





## Multi-Objective RL

Agent

Action  $A_t$ 

Environment



Next State  $S_{t+1}$ 

**Reward vector**  $\boldsymbol{R}_t = [R_1 \dots R_m]$ 



# 1. Environments MO-Gymnasium

- Standard API extending Gymnasium's API
- Part of the Farama Foundation
- MORL-specific wrappers
- Available on Pypi:

pip install mo-gymnasium

### Over 20 environments available!



#### Gymnasium's Half-Cheetah

ctrl cost = self.control cost(action) forward\_reward = self.\_forward\_reward\_weight \* x\_velocity reward = forward reward - ctrl cost

#### MO-Gymnasium's Half-Cheetah

reward = np.array([forward\_reward, -ctrl\_cost])

# A Toolkit for Reliable Benchmarking and Research in Multi-Objective Reinforcement Learning

Florian Felten\* Lucas N. Alegre\* Ann Nowé Ana L. C. Bazzan El Ghazali-Talbi Grégoire Danoy Bruno C. da Silva

Goal: Learn a set of policies, each specialized in a different preference over the *m* conflicting objectives.



• •	* * *		2	•
	•	2	•	Ø
•	3. ° •	•	÷	•
	*		$\varphi^{(1)} = \varphi^{(2)}$	*
÷.*.+	· •		· •	
*		*		40.0
	2.1	TAT		
	÷		÷	

# 2. Algorithms

# **MORL-Baselines**

Sol. 1

0

25

- Over 10 MORL algorithms implemented Clean, tested, and maintained code Automated logging and metrics Utilities for designing new algorithms

Algorithm	Single or	Utitlity	Observation	Action	
	multi-policy	function	space	space	
MOQL (Van Moffaert et al., 2013a)	Single	Linear	Disc.	Disc.	
EUPG (Rojjers et al. 2018a)	Single	Non-linear,	Disc	Disc.	
Lor O (Rohers et al., 2010a)	Single	ESR	Disc.		
MPMOQL (Van Moffaert et al., 2013a)	Multi	Linear	Disc.	Disc.	
POL (Van Moffgert and Nowé 2014)	Mult:	Non-linear,	Disc	Disc	
r QL (van Monaert and Nowe, 2014)	Iviuiti	SER (*)	Disc.	Disc.	
OLS (Roijers, 2016)	Multi	Linear	/ (**)	/ (**)	
Envelope (Yang et al., 2019)	Multi	Linear	Cont.	Disc.	
PGMORL (Xu et al., 2020)	Multi	Linear	Cont.	Cont.	
PCN (Reymond et al. 2022)	Multi	Non-linear,	Cont	Disc.	
r Civ (Reymond et al., 2022)	Iviuiti	ESR/SER (*)	Cont.		
GPI-LS &	Multi	Lincor	Cont	Any	
GPI-PD (Alegre et al., 2023)	iviuiti	Lineai	Cont.	Ally	
CAPQL (Lu et al., 2023)	Multi	Linear	Cont.	Cont.	



![](_page_0_Picture_39.jpeg)

- - comparison

![](_page_0_Picture_45.jpeg)

![](_page_0_Figure_46.jpeg)

![](_page_0_Figure_47.jpeg)

$\leftrightarrow$	C ( wandb.ai/openrlbenchmark/MORL-Baseli
(j Overview	Runs (540)
Workspace	Q .* =
	Name (40 visualized)
Runs	● – env_id: "deep-sea-treasure-concav
לא Jobs	Odeep-sea-treasure-concave-v0Mul
\$	@ deep-sea-treasure-concave-v0Mul
Automat.	@ deep-sea-treasure-concave-v0Mul
Sweeps	@ deep-sea-treasure-concave-v0Mul
Reports	deep-sea-treasure-concave-v0Mul
	deep-sea-treasure-concave-v0Mul
Artifacts	@ deep-sea-treasure-concave-v0Mul
	@ deep-sea-treasure-concave-v0Mul
	@ deep-sea-treasure-concave-v0Mul
	deep-sea-treasure-concave-v0Mul
	< 01-10 of 40 >
	<pre>&gt;&gt; env_id: "mo-mountaincarcontinue" 1 16</pre>
	Penv_id: "fruit-tree-vo"
	My Workspace

![](_page_0_Picture_49.jpeg)

![](_page_0_Picture_50.jpeg)

![](_page_0_Picture_51.jpeg)

florian.felten@uni.lu Inalegre@inf.ufrgs.br

![](_page_0_Picture_53.jpeg)

@FlorianFelten1

#### Motivation

Current challenges in MORL:

Lack of standardized environments Missing ready-to-use algorithms Need to re-run expensive baselines for

## 3. Benchmark Results

![](_page_0_Picture_59.jpeg)

All MORL-Baselines trained on all MO-Gymnasium environments Weights & Biases dashboard Tracked code version and hyperparameters Integrated with openrlbenchmark providing CLI for plotting

PCN Ma 1.0 0.5 0.0 0M	Envelope GPI-LS ax. Utility Loss (↓) Hypervolu 600 400 200 0.1M 0.2M 0M 0.1M Training steps	- GPI-PD ime ( $\uparrow$ ) 0.75 0.50 0.25 0.25 0.1M 0.2M
nes/workspace?workspace	e=user-florian-felten          Image: Search panels         runs.summary["eval/front"]         Image: Summary["eval/front"]         Image: Summary["eval/front"] <t< th=""><th>Image: Solution of the second seco</th></t<>	Image: Solution of the second seco
ItiPolicy MO QItiPolicy MO	eval/sparsity - algo: MultiPolicy MO Q-Learning (GPI-LS) _ algo: Pareto Q-Learning - algo: MultiPolicy MO Q-Learning _ algo: MultiPolicy MO Q-Learning (OLS) 1000 40	eval/igd         - algo: MultiPolicy M0 Q-Learning (CPT-LS) - algo: Pareto Q-Learning (OLS)         - algo: MultiPolicy M0 Q-Learning - algo: MultiPolicy M0 Q-Learning (OLS)         - algo: MultiPolicy M0 Q-Learning - algo: MultiPolicy M0 Q-Learning (OLS)         - algo: MultiPolicy M0 Q-Learning - algo: MultiPolicy M0 Q-Learning (OLS)         - algo: MultiPolicy M0 Q-Learning - algo: MultiPolicy M0 Q-Learning (OLS)         - algo: MultiPolicy M0 Q-Learning - algo: MultiPolicy M0 Q-Learning (OLS)         - algo: MultiPolicy M0 Q-Learning - algo: MultiPolicy M0 Q-Learning (OLS)         - algo: MultiPolicy M0 Q-Learning - algo: MultiPolicy M0 Q-Learning (OLS)         - algo: MultiPolicy M0 Q-Learning - algo: MultiPolicy M0 Q-Learning (OLS)         - algo: MultiPolicy M0 Q-Learning - algo: MultiPolicy M0 Q-Learning (OLS)         - algo: MultiPolicy M0 Q-Learning - algo: MultiPolicy M0 Q-Learning (OLS)         - algo: MultiPolicy M0 Q-Learning - algo: MultiPolicy M0 Q-Learning (OLS)         - algo: MultiPolicy M0 Q-Learning - algo: MultiPolicy M0 Q-Learning (OLS)         - algo: MultiPolicy M0 Q-Learning - algo: MultiPolicy M0 Q-Learning (OLS)         - algo: MultiPolicy M0 Q-Learning - algo: MultiPolicy M0 Q-Learning (OLS)         - algo: MultiPolicy M0 Q-Learning - algo: MultiPolicy M0 Q-Learning (OLS)         - algo: MultiPolicy M0 Q-Learning - algo: MultiPolicy M0 Q-Learning (OLS)         - algo: MultiPolicy M0 Q-Learning