Human-in-the-Loop Interpretable Reinforcement Learning via Evolutionary Decision Trees

Hao Jie Zhan[1 [0009-0007-3767-0533]], Stefano Genetti[1 [0009-0004-2417-0319]], and Giovanni Iacca[1 [0000-0001-9723-1830]]

University of Trento, Italy

Abstract. Despite the growing importance of interpretable AI in high-stakes domains, human-in-the-loop methodologies that allow domain experts to actively shape interpretable models during the optimization process remain largely unexplored. In this paper, we present IELDT (Interactive Evolutionary Learning of interpretable Decision Trees), a novel framework that integrates Evolutionary Reinforcement Learning with a rich human-driven refinement interface. IELDT evolves populations of interpretable decision tree policies and periodically exposes the best-performing tree to a domain expert through an intuitive web interface, allowing manual or natural-language-guided edits. The human-crafted policy is then re-injected into the evolutionary loop via a Lamarckian genotype reconstruction algorithm. The framework will be released as an open-source, modular Python library compatible with the OpenAI Gym interface, facilitating broad adoption and applicability.

Keywords: Interpretable AI · Human-in-the-loop optimization · Evolutionary Reinforcement Learning · Decision Trees · Interactive Evolutionary Computation.

1 Motivation and background

The increasing adoption of Artificial Intelligence (AI) in safety-critical and high-stakes domains has amplified the need for models whose decision-making process can be understood, inspected, and refined by human experts. This has motivated two complementary research directions: explainable AI (XAI), which provides post-hoc insight into black-box decisions, and interpretable AI (IAI), which produces models that are transparent and human-readable by design [1, 5].

A persistent and widely held assumption is that interpretable models are necessarily less performant than their black-box counterparts. Recent work has begun to challenge this narrative [2, 3, 6, 9]: in particular, evolutionary IAI approaches based on interpretable decision tree (DT) policies have demonstrated competitive performance against state-of-the-art deep Reinforcement Learning (RL) baselines on standard benchmarks and real-world industrial tasks. Yet, despite this progress, one underexplored dimension of IAI remains the exploitation of human expertise during the optimization loop itself. Because the evolved policies are interpretable by construction, a domain expert can, in principle, inspect them, identify shortcomings, and inject domain knowledge directly. Notably, this capability is simply not available when training opaque neural networks.

Prior work [4] introduced a first step in this direction by allowing a human to cast a vote between the two best-performing DTs at fixed intervals, indirectly biasing selection pressure. While promising, this interaction paradigm is rather limited: in fact, the user cannot modify the policy itself, and the implementation is highly problem-specific and difficult to adapt to new tasks. These limitations motivate the present work, where we aim to allow for a richer user interaction.

2 Proposed method

We introduce IELDT (Interactive Evolutionary Learning of interpretable Decision Trees), a human-in-the-loop IAI framework built on two tightly coupled components (Figure 1).

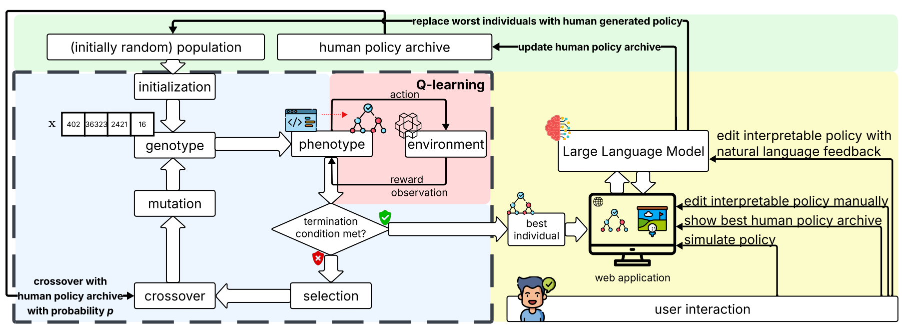

Fig. 1. Conceptual scheme of the proposed IELDT approach.

2.1 Evolutionary Reinforcement Learning backbone

The algorithmic core follows the two-level optimization scheme of [2]. An outer evolutionary loop based on Grammatical Evolution [10] searches the space of DT structures by encoding each individual as a fixed-length integer genotype, which is translated into a DT policy via a problem-dependent context-free grammar in Backus-Naur Form (BNF) [8]. An inner Q-learning loop then optimizes the action assignments at the DT leaves by interacting with an RL task, implemented as an OpenAI Gymnasium environment [12]. Fitness is measured as the mean cumulative reward over a fixed number of episodes, and the population evolves through tournament selection, subtree crossover, and uniform mutation at genotype level.

2.2 Human-driven refinement and feedback injection

Periodically, the best-performing DT of the current generation is presented to the human expert through a purpose-built web interface (Figure 2). After a post-processing step that prunes unreachable branches, the user can interact with the policy in several ways: they may simulate the policy on a live environment render, edit the DT directly through the graphical interface (adding, removing, or modifying nodes and branches), or issue free-form natural language instructions to a Large Language Model (LLM) middleware, which translates the feedback into concrete structural edits while enforcing grammar compliance. The user may also browse an archive of all previously crafted policies.

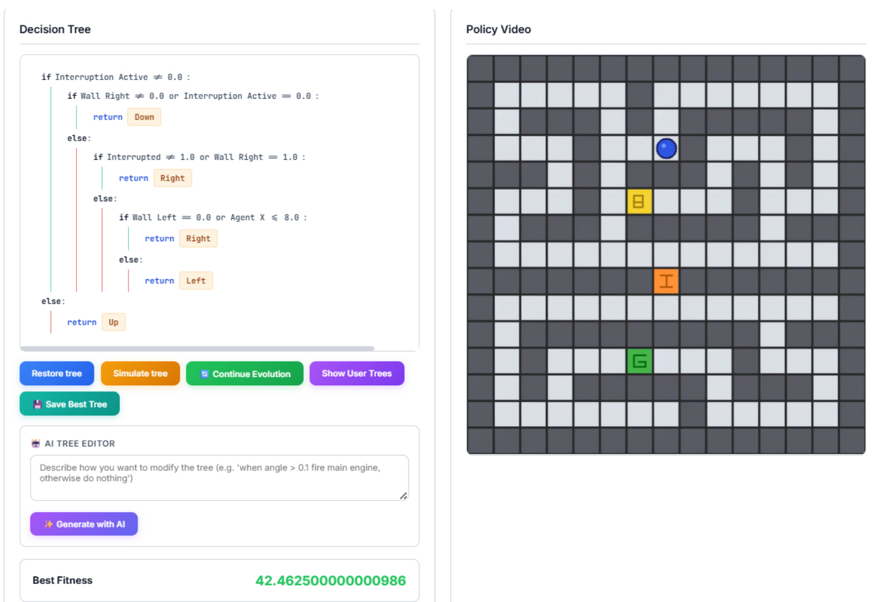

Fig. 2. Web interface implemented to allow the user to interact with the optimization process. In the example considered, a classical grid-world RL task inspired by the off-switch environment introduced in [7] is shown. On the left side, the user can see the interpretable DT, the performance achieved on the task, and the interaction commands. On the right side, the implementation of the `render` method allows the user to simulate the policy on the environment and visualize the results graphically.

Once the expert confirms the edited policy, re-injecting it into the evolutionary loop requires a translation from phenotype space back to genotype space. This is a non-trivial step that most Interactive Evolutionary Computation (IEC) [13] frameworks avoid. We address this through a deterministic Lamarckian reconstruction algorithm that converts any grammar-compliant DT phenotype back into a valid integer genotype. The human-crafted individual immediately replaces the worst member of the current population [11] and simultaneously seeds a dedicated archive H, which acts as a human-driven Hall-of-Fame. During subsequent generations, crossover is biased to preferentially combine offspring with individuals drawn from H with high probability, channelling evolution towards the structural preferences expressed by the domain expert.

2.3 Modular open-source infrastructure

A key contribution of IELDT is its generality. The framework will be released as a modular, open-source Python library. Applying IELDT to a new task requires: (i) encoding the problem as a standard OpenAI Gym environment, augmented with a `render` function and a textual `summary` method used by the LLM middleware; and (ii) specifying a problem-dependent BNF grammar. This design deliberately decouples the core optimization and interaction machinery from the task domain, making the framework reusable and highly customizable across RL tasks.

3 Research questions and expected contributions

The core hypothesis driving this work is that bidirectional human-AI interaction within an interpretable evolutionary RL loop can improve both optimization efficiency and solution quality while preserving, and indeed leveraging, the transparency of the evolved policies. Concretely, IELDT is designed to investigate the following research questions:

RQ1. Can interpretable DT-based policies match the performance of state-of-the-art black-box deep RL algorithms, directly challenging the assumed interpretability-performance trade-off?

RQ2. Does human-expert feedback injected through the IELDT interface accelerate convergence and improve the quality of the Pareto front in the performance-interpretability objective space, compared to the non-interactive evolutionary baseline?

RQ3. Can an LLM effectively serve as a surrogate domain expert for experimental evaluation, and how does LLM-guided optimization compare to genuine human-expert interaction?

Beyond answering these questions, we expect IELDT to contribute (i) a principled Lamarckian phenotype-to-genotype reconstruction algorithm for grammar-constrained DTs, (ii) an open-source, reusable human-in-the-loop IAI library, and (iii) insights into the design of LLM-mediated interaction for interpretable evolutionary optimization.

References

1. Barredo Arrieta, Alejandro and Díaz-Rodríguez, Natalia and Del Ser, Javier and Bennetot, Adrien and Tabik, Siham and Barbado, Alberto and Garcia, Salvador and Gil-Lopez, Sergio and Molina, Daniel and Benjamins, Richard and Chatila, Raja and Herrera, Francisco: Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion **58**, 82–115 (2020)
2. Custode, Leonardo Lucio and Iacca, Giovanni: Evolutionary learning of interpretable decision trees. IEEE Access **11**, 6169–6184 (2023)
3. Genetti, Stefano and Longobardi, Alberto and Iacca, Giovanni: Evolutionary reinforcement learning for interpretable decision-making in supply chain management. In: International Conference on the Applications of Evolutionary Computation (Part of EvoStar). pp. 187–203. Springer (2025)
4. Genetti, Stefano and Scarton, Giorgio and Formentini, Marco and Iacca, Giovanni: An intelligent Digital Twin based on machine learning for interpretable decision-making in manufacturing. International Journal of Production Economics p. 109841 (2025)
5. Gunning, David and Aha, David W.: DARPA's Explainable Artificial Intelligence Program. AI Magazine **40**(2), 44–58 (2019)
6. Kretowski, Marek: A memetic algorithm for global induction of decision trees. In: International Conference on Current Trends in Theory and Practice of Computer Science. pp. 531–540. Springer (2008)
7. Leike, Jan and Martic, Miljan and Krakovna, Victoria and Ortega, Pedro A and Everitt, Tom and Lefrancq, Andrew and Orseau, Laurent and Legg, Shane: AI safety gridworlds. arXiv preprint arXiv:1711.09883 (2017)
8. McCracken, Daniel D and Reilly, Edwin D: Backus-naur form (bnf). In: Encyclopedia of computer science, pp. 129–131. John Wiley and Sons Ltd. (2003)
9. Rudin, Cynthia: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature machine intelligence **1**(5), 206–215 (2019)
10. Ryan, Conor and Collins, John James and Neill, Michael O: Grammatical evolution: Evolving programs for an arbitrary language. In: Genetic Programming: First European Workshop. pp. 83–96 (1998)
11. Sakallioglu, Berfin and Santos, Frederico JJB and Parra, Daniel and Qiu, Yuxin and Nicolau, Miguel and Trujillo, Leonardo and Hidalgo, J Ignacio: Combining Grammatical Evolution with LLM-based Local Search to Improve Interpretability. In: International Conference on the Applications of Evolutionary Computation (Part of EvoStar). pp. 217–233. Springer (2026)
12. Towers, Mark and Kwiatkowski, Ariel and Balis, John and De Cola, Gianluca and Deleu, Tristan and Goulão, Manuel and Andreas, Kallinteris and Krimmel, Markus and Kg, Arjun and Perez-Vicente, Rodrigo and others: Gymnasium: A standard interface for reinforcement learning environments. Advances in Neural Information Processing Systems **38** (2026)
13. Wang, Yanan and Pei, Yan: A comprehensive survey on interactive evolutionary computation in the first two decades of the 21st century. Applied Soft Computing **164**, 111950 (2024)