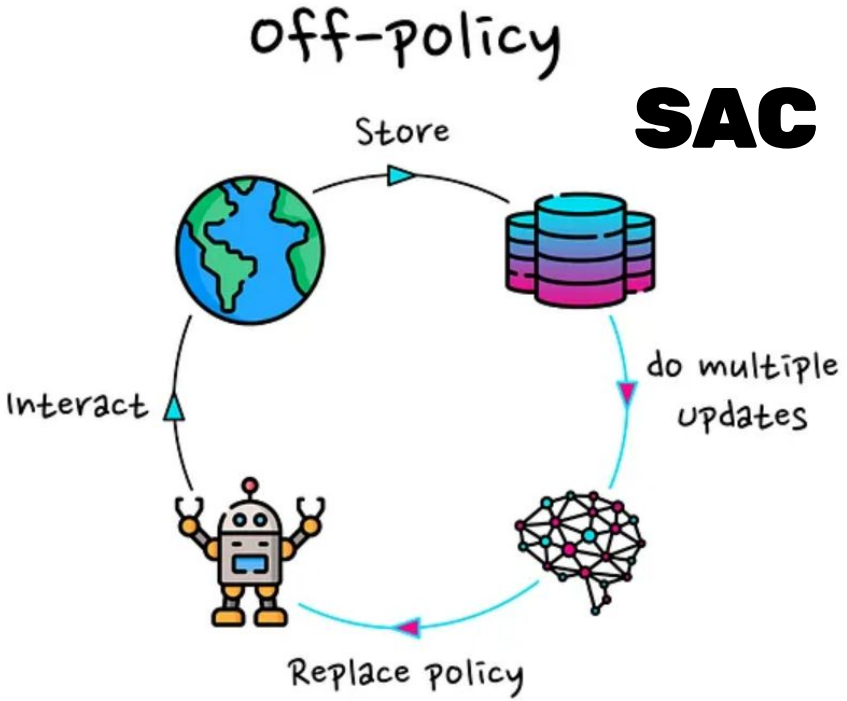
[5]



[6]

On-policy vs Off-policy methods?





[7]

On-policy vs Off-policy methods?

- **On-policy** : rely exclusively on the **most recent experiences** generated by the **current policy**

- **Off-policy** : can learn from a broader set of data, **including past experiences** collected under **different policies** , allowing them to store and reuse data over time.

SMAPE?

4.1 SMAPE

In this analysis, **the Symmetric Mean Absolute Percentage Error (SMAPE)** is employed to quantify the percentage difference between the learning curves of the best and worst performance cases. Originally SMAPE is a metric used to measure the accuracy of predictive models, particularly when comparing predicted values to actual values. It is calculated as follows:

$$\text{SMAPE} = \frac{1}{n} \sum_{i=1}^{n} \frac{2 \times |y_{\text{best},i} - y_{\text{worst},i}|}{|y_{\text{best},i}| + |y_{\text{worst},i}|} \times 100\%$$

where $y_{\text{best},i}$ represents the best performance value and $y_{\text{worst},i}$ represents the worst performance value at each time step i. The resulting SMAPE value represents the average percentage difference between these values, providing an intuitive measure of accuracy that is easy to interpret.

No CarRacing for SAC?

- It was not possible to run experiments with SAC for the CarRacing-v2 environment due to memory allocation issues.
- SAC, being an off-policy algorithm, relies on a replay buffer to store large amounts of experience data , including high-dimensional image states from CarRacing-v2.
- The attempt to store one million frames resulted in a memory allocation error. In contrast, PPO, an on-policy algorithm was able to run without encountering this issue.

Final seed range decision?

	env	buffer	noise	policy
SAC	[1 ... **15**]	[1 .. **10**]	[1 ... **16**]	[1 ... **9**]
PPO	[1 ... **23**]	[1 ... **12**]	[1 ... **15**]	[1 ... **12**]

Table 10: Final seed range decision

Bibliography

[5]
https://medium.com/geekculture/actor-critic-value-function-approximations-b8c118dbf723

[6]
https://www.researchgate.net/figure/The-structure-of-deep-actor-critic-reinforcement-learning_fig4_342358205

[7]
https://medium.com/@cedric.vandelaer/reinforcement-learning-an-introduction-part-3-4-e7d883dcbba2