Interpretability, Decision Trees, and Sequential Decision Making

Hector Kohler

Supervised by Dr. Riad Akrour (HdR) and Prof. Philippe Preux (HdR)
Université de Lille, CNRS, Inria, UMR CRIStAL 9189, France

November 22, 2025

Sequential decision making (SDM)

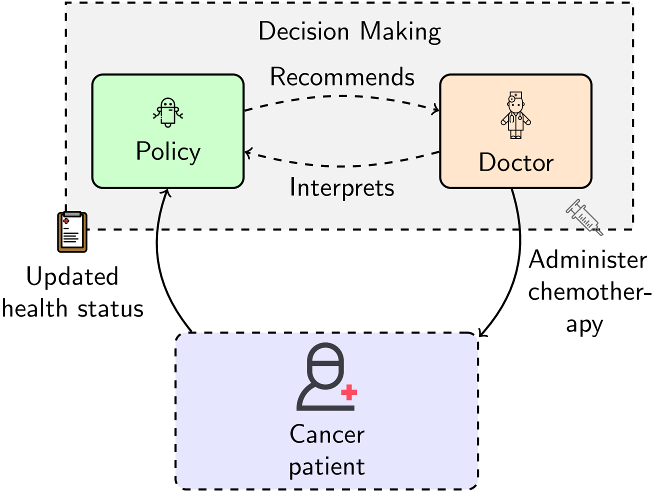


Figure: Sequential decision making in cancer treatment.

Machine learning (ML) of policies for SDM

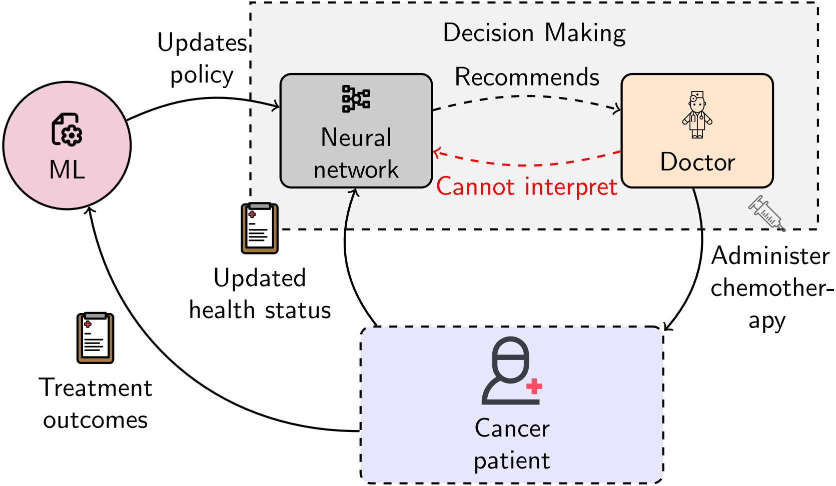


Figure: Machine learning of neural networks has many recent successes but neural networks are black-box.

Research direction

How to learn interpretable policies for sequential decision making?

Decision trees

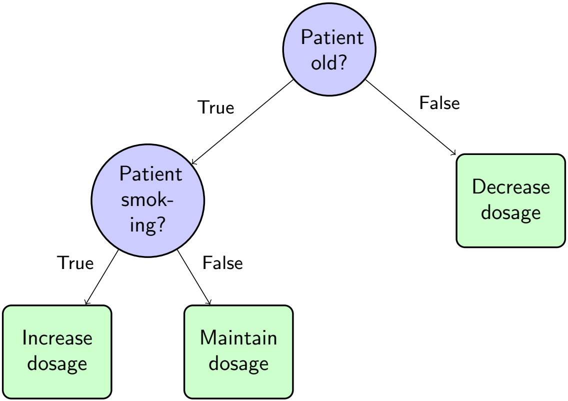


Figure: A generic decision tree of depth $D = 2$.

Algorithms are well understood for non-sequential learning: CART, OCT [7, 6, 9, 39, 25] .... What about SDM?

Markov decision processes

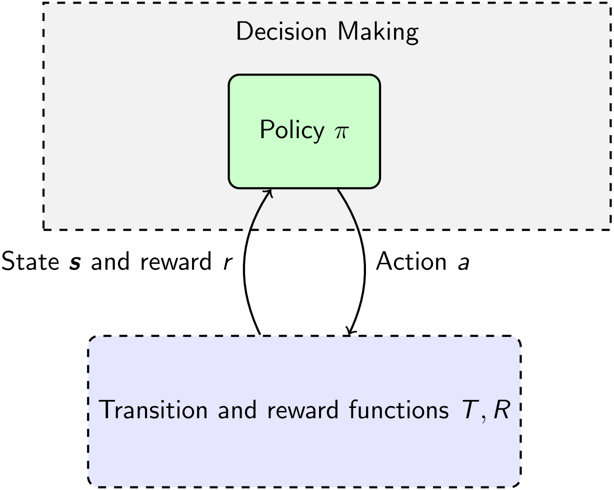


Figure: Markov decision process ([29]).

Reinforcement learning (RL) objective

- The goal of RL ([37]) for SDM is to find a policy, $\pi : S \rightarrow A$ that maximizes:

$$J(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 \sim T_0, a_t = \pi(s_t), s_{t+1} \sim T(s_t, a_t)\right] \tag{1}$$

- Value iteration, Q-learning, Sarsa, Deep Q Networks, Proximal Policy Optimization, ... ([5, 37, 27, 34])

Grid world MDP

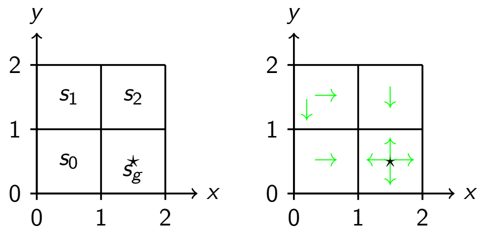


Figure: A grid world MDP and optimal actions.

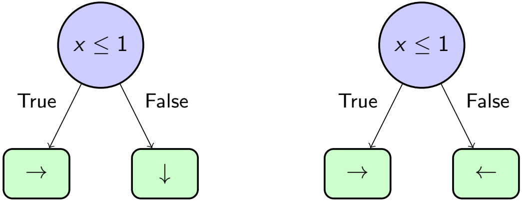


Figure: An optimal depth-1 decision tree policy and a sub-optimal depth-1 decision tree policy.

Indirect approach: imitation learning

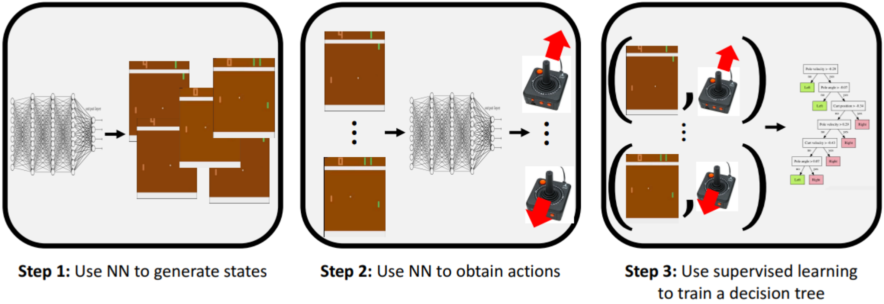


Figure: Imitation learning works well in practice to get interpretable policies ([26, 4, 33]) but no optimality guarantees.

Example: a decision tree policy for the grid world MDP

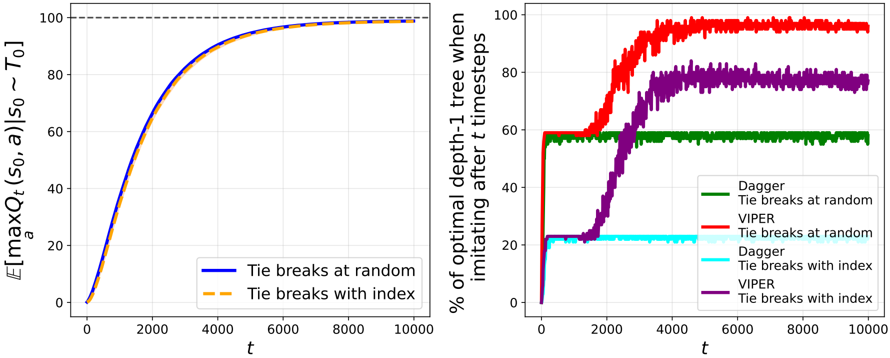


Figure: Sample complexity curve of Q-learning over 100 random seeds and performance of indirect interpretable methods when imitating the greedy policy with a tree at different Q-learning stages.

Direct approach to interpretable SDM

Q: Can we use reinforcement learning to directly optimize trade-offs of performance and interpretability in SDM?
A: direct reinforcement leargning is hard because it involves partial observability.

Iterative bounding Markov decision processes (IBMDP)

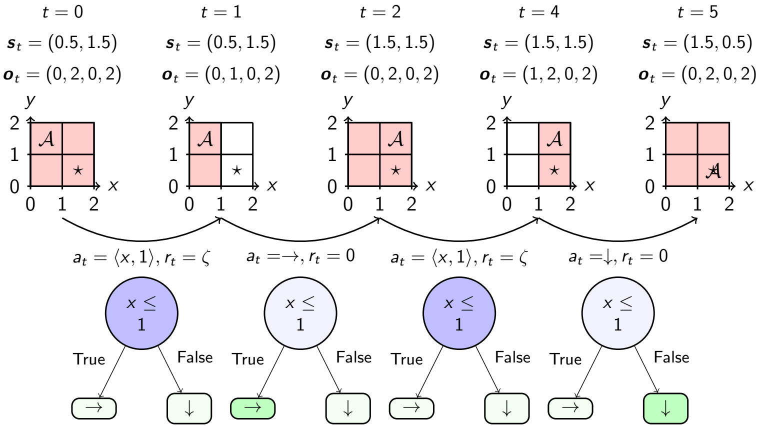


Figure: Trajectory in an IBMDP of the grid world MDP ([38]). Actions build a decision tree policy and rewards control the interpretability-performance trade-off.

Pros and cons of IBMDPs

Pros

- No need to design new algorithm: we can use deep RL.
- IBMDP rewards trade-off naturally interpretability and performances.

Cons

- Only **determinstic** and **partially observable** (a.k.a. memoryless or reactive) policies are equivalent to decision tree policies.
- Finding the best **deterministic** and **partially observable** policy is NP-hard ([21])!

Re-formulation

Q: Can we use reinforcement learning to directly optimize trade-offs of performance and interpretability in SDM? $\Leftrightarrow$
Q: How does RL perform for optimizing **deterministic** *and* **partially observable** *policies in IBMDPs?*

Result: RL cannot retrieve optimal depth-1 trees for the grid world MDP

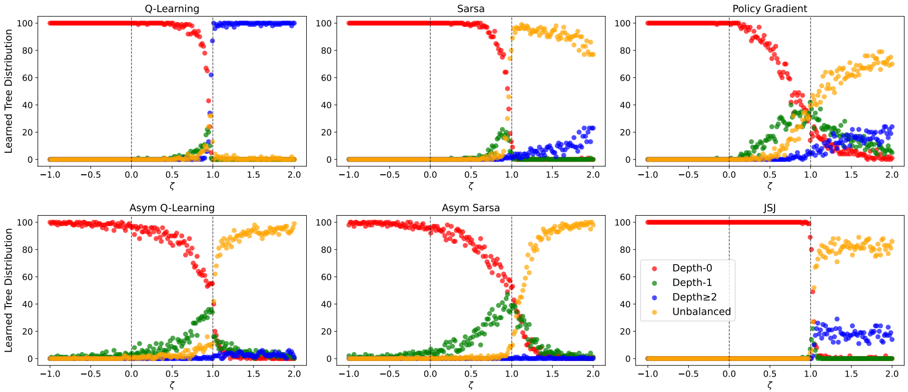


Figure: Distributions of final tree policies learned with various (asymmetric) RL algorithms ([37, 35, 22, 1, 2]) across 100 seeds. For each different performance-interpretability trade-off value ζ, each point represent the share of different trees.

Result: for similar problems, RL struggles when there is partial observability (not surprising)

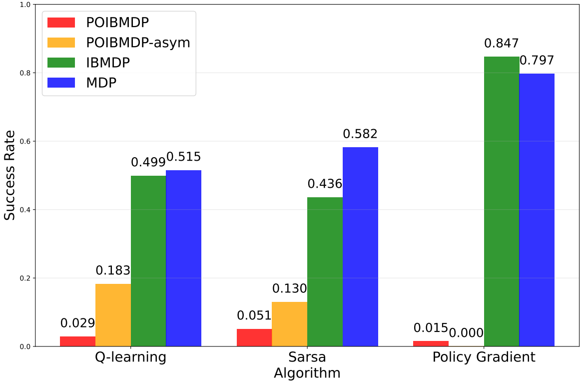


Figure: Success rates of different (asymmetric) RL algorithms over thousands of runs when applied to learning either deterministic partially observable policies in an IBMDP deterministic Markovian policies in the same IBMDP.

Interesting sub-class of MDPs: classification MDPs

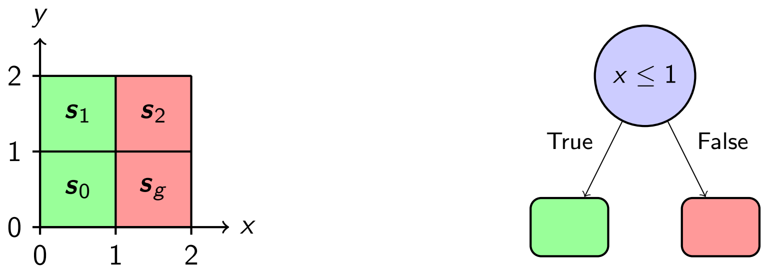


Figure: In this classification MDP, there are four data to which to assign either a green or red label. On the right, there is the unique optimal depth-1 tree for this particular classification MDP.

We show that in theory, deterministic partially observable policies for classification IBMDPs ($\Leftrightarrow$ decision tree policies) are in fact Markovian.

Perspectives for direct RL of decision tree policies.

- Interpretability for SDM problems is difficult because for most problems there is **partial observability**.
- Should we focus on indirect approach? Hybrid approaches ([32])?
- Fixing the policy tree structure a priori (paramteric trees, [24])?
- Design algorithms that learn deterministic partially observable policies ([16, 17])?

RL works in classification MDPs

Q: Can we leverage SDM in classification MDPs to design new decision tree induction algorithms for the supervised learning (no sequentiality) setting? **A: Yes!**

Decision trees in supervised learning

- N data points. Each x_i is described by p features and has a label $y_i \in \mathcal{Y}$.

$$\mathcal{L}(T) = \frac{1}{N}\sum_{i=1}^{N}\ell(y_i, f(x_i)) + \alpha C(T) \tag{2}$$

- Tree-based models are **interpretable** and perform really well on **tabular** data, often **better than deep neural nets** ([13]).

Optimal decision tree induction is NP-hard

- Greedy algorithms ([7, 31, 30]) **sub-optimal accuracy**, but time complexity in $\boldsymbol{O(2^D)}$.
- Optimal algorithms ([6, 9, 19, 8] ...) **optimal accuracy**, but time complexity in $\boldsymbol{O((2Np)^D)}$.

In between?

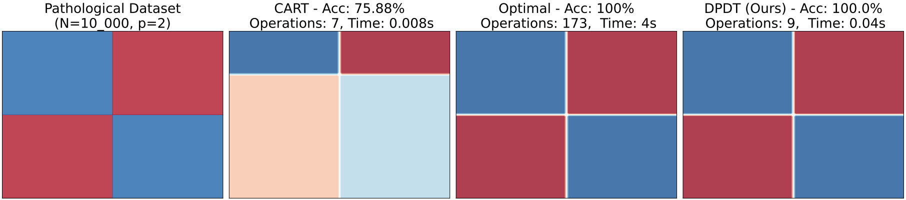


Figure: A checkers board data set highlights the limitations of existing works.

Decision tree induction as solving MDPs

Intuition

The induction of a decision tree is made of a sequence of decisions: at each node, we must decide whether it is better to split (a subset of) $\mathcal{E}$, or to create a leaf node.

- S: data subsets.
- A: test or leaf nodes that can be added to the tree.
- R: penalty or accuracies.
- T: node traversals.

Decision tree induction as solving MDPs

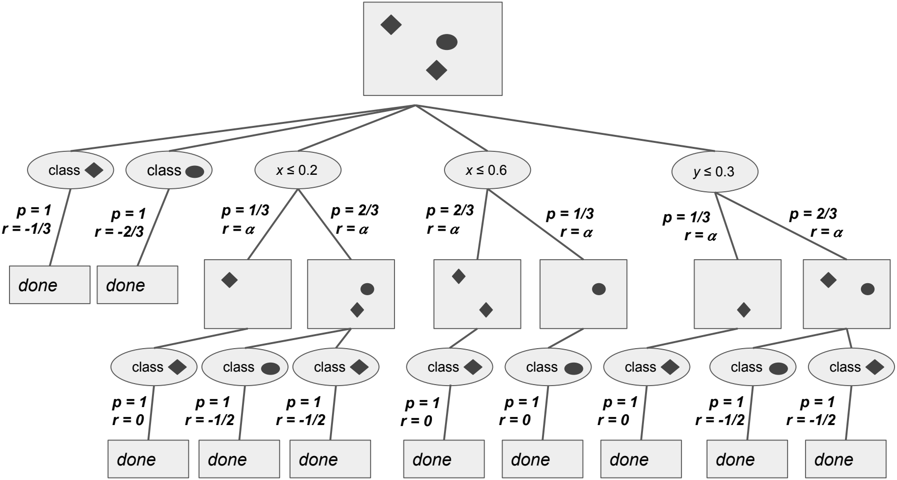


Figure: MDP formulation of a generic decision tree induction for a supervised learning task.

Controlling the time complexity of decision tree induction

- Greedy algorithms consider only one candidate action in each state which is the test that minimizes some impurity criterion $\rightarrow$ **MDP state space size is $O(2^D)$**.
- Optimal algorithms consider all possible actions in each state $\rightarrow$ **MDP state space size is $O((2Np)^D)$**.
- Let's choose candidate actions adaptively $\rightarrow$ for each MDP state consider B actions: **state space size is $O((2B)^D)$**.

Dynamic Programming Decision Trees (DPDT)[1]

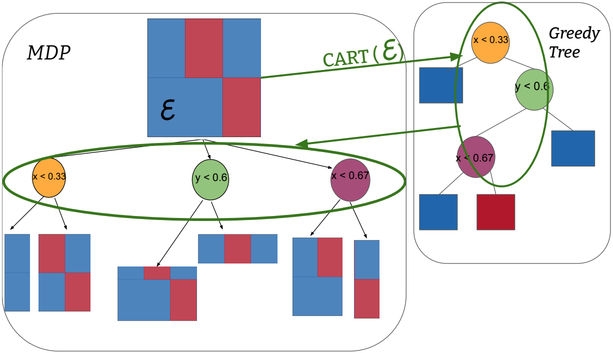


Figure: Overview of our algorithm DPDT presented at the 31st ACM SIGKDD conference.

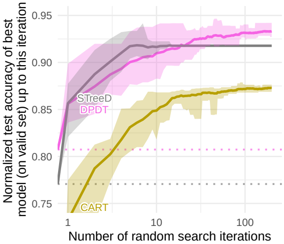
[1]Because states are entire datasets, we implement DPDT with a depth-first search to limit the space complexity.

Comparing tree accuracy to complexity

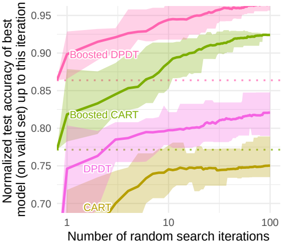
Table: Train accuracy and operation count when learning depth-3 decision trees.

			Accuracy				**Operations**			
Dataset	N	p	Opt Quant-BnB	Greedy CART	DPDT light	DPDT full	Opt Quant-BnB	Greedy CART	DPDT light	DPDT full
room	8103	16	**0.992**	0.968	0.991	**0.992**	10^6	15	286	16100
bean	10888	16	**0.871**	0.777	0.812	0.853	$5 \cdot 10^6$	15	295	25900
eeg	11984	14	**0.708**	0.666	0.689	0.706	$2 \cdot 10^6$	13	289	26000
avila	10430	10	**0.585**	0.532	0.574	**0.585**	$3 \cdot 10^7$	9	268	24700
magic	15216	10	**0.831**	0.801	0.822	0.828	$6 \cdot 10^6$	15	298	28000
htru	14318	8	**0.981**	0.979	0.979	0.980	$6 \cdot 10^7$	15	295	25300
occup.	8143	5	**0.994**	0.989	0.991	**0.994**	$7 \cdot 10^5$	13	280	16300
skin	196045	3	**0.969**	0.966	0.966	0.966	$7 \cdot 10^4$	15	301	23300
fault	1552	27	**0.682**	0.553	0.672	0.674	$9 \cdot 10^8$	13	295	24200
segment	1848	18	**0.887**	0.574	0.812	0.879	$2 \cdot 10^6$	7	220	16300
page	4378	10	**0.971**	0.964	0.970	0.970	10^7	15	298	22400
bidding	5056	9	**0.993**	0.981	0.985	**0.993**	$3 \cdot 10^5$	13	256	9360
raisin	720	7	**0.894**	0.869	0.879	0.886	$4 \cdot 10^6$	15	295	20900
rice	3048	7	**0.938**	0.933	0.934	0.937	$2 \cdot 10^7$	15	298	25500
wilt	4339	5	**0.996**	0.993	0.994	0.995	$3 \cdot 10^5$	13	274	11300
bank	1097	4	**0.983**	0.933	0.971	0.980	$6 \cdot 10^4$	13	271	7990

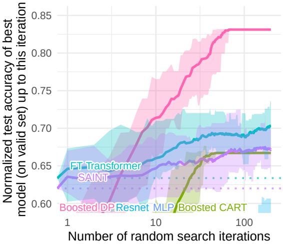
DPDT trees generalization



(a) DPDT depth-5 trees vs. other detph-5 trees



(b) Boosted DPDT vs. Boosted CART



(c) Boosted DPDT vs. other classifiers

Perspectives

- New SOTA decision tree induction with dynamic programming in MDPs.
- What about using DPDT for indirect decision tree policy learning for SDM?
- What performances could we reach with an industry-grade implementation of XGboost+DPDT?

Let us take a step back

Q: Are decision trees really the most interpretable model?

A: It depends.

Policy interpretability

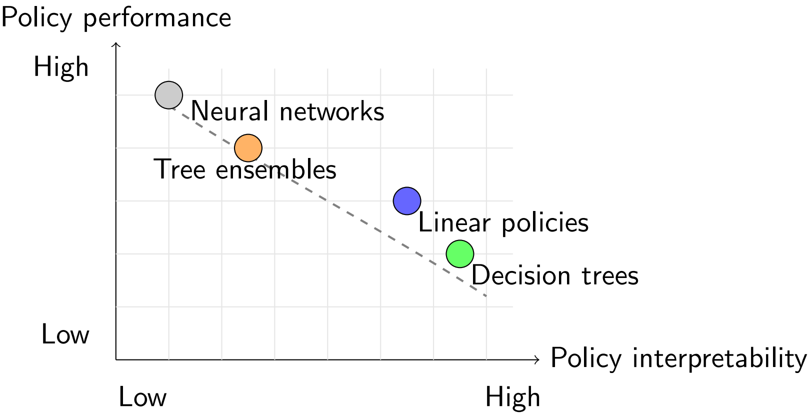


Figure: **Heuristic** interpretability-performance trade-offs of different policy classes. Interpretability is often presented in opposition to performances.

How to measure policy interpretability?

Challenges ([12, 20, 10])

- There is no clear definition of interpretability.
- Measuring interpretability might require humans.

The notion of *simulatability* ([20])

- Interpretability $\simeq$ how long it takes for human to make the same computations given an input.
- Interpretability $\simeq$ how much effort it would take a human to read through the entire policy once.
- Inside a given policy class, less parameters should mean more interpretability ([11, 18, 15, 36, 14]).
- The time required to formally verify a policy should decrease with interpretability ([4, 3]).

A methodology to measure policy interpretability without humans

Simulatability ([20])

1. How long it takes for human to make the same computations given an input $\simeq$ policy inference time.
2. How much effort it would take a human to read through the entire policy once $\simeq$ policy size in memory.

Not that simple in practice ([23])

- Different hardwares (tree policies are run on CPUs while neural policies are run on GPUs).
- Different implementations (neural policies compute outputs using matrix operations while tree operate fully sequentially) ...

We propose policy unfolding

```python
# Decision tree for Mountain Car
def play(x):
    if x[1] <= -0.2597:
        if x[1] <= -0.6378:
            return 0
        else:
            if x[0] <= -1.0021:
                return 2
            else:
                return 0
    else:
        if x[1] <= -0.0508:
            if x[0] <= 0.2979:
                if x[0] <= 0.0453:
                    return 2
                else:
                    if x[1] <=
   -0.2156:
                        return 0
                    else:
                        return 2
            else:
                return 0
        else:
            return 2
```

```python
# Small ReLU MLP for Pendulum
def play(x):
    h_layer_0_0 = 1.238*x[0]+0.971*x
      [1]
                  +0.430*x[2]+0.933
    h_layer_0_0 = max(0, h_layer_0_0
      )
    h_layer_0_1 = -1.221*x[0]+1.001
                  *x[1]-0.423*x[2]
                  +0.475
    h_layer_0_1 = max(0, h_layer_0_1
      )
    h_layer_1_0 = -0.109*h_layer_0_0
                  -0.377*h_layer_0_1
                  +1.694
    h_layer_1_0 = max(0, h_layer_1_0
      )
    h_layer_1_1 = -3.024*h_layer_0_0
                  -1.421*h_layer_0_1
                  +1.530
    h_layer_1_1 = max(0, h_layer_1_1
      )

    h_layer_2_0 = -1.790*h_layer_1_0
                  +2.840*h_layer_1_1
                  +0.658
    y_0 = h_layer_2_0
    return [y_0]
```

Empirical validation

1. Does our methodology respect consensus on policy interpretability?
2. Is policy unfolding necessary to respect the consensus?
3. What kind of results we can obtain using our proposed methodology?

We imitate $\sim$ 40000 expert policies from `stable-baselines3` using various policy classes/nb parameters on various environments.

Result: unfolding policies is necessary to respect consensus

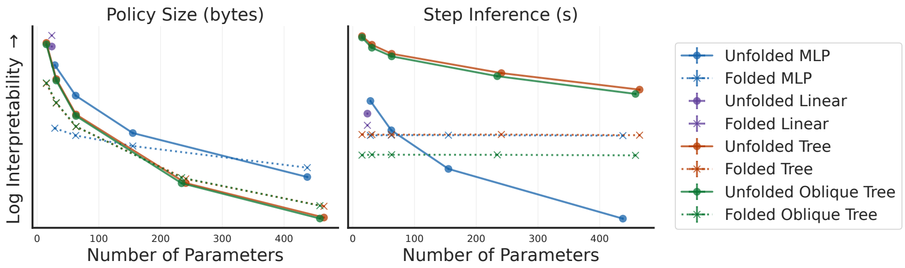


Figure: Policies interpretability on classic control environments. We plot 95% stratified bootstrapped confidence intervals around means in both axes. In each sub-plot, interpretability is measured with either bytes or inference speed.

Result: there is no dominating policy class for all environments

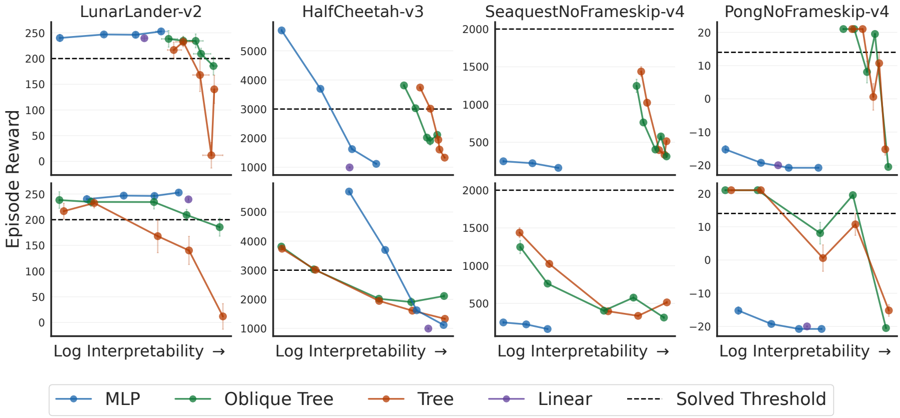


Figure: Interpretability-Performance trade-offs for representative environments. Top row, interpretability is measured with step inference times. Bottom row, the interpretability is measured with policy size.

Perspectives

- Because there is no dominating class for all problems in terms of interpretability-performance trade-offs, beliefs such as "trees are more interpretable than neural networks" should be used with caution.
- Tree-like policy classes can have good inductive bias for game-like envrionments.
- Can a human study confirm our results?
- Can our methodology be used for evaluating the interpretability of (very) big models?
- Can we use our policy programs as low level skills (hierarchical RL)?

Conclusion: interpretable machine learning is a difficult research topic

- Technical challenges: **partial observability** in SDM, NP-hardness.
  $\rightarrow$ Focus on indirect approaches and/or on POMDP research first.
- Fundamental challenges: **no definition**.
  $\rightarrow$ Discuss with the community (InterpPol workshop).
- **Decision trees offer good inductive bias for SDM in games or tabular data**.

My hope

Motivate interpretability by finding a real-world problem where interpretability is *really* necessary ([28]).

[1] Andrea Baisero and Christopher Amato. "Unbiased Asymmetric Reinforcement Learning under Partial Observability". In: *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. AAMAS '22. Virtual Event, New Zealand: International Foundation for Autonomous Agents and Multiagent Systems, 2022, pp. 44–52. ISBN: 9781450392136.

[2] Andrea Baisero, Brett Daley, and Christopher Amato. "Asymmetric DQN for partially observable reinforcement learning". In: *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*. Ed. by James Cussens and Kun Zhang. Vol. 180. Proceedings of Machine Learning Research. PMLR, Jan. 2022, pp. 107–117. URL: https://proceedings.mlr.press/v180/baisero22a.html.

[3] Pablo Barceló et al. "Model interpretability through the lens of computational complexity". In: *Advances in neural information processing systems* (2020).

[4] Osbert Bastani, Yewen Pu, and Armando Solar-Lezama. "Verifiable Reinforcement Learning via Policy Extraction". In: (2018).

[5] Richard Bellman. *Dynamic Programming*. 1957.

[6] Dimitris Bertsimas and Jack Dunn. "Optimal classification trees". In: *Machine Learning* 106 (2017), pp. 1039–1082.

[7] L Breiman et al. *Classification and Regression Trees*. Wadsworth, 1984.

[8] Ayman Chaouki, Jesse Read, and Albert Bifet. "Branches: A Fast Dynamic Programming and Branch & Bound algorithm for Optimal Decision Trees". In: (2024). arXiv: 2406.02175 [cs.LG]. URL: https://arxiv.org/abs/2406.02175.

[9] Emir Demirovic et al. "MurTree: Optimal Decision Trees via Dynamic Programming and Search". In: *Journal of Machine Learning Research* 23.26 (2022), pp. 1–47. URL: http://jmlr.org/papers/v23/20-520.html.

[10] Finale Doshi-Velez and Been Kim. "Towards A Rigorous Science of Interpretable Machine Learning". In: (2017). arXiv: 1702.08608 [stat.ML]. URL: https://arxiv.org/abs/1702.08608.

[11] Alex A. Freitas. “Comprehensible classification models: a position paper”. In: *SIGKDD Explor. Newsl.* 15.1 (Mar. 2014), pp. 1–10. ISSN: 1931-0145. DOI: 10.1145/2594473.2594475. URL: https://doi.org/10.1145/2594473.2594475.

[12] Claire Glanois et al. “A survey on interpretable reinforcement learning”. In: *Machine Learning* (2024), pp. 1–44.

[13] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. “Why do tree-based models still outperform deep learning on typical tabular data?” In: *Advances in neural information processing systems* 35 (2022), pp. 507–520.

[14] Johan Huysmans et al. “An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models”. In: *Decis. Support Syst.* 51.1 (Apr. 2011), pp. 141–154. ISSN: 0167-9236. DOI: 10.1016/j.dss.2010.12.003. URL: https://doi.org/10.1016/j.dss.2010.12.003.

[15] Isaac Lage et al. *An Evaluation of the Human-Interpretability of Explanation*. 2019. arXiv: 1902.00006 [cs.LG]. URL: https://arxiv.org/abs/1902.00006.

[16] Gaspard Lambrechts, Adrien Bolland, and Damien Ernst. "Informed POMDP: Leveraging Additional Information in Model-Based RL". In: *Reinforcement Learning Journal* 2 (2025), pp. 763–784.

[17] Gaspard Lambrechts, Damien Ernst, and Aditya Mahajan. "A Theoretical Justification for Asymmetric Actor-Critic algorithms". In: *Forty-second International Conference on Machine Learning*. 2025. URL: https://openreview.net/forum?id=F1yANMCnAn.

[18] Nada Lavrač. "Selected techniques for data mining in medicine". In: *Artificial Intelligence in Medicine* 16.1 (1999). Data Mining Techniques and Applications in Medicine, pp. 3–23. ISSN: 0933-3657. DOI: https://doi.org/10.1016/S0933-3657(98)00062-1. URL: https://www.sciencedirect.com/science/article/pii/S0933365798000621.

[19] Jacobus van der Linden, Mathijs de Weerdt, and Emir Demirović. "Necessary and Sufficient Conditions for Optimal Decision Trees using Dynamic Programming". In: *Advances in Neural Information Processing Systems* 36 (2023). Ed. by A. Oh et al., pp. 9173–9212.

[20] Zachary C. Lipton. "The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery.". In: *Queue* 16.3 (2018), pp. 31–57.

[21] Michael L. Littman. "Memoryless policies: theoretical limitations and practical results". In: *Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats 3: From Animals to Animats 3*. SAB94. Brighton, United Kingdom: MIT Press, 1994, pp. 238–245. ISBN: 0262531224.

[22] John Loch and Satinder P. Singh. "Using Eligibility Traces to Find the Best Memoryless Policy in Partially Observable Markov Decision Processes". In: *Proceedings of the Fifteenth International Conference on Machine Learning*. ICML '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 323–331. ISBN: 1558605568.

[23] Lirui Luo et al. "End-to-End Neuro-Symbolic Reinforcement Learning with Textual Explanations". In: *International Conference on Machine Learning (ICML)* (2024).

[24] Sascha Marton et al. "Mitigating Information Loss in Tree-Based Reinforcement Learning via Direct Optimization". In: (2025). URL: https://openreview.net/forum?id=qpXctF2aLZ.

[25] Rahul Mazumder, Xiang Meng, and Haoyue Wang. "Quant-BnB: A Scalable Branch-and-Bound Method for Optimal Decision Trees with Continuous Features". In: *Proceedings of the 39th International Conference on Machine Learning*. Proceedings of Machine Learning Research 162 (17–23 Jul 2022). Ed. by Kamalika Chaudhuri et al., pp. 15255–15277. URL: https://proceedings.mlr.press/v162/mazumder22a.html.

[26] Stephanie Milani et al. "Explainable Reinforcement Learning: A Survey and Comparative Review". In: *ACM Comput. Surv.* 56.7 (Apr. 2024). ISSN: 0360-0300. DOI: 10.1145/3616864. URL: https://doi.org/10.1145/3616864.

[27] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *nature* 518.7540 (2015), pp. 529–533.

[28] Myura Nagendran et al. "Eye tracking insights into physician behaviour with safe and unsafe explainable AI recommendations". In: *NPJ Digital Medicine* 7.1 (2024), p. 202.

[29] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 1994.

[30] J Ross Quinlan. "C4. 5: Programs for machine learning". In: *Morgan Kaufmann google schola* 2 (1993), pp. 203–228.

[31] J. R. Quinlan. "Induction of Decision Trees". In: *Mach. Learn.* 1.1 (1986), pp. 81–106.

[32] "Regional Tree Regularization for Interpretability in Deep Neural Networks". In: 34 (Apr. 2020), pp. 6413–6421. DOI: 10.1609/aaai.v34i04.6112. URL: https://ojs.aaai.org/index.php/AAAI/article/view/6112.

[33] Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. "A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning". In: (2010).

[34] John Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[35] Satinder P. Singh, Tommi S. Jaakkola, and Michael I. Jordan. "Learning without state-estimation in partially observable Markovian decision processes". In: *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*. ICML'94. New Brunswick, NJ, USA: Morgan Kaufmann Publishers Inc., 1994, pp. 284–292. ISBN: 1558603352.

[36] Dylan Slack et al. *Assessing the Local Interpretability of Machine Learning Models*. 2019. arXiv: 1902.03501 [cs.LG]. URL: https://arxiv.org/abs/1902.03501.

[37] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA: The MIT Press, 1998.

[38] Nicholay Topin et al. "Iterative bounding mdps: Learning interpretable policies via non-interpretable methods". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35 (2021), pp. 9923–9931.

[39] Sicco Verwer and Yingqian Zhang. "Learning optimal classification trees using a binary linear program formulation". In: *Proceedings of the AAAI conference on artificial intelligence* 33 (2019), pp. 1625–1632.