In HRI 2025 Workshop on Explainability for Human-Robot Collaboration: Real-World Concerns (X-HRI 2025)
Melbourne, Australia

Interpretability Analysis of Symbolic Representations for Sequential Decision-Making Systems

Pulkit Verma
MIT CSAIL
Cambridge, MA, USA
pulkitv@mit.edu

Julie A. Shah
MIT CSAIL
Cambridge, MA, USA
julie_a_shah@csail.mit.edu

Abstract

Interpretability in sequential decision-making (SDM) systems is critical for ensuring trust and transparency in human-robot collaboration scenarios. As robots increasingly work alongside humans in manufacturing, healthcare, and service environments, their decision-making processes must be understandable to their human collaborators. While significant progress has been made in interpretability for single-step decision-making systems, there remains a lack of consolidated research on interpretability techniques for SDM systems. This work analyzes various symbolic representations, evaluating their interpretability and applicability for effective human-robot teaming. We introduce a framework for analyzing these representations along key dimensions including interpretability, temporal expressiveness, and human-robot interaction capabilities. By synthesizing existing work and highlighting open challenges, this work guides researchers in selecting and designing interpretable symbolic representations that enhance trust in human-robot collaborative tasks.

Keywords

Interpretability, Sequential Decision-Making, Symbolic models

1 Introduction

While substantial research has been conducted on interpreting single-step decision-making systems [11, 17, 22, 28, 53], there remains a significant gap in consolidating interpretability techniques for sequential decision-making (SDM) systems, particularly in symbolic form. Existing work in this area has explored various approaches, such as PDDL-like languages [24], decision trees [31, 37, 40], and post-hoc explainability methods for reinforcement learning policies [42]. Additionally, some progress has been made in generating interpretable policies for reinforcement learning [15, 43] and generating environments that can lead to interpretable behavior of SDM systems [21]. However, these efforts are often fragmented, focusing on specific aspects of SDM interpretability. This highlights the need for a thorough review that organizes and synthesizes existing techniques, making the way for more cohesive advancements in SDM interpretability.

SDM systems, which form the core of autonomous robots and collaborative agents, present unique interpretability challenges due to their temporal dependencies and complex state spaces. In human-robot collaboration scenarios, understanding a robot's decision-making process becomes crucial for effective teamwork and trust. While neural approaches have shown strong performance in robot control and decision-making, their black-box nature limits human operators' ability to predict and understand robot behavior. Symbolic representations offer more structured, human-readable frameworks, ranging from Finite State Machines to Markov Decision Processes and Temporal Logic, making them valuable for collaborative settings where humans need to quickly grasp robot intentions and reasoning.

However, these symbolic approaches face trade-offs between interpretability and expressiveness. Simple representations like FSMs offer clarity but limited scalability, while more powerful frameworks like MDPs can become opaque as complexity grows - a critical concern when robots need to explain their actions to human collaborators in real-time. This work analyzes various symbolic representations across key dimensions including interpretability, temporal expressiveness, and abstraction level, providing insights for developing more interpretable robotic SDM systems that can facilitate natural and efficient human-robot collaboration in real-world applications.

2 Sequential Decision Making Systems

Sequential decision-making (SDM) systems are characterized by their ability to make a series of decisions over time, where each decision influences the subsequent state of the system. Most commonly, such systems can be represented using a Markov Decision Process (MDP) [33]. An MDP is defined by a set of states, actions, transition probabilities, and rewards. At each time step, the system is in a specific state, and an action is chosen based on a policy. The action leads to a transition to a new state according to a probabilistic transition function, and a reward is received based on the chosen action and the resulting state. The goal is to find an optimal policy that maximizes the cumulative reward over time. MDPs are widely used in reinforcement learning, robotics, and control systems due to their ability to model uncertainty and dynamic environments.

MDPs offer a structured and symbolic representation of decision-making processes. States and actions can be labeled with meaningful descriptions, making it easier to understand the high-level structure of the system. For example, in a robotics application, states might represent physical locations, and actions could correspond to movements like "move forward" or "turn left." This symbolic labeling allows humans to trace the sequence of decisions and transitions, providing insight into how the system operates.

However, the interpretability of MDPs is limited by their probabilistic nature. While the states and actions are interpretable, the transition probabilities between states are often represented as matrices or tables, which can be difficult to comprehend, especially in large systems. Additionally, the optimal policy derived from an MDP (e.g., via value or policy iteration) may not always be intuitive, as it is based on maximizing long-term rewards rather than following human-understandable rules. This can make it challenging to explain why certain actions are preferred over others in specific states. Despite this, there have been attempts to explain the decisions made by known-MDP-based systems [12, 20]

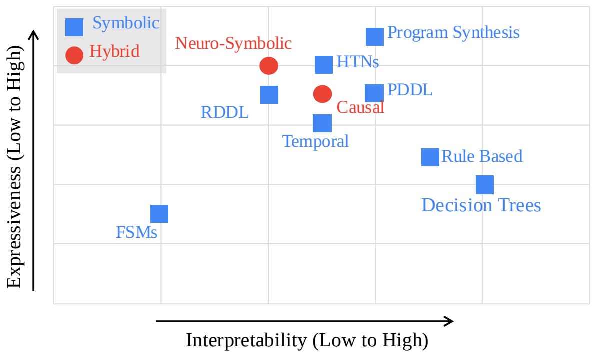

Figure 1: Trade-off between interpretability and expressiveness in sequential decision-making representations

Classification Dimensions To address the aforementioned challenges, researchers have developed alternative approaches to represent SDM systems. There are many criteria using which we can categorize these approaches. Tab. 1 presents seven such possible classification criteria. We use the formalism types to classify these approaches rather than the six other dimensions due to several practical considerations. First, many representations exhibit varying levels of capability across these dimensions depending on their specific implementation and application context. For example, a decision tree's interpretability level can range from high to low depending on its depth and complexity, making strict categorization challenging. Second, these dimensions often interact in complex ways - improvements in temporal expressiveness might come at the cost of interpretability level, while domain specificity could affect both abstraction level and explanation type. By focusing on representation types, we can better explore these interactions and trade-offs.

Classification based on Formalism Type As illustrated in Fig. 1, representations can be broadly categorized into three formalism types: symbolic, subsymbolic, and hybrid. Symbolic representations are explicit, logic-based, and rule-driven, making them highly interpretable and well-suited for structured environments. Subsymbolic representations, on the other hand, are data-driven and often rely on machine learning techniques, enabling them to handle complex, unstructured data but at the cost of interpretability. Hybrid representations combine the strengths of both symbolic and subsymbolic approaches, leveraging the interpretability of symbolic methods and the flexibility of subsymbolic techniques to address challenges in dynamic and uncertain environments. In the following sections, we will explore the symbolic representations, followed by the hybrid methods, highlighting their unique characteristics, applications, and trade-offs. Since we are focusing on symbolic representations, we exclude a detailed discussion on subsymbolic approaches.

3 Symbolic Representations

3.1 Finite State Machines (FSMs)

Finite State Machines (FSMs) are mathematical models that represent systems through distinct states, transitions, and associated actions. Similar to a flowchart, FSMs use nodes to represent states and arrows to show transitions between states based on specific inputs or conditions. In sequential decision-making systems, each decision point corresponds to a state, with transitions triggered by specific choices. For instance, a customer service workflow might include states like "awaiting response," "under review," and "resolved," with transitions driven by customer or representative actions.

Interpretability: FSMs excel in interpretability through several key mechanisms. States and transitions can be labeled with human-understandable descriptions, making the logic easily traceable [35]. [14] identified four aspects supporting FSM interpretability: intuitive graphical representation as cyclic, directed graphs; transparent computation enabling manual verification; generative nature allowing sampling and formal property queries; and extensive theoretical and practical study making them accessible for system design. Recent research has demonstrated FSMs' value in interpreting complex systems like recurrent neural networks (RNNs). By converting RNNs' intricate memory and observations into simplified states and transitions, FSMs can function like a digital map that highlights main routes while abstracting away unnecessary complexity [16, 9].

Strengths and Limitations: FSMs offer several significant advantages in sequential decision-making systems. They excel at simplifying and tracing complex decision sequences, proving highly effective for systems with clear, discrete decision points. Their ability to provide transparent and auditable decision trails makes them particularly valuable for small to medium-sized systems. However, FSMs also face notable limitations. They struggle with decisions requiring nuanced, continuous reasoning and face scalability challenges due to state explosion in complex systems. Additionally, FSMs are not well-suited for continuous or high-dimensional spaces, potentially falling short in highly complex scenarios where more sophisticated approaches might be necessary.

3.2 Decision Trees

Decision trees [2] are highly interpretable models structured in a tree-like format, where each node represents a decision based on certain specific conditions. The decision-making process begins at the root node and follows a series of splits determined by feature thresholds until reaching a leaf node that provides the predicted outcome. This hierarchical structure enables a clear, step-by-step decision process that makes SDM policies easily traceable and explainable [1].

Interpretability: The interpretability of decision trees stems from their ability to decompose predictions into individual feature contributions along the decision path. For any given instance, the prediction can be understood as the mean target value plus the contributions of each feature encountered during traversal. This transparency made decision trees particularly valuable for classification and regression tasks in data-driven systems [23]. Recent developments have further enhanced their application in complex domains like creating interpretable decision trees for challenging reinforcement learning tasks [8, 31, 37, 40].

Strengths and Limitations: Decision trees offer significant advantages in their interpretability and transparency, making them valuable when clear explanation of decisions is required. Their step-by-step nature allows for easy tracing of the decision process, and their hierarchical structure provides intuitive understanding of feature

Representation	Interpretability Level	Formalism Type	Temporal Expressiveness	Abstraction Level	Explanation Type	Domain Specificity	Human Interaction
Markov Decision Processes (MDPs)	Medium	Symbolic	Discrete Time	Low-Level	Global	General	Indirect
Finite State Machines (FSMs)	Medium	Symbolic	Discrete Time	Low-Level	Global	General	Direct
Decision Trees	High	Symbolic	N/A	Low-Level	Global	General	Direct
Rule-Based Systems	High	Symbolic	N/A	Low-Level	Global	General	Direct
Temporal Logic (LTL, STL, etc.)	Medium	Symbolic	Continuous Time	Low-Level	Global	Domain	Indirect
Program Synthesis	Low	Symbolic	N/A	High-Level	Global	Domain	Indirect
Planning Domain Definition Language (PDDL)	Medium	Symbolic	Discrete Time	High-Level	Global	Domain	Indirect
Causal Models	Medium	Hybrid	N/A	Multi-Level	Global	General	Indirect
Neuro-Symbolic Integration	Low	Hybrid	N/A	Multi-Level	Global	General	Indirect

Table 1: Classification of Interpretable Representations for Sequential Decision-Making Systems along different dimensions

importance. However, they face notable limitations that can impact their effectiveness. They are susceptible to overfitting, especially with increased tree depth, which can lead to complex and less generalizable models [28]. They also struggle with linear relationships, approximating them through step functions that can result in unstable and unintuitive predictions. [37] note their difficulty in handling online updates, particularly in reinforcement learning contexts. Despite these challenges, decision trees remain widely used for interpretable modeling, particularly when tree depth is constrained to maintain simplicity and understandability.

3.3 Rule-Based Systems

Rule-based systems (RBSs) are a type of symbolic representation where knowledge is encoded as “if-then” rules, consisting of a condition (the “if” part) and an action or conclusion (the "then" part). These systems mimic human reasoning by applying logical rules to input data to derive decisions. While much of the interpretability analysis of RBSs focuses on single-step decision-making [4, 54], they can be easily extended to sequential decision-making. E.g., in a traffic light control system, a rule might state: “If the north-south road has heavy traffic and the east-west road is clear, then extend the green light for north-south traffic.” RBSs are also used in industrial automation, robotics, and workflow management, where sequences of decisions must be made based on clear, logical rules.

Interpretability: Rule-based systems are highly interpretable due to their explicit and transparent structure. Each rule is a self-contained unit of logic that can be easily understood, validated, and modified. This transparency makes it straightforward to trace how a sequence of decisions was reached, which is critical in domains like traffic control, industrial automation, and robotics. For instance, in a traffic light system, operators can review and adjust rules to optimize flow or adapt to changing conditions. This interpretability is a key strength, particularly in applications where accountability and trust are essential.

Strengths and Limitations: Rule-based systems offer significant interpretability advantages, as their explicit "if-then" structure aligns well with human reasoning and is easy to understand. This makes them ideal for sequential decision-making in domains like traffic control, where complex sequences of decisions must be explainable and transparent. However, they struggle with scalability: as decision sequences grow more complex, the number of rules can explode, making systems difficult to manage. Additionally, rule-based systems are deterministic and lack flexibility in handling uncertainty or adapting to dynamic environments. These limitations highlight the need for hybrid approaches that combine the interpretability of rule-based systems with the adaptability of other techniques.

3.4 Temporal Logic

Temporal logic is a formalism used to reason about sequences of states or events over time. It provides a symbolic and mathematical framework for expressing temporal properties, such as “eventually,” “always,” or “until,” describing how a system evolves. Linear Temporal Logic (LTL) and Signal Temporal Logic (STL) are two prominent variants. LTL is used for discrete systems and focuses on sequences of states, while STL extends this to continuous signals, making it suitable for real-time systems. Property Specification Language is another such interpretable formalism used to express interpretable temporal models [34]. Temporal logic is widely used in formal verification, control systems, and robotics to specify and verify properties like safety, liveness, and reactivity. E.g., in a robotic system, an LTL formula might express: “The robot will eventually reach the goal while avoiding obstacles.”

Interpretability: Temporal logic offers high-level, symbolic descriptions of system behavior, which are interpretable when used correctly [3, 10, 7, 35]. Its formal syntax allows precise specification of temporal properties, such as “The system will always remain safe,” providing clear and unambiguous representations. However, interpretability depends on the user’s familiarity with its syntax and semantics. For non-experts, the abstract nature of temporal logic can be challenging, and interpreting verification results (e.g., why a property was violated) often requires expertise. Also, it has been shown that even for experts, STL can be tricky to interpret [38, 18].

Strengths and Limitations: Temporal logic is highly expressive and formal, making it powerful for specifying and verifying temporal properties in domains like control systems and robotics. Its symbolic nature enables precise descriptions of complex behaviors. However, this expressiveness comes with complexity: formulas can be difficult to understand, and scalability becomes an issue in large or highly complex systems. Despite these challenges, temporal logic remains a valuable tool, especially when paired with visualization or natural language translation to improve accessibility.

3.5 Program Synthesis

Program synthesis automatically generates human-readable programs that represent decision-making logic, often from high-level specifications or examples. These programs capture the rules or patterns of a system in a structured, interpretable form, unlike traditional "black-box" models. By producing code in familiar programming languages or symbolic forms, program synthesis bridges the gap between complex, data-driven models and interpretable representations. This makes it particularly valuable for sequential decision-making systems, where understanding the reasoning behind decisions is critical. For example, in robotics, synthesized programs can encode decision sequences for navigation or task execution, providing clear and actionable logic.

Interpretability: Program synthesis offers strong interpretability when the generated programs are simple, well-structured, and written in a language familiar to the end-user. This allows stakeholders to trace decision-making step-by-step, verify logic, and even refine the program [43, 41, 55]. For instance, in healthcare or finance, synthesized programs can provide transparent explanations of decisions, fostering trust and compliance with regulations. Additionally, program synthesis can compactly represent complex logic in a human-readable form, making it easier to communicate system behavior to non-experts. However, interpretability can be compromised if the synthesized programs become complex or use unfamiliar representations, limiting their accessibility.

Strengths and Limitations: Program synthesis excels in producing interpretable, structured representations of decision-making logic, making it ideal for domains requiring transparency, such as robotics, healthcare, and finance. Its ability to generate human-readable programs allows for verification, debugging, and refinement, aligning well with regulatory and ethical requirements. However, the approach has limitations: synthesized programs can become convoluted for complex systems, reducing interpretability. Also, program synthesis is computationally expensive and often restricted to domains where code can effectively represent logic, hence cannot be easily scaled [27]. These challenges highlight the need for careful design and evaluation to balance interpretability and scalability.

3.6 PDDL (Planning Domain Definition Language)

The Planning Domain Definition Language (PDDL) [26] is a symbolic language used to define planning problems and domains in artificial intelligence. It provides a formal framework for specifying actions, preconditions, effects, and goals. For example, in robotics, an action like "pick up an object" might have preconditions (e.g., the robot must be near the object) and effects (e.g., the object is now held by the robot). PDDL is widely used in AI planning for applications such as robotics, logistics, and autonomous systems, enabling the generation of plans to achieve goals.

Interpretability: PDDL is highly interpretable due to its explicit and modular representation of planning domains [46, 47, 44, 45]. Its structure separates actions, preconditions, effects, and goals, making it easy to trace how plans are generated and executed. This transparency is valuable for debugging and validation, as users can inspect each step to understand the reasoning behind actions. PDDL's human-readable syntax also makes it accessible to domain experts, fostering collaboration. PDDL domains can also be converted into graph representations for enhanced interpretability, particularly in smaller problem spaces. In such cases, states and actions become nodes and edges in the graph, with the visual and modular nature of the representation enabling users to understand system interactions more intuitively [50]. However, interpretability depends on the quality of the manually defined domain specification, which can be time-consuming and error-prone, especially in complex domains.

Strengths and Limitations: PDDL excels in interpretability, offering a clear and modular representation of planning problems. Its explicit separation of actions, preconditions, and effects makes it easy to trace and validate plans, which is critical in domains like robotics and logistics. Additionally, its human-readable syntax enables collaboration with domain experts. However, PDDL has limitations: manual domain specification is labor-intensive and prone to errors, and the language struggles with uncertainty and scalability in large or dynamic environments. These challenges highlight the need for complementary techniques, such as automated domain learning or probabilistic methods, to enhance its applicability [30, 36, 19, 6].

4 Hybrid Representations

4.1 Causal Models

Causal models [32] are frameworks designed to represent cause-and-effect relationships in decision-making systems, moving beyond traditional statistical correlations to uncover underlying mechanisms. Using tools like directed acyclic graphs (DAGs), structural equation models, and counterfactual reasoning, these models explicitly represent causal relationships between variables. They find widespread application in domains such as healthcare, economics, and policy-making, where understanding intervention impacts is crucial.

Interpretability: The interpretability of causal models stems from their ability to explain the "why" behind decisions, not just what happens. By explicitly modeling cause-and-effect relationships, they enable reasoning about interventions and counterfactuals [52, 48]. For instance, in healthcare, these models can explain why specific treatments lead to better outcomes, helping clinicians make informed decisions. However, their interpretability heavily depends on the accuracy of underlying assumptions and proper specification of causal structures.

Strengths and Limitations: Causal models excel in providing actionable insights into decision-making processes, offering clear explanations of why outcomes occur rather than just showing correlations [25, 29, 51]. This makes them particularly valuable for understanding intervention impacts in domains like healthcare and policy-making. However, they face significant challenges in construction and validation, requiring precise domain knowledge and often relying on assumptions that are difficult to verify. Their applicability is also limited to domains where causality is well-defined, making them less suitable for highly complex or uncertain environments.

4.2 Neuro-Symbolic Integration

Neuro-symbolic integration is an approach that combines the strengths of neural networks (sub-symbolic AI) with symbolic reasoning methods to create systems that are both powerful and interpretable [49]. Neural networks excel at handling unstructured data, such as images, text, and audio, and can learn complex patterns from large datasets. However, they often operate as black boxes, making their decision-making processes difficult to understand. Symbolic methods, on the other hand, use logical rules and structured representations to reason about problems, providing clear and interpretable explanations but struggling with scalability and flexibility. Neuro-symbolic integration seeks to bridge this gap by embedding symbolic reasoning into neural networks or using neural networks to enhance symbolic systems. For example, in a medical diagnosis system, a neuro-symbolic model might use a neural network to process patient data and a symbolic reasoning component to apply medical guidelines, ensuring both accuracy and interpretability.

Interpretability: Neuro-symbolic integration enhances interpretability by merging the learning power of neural networks with the transparency of symbolic methods [13]. This hybrid approach makes decision-making more understandable, as users can see both data-driven insights and logical rules. However, successful integration depends on balancing performance and clarity, as poorly designed systems may either complicate explanations or reduce accuracy.

Strengths and Limitations: Neuro-symbolic integration provides strong interpretability, especially for complex tasks requiring both data-driven learning and logical reasoning. It's valuable in high-stakes fields like healthcare and finance, where transparency is crucial. However, the field is still developing, with tools not as mature as purely neural or symbolic approaches. Designing effective systems requires careful balance, and integration can add complexity, potentially reducing interpretability if not done thoughtfully. Despite these challenges, neuro-symbolic integration shows promise for building capable and understandable AI systems.

5 Future Directions

Future research in interpretable AI should focus on hybrid neuro-symbolic approaches that combine the scalability of neural networks with the interpretability of symbolic reasoning, while addressing challenges like balancing performance and transparency. Scalability improvements, such as hierarchical representations and rule simplification, can mitigate state explosion in symbolic systems. Additionally, integrating probabilistic reasoning into symbolic frameworks will enhance their applicability in uncertain environments. Human-centered design, including natural language explanations and interactive visualizations, is essential for making interpretability accessible to non-experts. Standardized evaluation metrics for interpretability, domain-specific adaptations, and advancements in explainable reinforcement learning are also critical areas for exploration. Furthermore, despite some efforts [5, 39], most existing works including the ones covered in this paper, define interpretability and explainability loosely, highlighting the need for greater effort in formalizing these terms to establish clearer and more consistent frameworks. Finally, user studies involving domain experts and end-users will help identify effective interpretability techniques for SDM systems.

6 Analysis and Discussion

Interpretability in sequential decision-making systems is a critical challenge, particularly as these systems are increasingly deployed in high-stakes domains such as healthcare, autonomous systems, and finance. This work has explored a wide range of symbolic representations, each with its own strengths and limitations in terms of interpretability, expressiveness, and applicability. Finite State Machines (FSMs) and Decision Trees, for example, excel in transparency and simplicity but struggle with scalability and continuous reasoning. On the other hand, Temporal Logic provide powerful tools for modeling uncertainty and temporal dynamics but often sacrifice interpretability as system complexity grows. Rule-based systems and Program Synthesis offer modular and human-readable explanations but face challenges in handling uncertainty and scaling to complex environments.

A key insight from this work is that no single representation is universally superior. The choice of representation depends on the specific requirements of the application, including the need for transparency, the complexity of the decision-making process, and the domain-specific constraints. For instance, in domains where accountability and trust are paramount, such as healthcare or legal systems, rule-based systems (and their natural language explanations) may be preferred. In contrast, for dynamic and uncertain environments like robotics or autonomous driving, classic MDPs or neuro-symbolic integration may offer a better balance between performance and interpretability. Another important consideration is the trade-off between interpretability and expressiveness. Highly interpretable representations like FSMs and Decision Trees are often limited in their ability to model complex or continuous systems, while more expressive frameworks like MDPs and Temporal Logic can become difficult to interpret as they scale. This trade-off highlights the need for hybrid approaches that combine the strengths of different representations. For example, neuro-symbolic integration aim to bridge the gap between the high performance of sub-symbolic models and the transparency of symbolic methods.

7 Conclusion

Our comprehensive analysis of symbolic representations in sequential decision-making systems reveals that no single approach offers a universal solution. While simpler representations like FSMs and Decision Trees provide transparency, they struggle with scalability. Conversely, more sophisticated approaches like MDPs and Temporal Logic offer greater expressiveness but become opaque as complexity increases. This inherent trade-off between interpretability and expressiveness suggests the need for hybrid approaches, particularly in domains ranging from healthcare to autonomous systems where both performance and transparency are crucial.

Future work should address several key challenges: developing scalable methods for extracting interpretable symbolic representations, establishing standardized evaluation metrics for sequential interpretability, exploring effective combinations of multiple symbolic representations, and conducting empirical studies on how different stakeholders interact with these representations to inform user-centric design.

The future of interpretable sequential decision-making lies at the intersection of symbolic clarity and modern machine learning. As these systems become prevalent in critical applications, developing frameworks that balance transparency, scalability, and domain requirements becomes paramount. Our analysis provides a foundation for advancing the field toward transparent and trustworthy decision-making systems while maintaining real-world performance.

Acknowledgements

This research was sponsored by the Department of the Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of the Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] Osbert Bastani, Yewen Pu, and Armando Solar-Lezama. 2018. Verifiable reinforcement learning via policy extraction. In *Proc. NeurIPS*.

[2] Leo Breiman, Jerome Friedman, R.A. Olshen, and Charles J. Stone. 1984. *Classification and Regression Trees*. Routledge.

[3] Alberto Camacho and Sheila A. McIlraith. 2019. Learning interpretable models expressed in linear temporal logic. *Proc. ICAPS*.

[4] You Cao, Zhijie Zhou, Changhua Hu, Wei He, and Shuaiwen Tang. 2021. On the interpretability of belief rule-based expert systems. *IEEE Transactions on Fuzzy Systems*, 29, 11, 3489–3503.

[5] Tathagata Chakraborti, Anagha Kulkarni, Sarath Sreedharan, David E Smith, and Subbarao Kambhampati. 2019. Explicability? legibility? predictability? transparency? privacy? security? the emerging landscape of interpretable agent behavior. In *Proc. ICAPS*.

[6] Dillon Z. Chen, Pulkit Verma, Siddharth Srivastava, Michael Katz, and Sylvie Thiébaux. 2025. AI Planning: A primer and survey (preliminary report). In *AAAI 2025 PRL Workshop*.

[7] Glen Chou, Necmiye Ozay, and Dmitry Berenson. 2022. Learning temporal logic formulas from suboptimal demonstrations: theory and experiments. *Auton. Robots*, 46, 1, (Jan. 2022), 149–174.

[8] Vinícius G. Costa, Jorge Pérez-Aracil, Sancho Salcedo-Sanz, and Carlos E. Pedreira. 2024. Evolving interpretable decision trees for reinforcement learning. *Artificial Intelligence*, 327, 104057.

[9] Mohamad H Danesh, Anurag Koul, Alan Fern, and Saeed Khorram. 2021. Re-understanding finite-state representations of recurrent policy networks. In *Proc. ICML*.

[10] Jonathan DeCastro, Karen Leung, Nikos Aréchiga, and Marco Pavone. 2020. Interpretable policies from formally-specified temporal properties. In *Proc. ITSC*.

[11] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv*. https://arxiv.org/abs/1702.08608.

[12] Francisco Elizalde, Luis Enrique Sucar, Alberto Reyes, and Pablo Debuen. 2007. An MDP approach for explanation generation. In *AAAI 2007 ExaCt Workshop*.

[13] Peter Graf and Patrick Emami. 2024. Three pathways to neurosymbolic reinforcement learning with interpretable model and policy networks. *arXiv preprint arXiv:2402.05307*.

[14] Christian Albert Hammerschmidt, Sicco Verwer, Qin Lin, and Radu State. 2016. Interpreting finite automata for sequential data. In *NIPS 2016 IMLCS Workshop*.

[15] Daniel Hein, Steffen Udluft, and Thomas A. Runkler. 2018. Interpretable policies for reinforcement learning by genetic programming. *Engineering Applications of Artificial Intelligence*, 76, 158–169.

[16] Bo-Jian Hou and Zhi-Hua Zhou. 2020. Learning with interpretable structure from gated RNN. *IEEE Transactions on Neural Networks and Learning Systems*, 31, 7, 2267–2279.

[17] Weiche Hsieh et al. 2024. *A Comprehensive Guide to Explainable AI: From Classical Models to LLMs*. arXiv. arXiv: 2412.00800.

[18] Isabelle Hurley, Rohan R Paleja, Ashley Suh, Jaime Daniel Pena, and Ho Chit Siu. 2024. STL: Still tricky logic (for system validation, even when showing your work). In *Proc. NeurIPS*.

[19] Rushang Karia, Pulkit Verma, Gaurav Vipat, and Siddharth Srivastava. 2024. Epistemic exploration for generalizable planning and learning in non-stationary settings. In *Proc. ICAPS*.

[20] Omar Khan, Pascal Poupart, and James Black. 2009. Minimal sufficient explanations for factored markov decision processes.

[21] Anagha Kulkarni, Sarath Sreedharan, Sarah Keren, Tathagata Chakraborti, David E Smith, and Subbarao Kambhampati. 2020. Designing environments conducive to interpretable robot behavior. In *Proc. IROS*. IEEE.

[22] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, and David Madigan. 2015. Interpretable classifiers using rules and bayesian analysis: building a better stroke prediction model. *The Annals of Applied Statistics*, 9, 3, 1350.

[23] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Proc. NeurIPS*.

[24] Daoming Lyu, Fangkai Yang, Hugh Kwon, Wen Dong, Levent Yilmaz, and Bo Liu. 2021. TDM: Trustworthy decision-making via interpretability enhancement. *IEEE Trans. on Emerging Topics in Computational Intelligence*, 6, 3, 450–461.

[25] Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. 2020. Explainable reinforcement learning through a causal lens. In *Proc. AAAI*.

[26] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, A. Ram, Manuela Veloso, Daniel S. Weld, and David Wilkins. 1998. PDDL – The Planning Domain Definition Language. Tech. rep. CVC TR-98-003/DCS TR-1165. Yale Center for Computational Vision and Control.

[27] Eric J Michaud et al. 2024. Opening the AI black box: Program synthesis via mechanistic interpretability. *arXiv preprint arXiv:2402.05110*.

[28] Christoph Molnar. 2022. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. A Guide for Making Black Box Models Explainable*. (2nd ed.). ISBN: 979-8411463330. https://christophm.github.io/interpretable-ml-book.

[29] Samer B. Nashed, Saaduddin Mahmud, Claudia V. Goldman, and Shlomo Zilberstein. 2023. Causal explanations for sequential decision making under uncertainty. In *Proc. AAMAS*.

[30] Rashmeet Kaur Nayyar, Pulkit Verma, and Siddharth Srivastava. 2022. Differential assessment of black-box AI agents. In *Proc. AAAI*.

[31] Rohan Paleja et al. 2023. Interpretable reinforcement learning for robotics and continuous control. *arXiv preprint arXiv:2311.10041*.

[32] Judea Pearl. 2000. *Causality: Models, Reasoning, and Inference*. Cambridge University Press.

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

[34] Rajarshi Roy, Dana Fisman, and Daniel Neider. 2021. Learning interpretable models in the property specification language. In *Proc. IJCAI*.

[35] Rajarshi Roy, Jean-Raphaël Gaglione, Nasim Baharisangari, Daniel Neider, Zhe Xu, and Ufuk Topcu. 2023. Learning interpretable temporal properties from positive examples only. In *Proc. AAAI*.

[36] Naman Shah, Jayesh Nagpal, Pulkit Verma, and Siddharth Srivastava. 2024. From reals to logic and back: inventing symbolic vocabularies, actions and models for planning from raw data. *arXiv preprint arXiv:2402.11871*.

[37] Andrew Silva, Matthew Gombolay, Taylor Killian, Ivan Jimenez, and Sung-Hyun Son. 2020. Optimization methods for interpretable differentiable decision trees applied to reinforcement learning. In *Proc. AISTATS*.

[38] Ho Chit Siu, Kevin Leahy, and Makai Mann. 2023. STL: Surprisingly tricky logic (for system validation). In *Proc. IROS*.

[39] Sarath Sreedharan, Anagha Kulkarni, David Smith, and Subbarao Kambhampati. 2021. A unifying bayesian formulation of measures of interpretability in human-ai interaction. In *Proc. IJCAI*.

[40] Pradyumna Tambwekar and Matthew Gombolay. 2024. Towards reconciling usability and usefulness of policy explanations for sequential decision-making systems. *Frontiers in Robotics and AI*, 11, 1375490.

[41] Dweep Trivedi, Jesse Zhang, Shao-Hua Sun, and Joseph J Lim. 2021. Learning to synthesize programs as interpretable and generalizable policies. In *Proc. NeurIPS*.

[42] Jasper van der Waa, Jurriaan van Diggelen, Karel van den Bosch, and Mark Neerincx. 2018. Contrastive explanations for reinforcement learning in terms of expected consequences. In *IJCAI XAI Workshop*.

[43] Abhinav Verma, Vijayaraghavan Murali, Rishabh Singh, Pushmeet Kohli, and Swarat Chaudhuri. 2018. Programmatically interpretable reinforcement learning. In *Proc. ICML*.

[44] Pulkit Verma, Rushang Karia, and Siddharth Srivastava. 2023. Autonomous capability assessment of sequential decision-making systems in stochastic settings. In *Proc. NeurIPS*.

[45] Pulkit Verma, Rushang Karia, Gaurav Vipat, Anmol Gupta, and Siddharth Srivastava. 2023. Learning AI-system capabilities under stochasticity. In *NeurIPS 2023 GenPlan Workshop*.

[46] Pulkit Verma, Shashank Rao Marpally, and Siddharth Srivastava. 2021. Asking the Right Questions: Learning Interpretable Action Models Through Query Answering. In *Proc. AAAI*.

[47] Pulkit Verma, Shashank Rao Marpally, and Siddharth Srivastava. 2022. Discovering user-interpretable capabilities of black-box planning agents. In *Proc. KR*.

[48] Pulkit Verma and Siddharth Srivastava. 2024. Learning Causally Accurate Models for Autonomous Assessment of Deterministic Black-Box Agents. Tech. rep. TR-ASUSCAI-2024-001.

[49] Zishen Wan et al. 2024. Towards cognitive AI systems: a survey and prospective on neuro-symbolic AI. In *MLSys 2023 Workshop on Systems for Next-Gen AI Paradigms*.

[50] Tao Wang, Xiangwei Zheng, Lifeng Zhang, Zhen Cui, and Chunyan Xu. 2023. A graph-based interpretability method for deep neural networks. *Neurocomputing*, 555, 126651.

[51] Zizhao Wang, Caroline Wang, Xuesu Xiao, Yuke Zhu, and Peter Stone. 2024. Building minimal and reusable causal state abstractions for reinforcement learning. In *Proc. AAAI*.

[52] Zizhao Wang, Xuesu Xiao, Zifan Xu, Yuke Zhu, and Peter Stone. 2022. Causal dynamics learning for task-independent state abstraction. In *Proc. ICML*.

[53] Biao Xu and Guanci Yang. 2025. Interpretability research of deep learning: a literature survey. *Information Fusion*, 115, 102721.

[54] Fan Yang, Kai He, Linxiao Yang, Hongxia Du, Jingbang Yang, Bo Yang, and Liang Sun. 2021. Learning interpretable decision rule sets: a submodular optimization approach. In *Proc. NeurIPS*.

[55] Tianyi Zhang, Zhiyang Chen, Yuanli Zhu, Priyan Vaithilingam, Xinyu Wang, and Elena L. Glassman. 2021. Interpretable program synthesis. In *Proc. CHI*.

Received 12 February 2025; revised 26 February 2025; accepted