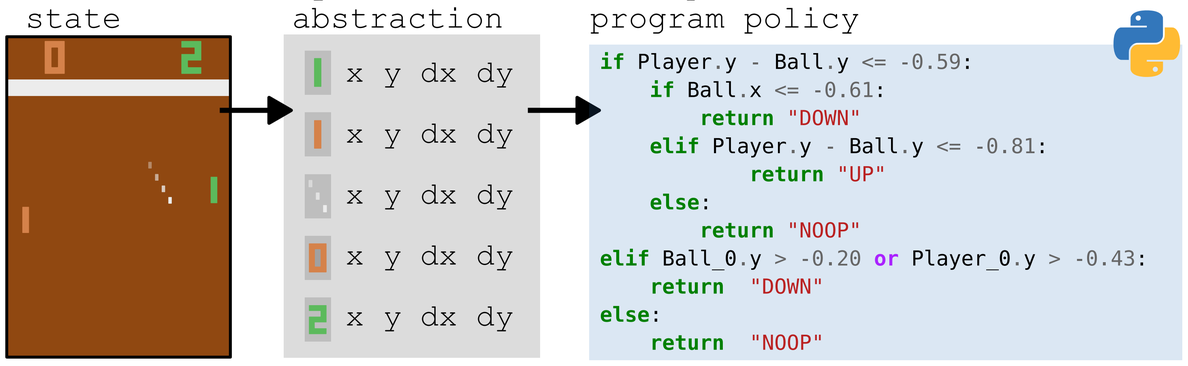
Interpretable and Editable Programmatic Tree Policies for Reinforcement Learning

Hector Kohler [*,1] Quentin Delfosse [*,2,3] Riad Akrour [1] Kristian Kersting [2,4] Philippe Preux [1]

Reinforcement Learning Conference

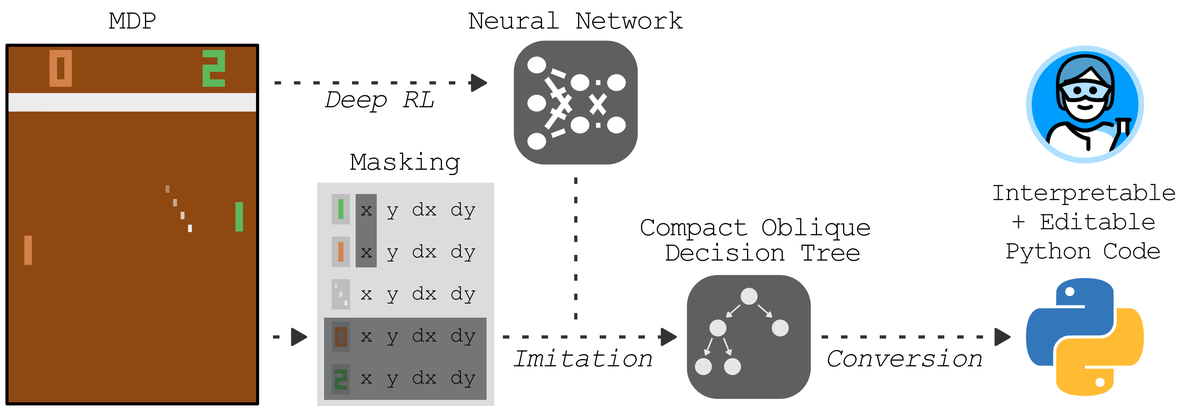
We can extract readable python policies from black box deep RL agents.

Goal: Readable RL policies

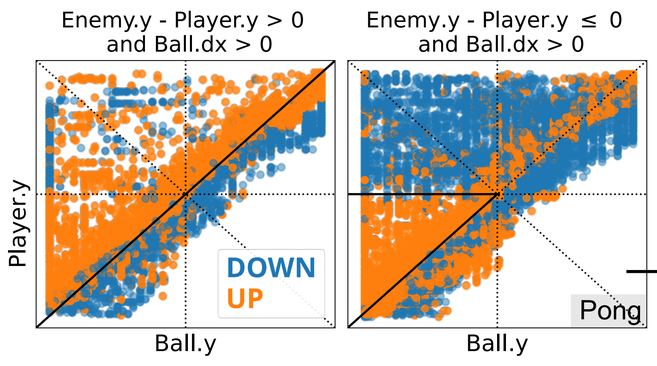


Short oblique decision trees are extracted from deep RL policies, and then **converted** into **readable and editable python programs**.

INTERPRETER: oblique trees to programs



Short oblique decision trees are extracted from deep RL policies, and then **converted** into **readable and editable python programs**.



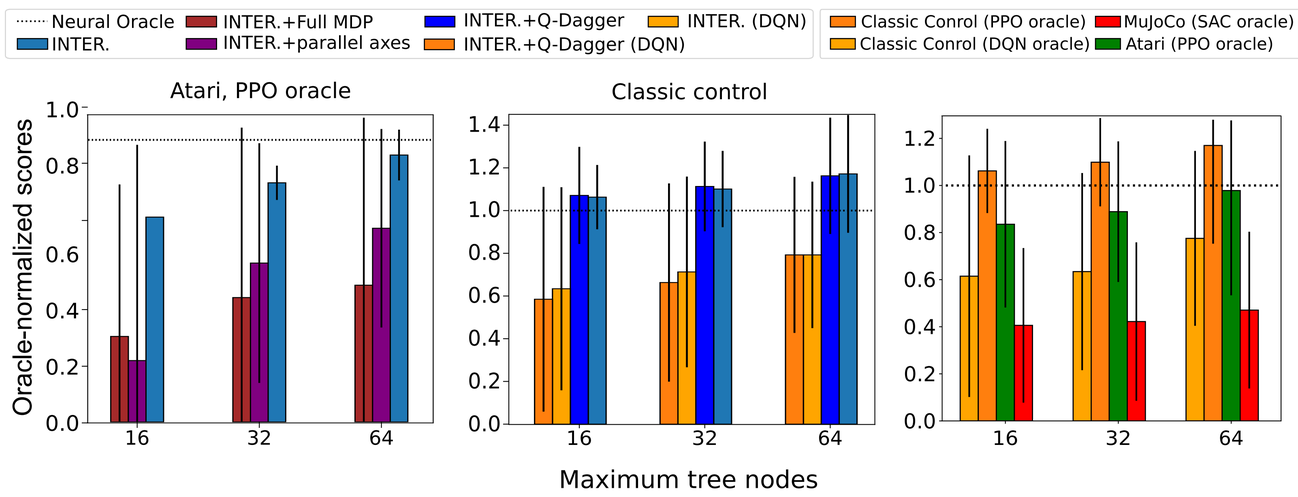
MDP	Aste.	Fre.	Kang.	Pong.	Sea.
Full	100	48	196	12	172
Simplified	90	22	28	8	54

↳ **Redundant** and **constant** features can be **pruned out**.

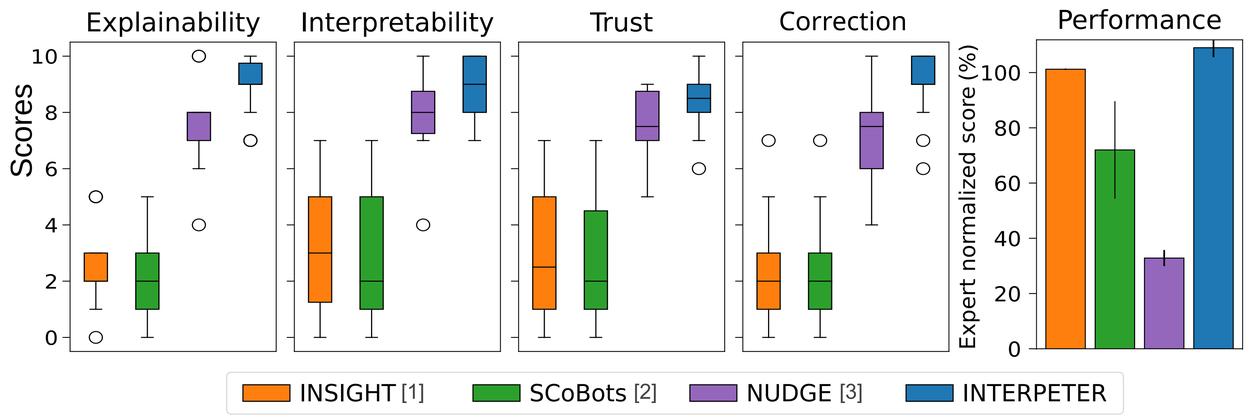
→ Most decisions are **oblique** (based on features' mixtures).

Deep agents' **decision boundaries are oblique** to the NS features, and **MDPs can be simplified**, leading to more **concise policies**.

Results: Interpretable competitive Programs



INTERPRETER's performances drop without **simplified MDPs** and **oblique decision boundaries**. Overall, the **extracted programs match their oracles' performances** on **Classic Control** and **Atari**. For mujoco, the learned programs **achieve the control goal.**



Our user study on Pong demonstrates **INTERPRETER's superior explainability**, **interpretability**, **trust** and **correction** abilities, while **outperforming** all the other interpretable baselines.

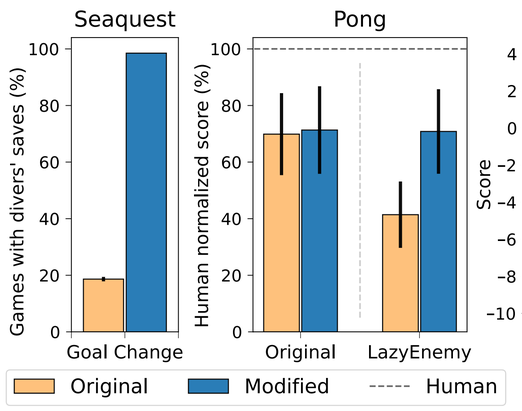
Conclusion

INTERPRETER provides **short readable** and **editable tree programs,** that **do not rely on expert provided prior knowledge.**

Pruning redundant MDP features and **using oblique tree** improve performances while simplying the policies.

The **tree programs allow to detect and correct misalignments**, to add safety conditions, and to merge skills trained in curriculum RL settings.

In our experimental evaluations, **VIPER did not improve (over DAGGER)** the performances of the extracted trees.




```
if nitrogen - days_planting < -17.50:
   if nitrogen - grain_weight < 13.50:
      if nitrogen - days_planting < -39.50:
         if nitrogen - maize_growing < -5.00:
            return "fertilizer_quant": 0.0
         else:

            return "fertilizer_quant": 27.0
      else:
         return "fertilizer_quant": 0.0
   else:
      if nitrogen - biomass < -930.64:
         return "fertilizer_quant": 54.0
      else:
         return "fertilizer_quant": 35.0
```

INTERPRETER's program allow for simple **additions** (Seaquest), **corrections** (Pong), and **merging** (Tennis) of their python policies. These programs can also be **extracted from human heuristic policies**, as shown on this fertilizer-aided plant harvesting problem.

[1] Luo et al. "INSIGHT: End-to-End Neuro-Symbolic Visual RL with Language Explanations." (2024).
[2] Delfosse et al. "Interpretable concept bottlenecks to align reinforcement learning agents." (2024).
[3] Delfosse et al. "Interp. and explai. logical policies via neurally guided symbolic abstraction." (2024).

Hector Kohler

Quentin Delfosse

1 Inria Université de Lille

2 AIML Lab TU Darmstadt

3 ATHENE

4 hessian.AI

5 dfki ai German Research Center for AI