Policy graphs in action: explaining single- and multi-agent behaviour using predicates

Adrian Tormos
Barcelona Supercomputing Center
adrian.tormos@bsc.es

Victor Gimenez-Abalos
Barcelona Supercomputing Center
victor.gimenez@bsc.es

Dmitry Gnatyshak
Barcelona Supercomputing Center
dmitry.gnatyshak@bsc.es

Sergio Alvarez-Napagao
Barcelona Supercomputing Center /
Universitat Politècnica de Catalunya
sergio.alvarez@bsc.es

Abstract

This demo shows that policy graphs (PGs) provide reliable explanations of the behavior of agents trained in two distinct environments. Additionally, this work shows the ability to generate surrogate agents using PGs that exhibit accurate behavioral resemblances to the original agents and that this feature allows us to validate the explanations given by the system. This facilitates transparent integration of opaque agents into socio-technical systems, ensuring explainability of their actions and decisions, enabling trust in hybrid human-AI environments, and ensuring cooperative efficacy. We present demonstrations based on two environments and we present a work-in-progress library that will allow integration with a broader range of environments and types of agent policies.

1 Introduction

The increasing use of Artificial Intelligence (AI) and Reinforcement Learning (RL) has introduced challenges related to the transparency of agents, particularly in scenarios where clear decision-making is vital for efficient human-agent interaction and cooperation. Providing explanations for agent behaviors across a diverse set of environments has become essential. We intend to demonstrate a method based on policy graphs that contributes to the topic.

1.1 Background

Explainability in Reinforcement Learning (XRL) has become increasingly important as the use of artificial agents in various applications grows. A detailed review by Krajna et al. [5] provides a wide-ranging overview of the challenges and solutions related to XRL, forming a basis for this research. When looking at the many methods available in the literature, it is important to keep a balance between providing detailed explanations while keeping them succinct and specific. Also, explanations can be provided after an action (post-hoc) or be integrated into the decision-making process (intrinsic). Other factors, such as environment type, policy type, and agent cardinality, also play a role in the complexity of XRL.

The approach that we follow in our research is Policy Graphs (PGs), which provide a possible solution to understanding the behavior of opaque agents based on a set of predicates. A PG is defined as a directed graph $G = (V, E)$, where discrete states are mapped to nodes and actions to edges, providing a clear representation of agent behavior. If a PG is expressive enough,

XAI in Action: Past, Present, and Future Applications @ NeurIPS 2023.

it can approximate the behavior of an opaque agent [1, 12, 9]. This method can be used in various tasks, spanning both single and multi-agent scenarios. Typically, explainable methods using PGs convert observed state and action into predicates after each environment iteration, providing post-hoc, proactive, and global explanations. These methods have been used mainly in single-agent tasks and are compatible with stochastic environments and policies [3, 1, 6].

Similar approaches that also use predicates to infer behavioral descriptions of opaque agents are those based in representing goal-directed behavior as transition models that can be formalized as sequences of operators or plans [11, 10, 2]. These approaches are especially suitable for agents that have been implemented or trained in a way that they follow a clear sequential decision making towards their goal. Policy graphs, on the other hand, do not make any assumption about the internal model of the agent nor about its decision making process, and thus should be also suitable in those cases where the decision making can take into account concurrent goals or where the decision making process is not optimally fitted to the goal. An example of this is in human agents or in agents that imitate human behavior, as we already studied in [9].

1.2 Objective of the Demonstration

The demonstration aims to present a practical application and visual example of how Policy Graphs (PGs) can represent the behavior of opaque agents in both single-agent and cooperative environments, particularly within the Cartpole and Overcooked-AI frameworks respectively. Among others, it will provide clear, illustrative examples of the process of understanding the actions and strategies of artificial agents, which is in particular crucial for facilitating effective collaboration with human users.

In more specific terms, the demonstration will showcase:

- The construction and utilization of PGs to map and represent agent behavior in discrete states and actions, employing directed graphs in the Cartpole and the Overcooked-AI environments.
- How PGs can be utilized to approximate the behavior of an opaque agent into what we call a PG-based agent, providing insights into the agent's decision-making process.
- Analysis of different PG complexities and how they correlate with agent behavior, studying various strategies of generation of PG-based agents such as Greedy Policy Graphs and Stochastic Policy Graphs.
- The application of PGs in both single-agent (Cartpole) and multi-agent (Overcooked-AI) tasks.
- A side-by-side comparison of agent behaviors with their respective PG representations, enabling viewers to observe the correlation between them and validate the efficacy of the PG model.

The intent of this demonstration is to offer a tangible and accessible means for viewers to explore the concepts discussed in the paper, catering to both those familiar and unfamiliar with the technicalities of agent behavior modeling and PG construction. Furthermore, by observing the application in a real-world simulation (Overcooked-AI), attendees will gain insights into the practical implications and potential of utilizing PGs for explaining opaque agent behavior in cooperative, task-oriented scenarios.

The demonstration is visual and interactive: the behavior of the agents is browseable in the form of a visual graph, it is possible to ask questions to the graph using a simple interface, and the behavior of the original and the PG-based agents can be compared by actually running them in the environment. Therefore, we aim to make the content of our research accessible and comprehensible to a broad audience, including those who might not have deep technical knowledge in the aspects of policy modeling and agent behavior analysis.

2 Policy graphs for explainability

From behavior to policy graphs The construction of a PG begins with observing an agent's interactions within the environment, systematically recording state transitions and subsequent actions. To render an agent's behavior into a coherent PG, we reduce the observations and actions (which might be continuous) into discrete states and actions, which are subsequently mapped into nodes and edges in the graph.

A crucial step is to decide which predicates to use for representing these discrete elements, trying to balance producing PGs that precisely mirror the observed behavior versus keeping the complexity of the resulting PG as low as possible. Having very complex PGs with many predicates and/or with many values for their parameters can compromise the understandability of the explanations that will be produced in later stages. The predicates used must be expressive enough to distinguish between states in which the policy differs substantially. This separability heuristic is a good guide for an initial design: include as many predicates as it is possible when obtaining data, and then compare different discretizers composed of subsets of the original pool using the proposed metrics.

Once these decisions are made, the PG is created through a frequentist approach, counting each time a transition between two states when the agent performed a certain action, and computing transition probabilities. This includes both the probability of performing an action in a state and the probability of arriving at a state when doing so. Each non-zero probability becomes an edge of the graph.

Producing explanations from policy graphs Generating understandable explanations of agent behavior through PGs involves traversing the directed graph to illustrate the decision-making process. There are three different types of questions that can be answered, and therefore there are three different algorithms that define exactly how this traversal process is executed:

- What will you do when you are in state X?
- When do you perform action A?
- Why did you not perform action A in state X?[1]

These three algorithms produce explanations that can be converted into natural language by means of the predicates chosen. Thus, explanations can be textual (through language) or visual (through graph traversal), offering easy ways to generate insights into the strategic choices of the agent.

Generating PG-based agents Another way to evaluate the quality of the explanation is by trying to understand whether these explanations really represent the behavior of the agent being analyzed. This can be achieved by generating a policy that uses the graph to decide the next action based on the node of the policy graph that more closely resembles the current state and checking the probabilities of the edges representing the next actions.

We call PG agent to an agent that uses this policy to try to mimic the behavior of the original agent. During operation, when the agent encounters a state, it refers to the PG to determine the next action, adhering to the policy depicted by the graph. The agent continues to navigate through the environment, at each step referencing the PG to dictate its actions.

The accuracy with which PG-based agents emulate the behavior of the original, potentially more opaque agent is vital, as it verifies the PG's reliability as a representative model: if the simulacra matches the original's performance, the actions critical to being performant must be captured and thus explanations regarding them should be trustworthy. For non-critical actions, those not related to rewards, we can make no claim on whether the two agents are similar.

3 The *pgeon* library

We are developing a library with the intention to offer a flexible framework to allow developers to generate and utilize policy graphs. It will be publicly and openly released at `https://github.com/HPAI-BSC/pgeon`. The core features of this library are:

- Policy graph (PG) generation via Gym/PettingZoo: this feature enables the generation of policy graphs utilizing standard RL interfaces. Users can model and evaluate RL agent

[1] Counterfactual explanation.

behaviors within a specified environment while automatically generating CSV traces that can be used to produce the PGs or analyzed independently.

- Policy graph generation from CSV traces: this feature facilitates the creation of policy graphs from agent behavior traces stored in CSV format.
- Policy generation based on policy graphs: this feature ensures users to generate RL policies grounded in policy graphs, applying both greedy and stochastic strategies. This enables an analytic approach to behavior analysis, by allowing developers to quickly test the behavior of agents that are explainable by default.
- Policy graph visualization: this feature enable users to visualize policy graphs, allowing users to find behavioral patterns and strategic decisions on the graph.
- Query algorithms on policy graphs: this feature will implement some basic graph traversal algorithms able to answer queries on policy graphs.

4 Practical demonstrations

We demonstrate the use of our library with two distinct settings: Cartpole for analyzing a single agent in a continuous environment, and Overcooked-AI for analyzing a multi-agent system in an environment with complex actions and the occasional need for cooperation. In this section, we summarize the main highlights of our research done in both environments that can be shown in the demonstration.

4.1 Cartpole

More information about this use case can be found at [1] and the code for the demonstration is available at `github.com/HPAI-BSC/pgeon`

In the Cartpole setting, a single agent interacts within a continuous environment, providing a suitable context to analyse specific agent behaviors and decision-making processes in the domain of reinforcement learning. The simplicity of Cartpole, consisting of a cart that moves along a track and a pole that is attached to it, offers clarity in interpreting and demonstrating fundamental principles of agent learning and behavior. This environment was utilized to observe how an agent learns to balance the pole vertically by applying forces to the cart. The agent's objective is to maximize the duration for which the pole remains upright by moving the cart left or right, which is measured through the number of time steps survived in the environment without the pole falling or the cart moving out of bounds. The observations and actions depend on real and therefore continuous value, so discretising the states and actions is a complex task in itself.

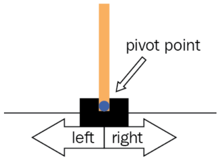


Figure 1: Cartpole scenario

The library is employed to generate explanations for the actions taken by the agent, thus allowing for an investigation into the policy learned during training. Special attention is paid to the agent's behavior in various key states, such as how it manages to recover the pole when it begins to fall and how it navigates to avoid reaching the track limits.

Our research shows that, with the proper set of predicates, an explainable PG agent can be created that has a similar behavior and performance in the environment.

4.2 Overcooked-AI

More information about this use case can be found at [12, 9] and the code for the demonstration is available at `github.com/MarcDV1999/Overcooked-Explainability`, `github.com/HPAI-BSC/explainable-agents-with-humans` and `github.com/HPAI-BSC/pgeon`

In the context of Overcooked-AI, our exploration mainly focuses on understanding agent behavior in the presence of other agents and the dynamics of cooperation in a structured, yet complex environment. The game, inspired by the popular video game Overcooked, involves two players who must work together to prepare and serve soups. Players navigate through the kitchen, preparing ingredients, cooking, and delivering finished dishes while managing spatial constraints (they can block each other) and task distribution.

Our demonstration primarily illustrates the process of mapping their behaviour onto PGs by means of a set of 10 predicates, two of them being related to the state of the other agent. These two predicates allow us to achieve explanations that involve the emergence of cooperation between the agents. Therefore, an interesting insight of this demonstration is that PGs can approximate the behavior of opaque agents if the set of predicates chosen is sufficiently expressive, an intelligible textual or visual representation of agent strategies and decisions in various game configurations.

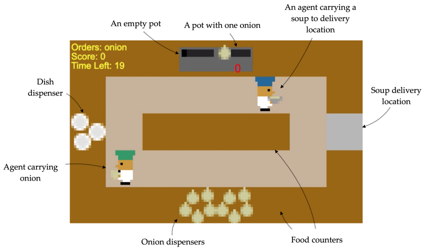
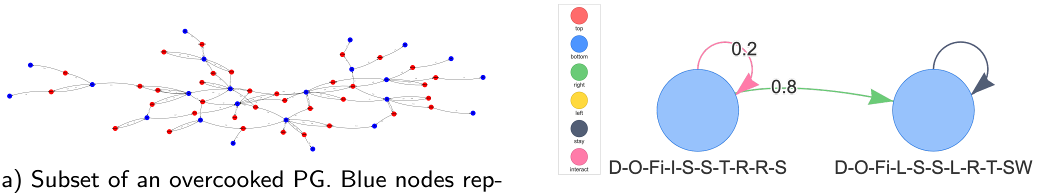


Figure 2: Overcooked-AI scenario

The demonstration includes simulations of the game in which PG agents that mimic the original (PPO [7] and GAIL [4]) agents. We allow for two different policy generation strategies –Greedy Policy Graphs and Stochastic Policy Graphs [9]– and elucidate how they affect the performance of the agent depending on the specific layout and the need for cooperation in each one.



(a) Subset of an overcooked PG. Blue nodes represent states, red nodes are actions. Each edge represents either the probability of an action $P(a|s)$ for blue-red edges, or the probability of a world update $P(s'|a, s)$ for red-blue edges.

(b) Two-state system example. In the left state, the agent either with 20% probability interacts and does not change the state, or moves right and switches to the right state.

5 Conclusions

In this paper, we have contextualized using policy graphs for achieving explainability of agents with opaque policies, we have discussed how to generate such policy graphs and use them to generate explanations and how to validate these explanations, e.g. by the creation of PG agents that mimic the behaviour of the original agent. Our research has led us to experiment with this explainability technique into two distinct environments, and that has led us to produce a library that abstracts the different algorithms that this method requires. We aim at demonstrating the library, especially through the two use cases, showing how to achieve textual and visual representations of the behavior of agents and how the PG agents, explainable by default, behave in the environment.

So far, in the scenarios we have studied, the predicates used have been designed through iterative refinement processes in where expert knowledge about the environments was combined with experimentation about the reliability of the resulting explanations depending on the available predicates. We aim to test and integrate approaches that exist in the literature for automatic exploration and modeling of predicates from opaque agents [8, 2]. More so, the information encoded in a PG may include potential for different kinds of explanation than those provided. We endeavour to find new uses for this information.

Acknowledgments and Disclosure of Funding

This work has been partially supported by the HORIZON-CL5-2022-D6-01 Trustworthy AI for CCAM (AI4CCAM, Grant agreement ID: 101076911) and the H2020 knowlEdge (Grant agreement ID: 957331) European projects. The authors declare no conflict of interest.

References

[1] Climent, A., Gnatyshak, D., Alvarez-Napagao, S.: Applying and Verifying an Explainability Method Based on Policy Graphs in the Context of Reinforcement Learning. In: Villaret, M., Alsinet, T., Fernández, C., Valls, A. (eds.) Frontiers in Artificial Intelligence and Applications. IOS Press (Oct 2021). https://doi.org/10.3233/FAIA210166, https://ebooks.iospress.nl/doi/10.3233/FAIA210166

[2] Das, D., Chernova, S., Kim, B.: State2Explanation: Concept-Based Explanations to Benefit Agent Learning and User Understanding (Nov 2023), https://openreview.net/forum?id=xGz0wAIJrS

[3] Hayes, B., Shah, J.A.: Improving Robot Controller Transparency Through Autonomous Policy Explanation. In: Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction. pp. 303–312. ACM, Vienna Austria (Mar 2017). https://doi.org/10.1145/2909824.3020233, https://dl.acm.org/doi/10.1145/2909824.3020233

[4] Ho, J., Ermon, S.: Generative Adversarial Imitation Learning. Proceedings of the 30th International Conference on Neural Information Processing Systems pp. 4572–4580 (2016)

[5] Krajna, A., Brcic, M., Lipic, T., Doncevic, J.: Explainability in reinforcement learning: perspective and position. arXiv preprint arXiv:2203.11547 (2022)

[6] Liu, T., McCalmon, J., Le, T., Rahman, M.A., Lee, D., Alqahtani, S.: A novel policy-graph approach with natural language and counterfactual abstractions for explaining reinforcement learning agents. Autonomous Agents and Multi-Agent Systems **37**(2), 34 (Aug 2023). https://doi.org/10.1007/s10458-023-09615-8, https://doi.org/10.1007/s10458-023-09615-8

[7] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal Policy Optimization Algorithms (Aug 2017). https://doi.org/10.48550/arXiv.1707.06347, http://arxiv.org/abs/1707.06347, arXiv:1707.06347 [cs]

[8] Silver, T., Chitnis, R., Kumar, N., McClinton, W., Lozano-Perez, T., Kaelbling, L.P., Tenenbaum, J.: Predicate Invention for Bilevel Planning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37 (10), pp. 12120–12129. arXiv (Nov 2022). https://doi.org/10.48550/arXiv.2203.09634, http://arxiv.org/abs/2203.09634, arXiv:2203.09634 [cs]

[9] Tormos, A., Gimenez-Abalos, V., Domènech i Vila, M., Gnatyshak, D., Alvarez-Napagao, S., Vázquez-Salceda, J.: Explainable agents adapt to human behaviour. In: Proceedings of the First International Workshop on Citizen-Centric Multi-Agent Systems (CMAS'23). pp. 42–48 (2023), https://upcommons.upc.edu/handle/2117/390757

[10] Verma, P., Karia, R., Srivastava, S.: Autonomous Capability Assessment of Sequential Decision-Making Systems in Stochastic Settings (Extended Version). In: Proceedings of the International Conference on Automated Planning and Scheduling. vol. 33 (1), pp. 189–197. arXiv (Oct 2023). https://doi.org/10.48550/arXiv.2306.04806, http://arxiv.org/abs/2306.04806, arXiv:2306.04806 [cs]

[11] Verma, P., Marpally, S.R., Srivastava, S.: Discovering User-Interpretable Capabilities of Black-Box Planning Agents. arXiv (May 2022). https://doi.org/10.48550/arXiv.2107.13668, http://arxiv.org/abs/2107.13668, arXiv:2107.13668 [cs]

[12] Domènech i Vila, M., Gnatyshak, D., Tormos, A., Alvarez-Napagao, S.: Testing Reinforcement Learning Explainability Methods in a Multi-Agent Cooperative Environment. In: Cortés, A., Grimaldo, F., Flaminio, T. (eds.) Frontiers in Artificial Intelligence and Applications. IOS Press (Oct 2022). https://doi.org/10.3233/FAIA220358, https://ebooks.iospress.nl/doi/10.3233/FAIA220358