Explainable Reinforcement Learning through Behavioral Cloning

Nikolaos Kaparinos[†] and Tzoulio Chamiti[†]

Contributing authors: kaparinos@csd.auth.gr; t.chamiti@csd.auth.gr;
[†]These authors contributed equally to this work.

Abstract

Reinforcement learning (RL) has emerged as a powerful paradigm for training agents to make autonomous decisions in complex environments. However, the lack of explainability in RL agents has been a significant hurdle in many real-world applications. In this study, we present an approach to address this challenge by combining Deep Reinforcement Learning (DRL) with Behavior Cloning (BC) to produce an agent that is both performant and explainable simultaneously. Initially, a conventional DRL agent is trained to optimize its policy through interaction with the environment. This process allows the agent to learn effective strategies, but the resulting policy may lack interpretability. To enhance explainability, we leverage the knowledge acquired by the non-explainable DRL agent and employ it to train an interpretable behavior cloning agent.Experimental results demonstrate that the proposed approach successfully achieves both high performance and explainability. The BC agent exhibits proficient behavior by inheriting the effective strategies learned by the DRL agent, while the interpretable classifier ensures the ability to provide human-understandable explanations for its actions. This combined approach presents a novel contribution towards addressing the trade-off between performance and explainability in RL, making it suitable for applications where trust, accountability and interpretability are crucial.

Keywords: Reinforcement Learning, Explainability, Behavior Cloning, Machine Learning

1 Introduction

Reinforcement learning (RL) has witnessed remarkable advancements in recent years, enabling agents to learn and make decisions in complex environments without explicit programming. RL algorithms, such as deep reinforcement learning (DRL), have achieved remarkable successes in various domains, including game playing [1], robotics [2] and recommendation systems [3]. However, one significant challenge in deploying RL agents in real-world applications lies in their lack of explainability, which hinders their adoption in critical domains where interpretability and transparency are essential.

Explainability refers to the ability to understand and provide comprehensible justifications for an agent's decisions and actions. While DRL agents excel at learning complex behaviors through trial-and-error interactions with the environment, they often operate as black boxes, making it challenging to understand why they take specific actions or behave in a particular manner. Extensive lack of transparency could lead to concerns regarding the reliability, trustworthiness and accountability of RL agents in critical applications, such as healthcare, finance and autonomous vehicles.

To address this challenge, a novel approach that combines DRL with behavior cloning (BC) to create agents that exhibit both high performance and explainability simultaneously is proposed. Behavior cloning is a supervised learning technique that involves training a model to imitate expert behavior by learning from their demonstrations. By leveraging the knowledge acquired by a pre-trained DRL agent, the aim of this work is to produce an agent that inherits the effective strategies while ensuring interpretability through an explainable classifier.

Our objective is to demonstrate that the proposed approach can strike a balance between performance and explainability, overcoming the traditional trade-off between these two critical aspects in RL. By providing interpretable justifications for their actions, our agents can address the growing need for transparency and accountability in RL applications. Furthermore, the ability to understand an agent's decision-making process could enhance trust and facilitate collaboration between humans and RL agents in various domains.

This paper, presents the methodology and experimental results that showcase the effectiveness of our proposed approach. The performance and explainability of our agents is evaluated by using standard RL and control benchmarks, highlighting the advantages of the proposed hybrid approach over traditional DRL methods.

2 Related Work

Explainable RL is essential for real-world applications where understanding an agent's decision-making process is crucial for trust, safety and deployment. This section presents a review of related works that focus on explainability in RL and behavior cloning techniques. Numerous studies have emphasized the importance of interpretability in RL and have proposed methods for explaining RL agents' behavior. Many of these works utilize Cascading Decision Trees (CDT) [4] and Soft Decision Trees (SDT) [5] to generate interpretable policies. CDT involve a sequential structure of decision trees, where the output of one tree feeds into the next. This cascading process aims to capture complex decision-making processes while maintaining transparency. On the other hand, SDT utilize probability distributions at internal nodes, enabling a soft and probabilistic interpretation of the RL agent's behavior. In contrast, this paper utilizes normal decision trees as a means of simplifying the interpretability process, making the explanations more accessible and understandable. Instead of the cascading or soft approach, the focus is on identifying the most important features that drive the RL agent's behavior. This enables the extraction of explicit rules and conditions that can be easily comprehended by humans, thereby enhancing the overall transparency and interpretability of the RL agent's decision-making process.

3 Methodology

The training of the DRL agent begins using the Proximal Policy Optimization (PPO) algorithm, a state-of-the-art RL method known for its stability and sample efficiency. The DRL agent is trained by interacting with the environment on the standard RL API Gymnasium[6]. Three popular discrete environments from the RL and control literature were chosen, namely CartPole[7], MountainCar[8] and Acrobot[9]. After the DRL agent is trained, its expertise is utilized in order to create a behavior cloning dataset. The dataset consisting of observation-action pairs is created by letting the agent interact with the environment for 10000 steps. For each step, the corresponding observation and the action chosen by the DRL agent are recorded.

Consequently, the BC agent is trained in a supervised manner using the behavior cloning dataset. Furthermore, decision trees and logistic regression models are employed as the classifiers for behavior cloning, as these models are considered "white box" and offer substantial interpretability. The decision trees varying depth is experimented upon, ranging from 1 to 4, in order to explore the trade-off between performance and explainability.

To assess the performance and explainability of the behavior cloning agents, evaluations are conducted on the RL environments. The BC agents are evaluated over 100 episodes in each environment. The performance metrics, such as the average episode reward, are compared with those of the original DRL agent. This evaluation allows the understanding of the impact BC has on performance while considering the interpretability of the BC agents at the same time. One of the advantages of utilizing decision trees and logistic regression models is the availability of feature importances. By analyzing these feature importances, insight into the decision-making process of the behavior cloning agents is achieved. The initial training of the DRL agent allows it to learn effective strategies in complex environments, while the BC agent, trained using the learned behavior of the DRL agent, provides a transparent and interpretable model for decision-making. This combination enables for both performance and explainability in RL agents.

The methodology presented above forms the basis for our experiments, where the performance and explainability of the proposed approach across multiple RL environments is evaluated, by comparing the BC agents to the original DRL agent. The results of these experiments provide insights into the effectiveness of our approach in addressing the trade-off between performance and explainability in RL agents.

4 Results

The results presented are obtained from our experiments evaluating the performance and explainability of the proposed approach, which combines DRL with BC to achieve simultaneous performance and explainability in RL agents. The performance of the BC agents is compared with that of the original DRL agent and afterwards, the interpretability of the BC agents is examined using decision trees and logistic regression models.

Specifically, for each environment, we present a barplot of the performance of each BC agent compared to the original DRL one. Additionally, we present the feature importance and a visualisation of the Decision Tree for the tree with max depth equals 3, since it proved to be the "sweet spot" between performance and interpretability.

4.1 CartPole

The environment describes a pole which is attached by an un-actuated joint to a cart, that moves along a frictionless track. The pendulum is placed upright on the cart and the goal is to balance the pole by applying forces in the left and right direction on the cart. The action and observation space of the environment are described in the following tables.

Table 1 The CartPole environment action space.

Num	Action
0	Push cart to the left
1	Push cart to the right

Table 2 The CartPole environment observation space.

Num	Observation	Min	Max
0	Cart Position	-4.8	4.8
1	Cart Velocity	-Inf	Inf
2	Pole Angle	∼-0.418 rad (-24°)	∼0.418 rad (24°)
3	Pole Angular Velocity	-Inf	Inf

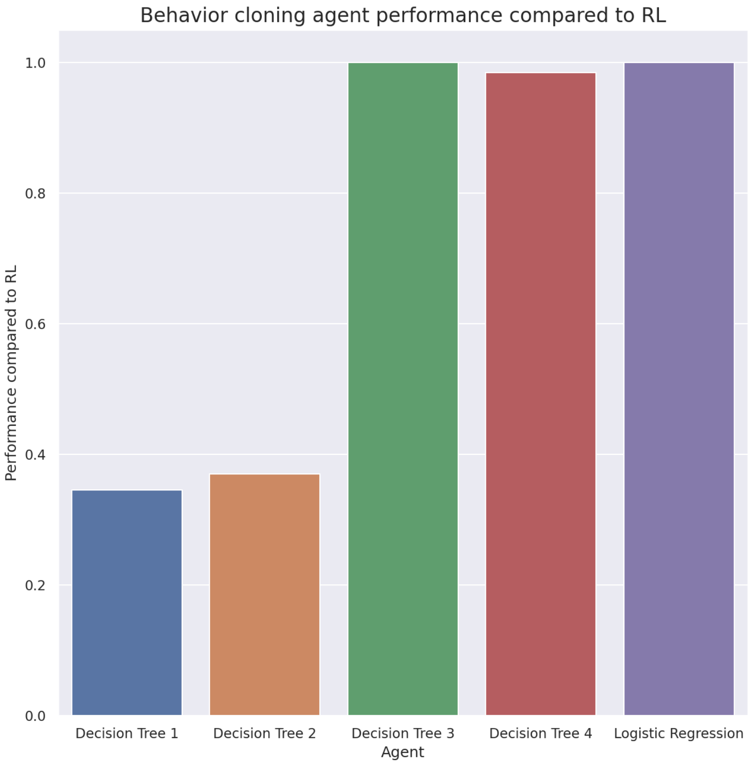

Fig. 1: BC models performance comparison in the CartPole environment.

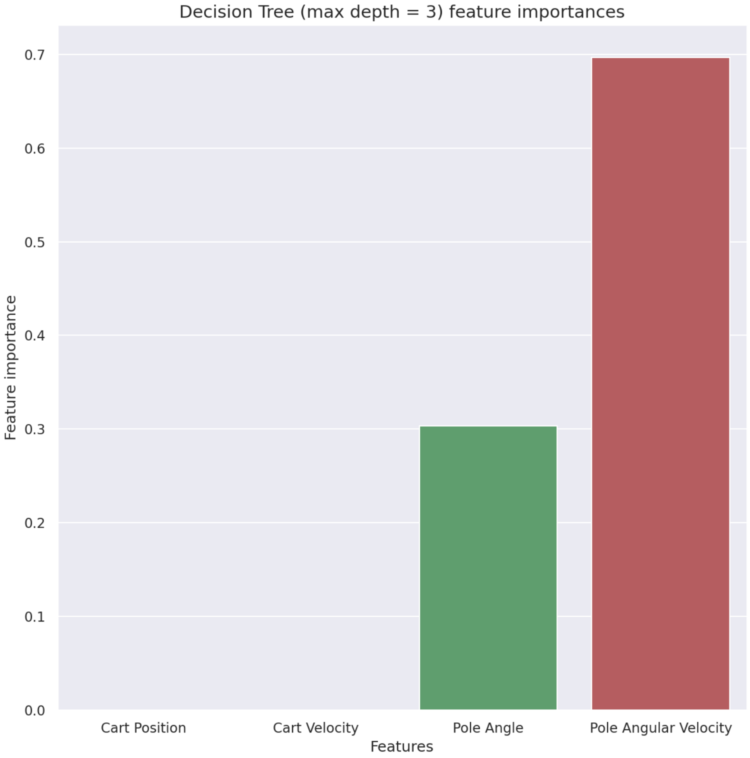

Fig. 2: Feature importance for Decision Tree with max depth = 3 in the CartPole environment.

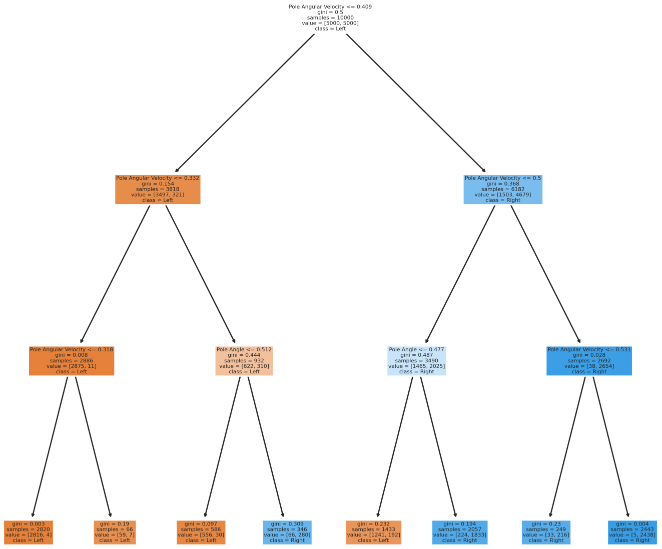

Fig. 3: Decision Tree with max depth = 3 visualization.

4.2 MountainCar

The environment consists of a car placed stochastically at the bottom of a sinusoidal valley, with the only possible actions being the accelerations that can be applied to the car in either direction. The goal is to strategically accelerate the car to reach the goal state on top of the right hill. The action and observation space of the environment are described in the following tables.

Table 3 The MountainCar environment action space.

Num	**Action**	**Value**	**Unit**
0	Accelerate to the left	Inf	position (m)
1	Don't accelerate	Inf	position (m)
2	Accelerate to the right	Inf	position (m)

Table 4 The MountainCar environment observation space.

Num	**Observation**	**Min**	**Max**	**Unit**
0	position of the car along the x-axis	-Inf	Inf	position (m)
1	velocity of the car	-Inf	Inf	position (m)

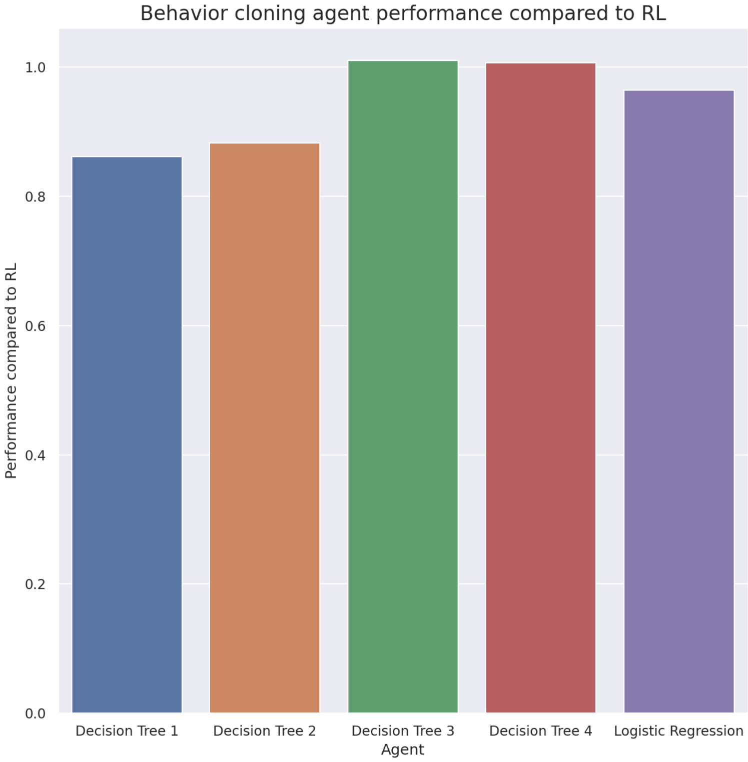

Fig. 4: BC models performance comparison in the MountainCar environment.

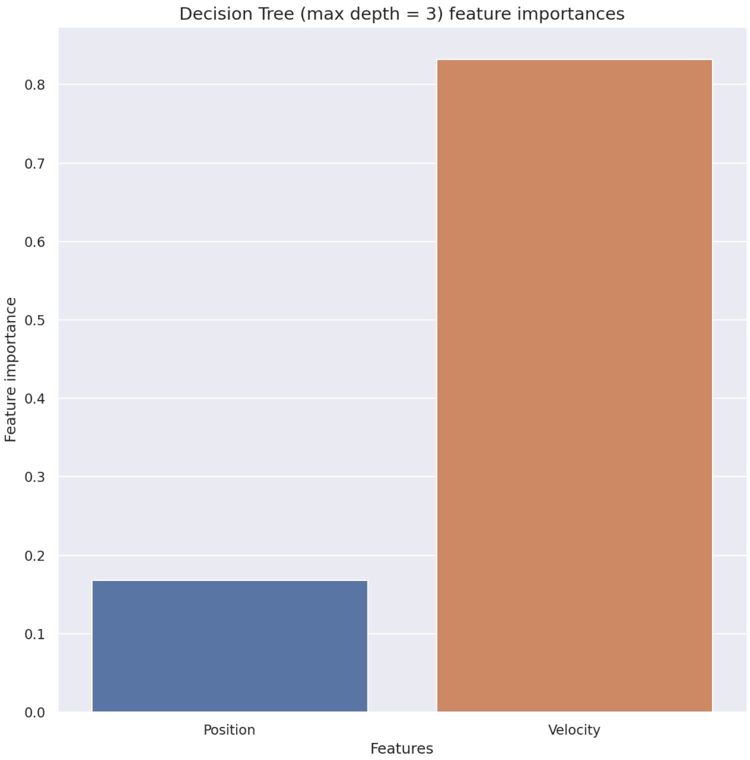

Fig. 5: Feature importance for Decision Tree with max depth = 3 in the MountainCar environment.

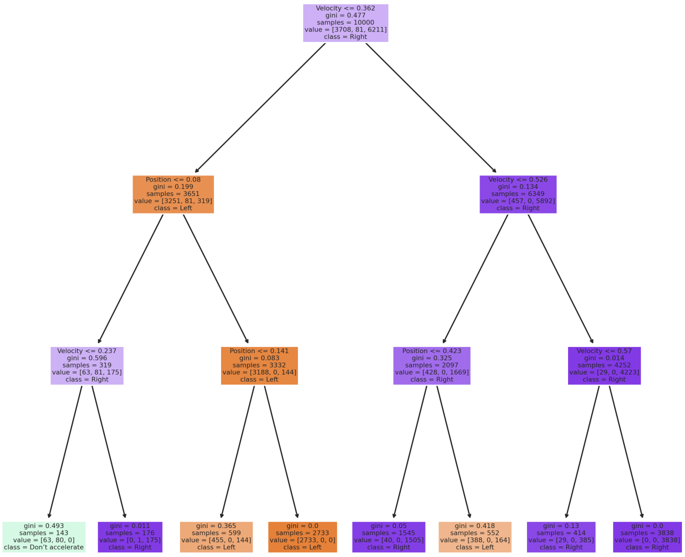

Fig. 6: Decision Tree with max depth = 3 visualization.

4.3 Acrobot

The environment consists of two links connected linearly to form a chain, with one end of the chain fixed. The joint between the two links is actuated. The goal is to apply torques on the actuated joint to swing the free end of the linear chain above a given height while starting from the initial state of hanging downwards. The action and observation space of the environment are described in the following tables.

Table 5 The Acrobot environment action space.

Num	Action	Unit
0	apply -1 torque to the actuated joint	torque (N m)
1	apply 0 torque to the actuated joint	torque (N m)
2	apply 1 torque to the actuated joint	torque (N m)

Table 6 The Acrobot environment observation space. **Theta1** is the angle of the first joint, where an angle of 0 indicates the first link is pointing directly downwards and **Theta2** is relative to the angle of the first link. An angle of 0 corresponds to having the same angle between the two links.

Num	Observation	Min	Max
0	Cosine of theta1	-1	1
1	Sine of theta1	-1	1
2	Cosine of theta2	-1	1
3	Sine of theta2	-1	1
4	Angular velocity of theta1	∼-12.567 (-4 * pi)	∼12.567 (4 * pi)
5	Angular velocity of theta2	∼-28.274 (-9 * pi)	∼28.274 (9 * pi)

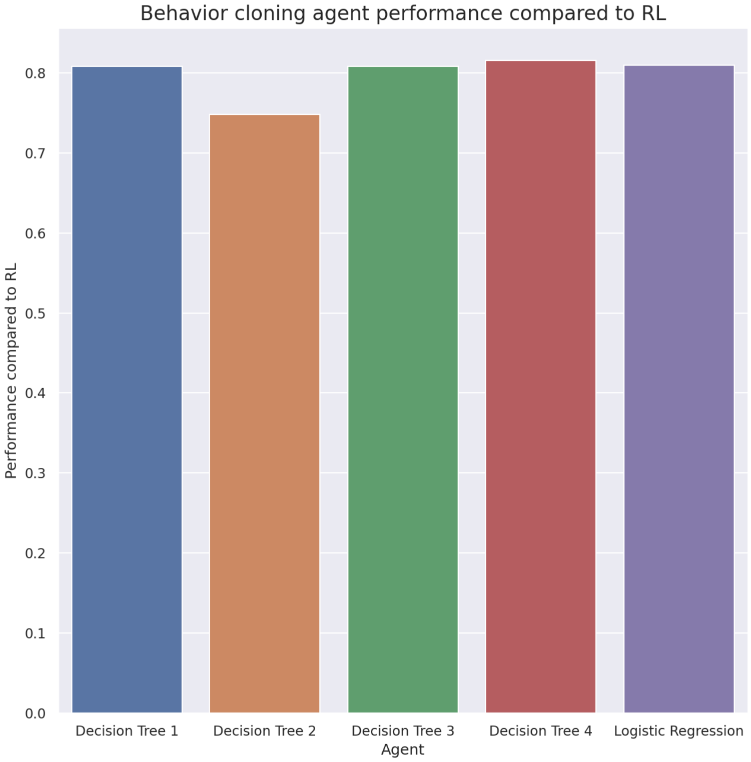

Fig. 7: BC models performance comparison in the Acrobot environment.

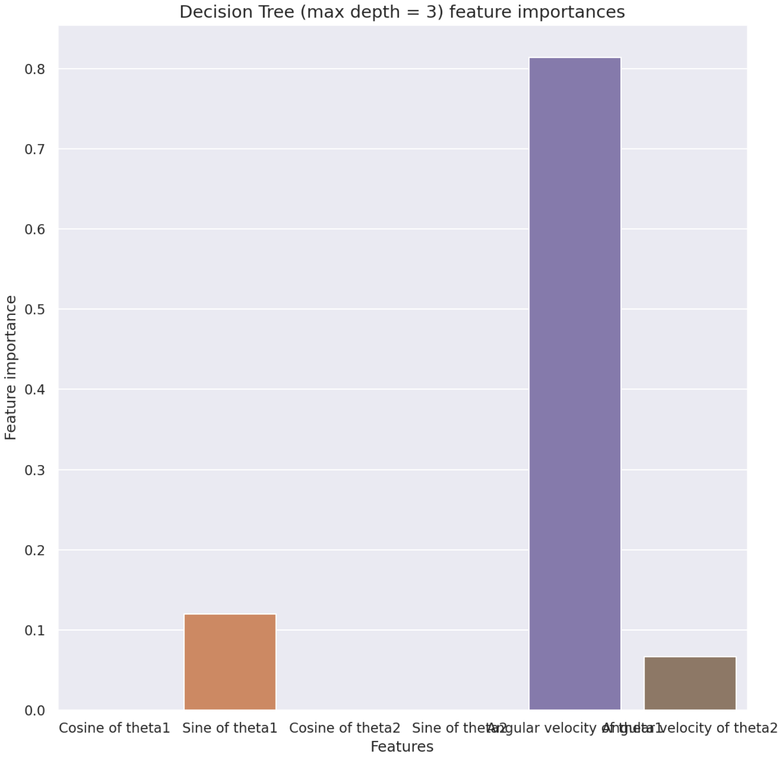

Fig. 8: Feature importance for Decision Tree with max depth = 3 in the Acrobot environment.

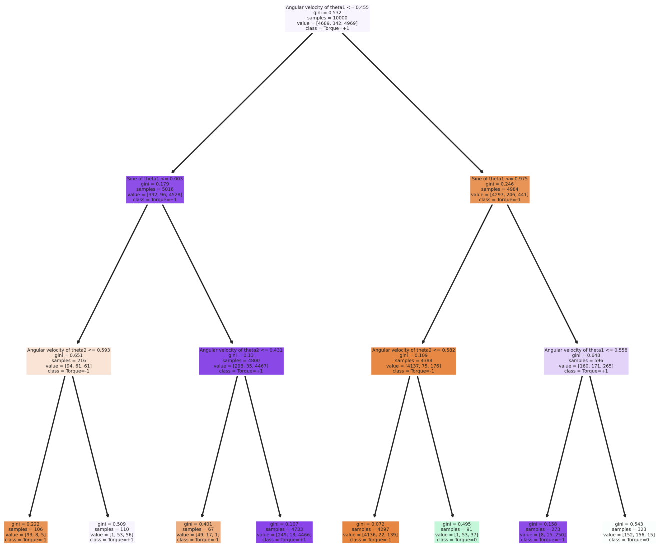

Fig. 9: Decision Tree with max depth = 3 visualization.

5 Conclusion

In this study, the challenge of achieving simultaneous performance and explainability in RL agents was addressed. By utilizing the data from DRL with BC in a form of transfer learning, a novel approach that combines the advantages of high-performance learning from DRL with the interpretability provided by ”white box” models was proposed. The results demonstrate the effectiveness of this approach in generating RL agents that exhibit both proficiency and transparency in decision-making. The evaluation of the behavior cloning agents on classic RL environments, including CartPole, MountainCar and Acrobot, demonstrated competitive performance compared to the original DRL agent. Although there may be a slight decrease in performance due to the behavior cloning process, the BC agents maintained proficient behavior, indicating the successful transfer of expertise from the DRL agent. Furthermore, the interpretability analysis revealed that decision trees and logistic regression models effectively captured the decision-making process of the BC agents. The feature importance's and tree visualizations provided by these models allowed for a comprehensive understanding of the factors influencing the agents' actions, enhancing their transparency and interpretability. The availability of understandable rules and feature importance ranking aims to encourage trust-building and collaboration between humans and RL agents.

The proposed approach contributes to addressing the challenges associated with the lack of transparency and interpretability in RL agents. The trade-off between performance and explainability in behavior cloning was also presented, highlighting that shallow decision trees which prioritized explainability do not always achieve satisfactory performance. On the other hand, decision trees with maximum depth equals 3 strike a reasonable balance between the two aspects, since they could still be considered sufficiently shallow and explainable. However, the optimal depth may vary depending on the specific RL environment and the desired levels of explainability in practical applications. In addition, by providing interpretable justifications for their actions, our agents facilitate their deployment in critical domains where trust, accountability and transparency are paramount. For future research, different models should be explored in an attempt to address possible generalization issues on the DRL dataset and to be able to tackle more complex environments. Such approaches could be rule-based models, distance or similarity based models and different DRL training procedures.

In conclusion, this study demonstrates the feasibility of creating RL agents that exhibit both performance and explainability by utilizing DRL with BC. The combination of these approaches opens new avenues for deploying RL agents in real-world applications, in an attempt at enabling transparency, interpretability and collaboration between humans and RL systems. By bridging the gap between performance and explainability, the study aims for the adoption of RL agents in critical domains that demand both proficiency and transparency.

References

[1] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., *et al.*: Mastering the game of go without human knowledge. nature **550**(7676), 354–359 (2017)

[2] Kober, J., Bagnell, J.A., Peters, J.: Reinforcement learning in robotics: A survey. The International Journal of Robotics Research **32**(11), 1238–1274 (2013)

[3] Zheng, G., Zhang, F., Zheng, Z., Xiang, Y., Yuan, N.J., Xie, X., Li, Z.: Drn: A deep reinforcement learning framework for news recommendation. In: Proceedings of the 2018 World Wide Web Conference, pp. 167–176 (2018)

[4] Ding, Z., Hernandez-Leal, P., Ding, G.W., Li, C., Huang, R.: CDT: cascading decision trees for explainable reinforcement learning. CoRR **abs/2011.07553** (2020) 2011.07553

[5] Coppens, Y., Efthymiadis, K., Lenaerts, T., Nowe, A.: Distilling deep reinforcement learning policies in soft decision trees. In: Miller, T., Weber, R., Magazzeni, D. (eds.) Proceedings of the IJCAI 2019 Workshop on Explainable Artificial Intelligence, pp. 1–6 (2019). IJCAI 2019 Workshop on Explainable Artificial Intelligence, XAI19 ; Conference date: 11-08-2019. https://sites.google.com/view/xai2019/home

[6] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: Openai gym. CoRR **abs/1606.01540** (2016) 1606.01540

[7] Barto, A.G., Sutton, R.S., Anderson, C.W.: Neuronlike adaptive elements that can solve difficult learning control problems. IEEE Transactions on Systems, Man, and Cybernetics **SMC-13**(5), 834–846 (1983) https://doi.org/10.1109/TSMC.1983.6313077

[8] Moore, A.W.: Efficient memory-based learning for robot control. Technical report, University of Cambridge (1990)

[9] Sutton, R.S.: Generalization in reinforcement learning: Successful examples using sparse coarse coding. In: Touretzky, D., Mozer, M.C., Hasselmo, M. (eds.) Advances in Neural Information Processing Systems, vol. 8. MIT Press, ??? (1995). https://proceedings.neurips.cc/paper_files/paper/1995/file/8f1d43620bc6bb580df6e80b0dc05c48-Paper.pdf