When RL and Control meet: lessons learned

Ann Nowé

AI lab
Vrije universiteit Brussel

ARTIFICIAL INTELLIGENCE LAB

LeCoPro

Learning Control of Production Machines

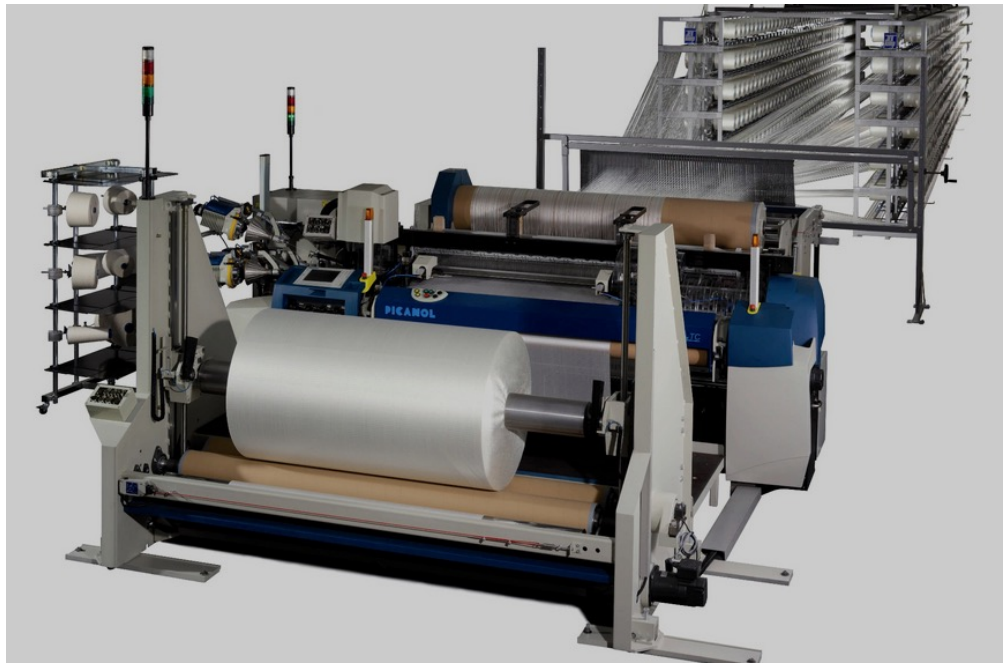
2009-2013

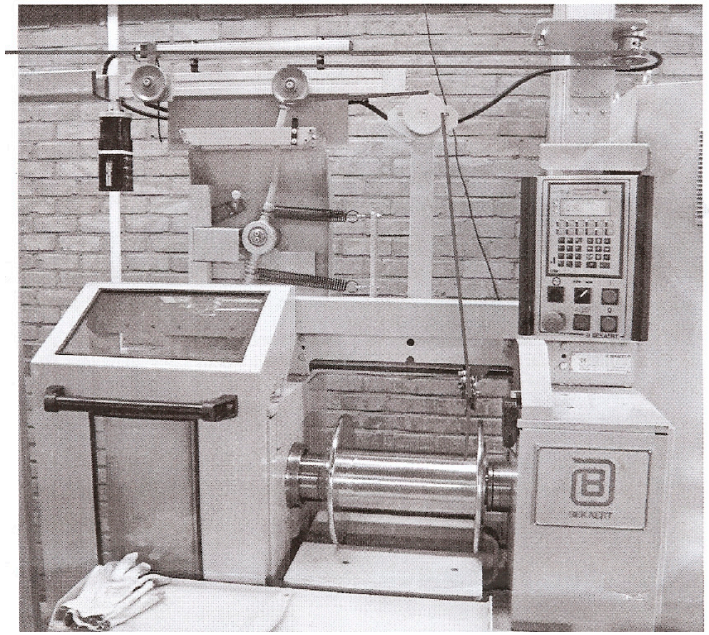
VUB ARTIFICIAL INTELLIGENCE LAB

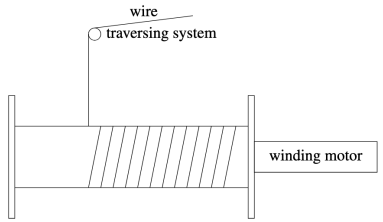
LeCoPro

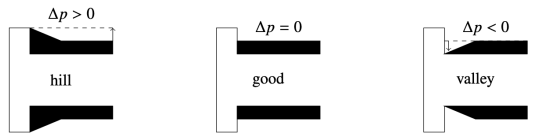
Learning Control of Production Machines

2009-2013







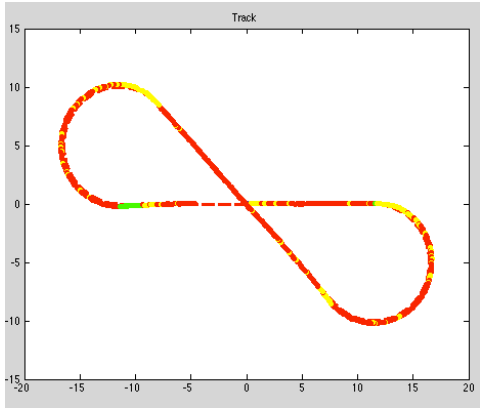


$r_k = -\Delta p$

VUB ARTIFICIAL INTELLIGENCE LAB

LeCoPro
Learning Control of Production Machines

- Basic PID to bootstrap RL-algorithm, i.e. to provide initial behavior to the Fuzzy Q-learning algorithm
- RL gradually improves with PID as backup, i.e. when RL proposes a potentially unsafe action, it can be overruled by the PID controller

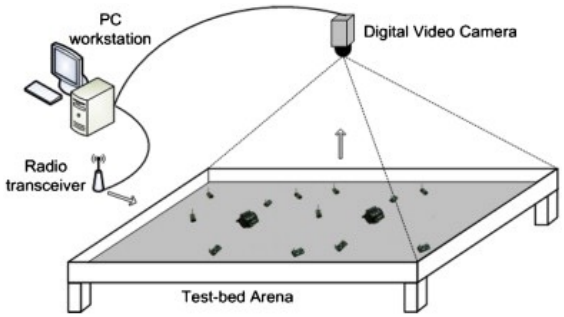


Red: Reinforcement Learning
Yellow: PID (RL uncertain)
Green: PID (out of bounds)

LeCoPro

Learning Control of Production Machines

2009-2013



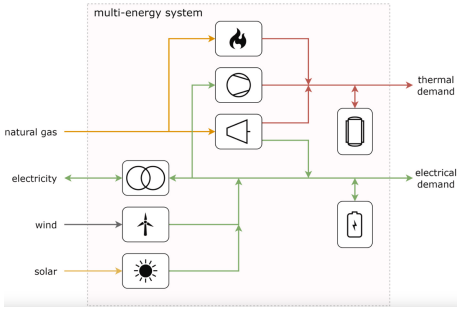
VUB ARTIFICIAL INTELLIGENCE LAB

Energy and AI
Volume 12, April 2023, 100227

Safe reinforcement learning for multi-energy management systems with known constraint functions

Glenn Ceusters [a b c], Luis Ramirez Camargo [b d], Rüdiger Franke [a], Ann Nowé [c], Maarten Messagie [b]

VLAIO Baekeland PhD with ABB



Algorithm 1: SafeFallback

Input: initialize RL algorithm, initialize constraint functions in sets X and U, initialize safe fallback policy π^{safe}
for $k = 0, 1, 2, \ldots$ **do**
  Observe state s and select action a
  if *constraint check* = *True* **then**
    keep selected action a as safe action a^{safe}
  else
    get safe action a^{safe} from safe fallback policy π^{safe}
  end
  Execute a^{safe} in the environment
  Observe next state s', reward r and done signal d to indicate whether s' is terminal
  Give experience tuple (s, a^{safe}, r, s', d) **and if** $a^{safe} \neq a$: $(s, a, r - c, s', d)$ with cost c
  If s' is terminal, reset environment state
end

Algorithm 2: GiveSafe

Input: initialize RL algorithm, initialize constraint functions in sets X and U
for $k = 0, 1, 2, \ldots$ **do**
  Observe state s and select action a
  if *constraint check* = *True* **then**
    keep selected action a as safe action a^{safe}
  else
    while *constraint check* = *False* **do**
      give experience tuple (s, a, c, s, d) with cost c
      agent selects new action a
      check constraints
    end
    return safe action a^{safe}
  end
  Execute a^{safe} in the environment
  Observe next state s', reward r and done signal d to indicate whether s' is terminal
  Give experience tuple (s, a^{safe}, r, s', d)
  If s' is terminal, reset environment state
end

Highlights

- RL learns policies that outperform MPC
- A (near-to) optimal multi-energy management policy can be learned safely.
- Constraints can be formulated independently from the (optimal) control technique
- Better policies can be found starting with an initial safe fallback policy.

VUB ARTIFICIAL INTELLIGENCE LAB

Ann Nowé

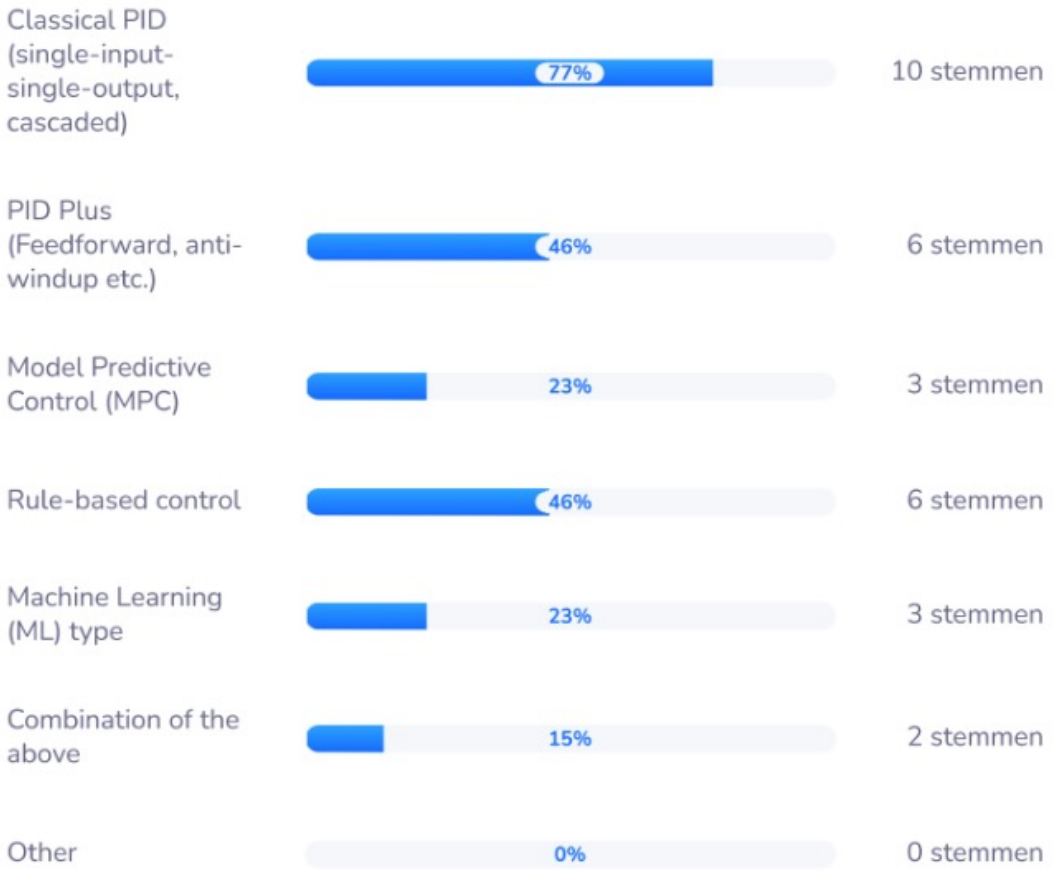
SAFE Reinforcement Learning

CTRL x AI

SBO with UGent

2. Do you have control structures available in your machines/robots/processes? If so, what type of controllers do you typically use right now?

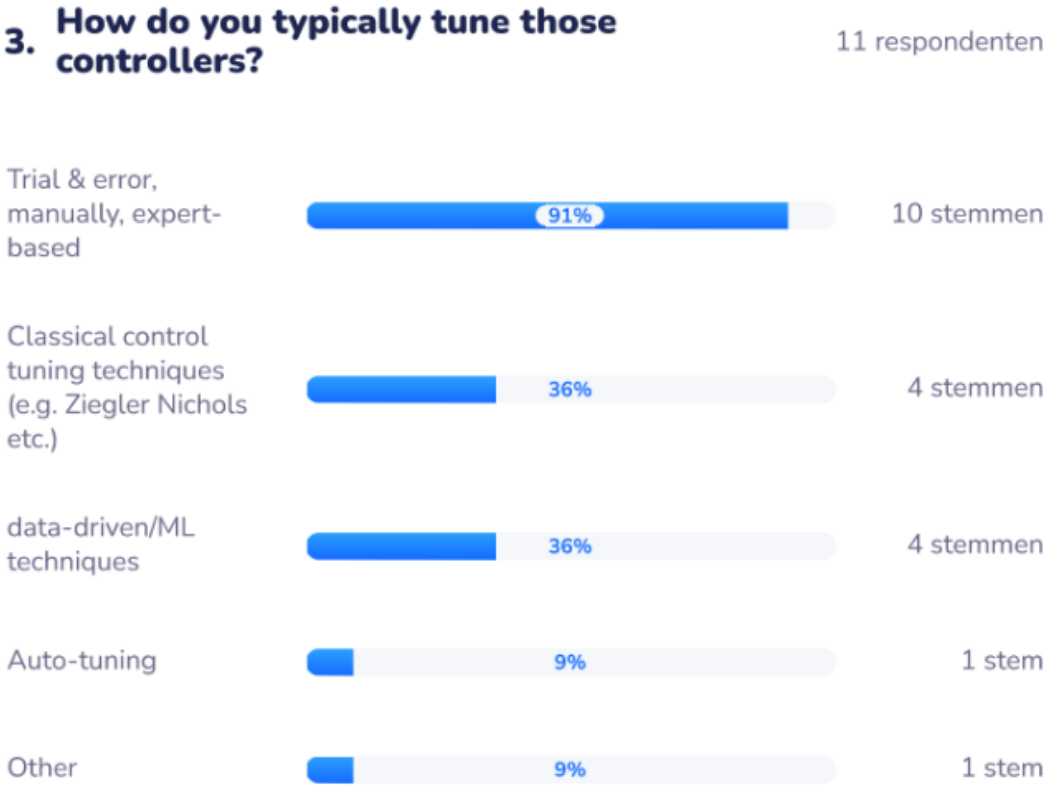
13 respondenten



VUB ART INTE LAB VUB NCE

CTRL x AI

SBO with UGent



ARTIFICIAL INTELLIGENCE LAB

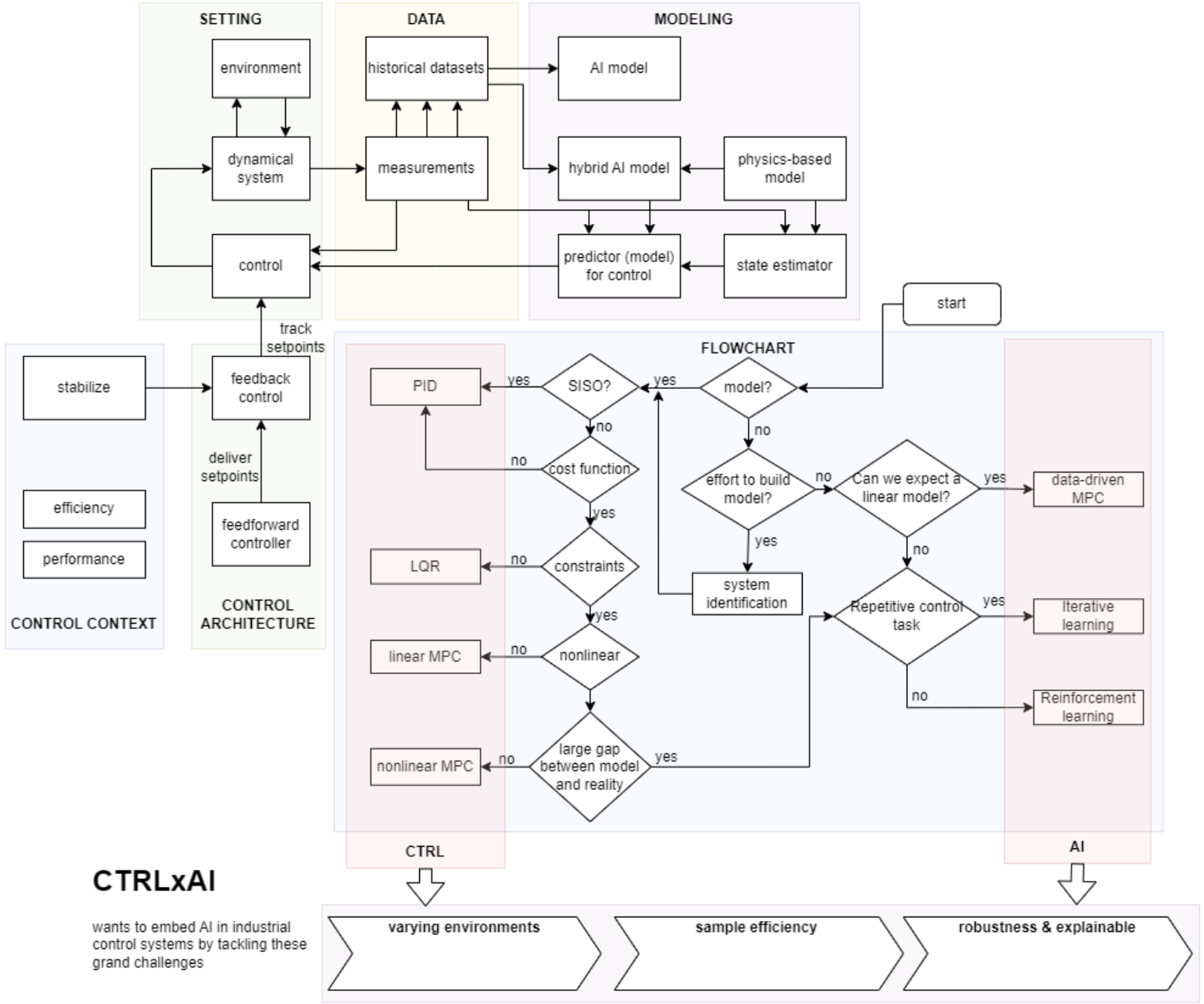
CTRL x AI

SBO with UGent

VUB ARTIFICIAL INTELLIGENCE LAB

CTRL x AI

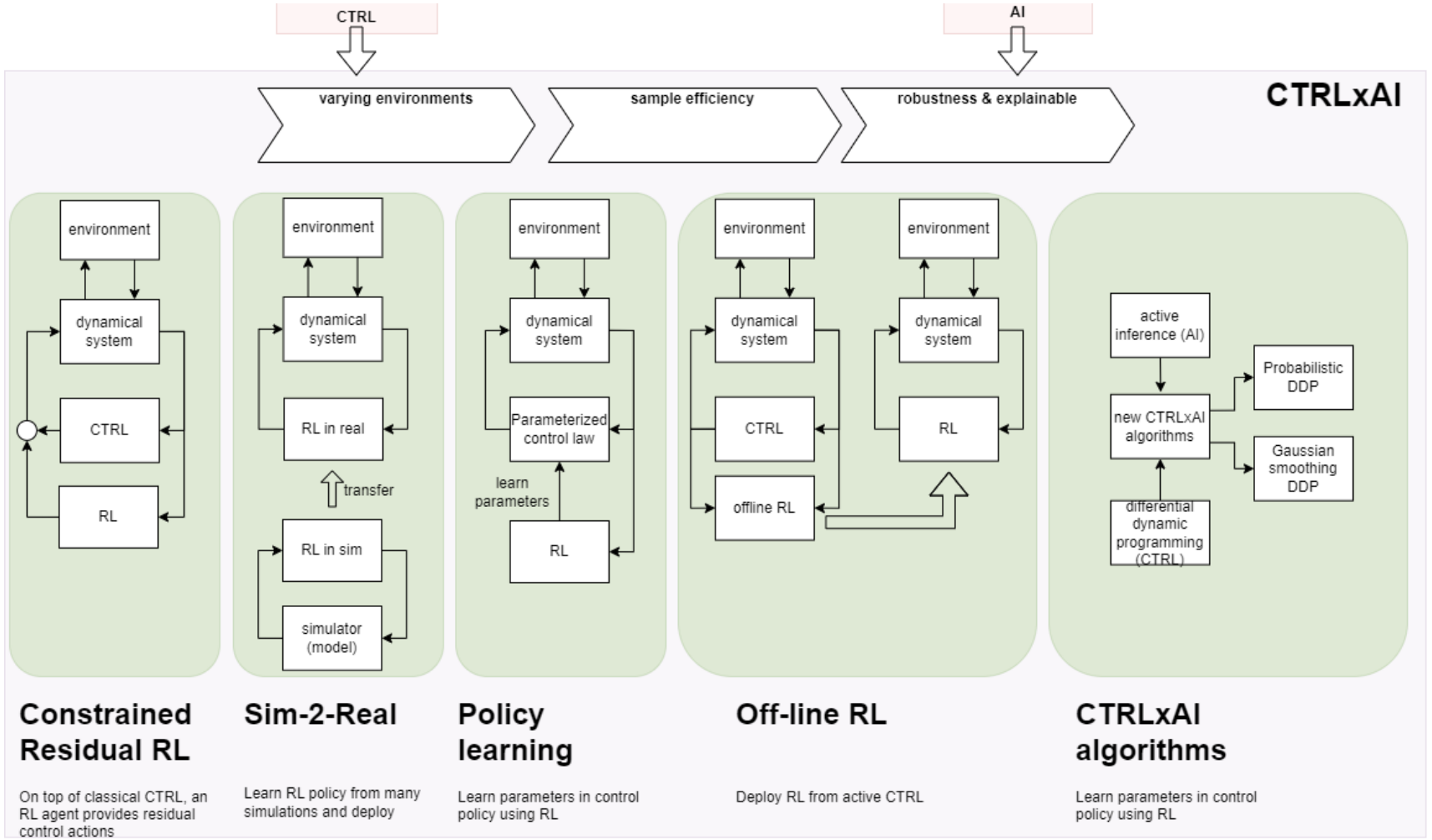
SBO with UGent



VUB ARTIFICIAL INTELLIGENCE LAB

CTRL x AI

SBO with UGent

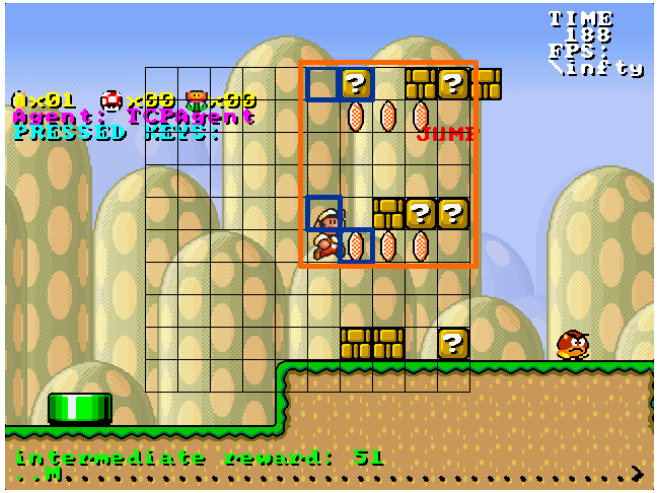


VUB ARTIFICIAL INTELLIGENCE LAB

Transparency / Interpretability / Explainability

Rule Distillation

- IF COND1 AND COND2 AND …. THEN ACTION1

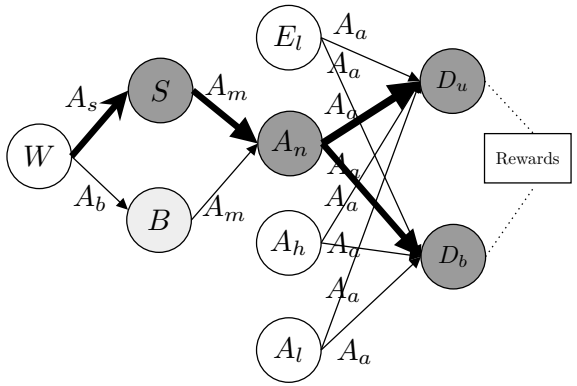


IF (1, 5)=BRICK **AND**
(1, 0)=COIN_RED
THEN Action=RIGHT

Action Influence Graphs

Structural Causal Models

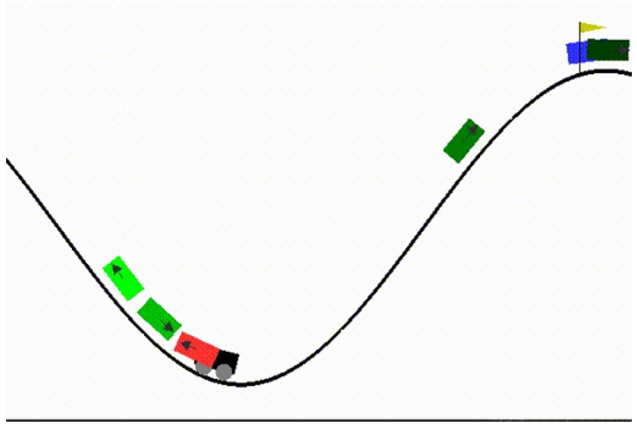
with actions



Question Why not *build_barracks* (A_b)?
Explanation Because it is more desirable to do action build_supply_depot (A_s) to have more Supply Depots (S) as the goal is to have more Destroyed Units (D_u) and Destroyed buildings (D_b).

I'm building up momentum

IF COND1 AND COND2 AND …. THEN ACTION1



ARTIFICIAL INTELLIGENCE LAB

Madumal, P., Miller, T., Sonenberg, L., Vetere, F.:
Explainable reinforcement learning through a causal lens. AAAI'20

Visualisation

ARTIFICIAL INTELLIGENCE LAB

Local Advantage Networks for Multi-Agent Reinforcement Learning in Dec-POMDPs
Raphaël Avalos et al, JMLR (2023)
Starcraft Multi-agent Challenge (SMAC)

Post-HOC Feature Importance

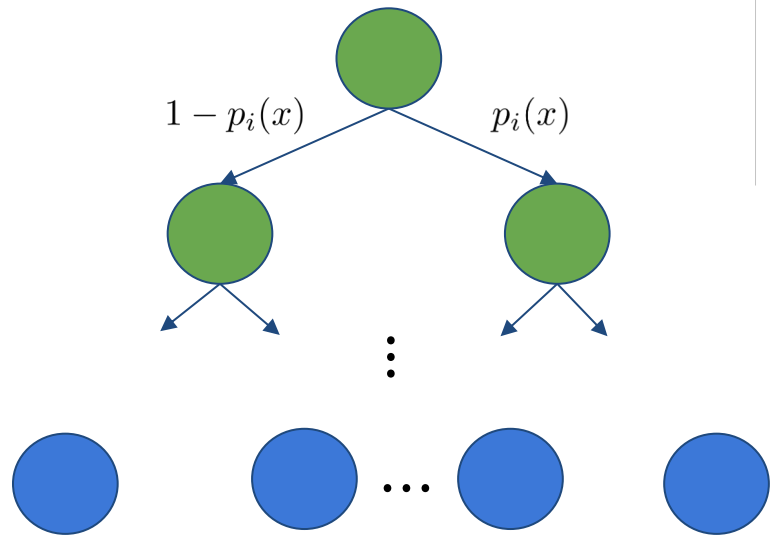
Soft Decision Trees (SDT)

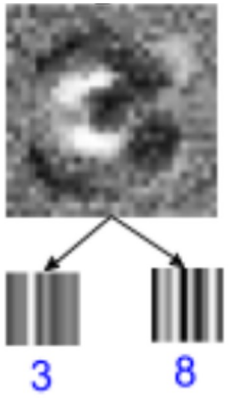
- Hybridization of Neural Networks and Decision Trees
  - Originally for image classification
- **Branching nodes** are perceptrons
  ⇒ **soft decisions for tree traversal**

$$p_i(x) = \sigma(\beta(xw_i + b_i))$$

- **Leaves** represent softmax distributions:

$$Q_k^l = \frac{\exp(\phi_k^l)}{\sum_{k'} \exp(\phi_{k'}^l)}$$

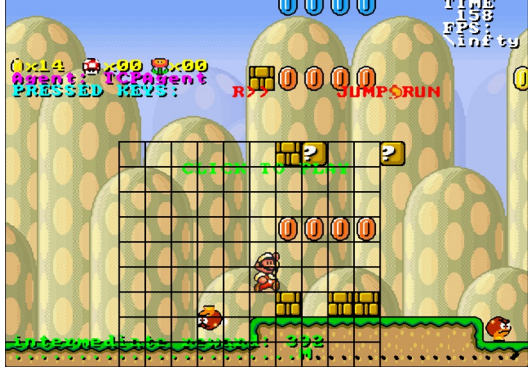




ARTIFICIAL INTELLIGENCE LAB

Frosst, N., & Hinton, G. (2017). Distilling a Neural Network Into a Soft Decision Tree. In T. R. Besold & O. Kutz (Eds.), *Proceedings of the First International Workshop on Comprehensibility and Explanation in AI and ML 2017* (Vol. 2071). Tarek R. Besold & Oliver Kutz.

Post-HOC Feature Importance



Soft Decision Trees (SDT)

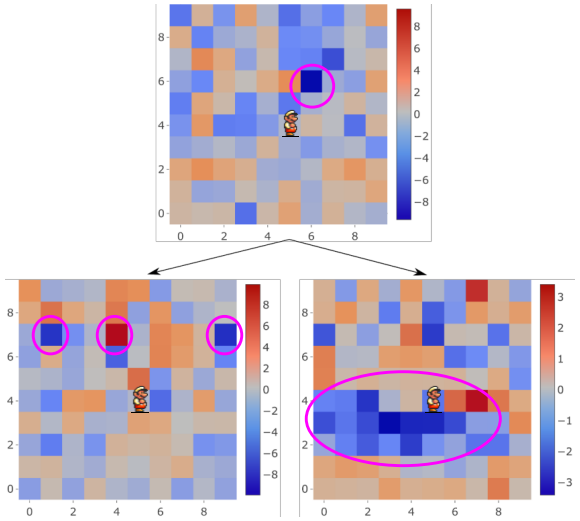
- Hybridization of Neural Networks and Decision Trees
  - Originally for image classification
- **Branching nodes** are perceptrons
  ⇒ **soft decisions for tree traversal**

$$p_i(x) = \sigma(\beta(xw_i + b_i))$$

- **Leaves** represent softmax distributions:

$$Q_k^l = \frac{\exp(\phi_k^l)}{\sum_{k'} \exp(\phi_{k'}^l)}$$

Examining branching nodes' filters shows important regions of the state

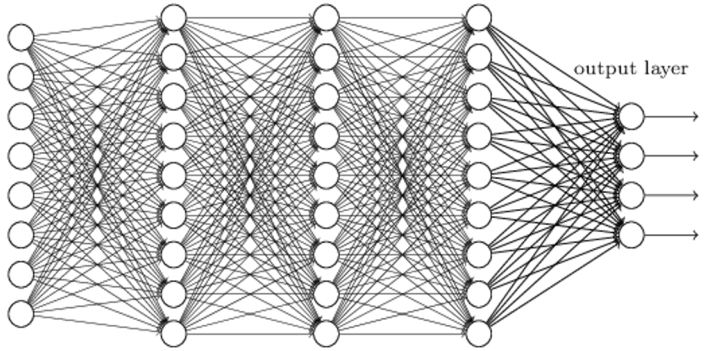


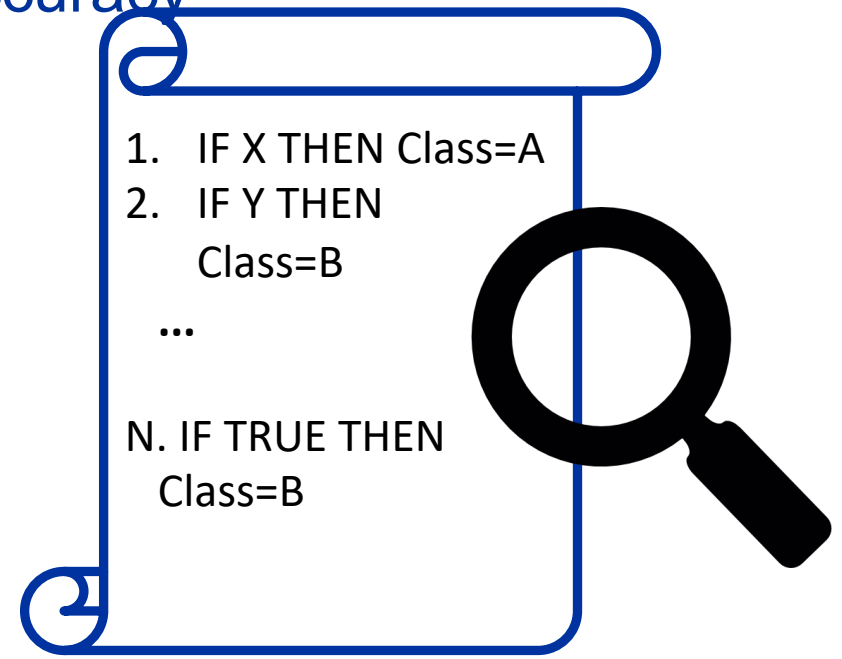
ARTIFICIAL INTELLIGENCE LAB

Coppens, Y., Efthymiadis, K., Lenaerts T., and Nowé A., (2019) Distilling Deep Reinforcement Learning Policies in Soft Decision Trees, XAI workshop at IJCAI 2019.

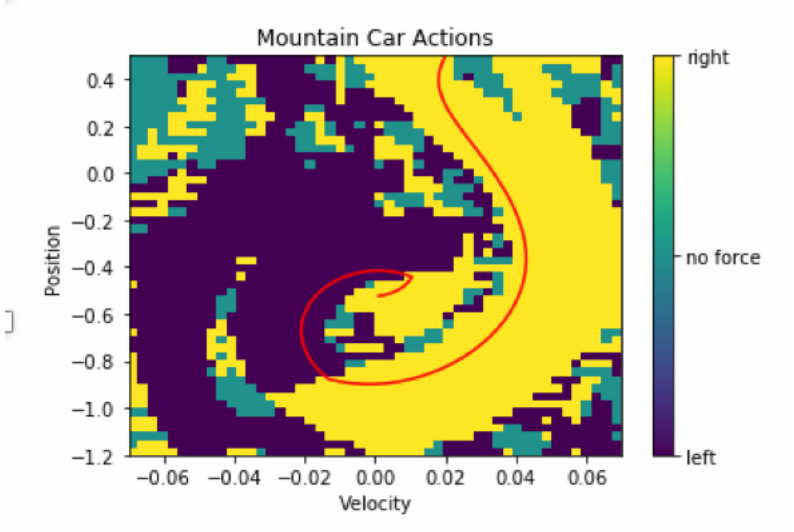
POST-HOC Policy distillation

- Summarize policy behavior into a surrogate white box model
- Driven by performance rather than model accuracy

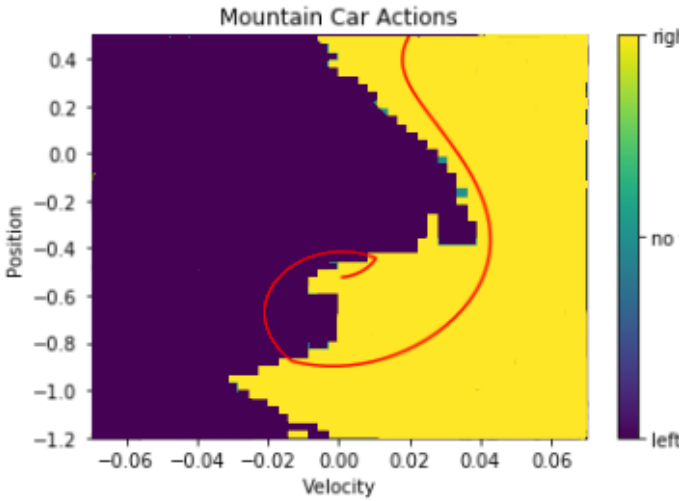




Exploiting meta information



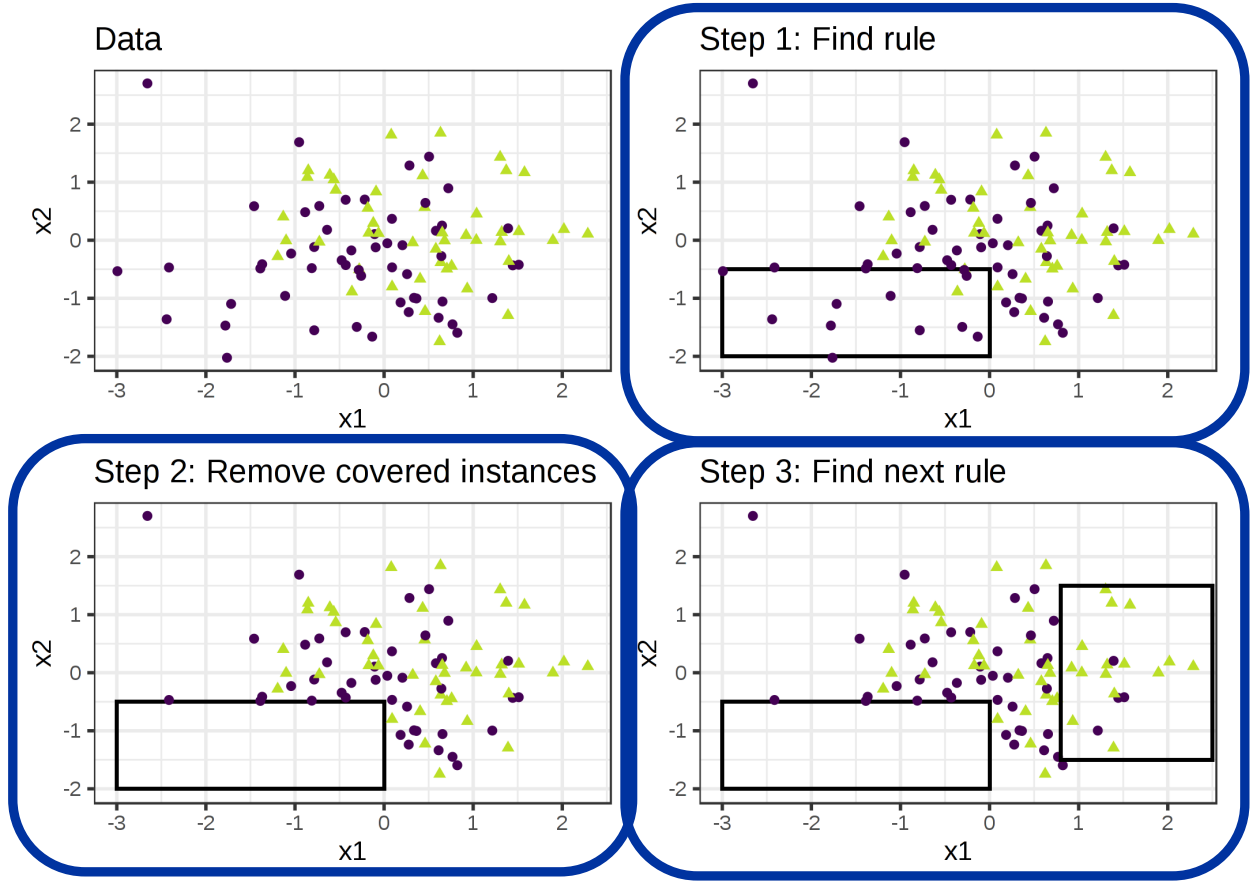
Greedy Policy



Distilled Policy

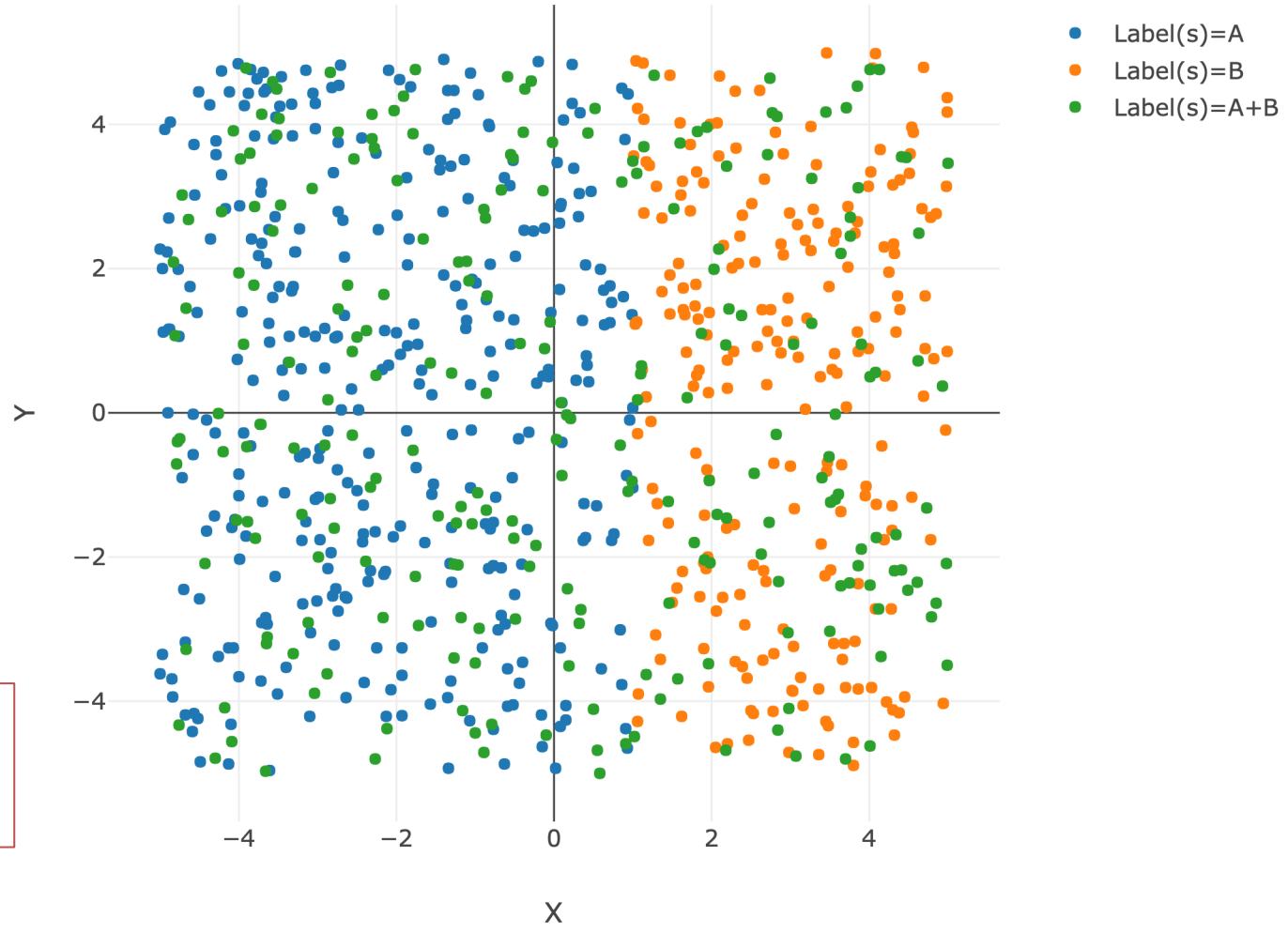
VUB ARTIFICIAL INTELLIGENCE LAB

Inductive Rule Learning



From *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*, by Christoph Molnar, 2019. https://christophm.github.io/interpretable-ml-book/. **(CC BY-NC-SA 4.0)**

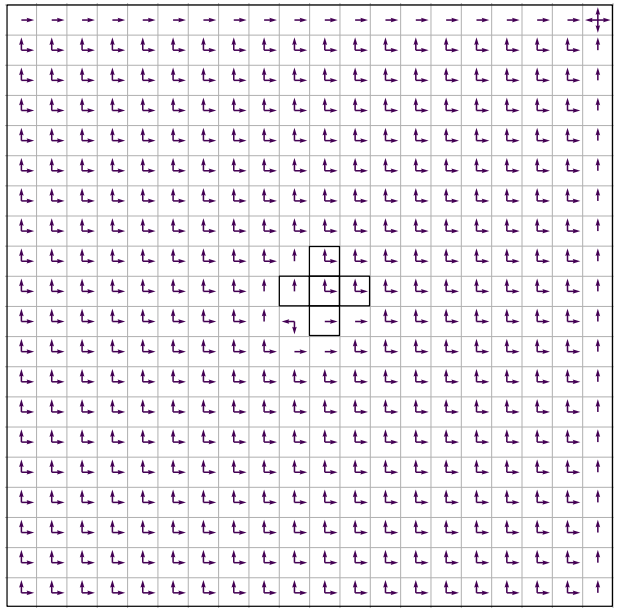
Mining rules from multi-labelled data



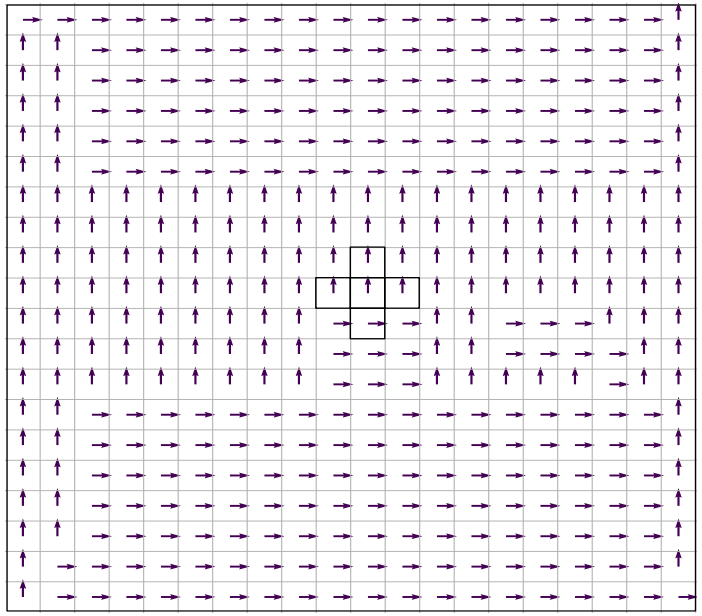
1. IF X<=1.0 THEN Class=A
2. IF X>=1.01 THEN Class=B

VUB ARTIFICIAL INTELLIGENCE LAB

Example Muddy world



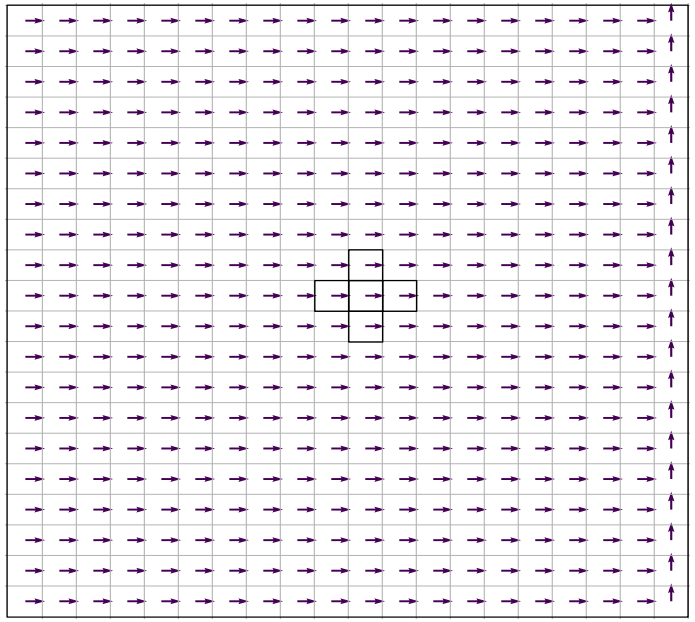
greedy policy



Rule mining on greedy policy

Phase 1

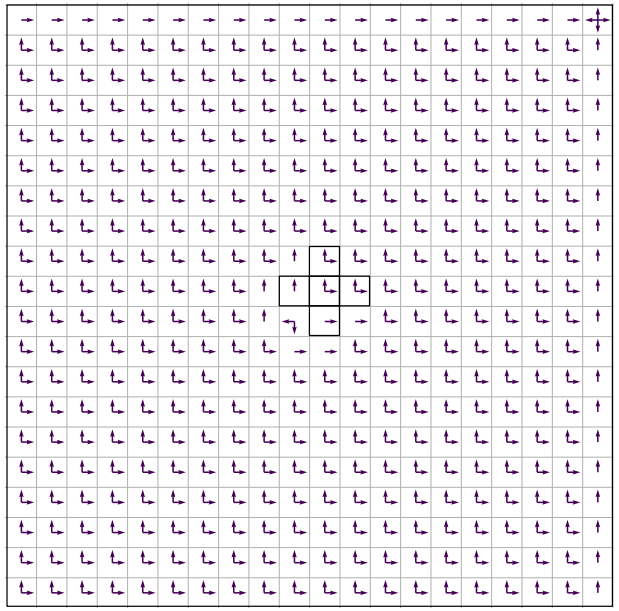
1. IF X<=18 THEN Class=RIGHT
2. IF X=19 THEN Class=UP

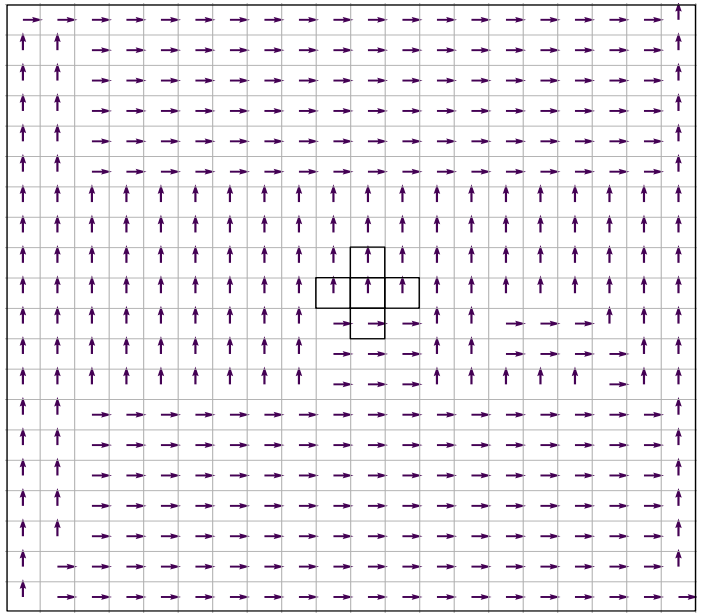


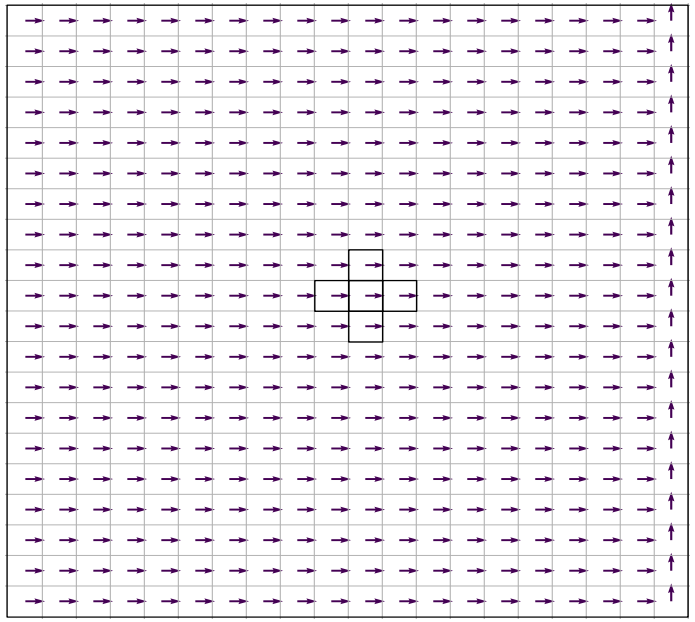
Example Muddy world

Phase 1

1. IF X<=18 THEN Class=RIGHT
2. IF X=19 THEN Class=UP





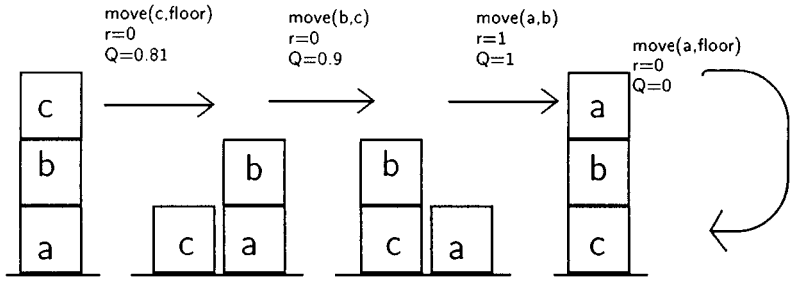


Phase 2

1.1 IF X<=18 AND Y=10 AND X>=8 AND X<=9 THEN Class=UP
1.2 IF X<=18 AND X>=10 THEN Class=RIGHT
1.3 IF X<=18 AND X<=8 THEN Class=RIGHT
1.4 IF X<=18 AND Y=11 THEN Class=UP
1.5 IF X<=18 AND Y=9 THEN Class=DOWN
1.6 IF X<=18 THEN Class=RIGHT
2 IF X=19 THEN Class=UP

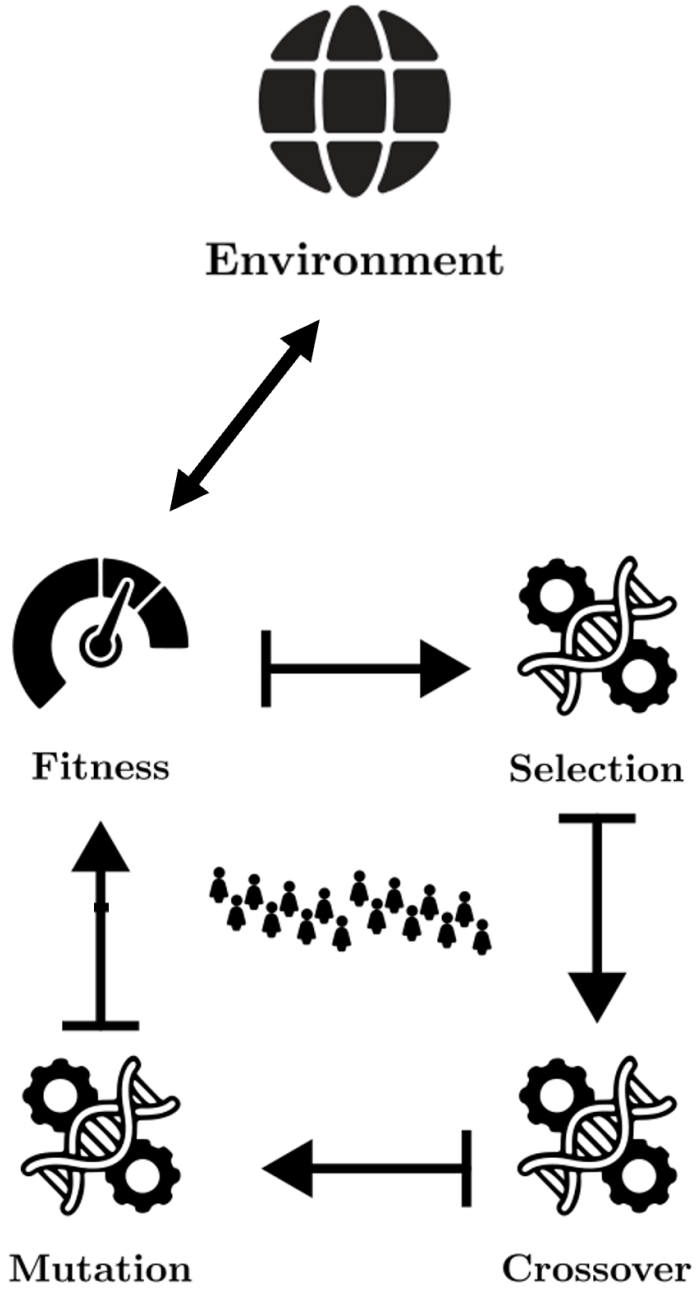
VUB ARTIFICIAL INTELLIGENCE LAB

Intrinsic transparency : **Relational RL**



Džeroski, S., De Raedt, L. & Driessens, K. Relational Reinforcement Learning. *Machine Learning* **43,** 7–52 (2001).

Intrinsic transparency



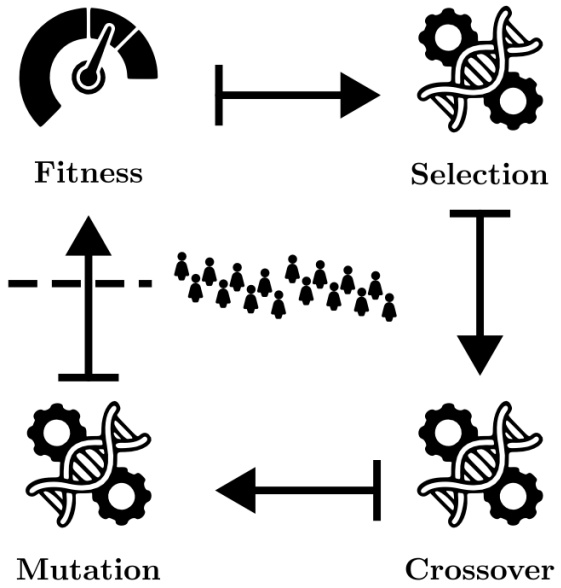
Utrecht University

BNAIC/BeNeLearn 2024

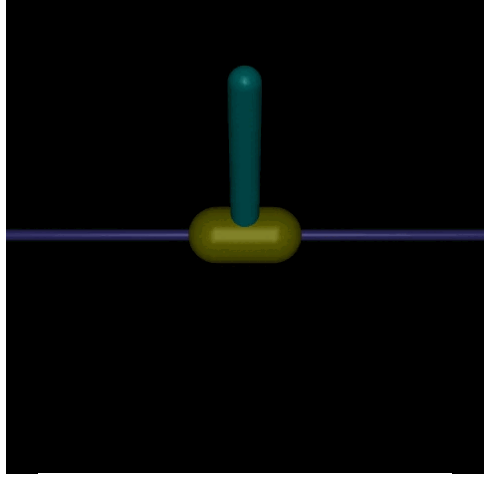
Critic-Moderated
Genetic Programming

VUB ARTIFICIAL INTELLIGENCE LAB

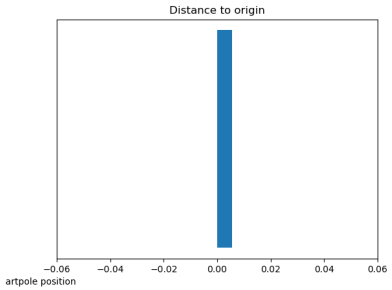
Intrinsic transparency

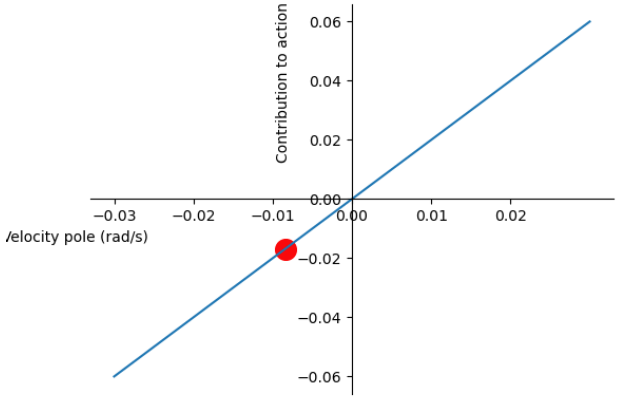


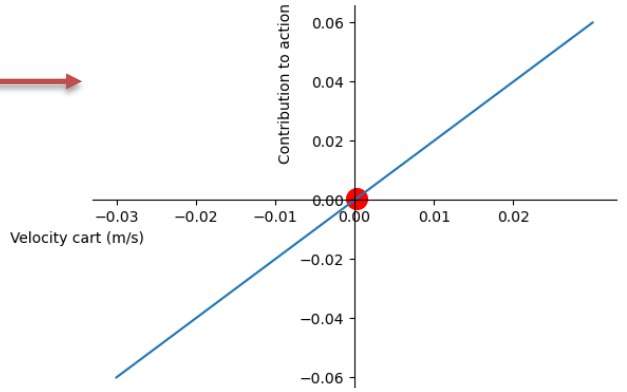
VUB ARTIFICIAL INTELLIGENCE LAB

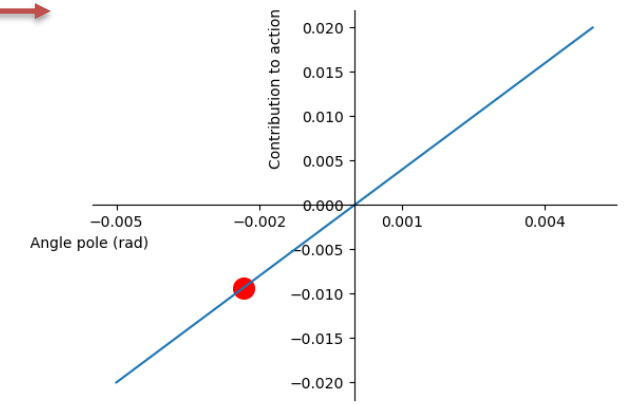


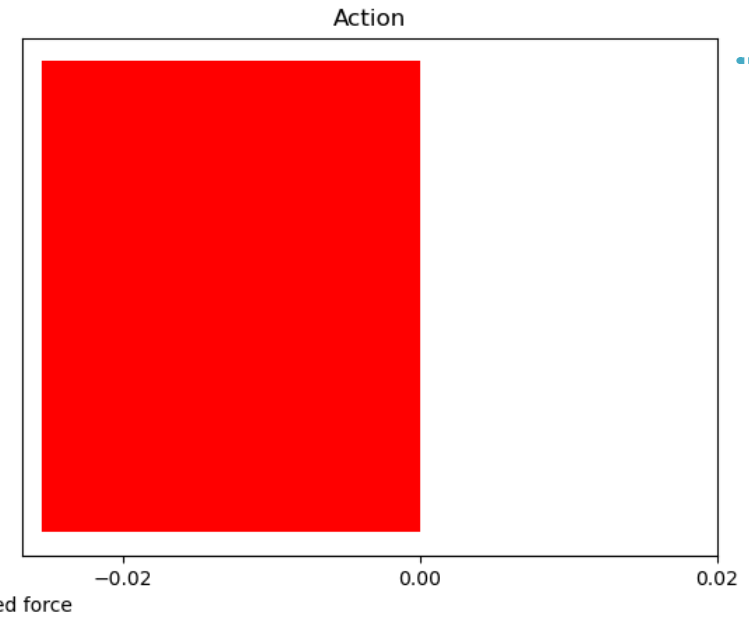
$$u = 4\angle_{pole} + 2v_{cart} + 2v_{pole}$$





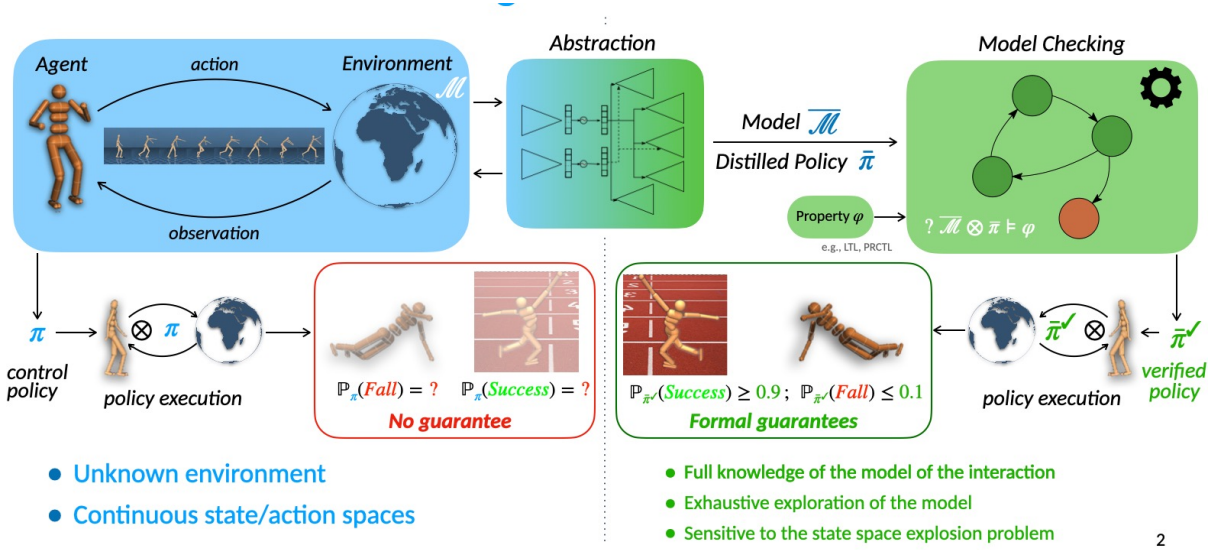






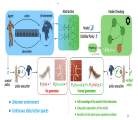
ARTIFICIAL INTELLIGENC LAB

Reinforcement Learning with Formal Guarantees



Framework for learning **discrete latent models** of **unknown continuous-spaces** environment with **bisimulation guarantees**

- Can be learned by executing an RL policy in the environment
- Yields a **distilled version of the RL policy**
- **New local losses bounds** for (i) bisimulation guarantees (ii) discrete setting (iii) action embedding function
- **PAC schemes** to formally retrieve confidence metrics to asses the quality of the learned model



Distillation of RL Policies with Formal Guarantees via Variational Abstraction of Markov Decision Processes
Florent Delgrange, Ann Nowé, Guillermo A. Pérez

AAAI-22

Wasserstein Auto-encoded MDPs: Formal Verification of Efficiently Distilled RL Policies with Many-sided Guarantees
Florent Delgrange, Ann Nowé, Guillermo A. Pérez

VUB ARTIFICIAL INTELLIGENCE LAB