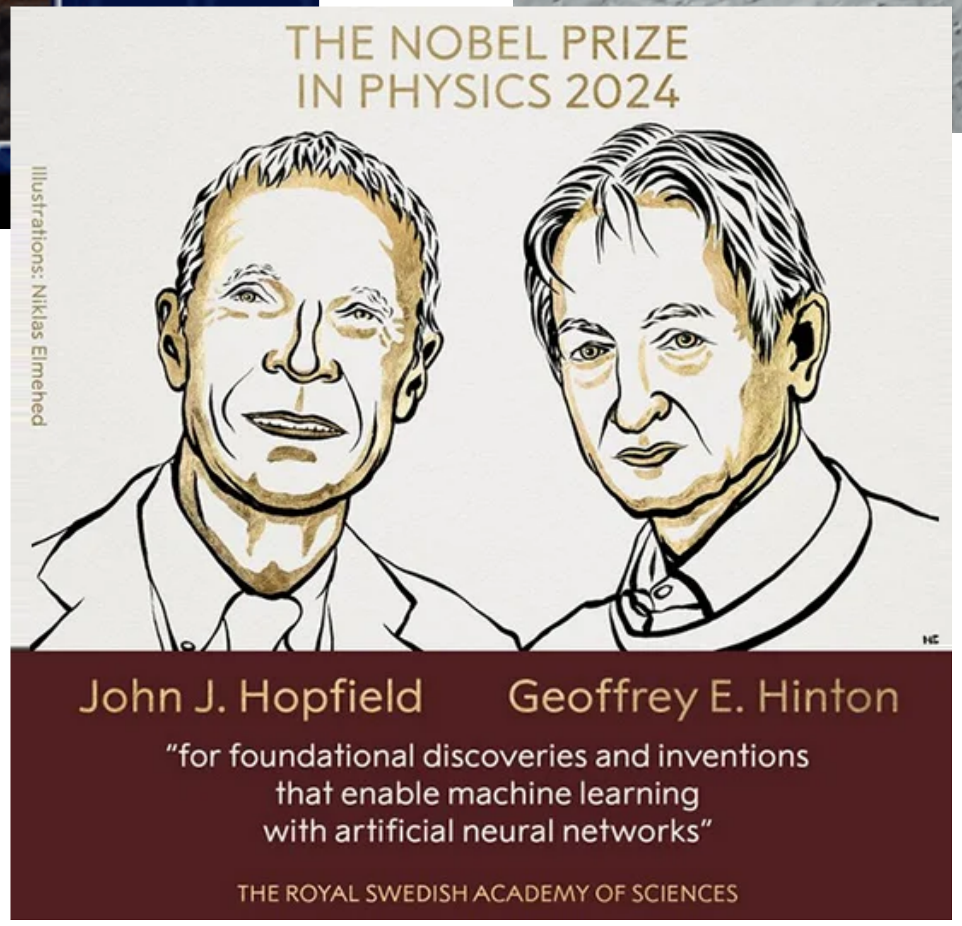
“It depends”: Dealing with Multiple Objectives in (MA)RL

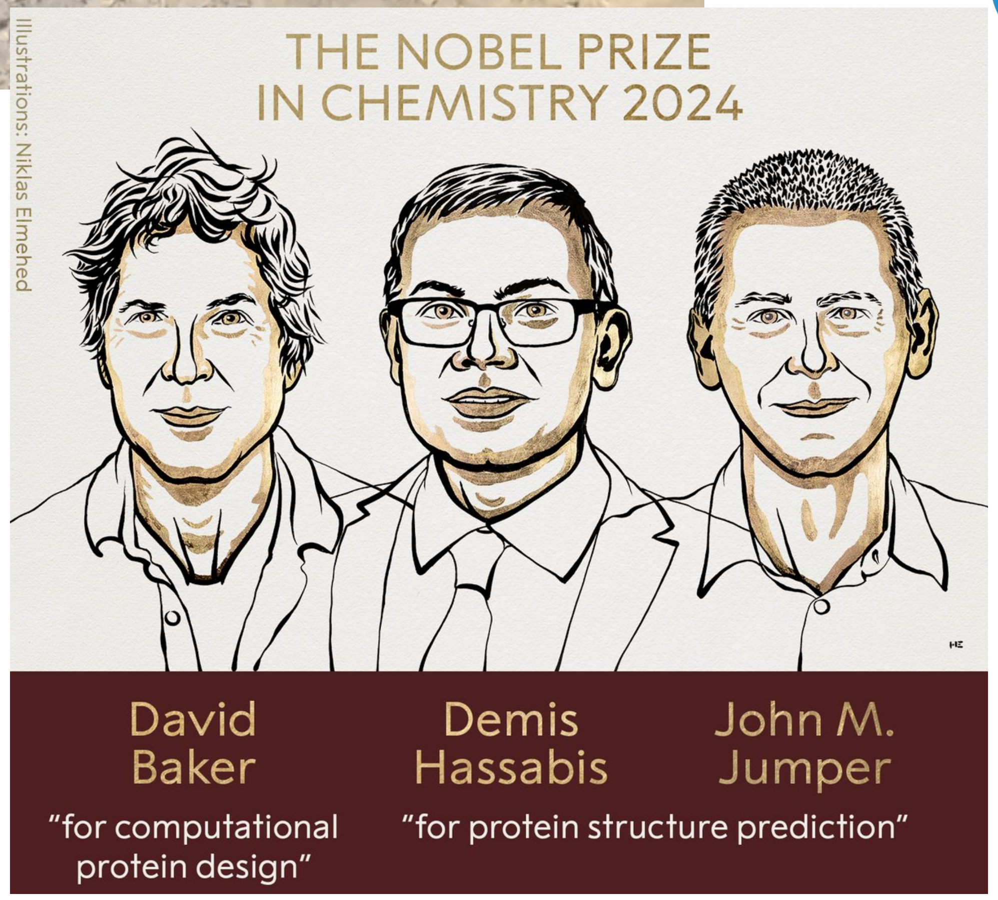
Florian Felten
ffelten@mavt.ethz.ch
ffelten

UNIVERSITÉ DU LUXEMBOURG

ETH zürich

The recent rise of Artificial Intelligence (AI)





Meta AI

MISTRAL AI_

OpenAI

cohere

Google DeepMind

[1] Silver, D. et al. "Mastering the game of Go without human knowledge." Nature, 2017.

[2] Smith, L. et al. "A Walk in the Park: Learning to Walk in 20 Minutes With Model-Free Reinforcement Learning." Proc. of the XIXth Conference on Robotics: Science and Systems, 2023.

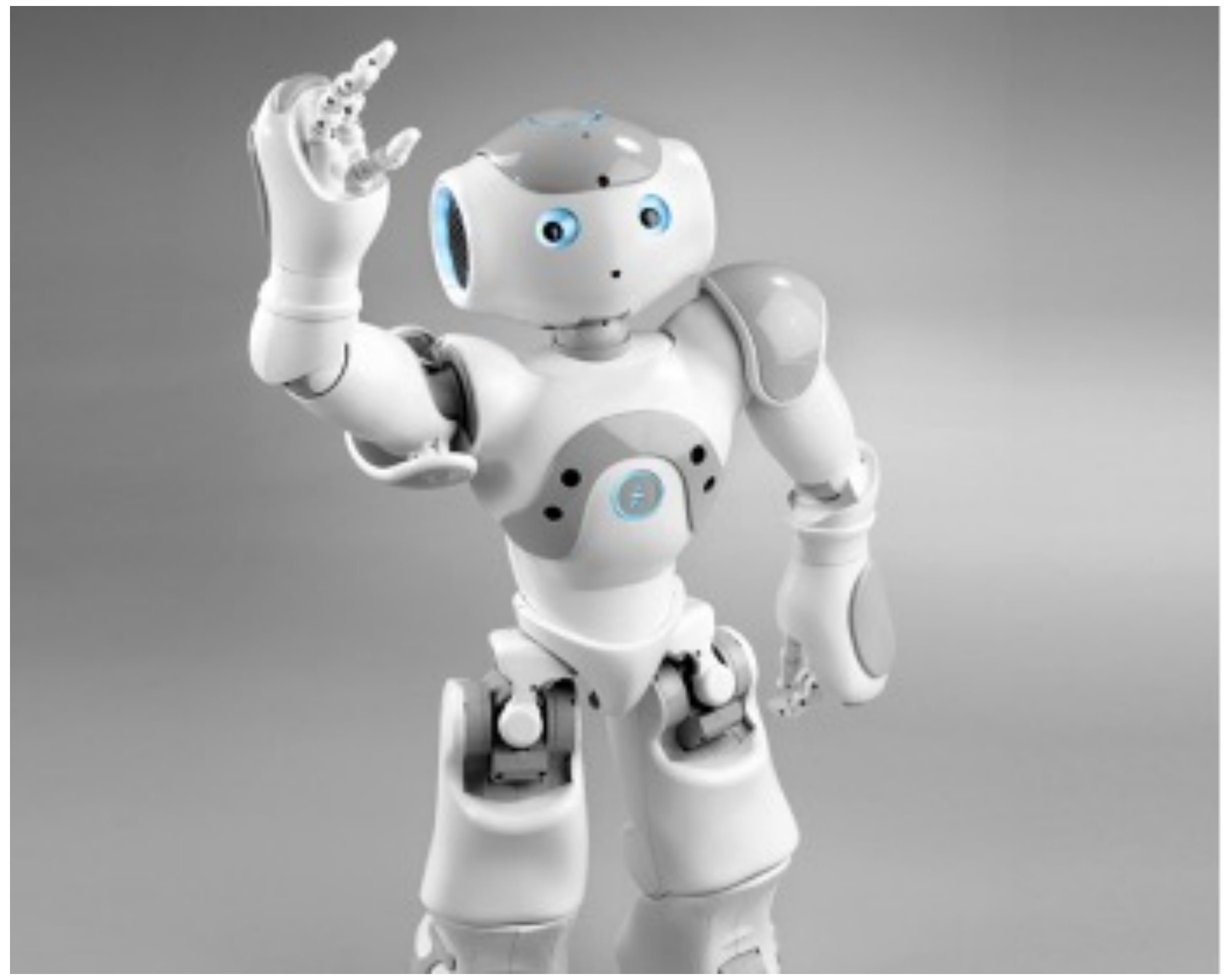
[3] https://www.nobelprize.org/prizes/lists/all-nobel-prizes/

Reinforcement Learning (RL)

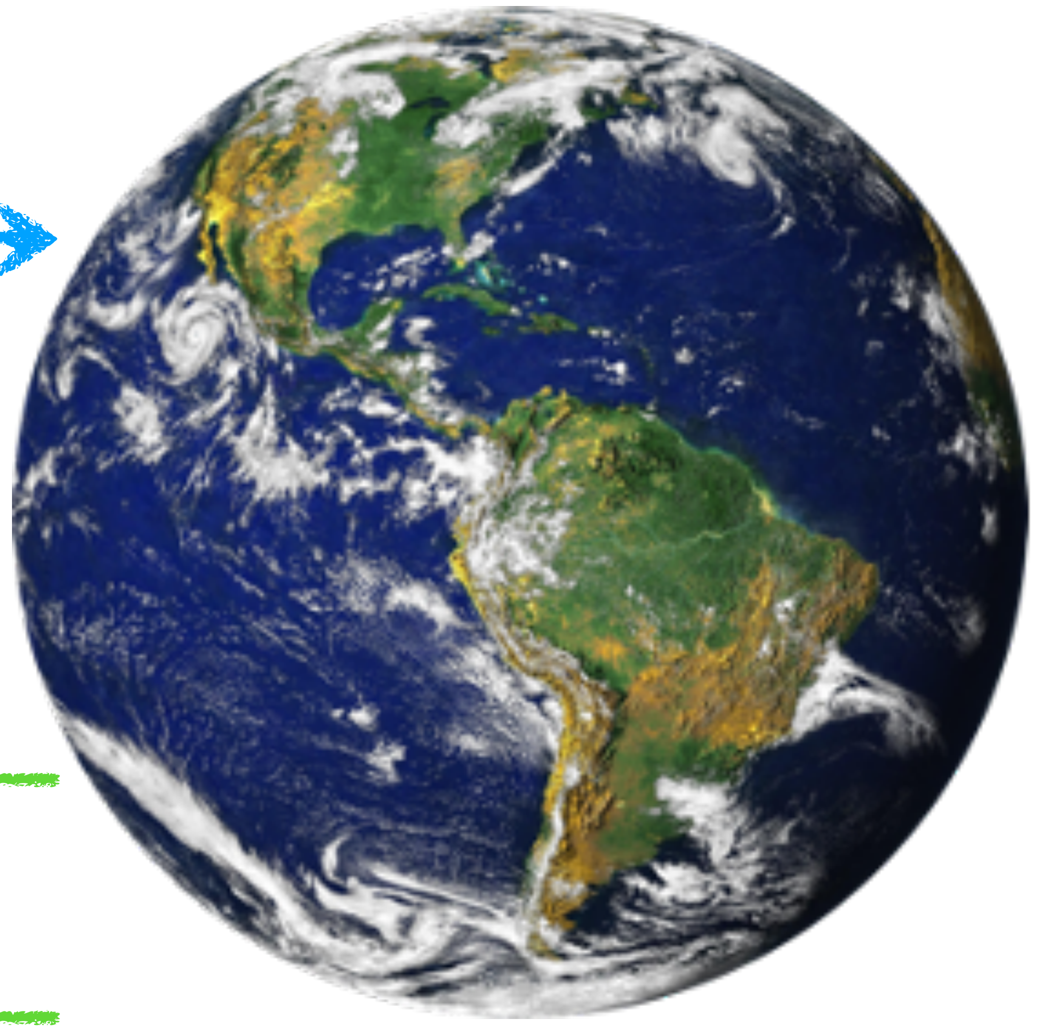
A key technique behind these advances

Markov Decision Process (MDP)

Agent



Environment



Action

State

Next state

Reward

🎯 **Learn to associate states to rewarding actions: a policy**

Sutton & Barto. "Reinforcement Learning: An Introduction.", 2018, MIT Press.

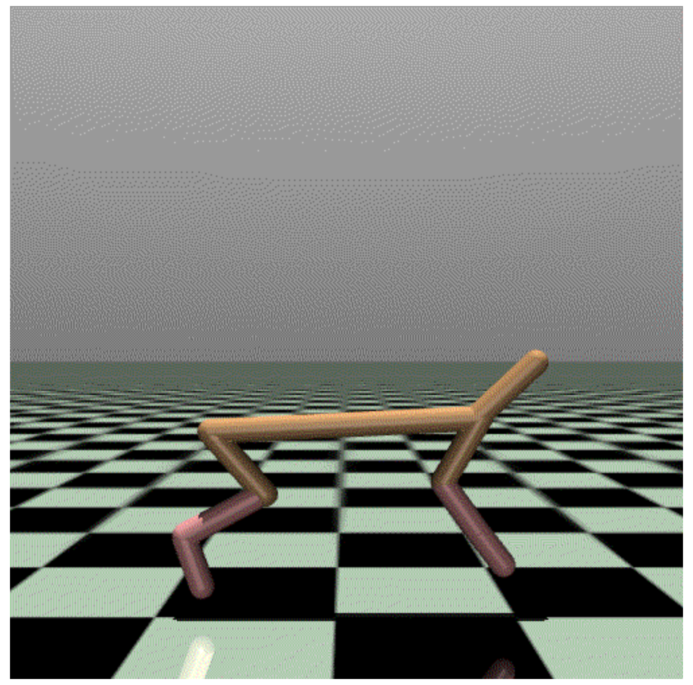
The need to consider multiple objectives

Games

Reward:

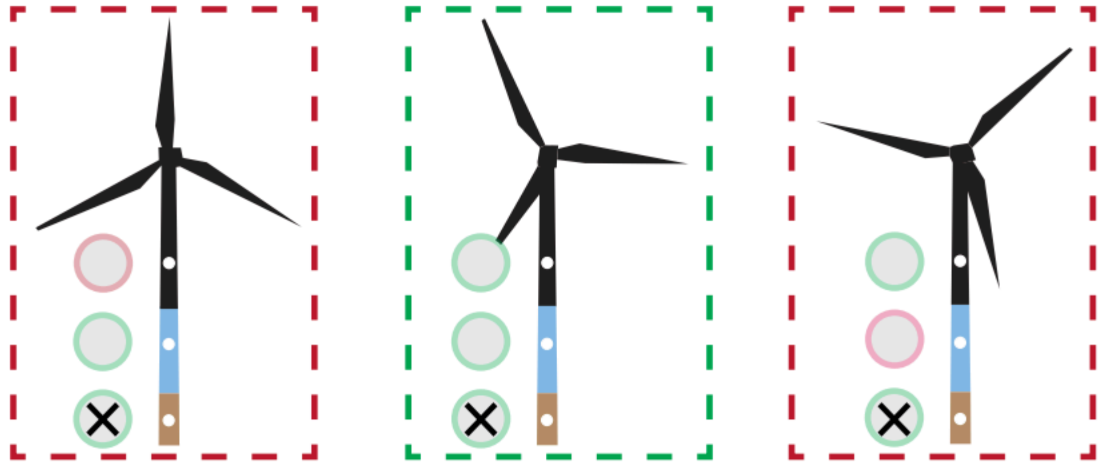
+1 if win, 0 if draw, -1 if lose

Real-world applications



Reward:

Speed vs.
energy consumption



[2]

Reward:

Risk vs. costs

[1] Vamplew, P. et al., "Scalar reward is not enough: a response to Silver, Singh, Precup and Sutton (2021)," Autonomous Agents and Multi-Agent Systems, 2022.

[2] P. Leroy, P. G. Morato, J. Pisane, A. Kolios, and D. Ernst, "IMP-MARL: a Suite of Environments for Large-scale Infrastructure Management Planning via MARL," NeurIPS, 2023.

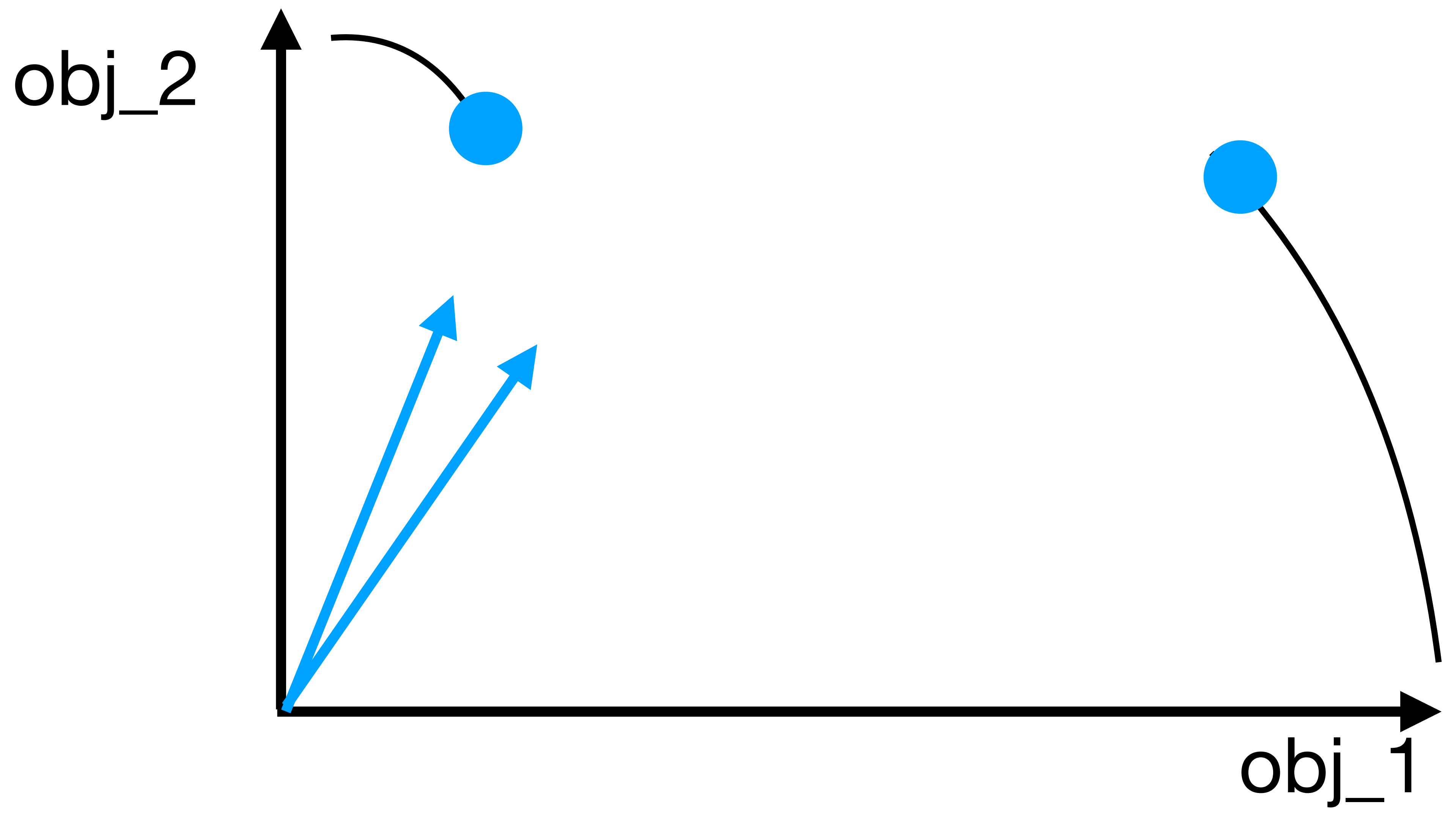
Traditional approach in RL

The trial and error

While not happy:

1. Set a weight/“importance” to each objective
2. Scalarize the objectives: **0.3** `* obj_1 +` **0.7** `* obj_2`
3. Train the RL agent ⬅️ This takes hours or days
4. Look at the resulting behavior

This is decided by the engineer, not the end user



We can do better !

Today’s menu

How to do better than the trial and error.

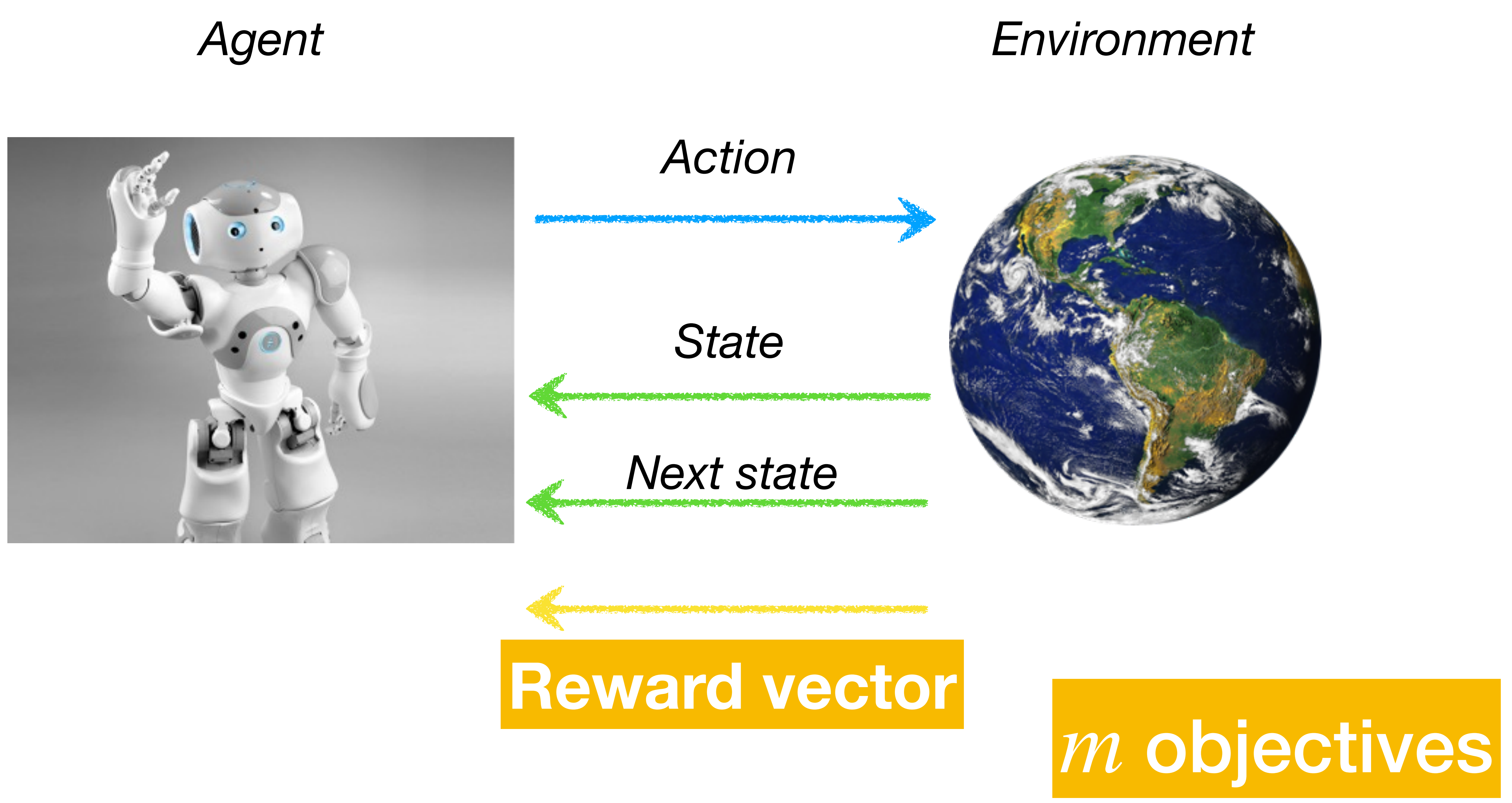
1. Single Agent MORL
2. Multi-Agent MORL
3. Example application

A glimpse of:

- Solution concepts
- Naive baselines
- Algorithmic improvements
- Tooling

1. Multi-Objective RL

Setup



[1] Roijers, D. et al., "A Survey of Multi-Objective Sequential Decision-Making," Journal of Artificial Intelligence, 2013.

[2] Hayes, C. et al., "A practical guide to multi-objective reinforcement learning and planning," Autonomous Agents and Multi-Agent Systems, 2022.

Optimal policy?

Optimal policy in single-objective RL
$$\pi^* = \arg\max_{\pi} \underbrace{\mathbb{E}_{a_t\sim\pi(s_t)}}_{\text{Averaged over various episodes}} \underbrace{\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t, s_{t+1}) \mid s_0\right]}_{\text{Discounted sum of rewards over one episode obtained by following the policy}}$$

… with vectorial rewards
$$\pi^* = \arg\max_{\pi} \mathbb{E}_{a_t\sim\pi(s_t)} \left[\sum_{t=0}^{\infty}\gamma^t \mathbf{r}(s_t, a_t, s_{t+1}) \mid s_0\right]$$

argmax is not defined on vectors…

💡 we can use a function $g : \mathbb{R}^m \mapsto \mathbb{R}$ that captures the user preferences to **scalarize the reward vector** (if we know them at training time)

Most common example: weighted sum $\sum_{i=1}^{m} \omega_i r_i$

Non-linear scalarization

… with vectorial rewards $\pi^* = \arg\max_\pi \mathbb{E}_{a_t \sim \pi(s_t)} \left[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}(s_t, a_t, s_{t+1}) \mid s_0 \right]$

Scalarizing after expectation
$$\pi^*_{\mathrm{SER}} = \arg\max_\pi g\left(\mathbb{E}_{a_t \sim \pi(s_t)} \left[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}(s_t, a_t, s_{t+1}) \mid s_0 \right] \right)$$

$$\neq$$

Scalarizing before expectation
$$\pi^*_{\mathrm{ESR}} = \arg\max_\pi \mathbb{E}_{a_t \sim \pi(s_t)} \left[g\left(\sum_{t=0}^{\infty} \gamma^t \mathbf{r}(s_t, a_t, s_{s+1}) \right) \mid s_0 \right] .$$

SER: when you want the agent to behave on average over various episodes, e.g., investing

ESR: when you want each application of the policy to be good, e.g., cancer detection

D. Roijers, D. Steckelmacher, and A. Nowe, "Multi-objective Reinforcement Learning for the Expected Utility of the Return," ALA workshop at ICML/AAMAS/IJCAI, 2018.

What if you don't know the user preferences at training time?

Solution concepts

MORL

Known utility

Unknown utility

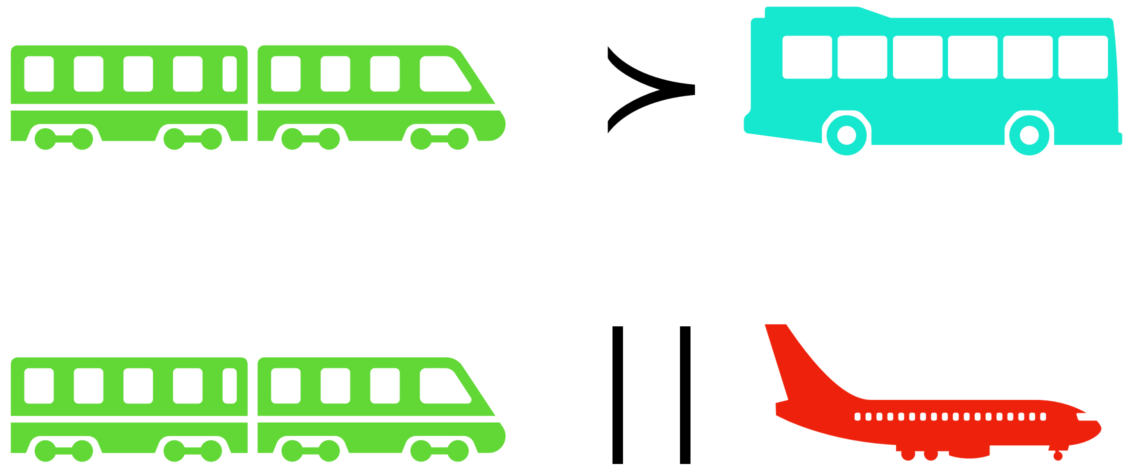
Scalarized MORL with
ESR vs. SER

Multi-policy MORL

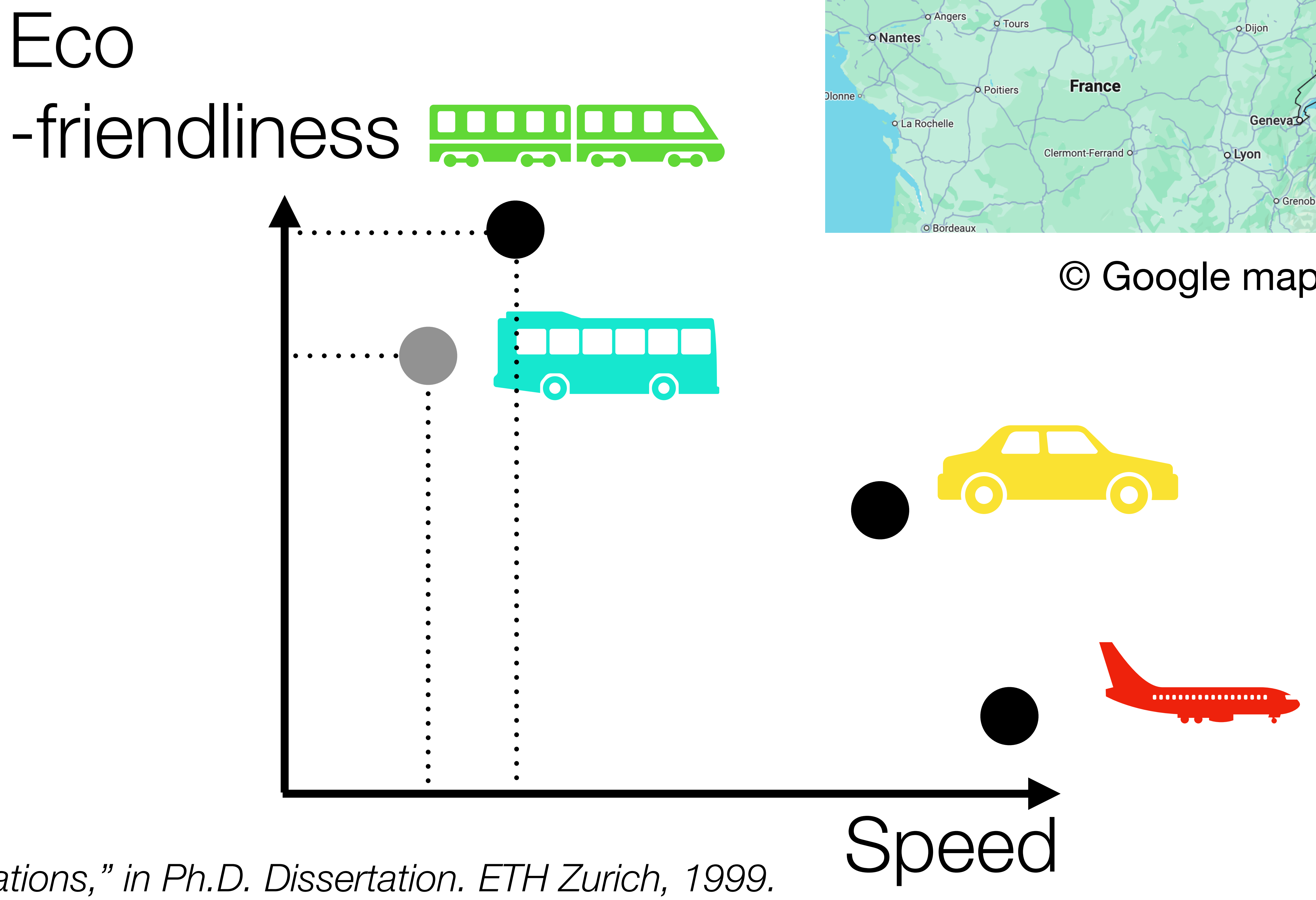
Multi-objective optimization (MOO)

Aims at **finding an assignment of decision variables** (≠ learning)**.**

Returns a solution set based on **Pareto optimality** [1,2]

dominated 

non-dominated



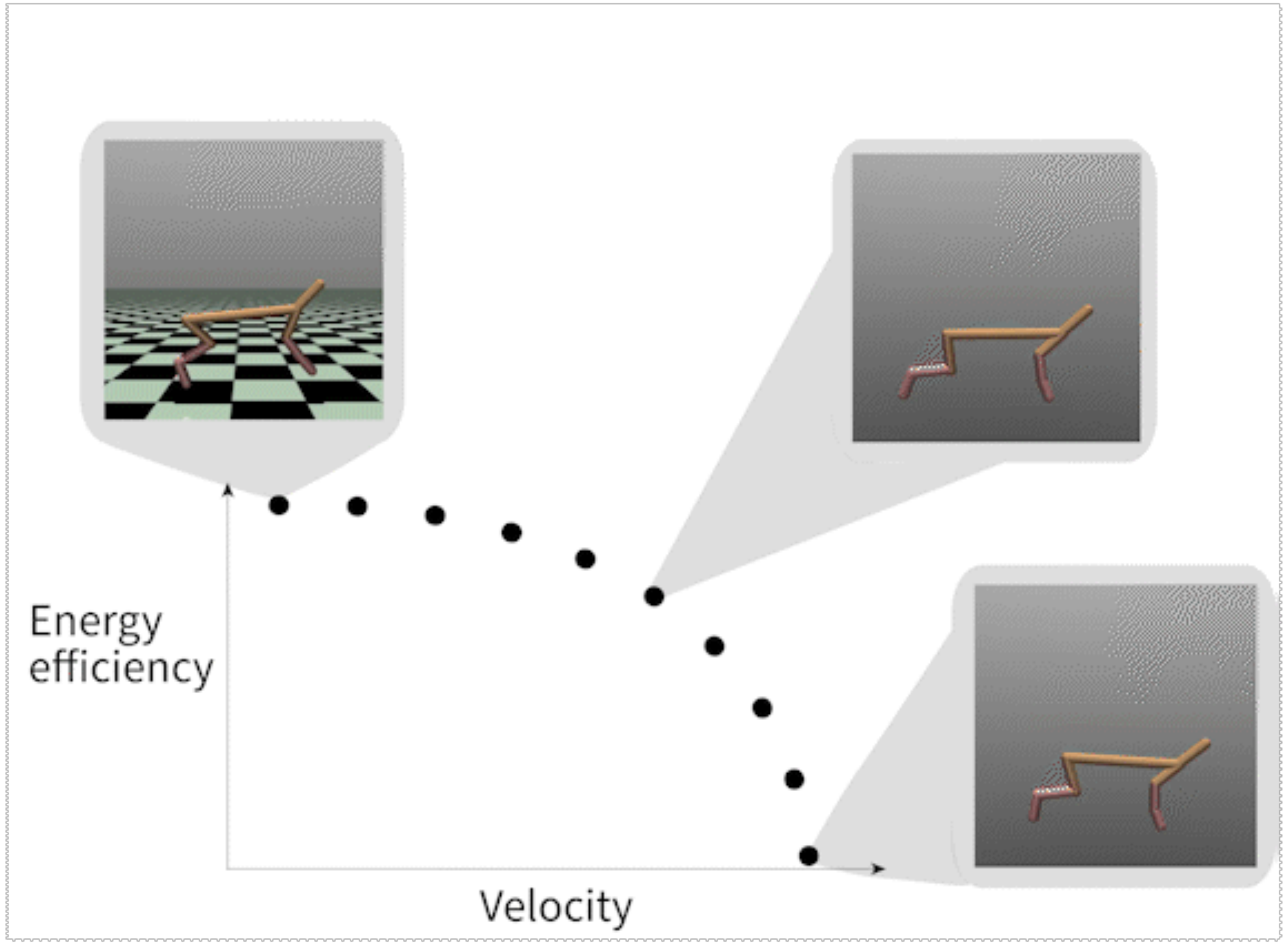
© Google maps

[1] Talbi, E.-G., "*Metaheuristics: From Design to Implementation.*" *Wiley Publishing, 2009.*

[2] *Zitzler, E., "Evolutionary algorithms for multiobjective optimization: methods and applications," in Ph.D. Dissertation. ETH Zurich, 1999.*

Multi-policy MORL

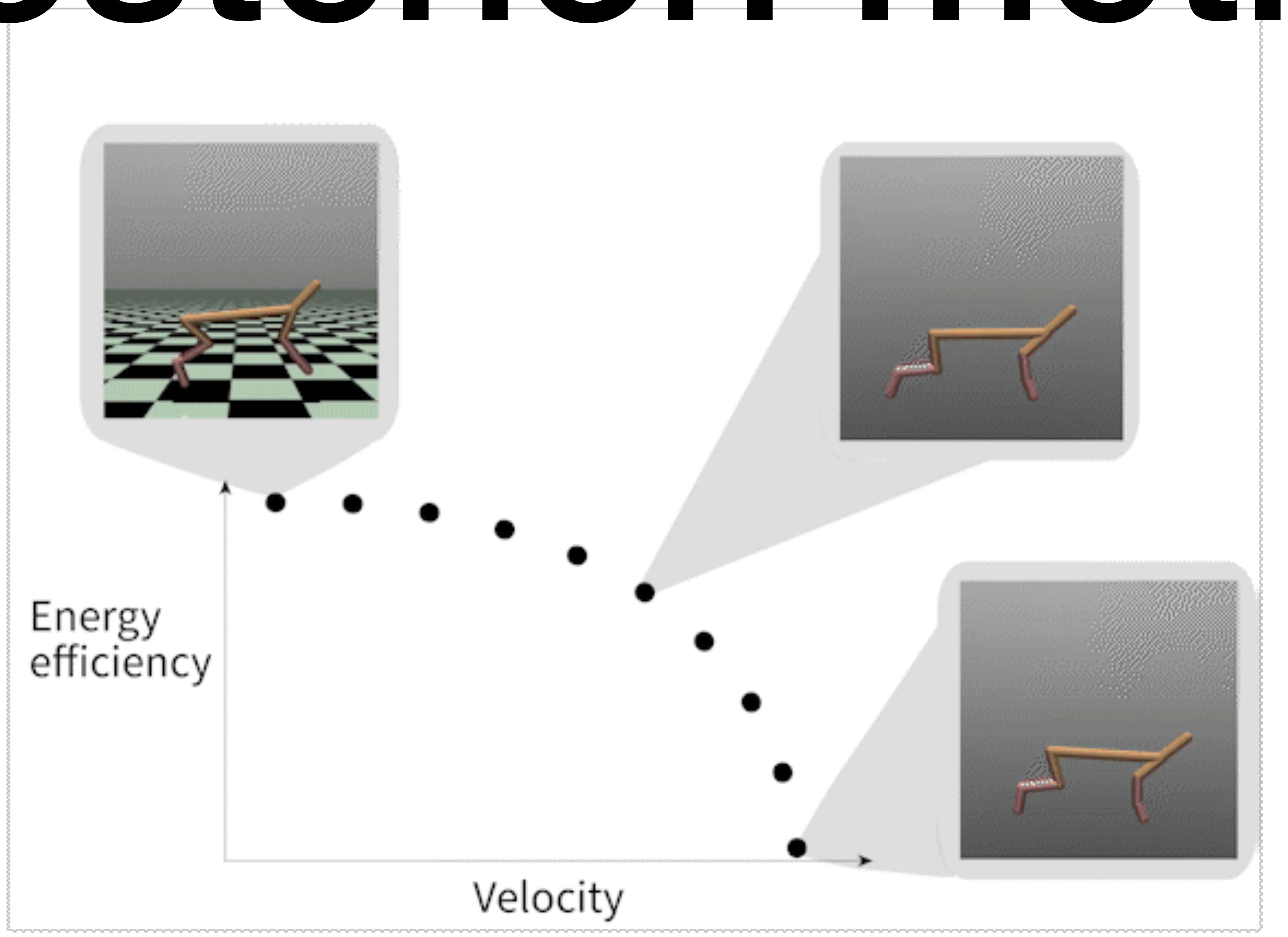
Learning behaviors associated with different compromises



[1] Roijers, D. et al., "A Survey of Multi-Objective Sequential Decision-Making," Journal of Artificial Intelligence, 2013.

[2] Hayes, C. et al., "A practical guide to multi-objective reinforcement learning and planning," Autonomous Agents and Multi-Agent Systems, 2022.

A posteriori methods



MOMDP → Algorithm → **Solution set** → Solution choice → Single solution

Training or learning phase

Selection phase

Execution or testing phase

[1] Roijers, D. et al., "A Survey of Multi-Objective Sequential Decision-Making," Journal of Artificial Intelligence, 2013.

[2] Hayes, C. et al., "A practical guide to multi-objective reinforcement learning and planning," Autonomous Agents and Multi-Agent Systems, 2022.

Families of multi-policy algorithms

Pareto-based

- Learn Pareto fronts for each state-action [1];
- Bootstraps on sets of vectors;
- ~ 5 existing works;
- Does not really scale to deep RL yet.

Decomposition-based

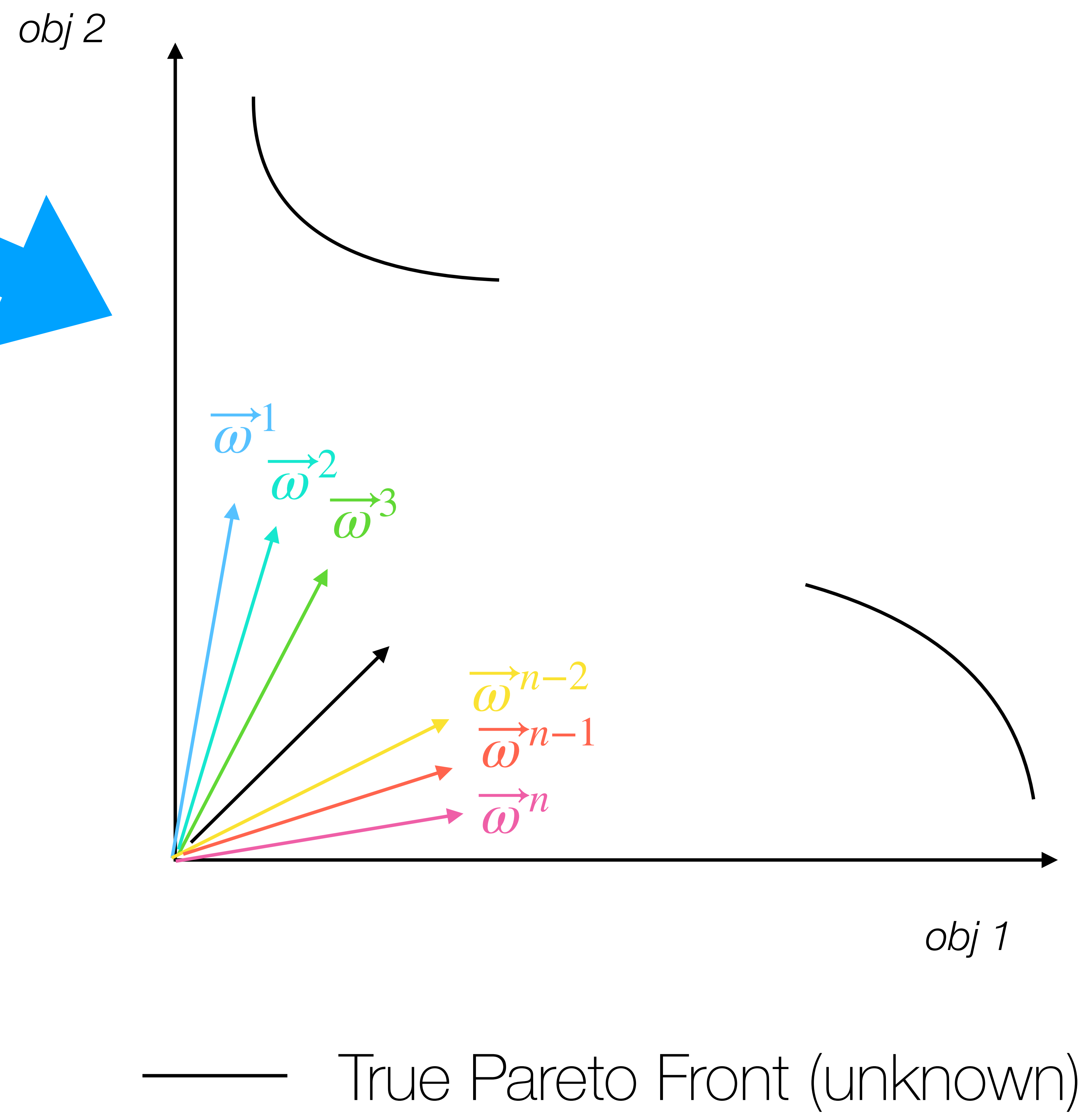
- Decompose the problem into several single-objective subproblems using a scalarization function [2];
- A large majority of existing works are decomposition-based;
- Trivial to scale to deep RL.

[1] K. Van Moffaert and A. Nowé, "Multi-objective reinforcement learning using sets of pareto dominating policies," The Journal of Machine Learning Research, 2014.

[2] F. Felten, E.-G. Talbi, and G. Danoy, "Multi-Objective Reinforcement Learning Based on Decomposition: A Taxonomy and Framework," Journal of Artificial Intelligence Research, 2024.

Naive MORL/D

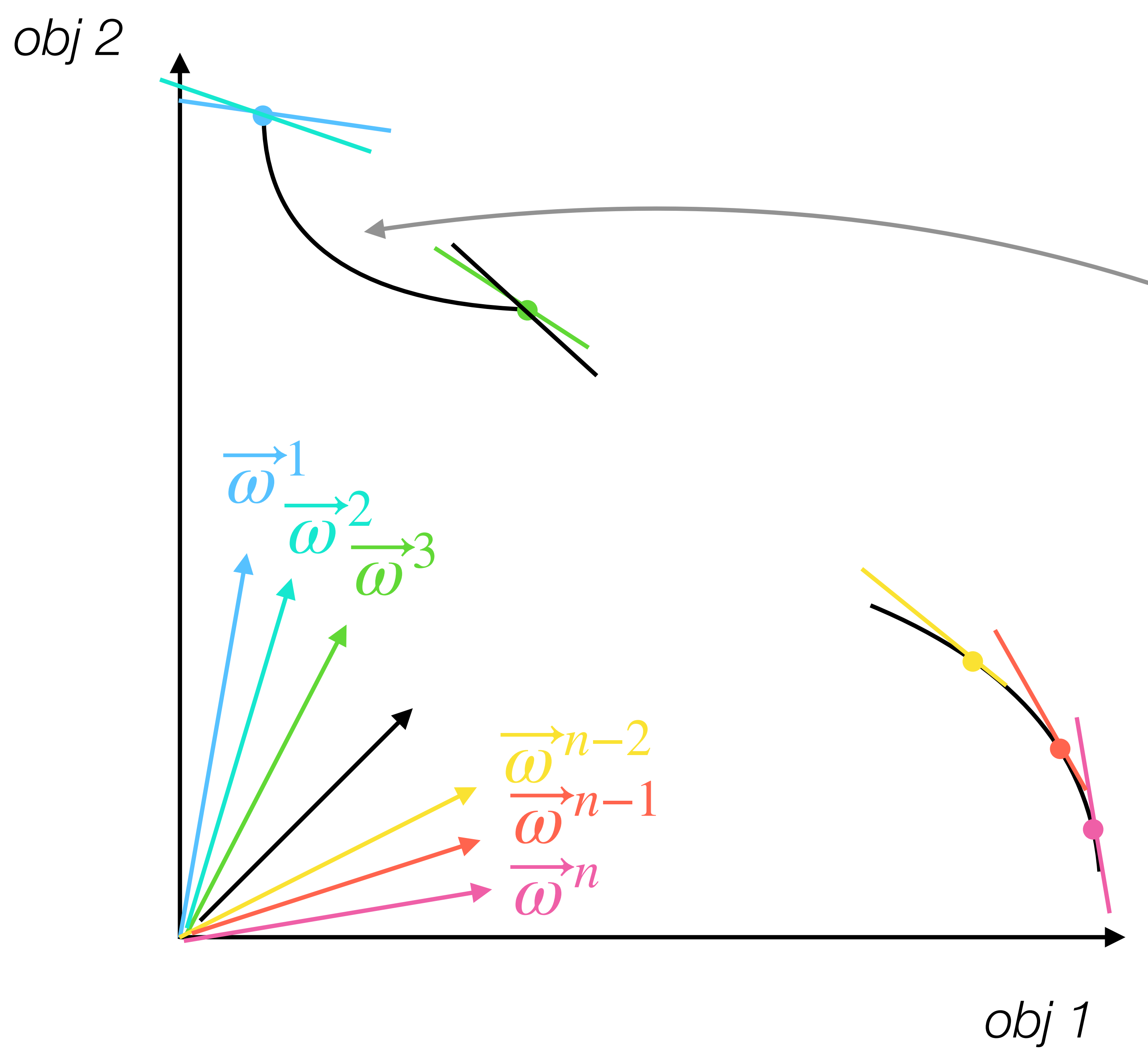
```
weights = generate_uniformly(n_objs)

policies = []

for w in weights:

    pi, v = train_rl(w, scalarization, env)

    policies.append((pi, v))

pareto_optimal = prune(policies)

return pareto_optimal
```



MORL research is about doing better than this…
And we often use existing methods from other fields such as MOO

Which scalarization function?

Linear? $$\sum_{i \in [1,2]} \omega_i \times obj_i$$



- Most common scalarization

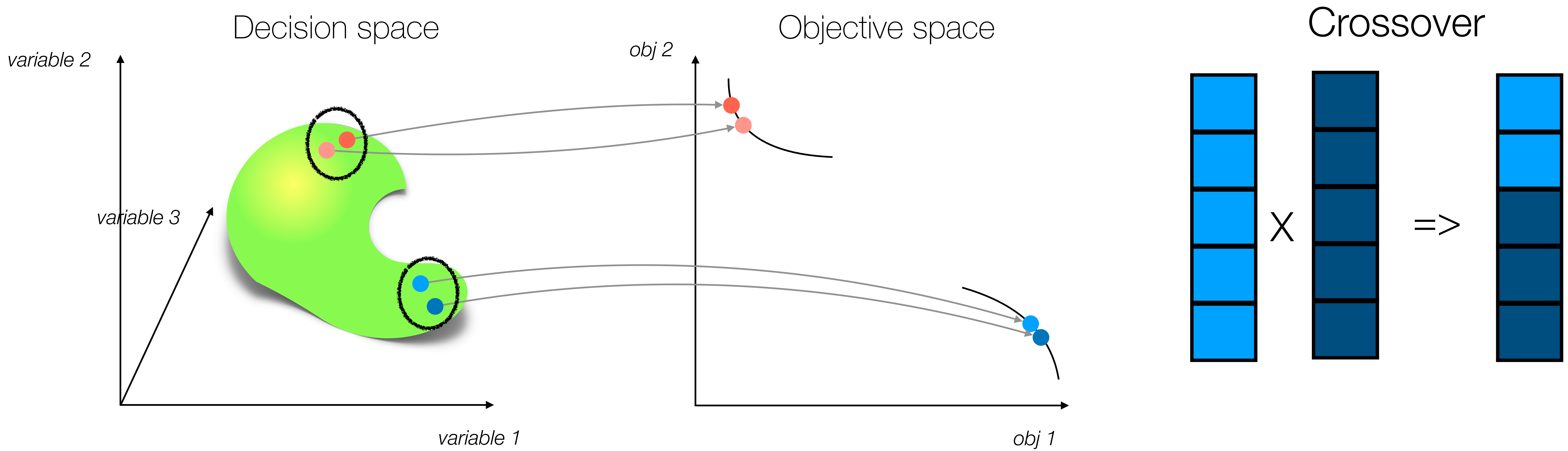
- But, cannot capture points in the concave parts of the PF;

  ➡️ Other non-linear functions exists, e.g. Chebyshev, PBI, etc. [1]

[1] Zhang, Q. and Li, H. "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," IEEE Transactions on Evolutionary Computation, 2007.

Can we use existing solutions to discover new ones?

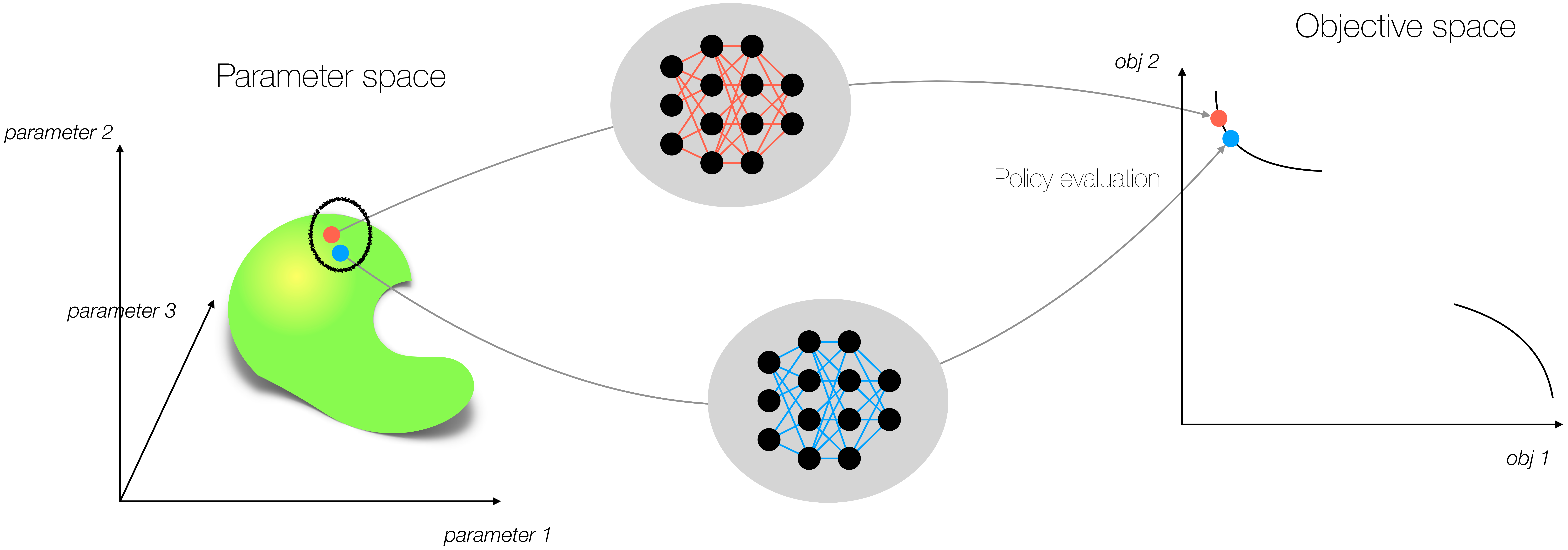
MOO: **Cooperation** techniques and similarity between neighbor solutions



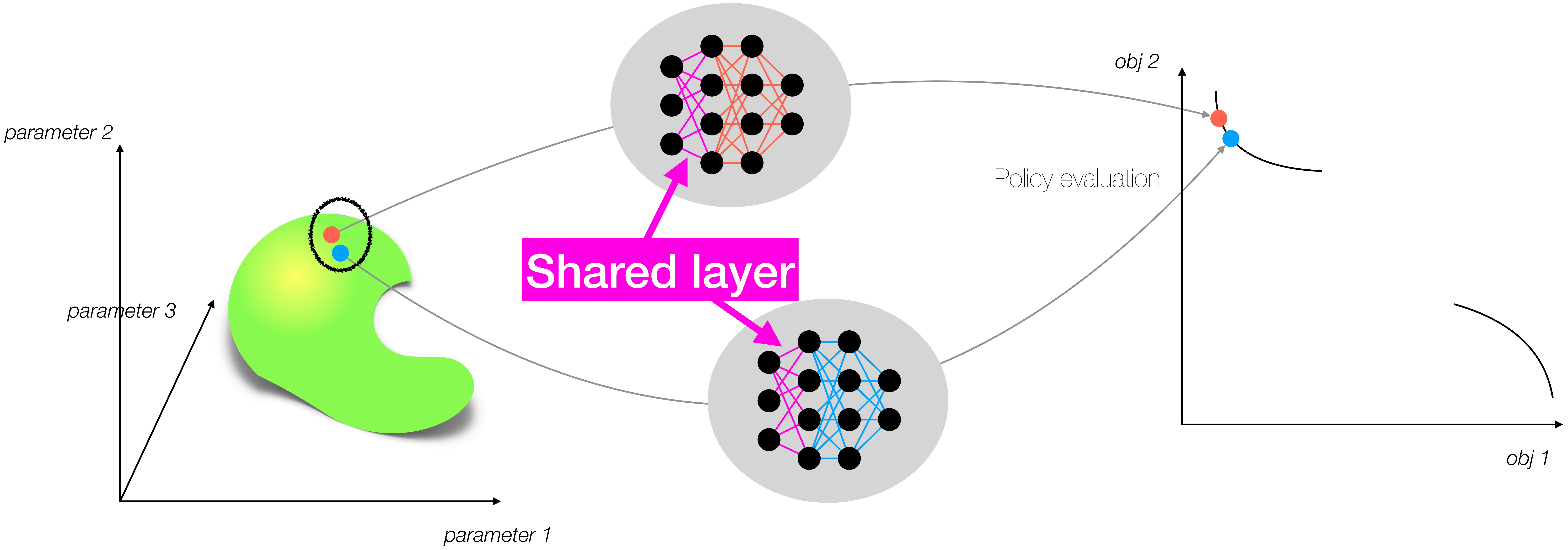
Multiple ways to “cooperate” exist: **crossover**, **shared search memory**, etc.

[1] Zhang, Q. and Li, H. “MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition,” IEEE Transactions on Evolutionary Computation, 2007.

Cooperation in MORL



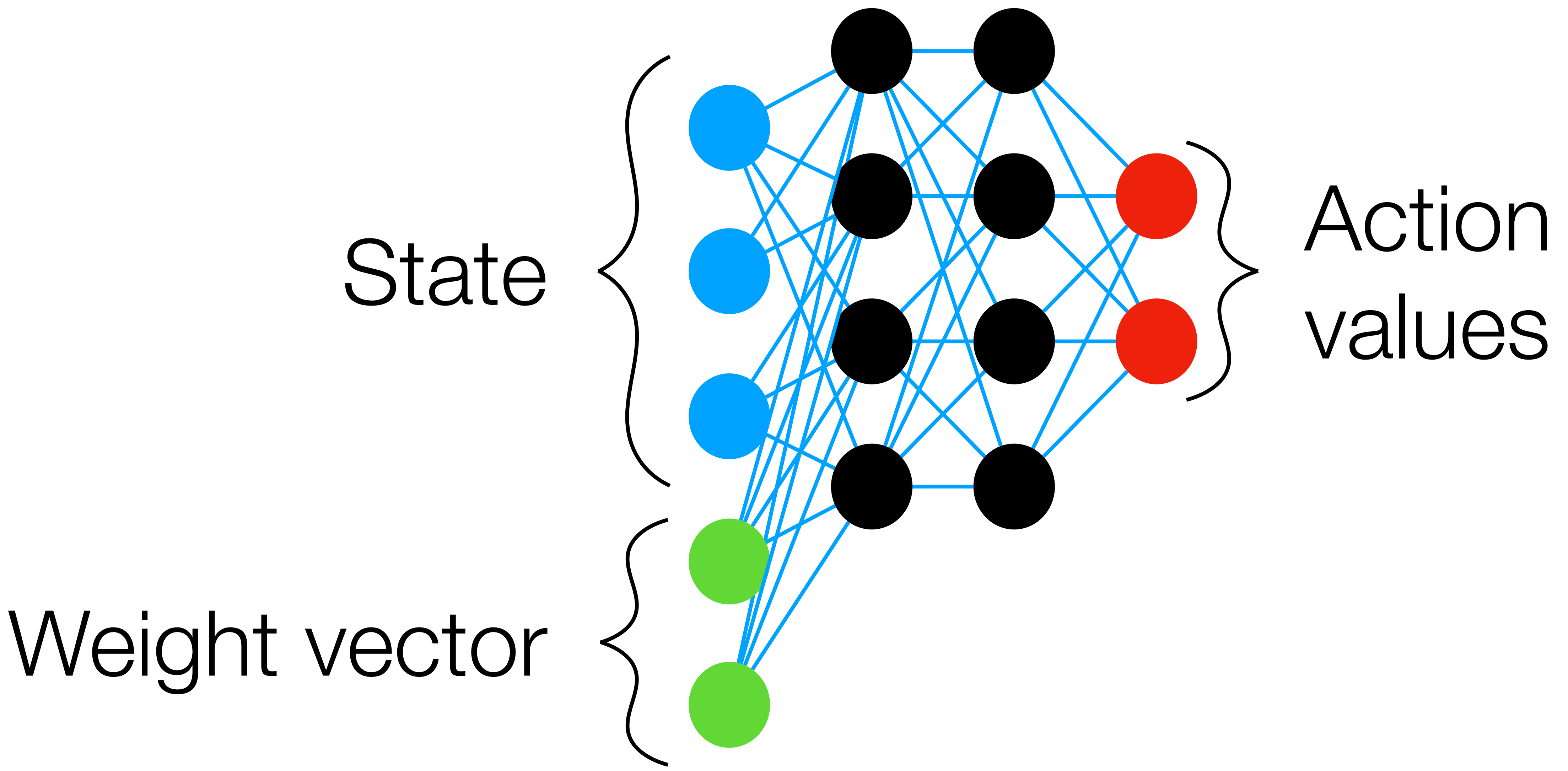
Cooperation in MORL



Chen, D., Wang, Y., and Gao, W., "Combining a gradient-based method and an evolution strategy for multi-objective reinforcement learning," Applied Intelligence, 2020.

Cooperation in MORL

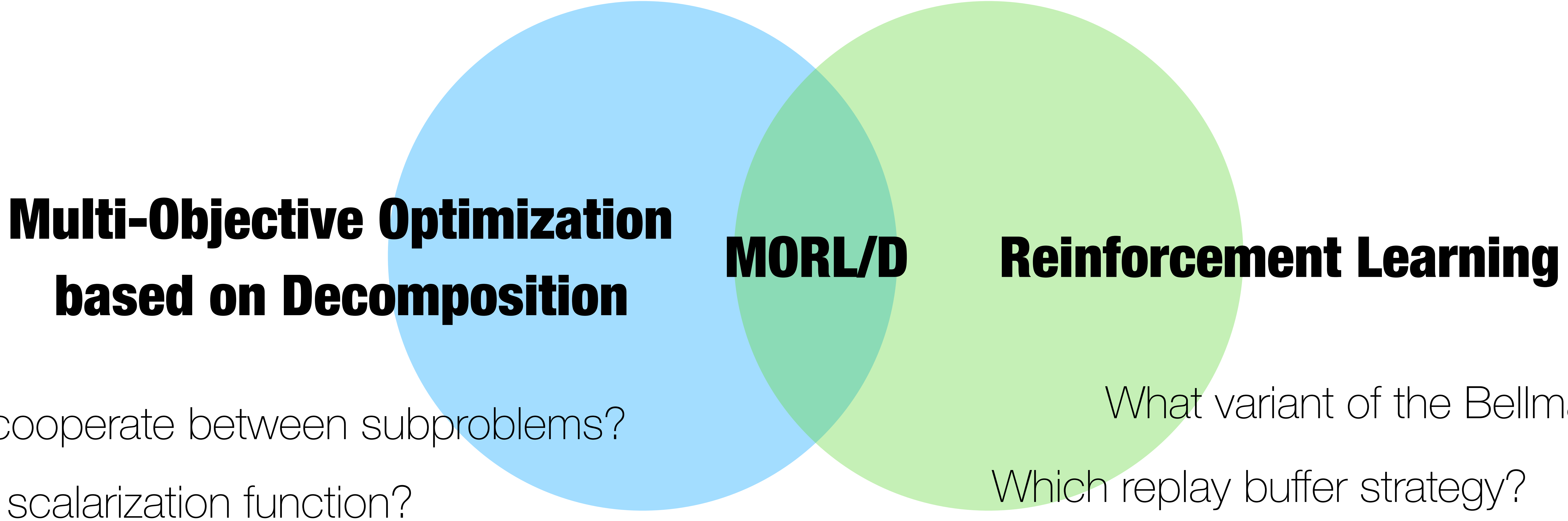
Conditioned network



1 network encodes multiple (all?) policies!

Abels, A., et al., "Dynamic Weights in Multi-Objective Deep Reinforcement Learning," in Proceedings of the 36th International Conference on Machine Learning (ICML), 2019.

Recurring topics



How to cooperate between subproblems?

Which scalarization function?

How to generate weight vectors?

What variant of the Bellman update?

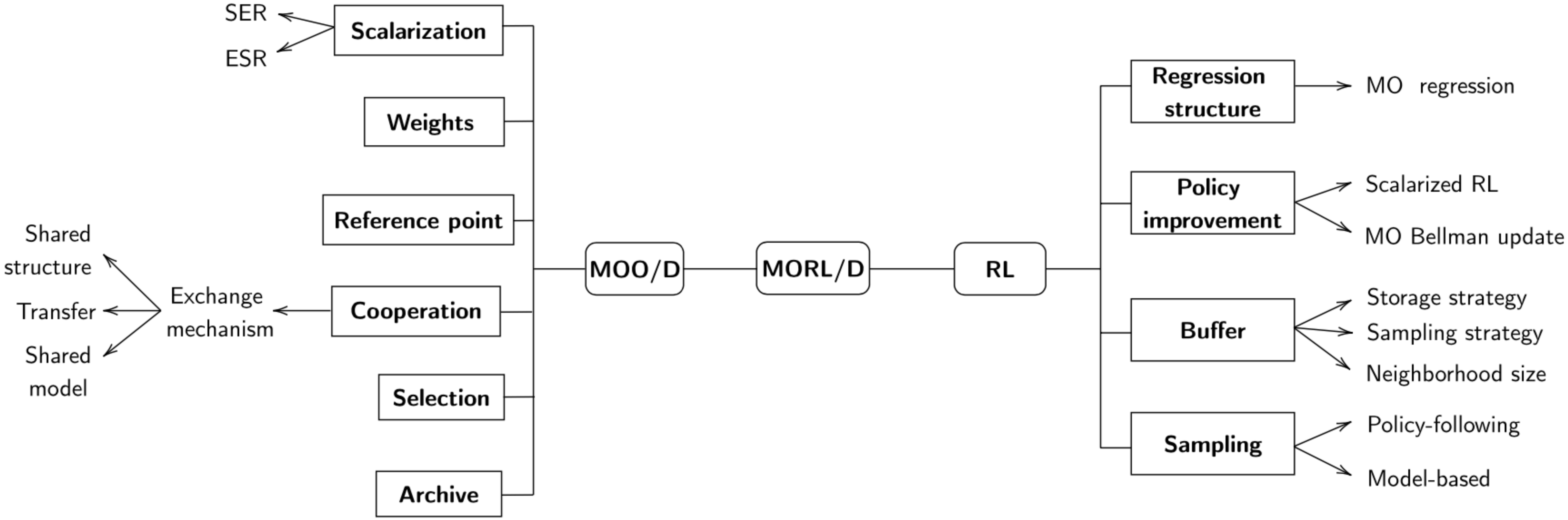
Which replay buffer strategy?

On or off-policy?

A lot of **existing techniques from MOO and RL can be applied to form new MORL/D methods**.

Actually, **various MORL contributions already use existing techniques**. But the **interactions between MOO/D, RL and MORL are not well identified**.

A taxonomy to classify existing and future works



We also propose a framework based on the taxonomy to construct adhoc algorithms

F. Felten, E.-G. Talbi, and G. Danoy, "Multi-Objective Reinforcement Learning Based on Decomposition: A Taxonomy and Framework," Journal of Artificial Intelligence Research, 2024.

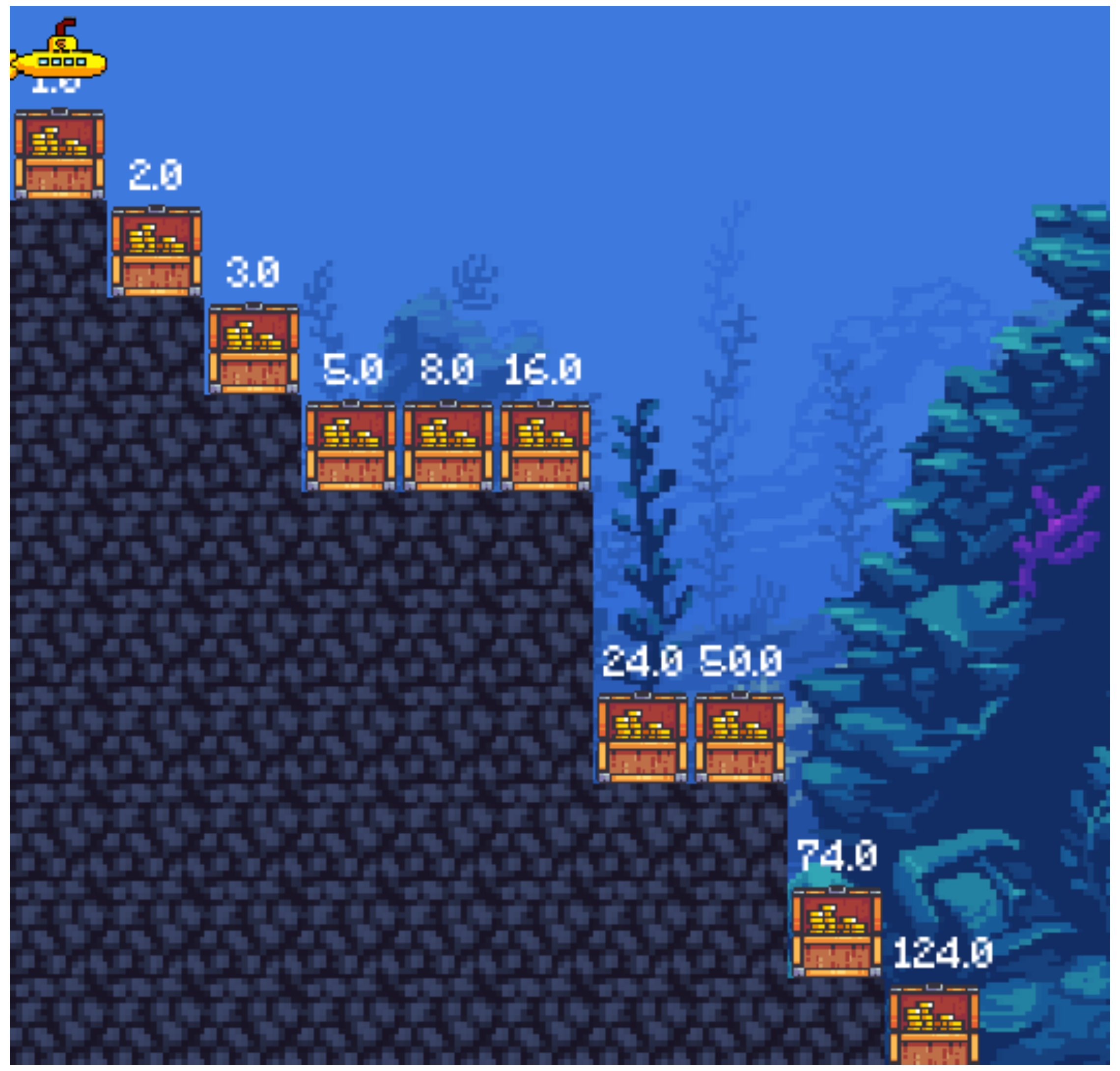
The MORL/D taxonomy

Bringing more clarity on ad-hoc contributions.

Reference	MOO					RL				
	Weight vectors		Cooperation			Regression structure	Policy improv.	Buffer		Sampling strategy
	When?	How?	Neighb.	Mechanism	Trigger			Neighb.	Storage & Sampling Strategy	
[Roijers et al., 2015b]	Dynamic	Adaptive - OLS	Single - Closest weight	Transfer	Periodic	$n\times$ Tabular	Scalarized POMDP solver	/	/	Policy following
[Mossalam et al., 2016]	Dynamic	Adaptive - OLS	Single - Closest weight	Transfer	Periodic	$n\times$ DNN + MO reg.	Scalarized DQN	Indep.	Recency + Uniform	Policy following
[Chen et al., 2020]	Static	Manual	All	Shared buffer Shared layers	Continuous	$n\times$ DNN	Scalarized SAC	All	Recency + Uniform	Parallel policy following
[Yang et al., 2019]	Dynamic	Random	All	CR	Continuous	$1\times$ DNN	Envelope DQN	All	HER + Recency + Uniform	Policy following
[Xu et al., 2020a]	Dynamic	Uniform	None	None	None	$n\times$ DNN + MO reg.	Scalarized PPO	Indep.	Recency + Uniform	Policy following
[Abels et al., 2019]	Dynamic	Random	All	CR	Continuous	$1\times$ DNN + MO reg.	Scalarized, Multi-weights DQN	All	HER + PER (Diversity)	Policy following
[Alegre et al., 2023]	Dynamic	Adaptive - GPI-LS	All	CR Shared model	Continuous	$1\times$ DNN + MO reg.	Scalarized, Multi-weights DQN or TD3	All	HER + PER (GPI)	Policy following
[Castelletti et al., 2013]	Dynamic	Random	All	CR	Continuous	$1\times$ Trees	Scalarized FQI	/	/	Historical dataset

F. Felten, E.-G. Talbi, and G. Danoy, "Multi-Objective Reinforcement Learning Based on Decomposition: A Taxonomy and Framework," Journal of Artificial Intelligence Research, 2024.

Framework instantiation



Discrete state/action spaces

Concave Pareto frontier

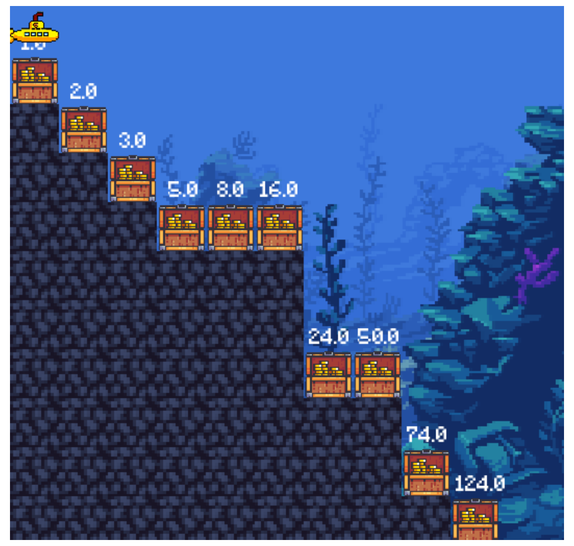
Looking for a deterministic policy

	Scalarization	Weight vector	Policy improvement
MORL/D	Chebyshev	Uniform, then adaptive technique from MOO [1]	Expected Utility Policy Gradient [2]

[1] Czyżżak, P. and Jaszkiewicz, A., “Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization,” Journal of Multi-Criteria Decision Analysis,1998.

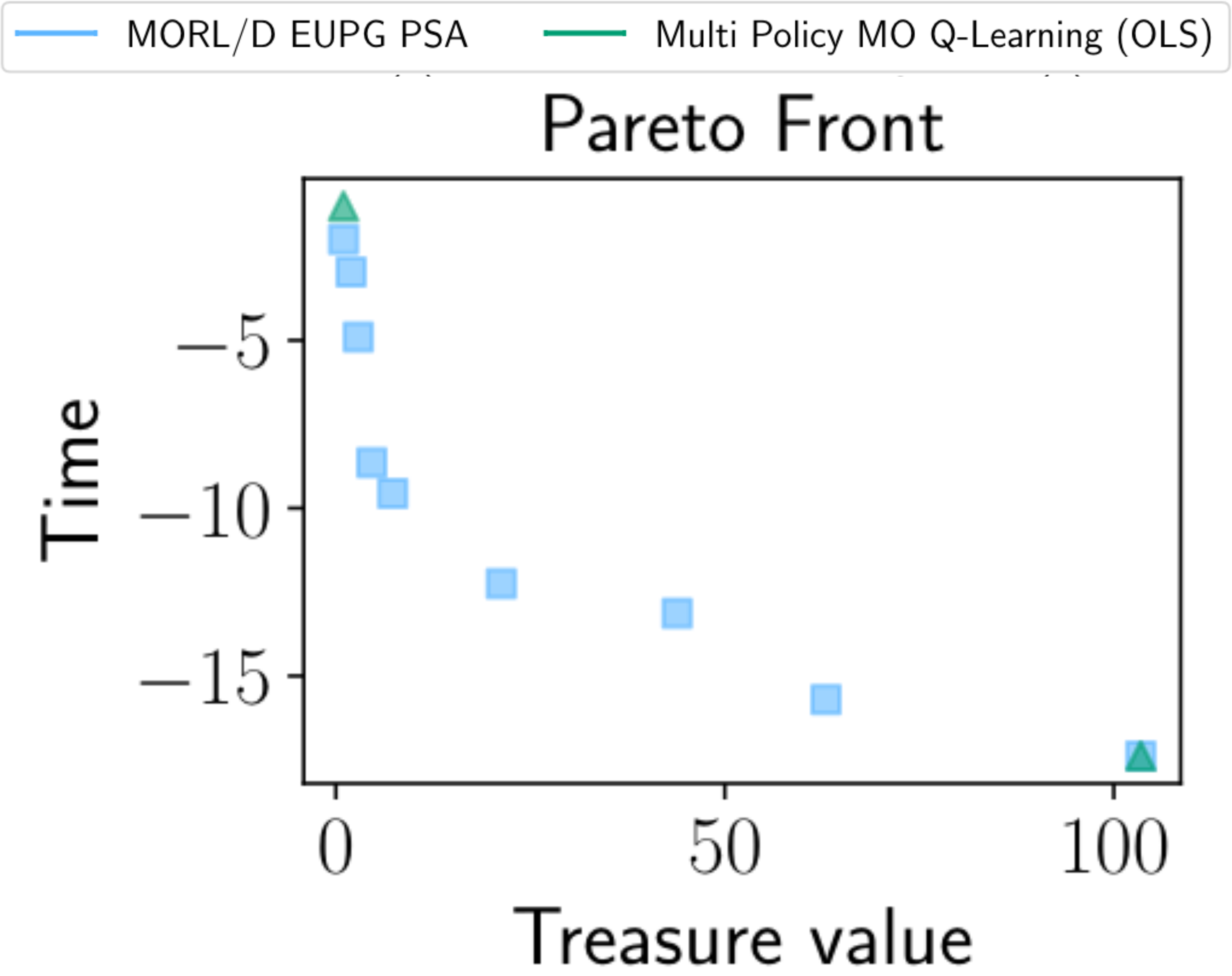
[2] Roijers, D., Steckelmacher, D., and Nowe, A., “Multi-objective Reinforcement Learning for the Expected Utility of the Return,” in Proceedings of the ALA workshop at ICML/AAMAS/IJCAI, 2018.

Framework instantiation



MORL/D can learn points in the concave part of the PF.

Finds different points thanks to the weight adaptation techniques from MOO literature.

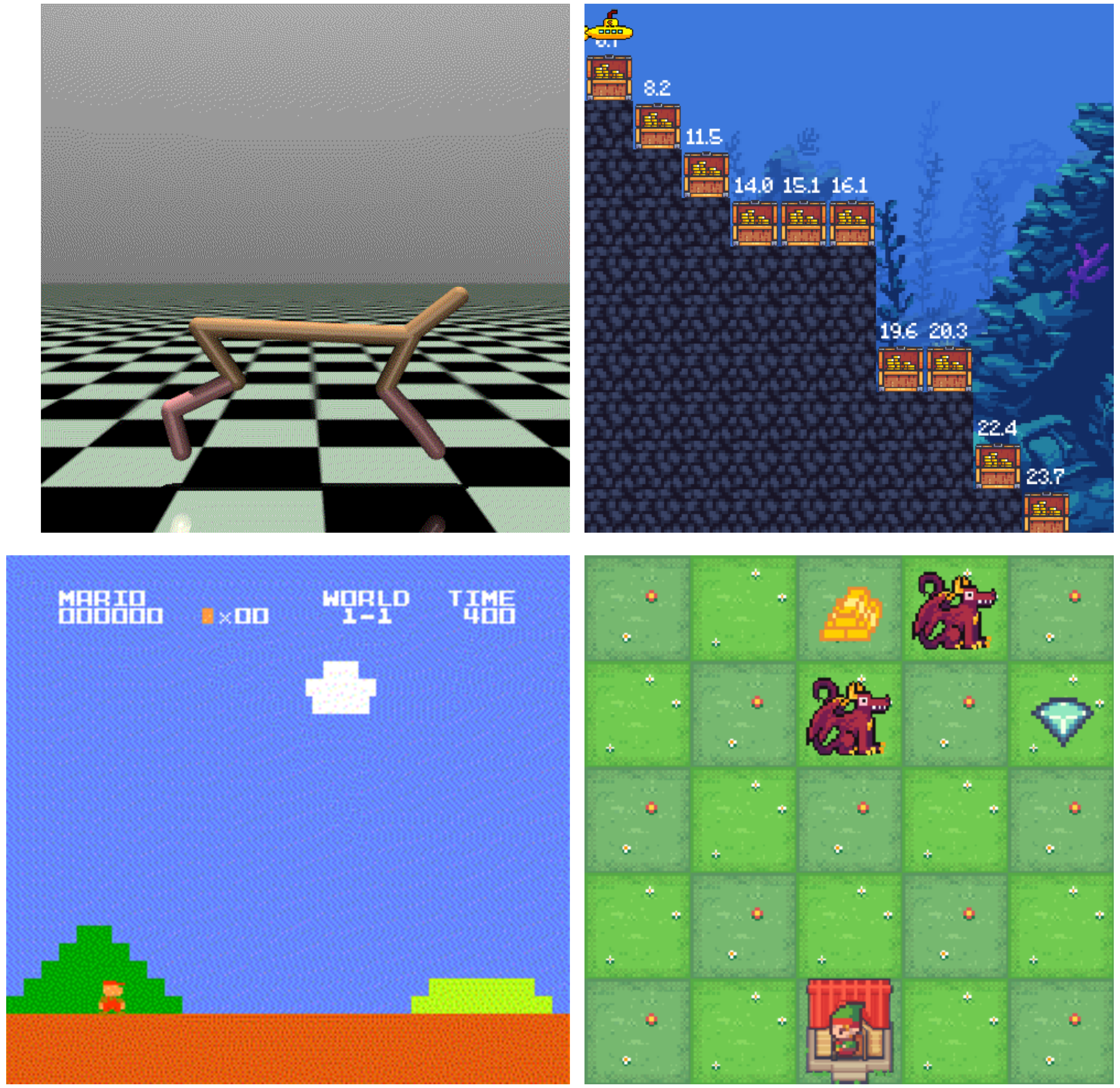


F. Felten, E.-G. Talbi, and G. Danoy, "Multi-Objective Reinforcement Learning Based on Decomposition: A Taxonomy and Framework," Journal of Artificial Intelligence Research, 2024.

Tooling

Standard environments

MO-Gymnasium



➡️ >25 MORL environments under a unified API

➡️ Open-source, part of the Farama Foundation since 2023

➡️ Useful and used

> 100k downloads in ~1.5 years

F. Felten, L. Alegre, et al., "A Toolkit for Reliable Benchmarking and Research in Multi-Objective Reinforcement Learning," NeurIPS, 2023.

Reliable implementations of algorithms

Algorithm	Single or multi-policy	Utitlity function	Observation space	Action space
MOQL [Van Moffaert et al., 2013]	Single	Linear	Disc.	Disc.
EUPG [Roijers et al., 2018]	Single	Non-linear, ESR	Disc.	Disc.
MPMOQL [Van Moffaert et al., 2013]	Multi	Linear	Disc.	Disc.
PQL [Van Moffaert and Nowé, 2014]	Multi	Non-linear, SER (*)	Disc.	Disc.
OLS [Roijers and Whiteson, 2017]	Multi	Linear	/ (**)	/ (**)
Envelope [Yang et al., 2019]	Multi	Linear	Cont.	Disc.
PGMORL [Xu et al., 2020a]	Multi	Linear	Cont.	Cont.
PCN [Reymond et al., 2022]	Multi	Non-linear, ESR/SER (*)	Cont.	Disc.
GPI-LS & GPI-PD [Alegre et al., 2023]	Multi	Linear	Cont.	Any
CAPQL [Lu et al., 2023]	Multi	Linear	Cont.	Cont.
MORL/D [Felten et al., 2024] (Section 2.2)	Multi	Any	Any	Any

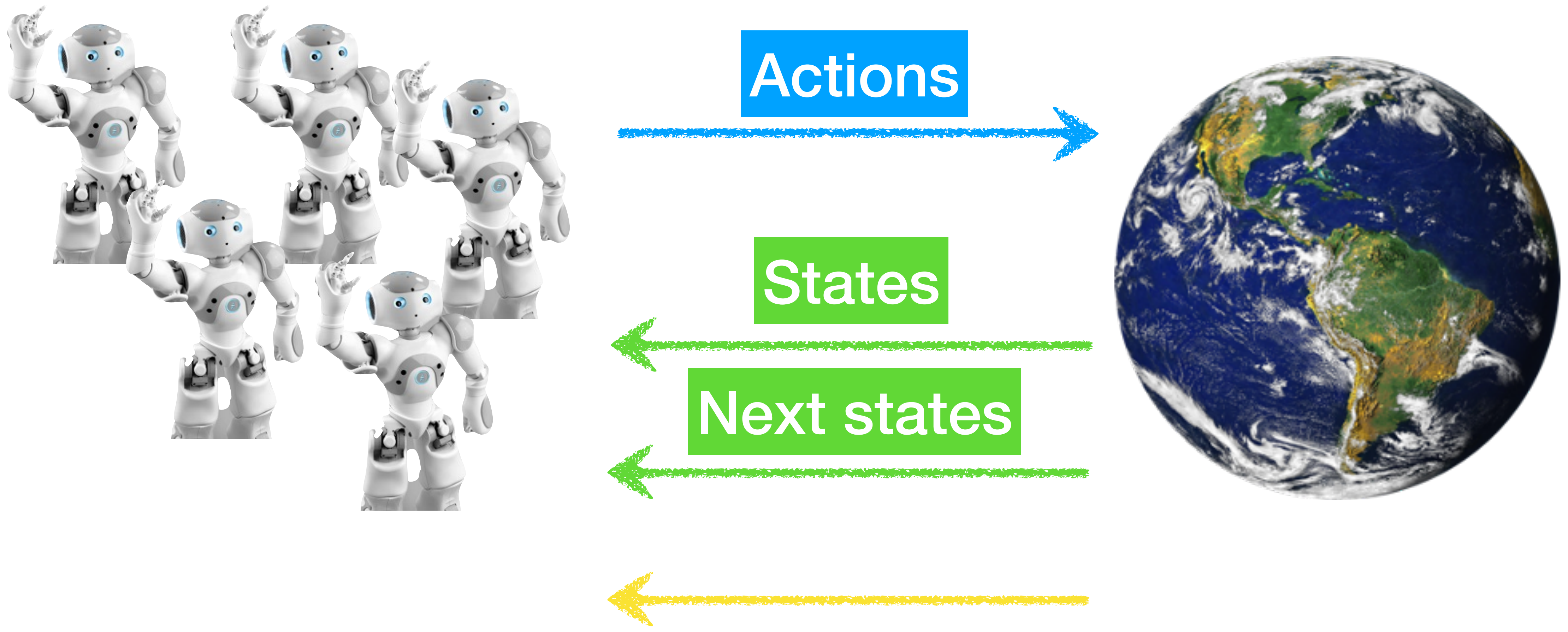
MORL-Baselines

- ➡️ > 10 MORL algorithms
- ➡️ Compatible with MO-Gymnasium
- ➡️ Clean, tested and documented code
- ➡️ Lots of utilities for MORL researchers

F. Felten, L. Alegre, et al., "A Toolkit for Reliable Benchmarking and Research in Multi-Objective Reinforcement Learning," NeurIPS, 2023.

2. Multi-Objective Multi-Agent RL (MOMARL)

Setup



Each agent receives a vectorial reward signal

Rădulescu, R. et al., "Multi-Objective Multi-Agent Decision Making: A Utility-based Analysis and Survey," Autonomous Agents and Multi-Agent Systems, 2020.

Solution concepts

MOMARL

Known utility

~MARL policy with
ESR vs. SER

Unknown utility

"Multi-compromise" MARL

Team reward

$$\mathbf{v}^{\pi}_{team} = \mathbf{v}^{\pi}_{1} = \ldots = \mathbf{v}^{\pi}_{n}$$

Pareto set of MA policies

Individual reward

Pareto-Nash sets
(no known algorithm)

In this setting, the value function is a matrix of size objs x agents

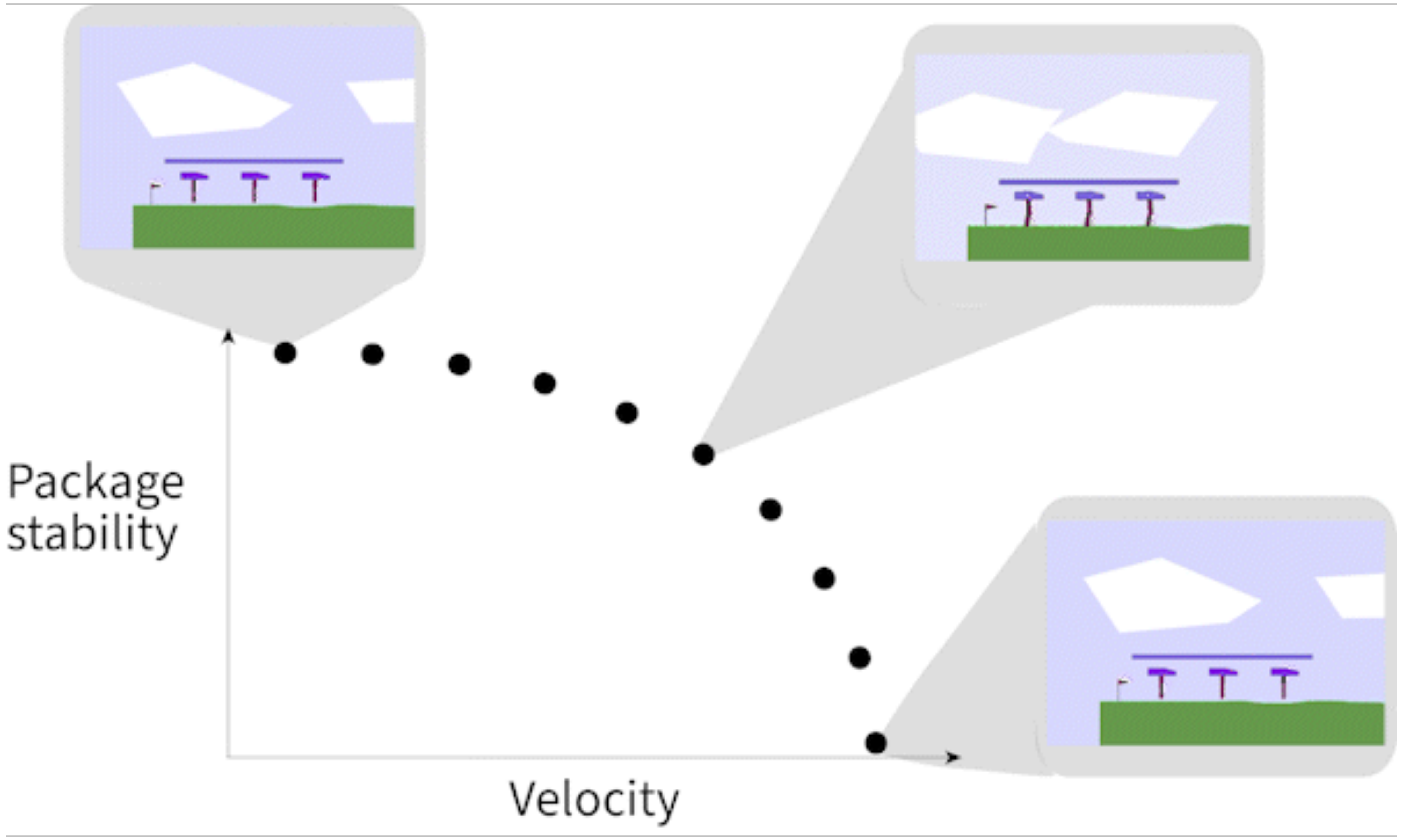
$$\mathbf{V}^{\pi} = [\mathbf{v}^{\pi}_{1} \ldots \mathbf{v}^{\pi}_{n}]^{T}$$

There are still relatively unexplored areas, e.g., heterogeneous utilities

[1] Rădulescu, R. et al., "Multi-Objective Multi-Agent Decision Making: A Utility-based Analysis and Survey," Autonomous Agents and Multi-Agent Systems, 2020.

[2] F. Felten et al., "MOMAland: A Set of Benchmarks for Multi-Objective Multi-Agent Reinforcement Learning," ArXiv, 2024.

Pareto set of MA policies



F. Felten et al., *"MOMAland: A Set of Benchmarks for Multi-Objective Multi-Agent Reinforcement Learning," ArXiv, 2024.*

Learning Pareto sets of MA policies

Option 1: Centralisation + MORL

```
MOMA_env = …

MO_env = CentraliseAgent(MOMA_env)

Pareto_policies = MORL(MO_env)
```

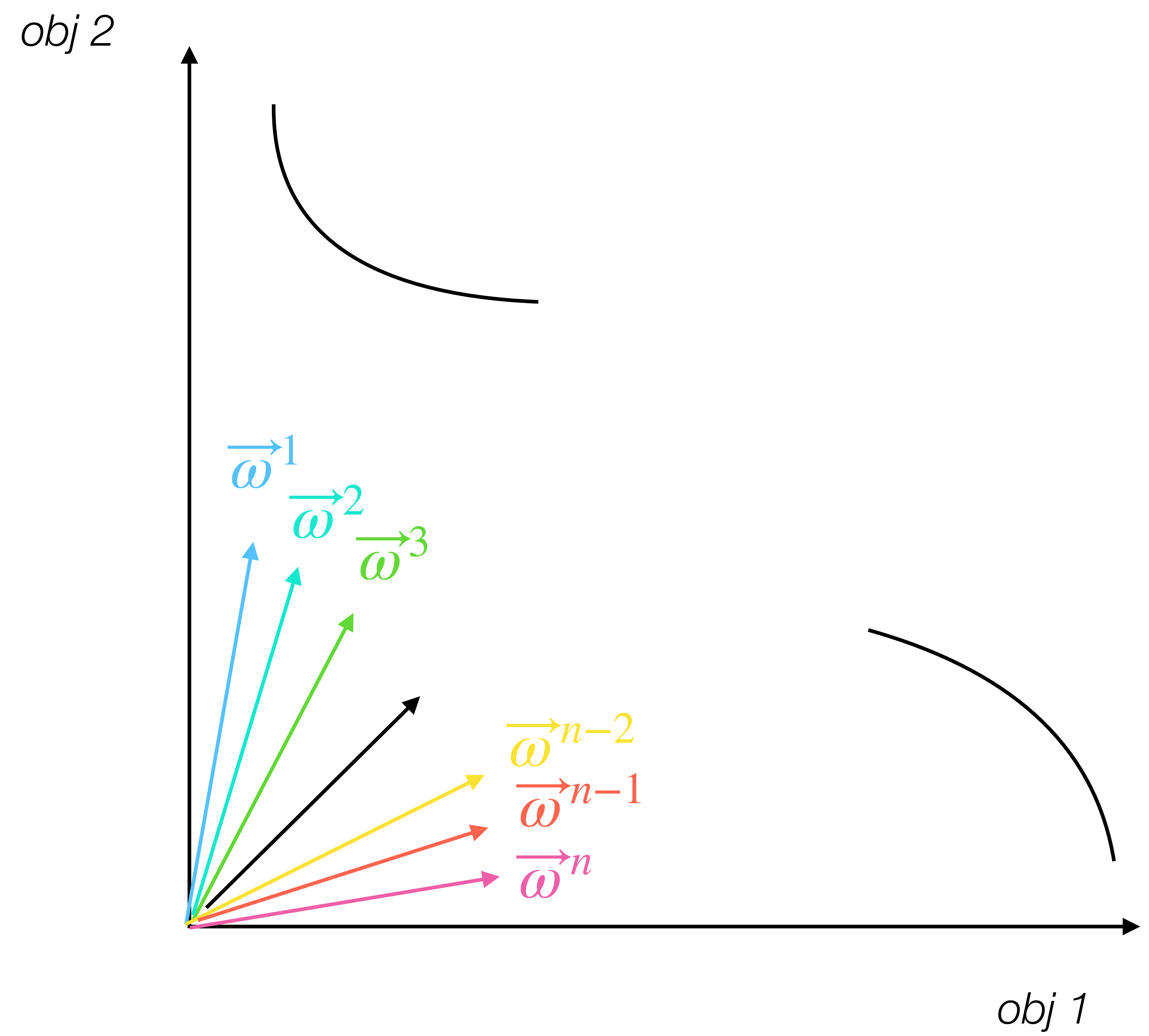
There are obvious problems with this approach, e.g., explosion of the action space
But it still gives a good baseline for future research

F. Felten et al., "MOMAland: A Set of Benchmarks for Multi-Objective Multi-Agent Reinforcement Learning," ArXiv, 2024.

Learning Pareto sets of MA policies

Option 2: Decomposition + MARL

```
MOMA_env = …

weights = generate_weights(n_objs)

for w in weights:

  MA_env = LinearizeRewards(MOMA_env, w)

  MA_policies.append(MARL(MA_env))

Pareto_policies = prune(MA_policies)
```



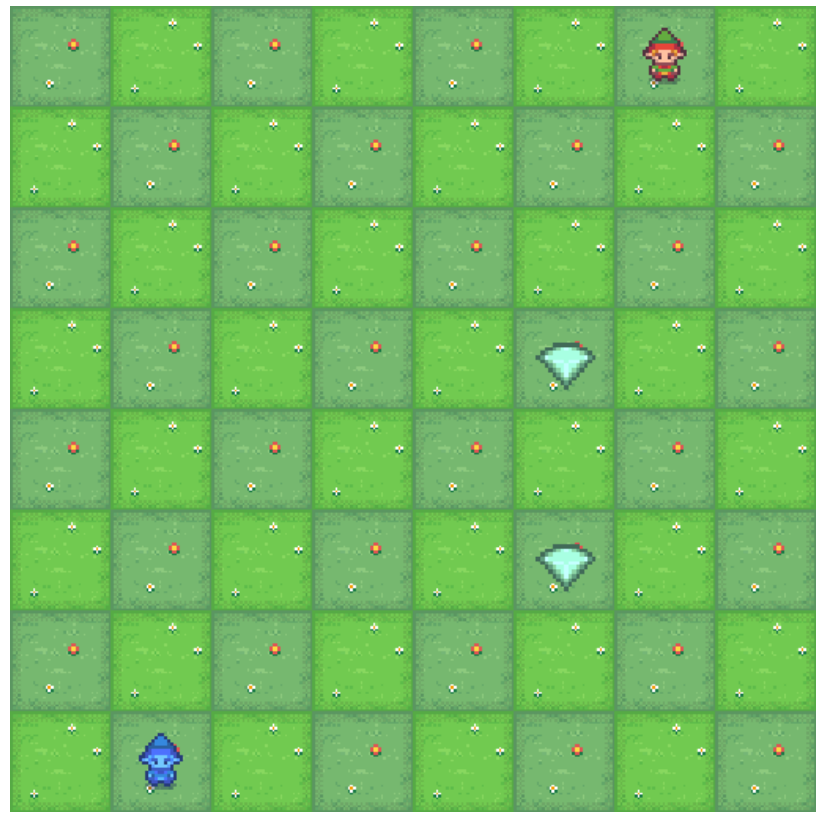
Naive baseline but we can transfer a lot of knowledge from MORL/D

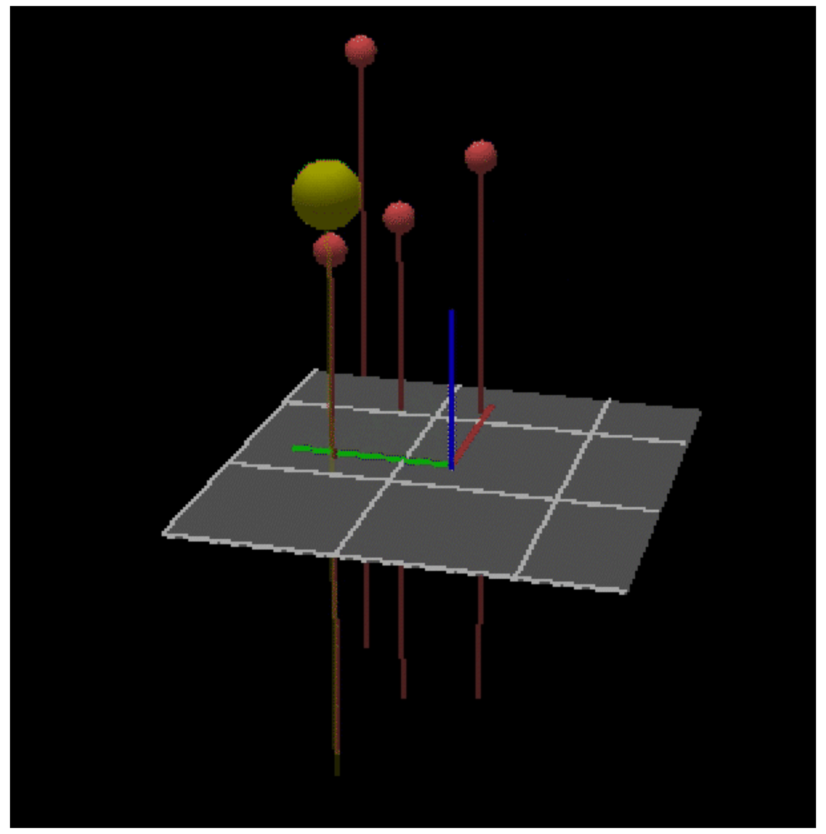
F. Felten et al., "MOMAland: A Set of Benchmarks for Multi-Objective Multi-Agent Reinforcement Learning," ArXiv, 2024.

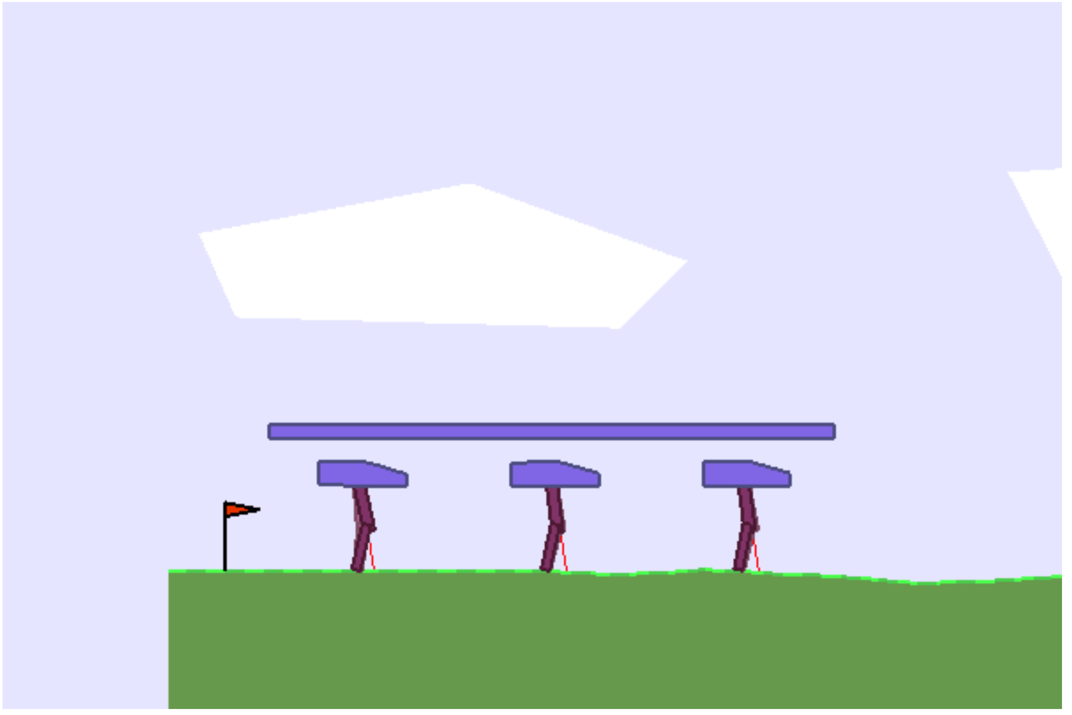
Tooling

Envs and baselines

MOMAland







➡️ ~10 MOMARL **environments** under a unified API

➡️ Open-source, part of the Farama Foundation

➡️ Also brings **utilities** and **learning algorithms**, e.g., MOMAPPO

F. Felten et al., “MOMAland: A Set of Benchmarks for Multi-Objective Multi-Agent Reinforcement Learning,” , arXiv, 2024.

Overview of the Farama ecosystem

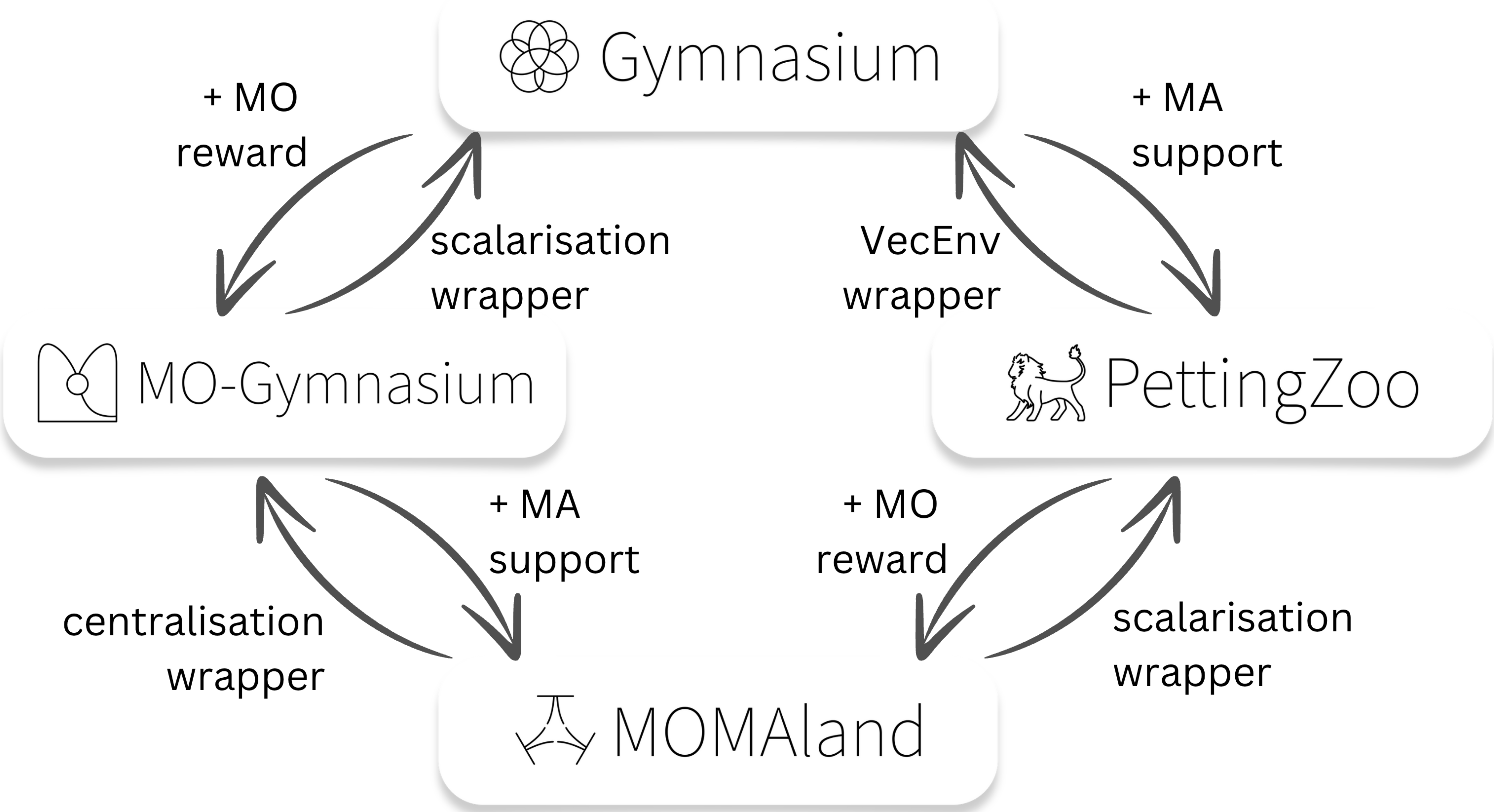


Image by Roxana Rădulescu

3. Application

CrazyRL

States:

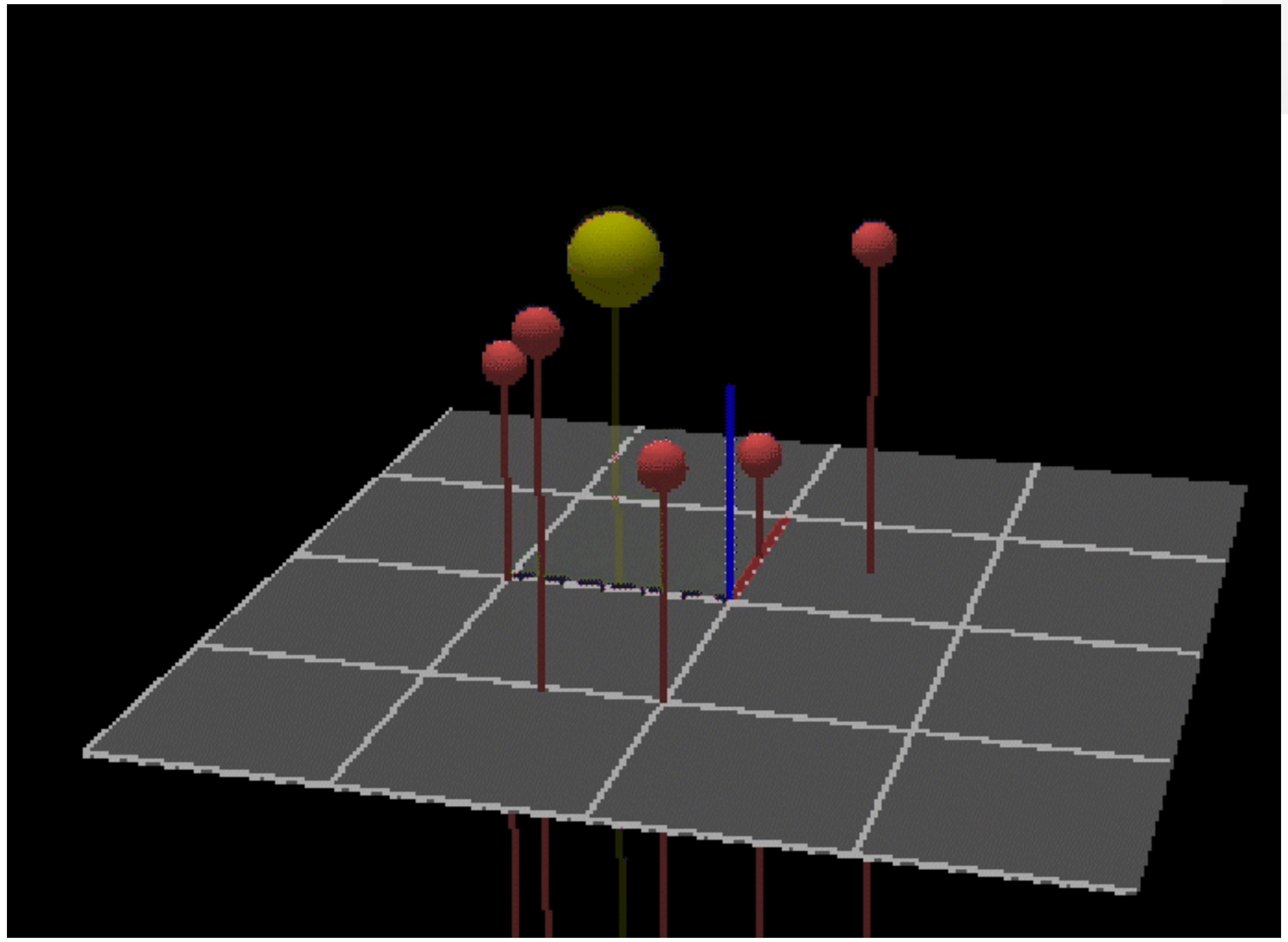
Each drone perceives x, y, z coordinates of everyone

Actions:

3D speed vector

Objectives:

- Close to target
- Far from other agents (avoid collisions & spread)



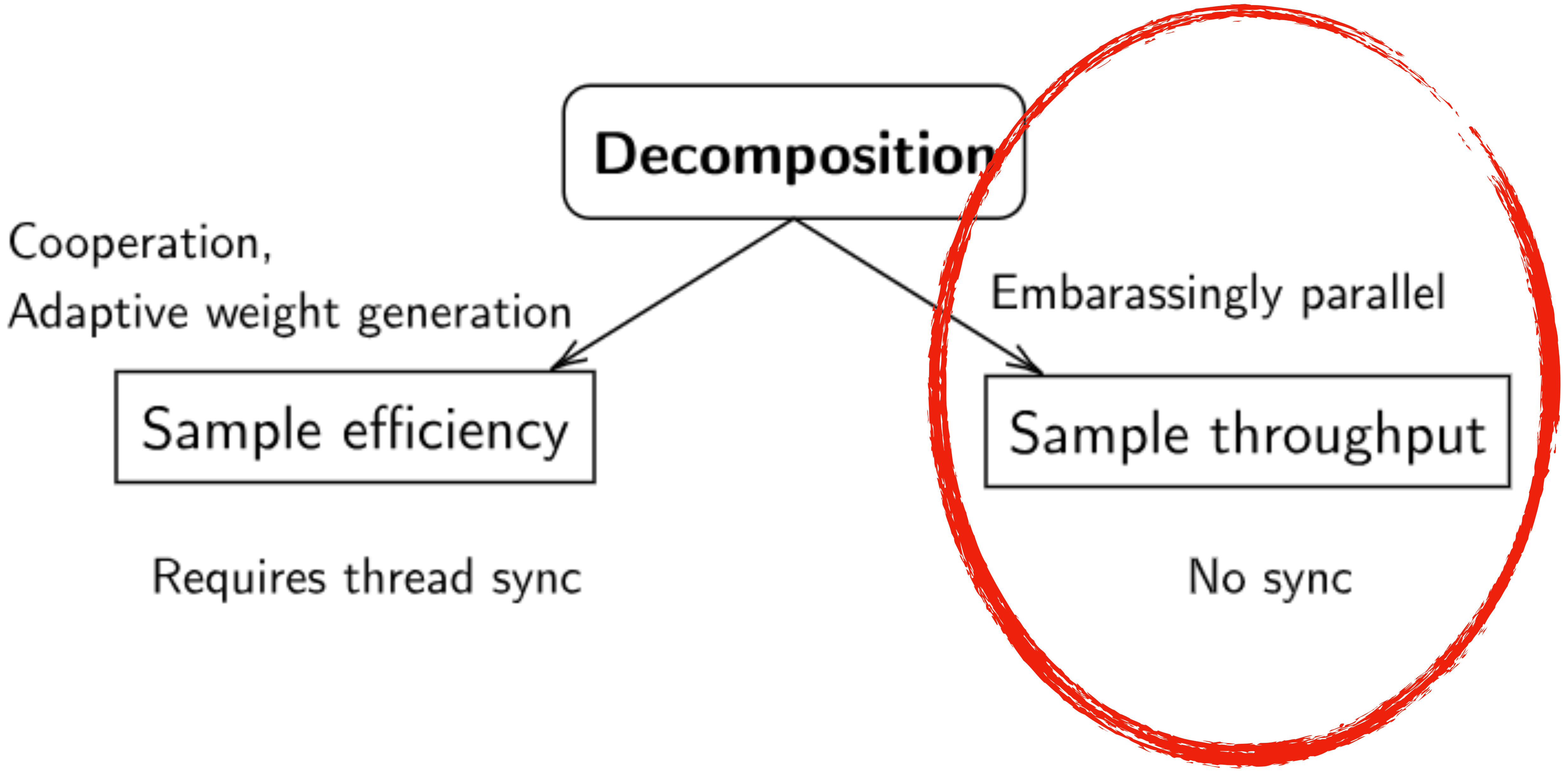
CrazyFlie [1]

[1] Giernacki, W., et al., "Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering," in 22nd International Conference on Methods and Models in Automation and Robotics (MMAR), 2017.

[2] F. Felten, "Multi-Objective Reinforcement Learning," PhD Thesis, Université du Luxembourg, 2024.

Accelerated decomposition

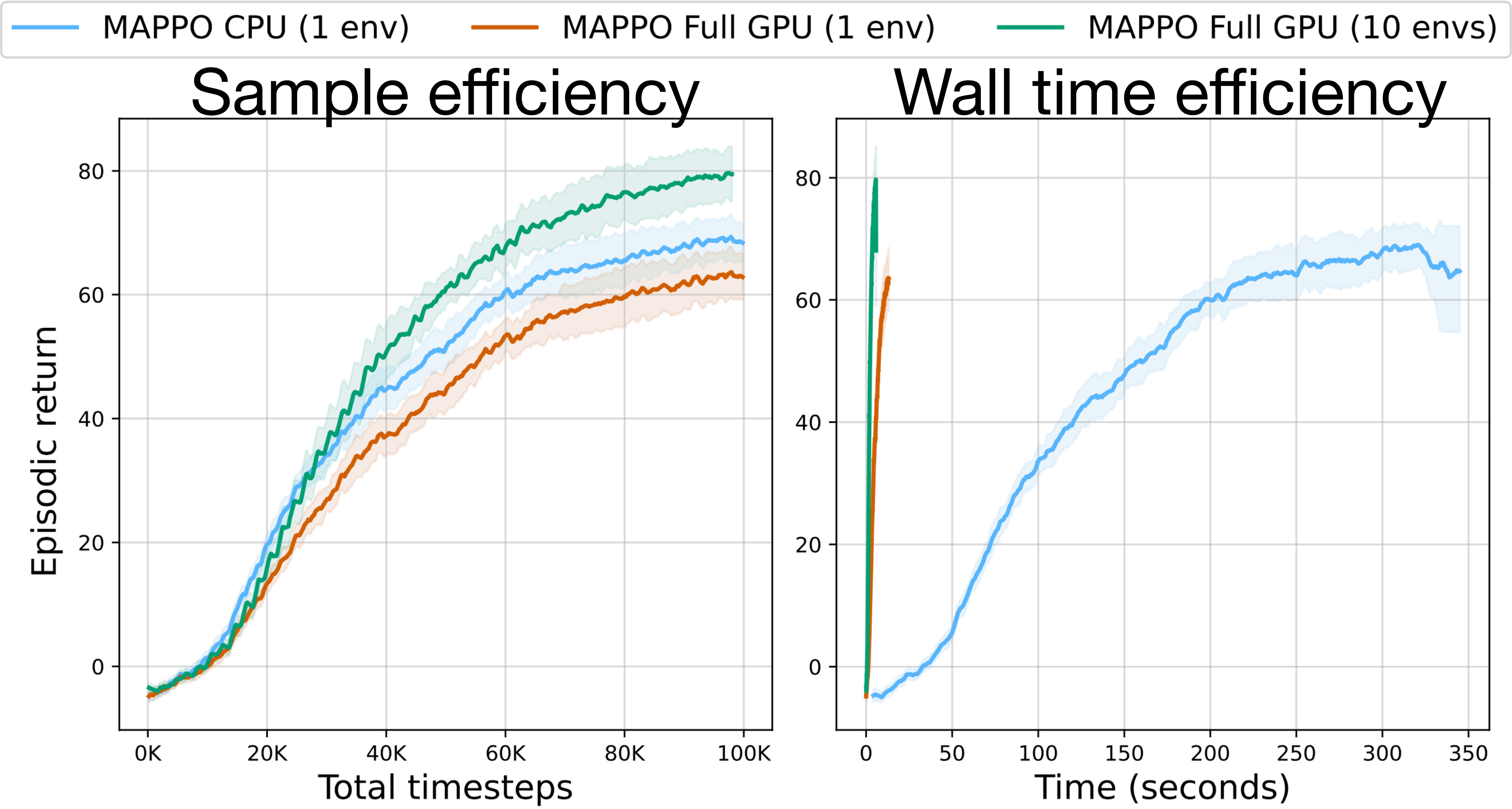
1. The CrazyRL environments can be run on a GPU (JAX-based implementation);
2. Learning and simulations co-located on the same accelerated hardware;
3. We can benefit from running the training of multiple trade-offs in parallel on the GPU.



F. Felten, *"Multi-Objective Reinforcement Learning," PhD Thesis, Université du Luxembourg, 2024.*

Learning + simulation on GPU

For 1 trade-off: MAPPO [1]



[1] C. Yu *et al.*, "The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games," in NeurIPS, 2022.

[2] F. Felten, "Multi-Objective Reinforcement Learning," PhD Thesis, Université du Luxembourg, 2024.

Accelerated decomposition

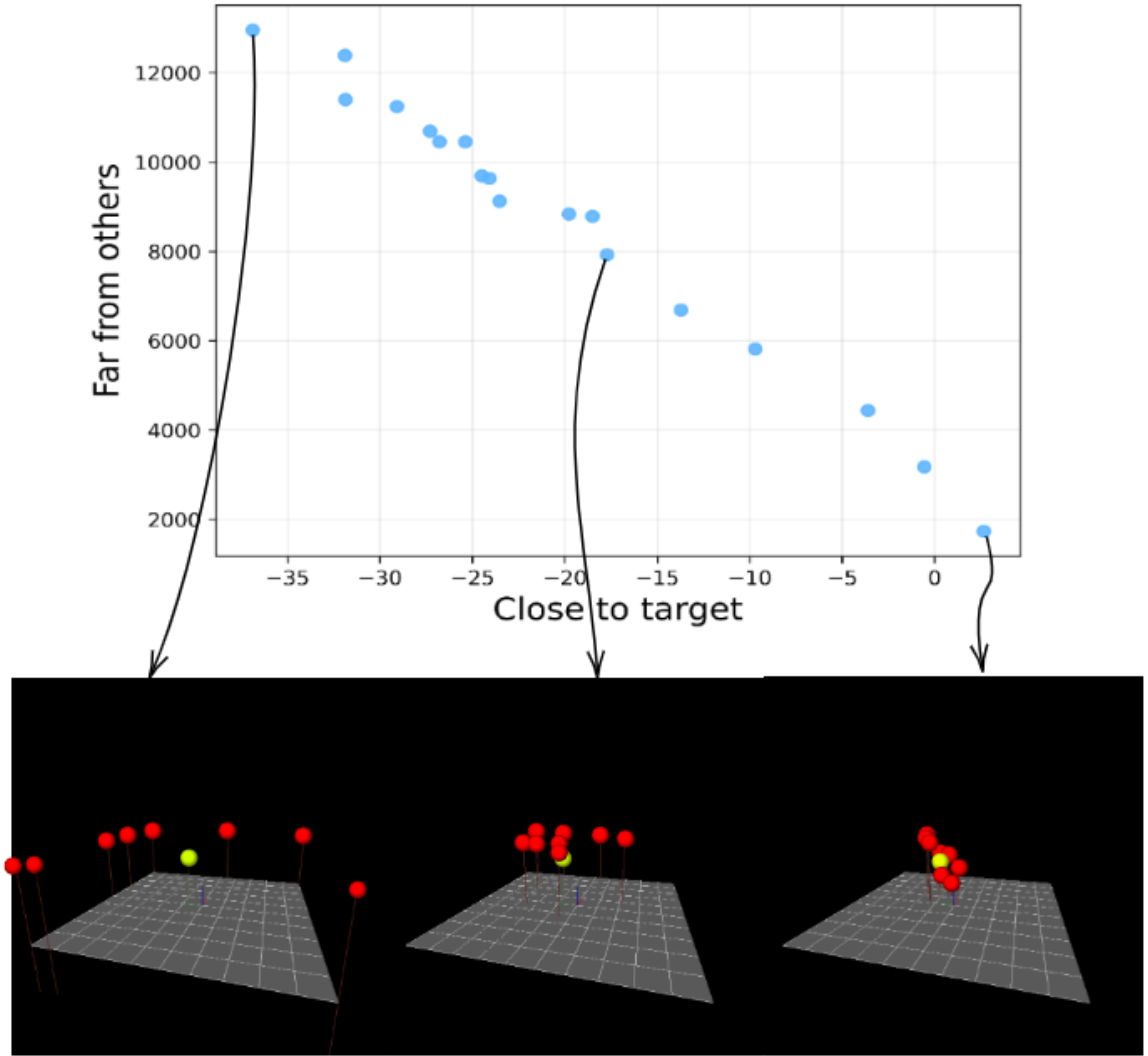
Training for various trade-offs in parallel on a GPU.

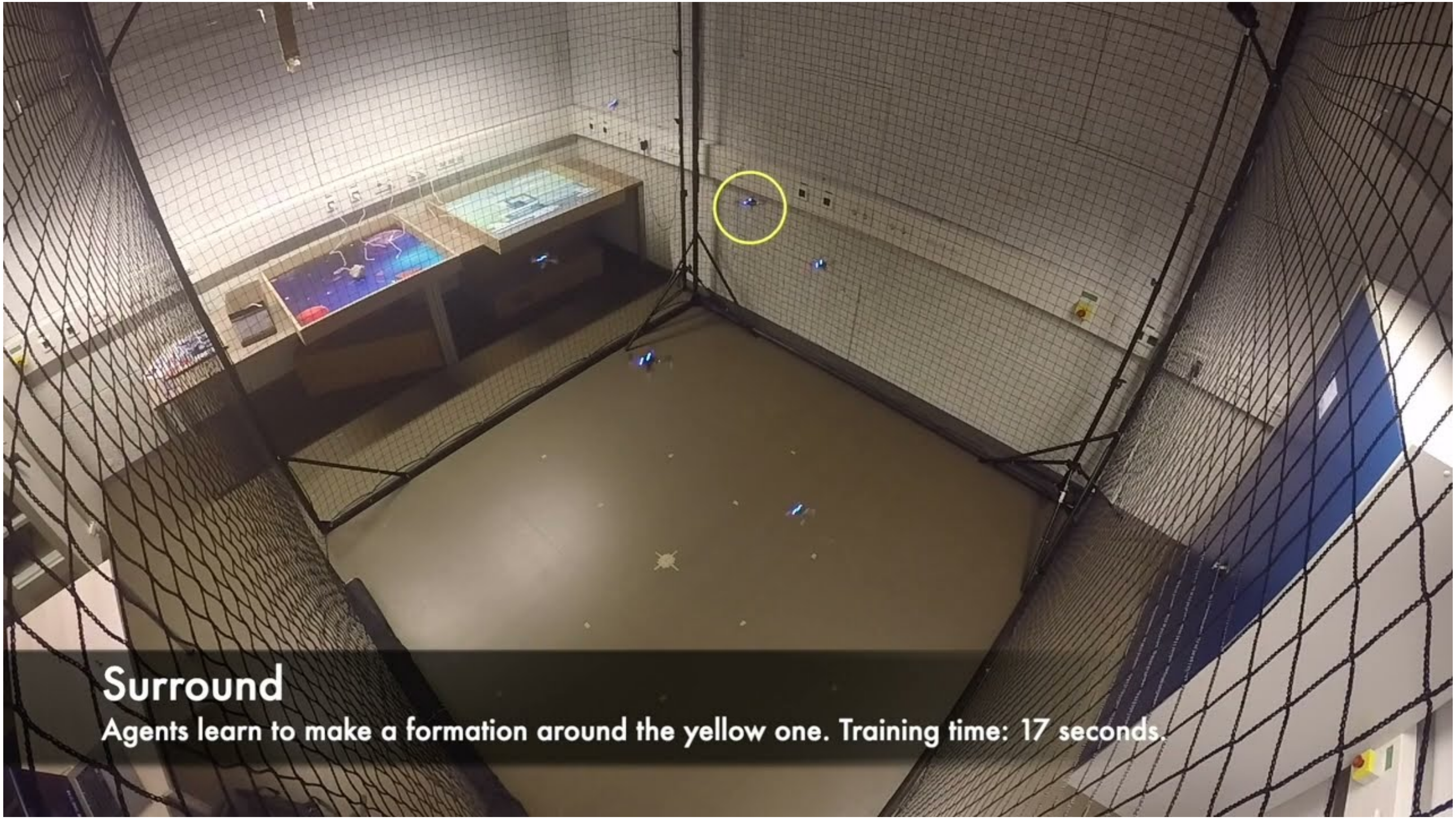
Number of policies	*1* (CPU)	*1*	*10*	*20*	*30*
Time	7228.6 ±22.8	10.6 ±0.3	35.9 ±0.9	56.9 ±0.4	78.8 ±0.8
SPS	415 ±1.3	282,251 ± 6809	837,251 ± 20,223	1,053,653 ± 7783	1,141,864 ± 10,858
Speedup	-	**≈680×**	**≈2017×**	**≈2539×**	**≈2751×**

Very few researchers look at wall-time in practice.

F. Felten, "Multi-Objective Reinforcement Learning," PhD Thesis, Université du Luxembourg, 2024.

Trade-offs


Far from others
12000
10000
8000
6000
4000
2000
−35
−30
−25
−20
−15
−10
−5
0
Close to target


Surround
Agents learn to make a formation around the yellow one. Training time: 17 seconds.

Wrapping up

- There are many problems which require optimizing multiple objectives
- Traditional (MA)RL overlook these aspects, and scalarizing rewards does not always give you what you want!
- MO(MA)RL are promising fields of research — lots of low hanging fruits
- We have tools for empirical evaluation — avoid the reproducibility crisis

Thank you!

ffelten@mavt.ethz.ch