WAE-PCN

Wasserstein Auto-encoded Pareto Conditioned Networks

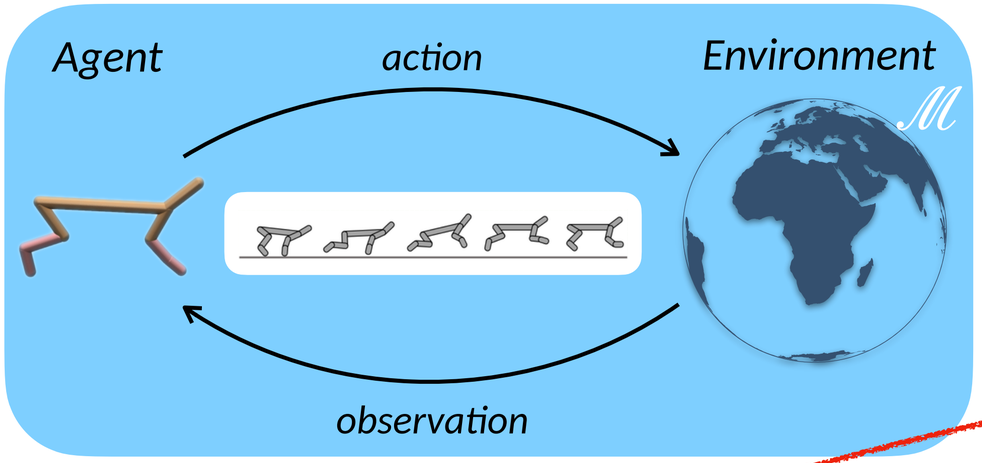
Florent Delgrange*, Mathieu Reymond*, Ann Nowé, Guillermo A. Pérez

*equal contribution

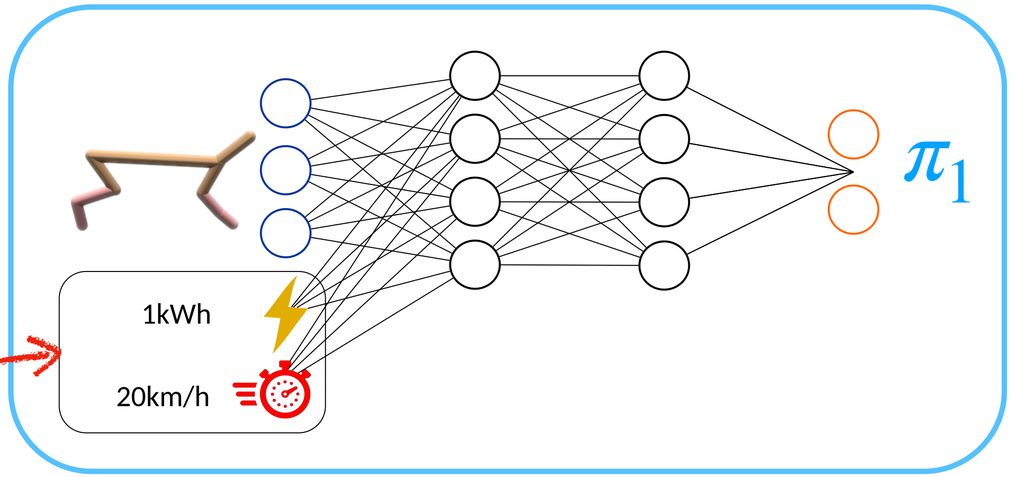
ALA 2023:
Adaptive and Learning Agents Workshop
at AAMAS, London, UK

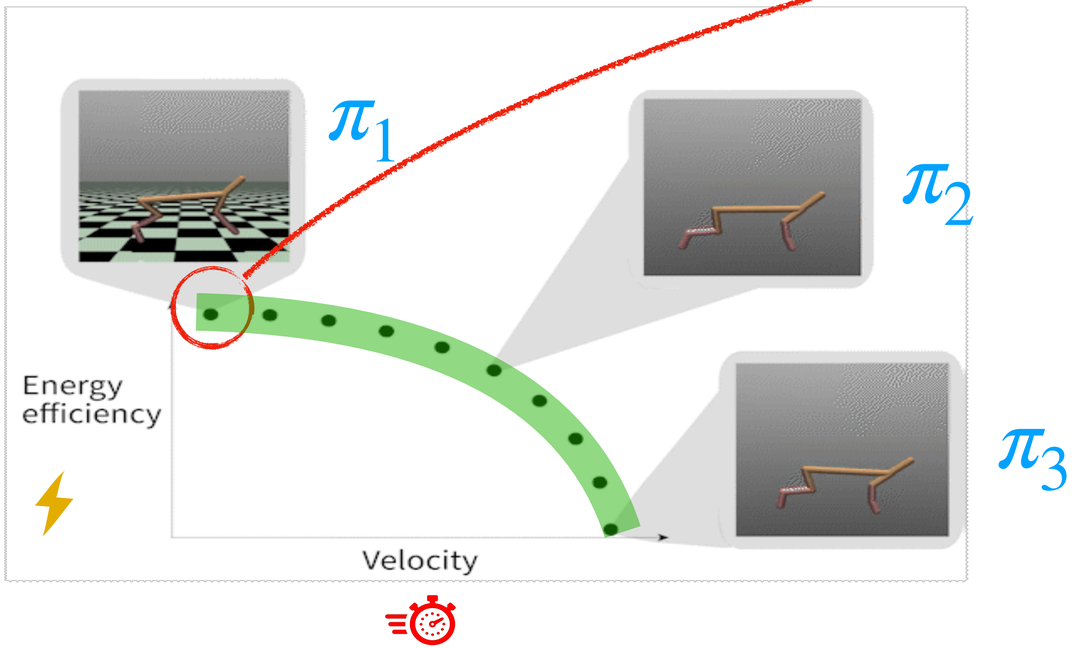
PCN: Pareto Conditioned Networks

Multi-objective Reinforcement Learning

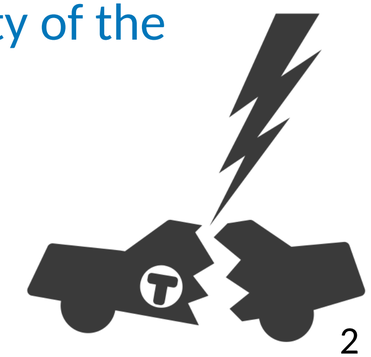


Pareto Conditioned Networks (PCN)



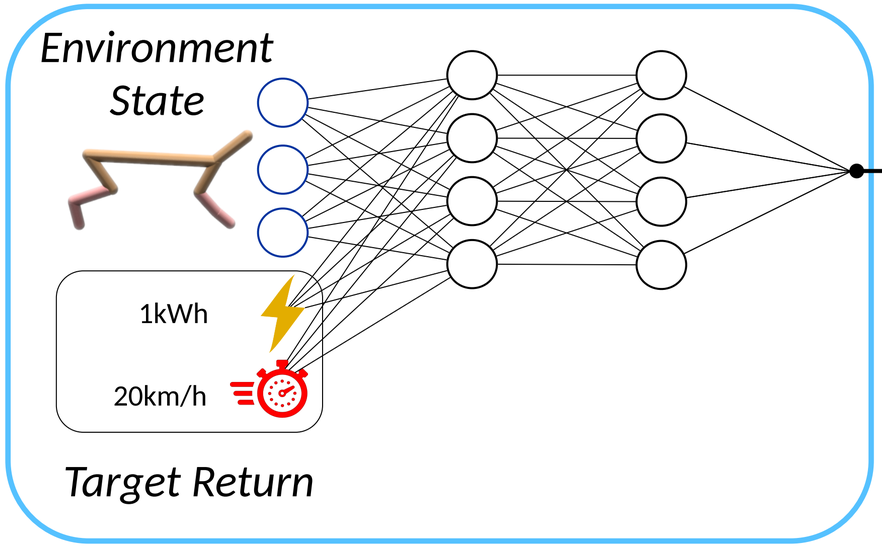


- Learn a set of ***Pareto-optimal*** policies
- Learn the set by ***conditioning π on the desired return to achieve*** (= the desired compromise between objectives)
- Do not take into account the stochasticity of the environment
  - ➡ Not robust to unexpected events
  - ➡ No guarantee that the target return will actually be achieved

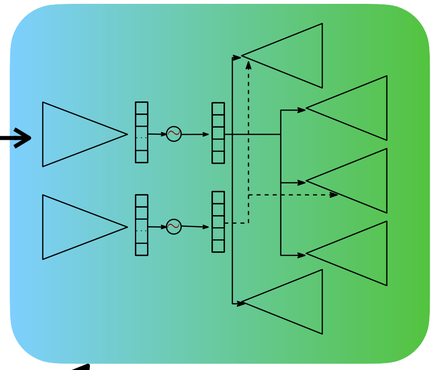


WAE-PCN: Wasserstein auto-encoded PCN

Pareto Conditioned Networks (PCN)



Wasserstein Auto-encoded Markov Decision Process

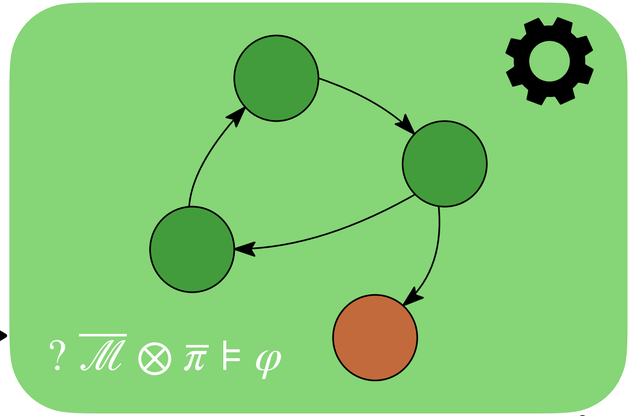


Abstraction

Latent Model $\overline{\mathscr{M}}$

Latent PCN $\bar{\pi}$

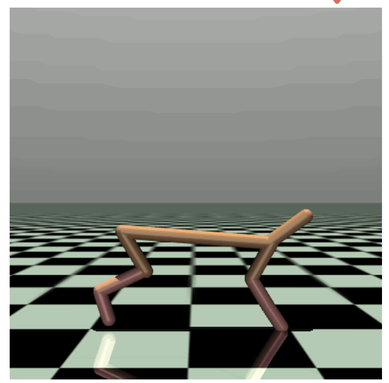
Model Checking



$\bar{\pi}^{\checkmark}$

verified policy

$? \; \pi^{\checkmark} \vDash \mathbb{P}_{\pi^{\checkmark}}(\text{Failure}) \leq 0.1$



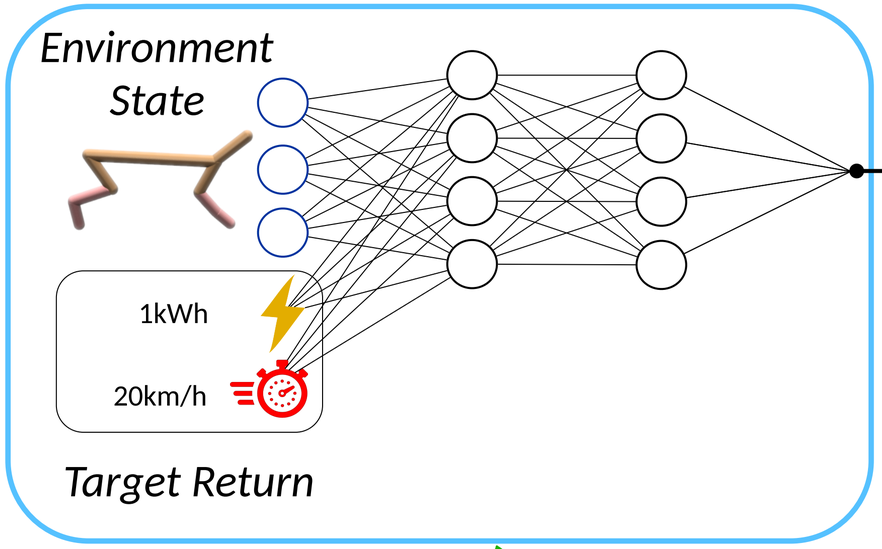
WAE-MDP guarantees:

- ***Bisimilarly close*** (= behaviourally equivalent) to the real environment
- The ***representation* of the state space** is guaranteed to ***capture the necessary information to optimise the policy***
- ***Formally Verifiable***

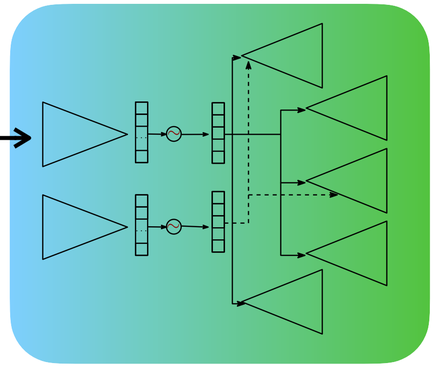
➡ Allows to check ***properties***, i.e., that the agent behaves as expected

WAE-PCN: Wasserstein auto-encoded PCN

Pareto Conditioned Networks



Wasserstein Auto-encoded Markov Decision Process

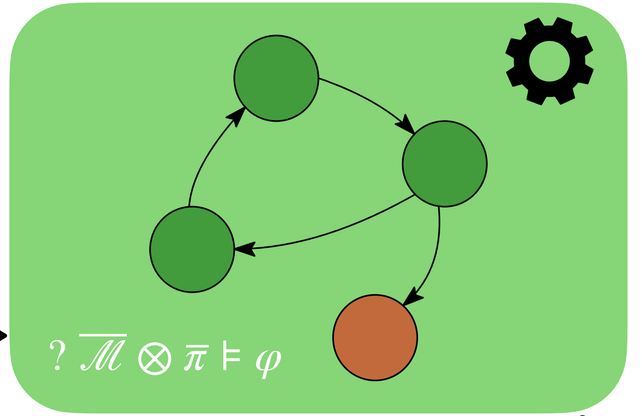


Latent Model $\overline{\mathscr{M}}$

Latent PCN $\bar{\pi}$

Property φ

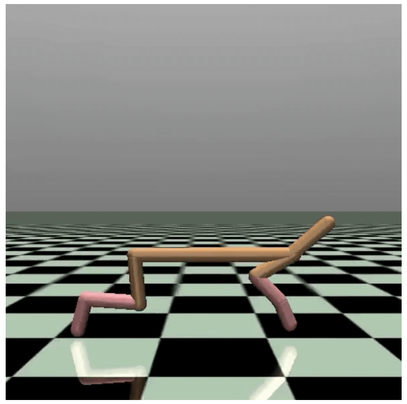
Model Checking



$\bar{\pi}^{\checkmark}$

verified policy

- **Feedback**: Expected return / Probability of achieving the input target return
- PCN updates its set of Pareto-optimal policies w.r.t. the received feedback



Compute the expected return:

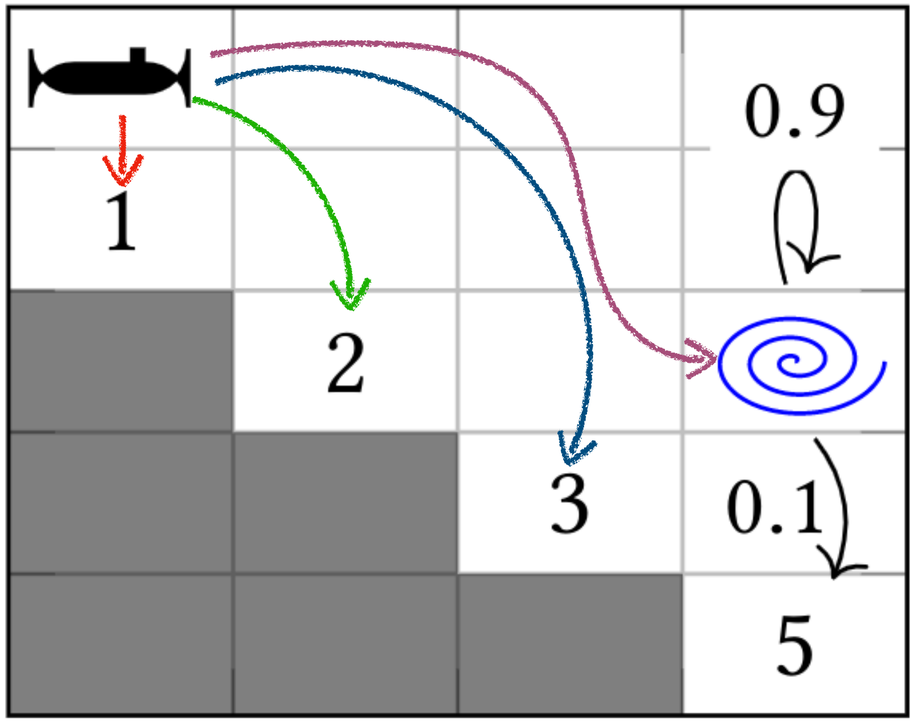
$$? \, \pi^{\checkmark} \vDash \mathbb{E}_{\pi^{\checkmark}}\left(\text{Return}\right) = \langle 1\text{kWh}, 20\text{km/h} \rangle$$

Check that the PCN policy achieves the target return:

$$? \, \pi^{\checkmark} \vDash \mathbb{P}_{\pi^{\checkmark}}\left(\text{Achieve return } \langle 1\text{kWh}, 20\text{km/h} \rangle\right) \geq 0.99$$

Experiments

Deep Sea Treasure

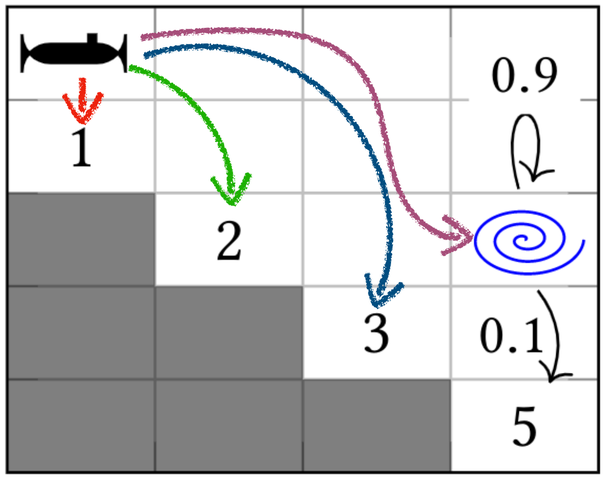


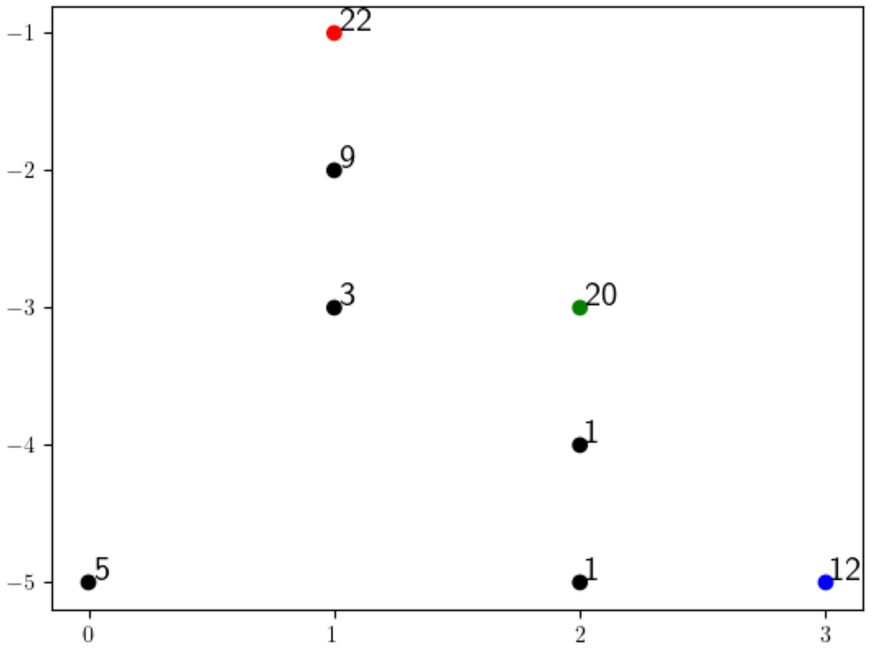
- π_1= (1,-1)
- π_2= (2,-3)
- π_3= (3,-5)
- π_4= 0.1 (5,-5) + 0.9 (0,-5) = (0.5, -5)

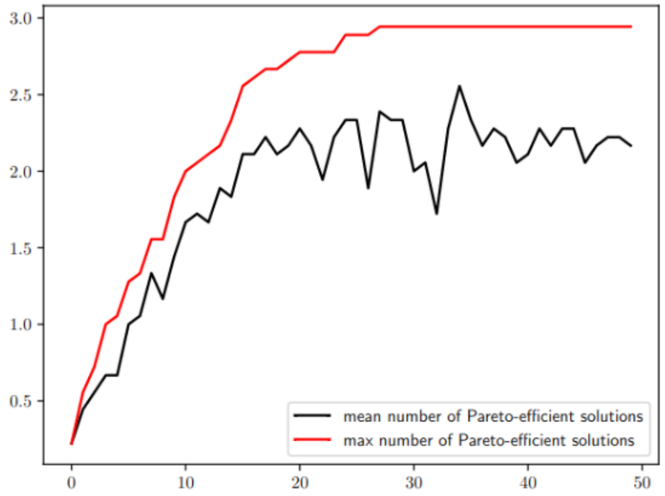
- Pareto front contains π_1, π_2, π_3
- Vanilla PCN **removes** π_3 and **keeps** π_4 !!
- WAE-PCN, because it learns the transition probabilities, ***learns to remove*** π_4

Experiments

Deep Sea Treasure







(c) Number of Pareto efficient policies over training time. Red illustrates that each run learns the full Pareto front during training.

More than half of the runs learn the full Pareto front

- *WAE-PCN is **able to learn the full Pareto front over the course of training**…*
- *… but sometimes struggles to keep it*
- *Stabilities issues due to the competition between PCN and WAE-MDPs*

Want to know more?

- Come to our poster!
- Check out the paper: