Università degli Studi della Tuscia di Viterbo

Dipartimento di Economia, Ingegneria, Società e Impresa

Corso di Dottorato di Ricerca in

ENGINEERING FOR ENERGY AND ENVIRONMENT

XXXVII ciclo

WHY UNDERSTANDING AI MATTERS: EXPLAINABLE REINFORCEMENT LEARNING IN HEALTHCARE FOR TYPE 1 DIABETES MANAGEMENT

Settore scientifico-disciplinare
IINF/05

Tesi di dottorato di:
Dott. Daniele Melloni
Firma

Tutore
Dott. Andrea Zingoni
Firma

Coordinatore del corso
Prof. A. L. Facci
Firma

Co-tutori
Dott. Juri Taborri
Firma

Prof. Giuseppe Calabrò
Firma

A.A. 2023/24 (A.Y. 2023/24)

Why Understanding AI Matters: Explainable Reinforcement Learning in Healthcare for Type 1 Diabetes Management

Daniele Melloni
Department of Economics, Engineering, Society
and Business Organization, University of Tuscia

September 19, 2025

Abstract

As artificial intelligence (AI) systems become increasingly prominent in everyday scenarios, the need for trustworthiness, accountability, and transparency becomes paramount for both current and future applications. This is especially true in high-stakes domains such as healthcare, where manual intervention is progressively replaced by autonomous agents and automated reasoning. In this context, the demand for transparent, reliable, and clinically meaningful decision-making is more urgent than ever.

This thesis explores the intersection of explainable AI and reinforcement learning (RL), referred to as eXplainable Reinforcement Learning (XRL), with a particular focus on the management of Type 1 Diabetes Mellitus (T1DM). The work is driven by the need to ensure interpretability and transparency in autonomous agents operating in complex clinical environments.

The research moves from foundational analysis to applied contributions. It begins with a comprehensive literature review of XRL, mapping the conceptual landscape and identifying key limitations in existing methods. It then focuses on the development of user-centered and model-agnostic explainability techniques, aimed at enhancing human understanding of agent behavior for both lay and expert audiences. To this end, methods such as reward-difference analysis, contrastive explanations, and post-hoc Markov abstractions are explored.

Building on these, a novel approach is introduced to interpret the internal mechanisms of deep RL agents. This method combines Shapley-based Q-value attributions from Deep Q-Networks (DQNs) with structural causal models (SCMs), enabling a principled analysis of decision-making dynamics and exposing the limitations of attribution-based explanations in capturing causality.

Through this multifaceted exploration, the thesis contributes a coherent framework for applying XRL in healthcare, grounded in the practical and clinically relevant use case of T1DM. By unifying theoretical insight with methodological innovation, it advances the broader goal of making AI (in particular, RL) more interpretable, robust, and aligned with real-world clinical needs.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Andrea Zingoni, for his invaluable guidance, constructive feedback, and continuous support throughout the course of this research. His expertise and mentorship have been instrumental in shaping the direction and depth of this work.

I also wish to thank Juri Taborri and Giuseppe Calabrò, whose insights and collaboration have significantly enriched the research outcomes.

I am grateful to the members of my doctoral committee for their time, feedback, and thoughtful evaluation of my work.

I would like to acknowledge the contributions of colleagues and collaborators within the University, whose discussions and support were deeply appreciated.

Deep and grateful thanks are given to Francesca, my spouse, which not only did she bear with me the sacrifices we made together to make this possible, but was always there for me, never profoundly regretting this decision. Not only are my most honest acknowledgments for being so supportive but also my all unconditioned love, as the biggest and most important part of my heart and life.

I extend my appreciation to Francesco, which has been the friendship cornerstone of my life, both in each aspect of it, and in particular in academic pursue. Thank you for never judging me once and to have been always there for me in all the most important moments.

I would like to express special thanks to all my family for the supportive love and affection that has given to me throughout the years. I am certain that you have always been with me and will always be, in all my lifelong decisions.

Furthermore, I would express my gratitude to Mum and Dad. For you Mum, because I know you always did your best to support me and always believed in me. For you Dad, you were here to see me start this journey and maybe will see me ending through another place, my thanks and thoughts to you will always accompany my days.

I am thankful to the “Gruppo del Gatto“ friends, which have shaped my knowledge and my self-awareness throughout all these years together.

I would also thank the many friends of the group coming from “Ferro di Cavallo“ area, which have been always supportive and open to my personal beliefs and goals.

Finally, I extend my appreciation to all my friends and acquaintances, along those who, directly or indirectly, contributed to the successful completion of this doctoral journey.

Contents

Chapter 1

Introduction

As Artificial Intelligence (AI) systems continue to advance and integrate into critical real-world domains, the demand for transparency, trustworthiness, and accountability grows increasingly urgent. This is especially true in high-stakes environments such as healthcare, where decisions influenced or made by AI can directly affect human lives. Despite remarkable progress in machine learning, many state-of-the-art models, particularly deep learning and Reinforcement Learning (RL) agents, often behave as "black boxes", delivering effective solutions without offering insight into the reasoning behind their decisions. This opacity hinders their safe deployment, limits user trust, and complicates validation by domain experts.

Explainability and interpretability have therefore emerged as essential properties for modern AI systems. These concepts enable stakeholders to understand, trust, and effectively interact with AI, as well as diagnose potential errors or biases. In the context of RL, where agents learn through trial-and-error interactions with dynamic environments, this need becomes even more pressing. The sequential nature of decision-making, the use of high-dimensional state and action spaces, and the delayed reward structures further complicate interpretation and transparency.

This thesis addresses the challenge of interpretability in reinforcement learning through the lens of eXplainable Reinforcement Learning (XRL). In this work, various XRL methods in the practical healthcare setting of Type 1 Diabetes Mellitus (T1DM) are applied. The focus is to ultimately bridge the gap between powerful, data-driven RL methods and the stringent requirements for transparency in healthcare. It contributes both methodological insights and practical tools for fostering more explainable and trustworthy AI systems in sensitive application domains.

1.1 Why Explainability Matters

As technology is becoming more and more ubiquitous, the progress of AI as a field has been growing exponentially in the past decades in both industry and research. The need to process big amounts of data [327] and the natural adaptability of AI made this area of information technology the suitable candidate to exploit the maximum potential in applications such as finance [33], healthcare [315], security [23], cyber-physical systems [280] and many other fields such as Internet of Things (IoT) [3], big data [200] and entertainment [142], just to cite a few.

Since the first discoveries and studies in the early developments in the field of AI, however, there has been multiple needs to explain AI mechanisms and decisions. This is due to the fact that, usually, understanding the processes of AI methods results in major cognitive efforts from non-expert users (also known as laypeople or laypersons) [223]. Indeed, oftentimes AI models are perceived as black-box (or opaque) models as their final decisions, from the initial inputs [231], are not trivial to comprehend and predict [214], so there has been a growing interest behind explaining and "graying" the *black-box* nature of such systems [58, 317]. In order to overcome such limitations, the field of eXplainable Artificial Intelligence (XAI) has provided a solid background for many ways to yield useful explanations and has adopted its own taxonomy [106, 59, 21], developed evidences of its benefits [56] as well as being targeted by critiques [154]. Indeed, transparency is crucial to obtain clear statements of AI algorithms in fields such as autonomous driving, autonomous robotic surgery mechanisms [202], and many more. In fact, fields that require an elevated level of responsibility of physical and virtual intervention, such as healthcare, aviation and automated law enforcement, could benefit from the assistance of AI-based tools but require also an high level of transparency in order to explain edge cases and worst-case scenarios that might happen [223].

In this regard, the field of XAI refers to the development and implementation of AI systems that can provide understandable and transparent explanations for their decisions, or that can produce human-readable outputs of their inner workings [1, 106]. It focuses on making the decision-making processes of AI models comprehensible to human users via the generation of readable explanations [107, 307, 10, 282]. Recently, there has been a rise in interpretable and explainable applications of AI and the field has seen an ever-growing success as emerging field [36], as XAI allows the technology to be more trustworthy and accountable [217]. Furthermore, XAI instills confidence in users, as they can verify the reliability and fairness of the system's outputs, leading to increased adoption of such technologies [307, 80]. In addition, XAI helps organizations ensure compliance with regulations and ethical guidelines [194, 310] by enabling them to explain AI decisions [144], especially in the sensitive areas mentioned above.

On the other hand, enhancing explainability may come at the cost of accuracy or efficiency. More complex algorithms and models are often less interpretable, requiring a balance between explainability and performance [10]. Also,

AI systems can be intricate, making it challenging to provide comprehensive explanations that can be easily understood by non-experts. Conversely, increasing interpretability often involves simplifying the models, which may reduce their ability to capture complex patterns and correlations in the data. Nonetheless, evidences in the XAI literature showed that the direction undertaken by past studies is indeed the way to go to establish trustworthiness and transparency in the AI field [60, 56].

As the field of AI has grown very vast over the years, more peculiar subfields such as Deep Learning (DL), Machine Learning (ML) and Reinforcement Learning (RL) may require a dedicated attention from the perspective of explainability. Indeed, approaches of ML, DL and RL may differ substantially across the various studies undertaken over the decades by practitioners and researchers [170, 195, 101]. In particular, the field of RL has shown promising results in many fields thanks to its unsupervised nature that allows this paradigm to be a flexible and powerful solution [123]. Alongside its development, thus, the need for explanation has also grown in the RL field as well, which has attracted particular attention recently in many different tasks [258]. In the light of this, the focus of this work is indeed aimed at exploring the current state of the art behind the field of eXplainable Reinforcement Learning (XRL), extract useful insights about its opportunities as well as underlining current open challenges and limitations.

1.1.1 Reinforcement Learning

Reinforcement Learning (RL) is a branch of AI that focuses on training agents to make sequential decisions in a given environment in order to maximize a specific objective or reward function. Reinforcement learning is applied in diverse fields such as gaming, robotics, autonomous vehicles, recommendation systems, healthcare, finance, Natural Language Processing (NLP), resource management, supply chain, game development, agriculture, drug discovery, adaptive user interfaces, control systems, education, cybersecurity, natural resource management, game AI, sports analytics, and social sciences. It draws inspiration from behavioral psychology of humans' and animals' instinctive behaviour and operates under the framework of trial-and-error learning [258].

A RL problem can be mathematically defined as a Markov Decision Process (MDP). A MDP consists of a tuple $(S, \mathcal{A}, \mathcal{P}, R, \gamma)$, where:

- $\mathcal{S}$ is the set of possible states in the environment,
- $\mathcal{A}$ is the set of possible actions that the agent can take,
- P is the state transition probability function, which defines the probability of transitioning from state s to state s' when a given action $a \in \mathcal{A}$ is taken: $P(s'|s, a)$,
- R is the reward function that maps a state-action pair to a scalar reward: $R : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$,

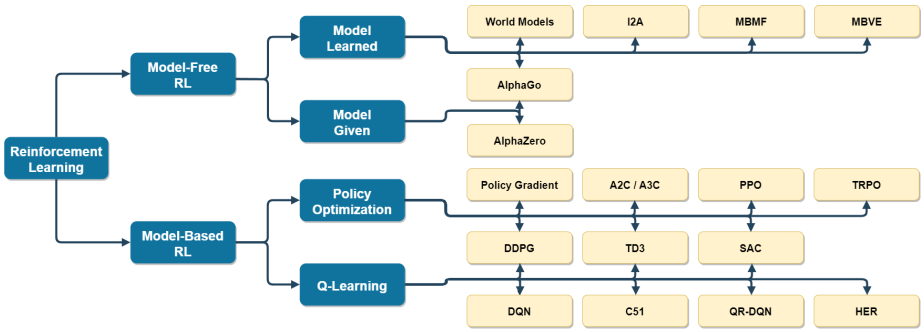


Figure 1.1: An example of taxonomy of RL algorithms

- γ is the discount factor that determines the importance of future rewards relative to immediate rewards.

A typical RL problem includes multiple components.

1. **Policy**: a policy π is a mapping from states to probabilities over actions, defined as: $\pi(a|s) = P(A_t = a|S_t = s)$ and defines the agent's behavior or strategy for selecting actions in different states.

2. **Value function**: a function that is used to evaluate the quality or utility of being in a particular state or taking a particular action. It represents the expected cumulative reward that an agent can achieve from a given state or state-action pair, following a specific policy.

There are two types of value functions, namely the **state value function**, which represents the expected cumulative reward from a given state s by following policy π ($v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s]$) and the **action value function** that represents the expected cumulative reward from taking action a in state s by following policy π ($q_\pi(s, a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a]$), where G_t is the discounted cumulative reward from time step t onwards. The main goal of RL is to reach (often by approximations of the functions, due to computability issues) the optimal value functions denoted as $V^*(s)$ and $Q^*(s, a)$, respectively. Various RL algorithms, such as Q-learning, SARSA, and Policy Gradient methods, are designed to solve the RL problem and converge to the optimal solutions [258]. An example of RL algorithms taxonomy is given in Fig. 1.1.

1.1.2 Explainable RL (XRL)

Explainable Reinforcement Learning (XRL) is an emerging field that aims to provide transparent and interpretable decision-making processes for RL agents. It combines concepts from both RL and interpretable ML to offer insights into how and why an agent makes specific decisions in its environment. Due to its

successes in fields such as robotics, game playing, recommendation systems, finance, and healthcare, there has been a need to thrive explainability onward particularly in the field of RL. Indeed, studies such as [7] overviews the capabilities and needs for the ongoing research behind XRL.

Specifically, motivations behind driving research of XRL, amongs others, are:

1. **trust and safety**, as AI systems become more prevalent in critical civilian applications, such as law enforcement and automated healthcare, understanding the decision-making process becomes crucial for ensuring trust and safety and XRL, in fact, helps build confidence in AI by making its actions more comprehensible to humans;

2. **regulatory compliance**, as in certain industries, regulations mandate the use of explainable AI, thus XRL helps companies adhere to these requirements while still leveraging the benefits of RL;

3. **human collaboration**, where XRL facilitates collaboration between AI agents and humans, enabling users to provide more effective guidance and feedback to the learning process.

In summary, Explainable Reinforcement Learning holds promise in enabling trust, understanding, and collaboration between humans and AI agents. However, addressing its limitations and exploring potential applications are essential for its widespread adoption and impact [130, 154]. In the current literature, multiple terms exists to express the objective of transparency, which are mainly explainability, explicability and interpretability. Throughout the scope of this work, these terms are used interchangeably to express the same process of obtaining insights of RL models and methods. That is, the effort to provide transparency and generating explanations that could be understood by the most widespread degree of cognitive effort possible for the direct final user.

1.2 Surveys on XRL

Even if the field of XRL research is still relatively new, it has been previously explored to a varying extent in different RL domains and across different tasks [218]. In this regard, the main contributions of the work here presented are manifold:

- to overview the recent advances and current state of the art on XAI applied in RL, that is XRL;

- to propose a new taxonomy of algorithms adopted in the field;

- to highlight the opportunities and challenges that XRL is still facing to date;

- to propose possible solutions to overcome such limitations.

In addition, in comparison with the other surveys listed below, this work focuses in a narrow fashion on XRL, other than considering pure RL or XAI, as proposed, for example, in preliminary works such as [218, 102] and [318]. Indeed, the studies included in this work are, for the greater part, focused solely on XRL methods and, as such, contribute to form the review hereby presented. For the sake of completeness, in this section, current surveys of XRL literature are presented and their strong and weak points are exposed in order to better assess the true potential of this field of research as well as to discuss aspects that could be improved [218, 81, 50, 300, 102, 287, 181, 318, 220, 4].

Among the most prominent literature surveys on XRL, important works that were found to be notable and worth mentioning are the following:

- In [218], explanations are divided in the forms of: local or global, based on the explanation range available to the methodologies proposed; intrinsic and post-hoc, based on where the explanations are generated. Even if it follows the XAI taxonomy, this survey doesn't look for a comprehensive analysis of the current XRL state of the art.

- In [81], explainability and interpretability are considered to be different concepts. The division is threefold for XRL works such as explainability of inputs, transitions functions (or models) and dynamics of environments. It could be of help to better generalize XRL works that don't fall into such categories or that share common traits with different classes of explainability.

- In [50], XRL interpretation by means of causal representations were introduced. The theoretical structure of this framework is rigorously stated and coherent (e.g. well-defined concepts of perceptions, expectations, events, actions, etc.). There is more need, however, to cover sources of particular events focused solely on perceptions and actions, cutting the scope of the current XRL state of the art to a limited portion of the proposed framework.

- In [300], the focus was on visual explanations of XRL which is a limited subset of the whole XRL framework. It is stated how the visual representation aids for the interpretation of outcomes, but visual representations solely are not enough to describe the entire XRL groundwork.

- In [102], authors borrow the concepts of post-hoc explainability and intrinsic explainability from [10] and solely made the distinction of XRL methods between the two. For this reason it is considered a preliminary work as the state of the art has greatly evolved and became multi-faceted over recent times. Such classification is somewhat simplified for XRL works to be divided only in two categories, as done in their work.

- In [287], the emphasis is put onto creating a symbolic framework of XRL and divides the work done in this field by agent and goal preference. No

further differentiation is given, thus posing an inherent limit to the classification of the actual state of the field.

- In [181] is a survey containing works from the field divided by a broad and general classification. It is, thus, to be considered as a valuable input for a starting practitioner but also limits itself to be a short statement of the state of the art behind XRL.
- The study in [318] is considered to be a preliminary assessment on explainability and XRL. It does, however, not focus on the actual implications about the field of XRL, but rather puts the basis for the areas of interest for applications of RL and the applicability of explainability to it.
- The work in [220] offers an overview of the XRL research field putting in place what has been done and what are still the open challenges and opportunites for this topic. The work, however, is structured differently from the one proposed here and some classes of works fall into categories that are not exactly representative of the advances of recent years.
- Lastly, [4] proposed an overview of the methods of XRL with the standard XAI taxonomy applied to the field of XRL. However, as a preliminary work, it lacks the applications and recent advances that has been growing exponentially over the past few years.

In light of literature evidence present in the various XRL studies and the reviews included above, a thorough and global methodology of generating a generic form of explanation, given a specific RL task, is almost totally lacking. Two main reasons behind this issue could be the lack of unification for different XRL paradigms, albeit some proposals have been made in literature, as well as standard evaluation criteria (or metrics) to coherently assess the efficacy and the quality of the generated explanations. Even if some of the previous work has the merit of clearly presenting some of the most used techniques of XRL, it appears noticeable that more effort should be devolved into addressing these principal issues. The taxonomy previously proposed is not fully aligned with the RL framework itself as it is, on average, inherited from XAI and, thus, drives away from the inner nature of the RL settings. Moreover, the existing surveys don't stress enough how human participation in XRL is a crucial element in assessing, evaluating, stabilizing and standardizing the quality and efficacy of explanations generated from the proposed methodologies. It is, indeed, of paramount importance to provide a human-in-the-loop verification of explanations given as it is the very main purpose of explanations, that is, directed to the end users. This iterative process not only enhances the reliability of the provided information but also ensures a more user-centric approach, acknowledging the dynamic nature of user needs and perceptions in this particular landscape.

In order to try to fill these gaps, this work is focused on dividing XRL works into components of partial explainability and to categorize, differently than before, the actual literature. The taxonomy proposed in the following sections is focused on dividing between practical XRL methodologies and its

applications. Based on the type of explanations generated, these categories are built by aiming at giving a novel categorization by considering the different specific mathematical models used in the numerous studies included in this work. This process results in a fine-grained division of XRL works that is precise with respect to the nature of RL and XRL, as compared to the already existing surveys. In addition, a whole separate section is devolved to the applications of XRL methods in different domains. This division was made in order to better classify, in terms of common background and applied concepts, between methodologies that were mainly exploratory (methodlogies) and studies which employ them in a practical real-word scenario (applications). In the next section, the opportunities and, consequently, what fuels the interest in XRL research, are presented and discussed.

1.3 Opportunities of XRL

The whole motivation behind this work is backed by what XRL offers both in terms of opportunities of developments and surfaces of applications. This section discusses the benefits of adopting and further improving XRL methodologies and what thrives the field in achieving impactful results. Indeed, XRL is still a promising area and various aspects of it still remain unsolved at the moment of writing this work: key components, such as actively involving and evaluating human subjects in the studies, remain unplumbed. The opportunities that XRL still offers to date are discussed below.

Complete Interpretability. Despite recent literature efforts and its sensitivity to the needs of global interpretability, a framework that could be defined as *complete* or *full* from an interpretable point of view is still lacking. Complete transparency and explanation are directly related to the complexity of the task and, thus, to the complexity of the methods proposed. Indeed, single contributions of XRL literature can introduce partial interpretability of RL while, for complete trust and explainability, all components of a given XRL model must be, by definition, explainable. Formalizing a thorough model that explains in all of its parts the process of a RL solution offers the possibility of further engaging third party entities into adopting RL, thus leaving major possibilities to XRL practitioners.

Metrics, Evaluations and Standards. XRL, as it is frequently also the case in XAI, suffers from the foundation of shared metrics and evaluations [104]. Indeed, current literature didn't yet establish a common ground amongst XRL that formalizes the evaluation of efficiency or goodness of proposed explanations. Studies don't stress enough how good effectively is a proposed explainable solution w.r.t. the plain RL method (i.e. there is no established way to evaluate how much transparent a method is compared to others). Furthermore, the quality of explanations should also be taken into account as the very final scope of the XRL paradigm is the end user. There is still need for the field for enabling a proper way of evaluation of given explanations based on the degree of quality it

can offer to laypeople.

Ethical AI. By making RL systems more transparent, XRL can help identify and mitigate biases and ethical concerns embedded in the learning process. Used in conjunction with evaluation metrics, protocols and standards could serve in defining precise targets that concern ethical, legal and usability aspects. Those account for the eployment in both research and industry fields and can drive to a greater extent the interest of practitioners in the field. Ethical concerns are one of the most argumented reasons behind the adoption of XAI and the same applies to XRL, leaving a window of opportunity to be addressed in further research.

Human Studies. As, also discussed previously, many of the current studies behind XRL take little to no account to human-in-the-loop process. This is, however, a key element for constructing appropriate trust and draw precise conclusions about the quality of the work [197]. Oftentimes human studies are provided with low individuals or with individuals recruited through virtual platforms such as Amazon Mechanical Turk which hinders the real benefits of adopting chosen individuals [31]. In fact, direct human feedback offers the possibility of modifying existing methodologies or enhancing parts of already established contribution, thus leaving margins of great interest from the psychological and cognitive effort required to successfully engage with the explanations generated.

1.4 XRL in Healthcare: Type 1 Diabetes Mellitus

Type 1 Diabetes (T1D) is a chronic autoimmune condition where the pancreas produces little to no insulin, a hormone that is crucial for regulating blood sugar levels. Unlike type 2 diabetes, which is often associated with lifestyle factors (sedentary, smoke, diet etc.), T1D is not preventable and typically develops in childhood or adolescence. T1D management is thus an essential requirement for affected individuals due to a variety of factors. First, it ensures blood sugar control, as without proper management, blood sugar levels can fluctuate dramatically, leading to hyperglycemia (high blood sugar) or hypoglycemia (low blood sugar) [45]. Consequently, management of T1D, through devices such as Continuous Glucose Monitors (CGMs) [1], is essential in improving the quality of life and can lead to healthier and more active individuals.

The widespread adoption in different sectors and recent progress in Artificial Intelligence (AI) has led this field to be widely applied to a vast range of different domains, including T1D [116]. Amongst all, one of the most successful paradigm of AI in T1D management is Reinforcement Learning (RL) [179]. Unlike supervised learning, RL operates without labeled data and relies

[1] A CGM is a device that continuously monitors the Blood Glucose (BG) level at predefined intervals (e.g. 3 minutes). It is widely used in T1D control because it helps at automating the process of monitoring BG levels.

on interaction with the environment to learn optimal actions over time [257]. This makes employed RL algorithms ideal to develop personalized insulin dosing strategies tailored to individual patients' glucose dynamics and lifestyle factors [266]. By continuously interacting with the patient's physiological data and observing the effects of insulin dosages, RL agents can learn to adapt dosing regimens to maintain optimal Blood Glucose (BG) levels while minimizing the risk of hypoglycemia or hyperglycemia.

However, the lack of transparency and interpretability can lead to distrust and skepticism towards automated systems, especially in critical domains where human oversight and accountability are crucial. By this issue it arises the need for eXplainable Artificial Intelligence (XAI) techniques in healthcare and, despite the paramount importance of such applications, XAI and eXplainable RL (XRL) methods applied to healthcare remain still rarely explored [35]. Additionally, in problems such as T1D management, model's opaqueness might hinder the possibility to discover more about the agent's dynamics as its complexity is very limited in terms of explanations generation. In this regard, the field of XRL [218] helps at better understanding and better interpreting what are the steps behind an automated decision-making process.

At the current state of the art, various frameworks of XRL algorithms exist. Examples are the use of causal models such as in [166], where the authors utilize causal relationships between variables of interest. The explanations are given, starting from agent's visual inputs, by highlighting the area of interest for the given tasks. Other XRL examples include the use of imitation learning such as the post-hoc technique of Programmatically Interpretable RL [281], the use of symbolic reasoning, in applications such as [77], where the authors propose an end-to-end learning that builds a set of easy symbolic rules, the use of decision trees, as in [155], where Linear Model U-Trees are used to generate an explanation detached to the learning agent, or the use of attention mechanisms, as in [192].

1.5 Problem Formulation

The task of achieving closed-loop control of T1DM using reinforcement learning (RL) can be modeled as a discrete-time, infinite-horizon Markov Decision Process (MDP), represented by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$. In this framework, $\mathcal{S}$ denotes the space of all possible system states, $\mathcal{A}$ comprises the set of available actions, $\mathcal{P}$ defines the probabilistic dynamics of transitions between states, $\mathcal{R}$ is the function that assigns rewards based on actions taken, and $\gamma \in [0, 1]$ represents the discount factor, which weighs future rewards during the learning process. Within this formulation, managing T1DM is viewed through the lens of this MDP structure: a state $s \in \mathcal{S}$ corresponds to a snapshot of continuous glucose monitor (CGM) readings, and an action $a_t \in \mathcal{A}$ reflects the decision made by the artificial pancreas (e.g., insulin delivery) at time step t. Upon executing an action (such as administering basal or bolus insulin), the system transitions to a new state $s' \in \mathcal{S}$, guided by the transition model $\mathcal{P}$. Simul-

taneously, the agent receives a scalar reward $r_t = \mathcal{R}(s_t, a_t)$ that quantifies the immediate outcome of its decision.

The behavior of the agent is governed by a policy π, which is optimized to maximize the cumulative expected reward over time, discounted by the factor γ to prioritize short-term vs. long-term gains. From this basis an action-value function $Q^\pi(s, a)$ can be defined as:

$$Q^\pi(s,a) = \mathbb{E}[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'} \,|s_t = s, a_t = a, \pi]. \tag{1.1}$$

Among the various methods of solving T1DM management via RL agents, one of the most promising one is the use of a Deep Q-Network (DQN) [265]. DQN was first introduced in [189], where a deep neural network is employed to approximate $Q(s, a)$ by delivering a function $Q(s, a; \theta)$, which relies on network's parameters θ.

In this context, the Loss function is defined as:

$$L_i(\theta_i) = E_{s,a,r,s'}[(y_i^{DQN} - Q(s,a;\theta_i))^2], \tag{1.2}$$

where y_i^{DQN} is the target Q-network parameterized by θ^-:

$$y_i^{DQN} = [r + \gamma \max_{a'} Q(s', a'; \theta^-)]. \tag{1.3}$$

Consequently, a gradient descent technique is utilized to update the parametrs by minimizing the square loss of Eq. 4.11. The Q-learning update rule then is defined as follows:

$$\nabla_{\theta_i} L_i(\theta_i) = E_{s,a,r,s'}[(r + \gamma \max_{a'} Q(s,a;\theta_{1-i}) - Q(s,a;\theta_i))\, \nabla_{\theta_i} Q(s,a;\theta_i)]. \tag{1.4}$$

The scenarios used are the ones included in the FDA-approved UVA/Padova in-silico T1DM simulator [171] and, in order to achieve closed-loop T1DM control, this work leverages the capabilities of the DQNs.

In particular, the mechanics of the simulator include glucose kinetics, insulin kinetics, meal model and insulin action. Glucose kinetics handles the glucose appearance, utilization and distribution by following the formulas

$$\frac{dQ_1}{dt} = F_{01}^C - x_1 \cdot Q_1 + k_{12} \cdot Q_2 + R_a(t) - EGP(t) \tag{1.5}$$

$$\frac{dQ_2}{dt} = x_1 \cdot Q_1 - k_{12} \cdot Q_2 \tag{1.6}$$

where F_{01}^C is non-insulin dependent glucose utilization, x_1 is the insulin action on glucose disappearance, $R_a(t)$ is the rate of glucose appearance from meals, $EGP(t)$ is the endogenous glucose production, Q_1 and Q_2 are the plasma and remote glucose compartments and t is the time variable.

On the other hand, insulin kinetics describe how subcutaneously administered insulin is absorbed and distributed by following:

$$\frac{dS_1}{dt} = k_{a1} \cdot S_1 + u(t) \tag{1.7}$$

$$\frac{dS_2}{dt} = -k_{a2} \cdot S_2 + k_{a1} \cdot S_1 \tag{1.8}$$

$$\frac{dI}{dt} = -k_e \cdot I + \frac{S_2}{V_I} \tag{1.9}$$

where k_{a1} and k_{a2} are the absorption rate constants, k_e is the insulin elimination rate, V_I is the insulin distribution volume, $u(t)$ is the insulin infusion input and S_1, S_2 are the subcutaneous compartments.

Furthermore, the glucose absorption modeled by the meal model describes how glucose from food enters the bloodstream and follows the equations:

$$\frac{dG_1}{dt} = -k_{g1} \cdot G_1 + D(t) \tag{1.10}$$

$$\frac{dG_2}{dt} = -k_{g2} \cdot G_2 + k_{g1} \cdot G_1 \tag{1.11}$$

$$R_a(t) = f \cdot k_{g2} \cdot G_2 \tag{1.12}$$

which models the gastric emptying and intestinal absorption G_1, G_2, where $D(t)$ is the meal carbohydrate intake function and f is the bioavailability factor.

Lastly, the insulin action is modeled by compartments that delay and smooth insulin's effect on glucose uptake and suppression of glucose production:

$$\frac{dx_1}{dt} = -p_2 \cdot x_1 + p_3 \cdot (I - I_b) \tag{1.13}$$

where x_1 is the insulin action, p_2, p_3 are the sensitivity and delay parameters respectively and I_b is the basal insulin level. By these equations all the 30 in-silico patients are then able to provide variability of the T1D kinetics and mechanics. In the following subsection an explanation of how to achieve better post-hoc interpretability of such mechanics is given.

Chapter 2

Theoretical Background

2.1 Methodologies of XRL

The methodologies of XRL represent the practical means by which explanations are generated and provided to the end users. In this section, the works in XRL are grouped by types of explainability mechanisms and, for each subsection, a brief description and the works in the relative methodology are given. The methodologies in this work are subdivided in the following subsections that will be treated in-depth in the following parts of this work:

- Subsect. 2.1.1 presents works that require visual input or produce visual feedbacks. In particular:
  - Subsect. 2.1.1 presents Decision Trees (DTs) as means of natural graphical explanations;
  - Subsect. 2.1.1 poses particular attention to the adoption saliency maps built throughout RL methods;
  - Subsect. 2.1.1 groups any other visual methodology that is not included in other subsections;
- Subsect. 2.1.2 includes works of symbolic methodologies of generating explanations;
- Subsect. 2.1.3 introduces causal models and methods of game theory applied to XRL;
- Subsect. 2.1.4 explores the field of contrastive explanations;
- Subsect. 2.1.5 groups the works that use Imitation Learning (IL) for explanation generation;
- Subsect. 2.1.6 focuses on means of explainability through the use of attention mechanisms;

- Subsect. 2.1.7 overviews the explanations based on textual and Natural Language Processing (NLP) approaches;
- Subsect. 2.1.8, finally, mentions works that do not fall into the previous categories and, thus, make a cluster on their own.

To facilitate the interested reader in how this work groups the papers, the studies included in this review of XRL are represented in Tab. 2.1 and in Fig. 2.1.

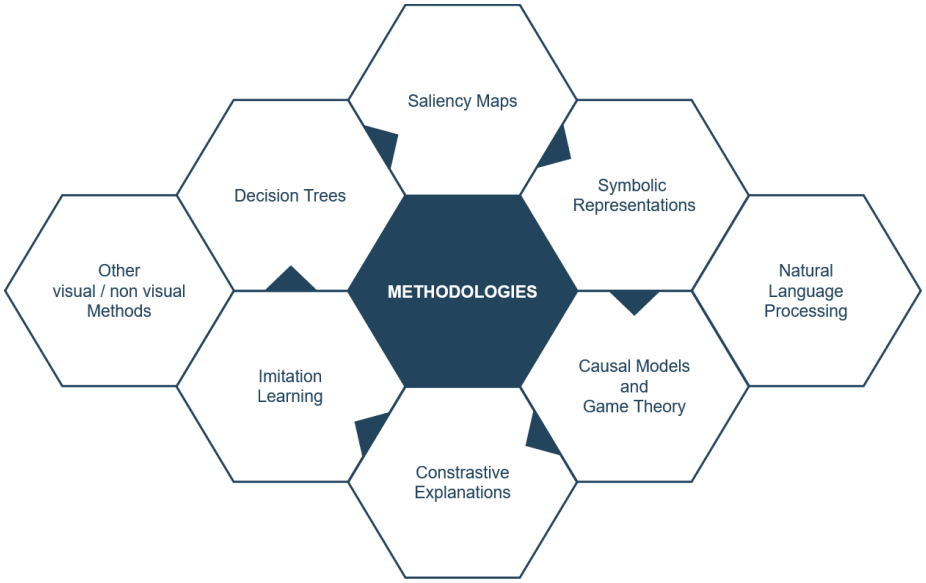


Figure 2.1: A scheme of the various methodologies included in this section. All categories are equivalent in terms of importance.

2.1.1 Visual Explanations

In this subsection, visual explanations are referred to means of explanations where either visual input is processed by the agents or some sort of visual feedback is given to the final user [159]. Different modalities of visual explanations exist in the current literature and range from highlighting parts of the agents' inputs to actual visual frameworks composed of various modules each explaining parts of the learning task. Visual explanations are a powerful tool in that partial explainability of XRL models can be obtained depending on the focus on which part of the model is explained. For example, visualizing the gradient [229], has been proposed as a method to visualize how the gradient changes over time during learning and execution. Dedicated following subsections are devolved to saliency maps and decision trees that are a common way of visual explanations in XRL that differs from other, more generic, visual methods.

Category	**Papers**
Decision Trees (DTs)	[269], [183], [19], [155], [46], [30], [238], [225]
Saliency Maps	[112], [90], [89], [306], [115], [113], [87], [199]
Other visual	[5], [230], [117], [293], [234], [255], [303], [186], [48], [270], [323], [317], [61], [263], [53]
Symbolic Representations	[77], [165], [42], [17], [128], [145]
Causal Models and Game Theory	[254], [319], [296], [193], [288], [52], [147], [201], [180], [166], [233]
Contrastive Explanations	[158], [2], [103], [277], [184], [153], [259], [129], [122], [215], [311], [291], [161], [66]
Imitation Learning	[203], [212], [111], [143], [328], [281], [241]
Attention Mechanisms	[304], [243], [292], [101], [309], [192], [67], [9]
Natural Language Processing (NLP)	[25], [96], [132], [39], [249], [57]
Other Methods	[68], [91], [152], [65], [222], [97], [167], [32], [268], [271], [305], [92], [245], [54], [236], [297], [47], [6], [135], [55], [62], [262], [196], [98], [99], [321]

Table 2.1: Categories and papers included in each category of the various XRL methods included.

Decision Trees

Decision Trees (DTs) are a popular machine learning algorithm used for both classification and regression tasks. They work by recursively partitioning the data into subsets based on the most informative features, creating a tree-like structure where each internal node represents a decision based on a feature, and each leaf node corresponds to a predicted outcome. Here, DTs are used as a form of increased interpretability for RL as they provide a clear and human-friendly structure of how decisions are taken and how discrimination between states is made. Indeed DTs are used in XRL, most commonly, to represent an agent's policy and decision making and are, as such, approximations of the actual policies of the agents.

This immediately comes, and naturally imposes, as discussed in Sect. 1.1, a barrier between the readability of the model and its actual performance. In most of the studies, in fact, there is particular attention devolved into the complexity of the tree that could grow exponentially with the state space. A good DT for explanation should be low in overall node count and represent as accurately as possible the real decisions made by the agent. This is not the case, however, for the vast majority of approaches as DTs have a limited capability in this as they are a simple tool to explain agent's actions. Nonetheless, DTs turned out to be a good approach for generating explanations of XAI models [73], in general, and their applications to the XRL are listed below.

Iterative bounding MDPs (IBMDPs) are introduced in [269], where the authors presented an approach of generating explanations in the form of DTs in hierarchical RL tasks. Authors define *CUSTARD*, a training algorithm that builds DT policies (DTPs) starting from IBMDPs. Empirical results in OpenAI Gym environments show that the tree size in correctly maintained small and reward improvements are highlighted for interpretability purposes. However, no user study is provided thus no human evaluation of the trees generated is provided.

Another approach of DTs for interpretability is [183], wchich introduced two algorithms for the MARL setting: IVIPER and MAVIPER that extend the VIPER [19] algorithm to the multi-agent case. Experiments in both cooperative-competitive and fully cooperative environments, compared with fitted q-iteration and imidation DT baselines, show that these two methodologies are capable of generating high-performing yet explainable policies with the use of DTs.

In [155] Linear Model U-Trees (LMUTs) are introduced, which are an extension of the Continuous U-Tree (CUT) [274], a variant of the classical U-Tree learning for continuous functions in the RL setting. The authors developed a mimic learning framework which constructs a LMUT that imitates a DQNs Q-function which is used to solve the RL problem. Here, a linear model is trained with SGD where the influence facture is obtained via variance reduction, the rule extraction is performed on the tree and, for further interpretation, super-pixels are used to give visual feedback. The work leverages the intrinsic interpretability of the proposed methodology due to the transparent nature of the DTs. LMUTs are more easily understandable in comparison to CUTs as

tend to have fewer leaves and are capable of better generalization by having a linear model at each leaf node, instead of a constant. The authors tested this methodology on three OpenAI Gym tasks [27] Mountain Car, Cart Pole and Flappy Bird.

Similarly, [46] presented a distillation method of RL policies via Soft Decision Trees (SDTs). They used a mimicking model of DTs which tries to explain how the policy operates in its inner features. The authors applied SDTs on the Mario AI benchmark which is a emulator of the classical Super Mario 2D platformer game based on Java so as to make game features available to the agent. The state representation of the surrounding environment of Mario is a 10x10 receptive field which describes the features around the agent with block resolution. The authors, however, didn't put considerable emphasis on the explanability of their methodology as the work limits itself only to evaluate the performances of SDTs compared to DNNs.

In [30], authors proposed an ensamble method of CART [26] trees via gradient boosting of DTs in the cart pole and mountain car environments. The proposed methodology implements a REINFORCE [301] algorithm with a softmax output layer to emulate MDPs policies. Experiments show that, except for the cart pole problem, this ensemble method is capable of achieving an overall good performance. No futher assessment of model interpretability, however, is given in this study and no practical or visual example of the actual explanations is provided.

The use of DTs is predominant also in [238] where an optimization method for explainable RL policies is introduced. The authors propose a Differentiable Decision Tree (DDT) as an interpretable function approximator for the agent's decisions. In the original DDT formulation a sigmoid activation function that smooths the transition between the *TRUE* and *FALSE* evaluations of the nodes is used in place of a discrete decision. In this work, the DDT is converted into a discrete tree by means of an $\arg\max_j(\beta_\eta^j)$ function and it is encoded with a one-hot vector. Furthermore the splitting criterion ϕ_η is divided by the node's weight β_η^j, forming all the discrete splits at each decision node. Also, differentiable rule lists are employed in the form of

$$T_\eta := \begin{cases} y_\eta, & \text{if leaf} \\ \mu_\eta(x) y_{\eta\swarrow} + (1 - \mu_\eta(x)) T_{\eta\searrow}(x) & \text{o/w} \end{cases}$$

keeping the DT discrete while also achieving performant results in RL tasks. The experiments were done in OpenAI Gym environments cart pole and lunar lander, a simulated wildfire tracking problem and a StarCraft II scenario. The agents are trained via PPO, the models are updated via the RMSProp optimizer and the performances are compared against a multilayer perceptron (MLP) baseline. The overall performances of the reduced trees are all above baselines and this shows a remarkably good insight on whether classical DDTs or reduced discrete DTs as employed in this work actually can perform similarly in given scenarios. An interpretability study was given over 15 participants. For each participant a discretized DDT, a discretized decision list and a sample one-hot MLP were

given as models and each model was evaluated in a Likert scale. Authors claim that the null hypothesis on **H1:** *A decision-based classifier is more interpretable than a MLP* and **H2:** *A decision-based classifier is more efficient than a MLP* are both rejected with strong evidence and thus infer that DDTs are indeed a powerful approach towards interpretable RL agents.

The proposed work in [225] introduced the Conservative Q-Improvement (CQI) methodology to explain agents' policies through DTs. To justify splits, the method proposed is conservative in the sense that each split of the DT is done by means of a decreasing threshold that is reset whenever a split occurs. This method allows the tree to have a straight increment in the number of leaf nodes and the whole trees converges towards leaves that are self-explained in that they represent agent's actions. The environment tested is a robot navigation problem which was also made publicly available to the OpenAI Gym framework. Evaluations are made in comparison with the method proposed in [219] and generally exhibits an overall good performance and results in having fewer nodes than the one in [219]. However, other than reducing tree size, the contribution of the work is limited to considering a better interpretability due to both tree size and leaf nodes, which in turn are made of agent's actions. Further work should be provided in order to better assess the goodness of the proposed solution.

Saliency Maps

Saliency maps are a category of visual representation that focuses on generating explainability via t-Stochastic Neighbor Embedding (t-SNE) visualizations [276]. They aim at highlighting parts of the input, or parts of the state space, to show what the agents perceive as meaningful or *salient* information. Different types of saliency maps, for example in conjuction with attention mechanisms, have been proposed and evaluated as XRL explanations. Saliceny maps are amongst the most common methodology to generate explanations as it is a quite a direct method of assessing agents' attentions and provides informative representations of how the agent evaluates meaningful features.

However, such saliency maps could be promising in low-dimensional and *trivial* environments (such as Atari videogames) as one could map the salient regions of the output with the input of the agents. It is discussed, however, whether that such works could offer relevant aid in real-world complex scenarios with by associating raw visual input to agents' behaviours. For example, initially, complex interactions between agents and environments in applications such as autonomous vehicles were not foreseen to be explained by such techniques as they would offer no practical way of explaining agents actions related to raw inputs. Some of the methods are compared and benchmarked in [112], providing a good base for drawing further conclusions. Nevertheless, such methods offer insights on the behaviour of agents and could be further exploited in different domains to showcase their efficacy.

Saliency maps for explanation are generated in [90] in order to verify the binary evaluative feedback hypothesis. To obtain such objective the authors, inspired by previous works such as [34] and [88], introduced *EXPAND (EXPla-*

nation AugmeNted feeDback) which is a method of augmenting the RL agent data with human feedback. This is done by training the agent with a loss defined by $L_F = \lambda_A L_A + \lambda_P L_P + \lambda_V L_V$, where the λ_i are given weights and L_i are predefined losses and A, P, V represent the *advantage*, *policy invariant* and *value invariant* terms respectively. The method is evaluated on a pixel taxi task and four atari games. Human feedback is collected every N_f episodes in form of binary evaluation for the queried trajectories. Results show that this is a promising approach for a human-in-the-loop RL paradigm. Their work was later extended in other works such as [89].

Interpretability through saliency maps is also used in [306], where the policy distillation is obtained via selective Input Gradient Regularization (DIGR). The Loss function is defined by $L_{\text{DIGR}} = \mathbb{E}_{s\sim D}[\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{[0,\infty)}(\lambda - M_{p_i}^t) \times M_{g_i}^\theta] + \alpha\mathbb{E}_{s\sim d_{\pi_\theta}}[D_{\text{KL}}(\pi_t(s)||\pi_\theta(s))]$. The interpretability of this method is presented through the use of adversarial attacks and results in atari environments effectively visualize the important features of the agents.

In [115] salient maps are produced that enable the visualization of agents' behaviours. The method consists of taking an image input, augment the channels of the input with the objects recognised by a dedicated module and then process the whole via a CNN. This procedure, named Object-sensitive Deep RL uses different networks as baselines such as the classical DQN [189], Double Deep-Q-Networks (DDQNs) [278], dueling Deep-Q-Networks [299] and Advanced Actor-Critic model (A3C) [187]. Normally, to obtain a saliency map, computations must be made in order to apply the first-order Taylor expansion $Q(s,a) \approx w^T s + b$, where w is the derivative of $Q(s,a)$ w.r.t. the state image s at state s_0. Then, the overall pixel saliency map is obtained through $w = \frac{\partial Q(s,a)}{\partial s}|s_0$, for each pixel. The authors here, however, introduce the notion of object saliency map which modifies the computation of w into $w = Q(s,a) - Q(s_o,a)$ where s_o is a new state formed from a given state s with masked out objects if they don't appear in the new state. w can then be positive for beneficial objects for the agent while negative w values yield objects that are *negative* for the agent's strategy (and corresponding rewards). The tests for this methodology were performed on the Mrs. Pacman videogame and humans were included in the experiments to evaluate the contribution of the saliency maps generated to the overall model transparency and explainability. In the tests, 40 participants were included and were prompted with different questions regarding the elements that they believed influenced more the agent. No claims were made on the skills of the recruited participants. Instead results show that showing saliency maps to them achieved better interpretability than showing only clips of agent interacting with the environment. In that regard, users generally perceived saliency maps could give additional info on the model used in situations where the agent's policy is not trivial to interpet only by analyzing its inputs.

An approach using saliency maps was presented in [113], where explanations of visual DRL agents are generated through a modified version of Layer-wise Relevance Propagation (LRP). The aim of the work is to give each unit (in

case of visual agents, input pixels) a weight of its contribution and this is done in a backward manner starting from the network's output back to the initial inputs. In addition to the base LRP concept, the authors used an arg max operator to omit messages that are not relevant for a particular passage. That is, $R^{l,l+1}_{j \leftarrow k} \coloneqq R^{l+1}_k$ if $j = \arg\max\{w_j a_j\}, 0$ if not, where R^l_k is the propagating relevance score of neuron k at layer k. To demonstrate the approach given, the authors chose a Dueling Q-Network [299] in which the contributions of the networks are taken into account applying LRP on both instead of evaluating their contributions separately. Evaluation, analogously with [299], is made in Atari games and results show that the z^+ rule introduced by the arg max operator by the authors is capable of generating well-balanced saliency maps. However, other than highlighting different but meaningful areas of the input space, this work doesn't contribute any further besides selective and focused saliency maps.

Similarly, [87] presents a visualization method employed through custom saliency maps. In place of utilizing Jacobian saliency maps [240], the authors used saliency maps for both $\pi(I_{1:t})$, the distribution over actions to take at time t (policy) and $V^\pi(I_{1:t})$, the expected future value of following policy π after observing $I_{1:t}$, on various A3C-trained agents. Authors here introduce a perturbation defined as $\phi(I_t, i, j) = I_t \odot (1 - M(i,j)) + A(I_t, \sigma_A) \odot M(i,j)$, which introduces, around the region centered in (i, j), spatial uncertainty in the form of a Hadamard product $\odot$ of the image and a 2D Gaussian blur $A(I_t, \sigma_A = 3)$ on that image. Saliency maps are then constructed using temporal differences, defined as $S_\pi(t, i, j)$, for every pixel in I_t. The interested reader could find further details on the definitions and steps provided in [87]. The agents are trained and tested in multiple OpenAI Gym tasks. Tests show that saliency highlights strategies of the agents which are not recognizable at first sight and reveal regions that appear somewhat meaningful to agents while being in certain states. Saliency maps generated in [87] were oriented at debugging the agents' policies by exploiting visual information on what the agents believes it's important the most in a given situation. Moreover, the work included also user testing with 31 students at Oregon State University by measuring how this technique could, in effect, help non-experts at understanding agents' behaviours. Results show that saliency maps generated by this technique help both experts and non-experts to identify salient inputs of the agents' by highlighting visually what are the major contribution to a particular policy at given time steps. A brief mention is made about memory and its importance on the explanation method in such, as implemented in the paper, LSTM cells and networks strongly rely on the concept of memory. The latter aspect of this methodology still needs further research in order to better combine memory information with visual representations of agents' mechanisms.

Saliency maps are also used in [199], where, differently from obtaining saliency in a post-hoc manner, the authors introduced a method to achieve built-in explainable features. The idea is based on the DRL models with attention modules integrating visual soft self-attention into a baseline model from [189]. The authors introduce the concept of *Free Lunch Saliency* (FLS) in that, with a small addition, baseline state-of-the-art models could be made more interpretable

without any additional computational overhead. Experiments were made in the Atari games environment where the agents are trained with PPO and perform below to equal to the baselines proposed. Evaluations are made by means of metrics such as Normalized Scanpath Saliency (NSS), the Kullback-Leibler (KL) divergence and the Shuffled AUC (Area Under Curve).

Other visual methods

Apart from DTs and saliency maps, other visual methods that are, in some way or another, generating explanations of visual information are grouped in the following subsection. This comprises a great variety of methods that don't have a common technique, as previous subsections, and were so grouped into the following.

In [5] the HIGHLIGHTS and HIGHLIGHTS-DIV procedures were presented. Those procedures are visual representations of an RL agent and its *trajectories* rewards obtained in order to pursue a given goal. More formally, the procedures start with choosing a set $T = \langle t_1, ..., t_k \rangle$ of possible trajectories of the agent, where each trajectory comprises of a sequence of consecutive states and actions take $\langle (s_i, a_i), ..., (s_{i+l-1}, a_{i+l-1}) \rangle$ where l is the sequence length. In order to generate sequences that do not exceed a meaningful and interpretable length, a limit budget k is given for the size of the summary such that $|T| = k$. This procedure led the authors into making short video sequences of two algorithms, namely HIGHLIGHTS and HIGHLIGHTS-DIV. Those two algorithms are identical but the latter has a different trajectory selection: HIGHLIGHTS tends to select a group of very similar trajectories to include due to their importance while HIGHLIGHTS-DIV, instead, tries to preserve also states that aren't strictly similar between each other and include also less important states in order to be more informative for the agent's decisions. Tests were made in the game of Mrs. Pacman and three agents (trained on different epochs) were compared. A user study was provided with users voting in a Likert scale from 1 to 7 the confidence of each agent relative to the video sequences generated. This is to show the importance of generating such video sequences of most relevant trajectories. However, other than implicit screen sequences in which the agent acts, no importance is given to the actual generation of a human-comprehensible explanation as the highlights don't tell more than how the environment looks and how the agent play the game. More effort should be given in order to explain, e.g. with NLP models, how states are encoded, how decisions are made and why such decisions affect the overall performance of the agent.

Moreover, the work in [230] presents a XRL technique based on the concept of introspective analysis and, in particular, follows the approach presented in [5]. The method applied is divided in few steps. First, elements of interest are extracted: **certain** and **uncertain** executions are obtained approximating the agent's interaction policy using $\hat{\pi}(s) = n(s, \mathcal{A})/n(s)$, where $n(s, \mathcal{A})$ and $n(s)$ are the number of times the agent has chosen a certain action $a \in \mathcal{A}$ when in state s and the number of times the agent has seen the state s respectively. Similarly, **minima** and **maxima** situations and most likely sequences to max-

ima are extracted by means of transition-value and sequence analysis, when a state-transition graph is constructed between any state and the most likely path weighted by value from an initial state s_i to a final state s_f is found. Introspective analysis is then applied and visual summaries are generated. Visual summaries consist of sequences of video clips highlighting the agents' performances during the whole test episodes. A user study was provided with 82 participants from psiTurk and Amazon Mechanical Turk in which, for each participant, were given video sequences of agents' gameplay. The results, however, don't conclude that the summarizations generated by this technique is perceived as beneficial or descriptive by the users from an explanatory point of view, due to the nature of the highlights generated by the procedure. More research is needed in order to generate explanatory visual descriptions that could better argue the reasoning of the agents' to non-experts users.

A visual tool was proposed in [117], which utilized an Asynchronous Actor-Critic (A2C) [187] algorithm to train an agent to operate in a POMDP navigation scenario of the 3D First Person Shooting (FPS) game Doom. The structure of the work consists of directly training an end-to-end mechanism capable of ingesting raw 2D RGB pixels and acting in a A2C fashion. The steps faced to obtain the resulting actions are: processing the input; transform an image into a feature vector; process the feature vector with a Gated Recurrent Unit (GRU); apply softmax operator to the GRU hidden state h_t to obtain the actions of the agent. The full set of parameters trained are $\theta = \theta_\phi, \theta_\psi, \theta_\xi$, where θ_ϕ are the parameters of the convolutional and fully-connected layers, θ_ψ are the parameters of the GRU layer and θ_ξ are the parameters of the trainable activation function $a = \xi(h_t, \theta_\xi)$ of the last softmax layer. The visualization framework propsed by the authors includes several components such as the memory timeline view in form of an horizontal heat-map, the derived metrics view which encodes measures of both inputs and outputs, a t-Distributed Stochastic Neighbor (t-SNE) embedding along with a module which highlights trajectories taken by the agent and other GUI adjustable parameters such as intervals, selected memory and episode number. This visual framework has indeed proven to be powerful but, as discussed in previous works and as presented in the paper, is more structured to being a debugging tool for experts rather than explaining to laypeople how a RL agent's inner mechanisms work. Indeed the user study provided with the paper has been conducted with three RL domain experts and the efficacy of DRLViz is evaluated consequently so it remains unclear how this framework, as useful as it gets, could be perceived from non-expert users.

The authors in [293] also use a visual representation to introduce DQNViz, a visual framework used to visualize DQNs' inner workings. The work is inspired by the state of the art at that time; like [41] and [157] applied to DQN, they state that the only visualization technique useful for such networks was t-SNE [276]. In this work, the network is trained in the Atari game breakout and the single components of DQNViz are analysed by the review of three domain experts in RL. The visual analytics system is composed of different modules: model statistics, which exposes summary statistics of rewards, number of games per epoch, average q value, etc.; epoch-level overview, marking the relationship

between actions and reward epoch-wise via visual charts; behavior patterns, such as trajectory view of episode-level exploration and summarization of event sequence data; segment analysis, useful to gain vision of what the agent sees from a trajectory segment. The case study is conducted via posing questions to the experts about the model statistics over training, agent's behavior patterns and about segment analysis. Results show that, overall, this tool is perceived as an useful tool for debugging agent's behavior under particular circumstances and that visualization of agent's states and memory could help give good insights on how the agent would reason. However, as for [117], this work is more focused towards giving insights available only to domain experts and it is not aimed at generating descriptive explanations to whom don't have know-how in the field.

Visual explanations in MARL are presented in [234] by a method called Multi-Agent Deep Deterministic Policy Gradient Visualization (MADDPGViz). The method proposed encapsulates different visual key elements such as: visualizing the state space via t-SNE (critic behavior view); average reward, loss and Q-values (statistics view); interactions between agents and reward sub-views of the environment. The models are trained in cooperative environments of communication and navigation and are observed by means of the proposed visualization tool. With the interaction views, users can easily track strategies of the agents and their relative positions with respect to the target, to get towards the cooperative strategies and assess whether the agents have learned a correct optimal behavior or not.

In [255], visual representations are presented with a particular focus on transfer learning. Indeed the authors, in order to account problem that derive from the nature of RL algorithms, proposed a causal view and generalization of interpretable and transferable models. The method is dived in multiple steps: first, the cusal graph of the augmented domain data is constructed; second, based on the causal knowledge built on the previous step, inference is done by probing the original model; lastly, the target model is used to train the policy on the target domain by offline RL learning.

Another approach, similar to [255], is [303], where transfer learning is used in combination with RL. This work, however, focuses on generating explanations for the application of Artificial Intelligence of Things (AIoT) by analyzing and displaying salient parts of the inputs. The RL agent extracts relevant spatio-temporal features from the visual inputs and then a mixture of transfer learning and RL is used in a self-paced manner. Weak explanation is then provided with a deep transfer learning model that the authors claim to be both post-hoc interpretable and performs better than other state-of-the-art models.

Visual representations in forms of sparse bayesian RL were presented in [186]. The methodology proposed is composed of three modules: the state image encoder (SIE), the Sparse Bayesian RL (SBRL) training module and the snapshot storage. The SIE module captures the visual input and encodes it into a 64-dimensional state representation. The SBRL module uses Relevance Vector Machines (RVM) to assign a posterior probability distribution over the snapshots taken. The snapshot storage maintains all the snapshots learned from SBRL and evaluates the relevance of the various snapshots over time. To

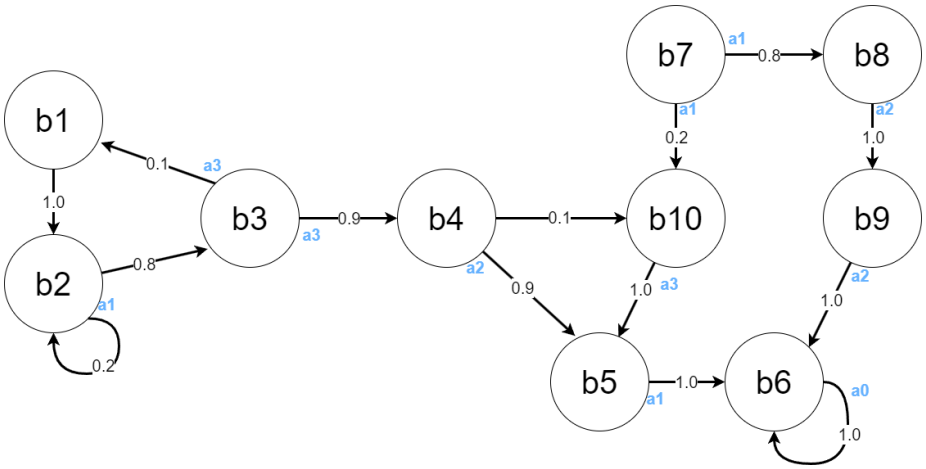


Figure 2.2: An example Abstract Policy Graph presented in [270].

test their methods, the authors used a navigation task in which Visual SBRL (V-SBRL) exhibits a good level of interpretability of RL policies.

Other visual representations are introduced in [48] where, in place of saliency maps, conceptual embedding intermediate representations are utilized as forms of explanations. The latent representation space of the employed deep RL models are utilized to capture causal factors and can, thus, yield inherent interpretable forms of agent behaviour nearer to human reasoning. In addition to this, the authors claim that the proposed method is also more efficient compared to the free DRL models used.

Visual representations by means of Abstract Policy Graphs (APGs) are given in [270], where explanations are presented in a lower complexity graphical form. The process adopts a *freature importance function* which caculates the conditional expected score for a feature w.r.t. a predefined function $g(s)$, i.e. $q_f(v) = \mathbb{E}(g(s)|s[f] = v)$. A representation of APGs generated by this method is given in Fig. 2.2. The algorithm used for APG generation (APG Gen) comprises of various steps such as the processing of states by an importance measure, splitting the graph with conditions over binary features, abstract state division and APG edge creation. The tests made on the prereqworld, introduced in this work, explain the better MDP policy readability and interpretability even when the state space grows exponentially.

In [323], a self-supervised deep learning framework was presented called Multi-level Abstraction Object-Oriented Predictor (MAOP). MAOP is made of several parts such as: the motion detection level, where potential regions of interacting dynamic objects are identified; the dynamic instance segmentation level, where proposed regions of *dynamic instances* are extracted from input; the object-oriented dynamics learning level, where the instance segmentations generated are modelled as object-based dynamics; the object detector and instance localization module, devolved in learning object masks from a sequence

of visual pixel input images (obtained via a CNN module). MAOP was then evaluated against several baselines and results show that it achieves a good performance and is able to correctly generalize in previously unseen environments. Visual intepretability is further given through object attention masks over the original input images.

Authors in [317] introduced a framework for visualizing DQNs by means of t-SNE. The environment used for these experiments were various Atari 2600 games in which DQNs are well known to perform above human level. The methodology is composed of multiple steps: train a DQN; evaluate the network, storing the hidden states, information about the graident and the raw pixels input; apply t-SNE; visualize data; analyze manually. After the feature extraction phase, t-SNE are calculated on pre-processed data (reduced dimensionality via Principal Component Analysis - PCA) and their respective saliency maps are constructed computing the Jacobian over all the network. A visual tool is then created that helps understanding what are the salient regions for activation of the DQN when playing games such as breakout, seaquest and pacman. Results show that, from the various clusters created, a user could further inspect moments of high attributions of reward, moments that are clustered together for similarity and terminal states. However, as in the case of the seaquest game, one can visualize that the agent seeks for immediate rewards in that is not able to generalize a more convenient strategy that pursues an higher but delayed reward. This suggests that the policy of the agent could be inspected through this visual tool and adjustements can be made to the reward design or agent's characteristics. No human study was provided and, thus, from [317] there's no evaluation method for assessing how beneficial this tool could be to the general public nor to experts.

In [61] the authors propose a visual tool for generating explanations extracted from classical MDPs. The process first highlights the recommended action and the relevant variable in a given situation, then it gives a visual representation of the former and, finally, gives a descriptive verbal explanation. The setting in which this system is applied is the generation of MDP-based explanation of emergency situations in a power plant. Given the expertise of the user (novice, intermediate and expert), different explanations are generated, obtained by the most relevant variable of the MDP in a given state. To extract the relevant variable of an MDP, this work uses the formula for obtaining $rel_s^V(X_i) = \max_{s' \in neigh_{x_i}(s)} V(s') - \min_{s' \in neigh_{x_i}(s)} V(s')$, where $neigh_{x_i}(s)$ is a set of states has give the same value as s for all other variables X_j $(j \neq i)$. This formula in then used to extract the most relevant, amongst all the variables, with $X_R^V = \arg\max_i(rel_s^V(X_i)), \forall(i)$. Templates of explanations are then extracted from a knowledge base and given to the user based on his experience. The operator then must choose between 5 different actions to bring back the plant to a non-critical state. Experiments were made with 10 different users and comparisons were made to the explanations given from a field expert. The authors point out the tests made for the automatic explanations, divided in 5 evaluation criteria, were, generally, perceived as useful from both domain experts and novice users. This is to be, however, intended as domain-specific

knowledge explanation generation and no further details besides the ones cited above are given in the tests.

Authors in [263] proposed a method of neuroevolution of self-interpretable RL agents. The key components of the work are concepts of neuroevolution and attention-based RL in which self-attention mechanisms are implemented for RL agents. The attention part is kept as simple as possible and only the self-attention module is used in the form of $A = \text{softmax}(\frac{1}{\sqrt{d_{in}}}(XW_k)(XW_q)^T)$, where d_{in} is the dimension of the input sequences $X \in \mathbf{R^{n \times d_{in}}}$ and $W_k, W_q \in \mathbf{R^{d_{in} \times d}}$ are the key and query matrices, respectively. Furthermore, a simplified processing of patches of the original image is used in place of the whole image due to computational purposes and a *Spatial Softmax* [70, 139] is used as input. Then, the processed input is used as an indirect encoding, i.e. the input is not used to search for the best parameter optimization of the NN of the agent but is used to restrict the attention only to key elements of the (image) input instead. This process leads to a vector called the *patch importance vector*, then only the k most important patches are considered and the agent is trained using Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [95] evolutionary algorithm. Experiments provided regard two vision-based RL tasks: DoomTakeCover, a VizDoom environment forked from the successful videogame Doom and CarRacing. The agent outperforms, in both tasks, the baselines of world model, Genetic Algorithm (GA), Proximal Policy Optimization (PPO) and Deep Innovation Protection (DIP). The authors also tested their agent in visually modified versions of the same tasks with the same objective and indeed found that the method proposed was able to generalize well across the selected domains while the baselines were not. This work, however, does not focus on interpretability apart from the inner visualization of the attention module. In addition, only very simple video game scenarios are considered and the model complexity is kept as simple as possible due to interpretability reasons. Further research must be done indeed to assess explainability of RL agents acting in indirect encodings via neuroevolution as proposed in this paper.

Another approach in [53] establishes a Probabilistic Graphical Model (PGM), introduced in [273], which aims at developing a model-agnostic explanation mechanism through graphical representations. The authors claim that th PGM-based approach requires little cognitive effort and is, though, a transparent way for generating explanations of RL agents' decisions. They provide also case studies that support this assessment and can thus help drive forward the idea of model-free XRL while also retaining transparency and trust.

2.1.2 Symbolic Representations

Symbolic explanations in XAI and XRL refer to providing human-understandable explanations for the decision-making process of an AI model using symbolic representations. These explanations often involve using human-interpretable symbols, rules, or logical constructs to describe the factors influencing the model's output. By presenting explanations in a symbolic form, users can better com-

prehend and trust the AI system's decisions, enhancing transparency and accountability. To date, symbolic representations do not constitute the majority of explanation methodologies as they tend to oversimplify agents' policies and thus are often used in limited or simple environments. There is, however, a substantial improved readability and model transparency through these methods as they are a powerful tool when it comes to generating model-agnostic explanations.

The use of symbolic AI for RL has been applied in [77], where the explainability is divided in three steps: low-level symbol generation, representation building and, lastly, reinforcement learning. This methodology was applied to an ad-hoc task where an agent is interacting with 4 variants of a grid environment and it can collect objects that give positive or negative rewards. Low-level symbol generation is done through a CNN 5x5 convolutional layer followed by a 2x2 pooling layer and then the corresponding decoding layers. Object detection is done by the sensing of the CNN and by the individuation of the salient regions of the image given in input. The output of this layer is fed to the second layers which builds an internal representation of the objects seen by elaborating spatial proximity, type transitions and neighbourhood processing. Each component of the latter layer is multiplied by certain weights w_1, w_2, w_3 which compose the object tracking equation $L = w_1 L_{dist} + w_2 L_{trans} + w_3 L_{neigh}$. Lastly, the output of the second layer is fed into a RL agent which processes the inputs and computes a Q function updated by the rule $Q^{ij}(s_t^{ij}, a_t) \leftarrow Q^{ij}(s_t^{ij}, a_t) + \alpha \left(r_{t+1} + \gamma(\max_a Q^{ij}(s_{t+1}^{ij}, a) - Q^{ij}(s_t^{ij}, a_t)) \right)$, where t is the time step, α is the learning rate, γ is the temporal discount factor and each state s^{ij} represents an interactions between two objects i and j. The overall process is calculated over a classical tabular Q-learning, given the statement simplicity of the game mechanics. The authors conclude that the methodology is a neural-symbolic, end-to-end RL architecture capable of decomposing the reward function and build an inner symbolic representation of the agent's perceptions.

The work presented in [165] propose a methodology of interpretability based on *Symbolic Planning* (SP) [42] and explains how RL + SP can help explain RL policies. The proposed framework is divided into three components: a planner, which uses symbolic knowledge to build a plan of symbolic actions sequences or subtasks; a controller, which uses DRL to learn a sub-task; a meta-controller which measures via extrinsic rewards the performance of the controllers and propose new intrinsic goals to the planner. The symbolic representation is formulated with the $\mathcal{BC}$ language and the translation of symbols into options, due to the hierarchical approach [17], is such that semi-Markov options are triples defined as (I, π, β), where $I = \{\tilde{s} \in \tilde{\mathbf{S}} : \mathbb{F}(s, \tilde{s}) = \mathbf{t}\}$ is the initiation set defined over a mapping $\mathbb{F} : S \times \tilde{S} \mapsto \{\mathbf{t}, \mathbf{f}\}$, $\pi : \tilde{\mathbf{S}} \mapsto \tilde{\mathbf{A}}$ is the intra-option policy, and β is the termination condition defined as $\beta(\tilde{s'}) = 1$ if $\mathbb{F}(s', \tilde{s'}) = \mathbf{t}$, for $\tilde{s'} \in \tilde{\mathbf{S}}$ and 0 otherwise. Here $\tilde{\mathbf{A}}$ is a set of primitive actions, $\tilde{\mathbf{S}}, \tilde{\mathbf{A}}$ are the sets of MDP state space and action space, respectively. The MDP tuple is then defined as $(\tilde{\mathbf{S}}, \tilde{\mathbf{A}}, \mathbf{P}^{\tilde{\mathbf{a}}}_{\mathbf{ss'}}, r, \gamma)$, where $\mathbf{P}^{\tilde{\mathbf{a}}}_{\mathbf{ss'}}$ is the transition matrix, r is the reward function and γ is the usual discount factor defined in a classical MDP. Rewards

are modeled as intrinsic, i.e. $r_i(\tilde{s'}) = \phi$ (ϕ encourages achieving optimal sub-task execution) if $\beta(\tilde{s'}) = 1$ and r (the usual RL reward from the environment in state s') otherwise, and extrinsic, i.e. $r_e = (s, g) = f(\epsilon)$, where f is defined as $f(\epsilon) = -\psi$ (ψ discourages exploring unlearnable sub-tasks) when ϵ is under a certain threshold and $f(\epsilon) = r(s, g)$ otherwise. In equations presented above, ϵ is a criterion that measures how a sub-task is executed with proficiency. Intrinsic rewards are consumed to learn sub-tasks while extrinsic rewards are consumed by the meta-controller where the controller itself is not able to learn useful policies for a specific sub-task. Finally, learning and planning is covered via symbolic planning where a controller uses deep Q-Learning with r_i and also uses experience replay. Meta-controller performs R-learning [228] by evaluating extrinsic reward signals and the *quality* of the symbolic plan is updated accordingly. An optimal plan Π^* is guaranteed to be reached when the R-learning converges, that is when the meta-controller stops and no further improvement to the plan can be made. Experiments are made in the Taxi domain [17] and Montezuma's Revenge Atari game. While the tests provided actually show that the methodology is indeed capable of abstracting higher level tasks with optimal sub-tasks, the limits of this work in terms of explainability are evident as this method is available only for finite horizon domains and no user study is provided, effectively quantifying how interpretable are the explanations generated in forms of symbolic planning.

Another symbolic representation of explanations is MARL Model Extraction library (MARLeME) [128], in which multiple components are brought together to form a framework. MARLeME is formed by the MARL system itself, the trajectory handler, the model extractor and model deployment / inspection parts. Few researches have focused their attention into MARL settings as it is inherently more complex to handle and to explain its dynamics. The outputs of MARLeME are agents based on Abstract Argumentation (AA) that are agents that make use of Value Argumentation Frameworks (VAFs) and produce approximations of the given models made more interpretable. A VAF is a tuple VAF $= (Arg, Att, V, val, Valpref)$ where Arg is a set of arguments, Att is a binary attack relation between arguments $Att \subseteq Arg \times Arg$, V is a set of possible argument values, $val : Arg \rightarrow V$ is a map between values and arguments and $Valpref$ is a given ordering among those values. Evaluations, both qualitative and quantitative, are made in the mountain car and robocup take-away environments where model fidelity is preserved and interpretable symbolic representations are given. Explanations generated via the MARLeME framework suggest that AA-based agents are a viable way to describe MARL system dynamics and behaviours.

Symbolic representations are used also in [145] towards the generation of deep symbolic policies for interpretable agents. The method comprises two components: the policy generator, which uses mathematical symbols and expressions to generate RL control policies, and the policy evaluator, which assess the goodness of the generated policies via episodic rewards. Policy evaluator will respond to the reward function defined by $R(\tau) = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T^{(i)}} r_t^{(i)}$, where

$T^{(i)}$ is the length of episode i, N is the total number of episodes and $r_t^{(i)}$ is the immediate reward of the episode i at time t following symbolic policy τ. The objective function $J_{\text{risk}}(\theta;\epsilon) = \mathbb{E}_{\tau\sim p(\tau|\theta)}[R(\tau)|R(\tau) \geq R_\epsilon(\theta)]$ is then approximated using Monte Carlo estimates. Evaluations on OpenAI Gym demonstrate that those models are inherently and provably stable and introduce an high interpretability component to the black-box nature of RL and NNs.

2.1.3 Causal Models and Game Theory

In this subsection, causal and game theory models used for XRL are presented. Causality refers to understanding the cause-and-effect relationships between input features and the model's outputs [254]. It aims to go beyond correlation and identify the factors that directly influence the predictions. Causal models in XRL involve constructing models that explicitly represent causal relationships and are often coupled with counterfactual explanations. Causality is composed of **causal discovery**, which is the identification of causal structures from raw observational data through statistical analysis of the variable of interest, and of **causal inference**, which is the process of inferring, by knowing partial or complete causal relations, causal effects of the variables taken into account For further details, the interested reader should refer to resources such as [319] and [254] which provide a good overview of causality in RL.

Together with causal models, game theory models come natural as similar approach to be grouped with causal explanations as, strictly speaking, both search relationships between components of the overall framework. Game theory and causal models in XAI and XRL share mutual focus on understanding decision-making processes by means of variable or situational analysis. Game theory analyzes strategic interactions between agents, seeking to predict outcomes based on their choices and preferences while causal models, on the other hand, aim to identify cause-and-effect relationships. One of the most widespread methodologies in game theory applied to XRL is the Shapley Value applied to the RL task which explains the decisions behind credit assignment.

Another application of explanations based on causal models is [296], in which Graph Neural Networks (GNNs) are employed and explained. The authors presented a novel methodology called Reinforced Causal Explainer (RC-Explainer), a NN-based RL agent that is devolved to the edge selection in order to perform causal screening of the proposed GNN. THe policy network of the explainer aims at learning the causal effect and dependencies of the edges to assess the influential sequence of the same.

Counterfactual questions of *What an agent need to do differently to obtain a different outcome* are presented also in [193]. The concept of counterfactual policy is introduced and explanations are generated with various experiments in a counterfactual fashion.

The work in [288] tries to address interpretability with characteristic functions. Agents controlled by neural networks are trained but not used for explanations. They are generated, instead, by computing the prime implicant explanations with the Frank-Wolfe (FW) method which are then compared with

the Shapley values in the connect four environment. The authors claim that, even if the FW-method performs slightly worse than other methods, it generally provides better interpretability via approximations of the neural-network-based agents.

Another explanation mechanisms that leverages counterfactual policies is proposed by [52], where the approach of FARE (eFficient counterfActual REcourse) is presented. The process aims at generating interventions through Monte Carlo Tree Search (MCTS) with learnt agent's policies π_x and π_f. To obtain this, however, the process id divided in four steps: create a node for each function f available in the DSL-defined functions library $\mathcal{F}$; connect each node after unrolling each sampled trace; store outgoing state-action pairs $(s_i, (f, x)_i)$ for each node f; train a DT for each node $f \in \mathcal{P}$, where $\mathcal{P}$ is a given graph, on the tuples $(s_i, (f, x)_i)$. Explanations are then generated via the interventions and interpretable boolean propositions. That is, for each node $f \in \mathcal{P}$, a DT is extracted and transitions (f', x') are followed afterwards.

Another application of Shapley value is [147], methodology applied to the MARL setting that attempts to address the credit assignment problem with a Shapley counterfactual approach. The method uses Centralized Training with Decentralized Execution (CTDE) in various cooperative StarCraft II scenarios. The Shapley Counterfactual Credits Q_i^ϕ are computed with the following equation: $Q_i^\phi = \frac{1}{N}\sum_{j=1}^{N} \frac{1}{\binom{N-1}{j-1}} \sum_{S \in S_j(i)} \Delta v(i, S)$, where $S_j(i)$ is the set comprising of agents with size j that includes the i-th agent, A the set of all agents, $\Delta v(i, S) = v_{S \text{ in } A} - v_{S \setminus i \text{ in } A}$ is the marginal contribution of agent i in the subset S of A with $S \setminus i$, the set of S removing i-th agent. Results and ablation studies show that the proposed methodology is able to perform well in the given environments and proposes increased interpretability through Shapley value estimations.

In [201], counterfactual explanations were generated in a generative RL setting to showcase the potentiality of such methods in RL interpretability. The idea behind the work is to generate counterfactual examples that resemble the original input in order to make the agent select an action a' instead of a, as with the original state. The process is then formulated as an optimization problem: minimize $||E_w(A(\mathbf{s})) - \mathbf{z}_\mathbf{w}^*||_2^2$ subject to $\arg\max_{a \in \mathcal{A}} \pi(D_W(\mathbf{z}_\mathbf{w}^*), a) = a'$, where $\mathbf{z}_\mathbf{w}^*$ a point in the latent space used to represent a possible state of the agent and D_W and E_W are the *Wasserstein* decoder and encoder, respectively. The agent is trained then with a deep convolutional neural network and used in user studies as well to detect the fidelity of counterfactual states and flaws of a given agent in Atari environments.

Another example of causal explanations is given in [180], where bayesian networks are used in order to generate causal graphs with the use of various algorithms. It is composed of four components: an RL agent, a bayesian network and two RNNs. The overall procedure produces a causal graph that aims at describing the causal relations of the features of the models engaged with the original input.

The work in [166] presents explanation in form of causal lens. The au-

thors employ a Structural Causal Model (SCM) in an RL setting of 6 different scenarios (divided between OpenAI Gym and a StarCraft II task) in which the technique presented generates, through NLP templates, conterfactual (*Why* and *Why not* questions) sentences to provide model-free RL explainability. The work is based on the concept of SCMs applied to human cognition and builds an *action influence graph*. The graph is then used to learn structural causal equations using a regression learner $\hat{\mathbb{L}}_{(s,a,r,s')\sim U(D)}$ in which updates are done only to equations with specific actions from an experience replay memory D, drawn uniformly as mini-batches. In addition, a user study was provided with **128** participants from *Amazon Mechanical Turk* [31] and the methodology was evaluated on task prediction, explanation *goodness* and trust. Authors claim that the model performs best on the first two evaluation criteria. Finally, this work is oriented towards implementing epistemic knowledge and not being constrained to provide the causal model beforehand.

The work proposed in [233] addressed vision-based RL policies explanations generation through the development of self-supervised agents by means of causal features. Provided causal explanations are said to have two properties such as *maximum behavior resemblance* and *minimum region retaining*. The former could be defined as $f(\tilde{s}_t) \approx f(s_t), t = 1, ..., T$, where $\tilde{s}_t = f_{exp}(s_t) \odot s_t$ is the attention-overlaid state corresponding to a state s_t of an explanation model f_{exp},T is the terminal state, f is the actor network and $\odot$ is the element-wise multiplication operator. The latter could be defined as $\min_\theta \|f^\theta_{exp}(s_t)\|_1$, where $\|\cdot\|_1$ is the L_1-norm and θ are the parameters of the explanation model f^θ_{exp}. Two components are used after the input is fed into the model, that is the Self-Supervised Interpretable Network (SSINet), which produces the overlaid state $\tilde{s}_t$, and the actor network which predicts the action to take afterwards. The learning then proceeds in two stages: first stage provides the generation of state-action pairs of the expert policy in order to train SSINet; second stage aims to learn interpretable attention masks to explain agent's behaviour, thus making the learning self-supervised. The experiments carried are in the OpenAI Gym's Duckietown and Atari 2600 games. The method proposed is then compared against Jacobian-based saliency maps [317], Gaussian perturbation-based saliency maps [87], attention augmented agent model (A3M) [192] and sparse free-lunch saliency (Sparse FLS) [199]. The evaluations are made with two introduced metrics: the *Feature Overlapping Rate* (FOR) and the *Background Elimination Rate* (BER). FOR is the ratio of overlapping between the true mask area and the learned mask while BER is the ratio of the area of the background eliminated by the mask to the area of the whole background. FOR is defined as $FOR(s) = \frac{S_{e,f} \cap S_{t,f}}{S_{t,f}}$ and BER is defined as $BER(s) = \frac{S_{t,b} - S_{t,b} \cap S_{e,f}}{S_{t,b}}$, where $S_{e,f}$ is the area of extracted features, $S_{t,f}$ is the area of true task-relevant features and $S_{t,b}$ is the area of true background. The proposed evaluations are accounting how much attention agents pay while learning and the proposed methodology has an overall good performance over the tasks presented. It is unclear, however, how beneficial to the general public are the explanations by means of the FOR and BER metrics and how could they affect non-experts in

the field.

2.1.4 Contrastive Explanations

Contrastive explanations are explanations in which policy features of the agents are decomposed and weighted in order to build contrasts between different features [248]. Usually, this is done in order to pose *Why* and *Why not* questions, based on the weights that the agent assigns to the various components. Common techniques of explaining why features are selected over others is Minimal Sufficient Explanations (MSX) and Reward Decomposition [129], which tend to destructure reward signals into multiple parts for extracting analytic properties.

The work in [158] discusses the problem of imitating a Deep Reinforcement Learning (DRL) policy into an interpretable model. They claim that existing approaches fail to fully utilize critical experience points due to heavy-tailed experience distributions in DRL tasks. The proposed method characterizes decision boundaries by identifying critical experience points close to the model's, resulting in a self-explainable structure in place of different more opaque models. Experimental results against state-of-the-art baselines demonstrate that the methodology proposed outperforms recent RL models on six games scenarios.

Another approach proposed in [2], generates policies that are inherently interpretable due to modified versions of policy evaluation and update. They propose to explore the state space by clustering it based on the actions closer to each cluster center. Their approach is tested in RoboSchool-Hopper and BipedalWalker continuous control tasks and interpretability is highlighted in the form of policies that approximate the problem differently from value or advantage functions, thus being naturally more readable.

Contrastive explanations by means of Shapley values are generated also in [103], where MADDPG and A3C algorithms were tested against sequential social dilemmas (such as Harvest) and multi-agent particles. The work didn't propose a new framework for explaining RL by means of reward decomposition or contrastive explanations but instead focused of the adaptation of existing methods to the multi-agent settings and implications of such methodologies in those environments.

Another method of explaining through Shapley values is presented in [20], where the Shapley Values for Explaining Reinforcement Learning (SVERL) methodology is introduced. The concept borrows ideas from the Shapley value construction and, when applied to RL, aims at building a value function that would be explained by Shapley values. However, in order to extract meaningful explanations from an agent's policy performance, two types of approaches are dervied: local and global SVERL-Performance (SVERL-P). The local SVERL-P takes the marginal contribution of state features from the agent's perspective and does so by calculating, for some policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ to be explained, the function $v^{\text{local}}(C) := \mathbb{E}_{\hat{\pi}}[\sum_{t=0}^{\infty} \gamma^t r_{t+1} | s_0 = s]$, where C is a given Shapley coalition and $\hat{\pi}(a_t|s_t) = \begin{cases} \pi_C(a_t|s_t) & \text{if} s_t = s, \\ \pi(a_t|s_t) & \text{otherwise} \end{cases}$. The function $\pi_C(a|s)$ is further

defined by $\pi_C(a|s) = \mathbb{E}_{p^\pi(s'|s_C)}[\pi(a|s')]$, where $p^\pi(s'|s_C)$ is the conditional limiting state occupancy distribution. These explanations will result in a policy that tries to best match the original policy but will not eventually be able to perfectly imitate π. Global explanations of the SVERL-P methodology are generated, similarly to v^{local}, by the function $v^{\text{global}}(C) := \mathbb{E}_{\pi_C}[\sum_{t=0}^{\infty} \gamma^t r_{t+1}|s_0 = s]$. This function allows to track the features importance on state s along with following states. It must be noted that this function alone cannot generalize for all the possible states due to the evaluation of v^{global} on a specific states. In order to globally explain the importance of the various state features in different states, the following function is used: $\Phi_i(v^{\text{global}}) = \mathbb{E}_{p^\pi(s)}[\phi_i(v^{\text{global}}, s)]$. Here, $\phi_i(f, x)$ represents the Shapley values of feature x_i to the prediction $y = f(x)$. The proposed solutions, however, acts as an approximation of the precise Shapley values, due to computational complexity. Evaluations are done in multiple simulated environments: Gridworld-B, Minesweeper and Taxi. Explanations are presented in tabular form and with histograms that represent the contributions of each features of the simulated environment.

The work in [277] uses contrastive explanations [184] applied to RL. This pilot study was employed in a simple grid world where a RL agent must move to achieve a goal and the authors provide explanations on what the agents evaluates at best decisions with an extension to the classical MDP. They construct the tuple $\langle S, A, R, T, \gamma, \mathbf{C}, \mathbf{O}, \mathbf{t}, \mathbf{k}\rangle$ where $\mathbf{C}$ and $\mathbf{O}$ are, respectively, state descriptions and action outcomes whereas $\mathbf{t}$ and $\mathbf{k}$ are transformation functions used to generate the explanations. Here, the Q-Value function is expanded from the classical formulation, incorporating also the Q-Value function for the users' preferences over some actions. The novel Q-function is defined as $Q_f(s, a) = Q_t(s, a) + Q_I(s, a), \forall s, a \in S, A$, where Q_I is the users' preferences function stated above. This generates a path of possible descriptions through the definition of a $Path$ function defined as $Path(s_t, \pi) = \{(c_0, o_0), ..., (c_n, o_n)\}, c_i = \mathbf{k}(s_i), o_i = \mathbf{t}(s_i, a_i), (s_i, a_i) \in \gamma(s_t, \pi)^*$, where $\gamma(s_t, \pi)^*$ is the most probable trajectory, obtained starting from a given state and following the transition with highest probability. A user study was then performed over 82 individuals to whom were presented the explanations generated by the reward signaling through the use of a Radial Basis Function (RBF) and different tuning parameters. In conclusion, however, this work didn't explicitly state qualitatively or quantitatively how beneficial are the explanations generated or how they were perceived from the participants. Thus, it remains unclear how, specifically, they can enhance model transparency and/or trustworthiness.

Embedded Self Prediction (ESP) model was presented in [153]. ESP follows from the Generalized Value Functions (GVFs) [259], a generalization of the classical value functions which decompose the cumulative attribution of the reward function to an arbitrary composition $F(s, a)$ of n-dimensional state-action feature functions $f_i(s, a)$, and is defined as $F(s, a,) = \langle f_1(s, a), ..., f_n(s, a)\rangle$. ESP is obtained by computing a function $\hat{Q}(s, a) = \hat{C}(\hat{Q}_F(s, a))$, where $\hat{Q}_F^{\hat{\pi}}$ is the GVF function of the agent's greedy policy $\hat{\pi}$, $\hat{Q}$ is the agents' Q-function and $\hat{C} : R^n \to R$ is a learned combining function from the single GVF vectors into

the action values themselves. The ESP-DQN algorithm trains a DQN via an ϵ-greedy policy by updating parameters θ_C and θ_F, which translates as optimizing Q^* and $\hat{Q}_F(s,a)$, respectively. A replay buffer is also used to store the exploration transitions. Authors claim that, analogous to the traditional Q-learning, given the sufficient prerequisites, the tabular version of ESP, ESP-Table, converges almost surely to an optimal policy. A proof is given in the appendix for the theoretical convergence for a given update interval K. However, a stronger convergence for $K = 1$ is still undetermined and remains an open problem, likely to the case of the base Q-learning. The ESP agents were tested on OpenAI Gym Cart Pole and Lunar Lander tasks as well as in the Real-Time Strategy (RTS) game StarCraft II environment called Tug of War. The decomposed GVFs are naturally more interpretable than the inner functions of the DQN used for DRL as they tend to extract single features from the reward function which highlight the agent's choices for the attribution of the single factors. The overall interpretability is given then by the concept of Minimal Sufficient Explanations (MSE or MSX) [129] which composes iteratively an explanation composed of features weighted by importance that are added until what the authors define a *sound explanation* is found. Even though the work presented in [153] is arguably solid for the construction of ESP, given the definitions of value functions and the aspect of features selection, a broader-level explanation method is still lacking for the general public to benefit as it still remains oriented towards domain experts. Authors suggest the possibility of extending the work to the more recent GVF Networks [259] in order to obtain different levels of granularity of the explanations.

Similarly, [122], proposes a method of explanation via reward decomposition. The main idea is to obtain a Reward Difference eXplanation (RDX) via MSX-MSX^+ and MSX^- for positive and negative consequences respectively-by decomposing the Q-vectors into differences $\delta(s,a_1,a_2) = \vec{Q}(s,a_1) - \vec{Q}(s,a_2)$ of two actions, namely a_1 and a_2. RDX can then build the MSX components by computing, respectively, $MSX^+ = \arg\min_{M \in 2^C} |M| s.t. \sum_{c \in M} \delta_c(s,a_1,a_2) > d$ and its counterpart $MSX^- = \arg\min_{M \in 2^C} |M| s.t. \sum_{c \in M} \delta_c(s,a_1,a_2) > v$, which is the same as MSX^+ except subtracting the term v, the sum of all reasons without the least contributing value. The method was tested in the CliffWorld and Lunar Lander OpenAI Gym tasks and the experiments show that some features learned by the agent were highlighted only by decomposing the Q-Value. This implies that the reward decomposition is indeed necessary to further investigate the reasons beyond the agent's policy and that without such explanatory values one cannot correctly identify which actions give a certain contribute when being in a given state. Whether the explanations cover all possible targets of people seeking explanations in RL remains scope for future work and can be still discussed as an argument.

Work in [215] is based on Reward Difference Explanation [66] and is modeled around the StarCraft II Learning Environment (SC2LE) [284]. Authors propose different tasks that are partially observable in order to provide further explanations of multiple agents acting in the simulated environment and are capable of generating explanations based on differences between rewards.

The work in [311] focused on explaining agent behaviour via implicit reward decomposition. The process aims to learn a map $\mathbf{H}$ called the *belief map* of discounted expected states alongside the standard Q-learning. The map is updated by the rule $\mathbf{H}(s_t, a_t) \leftarrow \mathbf{H}(s_t, a_t) + \alpha(1_{s_t,a_t} + \gamma\mathbf{H}(s_{t+1}, \arg\max_{a\in A} Q(s_{t+1}, a) - \mathbf{H}(s_t, a_t)$, where $1_{s_t,a_t} \in \mathrm{R}^{S\times A}$ is a binary indicator function that equals 1 at position s_t, a_t and 0 elsewhere. The map is then parameterised by θ_h and an action policy π_θ. Thus the update rule becomes $\theta_h \leftarrow \theta_h + \alpha\nabla_{\theta_h} L(\theta_h)$, where L is a predefined Deep Q-Learning loss function. Evaluations of such methods are made on three OpenAI Gym environments such as blackjack, cartpole and taxi. Contrastive explanations of such experiments are given in the form of tabular Q-learning. The authors claim that such explanations cannot be given post-hoc as one cannot reconstruct the agents' beliefs after a given policy and value function once learning has converged towards a certain point.

The work presented in [291] propose the Shapley Q-value Deep Deterministic Policy Gradient (SQDDPG), which The authors apply their methodology to their Extended Convex Game (ECG), an infinite-horizon variation, introduced in their work, of the Convex Game (CG), a popular multi-agent cooperative coalition scenario in game theory. This environment is, by nature, a MARL task and, as such, it brings up the problem of credit assignment where SQDDPG gives its contribution. The problem of credit assignment in ECG is faced here through Approximate Marginal Contribution (AMC) that directly estimates the marginal contributions of each coalition (i.e. $\Phi_i(\mathcal{C})$). The Shapley Q-value is approximated then by $Q^{\Phi_i}(s, a_i) \approx \frac{1}{M}\sum_{k=1}^{M} \hat{\Phi}_i(s, \mathbf{a}_{\mathcal{C}_k\cup\{i\}}), \forall\mathcal{C}_k \sim Pr(\mathcal{C}|\mathcal{N} \setminus \{i\})$, where $\mathbf{a}_C = (a_i)_{i\in C}$, $\mathcal{C}_k$ is an ordered coalition in which agent i would like to join in, $\mathcal{N}$ is the same as the CG and Φ is a function approximation of the marginal contribution. The interested reader can find more details on the implementations and mathematical definitions in [291]. The gradient of this function is then obtained through solving $\max_{\pi_i} Q^{\Phi_i}(s, a_i)$ for each agent i. Evaluations are done in three environments: cooperative navigation [161], prey-and-predator and traffic junction [252]. Various baselines were used for comparison such as Multi-Agent Deep Deterministic Policy Gradient (MADDPG) [161], COunterfactual Multi-Agent policy gradient (COMA) [71], Independent A2C (IA2C) [258] and Independent DDPG [150]. Overall, even if not the best performant technique in all the tasks, SQDDPG exhibits a better interpretability highlighting what's behind the decomposition of credit assignment in such a scenario.

In [66] the concept of minimal sufficient explanation (MSX) and Reward-Difference Explaination (RDX) were introduced to explain agent's decisions in Deep Adaptive Programs (DAPs) tasks. Adaptation Based Programming (ABP) is a framework in which programs that need to be executed have choice points through adaptive variables. RL explanations are then generated in this work by decomposing the original reward into smaller components with a SARSA-style algorithm. The RDX is then given by $\Delta_\pi(s, A, B) = Q_\pi(s, A) - Q_\pi(s, B)$, where the notation is the usual RL Q-value notation. MSX is then defined by having $\sum(Q) = \sum_{(r,x)\in Q} x$, where $R^-(Q) = \sum(\{r, x\} \in Q \wedge x < 0)$ and $Sub^+(Q) = \{S \subseteq Q | (r, x) \in S \implies x > 0\}$. Defined then $\mu(s, A, B) = S \in$

$Sub^{+}(Q) : \sum(S) > R^{-}(Q) \wedge |S|$, is minimal where $Q = \Delta_{\pi}(s, A, B)$. Tests were made in a custom OpenAI Gym environment composed of a grid world where the objective is to collect all the fruits and avoid being struck by random lightenings of squares: in either case the game ends with a positive reward (objective complete) or negative reward (agent dead).

2.1.5 Imitation Learning

Imitation learning (IL), also known as behavioral cloning or learning from demonstrations, is a type of ML approach used in the context of RL. In IL, an agent learns a policy by observing and imitating the behavior of an expert or a pre-existing demonstrator rather than interacting with the environment through trial and error, which is the typical RL approach [133]. Usually, the technique used is Inverse Reinforcement Learning (IRL) as, indeed, the process of agent learning is reverse [114]. That is, the agent is aiming at learning a function f that describes the behaviour of a human agent to try to replicate it in the simulated environment. IL can be beneficial in scenarios where gathering RL experiences through exploration is costly, time-consuming, or impractical.

In [203] an explainable generative adversarial imitation learning (xGAIL) method was introduced as a framework to learn a real-world human agent's decision-making policy of a taxi trajectory task. After preprocessing the data, a generator network and a discrimnator network are employed in order to train a policy function $\pi(a|s)$. Lastly, explanations are generated with Spatial Activation Maximization (SpatialAM) and Spatial Randomized Input Sampling Explanations (SpatialRISE). The objective function of SpatialAM is defined as $\mathbf{O}_s^*(a) = \arg\max_{\mathbf{O}_s}\{\pi(a|\mathbf{O}_s) + \lambda \cdot \text{Realness}(\mathbf{O}_s)\}$, where $\text{Realness}(\mathbf{O}_s) = -\text{Dist}(\mathbf{O}_s(a), \tilde{\mathbf{O}}_s(a))$ is a spatial regularization term, given $\text{Dist}(\mathbf{O}_s(a), \tilde{\mathbf{O}}_s(a))$ is the mean square distance between the terms and λ is the weight of the term of regularization. Differently, RISE [212] and, consequently, SpatialRISE aim to discover the salient regions of the input by randomly masking it and probing the original model in order to obtain its attentions.

Based on the work of [111], [143] proposes an IRL-based summary extraction with the aim on reconstructing human beliefs about how the agents may choose their actions. Explanations are conceived with the framework of maximum entropy RL (Max-Ent) [328] and Imitation Learning (IL). Those takes place via machine teaching (obtaining a reward function that is *behaviorally equivalent* to the optimal policy π^*) or active learning (producing a policy summary by the extraction of maximum policy-reconstruction quality of the trajectories). Tests are done in domein such as a random gridworld, pac-man grid and HIV simulator. Human-subject study further proves that, for some of the tasks, the explanations given by the IRL-based method are preferred over other summary types. For other tasks results show no direct evidence that human subjects perceived the explanations given as more explanatory in regard to the others.

Programmatically Interpretable Reinforcement Learning (PIRL) [281] is a kind of higher-level abstraction of DRL based on the concept of imitation learning [224, 227]. In this work, a specifically designed language called Neurally

Directed Program Search (NDPS) is constructed programmatically while constantly imitating a neural policy oracle $e_{\mathcal{N}}$ for a given environment. The objective of the NDPS is to define syntactically a set of programmatic policies called a *sketch*. The syntax of the language includes operators such as: c, for the constants; **peek**(x, i) which is used for inferring the i-th observation from a history x; **fold**, which is a higher-order combinator of sequences and the various x_i are unbound input variables. The framework of PIRL is tested on the The Open Racing Car Simulator (TORCS) as well as other OpenAI Gym tasks. As a consequence of the work presented here, NDPS results in a more interpretable model than the standard DQN and is, thus, more easily explainable. Due to its inherent transferable nature, NDPS is also capable of performing well in unseen tasks as well as being controllable by symbolic verification.

Authors in [241] proposed an algorithm called Adaptive Imitation - Interpretation algorithm (AI-Interpret) which is based on a solution of a Domain Specific Language (DSL) applied to meta-level RL policies. The aim is to build a simplified version of the policies taken into account represented by easily understandable flowcharts via the Logical Program Policies (LLP) [239] method. Generated explainable DTs are α-similar to the original ones in that they perform as good as the starting policies while differing at most α, a given parameter which is usually kept small in order to maintain consistency with the original policies. The authors here proposed a system of generating decision aids to particular planning tasks. The results in four different planning environments show that users, generally, prefer the explanations generated by the methodology instead of autonomous planning. The flowcharts generated, through imitation learning, are generally perceived more useful and easy to comprehend than previous works or methodologies.

2.1.6 Attention Mechanisms

Attention mechanisms became prominent in the ML scene as it turned out to be a very powerful tool to address very complex scenarios which were known to be previously intractable or difficult to solve [279, 246]. Attention mechanisms are essential components that facilitate capturing relevant information from input sequences. They consist of queries, keys, and values, and employ a score function to calculate the similarity between them. The resulting attention weights determine the contribution of each value to the context vector, a weighted sum of the values representing relevant information. Here, works of XRL that use attention mechanisms were grouped into the same category because they put forward the usage of such a technique to explain RL models.

The work in [304] proposed an attention-aware self-supervised RL for visual saliency explanations and representations. The work focuses on extracting intermediate feature maps of the CNN, assumed to be $f(x) \in \mathbb{R}^{H' \times W' \times C}$, via a self-supervised attention module $\Psi(x) \in \mathbb{R}^{H' \times W'}$. Policy and state-value function are predicted, thus, using a feed-forward network $\pi(a_t|\Psi(x)f(x))$, $V(\Psi(x)f(x))$ used as functions approximations and the training loss is defined as $\mathcal{L} = \mathcal{L}_{\text{RL}} + \mathcal{L}_{\text{attention}}$. The explanations are then given in the form of attention-

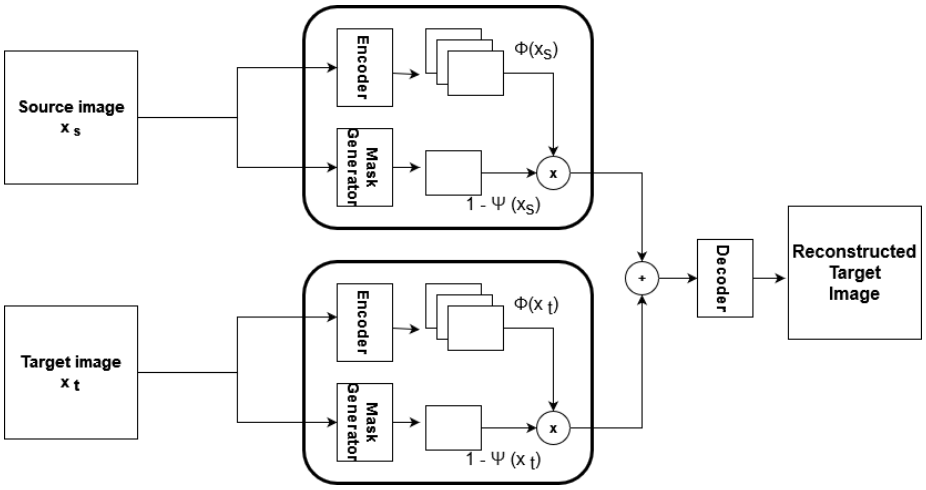


Figure 2.3: A schematic representation of the methodology used in [304]

aware saliency maps and processed visual inputs serve as a mean to obtain better interpretability.

The work of [243] proposed a multi-task RL methodology called CARE (Contextual Attention-based REpresentation learning) using context-based representations. The aim is to build an universal policy shared among all the tasks that uses task-specific metadata to select a precise representation for a particular task. The architecture is composed of k_i encoders that are mixed together and weighted by each α_i to a latent context, a context encoder that processes metadata with a language model and the both combined then define the final policy processed by the policy algorithm. The process is experimented in the RL benchmark of Meta-World [316] and tested against several baselines. Interpretation of the representations are given by computed cosine similarity and highlighted through visual representations.

An interpretable Deep RL method ws proposed in [292], where the model Alphastock was presented in order to address the problem of investment strategies. The framework proposed used a mixture of DRL and attention mechanisms while focusing on the interpretable aspect of the work. The setting is the *buy-winners-and-selling-losers* (BWSL) quantitative trading (QT) where an RL agent aims at optimizing the *Cross-Asset Attention Network* (CAAN) that models the interrelationships among stocks. Tests in the U.S. and Chinese markets demonstrate that Alphastock is capable of achieve an above-average performance while retaining the interpretable facet by measuring $\bar{\delta}_{x_p}$ (sensitivity of s to x_p, value of one feature w.r.t. a given period p).

A variant of the Rainbow model [101] was proposed in [309], where the authors retain attention mechanisms from the original model while proposing better interpretability introducing region-sensivie Rainbow (RS-Rainbow). The proposed architecture is composed of an image encoder that produces probabil-

ity distributions $\mathcal{P}_n$, a region-sensitive attention module and layers which are responsible for the policy's advantage and value streams. The interpretability focus in this work is on the region-sesitive module which enables users to see salient regions of the input space and allows a better understanding of the background learning mechanics. Tests were made in the Atari games environment but no further conclusions were made about explainability of the proposed solution.

An application of attention mechanisms toward interpretable RL policies is given in [192], where the an agent's network is augmented with a soft, top-down attention architecture. The model proposed has several **attention heads** where each attention head is composed of **queries** to a large input tensor and **answers** (low dimensional vector) to produce the output. The agents are trained with the Importance Weighted Actor-Learner Architecture [67] and are evaluated on 57 different Atari games. The tests showed that the agents are as performant as the state of the art at that time and that introduce a built-in saliency feature that is able to better explain the models. No user based tests are made, though, and no metrics on evaluating how beneficial it is are provided.

Attention mechanisms are used in networks presented in [9], where visual explanations are given through t-SNE visualizations. The work focuses on the attempt to obtain better interpretability of DQNs by highlighting the salient areas of the visual input of the networks. The training procedure is done with a loss defined by the linear combination of different losses, i.e. a loss of the form $\mathcal{L}_{\text{final}}(\theta) = \lambda_1\mathcal{L}_{\text{bellman}}(\theta) + \lambda_2\mathcal{L}_{\text{distrib}}(\theta) + \lambda_2\mathcal{L}_{\text{reconstruct}}(\theta) + \lambda_3\mathcal{L}_{\text{diversity}}(\theta)$. The agents are evaluated in 6 different Atari games and, by introducing attention layers and keys evaluation in the architecture, the system is viable for better intepretation while not heavily impacting the performances. Finally, the authors also introduced adversarial examples to better highlight salient regions and attentions of the agents but they dont' provide further analysis on the interpretability of the methodology proposed.

2.1.7 Natural Language Processing (NLP)

Forms of explanations using NLP were employed since the first advancements of the field and its adoption became prominent also in recent years. Even if this form of introducing transparency into the RL framework is promising, it is not one of the most common or widespread method. Forms of explanations include, but are not limited to, generating human-language sentences through NLP templates, qauerying a language model to argument agents' decisions and other techniques that NLP have been adopted in various works [15].

Query-based explanations are provided in [25] for MARL tasks in which agents' policies are abstracted, summarized and explained. The authors proposed explanations in forms of *when*, *why not* and *what* queries where the agents' policies are summarized and compared with a given baseline [96] in three different MARL scenarios. Authors provided also a user study that shows the inherent need for RL models to generate explanations that are understandable without excessive cognitive effort.

The autonomous vehicles task wass addressed by [132] as well by generating textual explanations in NLP format. The structure of the work is divided into two parts: one that is dedicated to train an end-to-end visual attention model that maps input images to controller's actions; the other video-to-text model produces explanations of actions in textual form. The methodology is focused on generating grounded introspective explanations and is evaluated in the Berkley DeepDrive eXplanation (BDD-X) dataset by comparing it with human annotations.

NLP generation in also present in [39] as a way to generate explanations of RL agents. The architecture of the framework is inspired from Imitation Learning (IL) and is composed of an high-level language generator and a low-level policy conditioned on such language. Multiple architectures are tested in a Minecraft-inspired 5x5 gridworld such as IL, IL with Generative Language, IL with Discriminative Language, plain RL, RL + IL, an autoencoder state reconstructur and a RNN state predictor. The method is able to generate demonstrations and explanations of hierarchical multi-task RL agents albeit not tested with a human-in-the-loop experiments.

Counterfactual explanations are given in NLP format by [249], where agents ought to act in partially observable environments. Here, explanations are generated via gradient descent in the form $\Delta Q(m_t, q_t, \sigma_t, a_h, a_t) \equiv Q(m_t, q_t, \sigma_t, a_h) - Q(m_t, q_t, \sigma_t, a_t)$, where $\sigma_t \in \mathbb{R}^{3\times N_s}$ is the list of N_s subgoal properties, m_t is the observed map, q_t is a robot pose for a given belief b_f. NLP explanations are generated using a rule-based grammar in two simulated environments and are evaluated for both accuracy and utility. The focus of the authors is to integrate *importance* ranking to guide explanations and planning under uncertainty. Such an approach has been promising even with its limitations wherever certain assumptions about the environment are violated and could generate poor explanations.

One example of NLP application of RL explanations generation is [57], where an MDP recommendation system is implemented and provided along with natural language explanations. The application domain evolves around recommending optimal study opportunities to university students that have low to none prior knowledge in choosing classes to attend. The main concept lies on the *action factored differential values* (AFDVs) which is calculated, from some state s, a policy π and a given value function V^π, by $V_2^\pi(s, a_1, a_2) - R(s) = \gamma \sum_{s'\in S} T(s'|s, a_1) \cdot [R(s') + \gamma \sum_{s''\in S} T(s''|s', a_2) \cdot V^\pi(s'')]$. From V_2^π a single AFDV is calculated by $\Delta^\pi(s, \pi, a_i, a) = V_2^\pi(s, \pi(s), a) - V_2^\pi(s, a_i, a)$ and to compute the full set of AFDVs the computation is done $\forall a_i \in A \setminus \pi(s), \forall a \in A$. The dataset used to generate the recommendations is taken from the University of Kentucky about students enrolled from 2001 and 2004, regarding all courses taken by such students. Transitions are considered atomically composed by [*scenario1* = {*var_ICP* = *A*} → *action* = *take_PDPS* → *scenario2* = {*var_PDPS* = *B*}], where a single transition is about a partial assignment of state variables *scenario1*, *action* taken after such assignment and then observing *scenario2*. The work then presents NLP explanations starting from a given template and recommending to final users which choice is best based on the past experience

and by foreseeing future actions that could be taken. Other than generating NLP-based explanations, however, the study doesn't state how much the explanations given are beneficial to a general public as a user study is not provided.

2.1.8 Other Methods

In this subsection all the methods that didn't fall, partially or entirely, into the categories presented above are grouped into a unique cluster hereby called "other methods", since not enough works have been published for each category to be coherently distinguished amongst the rest. At the same time, the various methodologies included in this subsection range from the use of specific mixed models to the use of more novel techniques which don't have anything in common with the studies already presented.

In [68], explanations are applied to the online RL problem where environment design could change over time and systems must adapt in a self-adaptive way. In their work, the authors explore the possibility of combining the reward decomposition [122] and interestingness elements [230] approaches to introduce a novel method for explaining online RL called XRL-DINE (eXplainable Reinforcement Learning - Decomposed Interestingess Elements).

The work in [91] presents an Integrated Decision and Control (IDC) framework for the autonomous vehicles task. The proposed solution decomposes the whole task into static path planning and dynamic optimal tracking and solves it in a hierarchical fashion. The authors indeed propose all the necessary rigorous mathematical framework needed to assess the increased interpretability and efficiency needed while addressing this particular problem.

In the application of [152], RL was used as an assistive tool for therapists' recommendations about four psychiatric conditions: depression, anxiety, schizophrenia and suicidal cases. The method proposed utilizes DRL agents with three working alliance ratings, such as task, bond and goal. The interpretability is given by running a 2D Principal Component Analysis (PCA) over the state space and transition matrices. The obtained system is capable of giving Disorder-Specific Multi-Objective Policies (DISMOP) which have a particular focus on explainable policy dynamics and, thus, hihglights the possibility of adopting interpretable methodologies in the field of psychiatry.

In [65], a sample-based rule extraction mechanism was proposed to introduce better explainability in RL models. The simple approach of model-agnostic and human-readable rule format is the key element of this work as it rpoposed iself to be an inherently explainable methodology for learned agents. The method was tested in various OpenAI Gym problems and shows a promising approach as it is also compatible with already existing DT-based methods.

In [222], a variation of the Tsetlin Machine (TM) [86] applied to RL tasks, was presented. Both off-policy and on-policy learning were proposed and the original TM was adapted to the dynamic nature of the value function iteration. The aim is to obtain explanations that are in the form of propositional logic and thus inherently interpretable by nature. The work can be considered still

experimental as few research has had been done in this field by adopting the TM architecture.

Particle swarm optimization in combination with fuzzy logic was instead presented in [97], where interpretable representations of RL policies are generated. The system uses fuzzy rules expressed in the form of $R^{(i)}$: IF s is $m^{(i)}$ THEN $o^{(i)}$, with $i \in 1, ..., C$, where $s \in \mathrm{R}^D$ is the input vector of the environment state, $m^{(i)}$ is the membership of fuzzy sets in the premise part, $o^{(i)}$ is a real number of the consequent part, considering the system to be D-inputs-single-outpt with predefined C rules and a $\mathbf{R}^{(i)}$. For particle swarm optimization, the best position for each particle, in the case of a maximization problem, is calculated as: $y_i(p) = x_i(p)$ if $\mathcal{F}(x_i(p)) > \mathcal{F}(y_i(p-1))$ and $y_i(p-1)$ otherwise. Here $\mathcal{F}$ is a given fitness function and the policy's parameters are repesenteded by particle positions. Those two concepts are then combined to form a fuzzy particle swarm reinforcement learning (FPSRL) with world models $\tilde{g}$ with inputs (s, a) to try to predict state s': $\Delta s'_m = \tilde{g}_{s_m}(s_1, s_2, ..., s_m, a)$. The resulting state is computed with $s' = (s_1 + \Delta s'_1, s_2 + \Delta s'_2, ..., s_m + \Delta s'_m)$ and the reward function is fed in $\mathcal{D}$ as per $\diagdown = \tilde{\mathbf{r}}(s, a, s')$. Explanations are given in the form of fuzzy rules and experiments in the OpenAI Gym proposed to be more interpretable than the black-box case. No human study is provided however as to assess the effective utility of generating such explanations for laypeople.

Authors in [167] proposed a framework of simple closed-form formulas for interpretability of policies. The idea is based on index-based policies and revolves around the low Kolmogorov complexity defined by $\kappa(\pi) = \min_{F \in D_F(\pi)} |F|$, where $D_F(\pi) = \{F \in \mathbb{F} | \pi_F = \pi\}$ is the set descripton formulas of the policy π, $|F|$ is the description length. The set of interpretable policies of lower or equal than K Kolmogorov complexity is then defined by $\prod_{\text{int}}^{K} = \{\pi | D_F(\pi) \neq \emptyset \text{ and } \kappa(\pi) \leq K\}$. The authors then pose the problem as a N-armed bandit problem of finding an high-performance policy in $\tilde{\prod}_{\text{int}}^{K}$, where $\tilde{\prod}_{\text{int}}^{K}$ is an approximated reduced set of policies, due to computational complexity reasons.

Another approach is given in [32], where collision prediction in combination with RL is used to address the autonomous vehicle risk-awareness task. The authors proposed, in a simulated autonomous vehicle environment, a framework for computing the collision prediction via vehicle dynamics and by stochastic position estimation. Their claim is to propose post-hoc explainability of vehicles in a given environment by means of mapping RL dynamics to the proposed vehicle dynamics.

A proposal of Self-Reward Design (SRD) was given in [268], where the authors adapt a NN to work in deep RL settings while also retaining interpretability. They propose a mechanism of an explainability design in various environments: a 1D Fish environment, a fish auction environment, a 2D robot lavaland and MuJoCo with SRD.into Much focus in this work has been put on interpretability but nonetheless, other than introducing SRD mechanisms in RL settings, no further contribution is given.

Another approach, in [271], proposed to synthesize programs in order to make more interpretable representations, introducing Learning Embeddings for

lAtent Program Synthesis framework, (LEAPS) [1]. So is done through the use of a predefined Domain-Specific Language (DSL) which constructs programs that include control flows, agent's perceptions and agent's actions. The methodology is applied to finite-horizon dicounted MDPs and, in order to reconstruct a particular MDP policy, and encoder q_ϕ and a decoder p_θ are used. The β-VAE loss is then defined as $\mathcal{L}^P_{\theta,\phi}(\rho) = -\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\rho)}[\log p_\theta(\rho|\mathbf{z})] + \beta D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\rho)||p_\theta(\mathbf{z}))$, where ρ is the given program, z is latent program and D_{KL} is the Kullback-Leibler divergence. The first step is to learn the embedding space and then the procedure searchs for a latent program that will maximize by the given MDP in the original task. Experiments are made in the Karel domain and ablation studies are also provided in order to represent the importance of each component, thus providing a promising common ground for future work with regaards of program synthesis in RL.

Progressive RL is adopted in [305], where Tree-Structured Policy-based Progressive RL (TSP-PRL) is presented as a mean to obtain explanations. The focus of this work is on temporal language grounding in video and TSP is used for hierarchical generation of policies, in combination with PRL for optimization. An alignment network for the *stop* signal is also provided where the binary cross-entropy loss is defined as $\mathcal{L}_{\mathrm{align}} = \frac{1}{M}\sum_{m=1}^{M}\sum_{t=1}^{T_{\max}}[U_{t-1}\log\sigma(C_t) + (1 - U_{t-1})\log(1-\sigma(C_t))]$, where C_t deontes a given confidence score, U_t is the temporal IoU (the interested reader should refer to [305] for formal definitions), $T_{\max}$ is the maximal timestep and M is the size of a mini-batch. The overall loss function then becomes $\mathcal{L} = \mathcal{L}_{\mathrm{tree}} + \lambda\mathcal{L}_{\mathrm{align}}$, where $\mathcal{L}_{\mathrm{tree}}$ is a linear combination of losses for root and leaf nodes.

The work in [92] proposed a framework called *EDGE* that is devolved towards strategy-level explanations of DRL agents. Interpretation is done by roughly understanding agent behaviour according to its inputs and further confirm first suggestions by launching adversarial attacks against its policy. Then an explanation-guided policy patch method tries to create a remediation policy that will be integrated with the original one and finally enhancing the agent by making it play against an adversarial opponent. Through these steps the method is self-explanatory in the sense that it provides explanations along with its training.

In [245] introduced stochastic and deterministic frameworks for portfolio management with RL focusing on explanations. Continuous portfolio optimization is done by means of DRL agents, in the stochastic case, are composed of a policy encoder network and a critic encoder network. Agents act with the use of maximum entropy RL training and distributional Q-value function, forming a Stochastic Policy with Distributional Q-network (SPDQ). Various performance metrics are used to evaluate the performances of the agents on U.S. stock market data and ablation studies are also performed to assess the importance of the various modules.

An application to the networking domain is given in [54], where a post-hoc explanation method is used to interpret the Pensieve model [174] to solve

[1] `https://clvrai.com/leaps`

the video bit rate adaptation problem. In order to explain the attribution of different weights to the various features present in the task, the authors used the LIME [223] tool. The experiments show that the conclusions drawn from selecting the features by the original model are not as straightforward as it may initially seem. Indeed, by removing features that are considered as most influential for the model's decisions, the authors show, by means of LIME, that the agent exhibits a different behavior than expected. Other than that, however, the contribution of this work is limited to the application of the LIME tool to a task in the application of RL agents in the networking field. Other steps would be needed in discovering how methods like this could impact more heavily the interpretability and explainability in RL.

The work in [236] presented a multi-task RL explainable method based on hierarchical reinforcement learning. The multi-task setting is defined as a collection of tasks $\mathcal{G}_k$ and subtasks g defined by a two-word tuple template as $g = \langle u_{\text{skill}}, u_{\text{item}} \rangle$. The overall task is composed of several stages. At each stage $k > 0$, the policy π_{k-1} is used as the base policy for solving the goal $\mathcal{G}_k$ which, in turn, is made of $\mathcal{G}_k$ used as the base goal. Then, the overall policy for the given goal $\mathcal{G}_k$ is composed of the base policy π_k^{inst} and an augmented flat policy π_k^{aug} in the case the base policy fails to solve the new task. A switch policy is used to determine, based on the binary variable e, when to switch to the new policy, i.e. $\pi_k^{\text{sw}}(e|s, g)$. In order to retain temporal relation the authors used the Stochastic Temporal Grammar (STG) based on the work of [237] and [213]. Learning is done by training the policies with the A2C algorithm [250] and by sampling the switch condition e and the goal g at each stage k, ensuring the goal $\mathcal{G}_k$ is always composed starting from the base goal $\mathcal{G}_{k-1}$. The gradient update is done by composing both the returns of the instruction and the switch policies and by alternating them every M iterations to avoid unstability of learning. The hierarchical approach combined with the STG showed a good performance when evaluated in the environment proposed, which is a Minecraft scenario with different tasks. However, the tasks proposed are too simplistic to be compared with a real-world scenario and are, thus, confined to the videogame environment proposed. The interpretability in this work is given by means of the inner structure of the hierarchical approach and no further conclusions are drawn neither on whether to make the method more interpretable towards laypeople nor when more complex environment are given.

An explainable RL framework was presented in [297], where the authors proposed a method to better interpret the choices of a recommendation system by means of model-agnostic RL agents. The architecture is based on couple agents π_θ^G, the generator, which is responsible of generating an explanation z given a state $s = (\mathbf{u}, \mathbf{v})$ and π_ϑ^D, the discriminator, whichdetermines whether user $\mathbf{u}$ will like item $\mathbf{v}$, i.e. $y = \pi_\vartheta^D(\mathbf{u}, \mathbf{v}, \mathbf{z})$. The two networks operate with various steps, such as processing the input via lookup layers, embedding the words via CNNs and an attention-based selection that outputs the probability of selecting a certain explanation. The authors also defined a measure of *goodness* of explainability composed of both readability $\Omega_r(z)$ and consistency $\Omega_c(z)$, thus giving equation $\Omega(z) = \lambda_r \Omega_r(z) + \lambda_c \Omega_c(z)$, where the terms $\lambda_r > 0$ and $\lambda_c > 0$

are introduce to balance the importance of the two components. The learning is carried by designing a reward function that weights both aspects and the learning proceeds with a doubly stochastic policy gradient. Experiments were then made both offline, with two different datasets, and evaluations done with human subhects. In the offline settings, the method proposed is compared against two baslines, one of which is random and the other borrowed from Neural Attentional Rating Regression with review-level Explanations (NARRE) [37]. Evaluations are made with the Pearson correlation M_c to test whether the explanations are consistent with the ratings and with the criterion M_e which measures the explainability. The tests that involved human participations were made with the collaboration of 4 Yelp reviewers that met a certain criterion. Results show that the method proposed is able to generate useful and meaningful explanations on why a certain item were recommended to a certain user and is able to outperform the state of the art at the time of their contribution.

A memory-based XRL method was proposed in [47], where state transitions towards a goal N_t are stored in memory to be explained via *Why* and *Why not* questions. This process saves the transitions from one state to another in a list T_{List} and each time the agents reaches a final state the probability $P_s \leftarrow T_s/T_t$ is computed, where T_t is the total number of transitions and T_s is a *success* sequence. The methodology was tested in a grid world with two variants: bounded and unbounded. Explanations are given in terms of probability instead of direct q-values. *Why* and *Why not* questions are generated, for each action in a given state, based on the probability of reaching a successful goal starting from that state. This study, however, does not present a user test scenario and doesn't go beyond simply translating the Q-values of the agent into sentences which express the probability of choosing a given action in a given state. In addition to that, experiments are done in a simple 3x4 grid world and are, thus, far from being suitable to real-world more complex applications.

A work for more interpretability of agents is [6] in which policies of agents summarization in order to introduce more transparency of agents' behavior is discussed. The key components of this concept are: intelligent state extraction, world state representation and strategy summary interface. The first step of intelligent state extraction is obtained through selecting a set of state-action pairs T and a limited budget k such that $|T| = k$ could also be included or not. The procedure called HIGHLIGHTS, formerly introduced in [5], can have two directions in generating summaries of policies such as by means of interest, based on policy reconstruction accuracy or via Peer-Designed Agents (PDAs) [175]. World state representation is highly dependent on the application domain and could vary based on the users expectations and knowledge. Lastly, authors present three main considerations around strategy summarization which include summary presentation, user-guide exploration of agents' strategies and understanding users' extrapolation from summaries. Evaluation criteria are also discussed where authors take into account evaluation domains, users and tasks characteristics and evaluation metrics.

Another view for the high-level explanations of inner mechanisms of NNs, specifically RNNs, applied to RL is [135], where the authors built a Moore

Machine Network (MMN) in order to explain hidden features of the network by means of finite state machines. The idea is to obtain a Quantized Bottleneck Network (QBN) that consumes quantized features and maintains quantized states and could be, thus, associated to a MMN. Functionally, QBNs are identical to autoencoders except for the latent representations which is not used, as for autoencodeers, for dimensionality reduction but for quantization of the activation layers instead. To quantize the encoding, the QBN processes $b(x) = D(\text{quantize}(E(x)))$, where E is the multilayer encoder which maps inputs x to the latent representation $E(x)$, D is a multilayer decoder and $b(x)$ is the final output. The outputs of $E(x)$ are assumed to be in the set $\{-1, 0, 1\}$ with an activaction function of the form $\phi(x) = 1.5\tanh(x) + 0.5\tanh(3x)$, a flatter version in the region around input zero of the classical tanh function. The process then maps two QBNs b_f and b_h to the F and H, sets of observation features and hidden states of the RNN respsectively. The resulting net is one where the tuples (o_t, f_t, h_t) are represented by a quantized version of the features and states by a QBN and thus can be viewed as a MMN. Experiments were done in various environment such as Mode Counter Environments (MCEs), a restricted version of POMDPs, Tomita Grammars, a popular benchmark for Finite State Machines (FSMs) and OpenAI Gym's Atari games. Explainability is given in forms of minimized Moore Machines obtained from the MMNs and can effectively reduce the number of features and observation states that a QBN processes. The outputted MMs representation are indeed more readable than direct policy rollout but, as the authors state however, this work is not intended to provide a full interpretation model in a primitive and minimal way to laypeople.

Early stages of explanations were provided in works such as [55], which aimed to find a correlation between explanations and RL. In their work, the authors outline the similarity of Explanation-Based Learning (EBL) and RL through simple grid-world problems and highlighted the possibility of linking framework such as Dynamic Programming (DP), by focusing on explanations, to RL. The work proposed also algorithms to solve such problems and comparisons were made in order to make better distinction between the two fields which might have been points in common.

Works such as [62] focus their attention on explaining factored MDPs by finding the most relevant variable in the policy. The work proposed two heuristic rules of finding a *relevant variable*, one based on utility and one based on the policy itself. The first on can be denoted by $rel_s^V(X_i) = \max_{s' \in neigh_{x_i}(s)} V(s') - \min_{s' \in neigh_{x_i}(s)} V(s')$, where $neigh_{x_i}(s)$ is the set of states which have same values as s for all other variables $X_j, j \neq i$; and a different value for X_i. The most relevant variable is then found by $X_R^V = \arg\max_i(rel_s^V(X + i)), \forall(i)$. The second one is denoted by $rel_s^A(X_i) = \#s' : s' \in neigh_{x_i}(s) \wedge \pi^*(s) \neq (s')$, where $neigh_{x_i}(s)$ is the same as before, $\pi^*(s)$ is the optimal action for state s and $\pi^*(s')$ is the action that the agent will take in other states such that $s' \in neigh_{x_i}(s)$. The relevant variable is then found by computing $X_R^A = \arg\max(rel_s^A(X_i)), \forall(i)$. The environment selected for experiments is the Intelligent Assistant for Operator Training (IAOT) [61], where the explanations generated by selecting the

most relevant variable for all the possible states are compared with those given by a domain expert.

In [262], Relational Forward Models (RFMs) are used in the context of MARL to introduce better interpretability. This method is based on Graph Networks (GNs) and is trained in a supervised fashion in order to be able to predict inner mechanisms of multi-agent systems. The input is a description of the state s of the environment and can output both a single action prediction for each agent or a prediction of the joint action of all the agents. RFM allows the agents to capture the information about the joint dynamics of the system and aids to speed up agents' learning. Agents are evaluated on three games of coop navigation, stag hunt and coin game. Introduced transparency is made by means of the RFM module which highlights behavioral trajectories of the agents and making them able to be analyzed in terms of social interactions.

Another concept brought to RL explainability is abstract Boolean algebra [196], where agents learn to optimize their behaviour in given environments and can be expressed under Boolean algebra. The work in [196] describes both the Boolean algebra for specific tasks and for value functions and the equivalency between the two. Starting from the base tasks, these are composed to form and solve tasks of higher complexity under Boolean algebra with the usual operators $\wedge, \vee$ and $\neg$. Experiments were done in a gridworld videogame environment and show that the agent is able to correctly optimize its behaviour toward a given goal and the explanations are given in forms of Boolean expressions.

The authors in [98, 99] propose a mix of Genetic Programming (GP) and RL which is called Genetic Programming Reinforcement Learning (GPRL). The most common high-leven approach of GP to RL is used here by means of interpreting the agents' policies. To do so, the authors implemented a Genetic Algorithm (GA) that has the common characteristics of GP like crossover, mutation, fitness function and selection but prioritized clarity of explanations. Explanations, here, are given in form of simple algebraic equations that can explain actions undertaken by the agent under certain circumstances. To favor simpler expressions, weights were given to each operator of the expressions as to keep the notations as clear and simple as possible while providing good insights for agent's reasoning. Due to the fact that this work is oriented towards applying RL policies to industrial applications (such as wind turbines), off-line learning is preferred as it is more suitable to such scenarios as training on-line RL algorithms that might initially fail to correctly explore the right policy is prohibitive. The learning is done on an off-line real-word complex scenario with a dataset composed of transition tuples of the form (s, a, s', r), where s, a, s', r are the same as in the usual RL notation. The agent's policy is approximated by a NN and trained via Monte Carlo updates. Three test scenarios were utilized as benchmarks: mountain cart, cart pole and a real-world application. The real world application is one where only partial observability is possible and the state action is continuous: the Industrial Benchmark (IB) [2]. Explanations were given for different level of complexities and authors claim that policies expressed

[2] `https://github.com/siemens/industrialbenchmark`

by algebraic equations are, generally, more readable and interpretable. Human case tests, however, were not given in such a study to fully comprehend how better interpretability this method gives to HMM and RL models.

The approach in [321] presented a hierarchical RL methodology based on Graph Deep Q-Networks (GDQNs) to introduce better interpretability. The work is structured to compute a sample-balanced deep SHAP value called *sample-balanced deep shapley additive explanation* (SE-DSHAP) that ranks each contribution of features of a model Φ, thus giving an explanations based on the subgraphs $G_S^B(v_i) = (\mathbf{X}_S, \mathbf{E}_S)$, where $\mathbf{X}_S$ and $\mathbf{E}_S$ are, respectively, the feature and adjacency matrices related to the subgraph and B is a positive integer. The methodology is tested in two active power correction control systems and are evaluated on the explanations graphs generated that could help human operators to better understand the inner working of the decision-making system.

2.2 Applications of XRL

Works that are accounted here as methodologies that address at practical and employed applications were grouped in this section in order to better represent potential XRL outcomes for real-world problems, as well as to overview its applications. Indeed, even if the methodologies proposed also lie in some of the categories presented in the previous section, as per the nature of the techniques applied, it seems more reasonable and natural to group them in a dedicated section. Principal applications of XRL methods are: robotics and Human-Robot Interaction (HRI), autonomous agents and self-driving vehicles, healthcare, finance and videogames (and simulated environments). The proposed clusters of applications that are here introduces are represented in Fig. 2.4, collected in Tab. 2.2 and further discussed in the following subsections.

Application	**Papers**
Human-Robot Interaction (HRI)	[232], [159], [29], [8], [176], [294], [295], [261], [96], [111], [74], [198], [110]
Autonomous Vehicles	[131], [132], [32], [204]
Healthcare	[242], [108], [40], [275]
Finance	[33], [140]

Table 2.2: Applications categories and papers included in each category.

2.2.1 Human-Robot Interaction (HRI)

XAI for HRI has been applied widely in the literature as per the current day [232, 159] with various methods [29] and various degree of trust and explanations [8,

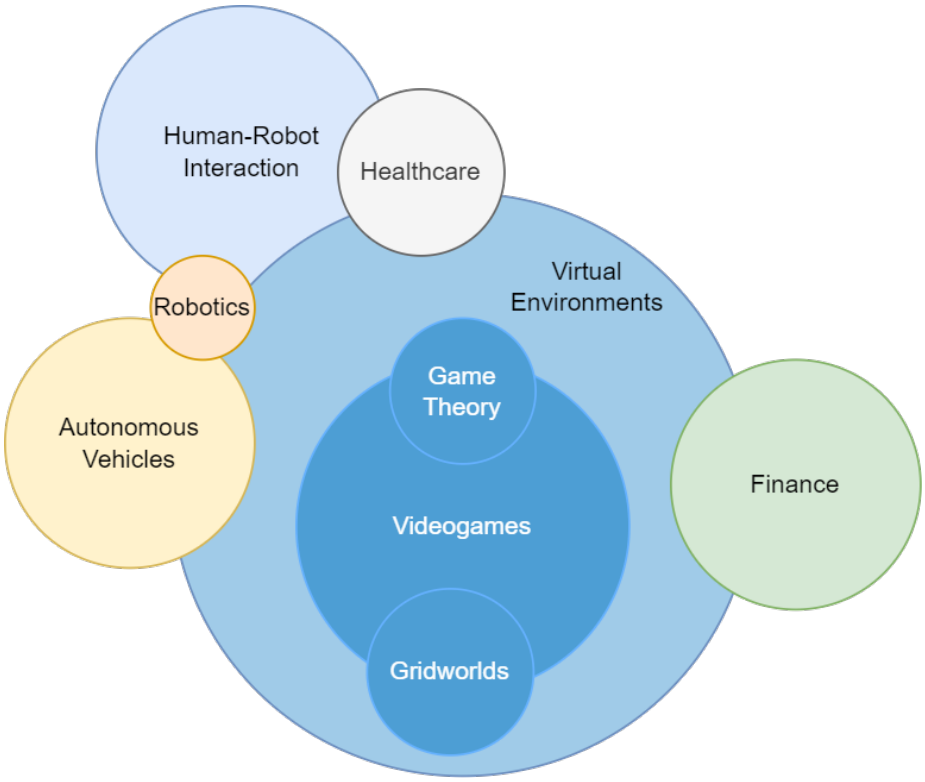


Figure 2.4: Applications of XRL represented as a bubble chart. Each bubble is a different application (size roughly represents the applications of XRL in the related field, the bigger the higher studies conducted). Elements contained in other circles represent subsets of applications. Many applications share methodologies with others and are represented as union of sets.

159]. One of the most common applications in XRL is indeed HRI, where agents represent the forms of robot and interact either in a simulated environment through human guide or directly with human subjects. Furthermore, trust and transparency are a required prerequisite in this field [29] and, consequently, a substantial subset of research has been put forward in HRI [232].

The work presented in [176] was oriented towards increasing the transparency of a human-like robot throught the use of emotional analysis and RL. Few tests using the iCub robot [3] were made to assess whether the 23 participants liked most between different emotional responses. Overall, the participant perceived that a response similar to that of a human was more likable.

Authors in works such as [294, 295] applied decision-theoretic planning and recursive modeling to a Human-Robot Interaction (HRI) scenario modelled as

[3] https://icub.iit.it/

a POMDP and point out why explanations are important in such field. The environment used for this work is a simulation environment called PsychSim [4] where a robot and a human agent, must cooperate in a reconnaissance task. The task consists in exploring different places and mark them as safe or not safe based on the likelihood of the current observation of the environment. The robot should make assumptions to make the human explore some place based on its current beliefs. Beliefs are made using sensors and giving a certain probability of getting false negatives and false positives. There is a considerable negative reward if the place is marked as safe while being not safe as the human can decide to wear protective equipment or not. If, for instance, the human decides not to wear protective gear and enters an unsafe zone it dies and the simulation is started over. In order to generate explanations in a NLP fashion, templates were used in natural language in order to provide humans with explanations regarding the robot's beliefs. Multiple factors built together the 2x4 design of the test, being *ability* (high or low) and *explanation* (four levels). The *ability* factor is referred to the correctness of the robot (high:100%, low: faulty camera) whereas the *explanation* factor could be either no explanation, explanation of two sensor readings, explanation of three sensor readings, and confidence-level explanation. Human users tests were made comprising of 220 participants from the USA from Amazon Mechanical Turk. This study's results highlight the link between explanations, trust, and the perceived value of explanations. A limitation in this study is that participants' understanding of the robot's decisions is self-reported, leaving uncertainty about their actual comprehension. Additionally, the measures in this study are aggregated from participants' responses after each of the 3 missions, posing another limitation.

Another application of XRL to HRI is [261] in which a Reward Augmentation Repair through Exploration (RARE) was presented. The framwework uses, in a POMDP environment, a family of Hidden Markov Models (HMM) to interact with a human collaborator. The steps needed to obtain such a goal are basically three: estimation of reward comprehension, which is needed to extract relevant features about the world and the reward signal; collaborative task execution and reward repair, that augments a POMDP with RARE components; explanation generation, that allows an agent to give an estimate of the human collaborator's reward function during the execution of the joint task. The environment given is a colored augmented Sudoku game in which the human agent has the possibility to make mistakes and the robot, in such case, gives suggestions to the human. Two versions of the robot were given, one with no explanation and one with minimal justification of which constraints of the game will be violated. The authors claim that the explanations given by the second robot was objectively perceived better. However, no further details were provided, the number of participants was not given, no further mentions of how the results were obtained and no explanation of how the tests were done was given.

The application in [96] attempts to solve policy explanation of autonomous robots in an NLP fashion. The process provides few steps on composing the

[4] `https://github.com/pynadath/psychsim`

explanations such as the composition of multiple functions through $f = C\_s \circ S\_a \circ R\_s \circ I\_q$. First, a query is transformed to a relevant set of states through the use of *Identify_question* (I_q) and *Resolve_states* (R_s) functions, where the text input of the query is mapped to a template and then it is resolved to a set of states. Then, *Summarize_attributes*(S_a) and *Compose_summary* (C_s) functions are used to summarize the relevant attributes of the states and translated to a language comprehensible by the interaction partner. The module that helps converting attributes to natural language and vice versa is that of a STRIPS-style [69] boolean classifier. In order to ground explanations, the authors apply their methodology to a scenario where an industrial robot is asked to inspect parts of a conveyor belt while being fed by a stock feed signal. Queries are expressed in the form of "When [do — will] you action?" when the agent is queried to explain its behavior and in the form of "Why [didn't — aren't] you action?" when the agent is queries to explain differences between expected and observed behaviors. Three case studies were provided, namely a delivery task, an inspection task and a stabilization task (CartPole). Although making the agents more transparent, this work is more focused on a task of debugging the agent code-wise (as the framing in the paper among the state of the art implies) rather than testing if the explanations make the user more at comfort with the agent. Even if the resulting explanations are generated with NLP, the usual queries and responses are parsed and generated through the use of templates and the scenarios adopted are generally simple in the state space. The work, however, is indeed interesting and could lie the base of future work applied to more complex scenarios and with additional components that could further improve the quality of explanations.

Another application of HRI is [111] in which a simulation of self-driving robot is made it communicate with an human user in an Inverse Reinforcement Learning (IRL) manner. The overall approach of the robot learning how humans make their inferences translates into solving an optimization model via Bayesian inference as: $P(\theta|\xi^{\theta^*}_{E_{1:n}}) \propto P(\xi^{\theta^*}_{E_{1:n}}|\theta) = P(\theta)\prod_{i=1}^{n} P(\xi^{\theta^*}_{E_{1:n}}|\theta)$, where $\xi^{\theta^*}_{E_{1:n}}$ is a robot's trajectory given some weights of reward function's features θ^* and the environment E and θ are the parameters of the objective function.

The work in [74] extended the idea borrowed from the field of Interactive Reinforcement Learning [267] agents towards explanations by presenting the method of Instruction-based Behavior Explanation (IBE). Explanations through IBE are constructed by estimating the changes in the environment states w.r.t. the agent's policy and are obtained by computing $\Delta s_t = s_{t+n} - s_t$ and by mapping the agent's behavior to the explanation signal. The latter step is done by a function $f : \Delta s \rightarrow m$ where m is an expression to explain agent's actions. Explanations is given in clusters $(m^1, m^2, ..., m^k)$ obtained by dividing the sets of Δs into clusters $C^1, C^2, ..., C^k$ and by computing $f(\Delta s_t) = m^i(\Delta s \in C^i)$. Expected values of the instructions M^{*i} are calculated with $e^i = E[m^*|m^* \in M^{*i}]$ and by normalizing e^i between the clusters to be $|m^i|$. Efficacy of the proposed methodology is tested in the lunar lander environment and nine male students aged between 21 and 28 who have never played the game before were given the IBE explanations. Results don't show noticeable evidence of benefits intro-

duced by IBE as: 1) the environment tested has both a low state space and is still considered a toy scenario that doesn't represent complex real-world systems; 2) the number of participants is both not adequate and not meaningful to the explainability as a whole; 3) results are not clear on how the experiments were made, how the explanations were generated and what effects are obtained through the use of IBE. In conclusion, no assessment could be made on whether this methodology is beneficial for the generation of meaningful real-world explanations of RL agents towards an audience of both experts and non-experts users.

Another application of HRI is in [111] in which a variant of the Inverse Reinforcement Learning (IRL) [198] algorithm is applied to robots enabling them to communicate their objectives. Robot's objectives are represented as linear combinations of features: $R_{\theta^*}(s_t, a_t, s_{t+1}; E) = \theta^{*T}\phi(s_t, a_t, s_{t+1}; E)$, where θ^* are the weights of the features, E is the environment descriptions and a_t, s_t are the usual state and action as in the RL definition. The trajectories are then defined as $\xi_E^\theta = \arg\max_{\xi_E \in \Xi_E} \theta^T \phi(\xi_E)$, where $\phi(\xi_E) = \sum_{t=0}^{T-1} \gamma^t \phi(s_t, a_t, s_{t+1}; E)$, Ξ_E refers to all the possible trajectories in E and γ is the discount factor between 0 and 1. The simulated environment is that of self-driving car in which an autonomous car and a non-autonomous car are present. Tests aim at developing approximate-inference models of users' behaviours in predicting how of one of the cars will bact in the environment. Various user studies are indeed provided in which exact-inference IRL is evaluated against approximate-inference IRL models. The utility of algorithmic teaching is then evaluated where coverage, analysed by providing actual behaviour samples that users can see, is generally perceived as a better component. The number of participants varies depending on the user study, whether was baselining the performances, evaluate the utility of algorithmic teaching or analyzing the utility of coverage. The work doesn't provide, however, visual or technical examples for the intepretability value generated by using such an approach.

Another application of HRI is in [110], where explanations are built upon computing the critical states of the agents by means of maximum-entropy RL. Such states are either policy-based or value-based, depending on the heuristic selected by the explanation generation mechanism. A definition of the set of critical states $\mathcal{C}_\pi$ for a stochastic policy π could then be: $\mathcal{C}_\pi = \{s | \mathcal{H}(\pi(\cdot|s)) < t\}$, where $\mathcal{H}(\pi(\cdot|s))$ is the definition of entropy of the policy's output action distribution at state s and $t \in \mathbb{R}$ is a certain threshold for it to be considered *critical*. The value-based case is similar but with a different definition; that is: $\mathcal{C}_\pi = \{s | (\max_a Q^\pi(s,a) - \frac{1}{|\mathcal{A}|}\sum_a Q^\pi(s,a)) > t\}$. A user study was conducted by the authors in which 72 participants recruited via Amazon Mechanical Turk rate the level of control they perceive when critical states are provided. No further conclusions are drawn after the discussion of the value that highlighting critical states gives and the limits this work has to face.

2.2.2 Autonomous Vehicles

XRL plays a crucial role in enhancing the safety, transparency, and trustworthiness of autonomous vehicles. By providing human-understandable explanations for the decision-making process, XRL enables stakeholders, such as developers, regulators, and end-users, to gain insights into how the self-driving vehicles operate in multiple scenarios [131]. Key applications of XRL in self-driving vehicles include safety and reliability improvement, human-machine collaboration, regulatory compliance, anomaly detection and diagnosis, end-user acceptance, transfer learning and many others.

XRL can be used to develop autonomous vehicle systems that can explain why they made certain decisions during critical situations. This is particularly important for debugging and ensuring that the system follows safety guidelines and adheres to ethical principles. For example, the effort devoted in [132] was employed towards explaining agent's decisions that represent attentions to visual stimuli in order to get better insight and debug agent's perceptions.

XRL can facilitate better interaction between the vehicle and its human passengers or operators. By providing explanations for its decisions, the vehicle can build trust with its users and enable a more natural and cooperative relationship [32]. Furthermore, autonomous vehicles are subject to stringent regulations to ensure public safety. By adopting XRL, developers can provide interpretable evidence to regulators, demonstrating that their system complies with safety standards and explaining how it handles potential edge cases.

Predictive control as a form of explanation is given in [204], where autonomous driving agents act in few simulated environments. In particular, Semantic Predictive Control (SPC) is used in order to provide a form of inherent explainabiliity by highlighting future predictions of the evolution of the environment by the agents.

XRL can assist in detecting and diagnosing abnormal behavior in an autonomous vehicle systems. If the system deviates from expected norms, it can provide explanations for its actions, helping developers understand potential faults and improve system robustness. With complex models like RL agents, it can be challenging to pinpoint issues and understand why certain behaviors occur. XRL could assist developers in validating the models, identify potential biases, and uncover unintended consequences. What is best is that XRL can be easily deployed to generate explanations in simulation environments, allowing developers to examine and understand the behavior of autonomous vehicles in various scenarios without real-world risks. Finally, explanations can facilitate knowledge sharing across different autonomous vehicle systems and insights gained from one model can be transferred and utilized in the development of other systems, leading to improved overall performance and efficiency.

2.2.3 Healthcare

XRL in healthcare aims at introducing transparency, interpretability, and safety in healthcare treatments and building trust in patients. By providing human-

understandable explanations for the actions and decisions of AI models, XRL can significantly impact various areas of healthcare [242]. It is, however, more widespread among XAI [315] rather than XRL per se as a general approach given the fact that self-supervised or semi-supervised learners are difficult to employ in those tasks. Nevertheless, XRL offers opportunities of development in this field, when applicable, and applications to healthcare could be further expanded.

One example of clinical decision support is [108], that uses Fuzzy Logic Controller (FLC) rules. The algorithm proposed, the Temporal Granule Pattern (TGP) mining, aims at increasing interpretability of FLC rules further adopting a deep RL framing. XRLcan be used, indeed, to develop models that provide explanations for their treatment recommendations, aiding healthcare professionals in understanding the reasoning behind the suggested course of action.

Another example is [40], in which contrastive explanations are generated and applied to common healthcare datasets. This exploits personalized treatment plans in which XRL can assist in explaining the reasons behind personalized treatment for individual patients. This can help patients and their families comprehend the rationale behind specific interventions and foster greater adherence to treatment regimens. In particular, this approach generates counterfactual examples via DRL with discrete-continuous hybrid action space.

Other methods of XRL, among others, could offer the opportunities of disease diagnosis and prediction, drug discovery and development, monitoring and anomaly detection and many more. By generating explanations for model predictions, clinicians can gain insights into the contributing factors and markers used by the system, potentially aiding in early diagnosis and intervention. In addition, XRL can be utilized to create AI models that explain the effects of potential drug candidates on biological systems.

For example, in [275], XRL methodologies are applied for data collected in the Intensive Care Unit (ICU). Given that the problem addressed in this work is partially observable, RL comes as a great candidate and also particular attention was given to the interpretability of the method. Indeed, RL can enable AI models to explain the factors contributing to a patient's risk of developing specific conditions or complications. This information can support clinicians in designing targeted preventive strategies and interventions.

Furthermore, another possibility is to build regulatory compliance and ethics. In the healthcare domain, interpretability is crucial for ensuring that AI models meet regulatory requirements and ethical standards. This could build trustu also for medical personnel as XRL can aid in medical education by providing detailed explanations of AI-generated diagnoses and treatment plans as in [275]. In addition, by providing transparent explanations for AI-driven medical decisions, XRL can enhance public understanding and acceptance of AI technologies in healthcare. This can reduce skepticism and improve the adoption of AI solutions for medical applications.

2.2.4 Finance

XRL in finance refers to the application of RL techniques in the financial domain while ensuring that the decision-making process of the RL algorithm is transparent and interpretable. The need for explainability in finance arises from regulatory requirements, risk management, and the desire to build trust with stakeholders. In many financial applications, stakeholders want to understand why a particular decision was made, what factors influenced the decision, and how the model arrived at a specific conclusion. This understanding is crucial for risk assessment, compliance, and improving the model's performance over time [33].

Unfortunately, only few examples are available in the current XRL literature as the field of application of finance is not as trivial as it might at a first glance seem as it poses problems of economic nature and intrinsic high unpredictability. This hinders the general use of RL techniques as high uncertainty environments very often introduce non-stationarity of MDP processes. Nonetheless, examples such as [140], show that XRL can be a valuable tool for interpretable financial AI models. In their work, the approach used is through the use of a DQN. Then, a SHapley Additive exPlanation (SHAP) approach was introduced where single features are weighted and added together according to the following equation:

$$\phi_j(N) = \sum_{S \subset N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!}(v(S \cup \{i\}) - v(S)),$$

where S is the subset containing the features used by the model, N is a set containing all the possible permutations of various features, $v(S)$ is a given prediction for a small feature value in S, compared to features not in S. Similarly to other approaches that use Shapley values, the explanations are generated in a contrastive fashion through the elaboration of features of reward signals. The dataset used for the experiments are 2 stock market (SENSEX and DJIA) comprising 30 companies trading in such markets. Thus, determining the importance of different features or inputs to the RL model can provide insights into which factors have the most significant impact on the decision-making process.

Thus, XRL could aid in explaining the entire model's behavior, local explanations focus on explaining the model's decisions for a specific instance or a small set of instances. Furthermore, representing the agent's decision-making process through visualizations, such as heatmaps or saliency maps, can help stakeholders grasp the model's behavior more intuitively. Simulating the model's behavior in different scenarios and comparing the results with actual historical data can provide a clearer understanding of the model's strengths and limitations. Overall, the field of XRL in finance is still a prosperous area of research and development, driven by the need to build trustworthy AI systems and ensure transparency in decision-making in critical financial domains.

2.2.5 Entertainment

Entertainment and, in particular, videogames are amongst the most common fields of application of both RL and XRL as they come in handy in experiment testing, benchmarking and reproducibility. It is worth mentioning, in this section, that videogames are a common category of application both for RL and, consequently, for XRL. Many of the papers in videogame applications were already included in the categories cited above and were thus classified as methodologies but the same still applies for the section of applications per se. Indeed, the interactive and real-time nature of videogames provide a dynamic environment where RL agents can make decisions in real-time and in continuously evolving scenarios. This real-time aspect allows players to observe RL agents' actions and outcomes immediately, making it easier to assess and evaluate the AI's behavior. The vast majority of works in XRL are divided into several environments such as OpenAI Gym [27], StarCraft II [284], simulated gridworlds or, most commonly, Atari Videogames [199, 112, 309, 188].

So much so that videogames are prominent in XRL as come in handy as a *toy* scenario and because, after the advent of advances in DRL in environments such as Atari games, potiential of RL turned out to be quite successful. As it came natural, a widespread testbed for RL algorithms was then adopted afterwards by XRL practitioners and a vast majority of XRL applications were employed in such scenarios. This, however, poses a limit for the actual benefit users could obtain from XRL methodologies as videogames are complex but distant environments compared to real-world applications that use sensors (autonomous vehicles), day-to-day machinery and equipment. There is a need, though, to thrive the research forward by applying methods that were used in videogames to actual XRL problems in order to better assess its utility in more practical applications.

Furthermore, videogames offer engaging and challenging scenarios that can serve as valuable testbeds for XRL research and development. Researchers can design diverse game environments to study the impact of explainability techniques on agents' performances and player experiences. Videogames have a vast and diverse user base, making it an area of significant interest to both researchers and the gaming industry. Applying XRL in video games can attract attention and promote the understanding and adoption of explainable AI methods.

Video games can be also used for educational purposes, and XRL can enhance the learning experience. By providing comprehensible explanations for agents' actions, players can learn strategies and tactics, which can be valuable for educational or training in games. Lastly, game developers often need to fine-tune AI behavior to create challenging and enjoyable gameplay experiences. XRL can help videogame industry protagonists understand AI agent decisions and improve their performance, leading to more compelling games, such as AlphaStar by OpenAI [283].

2.3 Current Limitations and Open Challenges

After exposing the numerous and undoubtful opportunities offered by XRL, an insight of the limitations of the current technologies and of the still open challenges will now be presented. A graphical overview that summarizes the topic of this section is reported in Fig. 2.5. Then, each single aspect is listed and explained in the following subsections.

Figure 2.5: Open challenges and limitations of the field of XRL presented in this work.

Interpretability-Performance Trade-off. Often, increasing the interpretability of a learning system can come at the cost of reduced performance. Highly interpretable models may not achieve the same level of complexity and optimization as their black-box counterparts. Indeed, there is a tangible degree of agreement that this trade-off exists to an extent where both simplicity and model explainability cannot be obtained jointly [18]. This is because, in order to obtain better transparency, the models and methods must shrink in terms of complexity of behaviours and, as such, represent simpler approximations of the original models [24]. Works such as [172] and [244], however, suggest that maintaining simpler methods can overcome this limitation of performance-readability trade-off in order to retain transparency. On the other hand, there are ongoing arguments where this phenomenon might be a false myth in fields such as ML [226], where athe authors discourage the use of explanations mechanisms, or in

specific RL tasks such as 2D mazes [173], where the authors claim that there is no accuracy-interpretability trade-off for such problems. As a consequence, it is still an open question whether there exists an interpretablity-performance trade-off in general XRL applications as well, and whether it exists and, thus, that it is recommended to develop a-priori interpretable models in place of post-hoc interpretations [289].

Complexity and Scalability. Developing effective XRL methods can be challenging due to the inherent complexity of the initial problems. Indeed, as [211] argues, the problem of obtaining interpretable MDPs policies could be a NP-Hard problem. In such a study, the problem of dynamic customer segment recommendations is addressed and interpretable policies are derived by using Mixed Integer Nonlinear Programming and McCormick inequalities. However, the authors show that, in order to maintain the policy transparent and interpretable, the number of binary variables grows exponentially with the number of recommended products. Thus, apart from cases of strict intractability, obtaining full interpretability might introduce computational issues and practically inaccessible requirements. Also, even if a boundary between complexity and interpretability exists, precisely defining it remains an open question that needs further investigation. A plausible solution that have been proposed, is to address scalable problems with simpler models [84], but such an approach doesn't address scenarios where dimensionality introduces intractability. In order to ascertain if this boundary constitutes an upper limitation or it could be in any way solvable is a current open challenge in XRL.

Real-World Problems. There is a need for XRL to face real-world problems as it is often utilized to solve toy problems or too simplistic scenarios to be of practical advantage. This usually obfuscates the true capability of generating explanations to non-experts in complex real-word applications and, thus, hinders its real pragmatic and theoretical value [181, 102]. This is due to the natural complexity that real-world applications have and, consequently, the XRL algorithms employed for simple environments, such as gridworlds, videogames, etc., cannot be applied to complex environments. As also discussed above, in the literature, whether certain degrees of complexity are achievable or not is still an open problem. For instance, papers such as [79] argue that there is a false hope behind scaling explainability in fields like healthcare. Nontheless, to establish more evidence, future research directions of XRL should indeed focus more on problems that are relevant from a real-world standpoint and could develop real XRL framework despite the level of complexity proposed.

Formal verification. Formal verification is the process of obtaining a certification of expected or unexpected state transitions in the state space that could enhance the safety and trustworthiness of RL paradigms through a rigorous mathematical framework [126]. One example of formal verification is given in [127], where a framework called *Verily* considers the space of all possible states in the environment. Other examples such as [127] and [324, 19] propose formal frameworks for verifying NNs and DNNs systems in the context of RL. In [127], verification is done through the concepts of *good* or *bad* states

and through a *liveness* metric defined as $\exists x_1, ..., x_n$ s.t. $I(x_1) \wedge (\bigwedge_{i=1}^{k-1} x_{i+1} \in \epsilon(x_i)) \wedge (\bigwedge_{i=1}^{k} \neg G(x_i)) \wedge (\bigwedge_{i=1}^{k-1} x_k = x_i)$, where $\epsilon(x)$ is the set of all possible states reachable from x in a timestep, $G(x)$ is a given predicate and $I(x)$ is a predicate that is true iff x is a possible initial state of the system. The process of obtaining a formal proof of a model's behaviour is a predominant problem as it can provide and irrefutable statement about its decisions. This allows the model to obtain better transparency and, as a direct consequence, better interpretability.

While the issue of obtaining formal verification is still discussed in the literature, it becomes of paramount importance in tasks such as safe RL, that is obtaining RL systems that account human safety constraints. Some works such as [125] argue that the problem of *safe* systems through DNNs verification is NP-complete. Furthermore, the concept of *safe* RL poses inherent challenges and limitations, as per [76, 164] and [75, 120], but nonetheless safe RL frameworks could greatly benefit from formal verification processes. It remains to date an open challenge where its tractability could also be limited by the inner nature of the problems and, as such, further research must be done in order to better assess established evidence.

Chapter 3

Related Work

Recent studies have explored the use of RL and Deep RL (DRL) in T1DM. RL has emerged as a promising methodology for adaptive clinical decision-making. By framing medical interventions as sequential decision processes, RL offers the ability to optimize long-term outcomes in dynamic and uncertain environments. These characteristics make RL attractive in healthcare contexts where treatments must be personalized, responsive to evolving patient states, and robust to variability in outcomes. However, despite significant progress in applying RL across critical care, oncology, and chronic disease management, two central challenges remain unresolved: safety and interpretability. Without robust safeguards and transparent reasoning, RL models risk reinforcing biases, producing unsafe recommendations, and alienating clinicians and patients who must ultimately trust these systems.

In this chapter, I review existing works relevant to my research on explainable reinforcement learning (XRL) for Type 1 diabetes management. The literature is clustered into five thematic groups: (i) reinforcement learning in clinical decision-making, (ii) causal and personalized treatment regimes, (iii) RL in digital health and diabetes management, (iv) theoretical/model-based RL advances, and (v) reviews and surveys. For each group, I summarize representative studies, extract key insights, and highlight their limitations relative to my focus on chronic disease explainability. It is evident, from current literature, that this approach is indeed interesting and successful.

3.0.1 Reinforcement Learning in Clinical Decision-Making

The majority of early RL applications in medicine emerged in acute care environments, such as intensive care units (ICUs), where decisions must be made quickly, stakes are high, and rich longitudinal data from electronic health records (EHRs) are available. These settings are ideal testbeds for RL because interventions can be framed as MDPs: the patient's physiological state evolves over time, clinicians take actions (e.g., adjusting ventilator settings, administering fluids), and outcomes can be measured over episodes of care. In [156], Liu and colleagues

focus on sepsis, a life-threatening condition with high variability in presentation and treatment response. Using offline RL methods, they propose uncertainty quantification mechanisms to assess the reliability of learned treatment strategies. Importantly, they address the problem of structural bias in RL models, where aggressive interventions appear statistically linked to poor outcomes due to confounding by indication. By applying subspace learning, they attempt to disentangle true causal effects from artifacts of observational data. *Significance*: This study represents a step toward trustworthy RL in medicine, as it acknowledges the importance of uncertainty in policy recommendations. Without such mechanisms, RL risks overconfidence in spurious policies. *Contrast*: This work addresses a different axis of trust. While [156] mitigate uncertainty, they do not make policies interpretable to end-users. In T1DM, patients must not only trust that recommendations are safe but also understand why particular insulin dosing adjustments are made. Thus, I position explainability as complementary to uncertainty quantification.

Another approach is given in [314], where Yu and colleagues use Inverse RL (IRL) to infer reward functions from observed clinical behavior in intensive care unit settings. By analyzing ventilation and sedation strategies, the system attempts to replicate the implicit expertise embedded in clinician decisions. The advantage of IRL is its ability to model complex decision-making without explicit specification of reward functions, which are often difficult to design in healthcare contexts. *Significance*: IRL highlights the potential of learning from expert demonstrations, especially in domains where clinician intuition and tacit knowledge are critical. *Contrast*: IRL serves as a black-box inference mechanism. It uncovers hidden reward structures but does not inherently provide explanations. My XRL approach instead seeks to explicitly communicate decision rationales, bridging the interpretability gap that [314] leave unaddressed.

The study included in [141] proposes an RL-based method for personalized treatment optimization, with emphasis on patient trajectory modeling and adaptive decision-support. It leverages electronic health records to train RL policies while employing safety constraints. This paper is indeed highly relevant in terms of literature contribution to the adoption of RL in healthcare. It demonstrates how RL can learn treatment strategies from historical data, similar in spirit to the approach proposed in this work. However, it lacks a focus on explainability. My work innovates by integrating XRL, giving patients and clinicians interpretable feedback about policy decisions in T1DM.

Conversely, in [118], the authors synthesize RL's potential in medicine for a clinical audience. It outlines basic RL concepts, provides examples of medical applications, and, most importantly, identifies barriers to clinical translation: data scarcity, safety, fairness, and interpretability. *Significance*: The article has pedagogical value but also highlights interpretability as a major stumbling block. Without transparency, RL risks being dismissed as a "black-box" tool ill-suited for clinical workflows. *Contrast*: This contribution directly operationalizes this insight by embedding explainability mechanisms within the RL framework for T1DM, translating a recognized challenge into a concrete design principle.

3.0.2 Causal and Personalized Treatment Regimes

Chronic disease management often requires multi-stage, long-term decision-making. Dynamic treatment regimes (DTRs) formalize this process, specifying treatment decisions at each stage based on patient history and evolving state. RL is a natural fit, but causal reasoning is necessary to avoid biases and ensure generalizability.

In this regard, [320] introduces causal reinforcement learning methods tailored to DTRs. By leveraging causal graphs, the approach ensures that policies respect underlying causal structures, avoiding spurious associations. The proposed algorithms, OFU-DTR and PS-DTR, achieve improved regret bounds by explicitly incorporating causal information. *Significance*: This represents a major methodological advance, merging causal inference and RL to improve both sample efficiency and policy robustness. *Contrast*: My approach shares [320] commitment to personalization but differs in two respects: (i) the author's work remains largely theoretical, whereas I apply XRL in a real-world clinical domain; (ii) the author optimizes regret bounds, while I optimize explainability as a design objective.

On the othre hand, studies like [22] develop causal Bayesian networks to provide counterfactual explanations for cardiovascular risk in adolescent breast cancer survivors. By modeling necessary and sufficient causes, the system generates human-interpretable explanations for risk assessments, supporting shared decision-making. *Significance*: the authors demonstrate the utility of counterfactual reasoning for medical explainability, showing how patients and clinicians can engage with AI outputs in an interpretable way. *Contrast*: My approach extends this logic to RL. Instead of providing explanations for static risk predictions, I embed explainability in sequential treatment recommendations, thereby addressing the dynamic complexity of T1DM.

3.0.3 RL in Digital Health and Diabetes-Oriented AI

T1DM has been a fertile ground for AI applications, particularly in glucose prediction and automated insulin delivery. However, most systems prioritize accuracy and automation, often neglecting explainability.

In [216], the authors review AI-driven approaches to T1D management, focusing on predictive glucose modeling and closed-loop insulin pumps. RL is identified as a promising strategy for adaptive insulin dosing. *Significance*: The paper highlights the technical feasibility of AI in T1D, setting the stage for RL-driven innovations. *Contrast*: While automation is emphasized, patient comprehension is not. My XRL framework addresses this by ensuring that patients understand and can trust automated insulin recommendations.

In [322], the study emphasizes safety in RL applications. Key risks include policy misalignment, poor generalization, and vulnerability to data biases. *Contrast*: This work complements this perspective by addressing not just safety but also interpretability, thereby tackling two of the main barriers to clinical adoption simultaneously.

Another study [138] explores safe DRL in therapeutics. This preprint proposes mechanisms for embedding safety constraints in RL-driven digital therapeutics. *Contrast*: I share the concern for safety but add interpretability as a fundamental design principle, particularly critical in patient-managed chronic conditions.

3.0.4 Theoretical and Model-Based RL Advances

Although much of the healthcare-focused RL literature is application-oriented, theoretical research provides the algorithmic backbone for safe, efficient, and generalizable deployment. Model-based approaches, off-policy evaluation methods, and stochastic RL frameworks contribute to making RL applicable in noisy, high-risk environments such as medicine.

The work proposed in [119], introduce an innovative RL method that operates in a continuous-time stochastic setting. Traditional RL assumes discrete-time MDPs, but in healthcare many systems (e.g., blood glucose fluctuations, hemodynamics) evolve continuously. The authors propose a nonparametric smoothing method combined with temporal-difference learning to approximate value functions when the underlying dynamics are governed by differential equations. Their framework accounts for noisy observations generated from stochastic differential equations, making it theoretically well-suited to domains with irregular sampling or noisy physiological data. This work contributes robustness at the modeling level, showing that RL can be extended beyond simplistic discrete approximations. For conditions like diabetes, where glucose levels evolve continuously and monitoring signals are noisy, such theoretical innovations are highly relevant. In contrast with this contribution,while they advance the mathematical underpinnings of RL, they do not address patient-facing interpretability. My approach applies RL to T1DM in a manner that explicitly prioritizes explainability, bridging the gap between robust theoretical foundations and human-centered applications.

Simlarly, the study [177] focuses on safe deployment of RL in digital health. It proposes hybrid methods that combine model-based planning with model-free RL to ensure safe and stable decision-making in uncertain clinical environments. The authors of this work directly tackle the relevant implications of the emphasis given on safety. The approach strengthens the methodological foundation that this project builds on. The key difference, however, is that the paper prioritizes algorithmic safety guarantees, while my contribution balances both safety and explainability, making the approach clinically transparent for T1D patients and providers.

Another work included in [162] represents a thorough exploration of how reinforcement learning can be applied to optimize treatment strategies using real-world hospital data. The study investigates different RL paradigms, including value-based and policy-gradient methods, applied to retrospective patient trajectories. Key contributions include:

- Algorithmic benchmarking: The thesis compares several RL algorithms

on medical datasets, analyzing their performance in handling sparse and delayed rewards—both common in healthcare scenarios.

- Handling observational bias: By carefully preprocessing and structuring clinical data, the work highlights the challenges of confounding and treatment assignment bias, an issue mirrored in many large-scale EHR studies.

- Evaluation frameworks: Since deploying RL directly in clinical environments is infeasible, the thesis dedicates attention to off-policy evaluation (OPE), assessing how well RL models perform using historical patient data without online experimentation.

This workprovides a detailed case study of RL in healthcare from a practical perspective, bridging the gap between theory and application. It demonstrates the feasibility of RL for treatment optimization while also revealing critical obstacles such as data bias, sample inefficiency, and evaluation complexity. However, while the author emphasizes algorithmic benchmarking and evaluation strategies, the thesis does not address interpretability or transparency. My work extends beyond algorithmic feasibility to propose explainable reinforcement learning for T1DM, directly tackling the interpretability and trust barriers that the author acknowledges but does not solve.

Lastly, the work in [286] focuses on the reliability of evaluating RL policies without online deployment. They conduct an empirical study of off-policy evaluation methods, benchmarking their ability to provide unbiased and low-variance estimates of policy performance. Such methods are indispensable in healthcare, where testing unverified RL policies on patients is unethical. While these methodologies strengthen the safety of RL by ensuring that policies are vetted before clinical use, my contribution lies in making RL understandable rather than merely evaluatable.

3.0.5 Summaries and overviews

An overview of the various RL and DRL algorithms used in T1DM is included in [265] and [312], where thorough comparisons of deep Q-learning studies in the field is provided. The cited examples include the use of noisy DQN for certain experiments and dueling DQN for others. The authors assess that such architectures outperform the classical bolus treatment without DRL. Another example is offline RL such as [63] and [64], where human feedback is used to obtain flexible insulin dosages suited on given patient records. In this framework, the authors obtained fine-tuned RL agents that were able to solve multi-task problem of adapting to meals and habits of the patients. Similar approaches are also presented in studies like [298], [325] and [72], where efficiency and accuracy are measured by applying RL and DRL methods to T1DM. These studies further affirm evidence that such agents are capable of solving these types of tasks.

For instance, the review proposed in [13] comprehensively examines RL applications across multiple domains, including critical care, oncology, mental

health, and chronic disease management. The authors underscore four recurring challenges: Safety - Ensuring RL policies do not recommend harmful interventions; Interpretability - Overcoming the "black-box" problem of deep RL models; Bias - Addressing confounding and imbalances in observational healthcare data; Validation - Establishing robust clinical evaluation frameworks, including offline policy evaluations and simulation environments. The review concludes that RL's clinical adoption depends on not only technical performance but also alignment with clinical workflows and patient-centered values. In contrast with this work, which directly addresses interpretability and patient-centeredness in the context of T1DM, they design policies that are both adaptive and explainable. My contribution tackles two of the most prominent obstacles highlighted by the review, thereby pushing RL one step closer to safe and trustworthy deployment in everyday clinical practice.

Another work proposed in [82] discusses how machine learning can optimize the management and organization of healthcare records, focusing on improving efficiency, accessibility, and decision-making support through automated classification and record handling. Relevant to this work only in terms of infrastructure, it is accurate at record management, which is essential for RL policy training. However, it does not contribute to interpretability or decision-making frameworks. My approach directly targets clinical treatment optimization and explainability in T1DM, while this paper is more about data logistics.

However, while these approaches generally succeed in solving the problem of T1DM, they do not account for the interpretability of the models, which is a crucial aspect for healthcare applications. This very likely translates into opaqueness, thus hindering proper accountability of both the algorithm and the adoption of such techniques in T1DM. A successful approach in building trust and transparency into RL applications is the use of Shapley values, which are borrowed from game theory and applied in the RL domain. The approach of Shapley value has been widely adopted in the field of XAI and it was verified in the RL domain as well. Studies such as [291] and [290] apply the concept of distribution of the Q-value function to identify each agent's contribution to the global reward. These papers address the credit assignment problem in cooperative multi-agent RL by introducing the extended convex game framework and proposing the Shapley Q-value, a local reward approach that better reflects each agent's contribution than shared global rewards. They develop a new algorithm, Shapley Q-value deep deterministic policy gradient and SHAQ, which demonstrate improved convergence and fairer credit distribution in benchmark tasks compared to existing methods. As these works offer great insight in the broader domain of RL applicability, they do not address the problem of T1DM directly as their scope is focused on Shapley Q-values only. Furthermore, their work is focused on cooperative multi-agent settings, which have intrinsic different mechanics from single-agent scenarios as the one addressed by this work.

As of XRL applied to healthcare, the work provided in [12] addresses challenges in enabling efficient local and distributed inference of deep neural networks. It proposes a novel online pruning strategy that uses XAI to dynamically prune networks without retraining, overcoming data scarcity and bias issues in

local healthcare environments. By combining XAI with RL, the agents adapt to changing system conditions. Experiments show this method outperforms existing pruning techniques by reducing resource consumption while maintaining high accuracy in distributed healthcare inference tasks.

Even if AI, RL and DRL have demonstrated wide adoption also in the context of healthcare and T1DM, at the current state of the art, the field of XAI and XRL in diabetes is not stressed enough. Indeed, there still are open challenges in order to implement a full transparency-driven closed-loop control which is accountable and trustworthy by all the parties. In this context, fully automated closed-loop control in T1DM is still an open challenge, and is crucial for automating manual intervention and minimizing risks associated with it. The final aim of building an artificial pancreas would include an automated system that continuously monitors BG levels and adjusts insulin delivery in real time to maintain glucose within a target range. It would integrate a glucose sensor, an insulin pump, and a control algorithm to mimic the body's natural insulin regulation. It is, thus, of paramount importance to the algorithm regulating such cycle to be held accountable and transparent enough so that each step of the decisions of the agents are explicitly defined.

Towards interpretability in diabetes there are works such as [134], where the authors show how it reveals feature influences in specific patient groups that global interpretation might miss. By comparing three models such as gradient boosting machine, random forest, and generalized linear model with regularization, no significant differences in prediction accuracy were found, but local interpretation identified additional factors like depression and smoking as important in certain subgroups. The study emphasizes that model selection should consider interpretability, not just prediction performance, especially in personalized healthcare. Another example of XRL in diabetes is [100], where a mobile app, XAI4Diabetes, is developed to address interpretability. The authors built a multimodule platform that combines machine learning, knowledge graphs, and ontologies to predict diabetes risk and explain its predictions. It includes modules for knowledge base, matching, prediction, and interpretation, and a user study showed it improves understanding of machine learning predictions in real patient scenarios. While these applications are framed correctly in terms of the field of study, both don't account into applying critical-reasoning algorithms such as single-agent RL automata and focusing on explaining its decisions.

Conversely, in T1DM a different approach appears in [151], where the authors implement a soft actor-critic controller to maintain BG levels in target range. In such work, to address interpretability, a random forest regressor and a dual attention network were utilized to extend the state variables and later map them to the actual predictions of the model. Although the work proposes a validated solution, the study is limited to few methods and other types of explainability techniques should be taken into account.

Another example of XRL in diabetes is [179], which is a study comparing three different methods of interpretability. The authors used the concept of reward decomposition and belief maps to obtain better explainability and they analyzed the results through questionnaires to T1DM patients and physicians.

For similar reasons, this study is interesting from an explainability standpoint but it suffers generalization by adopting only the three methods included. Furthermore, the study does not explore how these interpretability methods perform under varying clinical scenarios or across different patient profiles, limiting its applicability in real-world decision support systems.

Taken together, the literature reveals significant advances in RL methodology, clinical application, and safety assurance. However, explainability still remains underexplored. Acute care studies prioritize uncertainty and robustness, causal RL works emphasize personalization, and diabetes-specific AI emphasizes automation. My contribution integrates these strands by embedding explainable reinforcement learning into T1DM, ensuring safety, personalization, and interpretability in one framework. Thus, to help the current state of the art bridge this kind of gap, in this work I pioneer the integration of Structural Causal Models (SCMs) derived from Shapley Q-value attributions of Deep Q-Network (DQN) internals, aiming to advance the frontier of interpretability in T1DM management. By mapping the agent's learned value structure into an explicit causal graph, I not only provide transparent explanations of its decision-making process but also enable principled reasoning about counterfactuals and intervention effects, capabilities critical for building trust in clinical applications of RL. The approach is described in better detail in the following sections.

Chapter 4

Applying XRL to T1DM

As RL systems continue to evolve and find application in increasingly sensitive domains such as healthcare, the demand for explainability becomes not only a matter of usability but also of ethical and regulatory importance. Explainability in this context serves two essential roles: first, to make agent behavior interpretable and meaningful to human stakeholders, and second, to deepen the understanding of the algorithmic processes that govern autonomous decision-making. These complementary goals inform the structure of this chapter, which is organized around two principal axes: *Explaining to People* and *Explaining the Algorithms*.

The *Explaining to People* section focuses on generating human-understandable narratives of agent behavior, tailored to users such as clinicians, patients, or system operators. Here, the emphasis is on developing model-agnostic and user-centered techniques (such as reward-difference analysis and contrastive explanations, along with post-hoc explainability) that help external observers make sense of RL decisions without requiring deep technical expertise. These techniques aim to enhance the trust, transparency, and accountability of RL systems in real-world settings, especially in high-stakes contexts like T1DM management.

In contrast, the *Explaining the Algorithms* section turns the lens inward, examining the internal mechanisms and latent representations that drive agent behavior. It investigates methods that are less concerned with user accessibility and more with theoretical clarity and diagnostic insight. In particular, it explores Shapley-based Q-value attribution for Deep Q-Networks (DQNs), and introduces a novel integration with Structural Causal Models (SCMs) to capture causal relationships underpinning policy decisions. This deeper layer of analysis not only informs the development of more robust and interpretable RL agents, but also reveals important limitations in conventional attribution-based explanation techniques.

Together, these two approaches form a comprehensive exploration of explainability in RL, one that spans the spectrum from user interaction to algorithmic transparency. By addressing both sides of the XRL challenge, this chapter con-

tributes to the overarching goal of aligning AI behavior with human expectations and values: an essential step toward responsible and trustworthy deployment in clinical environments.

4.1 Explaining to People

In the context of clinical applications such as T1DM, adopting interpretability in RL is not just an academic preference—it is a practical necessity. When decisions involve human health, particularly in conditions that demand constant and precise management like T1D, the consequences of algorithmic recommendations are deeply personal and often immediate. Here, interpretability serves a crucial role: not to demystify the internal workings of the algorithm per se, but to make its outputs understandable, relatable, and justifiable to the people who are affected by them. Patients, caregivers, and clinicians alike must be able to grasp the reasoning behind the actions suggested by a reinforcement learning agent. Without this layer of communicative transparency, trust in automated decision-making systems can erode quickly, regardless of their performance metrics.

This form of interpretability focuses less on the inner mechanics of the model and more on rendering its behavior accessible to human reasoning. It addresses the essential question of why a particular decision was made, in a language and format that resonate with human experience and medical understanding. In doing so, it empowers users to integrate algorithmic insights with their own judgment, rather than passively accept opaque outputs. This chapter is dedicated to that human-centered dimension of interpretability—setting the stage for the following technical exploration of how the algorithm itself works—because building trust and understanding among users is foundational to the safe and effective deployment of reinforcement learning in sensitive, real-world settings like T1D care.

4.2 Contrastive Explanations

Contrastive explanations offer a compelling route to model transparency because they mirror how people naturally reason about decisions. When people try to understand a choice, especially one made by a complex system like a ML model, they often don't just want to know why it made that choice. They want to know why it made that choice instead of another. This "why A rather than B?" framing is the essence of contrastive explanation. It's not simply about identifying influential features, as many traditional interpretability methods do, but about articulating what would have had to change in the input to yield a different outcome.

This approach creates more intuitively satisfying explanations. By focusing on what minimal or critical differences would flip the decision, contrastive explanations can surface the structure of the model's decision boundary in a way

that is easy for humans to grasp. Unlike methods that merely highlight input regions or assign scores to features, contrastive explanations provide counterfactual narratives, stories about what could have been, which resonate more strongly with how diagnose, debug, and learn are done.

Because of this alignment with human intuition, contrastive explanations often lead to deeper trust and clearer insight, especially in settings where understanding the "why not" is just as important as understanding the "why."

Contrastive methods highlight differences between outcomes or scenarios, clarifying why a model made a specific prediction by comparing it to alternative possibilities. Use of contrastive explanations is found in works such as [277], where the dynamics of an RL agent are explained in terms of expected consequences. This technique first builds a set of user-friendly descriptions, queries the policy and its beliefs and, finally, outputs explanations in a human-readable format. Although this approach is interesting and valuable, it requires to have images as agent's inputs, hence it is not applicable here. Another example is [291], where explanations are generated by constructing Shapley Q-values in Multi-Agent RL (MARL) systems. Unfortunately, dynamics of MARL systems are not trivially translated to single agent ones, which is not the case of T1D management.

Other works such as [122] and [66] introduced, instead, key concepts such as Reward Difference eXplanations (RDX) and Minimal Sufficient eXplanations (MSX). The authors in these work build explanations by means of differences in reward values, which highlight what are the expected future return values for each action at a given state. These form of explanations can greatly help in increasing the comprehension of how outcomes could change based on different states, and makes this approach ideal towards better interpretability in T1D management via XRL. Another interesting work is [311], where Belief Maps (BMs) were first introduced as a form of contrastive explanations. Analogously, this technique builds a visual representation of the perceived value of future state transitions which is consistent with the Q-value map of the agent. This method not only allows to easily visualize agents beliefs about future states but also helps understand the learning mechanism by visualizing agent's best choice for a given situation.

In this work, three contrastive explanations are employed alongside the RL algorithm in charge of explaining T1D mechanics, namely Decomposed Q-values (DQv), MSX / RDX graphs and BMs. Results show that BMs and MSX / RDX graphs, as forms of visual explanations, are usually perceived as better explanation quality and trust towards the automated process. This highlights that contrastive or visual explanations are on average preferred over no explanations or other non-visual forms of explanations. In this section, explanations are employed as follows: firstly, a mathematical formulation of the problem will be given; then, the problem will be stated as an RL statement; finally, three contrastive XRL methods (DQv, RDX and BMs) are described.

4.2.1 Problem Formulation

The problem of closed-loop glucose control in T1D management can be formulated as a continuous and infinite-state Markov Decision Process (MDP) by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma \rangle$, where $\mathcal{S}$ is the set of possible states, $\mathcal{A}$ is the set of possible agent's actions, $\mathcal{P}$ is the probability function of transitioning from a given state s to a state s', R is the reward function and $\gamma \in [0, 1]$ is the discount factor for future states. At each time step t, an observation (i.e. from the CGM measurements) is given by the environment that forms the state $s \in S$. The agent then chooses an action $a \in A$ to be performed, the environment then transitions into a new state $s' \in S$, following the function $\mathcal{P}$, and gives the agent a real-valued reward $r \in R$. What action to choose at a given state is defined by the agent's policy π. The aim of the agent is to maximize the expected future reward $r_t = R(s_t, a_t)$ given by the environment at each time step t. Generally speaking, the Q-Value function can then be defined by:

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}[\sum_{k}^{\infty} \gamma^{k} r_{t+k+1} | s_t = s, a_t = a], \tag{4.1}$$

and its optimal function by $Q^*(s, a) = max_{\pi} Q^{\pi}(s, a)$, which can be found by solving its corresponding Bellman equation. In T1D management, the set of possible states is the set of observed CGM measurements, actions of the agent are to give basal insulin, bolus insulin or no bolus at all.

4.2.2 Environment Setup

The work here presented is based on the UVA / Padova simulator [171], which is a glucose-insulin dynamics simulator, jointly developed by the University of Virginia (US) and the University of Padova (Italy), and which has gained FDA acceptance for pre-clinical studies. The implementation of the simulator used is based on the library simglucose [1], which also supports the Python interface provided by OpenAI Gym [2]. This version of the simulator provides 30 in-silico patients (10 children, 10 adolescents and 10 adults) that follow the equations of glucose kinetics defined in [171] (S2008 version). Observations consist of CGM measurements while agent's actions is a basal insulin dose varying in the interval $[0, 30]$ units. The reward function was structured to be 1 in case of normal BG levels and -2 elsewhere, that is BG $> G_h$ or BG $< G_l$, where G_h and G_l are fixed maximum and minimum glucose levels, respectively. Here, constant values of $G_l = 70$ mg/dl and $G_h = 230$ mg/dl are set as the bounds for the hypoglicemia and hyperglicemia, respectively. Thus, the goal of the agent is to maintain, as most continuously as possible, the BG levels in the control zone, i.e. always over G_l and under G_h. The observations of the agent consist of a tuple (BG, CHO), where BG is the blood glucose level coming from CGM measurements and CHO are the current carbohydrates (CHO) intake announced by the user.

[1] `https://github.com/jxx123/simglucose`
[2] `https://github.com/openai/gym`

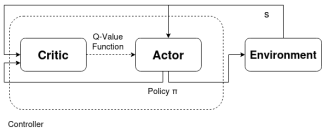


Figure 4.1: Overview of the AC controller as in [49], where π is the control policy and s is the state the environment transitions to. The Actor is in charge of policy improvement and aims at maximizing average cost α, while the Critic is in charge of policy evaluation and outputs the Q-Value function. Both components interact with each other in a push-and-pull manner towards the main objective (T1D optimal control).

4.2.3 T1D Management through RL

As cited above, online and offline RL algorithms have made great impact on closed-loop control of T1D thanks to their adaptability to new situations without continuous supervision. In order to demonstrate its capability, an optimal Actor-Critic (AC) method for solving T1D is taken into account as an example of a black-box RL algorithm. The functions employed in this work are the ones presented in [49]. The control policy π of the agent is formulated as a parameterized conditional probability $\mu_\theta(s'|s,a)$ with parameters θ. To obtain optimal control, consider the following average cost function:

$$\alpha(\theta) = \sum_{s\in S, a\in A} c(s,a)\eta_\theta(s,a), \tag{4.2}$$

where $c(s,a)$ is a cost function describing the quality of each assigned state $s \in S$ and $\eta_\theta(s,a) = \pi_\theta(s)\mu(a|s,\theta)$ is a function representing the Markov chains with stationary probabilities $\pi_\theta(s)$.

Then, the gradient method is employed in order to maximize the Q-value function:

$$\nabla_\theta \alpha(\theta) = \sum_{s,a} \eta_\theta(s,a) Q_\theta(s,a)\psi_\theta(s,a), \tag{4.3}$$

where $\psi_\theta(s,a) = \nabla \ln(\mu_\theta(s|a))$.

AC algorithms are composed of two components: an *actor*, who is responsible for the approximation of the optimal policy π and the *critic*, that is responsible for the approximation of the Q-Value function $Q(s,a)$. Fig. 4.1 represent an high-level overview of AC method used for diabetes management. The algorithm used in this work is the same as described by the authors but only the basal version of the algorithm is employed. This is because, in optimal situations, the RL environment should work solely by basal insulin and without active bolus declarations by the patient. Similarly to [49], functions $F_1 = max(G_{max} - G_h, 0)$ and $F_2 = max(G_l - G_{min}, 0)$ are built analogously. The cost associated to each day is defined by $c(s_t) = a_h F_1^t + a_l F_2^t$, where $a_h = 0.01$ and $a_l = 0.1$ are two parameters used for scaling the two features. The same considerations of [49] are followed for what concerns learning rates and hyperparameter settings. Profiles $P1$, with hourly variable basal rate and $P2$, which uses a constant basal insulin

infusion are built accordingly. $P1$ has a basal rate of $0.5B/24$ in hours ranging from midnight to 04:00, $1.5B/24$ from 04:00 to 10:00 and $1B/24$ from 10:00 to midnight, while $P2$ has a constant $B/24$ at each hour. Both profiles were chosen in this scenario in order to compare the explanations generated by the different methods at a given time step t.

To obtain optimal T1D treatment, this algorithm expects to have both basal and bolus, whereas in this work only the basal insulin is used. Nonetheless, this approach remains valid because the primary aim of this work is not to find the optimal T1D treatment but to highlight and clarify the inner workings of the methods using XRL. The primary focus on explanation, rather than optimization, helps build trust and transparency in the technology from a user's standpoint.

4.2.4 Intepreting RL agents

In this work, in order to explain the RL algorithm described above, three constrastive XRL methods are employed to interpret T1D management. The first method employed here is Decomposed Q-values (DQv), introduced in [122], which explains the state-value function as a composition of reward and Q-values differences. This presents a starting form of contrastive explanation and is considered here as a sort of benchmark of explanation quality and trust. One can expect that mathematical explanations of the Q-value function offers little to no insight on what happens in the background. Results indeed tend to agree to this sentiment but they are nevertheless considered as helpful comparison.

The second method considered here are the grouping of RDX and MSX explanations. RDX is obtained as a difference between Decomposed Q-vectors $\Lambda(s, a_1, a_2) = \overrightarrow{Q}(s, a_1) - \overrightarrow{Q}(s, a_2)$. From the concept of RDX, one can derive the MSX, which is a tuple (MSX^+, MSX^-) composed of, respectively, the positive and negative reasons of preference for actions $a_1, a_2 \in A$ in state s. Mathematically, the positive feature of the MSX is defined by the following

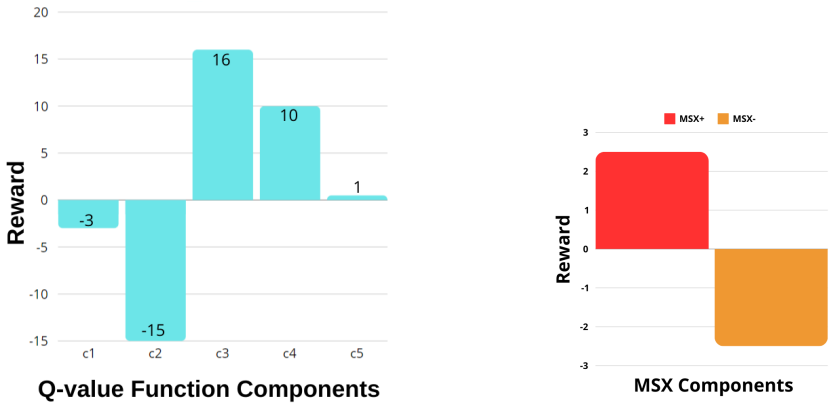
$$MSX^+ = argmin_{M \in 2^C} |M| \quad \text{s.t.} \sum_{c \in M} \Delta_c(s, a_1, a_2) > d. \tag{4.4}$$

Here, C is the set of *reward components / types*, d is the disadvantage function of selecting a_1 over a_2, $\Delta_c(s, a_1, a_2)$ is a component of the RDX w.r.t. the reward type c. Here, a choice of $c = 6$ was made in order to represent what could naturally be divded into reward components for diabetes dynamics, i.e. glucose and insulin kinetics, insulin action on production, insulin action in the liver and subcutaneous insulin kinetics. The sets M are the sets of the smallest size composed of the positive reasons that overcomes the disadvantage.

Similarly, the negative feature of the MSX is defined by:

$$MSX^- = argmin_{M \in 2^C} |M| \quad \text{s.t.} \sum_{c \in M} -\Delta_c(s, a_1, a_2) > v, \tag{4.5}$$

where v is the just-insufficient advantage as defined in [122]. The decomposition of the Q-values this way is considered here as a form explanation based on the



(a) (b)

Figure 4.2: MSX and RDX representations of the Decomposed Q-values form critic's perspective. Fig. 4.2a represents a sample of a RDX graph. c_1 through c_5 are the components of the reward function from critic's perspective. The components are built upon reward signaling and could represent inner insulin kinetics. Fig. 4.2b represents a sample of a MSX graph from critic's perspective. The actions compared in this case are: no basal insulin or amount of basal insulin suggested by the control policy. Both sample refer to the adolescent n.6 at h. 17.00.

mathematical representation of the inner mechanics. For the problem addressed in this work, it serves as a sort of proper XRL contrastive baseline method.

Oftentimes, however, graphical explanations are usually preferred and commonly perceived more useful in terms of understanding and trusting the process. So, the RDX and MSX graphical interpretations are taken into account as well.

The last method considered in this work is the concept of BMs introduced in [311]. A *Belief Map* (BM) is a map of discounted expected states $\mathbf{H}(s_t, a_t) \in \mathcal{R}^{S \times A}$, which defines the expected discounted sum of perceived advantage by the agent for visiting future states. In the classical Q-Learning setting, this function can be defined as

$$\begin{aligned} \mathbf{H}(s_t, a_t) \leftarrow \mathbf{H}(s_t, a_t) + \alpha[1_{s_t, a_t} + \\ \gamma \mathbf{H}(s_{t+1}, argmax_{a \in A} Q(s_{t+1}, a) - \mathbf{H}(s_t, a_t)], \end{aligned} \tag{4.6}$$

where α is the learning rate, as commonly used in Q-Learning applications, and $1_{s_t, a_t} \in \mathcal{R}^{S \times A}$ is a binary indicator function of 1 at position (s_t, a_t) and 0 elsewhere. Thus, Decomposed Q-values, MSX and RDX graphs and BMs are used for the comparison of explanation in the proposed environment.

4.2.5 Evaluation Metrics

In order to assess the significance of the given explanations, quality and trust were evaluated separately. A one-way ANOVA test was made for both with

the following null hypothesis: H_0 is such that every explanation type gives no influence at all $H_0 : \mu_{DQv} = \mu_{RDX} = \mu_{BM} = \mu_N$, where DQv represents the Decomposed Q-values mathematical explanations, RDX the explanations given by MSX/RDX graphs, BM is the explanations given by BMs and N is a simple verbal description of the RL environment without any form of explanation generation. Then, a pairwise Tukey comparison was made in order to better understand if the population means of the votes were statistically significant for each of the proposed metrics. For the evaluation of quality, the metrics used for the comparisons are: understandable, satisfying, sufficiently detailed and complete. On the other hand, to investigate if explanations promote trust towards the environment, other metrics are used: confidence, predictability, reliability and safety.

4.3 Post-hoc Explanations

Having previously explored how transparency can be designed directly into RL systems—through the use of inherently interpretable models and architectures—[178, 266], I now shift the focus to a different, yet equally important, approach: the use of post-hoc interpretability methods. While inherently transparent systems offer clarity by construction, they often come at the cost of model complexity, flexibility, or performance, particularly in high-dimensional, dynamic environments such as those encountered in T1DM. Methods like [298] and [64] are few examples of employing both online and offline RL algorithms that can successfully maintain the BG levels in target.

In such cases, more expressive models, like deep RL agents, may be preferred for their ability to capture subtle patterns and adapt to complex patient behaviors [182]. However, these models are typically opaque, which brings us back to the core concern of interpretability. One example is Shapley Q-values, which are used to obtain better understanding of the relationships between random variables perceived in the environment by the agents [291]. Another example is the work in [179], which addressed the problem of interpreting T1DM via transparency methods. In particular, the study uses belief maps along other non-visual methods in order to provide explanations of actor-critic agent's decisions. In addition, works such as [208] focused on applying inherent natural language explanations generated in text-based game environments to implement an hierarchically-explainable RL agent.

However, despite their promise, transparency methods face significant limitations. Transparency is tightly coupled with the complexity of the underlying agent or model. While transparent models provide an intrinsic window into decision processes, their explanations often suffer from the same cognitive complexity as the models themselves. That is, transparent explanations can be too detailed or technical for human users to intuitively understand, especially in high-dimensional or deep learning-based agents. This creates a paradox where transparency does not necessarily translate into effective interpretability.

Conversely, post-hoc explainability techniques attempt to bridge this gap

by generating simplified, human-understandable explanations after the model is trained, without requiring the model itself to be inherently transparent. These methods, such as surrogate models [146], feature attribution techniques [256], or natural language generation [166, 60], aim to provide actionable insights that are more accessible to users while still maintaining fidelity to the original agent's behavior. Examples of works in which post-hoc is used to address practical application include the use of XRL methods for categorization of intrusion detection [146], or in the case of portfolio building [51].

Cases where post-hoc explainability reaches its best expressions are when explanations are visualized through methods like Probabilistic Graphical Models (PGMs) [251]. For example, in [53], the authors propose an algorithm-agnostic method using PGMs to generate human-understandable explanations for RL agents. Such explanations, generated through the use of PGMs, have demonstrated their applicability towards the construction of accountable, reliable and trustworthy RL agents. Another example is [137], where the interpretability of recurrent neural networks is improved by combining them with hidden Markov models, which are simpler and more transparent than the original models.

In such works, explainability is the main focus as it enhances operator trust and regulatory compliance, paving the way for accountable, reliable, and trustworthy RL systems in real-world applications. In particular, in the context of PGMs, there are a subset of such graphs called Markov networks which operate on a random field and are built upon the structure of the considered variables. Works such as [205] have demonstrated their applicability in explainability contexts. The authors propose a novel, feature-dependent Markov chain-based system for forecasting complex demand patterns through relevance and informativeness metrics. Those methods become important when they can bridge the gap between advanced AI models and domain experts' understanding, promoting adoption across business functions while handling high-dimensional data effectively.

In the purpose of this work, the use of Markov networks as means of obtaining better interpretability is explored in the domain of solving T1DM management through a deep RL agent. The contribution of this work is focused on highlighting how post-hoc Markov networks are able to enhance qualitatively the explanations, in the context of healthcare. This is discussed in the following sections.

4.4 Explaining Using Markov Networks

In order to achieve better explainability in healthcare contexts such as T1DM, the problem formulation is first stated and then the Markov network graphs are then introduced. To explain the DQN agent, Markov networks are used in this work to model the dependencies between various health-related variables, such as BG levels, insulin sensitivity, meal timing, and other factors that are crucial for managing diabetes. Markov networks are undirected graphical models based on the concept of *random fields*, which satisfy the locality property: $P(F_i|\mathbf{F_c}) =$

$P(F_i|Nei(F_i))$, where $\mathbf{F_c}$ is the set of all random variables in the field except F_i and $Nei(F_i)$ are the neighbors of F_i.

Each node in the Markov network represents a variable (e.g., insulin dose, insulin action, etc.). The edges (or lack thereof) represent dependencies or conditional independencies between those variables. The absence of directed edges indicates that the model assumes no direct causality between two connected variables, but rather a mutual influence. Based on the variants of the glucose kinetics, insulin kinetics, meal model and insulin action mechanics of the simulator, three different graphs are constructed. The labels to be assigned to the Markov networks are described in Tab. 4.1.

Letter	**Description**
A	Insulin Action
B	Insulin Concentration
C	Glucose Absorption
D	Plasma Glucose
E	Insulin Sensitivity
F	Noise/Disturbance
O	Insulin Dose

Table 4.1: Association between Markov variables and alphabet letters.

Given the availability of the data and the arbitrariness of constructing such causal models starting from variables in the random field, three graphs are constructed and used as means of explanations. The first graph is called a *Full Model* and represents the complete mechanistic model of T1D dynamics: Insulin (B, E, A) and glucose (C, D) subsystems are fully integrated, external factors like meal absorption (C) and noise/disturbances (F) are explicitly modeled and the insulin dosing decision (O) is influenced by both physiological signals and external noise. This graph is best for explaining biologically accurate simulation, system identification and best represents the DQN agent. The graph is depicted in Fig. 4.3.

The second graph is focused on a latent control-oriented abstraction and is called the *Reduced Model*. This version simplifies the system by removing explicit environmental inputs (like meals and disturbances), focusing instead on: the internal insulin pathway: E $\rightarrow$ A $\rightarrow$ D; how internal physiology and glucose levels influence the dose decision (O). This use case is ideal for model-based control, robust policy design, or testing agent generalization when noise is abstracted. This graph is represented in Fig. 4.4.

The last graph is called a *Decision-Centric* model in that it is designed around decision logic, prioritizing: what the agent can observe (D, A); what it can act on (O); key latent contributors (E, C), while collapsing or skipping intermediary signals. It assumes the agent can reason over a simplified state for dosing and is tailored to explain hte behavior of the DQN where only actionable or observable variables are kept for efficient learning. This kind of graph is depicted in Fig. 4.5.

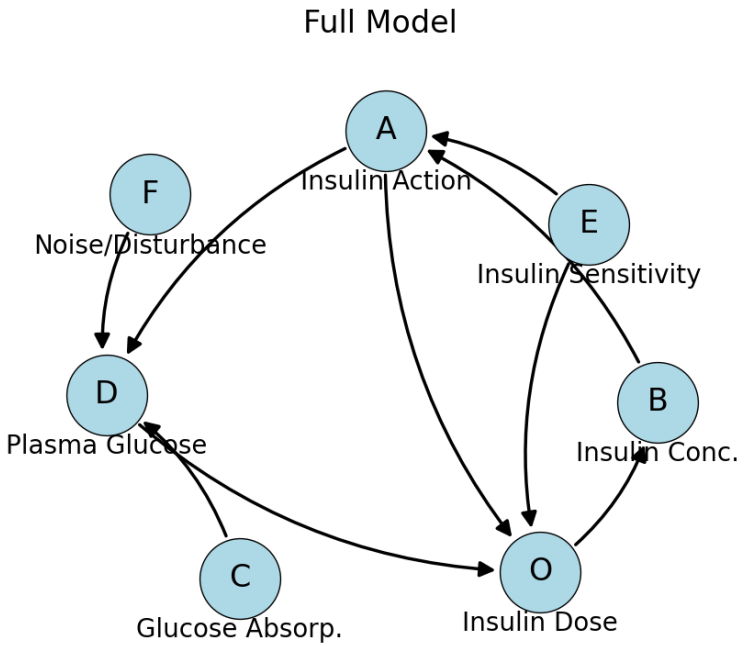


Figure 4.3: Comprehensive causal graph representing the physiological interactions of the patient model. The network includes insulin sensitivity (E), insulin concentration (B), insulin action (A), glucose absorption (C), plasma glucose (D), disturbance/noise (F), and insulin dosing decision (O). This model captures the full endogenous and exogenous dynamics used.

4.5 Summary

Managing chronic conditions like T1DM requires continuous, nuanced decision-making that balances multiple dynamic factors, including glucose levels, insulin dosing, diet, and physical activity. Deep Reinforcement Learning (DRL) offers great promise in this context, enabling adaptive and personalized treatment strategies through iterative learning. However, a persistent barrier to clinical adoption remains: the lack of transparency in decision-making, which is crucial for user trust and safety.

In this work, the role of XRL methods—both integrated and post-hoc—in enhancing transparency in a DRL-based T1DM treatment system using the UVA/Padova simulator is explored. For transparency, an Actor-Critic algorithm is employed and explanations are generated through various techniques: Decomposed Q-values (DQv), MSX and RDX graphs, Belief Maps (BMs), and hybrid probabilistic frameworks that integrate Markov Networks with Deep Q-

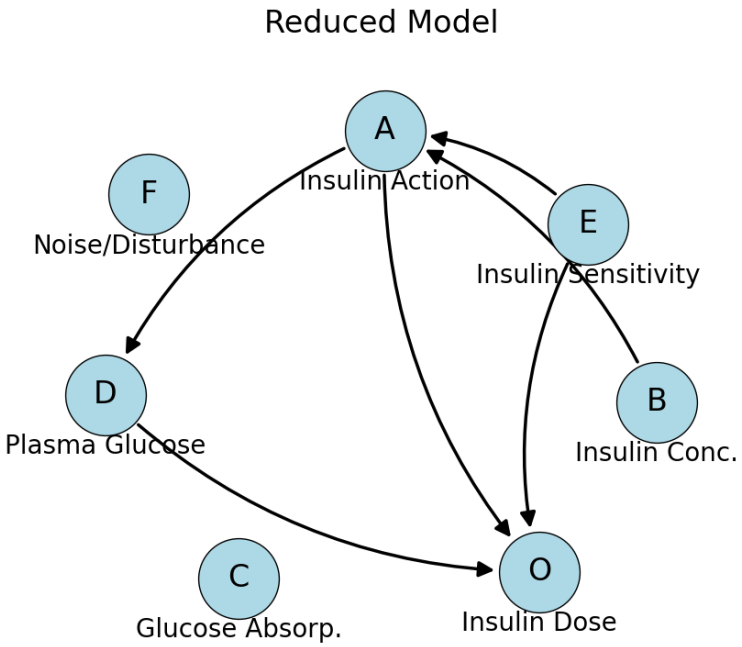


Figure 4.4: Simplified causal structure focusing on the internal insulin-glucose regulation loop. External factors like meal absorption and disturbances are excluded, highlighting latent physiological pathways (E $\rightarrow$ A $\rightarrow$ D) and their influence on insulin dosing (O).

Networks (DQNs).

To assess the impact of these methods, two user studies were conducted involving real T1D patients and clinicians. In total, 60 participants evaluated the system's transparency, quality, and trustworthiness through structured questionnaires. Results show that visual and post-hoc explanations—such as MSX, RDX, BMs, and probabilistic graphs—consistently increased user trust and understanding, with a 28% improvement in willingness to adopt the system. Notably, methods that visualize interdependencies between physiological and behavioral variables provided the most actionable insights. These findings underscore the necessity of explainability in AI-driven healthcare tools—not only to validate models clinically, but also to engage users meaningfully.

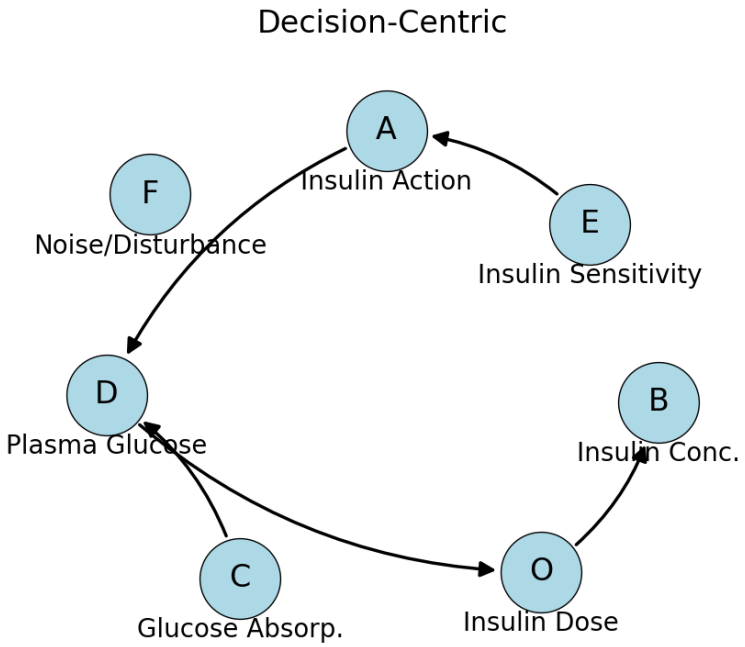


Figure 4.5: Control-oriented graph emphasizing observable and actionable variables for decision-making. Retains essential contributors to plasma glucose (D) and insulin dynamics, while abstracting or omitting exogenous variables.

4.6 Explaining the Algorithms

Type 1 Diabetes Mellitus (T1DM) is a chronic autoimmune condition, that requires lifelong management. It occurs when the immune system mistakenly attacks the insulin-producing beta cells in the pancreas, leading to little or no insulin production [105]. Diabetes, both type 1 and type 2, can be considered an important issue as of today, due to its rapidly increasing prevalence worldwide [272].

Without a proper management, T1DM can lead to severe short- and long-term complications, including diabetic ketoacidosis, heart disease, nerve damage, kidney failure, and vision loss [168]. T1DM is characterized by inadequate insulin production by the pancreatic β-cells, necessitating the administration of exogenous insulin [45]. Standard insulin replacement therapy for T1DM involves a bolus of fast-acting insulin to address the rapid glucose rise following meal consumption, as well as the administration of slow-acting basal insulin to maintain glucose levels within the target range during fasting conditions. By maintaining Blood Glucose (BG) levels within a target range through a com-

bination of insulin therapy, regular monitoring, healthy eating, and physical activity, individuals can reduce the risk of complications and lead full, active lives.

In this regard, the best way to achieve optimal T1DM is to handle, as long as possible, the factors that are out of patients' control, such as BG and insulin levels during sleep, or during a long period after digestion. To obtain such management, automated insulin delivery systems have been proven successful for a wide variety of patients over recent years [221]. Furthermore, by using Continuous Glucose Monitors (CGMs) and insulin pumps, automated management of T1DM have reached satisfactory results in both reliability and accuracy, in a drive towards artificial pancreas [136].

Indeed, the growing abundance of medical data and significant improvements in computational capabilities have propelled the use of machine learning, especially deep learning, in numerous healthcare applications that were previously unattainable [116, 44]. The prominence of Artificial Intelligence (AI) algorithms in this field is evident in works where artificial neural networks [209], recurrent neural networks [38] or convolutional neural networks [149] were employed for glucose prediction or forecasting, fostering the use of AI in critical applications such as T1DM management.

The adoption of AI algorithms in healthcare applications offers significant potential, but it comes with challenges related to the interpretability of AI decisions. In this regard, eXplainable AI (XAI) offers a set of techniques in the field of artificial intelligence that aim to make the decision-making processes of AI systems more transparent and understandable to humans [160]. The goal of XAI is to ensure that AI models, which are often considered *black boxes*, can provide clear and interpretable explanations for their predictions or actions [223]. This is especially important in fields like finance law or healthcare (which is the focus of this work), where understanding AI decisions is as fundamental as obtaining accurate results, for reasons of trust, accountability, and regulatory compliance.

Moreover, Reinforcement Learning (RL), a subfield of AI, tackles a more complex class of problems: those involving sequential, adaptive decision-making under uncertainty. Unlike traditional AI methods that passively analyze data, RL actively learns by interacting with an environment and receiving feedback in the form of rewards or penalties [123]. This makes it especially well-suited for healthcare scenarios like personalized treatment planning, dynamic drug dosing, where decisions must adapt over time to a patient's changing condition [313]. However, simlarly to AI models, RL algorithms offer better insights in specific complex tasks but they often lack interpretability as well.

Bridging the gap between explainability and black-box RL is eXplainable RL (XRL), a specific subset of XAI focused on enhancing transparency and interpretability under RL domains [182]. It helps users understand why a RL agent takes specific actions in a given situation, providing insights into the agent's learned policies and making it easier to trust and validate the agent's behavior in real-world applications [181]. Indeed, XRL is especially useful in healthcare because it makes complex, adaptive decision-making systems more

transparent and trustworthy to clinicians. Studies like [308] highlight that, while RL agents learn to act, their actions may be difficult to interpret and XRL helps revealing the rationale behind treatment choices, making it easier for medical professionals to validate, trust, and act upon the recommendations.

For these reasons, this study is focused on applying a novel XRL pipeline on in-silico patients via the FDA-accepted UVA/Padova T1DM simulator [171]. While RL has shown promise in optimizing treatment strategies for chronic conditions like T1DM, its black-box nature poses significant challenges for clinical adoption. To address this, I integrate explainability directly into the learning process by extracting Shapley-based attributions from the agent's Q-values and further modeling them through a structural causal framework. This causal approach not only enhances interpretability but also allows us to identify and reason about the underlying decision structure and counterfactual effects, offering a novel pathway toward trustworthy and clinically aligned AI-driven therapy optimization.

4.7 Methodology

This section introduces the challenge of interpretable T1DM closed-loop control and frames it within the context of applying XRL to DRL methodologies. Building on this foundation, I deepen the problem formulation by leveraging Shapley value attributions to quantify the contribution of individual features to the learned policy, and propose a novel approach for constructing SCMs from these values, thereby enhancing the interpretability and causal understanding of the control strategy.

4.7.1 Problem Formulation

The problem of solving closed-loop control of T1DM via RL can be formulated as a discrete infinite MDP tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ is the set of all possible states, $\mathcal{P}$ is the probability function of transition between states, $\mathcal{A}$ is the set of all possible actions, $\mathcal{R}$ is defined as the reward function and $\gamma \in [0, 1]$ is commonly known as the discount factor (for the process of learning through exploration). Formulated this way, T1DM can be interpreted as the tuple where $s \in \mathcal{S}$ represents the value of the CGM measurements, $a_t \in \mathcal{A}$ is the (artificial pump) agent's action at a given timestep t. After the action has been chosen and performed (e.g. delivering of basal or bolus insulin), the state of the environment transitions to a new state $s' \in \mathcal{S}$, according to the state transition function $\mathcal{P}$ and a reward $r_t = R(s_t, a_t)$ is fed to the agent according to its performance. The agent is said to be following a policy π which aims at maximizing the expected future discounted reward (discounted using the discount factor γ).

From this basis an action-value function $Q^\pi(s, a)$ can be defined as:

$$Q^\pi(s, a) = \mathbb{E}[\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'} \, | s_t = s, a_t = a, \pi]. \tag{4.7}$$

According to the Bellman equations [16], the optimal value for this function is defined as $Q^*(s,a) = \max_\pi Q^\pi(s,a)$, can be achieved via solving:

$$Q^*(s,a) = \mathbb{E}_{s'}[R(s,a) + \gamma \max_{a'} Q^*(s',a')]. \tag{4.8}$$

As per the literature, directly iteratively solving $Q*$ is known to be computationally expensive and various methods of approximations are employed instead, which have been demonstrated to be a sound alternative.

In this work, the state space is composed of the CGM measurements of BG levels, which determine the environment updates according to the simulator's dynamics. Thus, each state $s_t \in S$ includes the last 30 BG observations (last 2.5 hours), as well as the last 5 insulin actions taken by the agent. The resolution used is of 5 minutes, i.e. $s_t = [G_t, I_t]$, where $I_t = [i_{t-4}, i_{t-3}, i_{t-2}, i_{t-1}, i_t]$ and $G_t = [g_{t-29}, g_{t-28}, ..., g_t]$. Each g_t is a number in the range $[0, 500]$, for sake of completeness of possibilities of each BG level measurements, and i_t is a positive real number. Therefore, g_t is measured in mg/dL, i_t is measured in mU/min and t is the time index for each step of 30 minutes time. Additionally, the time step window is limited to the last 72 steps, to include the last 72 x 30 = 2160 minutes, which in turn equals the last 1.5 days of patient's data.

Regarding the agent's actions, they are defined as $A = b_0, b/2, b, 2b, 3b$, where b_0 corresponds to giving no insulin at all and b is a reference value [265], considered to be in most cases the optimal basal insulin rate of 6.43 mU/min. Here, timesteps t are analogous to the ones considered in the state space, namely 5 minute intervals, to keep the formalization of problem formulation as simple as possible.

Every time a new state is obtained, the agent performs an action a_t and the next state s_{t+1} is given by the environment along with a reward $r_t \in \mathbb{R}$, which is fed to the controller. The reward function is the same defined in [265] as

$$R(g_t) = e^{-\frac{1}{2}(g_t - b_r)^2/900}, \tag{4.9}$$

where g_t is the BG level measurement as defined above and b_r is a target BG reference value set to 108 mg/dL. This reward is given to the agent as long as BG is kept in target range of $[70, 180]$. If the BG level is found outside such range, the agent is rewarded a penalty of $r_t = -1000$ in order to discourage risks of hyper-glycemia or hypo-glycemia. In this work, however, a DQN is employed and further considerations are described in the following subsection.

4.7.2 Solving T1DM via DRL

The framework of DRL was introduced in works such as [189], where a DQN was used as the main agent's policy approximator. DQNs use a deep neural network to approximate $Q(s,a)$ by delivering a function $Q(s,a;\theta)$, which relies on network's parameters θ. The learning then proceeds off-policy with a predefined target through exploration of an ϵ-greedy policy.

The Loss function is then defined as:

$$L_i(\theta_i) = E_{s,a,r,s'}[(y_i^{DQN} - Q(s,a;\theta_i))^2], \tag{4.10}$$

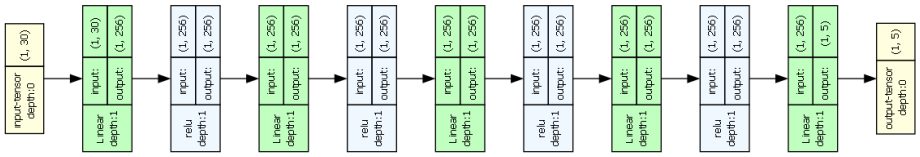


Figure 4.6: The structure of the adopted DQN. Without loss of generality, the complexity is kept as low as possible so as to adhere to the well-established standards of the DQN application. More complex DQNs or other DRL methods could be used as well. Such methods would maintain the interpretability value of this work as the main focus is on causal representations of the algorithms.

where y_i^{DQN} is the target Q-network parameterized by θ^-, expressed as:

$$y_i^{DQN} = [r + \gamma \max_{a'} Q(s', a'; \theta^-)]. \tag{4.11}$$

The parameters are upgraded via gradient descent by minimizing the square loss of Eq. 4.11. The Q-learning update is then defined by:

$$\nabla_{\theta_i} L_i(\theta_i) = E_{s,a,r,s'}[(r+\gamma \max_{a'} Q(s, a; \theta_{1-i}) - Q(s, a; \theta_i))\ \nabla_{\theta_i} Q(s, a; \theta_i)]. \tag{4.12}$$

In this work, closed-loop T1DM control is achieved through the use of a DQN. The overall architecture of the DQN used in this work is depicted in Fig 4.6.

4.7.3 Shapley Q-Values

One approach of explaining RL models is through post-hoc methods, i.e. explaining the model after its predictions. However, post-hoc methods could have disadvantages such as misrepresenting the original model by building other representations, for example by over-simplificating the model. This work focuses instead on maximizing the transparency of the interpretations by building upon the framework of SHAP (SHapley Additive exPlanations) [163].

Shapley values are a concept in cooperative game theory that fairly distributes the total payoff among players based on their individual contributions to various coalitions. The Shapley value of a player represents their average marginal contribution across all possible coalitions [302].

A valid way to build Shapley values of a multi-agent RL environment is the use of Shapley Q-Values, as proposed in [290]. The proposed approach focuses on building Shapley-Bellman optimality equations and operator through Markov convex games. While this is valuable in multi-agent settings, it is not trivial on how to construct such interpretations in single-agent scenarios, which is the scope of this work.

To address this issue, a practical way of obtaining Shapley values from a DRL model in the single agent case is to use the SHAP Deep Explainer, which is based on the Deep Learning Important FeaTures (DeepLIFT) algorithm [148,

235]. Indeed, in [163], the authors showed that the attribution rules chosen in DeepLIFT can well approximate Shapley values in various practical scenarios. That is, by integration of multiple samples, DeepLIFT estimates sum up to the difference between the model's expected output on past samples and current output (i.e. $f(x) - E[f(x)]$).

This method has proven successful in applications such as explaining DQN's experience replay mechanism [253]. In this work, however, in order to maximize transparency of the proposed solution, major interest is posed over DQN inner parameters. To accomplish this, Shapley values are constructed with the DQN layers. Indeed, Shapley values, in this sense, are a thorough way of establishing attribution of importance to the different features. Examples of Shapley values interpretations of the algorithm used in this work is depicted in Figs. 4.7, 4.9 and 4.10.

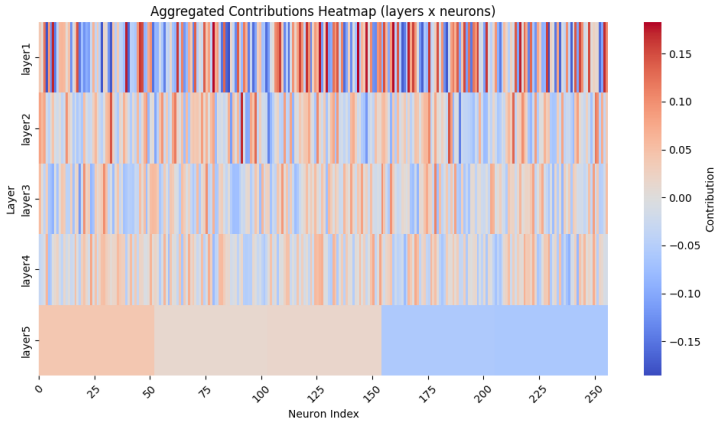
4.7.4 Assessing Causality

Although Shapley values offer great insight on agent's decision, an inherent problem with this technique is that it is hard to infer causality in more complex models, such as deep networks. Indeed, DQNs are often perceived as black boxes, due to their nature, and this is a known cause of distrust in the model caused by the lack of understanding of its inner mechanisms. In order to obtain more transparency of the model without affecting its complexity, a possible solution is to extract causality. In this regard, Shapley values alone are not enough to assess causal relationships behind agent's decisions.

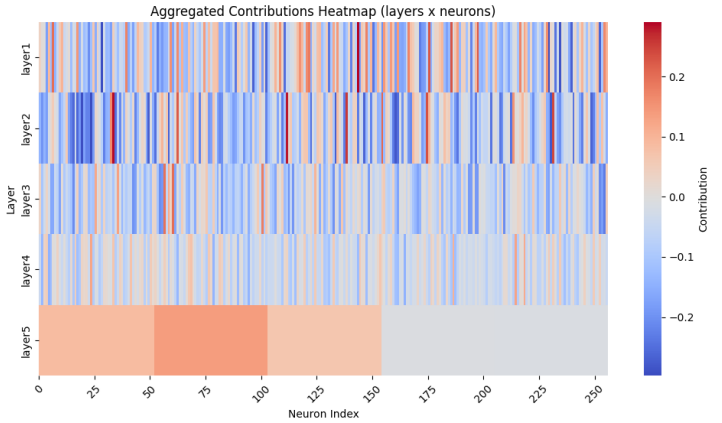
Current research behind causal models suggests multiple ways of constructing causality. A method to build causality is to use constraint-based methods, which are statistical techniques that rely on observed conditional independencies in the data to infer the structure of a causal graph. These methods operate under the assumption that the underlying data distribution follows a causal Markov condition and faithfulness assumption, meaning that the independencies in the data reflect the structure of the true causal graph.

Structural Causal Models (SCMs) are a formal framework for representing and analyzing causal relationships using structural equations and a corresponding graphical model, typically a Directed Acyclic Graph (DAG) [206]. Each variable in an SCM is expressed as a deterministic function of its direct causes (parents in the graph) and an independent noise term, allowing for clear modeling of causal mechanisms. SCMs enable counterfactual reasoning, causal inference, and the prediction of outcomes under interventions (e.g., do-calculus) [207].

SCMs are not the only causal method and one other approach is the Peter-Clark (PC) algorithm [247], a constraint-based method which starts with a fully connected graph and removes edges by testing for conditional independence between variables. This process continues iteratively, refining the causal graph by examining increasingly complex conditional independencies. From the PC interpretation of the data, a DAG [43] is built using a series of conditional independence tests to first determine the skeleton of the graph, and then ori-



(a)



(b)

Figure 4.7: Heatmap of the layers as per Shapley contributions. Fig. 4.7a represents Shapley values calculated in the frozen regime, while Fig.4.7b represents the heatmap obtained under normal learning circumstances.

enting edges based on identified v-structures and additional orientation rules that preserve acyclicity and consistency with the observed independencies. The PC algorithm is still relevant for causality as it is still used to date as a baseline reference for assessing causality of the models [109]. However, the use of SCMs in the context of T1DM is naturally more appealing as, in comparison with constraint-based methods, such as the PC algorithm, they define how variables causally affect each other, thus introducing better liabilities in the decision making.

A different approach is Granger causality, which is a statistical method to determine whether one time series can predict another [85]. It is used in applications such as monitoring electrical activity of individual cells over time, as in [14], or to capture spatial relationships in multielectrode EEG data, as in [264]. It is a useful tool since it enables causal connections between variable series to be identified by assessing whether past values of one variable help predict future values of another, beyond what the latter's own past values can predict. However, using Granger causality and other time-series methods only assess predictive relationships and do not account for causal effects. Instead, SCMs use what's called "do-calculus" to model interventions, enabling true causal inference.

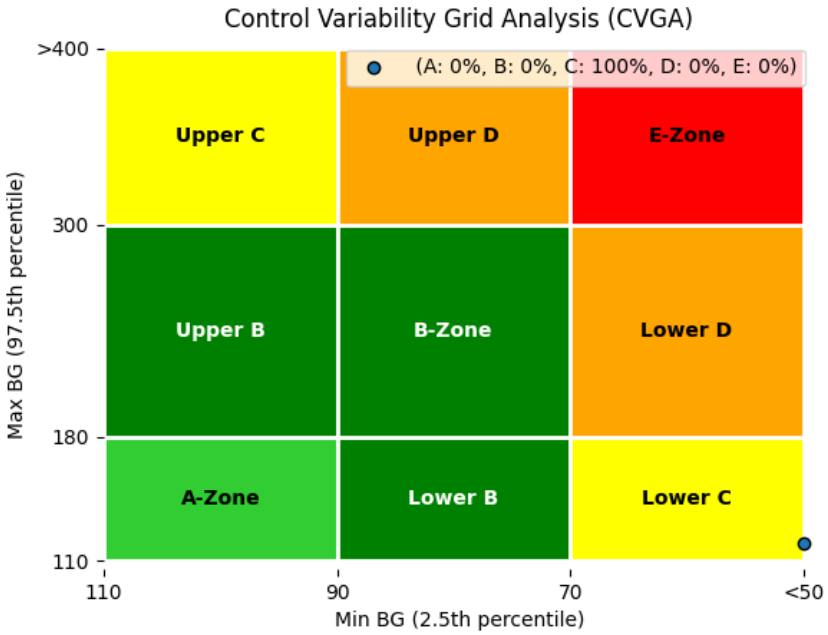
Other applications explore the possibility of using probabilistic graphical models such as Bayesian networks to express causality [191]. In this other way, the causal interventions are represented by using do-calculus, which allows the modeling of changes in the network structure by explicitly setting variables to specific values, independent of their usual causes. Still, the PC Algorithm, Bayesian networks, and constraint-based methods rely on observational data, making them vulnerable to confounders (hidden common causes). Instead, SCMs explicitly model confounders by introducing latent variables, making them more robust in real-world applications.

As a consequence, in this work SCMs are used in place of other methods. Indeed, the main focus of this contribution is to infer causal relationships in the model's mechanisms. When accounting for probabilistic inference and discovering relationships in an unknown system, methods such as Bayesian Networks are to be preferred. However, by deriving Shapley values the causal origin of the data is already known. Thus, in order to analyze interventions and counterfactuals, SCMs are chosen instead.

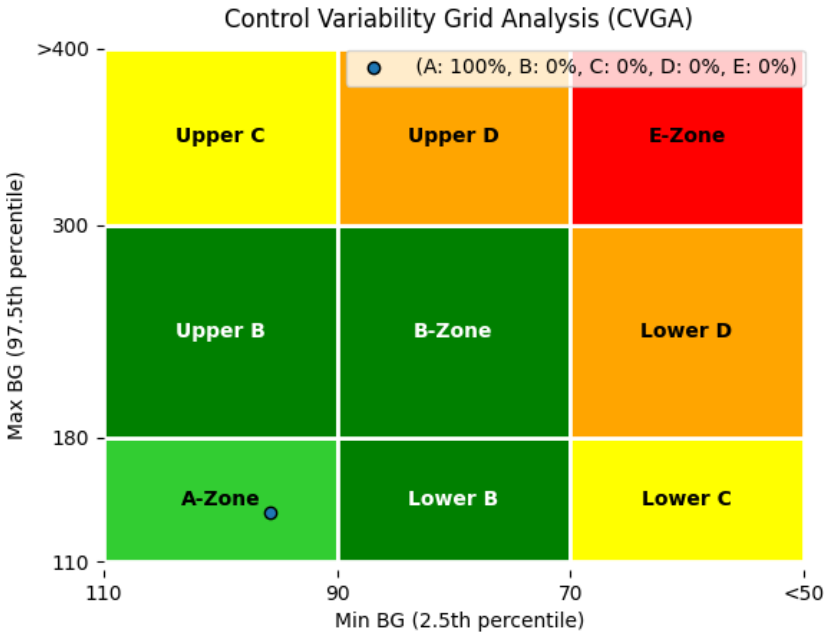
SCMs [210] can be formally defined as a collection $\mathcal{C} := (\mathbf{S}, P_N)$ of d structural assignments $X_j := f_j(\mathbf{PA}_j, N_j), j = 1, ..., d$, where $\mathbf{PA}_j \subseteq \{X_1, ..., X_d\} \setminus \{X_j\}$ are called parents of X_j and $P_N = P_{N_1,...,N_d}$ is a jointly independent product distribution over the noise variables. In the framework of SCMs, given multiple functions that describe a non-zero dependency, structural minimality is commonly preferred.

SCMs are used in forms of XAI such as contrastive explanations via counterfactual explananda [185]. Indeed, one of the key strengths of SCMs lies in their ability to facilitate causal identification, which is the process of determining whether a causal relationship can be inferred from observed data.

To enhance interpretability of the RL agent in T1DM management, I devel-



(a)



(b)

Figure 4.8: An example of Control Variability Grid Analysis of the network in the two regimes of training. In Fig.4.8a, the network is not able to correctly learn a proper way to maximize the time in-range of BG levels, while Fig.4.8b is the expected graph in a controlled environment.

oped a three-stage explainability pipeline: extracting DQN weights → building Shapley Q-values from weights → building the SCM graph. This pipeline allows us to move beyond attribution and toward causal understanding of the agent's decision-making process.

4.7.5 Training Procedure

The DQN model was trained using a virtual population of 30 subjects generated by the UVA/Padova T1DM simulator, stratified equally across three age groups: children (n=10), adolescents (n=10), and adults (n=10). The actions are defined by the set $A = b_0, b/2, b, 2b, 3b$, where b_0 corresponds to giving no insulin at all and b is 6.43 mU/min. The DQN is structured as a fully connected neural network, which is designed to approximate Q-values in the proposed T1DM setting, as depicted in Fig. 4.6. It takes a state as input and outputs a Q-value for each possible action. The network has five linear layers, with the first four hidden layers consisting of 256 units each. The first three hidden layers are followed by batch normalization and dropout (p=0.5) to improve training stability and reduce overfitting. The final layer outputs raw Q-values for each action, defined in section 4.7.2. While raw network weights encode this learned policy, they offer limited transparency. To address this, I compute Shapley Q-values, that assign importance scores to input features by considering their marginal contributions to predicted Q-values.

To investigate the internal representation dynamics relevant to interpretability and causal attribution, training was conducted under two regimes: a standard learning regime, in which all layers were trainable, and a frozen-layer regime, where the third hidden layer was held fixed during training by disabling gradient updates (require_grad = False). Both networks were initialized with the same random seed and trained with identical hyperparameters to ensure consistent experimental conditions.

The optimizer used was Adam with a learning rate of 1×10^{-4}, and training was performed using mini-batches of size 64 sampled from a shared experience replay buffer containing up to 100,000 transitions. The agent followed an ϵ-greedy exploration policy, where ϵ was linearly annealed from 0.5 to 0.05 over the first 10,000 episodes, and held constant thereafter. Training proceeded for a total of 500,000 environment steps. A separate target network was used to stabilize Q-value updates, and was synchronized with the main network every 1,000 steps. Both agents received identical sequences of training data by drawing from the same experience buffer and applying identical update steps.

Post-training, each model's learned Q-values were decomposed using Shapley-based attribution computed directly from the trained weights and a sampled state, and these were used to construct DAGs that reflect the agent's inferred causal structure between state features and insulin action values. The frozen-layer regime served as a diagnostic condition for impaired abstraction and representation learning, thereby highlighting the role of depth-specific adaptability in generating meaningful causal explanations.

To highlight Shapley contribution, I employ two NetworkX [93] visualizations of heatmaps and bar plots. The heatmap visualizes the *weighted contributions* of the input features to the output neurons of the network. Each cell in the heatmap corresponds to the combined SHAP value for a specific feature and output neuron (x-axis), given the input features (y-axis), weighted by the learned connection weight in the neural network. The color intensity of each cell indicates the magnitude of the contribution. On the other hand, the bar plot isolates and visualizes *the total contribution of input features* (x-axis) to a single output neuron. The bars (y-axis) represent the aggregated impact of each input feature on a specific output feature after combining SHAP values with model weights.

In Figs. 4.7, 4.9 and 4.10, I visualize these Shapley values across representative training episodes. These heatmaps and bar plots offer a view into which features are driving the agent's decisions, highlighting the relative influence of physiological signals such as glucose trends, carbohydrate intake, and prior insulin dosages, at each layer.

However, as discussed before, Shapley values are inherently associative and not causal. They describe which inputs were influential, but not why the learning progressed in a particular direction. This limits their explanatory depth, especially when trying to interpret model dynamics across training regimes.

While Shapley Q-values offer insight into the relative importance of input features for the Q-function, they remain fundamentally associative. That is, they reflect correlations between inputs and Q-values but cannot determine whether a change in a feature or parameter causes a change in the agent's decision-making. To bridge this gap, I construct SCM graphs that allow for reasoning about directional dependencies and interventional dynamics.

Indeed, as explained beforehand, to facilitate a controlled causal analysis, I purposefully introduce a second training regime in which learning is artificially constrained: the frozen-learning condition. The visual explanations generated by these SCM graphs (see Figs. 4.11 and 4.12) are obtained from observational data extracted separately between the two trainings. Thus, two main experimental conditions are considered:

1. **Standard Training** (Fig. 4.12): The SCM reveals a well-connected causal structure, indicating balanced contributions across multiple weights and a coherent flow of influence from relevant features to Q-value formation.

2. **Frozen-Learning Regime** (Fig. 4.11): Here, gradient updates are blocked after an early phase. While the Shapley heatmaps remain visually similar (suggesting continued attribution to input features), the SCM graph reveals disrupted or diminished causal pathways, especially from adaptive features like glucose variability.

In particular, Shapley values alone cannot distinguish between active learning and stagnation. Since they are based on feature marginal contributions within a fixed model, they continue to produce apparently reasonable attribution maps, even when the model is no longer adapting (Fig.4.11). By freezing

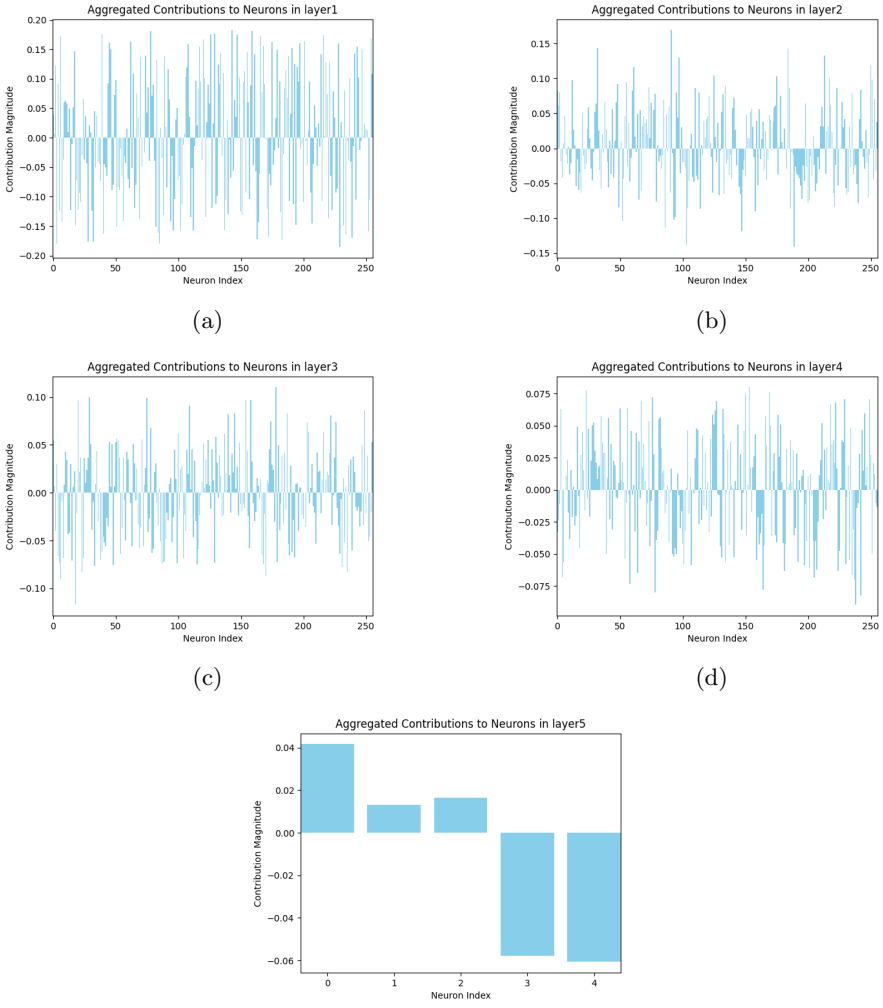


Figure 4.9: Bar plots of the hidden layers of the DQN under frozen regime: first hidden layer 4.9a, second hidden layer 4.9b, third hidden layer 4.9c, fourth hidden layer 4.9d and the output layer 4.9e. These plots are constructed after extracting Shapley values attributions across all layers. No particular evidence of issue in the learning can be visually seen in these plots.

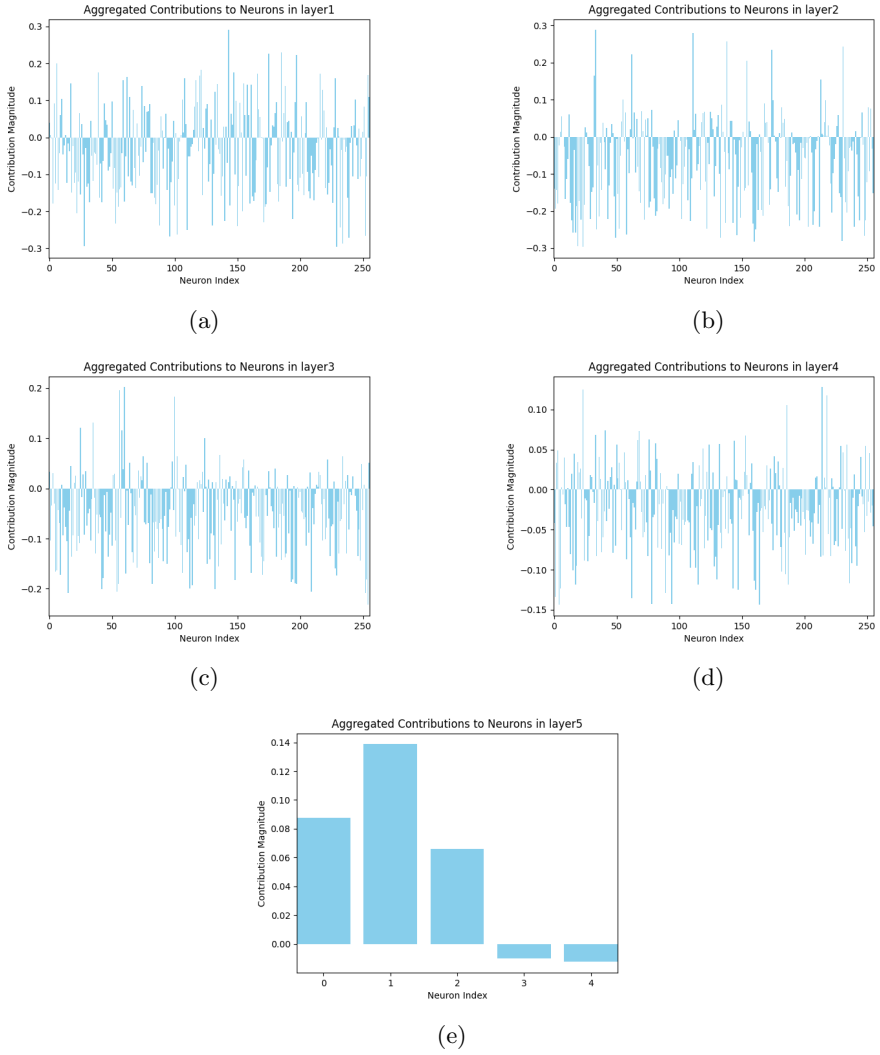


Figure 4.10: Bar plots of the hidden layers of the DQN under the normal training regime: first hidden layer 4.10a, second hidden layer 4.10b, third hidden layer 4.10c, fourth hidden layer 4.10d and the output layer 4.10e. In comparison with the bar plots of the frozen learning regime, no particular evidence of difference in learning can be visually extracted at first sight.

the learning, I create a scenario where the observed Shapley values may still point to relevant inputs, yet the agent's behavior deteriorates, as verified by the cumulated discounted reward obtained.

The SCMs generated under the frozen-learning condition (Fig. 4.11) reflect this explicitly. Compared to the balanced, interconnected graph of the standard training condition (Fig.4.12), the frozen-learning SCM shows disrupted causal pathways, such as weakened or broken links from key physiological features to Q-value formation (for example, layer 3). This structural breakdown aligns with observed performance deficits, but it is not detectable through Shapley analysis alone. By purposefully freezing learning, I highlight the central contribution of this work: only through SCMs can I access the internal causal logic of RL agents, beyond what attribution maps suggest.

Furthermore, given the structure of the DQN, to enhance the interpretability SCM graphs, I aggregated neurons into fixed groups of 32 based on their index in the network architecture. This aggregation is a purely procedural decision, not based on any clustering or functional similarity, and serves to reduce the dimensionality of the analysis (mainly graph dimensions and number of links) for a clearer presentation, while causal assessment is still preserved. Indeed, examining each individual neuron would result in high visual complexity and obscure the broader patterns I aim to surface. By grouping adjacent neurons (e.g., Neurons 0–31, 32–63, etc.), I retain enough granularity to observe localized patterns of influence, while enabling compact causal graphs and Shapley summaries that are visually tractable and cognitively accessible. This grouping strategy is consistently applied across all attribution maps and causal graphs (e.g., Figs. 4.11, 4.12), where each node or feature corresponds to a block of 32 neurons. Furthermore, the influence of the neurons was also included in the visual representation as well as it could better highlight causality layer-wise.

For better clarity, the DAG network visualizes the relationship between input features by representing: nodes as entities, by using input features; output neurons, represented as another set of nodes; edges, representing connections between input features and output neurons - with edge weights indicating the strength and polarity of the contribution. Derived from the normalized edge weights (SHAP value $\times$ weight), the edge links are mapped to a colormap, where bright colors (e.g., yellow) indicate strong positive contributions, while dark colors (e.g., purple or blue) indicate either no contribution at all or a strong negative contribution. In summary, the graph network shows the pathways through which input features influence output neurons and highlights the most influential input-output connections based on weights. In addition, it visualizes how certain features might dominate influence over specific neurons. Such methodology is supposed to introduce better interpretability and clarity over model reasoning and causality.

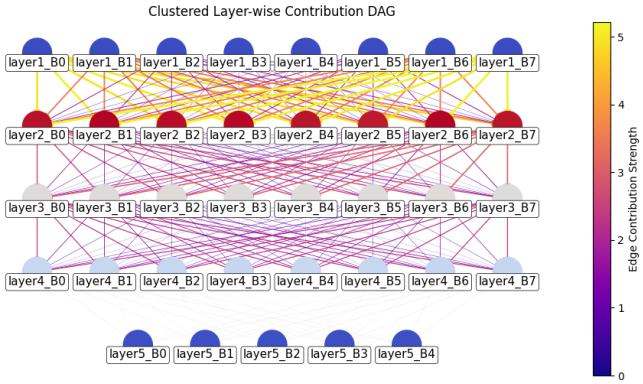


Figure 4.11: The DAG constructed from the SCM derived from the DQN agent. This DAG refers to the frozen learning regime and it can be visually seen that layer 3 is kept fixed during learning (no gradient updates). This Fig. showcases the potential to spot eventual issues that may arise that are not discoverable by Shapley Q-Values alone.

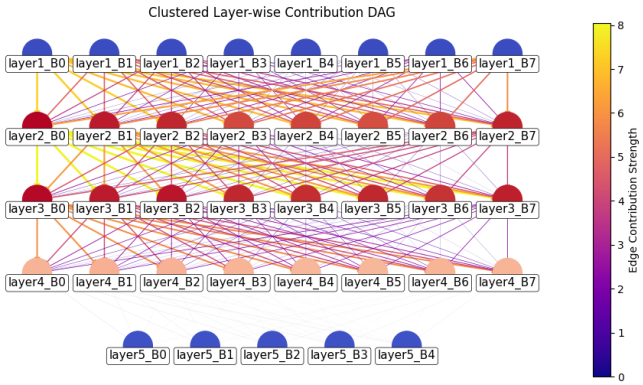


Figure 4.12: The DAG constructed from the SCM in the normal training regime. From this Fig. it can be seen that the contribution values across the graph is equally distributed (as expected), thus visually confirming that the learning is not hindered by network internal mechanisms.

Chapter 5

Results

5.1 Results of Transparency Explanations

The AC algorithm was tested on the in-silico 10 adults, 10 children and 10 adolescents provided in the educational version of the UVA / Padova simulation environment. The simulation was run for a 30-day period and the corresponding Control Variability Grid Analysis (CVGA), zone stats and risk stats are analogous to the ones obtained in [49]. The two profiles described in the original work *P1* and *P2* were both employed in the study.

To evaluate the proposed explanations during the learning process, I presented two questionnaires to 19 T1D patients and 5 physicians. Each questionnaire used a Likert scale of 1 to 5 to assess two factors: **quality** and **trust**. For each explanation methodology, *P1* and *P2* were presented jointly to obtain a general sentiment of the presented explanations. The patients were aged between 19 to 35 ($\mu = 24.4$) comprising of 10 males and 9 females, while the physicians were aged between 30 and 59 ($\mu = 48.2$) comprising of 3 females and 2 males. All the physicians had the necessary knowledge to understand T1D mechanics but not all of them were previously exposed to RL or to T1D through RL agents. All the participants were used in the evaluation, regardless of the background preparation in understanding such methods. This is made on purpose as the final aim of XRL is to provide the same types of explanations to different expertise level, ideally matching all of them. It was chosen to group MSX and RDX graphs together for a matter of simplicity of proposed explanations. Indeed, because one is the derivation of the other and both are represented by bar charts, it makes no significant difference to split them into two different explanations methodologies.

The MSX and RDX graphs obtained in the environment by the algorithm proposed are depicted in Fig. 4.2. Samples of the explanations generated by BMs are depicted in Fig. 5.1. Here, as cited before, the choice of having 6 components was set empirically after evaluating different values. Indeed, much lower or much higher values would make the final graph almost identical in terms

Metric (Means)	Pair	Mean-diff	Median-diff	p-value
Understandable	DQv-RDX	0.63	0.50	0.079
(μ_{DQv}=1.71,	DQv-BM	2.21	2.50	**0.001**
μ_{RDX}=2.33,	DQv-N	0.38	-0.50	0.469
μ_{BM}=3.92,	RDX-BM	1.58	2.00	**0.002**
μ_{N}=1.33)	RDX-N	1.00	1.00	**0.001**
	BM-N	2.58	3.00	**0.000**
Complete	DQv-RDX	0.08	0.00	0.992
(μ_{DQv}=2.21,	DQv-BM	0.88	-1.00	**0.021**
μ_{RDX}=2.13,	DQv-N	0.92	1.00	**0.014**
μ_{BM}=3.08,	RDX-BM	0.96	-1.00	**0.009**
μ_{N}=1.29)	RDX-N	0.83	1.00	**0.030**
	BM-N	1.79	2.00	**0.001**
Detailed	DQv-RDX	0.38	1.00	0.604
(μ_{DQv}=2.88,	DQv-BM	0.29	0.00	0.770
μ_{RDX}=2.50,	DQv-N	1.54	2.00	**0.001**
μ_{BM}=3.17,	RDX-BM	0.67	-1.00	0.130
μ_{N}=1.33)	RDX-N	1.17	1.00	**0.001**
	BM-N	1.83	2.00	**0.006**
Satisfying	DQv-RDX	0.79	0.00	**0.046**
(μ_{DQv}=1.92,	DQv-BM	1.08	-1.00	**0.002**
μ_{RDX}=2.71,	DQv-N	0.46	1.00	0.422
μ_{BM}=3.00,	RDX-BM	0.29	-1.00	0.763
μ_{N}=1.46)	RDX-N	1.25	1.00	**0.003**
	BM-N	1.54	2.00	**0.002**

Table 5.1: Results of the one way ANOVA test in terms of pair, mean difference, median difference (positive higher values are better) and p-values. Under the metrics, in parenthesis, are the means of the votes in the Likert scale for each methodology. The results refer to the votes of the questionnaire about quality. Perceived quality of the explanations provided in this work in terms of quality of explanations (Likert scale 1-5). Bold values are where the p-value is < 0.05.

of contributions to the Q-Value function and so a value of 6 components was kept during the experiments. The results of the comparisons are represented in Tab. 5.1 and Tab. 5.2.

Evidences in mean differences highlight that, generally speaking for all the quality metrics, the p-values over the pairwise comparisons for BM and RDX are < 0.05 and thus it's safe to reject the null hypothesis. Indeed, it is generally perceived by the users as beneficial to have a graphical representation of the dynamics as with BM or MSX/RDX: those two techniques, compared with the others usually got a better average vote on the questionnaires. In addition, N has the lowest score in all metrics suggesting that any kind of explanation considered in the study is still better than no explanation given. In terms of quality, it is evident that BM is perceived as the best on all metrics, followed by RDX, DQv and N, respectively. This demonstrates that the hypothesis of better explainability via RDX or BM is matched with actual preference data. The results of the pairwise comparison of the Likert scale for the quality of explanations are represented in Tab. 5.1.

As opposed to the expectations, however, when addressing trust and safety, apart from few examples, no method provided evidence of increased trust to-

Metric (Means)	Pair	Mean-diff	Median-diff	p-value
Confidence	DQv-RDX	1.08	-1.00	**0.006**
(μ_{DQv}=1.54,	DQv-BM	1.38	-1.00	**0.001**
μ_{RDX}=2.63,	DQv-N	0.42	-1.00	0.568
μ_{BM}=2.92,	RDX-BM	0.29	0.00	0.801
μ_{N}=1.96)	RDX-N	0.67	0.00	0.170
	BM-N	0.96	0.00	**0.019**
Predictability	DQv-RDX	0.46	1.00	0.596
(μ_{DQv}=3.00,	DQv-BM	0.13	1.00	0.986
μ_{RDX}=2.54,	DQv-N	0.38	1.00	0.736
μ_{BM}=2.88,	RDX-BM	0.33	0.00	0.799
μ_{N}=2.63)	RDX-N	0.08	0.00	0.995
	BM-N	0.25	0.00	0.903
Reliability	DQv-RDX	0.38	0.00	0.599
(μ_{DQv}=2.21,	DQv-BM	1.04	-1.00	**0.004**
μ_{RDX}=2.58,	DQv-N	0.58	0.50	0.219
μ_{BM}=3.25,	RDX-BM	0.67	-1.00	0.126
μ_{N}=1.50)	RDX-N	0.96	0.50	**0.010**
	BM-N	1.63	1.50	**0.007**
Safety	DQv-RDX	0.13	0.00	0.962
(μ_{DQv}=2.13,	DQv-BM	0.04	0.00	0.998
μ_{RDX}=2.00,	DQv-N	0.71	1.00	**0.036**
μ_{BM}=2.17,	RDX-BM	0.17	0.00	0.917
μ_{N}=1.42)	RDX-N	0.58	1.00	0.116
	BM-N	0.75	1.00	**0.023**

Table 5.2: Results of the one way ANOVA test in terms of pair, mean difference, median difference (positive higher values are better) and p-values. Under the metrics, in parenthesis, are the means of the votes in the Likert scale for each methodology. Perceived quality of the explanations provided in this work in trust towards the explanations (Likert scale 1-5). The results refer to the trust factor. Bold values are where the p-value is < 0.05.

wards the system. This could be due either by patients' or physicians' prior assumptions of a given device for T1D control or simply no gain in trust overall. Other methods could be tested to ascertain if trust is a major issue in this problem or it is dependent on the methods used. The results of the pairwise comparison of the Likert scale for the quality of explanations are represented in Tab. 5.2.

5.2 Results of Post-Hoc Explanations

In this Subsect. the results of the Post-Hoc explanations are given both in terms of explaining to laypeople and explaining the algorithms through causality.

5.2.1 Results of Markov Networks

To evaluate the impact of system intepretability on patient trust and system adoption, a structured explanation protocol was administered to T1DM patients following an initial survey. Prior to the intervention, participants (n = 41) com-

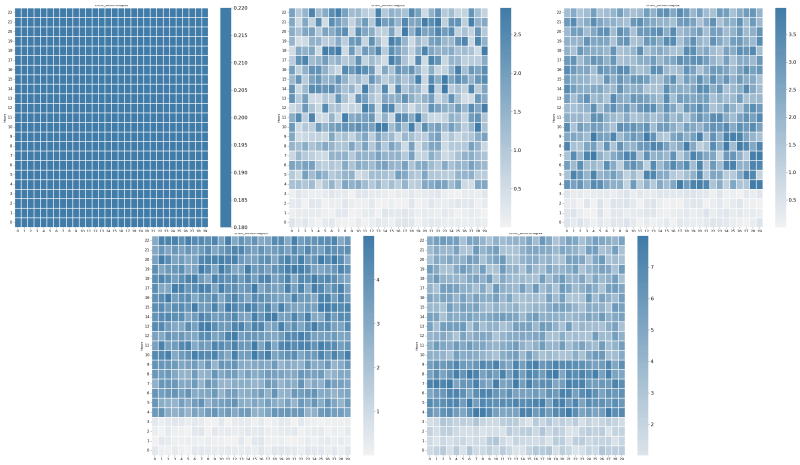


Figure 5.1: Examples of explanations generated by BMs that change over time, following profile P2 from critic's perspective. Darker areas represent higher expected future reward. The X axis represent a projection of future 30-day periods, while the Y axis represents the hours of the day. From top-left to bottom-right is the evolution over time of agent's beliefs. Top-left figure represents the initial state of the learning, that is the agent maps all hours of the day to the same value. Middle-top figure starts differentiating between hours of the day. The agent then learns the P2 profile and it becomes visually evident through figures from top-right, bottom-left and, finally, bottom-right, where the agent has correctly mapped the optimal control.

pleted a baseline assessment that captured their current treatment methods, prior experience with automated insulin delivery, and attitudes toward closed-loop systems. Trust was measured using a 5-point Likert scale, and patients indicated whether they would consider using a closed-loop fully-automated controller.

Patients then participated in a 15–20 minute explanation session. This included a visual walkthrough of the reinforcement learning agent's decision process, modeled as a MDP. State variables such as BG level, carbohydrates (CHO) intake, insulin on board (IoB), and physical activity were illustrated, along with the corresponding actions and reward functions.

The session also included an interactive simulation demonstrating how the agent responded to common scenarios (e.g., skipped meals, exercise). Analogies, such as comparing the system to automotive cruise control, were used to frame the system as assistive rather than fully autonomous. The session then concluded with a short Q&A.

After the explanation, the same trust and adoption questions were administered again after showing the three Markov networks to assess changes in perceptions of explanation levels. This approach aimed to bridge the gap between technical complexity and patient understanding, with the goal of increasing

trust in automated diabetes management systems. On the real patients pool, comprising of 22 males and 19 females aged between 18-50 (s.d. ±8.5), only about 68% of the patients were involved in a closed-loop intervention. The other patients actively neglected the use of an artificial pump due to lack of trust or due to personal preference. Indeed, only around 15% of the patients involved were willing to adopt a fully automated artificial pancreas closed-loop mechanism before the explanations.

The questions administered to the patients were:

- **Q1** (Likert): On a scale of 1–5, how much do you trust an automated insulin delivery system (closed-loop)?
- **Q2** (binary): Would you consider using a closed-loop controller in your diabetes management?

After modeling the Markov network and showing the patients the relationships between agent's decisions and its driving mechanics, a 28% increase in trust towards the system and were able to better understand the behavior of the controller agent. This resulted in 43% of the patients with an increased trust towards the system that were willing to adopt the automated agent afterwards. The results are represented in Tab. 5.3 and are depicted in Fig 5.2.

Metric	Before	After
Willing to adopt closed-loop	15% (6 patients)	43% (17 patients)
High trust (Likert 4–5)	18% (7 patients)	46% (18 patients)
Average trust score (Likert 1–5)	2.51	3.27

Table 5.3: Trust and Willingness to Adopt Before and After Explanation

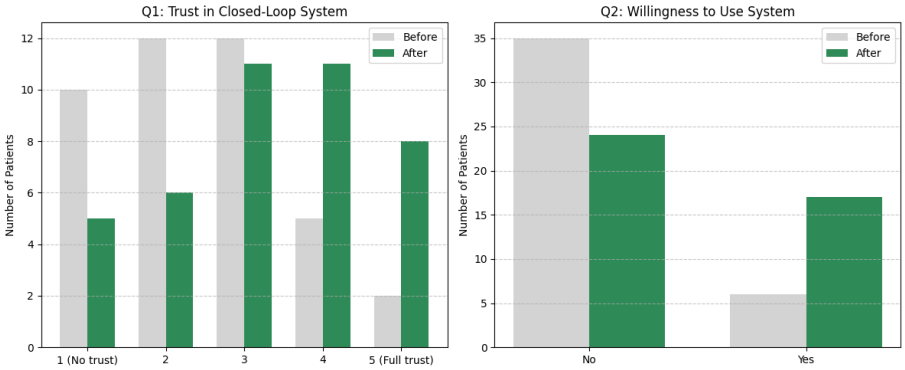


Figure 5.2: Results of the questionnaires before and after the Markov network graph provided to patients.

5.2.2 Results of Structural Causal Models

Training under both regimes proceeded smoothly, with stable convergence of the Q-value loss observed after approximately 400,000 environment steps. The shared experience replay and identical hyperparameter settings ensured comparable learning conditions, allowing for a controlled comparison of representational dynamics. In the standard learning regime, all network layers adapted through gradient updates, resulting in robust policies capable of achieving effective glycemic control across the diverse patient cohort (see Fig. 4.8b). Conversely, in the frozen-layer regime, the fixed third hidden layer constrained the network's capacity for deep feature abstraction, yielding noticeably slower improvement in action-value estimates and a modestly degraded policy performance (see Fig. 4.8a).

From an interpretability standpoint, the post-training analysis revealed critical differences between the two regimes. When examining Shapley Q-Values derived from the learned weights, both models exhibited broadly similar attributions across input features, indicating that the Shapley decomposition alone was insufficient to clearly distinguish the effect of freezing intermediate representations (see Figs. 4.9 and 4.10). While Shapley values provided quantitative attributions, they lacked the granularity needed to expose structural representational deficiencies caused by the frozen layer.

In contrast, the SCMs constructed as DAGs from the same attribution data offered a more nuanced and visually salient explanation. Under the frozen-layer regime, the DAGs exhibited a pronounced disruption in causal pathways associated with the features mediated by the frozen third layer (see Fig. 4.12). Specifically, the expected edges representing functional dependencies between intermediate feature representations and downstream Q-values were either absent or markedly weakened, visually illustrating the blockage in information flow caused by the frozen parameters. This disruption was not present in DAGs from the standard learning regime (see Fig. 4.11), where fully trainable layers enabled the emergence of coherent and dense causal structures linking inputs to actions.

These results highlight that while Shapley value attributions remain a powerful tool for feature importance ranking, their interpretability is limited when internal representational learning is impaired. The SCM-based causal graphs, by explicitly modeling the network's hierarchical dependencies, provided a complementary and more sensitive diagnostic framework capable of revealing latent issues in the network's learning and causal inference capabilities. This contrast underscores the necessity of combining value-based attributions with structural causal analysis to gain deeper insights into the representational integrity and decision-making processes of deep reinforcement learning agents, especially in high-stakes clinical applications such as T1DM management.

Chapter 6

Conclusions

This thesis explored the integration of explainable artificial intelligence (XAI) and reinforcement learning (RL), with a specific focus on their application to the management of Type 1 Diabetes Mellitus (T1DM). As autonomous agents increasingly support or replace manual processes in healthcare, ensuring transparency and trust in their decision-making becomes critical. To address this, the research investigated a range of techniques under the umbrella of eXplainable Reinforcement Learning (XRL), aiming to clarify the behavior of RL-based agents in high-stakes clinical environments.

The work spans from foundational analysis to practical applications. It begins with a comprehensive literature review of XRL, offering a structured understanding of current trends, methodological gaps, and opportunities in the field. This is followed by the development and evaluation of user-centered and model-agnostic explanation techniques, as well as an investigation into novel, theoretically grounded approaches for interpreting agent behavior.

The contributions of this thesis can be summarized across three main dimensions: foundational, methodological, and applied. On the foundational side, the thesis provides a detailed conceptual and methodological review of the XRL literature. By classifying existing approaches and identifying limitations, it offers a structured reference point for future research. This work highlights the fragmented nature of current XRL efforts and emphasizes the need for standardization and interdisciplinary perspectives, particularly when applied to sensitive domains like healthcare.

Methodologically, several explanation techniques were proposed and evaluated. These include reward-difference analysis and contrastive explanations, designed to offer user-centered insights into agent decisions. Additionally, a novel post-hoc abstraction technique based on Markov models was developed to provide model-agnostic interpretations of agent behavior across trajectories. These methods collectively aim to improve interpretability for both technical and non-technical audiences.

From an applied perspective, the thesis introduces an innovative approach combining Shapley-based Q-value attributions from Deep Q-Networks (DQNs)

with Structural Causal Models (SCMs). This method provides a causal perspective on RL behavior, highlighting the limitations of attribution-based techniques in capturing true causal dependencies. The approach enables a more principled and explainable analysis of agent decision-making processes in the context of T1DM.

This research contributes to the growing body of work aimed at making AI systems more transparent, accountable, and suitable for deployment in high-stakes environments. By focusing on healthcare and specifically T1DM, it grounds theoretical advancements in practical, real-world challenges. The methods developed herein address the interpretability needs of diverse user groups, ranging from clinicians and researchers to patients and caregivers. The thesis bridges the gap between technical innovation and ethical deployment, highlighting how explainability can enhance trust and facilitate regulatory compliance. In doing so, it supports a vision of AI that complements human expertise, rather than replacing it blindly.

While the contributions can be considered substantial, certain limitations remain. Some of the proposed explanation techniques may not generalize easily to continuous control settings or multi-agent environments. Additionally, the clinical realism of the simulated environments used in this work may not fully capture the complexity of real-world T1DM management. Another limitation lies in the subjective nature of explanation evaluation. While the proposed methods aim to serve both expert and lay audiences, initial human-centered validation, through user studies involving clinicians, was conducted, but further large-scale evaluations are still warranted.

Future research could extend the current methods to other chronic conditions and more complex treatment regimens. For instance, the framework introduced here could be adapted to settings involving comorbidities or multi-agent decision systems. Moreover, additional work is needed to scale explanation techniques to large-scale and real-time RL applications. Enhancing human-in-the-loop feedback mechanisms, improving the interpretability of policy-gradient methods, and integrating causal modeling with other deep RL architectures are promising directions. Finally, broader empirical validation, including qualitative studies with domain experts, will be crucial to ensure the practical utility and adoption of these techniques.

Bibliography

[1] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018.

[2] Riad Akrour, Davide Tateo, and Jan Peters. Towards reinforcement learning of human readable policies. In *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases: The 1st Workshop on Deep Continuous-Discrete Machine Learning*, 2019.

[3] Fadi Al-Turjman. *Artificial intelligence in IoT*. Springer, 2019.

[4] Alnour Alharin, Thanh-Nam Doan, and Mina Sartipi. Reinforcement learning interpretation methods: A survey. *IEEE Access*, 8:171058–171077, 2020.

[5] Dan Amir and Ofra Amir. Highlights: Summarizing agent behavior to people. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1168–1176, 2018.

[6] Ofra Amir, Finale Doshi-Velez, and David Sarne. Summarizing agent strategies. *Autonomous Agents and Multi-Agent Systems*, 33:628–644, 2019.

[7] Andrew Anderson, Jonathan Dodge, Amrita Sadarangani, Zoe Juozapaitis, Evan Newman, Jed Irvine, Souti Chattopadhyay, Alan Fern, and Margaret Burnett. Explaining reinforcement learning to mere mortals: An empirical study. *arXiv preprint arXiv:1903.09708*, 2019.

[8] Sule Anjomshoae, Amro Najjar, Davide Calvaresi, and Kary Främling. Explainable agents and robots: Results from a systematic literature review. In *18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019*, pages 1078–1088. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[9] Raghuram Mandyam Annasamy and Katia Sycara. Towards better interpretability in deep q-networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4561–4569, 2019.

[10] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020.

[11] Mark A Atkinson, George S Eisenbarth, and Aaron W Michels. Type 1 diabetes. *The lancet*, 383(9911):69–82, 2014.

[12] Emna Baccour, Aiman Erbad, Amr Mohamed, Mounir Hamdi, and Mohsen Guizani. Reinforcement learning-based dynamic pruning for distributed inference via explainable ai in healthcare iot systems. *Future Generation Computer Systems*, 155:1–17, 2024.

[13] K Banumathi, Latha Venkatesan, Lizy Sonia Benjamin, K Vijayalakshmi, Nesa Sathya Satchi, and Nesa Sathya Satchi IV. Reinforcement learning in personalized medicine: A comprehensive review of treatment optimization strategies. *Cureus*, 17(4), 2025.

[14] Lionel Barnett and Anil K Seth. The mvgc multivariate granger causality toolbox: a new approach to granger-causal inference. *Journal of neuroscience methods*, 223:50–68, 2014.

[15] Marco Baroni. Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society B*, 375(1791):20190307, 2020.

[16] Andrew G Barto. Reinforcement learning: An introduction. by richard's sutton. *SIAM Rev*, 6(2):423, 2021.

[17] Andrew G Barto and Sridhar Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete event dynamic systems*, 13(1-2):41–77, 2003.

[18] George Baryannis, Samir Dani, and Grigoris Antoniou. Predicting supply chain risks using machine learning: The trade-off between performance and interpretability. *Future Generation Computer Systems*, 101:993–1004, 2019.

[19] Osbert Bastani, Yewen Pu, and Armando Solar-Lezama. Verifiable reinforcement learning via policy extraction. *Advances in neural information processing systems*, 31, 2018.

[20] Daniel Beechey, Thomas MS Smith, and Özgür Şimşek. Explaining reinforcement learning with shapley values. In *International Conference on Machine Learning*, pages 2003–2014. PMLR, 2023.

[21] Vaishak Belle and Ioannis Papantonis. Principles and practice of explainable machine learning. *Frontiers in big Data*, page 39, 2021.

[22] Alice Bernasconi, Antonio Balordi, Alessio Zanga, Rafael Cabañas, et al. On counterfactual explanations of cardiovascular risk in adolescent and young adult breast cancer survivors. In *CEUR WORKSHOP PROCEEDINGS*, volume 3880, pages 211–223, 2024.

[23] Elisa Bertino, Murat Kantarcioglu, Cuneyt Gurcan Akcora, Sagar Samtani, Sudip Mittal, and Maanak Gupta. Ai for security and security for ai. In *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*, pages 333–334, 2021.

[24] Dimitris Bertsimas, Arthur Delarue, Patrick Jaillet, and Sebastien Martin. The price of interpretability, 2019.

[25] Kayla Boggess, Sarit Kraus, and Lu Feng. Toward policy explanations for multi-agent reinforcement learning. *arXiv preprint arXiv:2204.12568*, 2022.

[26] Leo Breiman. *Classification and regression trees*. Routledge, 2017.

[27] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[28] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. `https://arxiv.org/abs/1606.01540`, 2016. arXiv:1606.01540.

[29] Daniel Brooks, Abraham Shultz, Munjal Desai, Philip Kovac, and Holly A Yanco. Towards state summarization for autonomous robots. In *2010 AAAI Fall Symposium Series*, 2010.

[30] Alexander Brown and Marek Petrik. Interpretable reinforcement learning with ensemble methods. *arXiv preprint arXiv:1809.06995*, 2018.

[31] Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on psychological science*, 6(1):3–5, 2011.

[32] Eduardo Candela, Olivier Doustaly, Leandro Parada, Felix Feng, Yiannis Demiris, and Panagiotis Angeloudis. Risk-aware controller for autonomous vehicles using model-based collision prediction and reinforcement learning. *Artificial Intelligence*, 320:103923, 2023.

[33] Longbing Cao. Ai in finance: challenges, techniques, and opportunities. *ACM Computing Surveys (CSUR)*, 55(3):1–38, 2022.

[34] Thomas Cederborg, Ishaan Grover, Charles L Isbell Jr, and Andrea Lockerd Thomaz. Policy shaping with human teachers. In *IJCAI*, pages 3366–3372, 2015.

[35] Ahmad Chaddad, Jihao Peng, Jian Xu, and Ahmed Bouridane. Survey of explainable ai techniques in healthcare. *Sensors*, 23(2):634, 2023.

[36] Tathagata Chakraborti, Sarath Sreedharan, and Subbarao Kambhampati. The emerging landscape of explainable ai planning and decision making. *arXiv preprint arXiv:2002.11697*, 2020.

[37] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 world wide web conference*, pages 1583–1592, 2018.

[38] Jianwei Chen, Kezhi Li, Pau Herrero, Taiyu Zhu, and Pantelis Georgiou. Dilated recurrent neural network for short-time prediction of glucose concentration. In *KDH@ IJCAI*, pages 69–73, 2018.

[39] Valerie Chen, Abhinav Gupta, and Kenneth Marino. Ask your humans: Using human instructions to improve generalization in reinforcement learning. *arXiv preprint arXiv:2011.00517*, 2020.

[40] Ziheng Chen, Fabrizio Silvestri, Gabriele Tolomei, He Zhu, Jia Wang, and Hongshik Ahn. Relace: Reinforcement learning agent for counterfactual explanations of arbitrary predictive models. *arXiv preprint arXiv:2110.11960*, 2021.

[41] Jaegul Choo and Shixia Liu. Visual analytics for explainable deep learning. *IEEE computer graphics and applications*, 38(4):84–92, 2018.

[42] Alessandro Cimatti, Marco Pistore, and Paolo Traverso. Automated planning. *Foundations of Artificial Intelligence*, 3:841–867, 2008.

[43] Diego Colombo, Marloes H Maathuis, Markus Kalisch, and Thomas S Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, pages 294–321, 2012.

[44] Ivan Contreras and Josep Vehi. Artificial intelligence for diabetes management and decision support: literature review. *Journal of medical Internet research*, 20(5):e10775, 2018.

[45] Diabetes Control, Complications Trial, Epidemiology of Diabetes Interventions, Complications (DCCT/EDIC) Research Group, et al. Risk factors for cardiovascular disease in type 1 diabetes. *Diabetes*, 65(5):1370, 2016.

[46] Youri Coppens, Kyriakos Efthymiadis, Tom Lenaerts, Ann Nowé, Tim Miller, Rosina Weber, and Daniele Magazzeni. Distilling deep reinforcement learning policies in soft decision trees. In *Proceedings of the IJCAI 2019 workshop on explainable artificial intelligence*, pages 1–6, 2019.

[47] Francisco Cruz, Richard Dazeley, and Peter Vamplew. Memory-based explainable reinforcement learning. In *AI 2019: Advances in Artificial Intelligence: 32nd Australasian Joint Conference, Adelaide, SA, Australia, December 2–5, 2019, Proceedings 32*, pages 66–77. Springer, 2019.

[48] Yinglong Dai, Haibin Ouyang, Hong Zheng, Han Long, and Xiaojun Duan. Interpreting a deep reinforcement learning model with conceptual embedding and performance analysis. *Applied Intelligence*, 53(6):6936–6952, 2023.

[49] Elena Daskalaki, Peter Diem, and Stavroula G Mougiakakou. An actor–critic based controller for glucose regulation in type 1 diabetes. *Computer methods and programs in biomedicine*, 109(2):116–125, 2013.

[50] Richard Dazeley, Peter Vamplew, and Francisco Cruz. Explainable reinforcement learning for broad-xai: a conceptual framework and survey. *Neural Computing and Applications*, pages 1–24, 2023.

[51] Alejandra de-la Rica-Escudero, Eduardo C Garrido-Merchán, and María Coronado-Vaca. Explainable post hoc portfolio management financial policy of a deep reinforcement learning agent. *PloS one*, 20(1):e0315528, 2025.

[52] Giovanni De Toni, Bruno Lepri, and Andrea Passerini. Synthesizing explainable counterfactual policies for algorithmic recourse with program synthesis. *Machine Learning*, 112(4):1389–1409, 2023.

[53] Saurabh Deshpande, Rahee Walambe, Ketan Kotecha, and Marina Marjanović Jakovljević. Post-hoc explainable reinforcement learning using probabilistic graphical models. In *International Advanced Computing Conference*, pages 362–376. Springer, 2021.

[54] Arnaud Dethise, Marco Canini, and Srikanth Kandula. Cracking open the black box: What observations can tell us about reinforcement learning agents. In *Proceedings of the 2019 Workshop on Network Meets AI & ML*, pages 29–36, 2019.

[55] Thomas G Dietterich and Nicholas S Flann. Explanation-based learning and reinforcement learning: A unified view. *Machine Learning*, 28:169–210, 1997.

[56] Jonathan Dodge, Andrew Anderson, Roli Khanna, Jed Irvine, Rupika Dikkala, Kin-Ho Lam, Delyar Tabatabai, Anita Ruangrotsakun, Zeyad Shureih, Minsuk Kahng, et al. From "no clear winner" to an effective explainable artificial intelligence process: An empirical journey. *Applied AI Letters*, 2(4):e36, 2021.

[57] Thomas Dodson, Nicholas Mattei, and Judy Goldsmith. A natural language argumentation interface for explanation generation in markov decision processes. In *Algorithmic Decision Theory: Second International*

Conference, ADT 2011, Piscataway, NJ, USA, October 26-28, 2011. Proceedings 2, pages 42–55. Springer, 2011.

[58] Yinpeng Dong, Hang Su, Jun Zhu, and Fan Bao. Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv preprint arXiv:1708.05493*, 2017.

[59] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. Explainable artificial intelligence: A survey. In *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 0210–0215. IEEE, 2018.

[60] Upol Ehsan, Pradyumna Tambwekar, Larry Chan, Brent Harrison, and Mark O Riedl. Automated rationale generation: a technique for explainable ai and its effects on human perceptions. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, pages 263–274, 2019.

[61] Francisco Elizalde, Enrique Sucar, Julieta Noguez, and Alberto Reyes. Generating explanations based on markov decision processes. In *MICAI 2009: Advances in Artificial Intelligence: 8th Mexican International Conference on Artificial Intelligence, Guanajuato, México, November 9-13, 2009. Proceedings 8*, pages 51–62. Springer, 2009.

[62] Francisco Elizalde, L Enrique Sucar, Manuel Luque, J Diez, and Alberto Reyes. Policy explanation in factored markov decision processes. In *Proceedings of the 4th European workshop on probabilistic graphical models (PGM 2008)*, pages 97–104, 2008.

[63] Harry Emerson, Matthew Guy, and Ryan McConville. Offline reinforcement learning for safer blood glucose control in people with type 1 diabetes. *Journal of Biomedical Informatics*, 142:104376, 2023.

[64] Harry Emerson, Sam Gordon James, Matthew Guy, and Ryan McConville. Flexible blood glucose control: Offline reinforcement learning from human feedback. *arXiv preprint arXiv:2501.15972*, 2025.

[65] Raphael C Engelhardt, Moritz Lange, Laurenz Wiskott, and Wolfgang Konen. Sample-based rule extraction for explainable reinforcement learning. In *International Conference on Machine Learning, Optimization, and Data Science*, pages 330–345. Springer, 2022.

[66] Martin Erwig, Alan Fern, Magesh Murali, and Anurag Koul. Explaining deep adaptive programs via reward decomposition. In *IJCAI/ECAI workshop on explainable artificial intelligence*, 2018.

[67] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al.

Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, pages 1407–1416. PMLR, 2018.

[68] Felix Feit, Andreas Metzger, and Klaus Pohl. Explaining online reinforcement learning decisions of self-adaptive systems. In *2022 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*, pages 51–60, 2022.

[69] Richard E Fikes and Nils J Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208, 1971.

[70] Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Learning visual feature spaces for robotic manipulation with deep spatial autoencoders. *CoRR*, abs/1509.06113, 2015.

[71] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[72] Ian Fox, Joyce Lee, Rodica Pop-Busui, and Jenna Wiens. Deep reinforcement learning for closed-loop blood glucose control. In *Machine Learning for Healthcare Conference*, pages 508–536. PMLR, 2020.

[73] Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.

[74] Yosuke Fukuchi, Masahiko Osawa, Hiroshi Yamakawa, and Michita Imai. Autonomous self-explanation of behavior for interactive reinforcement learning agents. In *Proceedings of the 5th International Conference on Human Agent Interaction*, pages 97–101, 2017.

[75] Nathan Fulton and André Platzer. Safe reinforcement learning via formal methods: Toward safe control through proof and learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[76] Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

[77] Marta Garnelo, Kai Arulkumaran, and Murray Shanahan. Towards deep symbolic reinforcement learning. *arXiv preprint arXiv:1609.05518*, 2016.

[78] Julie Gerlings, Millie Søndergaard Jensen, and Arisa Shollo. Explainable ai, but explainable to whom? an exploratory case study of xai in healthcare. *Handbook of Artificial Intelligence in Healthcare: Vol 2: Practicalities and Prospects*, pages 169–198, 2022.

[79] Marzyeh Ghassemi, Luke Oakden-Rayner, and Andrew L Beam. The false hope of current approaches to explainable artificial intelligence in health care. *The Lancet Digital Health*, 3(11):e745–e750, 2021.

[80] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018.

[81] Claire Glanois, Paul Weng, Matthieu Zimmer, Dong Li, Tianpei Yang, Jianye Hao, and Wulong Liu. A survey on interpretable reinforcement learning. *arXiv preprint arXiv:2112.13112*, 2021.

[82] Leela Prasad Gorrepati and Ravi Teja Potla. A reinforcement learning framework for real-time personalized treatment planning in clinical environments. *Engineering, Technology & Applied Science Research*, 15(4):24698–24704, 2025.

[83] Abhijit Gosavi. Reinforcement learning: A tutorial survey and recent advances. *INFORMS Journal on Computing*, 21(2):178–192, 2009.

[84] Alicja Gosiewska, Anna Kozak, and Przemysław Biecek. Simpler is better: Lifting interpretability-performance trade-off via automated feature engineering. *Decision Support Systems*, 150:113556, 2021. Interpretable Data Science For Decision Making.

[85] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.

[86] Ole-Christoffer Granmo. The tsetlin machine–a game theoretic bandit driven approach to optimal pattern recognition with propositional logic. *arXiv preprint arXiv:1804.01508*, 2018.

[87] Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding atari agents. In *International conference on machine learning*, pages 1792–1801. PMLR, 2018.

[88] Shane Griffith, Kaushik Subramanian, Jonathan Scholz, Charles L Isbell, and Andrea L Thomaz. Policy shaping: Integrating human feedback with reinforcement learning. *Advances in neural information processing systems*, 26, 2013.

[89] Lin Guan, Mudit Verma, Suna Sihang Guo, Ruohan Zhang, and Subbarao Kambhampati. Widening the pipeline in human-guided reinforcement learning with explanation and context-aware data augmentation. *Advances in Neural Information Processing Systems*, 34:21885–21897, 2021.

[90] Lin Guan, Mudit Verma, and Subbarao Kambhampati. Explanation augmented feedback in human-in-the-loop reinforcement learning. *arXiv preprint arXiv:2006.14804*, 2020.

[91] Yang Guan, Yangang Ren, Qi Sun, Shengbo Eben Li, Haitong Ma, Jingliang Duan, Yifan Dai, and Bo Cheng. Integrated decision and control: Toward interpretable and computationally efficient driving intelligence. *IEEE Transactions on Cybernetics*, 53(2):859–873, 2023.

[92] Wenbo Guo, Xian Wu, Usmann Khan, and Xinyu Xing. Edge: Explaining deep reinforcement learning policies. *Advances in Neural Information Processing Systems*, 34:12222–12236, 2021.

[93] Aric Hagberg, Pieter J Swart, and Daniel A Schult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), 2008.

[94] G. Hailu and G. Sommer. Integrating symbolic knowledge in reinforcement learning. In *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)*, volume 2, pages 1491–1496 vol.2, 1998.

[95] Nikolaus Hansen. The cma evolution strategy: a comparing review. *Towards a new evolutionary computation: Advances in the estimation of distribution algorithms*, pages 75–102, 2006.

[96] Bradley Hayes and Julie A Shah. Improving robot controller transparency through autonomous policy explanation. In *Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction*, pages 303–312, 2017.

[97] Daniel Hein, Alexander Hentschel, Thomas Runkler, and Steffen Udluft. Particle swarm optimization for generating interpretable fuzzy reinforcement learning policies. *Engineering Applications of Artificial Intelligence*, 65:87–98, 2017.

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

[99] Daniel Hein, Steffen Udluft, and Thomas A Runkler. Generating interpretable reinforcement learning policies using genetic programming. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 23–24, 2019.

[100] Rasha Hendawi, Juan Li, Souradip Roy, et al. A mobile app that addresses interpretability challenges in machine learning–based diabetes predictions: survey-based user study. *JMIR Formative Research*, 7(1):e50328, 2023.

[101] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[102] Alexandre Heuillet, Fabien Couthouis, and Natalia Díaz-Rodríguez. Explainability in deep reinforcement learning. *Knowledge-Based Systems*, 214:106685, 2021.

[103] Alexandre Heuillet, Fabien Couthouis, and Natalia Díaz-Rodríguez. Collective explainable ai: Explaining cooperative strategies and agent contribution in multiagent reinforcement learning with shapley values. *IEEE Computational Intelligence Magazine*, 17(1):59–71, 2022.

[104] Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. Metrics for explainable ai: Challenges and prospects. *arXiv preprint arXiv:1812.04608*, 2018.

[105] Richard IG Holt and Allan Flyvbjerg. *Textbook of diabetes*. John Wiley & Sons, 2024.

[106] Andreas Holzinger. From machine learning to explainable ai. In *2018 world symposium on digital intelligence for systems and machines (DISA)*, pages 55–66. IEEE, 2018.

[107] Andreas Holzinger, Randy Goebel, Ruth Fong, Taesup Moon, Klaus-Robert Müller, and Wojciech Samek. xxai-beyond explainable artificial intelligence. In *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*, pages 3–10. Springer, 2020.

[108] John Wesley Hostetter, Mark Abdelshiheed, Tiffany Barnes, and Min Chi. Leveraging fuzzy logic towards more explainable reinforcement learning-induced pedagogical policies on intelligent tutoring systems. In *2023 IEEE international conference on fuzzy systems*, 2023.

[109] Biwei Huang, Kun Zhang, Jiji Zhang, Joseph Ramsey, Ruben Sanchez-Romero, Clark Glymour, and Bernhard Schölkopf. Causal discovery from heterogeneous/nonstationary data. *Journal of Machine Learning Research*, 21(89):1–53, 2020.

[110] Sandy H Huang, Kush Bhatia, Pieter Abbeel, and Anca D Dragan. Establishing appropriate trust via critical states. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3929–3936. IEEE, 2018.

[111] Sandy H Huang, David Held, Pieter Abbeel, and Anca D Dragan. Enabling robots to communicate their objectives. *Autonomous Robots*, 43:309–326, 2019.

[112] Tobias Huber, Benedikt Limmer, and Elisabeth André. Benchmarking perturbation-based saliency maps for explaining atari agents. *Frontiers in Artificial Intelligence*, 5:903875, 2022.

[113] Tobias Huber, Dominik Schiller, and Elisabeth André. Enhancing explainability of deep reinforcement learning through selective layer-wise relevance propagation. In *KI 2019: Advances in Artificial Intelligence: 42nd German Conference on AI, Kassel, Germany, September 23–26, 2019, Proceedings 42*, pages 188–202. Springer, 2019.

[114] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

[115] Rahul Iyer, Yuezhang Li, Huao Li, Michael Lewis, Ramitha Sundar, and Katia Sycara. Transparency and explanation in deep reinforcement learning neural networks. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 144–150, 2018.

[116] Peter G Jacobs, Pau Herrero, Andrea Facchinetti, Josep Vehi, Boris Kovatchev, Marc Breton, Ali Cinar, Konstantina Nikita, Frank Doyle, Jorge Bondia, et al. Artificial intelligence and machine learning for improving glycemic control in diabetes: best practices, pitfalls and opportunities. *IEEE reviews in biomedical engineering*, 2023.

[117] Theo Jaunet, Romain Vuillemot, and Christian Wolf. Drlviz: Understanding decisions and memory in deep reinforcement learning. In *Computer Graphics Forum*, volume 39, pages 49–61. Wiley Online Library, 2020.

[118] Pushkala Jayaraman, Jacob Desman, Moein Sabounchi, Girish N Nadkarni, and Ankit Sakhuja. A primer on reinforcement learning in medicine for clinicians. *NPJ Digital Medicine*, 7(1):337, 2024.

[119] Chenyang Jiang, Bowen Hu, Yazhen Wang, and Shang Wu. Reinforcement learning via nonparametric smoothing in a continuous-time stochastic setting with noisy data. *Statistica Sinica*, 35:831–852, 2025.

[120] Peng Jin, Jiaxu Tian, Dapeng Zhi, Xuejun Wen, and Min Zhang. Trainify: A cegar-driven training and verification framework for safe deep reinforcement learning. In *International Conference on Computer Aided Verification*, pages 193–218. Springer, 2022.

[121] Jinyu Xie. Simglucose v0.2.1 (2018). Accessed: 2024-06-05.

[122] Zoe Juozapaitis, Anurag Koul, Alan Fern, Martin Erwig, and Finale Doshi-Velez. Explainable reinforcement learning via reward decomposition. In *IJCAI/ECAI Workshop on explainable artificial intelligence*, 2019.

[123] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[124] Frank Kaptein, Joost Broekens, Koen Hindriks, and Mark Neerincx. The role of emotion in self-explanations by cognitive agents. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 88–93. IEEE, 2017.

[125] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*, pages 97–117. Springer, 2017.

[126] Guy Katz, Derek A Huang, Duligur Ibeling, Kyle Julian, Christopher Lazarus, Rachel Lim, Parth Shah, Shantanu Thakoor, Haoze Wu, Aleksandar Zeljić, et al. The marabou framework for verification and analysis of deep neural networks. In *Computer Aided Verification: 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part I 31*, pages 443–452. Springer, 2019.

[127] Yafim Kazak, Clark Barrett, Guy Katz, and Michael Schapira. Verifying deep-rl-driven systems. In *Proceedings of the 2019 workshop on network meets AI & ML*, pages 83–89, 2019.

[128] Dmitry Kazhdan, Zohreh Shams, and Pietro Liò. Marleme: A multi-agent reinforcement learning model extraction library. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[129] Omar Khan, Pascal Poupart, and James Black. Minimal sufficient explanations for factored markov decision processes. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 19, pages 194–200, 2009.

[130] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. *Advances in neural information processing systems*, 29, 2016.

[131] Jinkyu Kim, Suhong Moon, Anna Rohrbach, Trevor Darrell, and John Canny. Advisable learning for self-driving vehicles by internalizing observation-to-action rules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9661–9670, 2020.

[132] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles. In *Proceedings of the European conference on computer vision (ECCV)*, pages 563–578, 2018.

[133] Jens Kober and Jan Peters. Imitation and reinforcement learning. *IEEE Robotics & Automation Magazine*, 17(2):55–62, 2010.

[134] Leon Kopitar, Leona Cilar, Primoz Kocbek, and Gregor Stiglic. Local vs. global interpretability of machine learning models in type 2 diabetes mellitus screening. In *International Workshop on Knowledge Representation for Health Care*, pages 108–119. Springer, 2019.

[135] Anurag Koul, Sam Greydanus, and Alan Fern. Learning finite state representations of recurrent policy networks. *arXiv preprint arXiv:1811.12530*, 2018.

[136] Boris Kovatchev. A century of diabetes technology: signals, models, and artificial pancreas control. *Trends in Endocrinology & Metabolism*, 30(7):432–444, 2019.

[137] Viktoriya Krakovna and Finale Doshi-Velez. Increasing the interpretability of recurrent neural networks using hidden markov models. *arXiv preprint arXiv:1606.05320*, 2016.

[138] Maik Kschischo and Philipp Wendland. Counterfactual ai for dynamic dose optimization with side-effect constraints. *Authorea Preprints*, 2025.

[139] Tejas D. Kulkarni, Ankush Gupta, Catalin Ionescu, Sebastian Borgeaud, Malcolm Reynolds, Andrew Zisserman, and Volodymyr Mnih. Unsupervised learning of object keypoints for perception and control. *CoRR*, abs/1906.11883, 2019.

[140] Satyam Kumar, Mendhikar Vishal, and Vadlamani Ravi. Explainable reinforcement learning on financial stock trading using shap. *arXiv preprint arXiv:2208.08790*, 2022.

[141] Sushant Kumar, Sumit Datta, Vishakha Singh, Deepanwita Datta, Sanjay Kumar Singh, and Ritesh Sharma. Applications, challenges, and future directions of human-in-the-loop learning. *IEEE Access*, 12:75735–75760, 2024.

[142] Richard Lachman and Michael Joffe. Applications of artificial intelligence in media and entertainment. In *Analyzing future applications of AI, sensors, and robotics in society*, pages 201–220. IGI Global, 2021.

[143] Isaac Lage, Daphna Lifschitz, Finale Doshi-Velez, and Ofra Amir. Exploring computational user models for agent policy summarization. In *IJCAI: proceedings of the conference*, volume 28, page 1401. NIH Public Access, 2019.

[144] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253, 2017.

[145] Mikel Landajuela, Brenden K Petersen, Sookyung Kim, Claudio P Santiago, Ruben Glatt, Nathan Mundhenk, Jacob F Pettit, and Daniel Faissol.

Discovering symbolic policies with deep reinforcement learning. In *International Conference on Machine Learning*, pages 5979–5989. PMLR, 2021.

[146] Xavier Larriva-Novo, Luis Pérez Miguel, Victor A Villagra, Manuel Álvarez-Campana, Carmen Sanchez-Zas, and Óscar Jover. Post-hoc categorization based on explainable ai and reinforcement learning for improved intrusion detection. *Applied Sciences*, 14(24):11511, 2024.

[147] Jiahui Li, Kun Kuang, Baoxiang Wang, Furui Liu, Long Chen, Fei Wu, and Jun Xiao. Shapley counterfactual credits for multi-agent reinforcement learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 934–942, 2021.

[148] Junbing Li, Changqing Zhang, Joey Tianyi Zhou, Huazhu Fu, Shuyin Xia, and Qinghua Hu. Deep-lift: Deep label-specific feature learning for image annotation. *IEEE transactions on Cybernetics*, 52(8):7732–7741, 2021.

[149] Kezhi Li, Chengyuan Liu, Taiyu Zhu, Pau Herrero, and Pantelis Georgiou. Glunet: A deep learning framework for accurate glucose forecasting. *IEEE journal of biomedical and health informatics*, 24(2):414–423, 2019.

[150] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[151] Min Hyuk Lim, Woo Hyung Lee, Byoungjun Jeon, and Sungwan Kim. A blood glucose control framework based on reinforcement learning with safety and interpretability: In silico validation. *IEEE Access*, 9:105756–105775, 2021.

[152] Baihan Lin, Guillermo Cecchi, and Djallel Bouneffouf. Psychotherapy ai companion with reinforcement learning recommendations and interpretable policy dynamics. WWW '23 Companion, page 932–939, New York, NY, USA, 2023. Association for Computing Machinery.

[153] Zhengxian Lin, Kim-Ho Lam, and Alan Fern. Contrastive explanations for reinforcement learning via embedded self predictions. *arXiv preprint arXiv:2010.05180*, 2020.

[154] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

[155] Guiliang Liu, Oliver Schulte, Wang Zhu, and Qingcan Li. Toward interpretable deep reinforcement learning with linear model u-trees. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part II 18*, pages 414–429. Springer, 2019.

[156] Ran Liu, Joseph L Greenstein, James C Fackler, Jules Bergmann, Melania M Bembea, and Raimond L Winslow. Offline reinforcement learning with uncertainty for treatment strategies in sepsis. *arXiv preprint arXiv:2107.04491*, 2021.

[157] Shixia Liu, Xiting Wang, Mengchen Liu, and Jun Zhu. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics*, 1(1):48–56, 2017.

[158] Xiao Liu, Shuyang Liu, Bo An, Yang Gao, Shangdong Yang, and Wenbin Li. Effective interpretable policy distillation via critical experience point identification. *IEEE Intelligent Systems*, pages 1–10, 2023.

[159] Meghann Lomas, Robert Chevalier, Ernest Vincent Cross, Robert Christopher Garrett, John Hoare, and Michael Kopack. Explaining robot actions. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 187–188, 2012.

[160] Luca Longo, Randy Goebel, Freddy Lecue, Peter Kieseberg, and Andreas Holzinger. Explainable artificial intelligence: Concepts, applications, research challenges and visions. In *International cross-domain conference for machine learning and knowledge extraction*, pages 1–16. Springer, 2020.

[161] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

[162] SJA Stefan Luijten, V Vlado Menkovski, and M Marwan Hassani. Improving therapy adherence using smart persuasive interventions generated by an rnn-rl framework.

[163] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.

[164] Björn Lütjens, Michael Everett, and Jonathan P How. Safe reinforcement learning with model uncertainty estimates. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8662–8668. IEEE, 2019.

[165] Daoming Lyu, Fangkai Yang, Bo Liu, and Steven Gustafson. Sdrl: interpretable and data-efficient deep reinforcement learning leveraging symbolic planning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2970–2977, 2019.

[166] Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. Explainable reinforcement learning through a causal lens. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 2493–2500, 2020.

[167] Francis Maes, Raphael Fonteneau, Louis Wehenkel, and Damien Ernst. Policy search in a space of simple closed-form formulas: Towards interpretability of reinforcement learning. In *Discovery Science: 15th International Conference, DS 2012, Lyon, France, October 29-31, 2012. Proceedings 15*, pages 37–51. Springer, 2012.

[168] Dianna J. Magliano, Edward J. Boyko, and IDF Diabetes Atlas 10th edition scientific committee. Idf diabetes atlas, 2021.

[169] Lalo Magni, Davide M Raimondo, Chiara Dalla Man, Marc Breton, Stephen Patek, Giuseppe De Nicolao, Claudio Cobelli, and Boris P Kovatchev. Evaluating the efficacy of closed-loop glucose regulation via control-variability grid analysis. *Journal of diabetes science and technology*, 2(4):630–635, 2008.

[170] A Rupam Mahmood, Dmytro Korenkevych, Gautham Vasan, William Ma, and James Bergstra. Benchmarking reinforcement learning algorithms on real-world robots. In *Conference on robot learning*, pages 561–591. PMLR, 2018.

[171] Chiara Dalla Man, Francesco Micheletto, Dayu Lv, Marc Breton, Boris Kovatchev, and Claudio Cobelli. The uva/padova type 1 diabetes simulator: new features. *Journal of diabetes science and technology*, 8(1):26–34, 2014.

[172] Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search of static linear policies is competitive for reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

[173] Yishay Mansour, Michal Moshkovitz, and Cynthia Rudin. There is no accuracy-interpretability tradeoff in reinforcement learning for mazes, 2022.

[174] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. Neural adaptive video streaming with pensieve. In *Proceedings of the conference of the ACM special interest group on data communication*, pages 197–210, 2017.

[175] Moshe Mash, Raz Lin, and David Sarne. Peer-design agents for reliably evaluating distribution of outcomes in environments involving people. In *Proceedings of the 2014 international conference on autonomous agents and multi-agent systems*, pages 949–956, 2014.

[176] Marco Matarese, Silvia Rossi, Alessandra Sciutti, and Francesco Rea. Towards transparency of td-rl robotic systems with a human teacher. *arXiv preprint arXiv:2005.05926*, 2020.

[177] Roshan P Mathews, Mahesh Raveendranatha Panicker, Abhilash R Hareendranathan, Yale Tung Chen, Jacob L Jaremko, Brian Buchanan, Kiran Vishnu Narayan, Greeta Mathews, et al. Unsupervised multi-latent space rl framework for video summarization in ultrasound imaging. *IEEE Journal of Biomedical and Health Informatics*, 27(1):227–238, 2022.

[178] Yutaka Matsuo, Yann LeCun, Maneesh Sahani, Doina Precup, David Silver, Masashi Sugiyama, Eiji Uchibe, and Jun Morimoto. Deep learning, reinforcement learning, and world models. *Neural Networks*, 152:267–275, 2022.

[179] Daniele Melloni and Andrea Zingoni. Interpreting type 1 diabetes management via contrastive explanations. In *2024 IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE)*, pages 692–697. IEEE, 2024.

[180] Rudy Milani, Maximilian Moll, Renato De Leone, and Stefan Pickl. A bayesian network approach to explainable reinforcement learning with distal information. *Sensors*, 23(4):2013, 2023.

[181] Stephanie Milani, Nicholay Topin, Manuela Veloso, and Fei Fang. A survey of explainable reinforcement learning. *arXiv preprint arXiv:2202.08434*, 2022.

[182] Stephanie Milani, Nicholay Topin, Manuela Veloso, and Fei Fang. Explainable reinforcement learning: A survey and comparative review. *ACM Computing Surveys*, 56(7):1–36, 2024.

[183] Stephanie Milani, Zhicheng Zhang, Nicholay Topin, Zheyuan Ryan Shi, Charles Kamhoua, Evangelos E Papalexakis, and Fei Fang. Maviper: Learning decision tree policies for interpretable multi-agent reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 251–266. Springer, 2022.

[184] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[185] Tim Miller. Contrastive explanation: A structural-model approach. *The Knowledge Engineering Review*, 36:e14, 2021.

[186] Indrajeet Mishra, Giang Dao, and Minwoo Lee. Visual sparse bayesian reinforcement learning: a framework for interpreting what an agent has learned. In *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1427–1434. IEEE, 2018.

[187] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[188] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[189] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[190] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 193–209, 2019.

[191] Joris M Mooij, Jonas Peters, Dominik Janzing, Jakob Zscheischler, and Bernhard Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *Journal of Machine Learning Research*, 17(32):1–102, 2016.

[192] Alexander Mott, Daniel Zoran, Mike Chrzanowski, Daan Wierstra, and Danilo Jimenez Rezende. Towards interpretable reinforcement learning using attention augmented agents. *Advances in neural information processing systems*, 32, 2019.

[193] Maria Movin, Guilherme Dinis Junior, Jaakko Hollmén, and Panagiotis Papapetrou. Explaining black box reinforcement learning agents through counterfactual policies. In *International Symposium on Intelligent Data Analysis*, pages 314–326. Springer, 2023.

[194] Dena F Mujtaba and Nihar R Mahapatra. Ethical considerations in ai-based recruitment. In *2019 IEEE International Symposium on Technology and Society (ISTAS)*, pages 1–7. IEEE, 2019.

[195] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

[196] Geraud Nangue Tasse, Steven James, and Benjamin Rosman. A boolean task algebra for reinforcement learning. *Advances in Neural Information Processing Systems*, 33:9497–9507, 2020.

[197] Mark A Neerincx, Jasper van der Waa, Frank Kaptein, and Jurriaan van Diggelen. Using perceptual and cognitive explanations for enhanced human-agent team performance. In *Engineering Psychology and Cognitive Ergonomics: 15th International Conference, EPCE 2018, Held as Part of HCI International 2018, Las Vegas, NV, USA, July 15-20, 2018, Proceedings 15*, pages 204–214. Springer, 2018.

[198] Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[199] Dmitry Nikulin, Anastasia Ianina, Vladimir Aliev, and Sergey Nikolenko. Free-lunch saliency via attention in atari agents. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4240–4249. IEEE, 2019.

[200] Martin Obschonka and David B Audretsch. Artificial intelligence and big data in entrepreneurship: a new era has begun. *Small Business Economics*, 55:529–539, 2020.

[201] Matthew L Olson, Roli Khanna, Lawrence Neal, Fuxin Li, and Weng-Keen Wong. Counterfactual state explanations for reinforcement learning agents via generative deep learning. *Artificial Intelligence*, 295:103455, 2021.

[202] Shane O'Sullivan, Nathalie Nevejans, Colin Allen, Andrew Blyth, Simon Leonard, Ugo Pagallo, Katharina Holzinger, Andreas Holzinger, Mohammed Imran Sajid, and Hutan Ashrafian. Legal, regulatory, and ethical frameworks for development of standards in artificial intelligence (ai) and autonomous robotic surgery. *The international journal of medical robotics and computer assisted surgery*, 15(1):e1968, 2019.

[203] Menghai Pan, Weixiao Huang, Yanhua Li, Xun Zhou, and Jun Luo. xgail: Explainable generative adversarial imitation learning for explainable human decision analysis. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1334–1343, 2020.

[204] Xinlei Pan, Xiangyu Chen, Qizhi Cai, John Canny, and Fisher Yu. Semantic predictive control for explainable and efficient policy learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3203–3209. IEEE, 2019.

[205] Debdeep Paul, Chandra Wijaya, Sahim Yamaura, Koji Miura, and Yosuke Tajika. Markov chain based explainable pattern forecasting. In *IECON 2023-49th Annual Conference of the IEEE Industrial Electronics Society*, pages 1–7. IEEE, 2023.

[206] Judea Pearl. Causality: Models, rreasoning, and inference. *Econometric Theory*, 19(4):675–685, 2000.

[207] Judea Pearl. Causal inference. *Causality: objectives and assessment*, pages 39–58, 2010.

[208] Xiangyu Peng, Mark Riedl, and Prithviraj Ammanabrolu. Inherently explainable reinforcement learning in natural language. *Advances in Neural Information Processing Systems*, 35:16178–16190, 2022.

[209] Carmen Pérez-Gandía, Andrea Facchinetti, Giovanni Sparacino, Claudio Cobelli, EJ Gómez, M Rigla, Alberto de Leiva, and ME Hernando. Artificial neural network algorithm for online glucose prediction from continuous glucose monitoring. *Diabetes technology & therapeutics*, 12(1):81–88, 2010.

[210] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[211] Marek Petrik and Ronny Luss. Interpretable policies for dynamic product recommendations. In *UAI*, 2016.

[212] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

[213] Hamed Pirsiavash and Deva Ramanan. Parsing videos of actions with segmental grammars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 612–619, 2014.

[214] Gregory Plumb, Maruan Al-Shedivat, Ángel Alexander Cabrera, Adam Perer, Eric Xing, and Ameet Talwalkar. Regularizing black-box models for improved interpretability. *Advances in Neural Information Processing Systems*, 33:10526–10536, 2020.

[215] Rey Pocius, Lawrence Neal, and Alan Fern. Strategic tasks for explainable reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 10007–10008, 2019.

[216] Mattia Prosperi, Yi Guo, Matt Sperrin, James S Koopman, Jae S Min, Xing He, Shannan Rich, Mo Wang, Iain E Buchan, and Jiang Bian. Causal inference and counterfactual prediction in machine learning for actionable healthcare. *Nature Machine Intelligence*, 2(7):369–375, 2020.

[217] Pearl Pu and Li Chen. Trust building with explanation interfaces. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 93–100, 2006.

[218] Erika Puiutta and Eric MSP Veith. Explainable reinforcement learning: A survey. In *Machine Learning and Knowledge Extraction: 4th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2020, Dublin, Ireland, August 25–28, 2020, Proceedings 4*, pages 77–95. Springer, 2020.

[219] Larry D Pyeatt, Adele E Howe, et al. Decision tree function approximation in reinforcement learning. In *Proceedings of the third international symposium on adaptive systems: evolutionary computation and probabilistic graphical models*, volume 2, pages 70–77. Cuba, 2001.

[220] Yunpeng Qing, Shunyu Liu, Jie Song, and Mingli Song. A survey on explainable reinforcement learning: Concepts, algorithms, challenges. *arXiv preprint arXiv:2211.06665*, 2022.

[221] Griselda Quiroz. The evolution of control algorithms in artificial pancreas: A historical perspective. *Annual Reviews in Control*, 48:222–232, 2019.

[222] Saeed Rahimi Gorji and Ole-Christoffer Granmo. Off-policy and on-policy reinforcement learning with the tsetlin machine. *Applied Intelligence*, pages 1–18, 2023.

[223] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[224] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[225] Aaron M Roth, Nicholay Topin, Pooyan Jamshidi, and Manuela Veloso. Conservative q-improvement: Reinforcement learning for an interpretable decision-tree policy. *arXiv preprint arXiv:1907.01180*, 2019.

[226] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, 2019.

[227] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242, 1999.

[228] Anton Schwartz. A reinforcement learning method for maximizing undiscounted rewards. In *Proceedings of the tenth international conference on machine learning*, volume 298, pages 298–305, 1993.

[229] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[230] Pedro Sequeira and Melinda Gervasio. Interestingness elements for explainable reinforcement learning: Understanding agents' capabilities and limitations. *Artificial Intelligence*, 288:103367, 2020.

[231] Murray Shanahan. Perception as abduction: Turning sensor data into meaningful representation. *Cognitive science*, 29(1):103–134, 2005.

[232] Raymond K Sheh. Different xai for different hri. In *2017 AAAI Fall Symposium Series*, 2017.

[233] Wenjie Shi, Zhuoyuan Wang, Shiji Song, and Gao Huang. Self-supervised discovering of causal features: Towards interpretable reinforcement learning. *arXiv preprint arXiv:2003.07069*, 2020.

[234] Xiaoying Shi, Jiaming Zhang, Ziyi Liang, and Dewen Seng. Maddpgviz: a visual analytics approach to understand multi-agent deep reinforcement learning. *Journal of Visualization*, pages 1–17, 2023.

[235] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMlR, 2017.

[236] Tianmin Shu, Caiming Xiong, and Richard Socher. Hierarchical and interpretable skill acquisition in multi-task reinforcement learning. *arXiv preprint arXiv:1712.07294*, 2017.

[237] Zhangzhang Si, Mingtao Pei, Benjamin Yao, and Song-Chun Zhu. Unsupervised learning of event and-or grammar and semantics from video. In *2011 International Conference on Computer Vision*, pages 41–48. IEEE, 2011.

[238] Andrew Silva, Taylor Killian, Ivan Dario Jimenez Rodriguez, Sung-Hyun Son, and Matthew Gombolay. Optimization methods for interpretable differentiable decision trees in reinforcement learning. *arXiv preprint arXiv:1903.09338*, 2019.

[239] Tom Silver, Kelsey R Allen, Alex K Lew, Leslie Pack Kaelbling, and Josh Tenenbaum. Few-shot bayesian imitation learning with logical program policies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10251–10258, 2020.

[240] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[241] Julian Skirzyński, Frederic Becker, and Falk Lieder. Automatic discovery of interpretable planning strategies. *Machine Learning*, 110:2641–2683, 2021.

[242] Benjamin Smith, Anahita Khojandi, and Rama Vasudevan. Bias in reinforcement learning: A review in healthcare applications. *ACM Computing Surveys*, 2023.

[243] Shagun Sodhani, Amy Zhang, and Joelle Pineau. Multi-task reinforcement learning with context-based representations. In *International Conference on Machine Learning*, pages 9767–9779. PMLR, 2021.

[244] Kacper Sokol and Peter Flach. One explanation does not fit all: The promise of interactive explanations for machine learning transparency. *KI-Künstliche Intelligenz*, 34(2):235–250, 2020.

[245] Zitao Song, Yining Wang, Pin Qian, Sifan Song, Frans Coenen, Zhengyong Jiang, and Jionglong Su. From deterministic to stochastic: an interpretable stochastic model-free reinforcement learning framework for portfolio optimization. *Applied Intelligence*, pages 1–16, 2022.

[246] Ivan Sorokin, Alexey Seleznev, Mikhail Pavlov, Aleksandr Fedorov, and Anastasiia Ignateva. Deep attention recurrent q-network. *arXiv preprint arXiv:1512.01693*, 2015.

[247] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2001.

[248] Sarath Sreedharan, Siddharth Srivastava, and Subbarao Kambhampati. Hierarchical expertise level modeling for user specific contrastive explanations. In *IJCAI*, pages 4829–4836, 2018.

[249] Gregory Stein. Generating high-quality explanations for navigation in partially-revealed environments. *Advances in neural information processing systems*, 34:17493–17506, 2021.

[250] Pei-Hao Su, Pawel Budzianowski, Stefan Ultes, Milica Gasic, and Steve Young. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. *arXiv preprint arXiv:1707.00130*, 2017.

[251] Luis Enrique Sucar. *Probabilistic graphical models*. Springer, 2021.

[252] Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29, 2016.

[253] Robert S Sullivan and Luca Longo. Explaining deep q-learning experience replay with shapley additive explanations. *Machine Learning and Knowledge Extraction*, 5(4):1433–1455, 2023.

[254] Yue-Wen Sun, Wen-Zhang Liu, and Chang-Yin Sun. Causality in reinforcement learning control: The state of the art and prospects. *Zidonghua Xuebao/Acta Automatica Sinica*, 49(3):661 – 677, 2023. Cited by: 0.

[255] Yuewen Sun, Kun Zhang, and Changyin Sun. Model-based transfer reinforcement learning based on graphical model representations. *IEEE Transactions on Neural Networks and Learning Systems*, 34(2):1035–1048, 2023.

[256] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[257] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. *Robotica*, 17(2):229–235, 1999.

[258] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[259] Richard S Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M Pilarski, Adam White, and Doina Precup. Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 761–768, 2011.

[260] Maxwell Szymanski, Martijn Millecamp, and Katrien Verbert. Visual, textual or hybrid: the effect of user expertise on different explanations. In *26th International Conference on Intelligent User Interfaces*, pages 109–119, 2021.

[261] Aaquib Tabrez and Bradley Hayes. Improving human-robot interaction through explainable reinforcement learning. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 751–753. IEEE, 2019.

[262] Andrea Tacchetti, H Francis Song, Pedro AM Mediano, Vinicius Zambaldi, Neil C Rabinowitz, Thore Graepel, Matthew Botvinick, and Peter W Battaglia. Relational forward models for multi-agent learning. *arXiv preprint arXiv:1809.11044*, 2018.

[263] Yujin Tang, Duong Nguyen, and David Ha. Neuroevolution of self-interpretable agents. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pages 414–424, 2020.

[264] Alex Tank, Ian Covert, Nicholas Foti, Ali Shojaie, and Emily B. Fox. Neural granger causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4267–4279, 2022.

[265] Miguel Tejedor, Sigurd Nordtveit Hjerde, Jonas Nordhaug Myhre, and Fred Godtliebsen. Evaluating deep q-learning algorithms for controlling blood glucose in in silico type 1 diabetes. *Diagnostics*, 13(19):3150, 2023.

[266] Miguel Tejedor, Ashenafi Zebene Woldaregay, and Fred Godtliebsen. Reinforcement learning application in diabetes blood glucose control: A systematic review. *Artificial intelligence in medicine*, 104:101836, 2020.

[267] Andrea Lockerd Thomaz, Guy Hoffman, and Cynthia Breazeal. Real-time interactive reinforcement learning for robots. In *AAAI 2005 workshop on human comprehensible machine learning*, volume 3, 2005.

[268] Erico Tjoa and Cuntai Guan. Self reward design with fine-grained interpretability. *Scientific Reports*, 13(1):1638, 2023.

[269] Nicholay Topin, Stephanie Milani, Fei Fang, and Manuela Veloso. Iterative bounding mdps: Learning interpretable policies via non-interpretable methods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9923–9931, 2021.

[270] Nicholay Topin and Manuela Veloso. Generation of policy-level explanations for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2514–2521, 2019.

[271] Dweep Trivedi, Jesse Zhang, Shao-Hua Sun, and Joseph J Lim. Learning to synthesize programs as interpretable and generalizable policies. *Advances in neural information processing systems*, 34:25146–25163, 2021.

[272] Jaakko Tuomilehto. The emerging global epidemic of type 1 diabetes. *Current diabetes reports*, 13(6):795–804, 2013.

[273] Onder Tutsoy and Martin Brown. An analysis of value function learning with piecewise linear control. *Journal of Experimental & Theoretical Artificial Intelligence*, 28(3):529–545, 2016.

[274] William TB Uther and Manuela M Veloso. Tree based discretization for continuous state space reinforcement learning. *Aaai/iaai*, 98:769–774, 1998.

[275] Chandra Prasetyo Utomo, Xue Li, and Weitong Chen. Treatment recommendation in critical care: A scalable and interpretable approach in partially observable health states. 2018. Cited by: 4.

[276] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[277] Jasper van der Waa, Jurriaan van Diggelen, Karel van den Bosch, and Mark Neerincx. Contrastive explanations for reinforcement learning in terms of expected consequences. *arXiv preprint arXiv:1807.08706*, 2018.

[278] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[279] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[280] Eric MSP Veith, Lars Fischer, Martin Tröschel, and Astrid Nieße. Analyzing cyber-physical systems from the perspective of artificial intelligence. In *Proceedings of the 2019 International Conference on Artificial Intelligence, Robotics and Control*, pages 85–95, 2019.

[281] Abhinav Verma, Vijayaraghavan Murali, Rishabh Singh, Pushmeet Kohli, and Swarat Chaudhuri. Programmatically interpretable reinforcement learning. In *International Conference on Machine Learning*, pages 5045–5054. PMLR, 2018.

[282] Giulia Vilone and Luca Longo. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion*, 76:89–106, 2021.

[283] Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojtek Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, Timo Ewalds, Dan Horgan, Manuel Kroiss, Ivo Danihelka, John Agapiou, Junhyuk Oh, Valentin Dalibard, David Choi, Laurent Sifre, Yury Sulsky, Sasha Vezhnevets, James Molloy, Trevor Cai, David Budden, Tom Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Toby Pohlen, Dani Yogatama, Julia Cohen, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Chris Apps, Koray Kavukcuoglu, Demis Hassabis, and David Silver. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. `https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/`, 2019.

[284] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, et al. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*, 2017.

[285] Dimitrios Vogiatzis and Andreas Stafylopatis. Reinforcement learning for symbolic expression induction. *Mathematics and computers in simulation*, 51(3-4):169–179, 2000.

[286] Cameron Voloshin, Hoang M Le, Nan Jiang, and Yisong Yue. Empirical study of off-policy policy evaluation for reinforcement learning. *arXiv preprint arXiv:1911.06854*, 2019.

[287] George A Vouros. Explainable deep reinforcement learning: state of the art and challenges. *ACM Computing Surveys*, 55(5):1–39, 2022.

[288] Stephan Wäldchen, Sebastian Pokutta, and Felix Huber. Training characteristic functions with reinforcement learning: Xai-methods play connect four. In *International Conference on Machine Learning*, pages 22457–22474. PMLR, 2022.

[289] Sebastian Wallkötter, Silvia Tulli, Ginevra Castellano, Ana Paiva, and Mohamed Chetouani. Explainable embodied agents through social cues: a review. *ACM Transactions on Human-Robot Interaction (THRI)*, 10(3):1–24, 2021.

[290] Jianhong Wang, Yuan Zhang, Yunjie Gu, and Tae-Kyun Kim. Shaq: Incorporating shapley value theory into multi-agent q-learning. *Advances in Neural Information Processing Systems*, 35:5941–5954, 2022.

[291] Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. Shapley q-value: A local reward approach to solve global reward games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7285–7292, 2020.

[292] Jingyuan Wang, Yang Zhang, Ke Tang, Junjie Wu, and Zhang Xiong. Alphastock: A buying-winners-and-selling-losers investment strategy using interpretable deep reinforcement attention networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1900–1908, 2019.

[293] Junpeng Wang, Liang Gou, Han-Wei Shen, and Hao Yang. Dqnviz: A visual analytics approach to understand deep q-networks. *IEEE transactions on visualization and computer graphics*, 25(1):288–298, 2018.

[294] Ning Wang, David V Pynadath, and Susan G Hill. The impact of pomdp-generated explanations on trust and performance in human-robot teams. In *Proceedings of the 2016 international conference on autonomous agents & multiagent systems*, pages 997–1005, 2016.

[295] Ning Wang, David V Pynadath, and Susan G Hill. Trust calibration within a human-robot team: Comparing automatically generated explanations. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 109–116. IEEE, 2016.

[296] Xiang Wang, Yingxin Wu, An Zhang, Fuli Feng, Xiangnan He, and Tat-Seng Chua. Reinforced causal explainer for graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):2297–2309, 2023.

[297] Xiting Wang, Yiru Chen, Jie Yang, Le Wu, Zhengtao Wu, and Xing Xie. A reinforcement learning framework for explainable recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 587–596. IEEE, 2018.

[298] Zihao Wang, Zhiqiang Xie, Enmei Tu, Alex Zhong, Yingying Liu, Jichang Ding, and Jie Yang. Reinforcement learning-based insulin injection time and dosages optimization. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[299] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pages 1995–2003. PMLR, 2016.

[300] Lindsay Wells and Tomasz Bednarz. Explainable ai and reinforcement learning—a systematic review of current approaches and trends. *Frontiers in artificial intelligence*, 4:550030, 2021.

[301] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, pages 5–32, 1992.

[302] Eyal Winter. The shapley value. *Handbook of game theory with economic applications*, 3:2025–2054, 2002.

[303] Bin Wu and Sean He. Self-learning and explainable deep learning network toward the security of artificial intelligence of things. *The Journal of Supercomputing*, 79(4):4436–4467, 2023.

[304] Haiping Wu, Khimya Khetarpal, and Doina Precup. Self-supervised attention-aware reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10311–10319, 2021.

[305] Jie Wu, Guanbin Li, Si Liu, and Liang Lin. Tree-structured policy based progressive reinforcement learning for temporally language grounding in video. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12386–12393, 2020.

[306] Jinwei Xing, Takashi Nagata, Xinyun Zou, Emre Neftci, and Jeffrey L Krichmar. Achieving efficient interpretability of reinforcement learning via policy distillation and selective input gradient regularization. *Neural Networks*, 161:228–241, 2023.

[307] Feiyu Xu, Hans Uszkoreit, Yangzhou Du, Wei Fan, Dongyan Zhao, and Jun Zhu. Explainable ai: A brief survey on history, research areas, approaches and challenges. In *Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II 8*, pages 563–574. Springer, 2019.

[308] Christopher C Yang. Explainable artificial intelligence for predictive modeling in healthcare. *Journal of healthcare informatics research*, 6(2):228–239, 2022.

[309] Zhao Yang, Song Bai, Li Zhang, and Philip HS Torr. Learn to interpret atari agents. *arXiv preprint arXiv:1812.11276*, 2018.

[310] Adrienne Yapo and Joseph Weiss. Ethical implications of bias in machine learning. 2018.

[311] Herman Yau, Chris Russell, and Simon Hadfield. What did you think would happen? explaining agent behaviour through intended outcomes. *Advances in Neural Information Processing Systems*, 33:18375–18386, 2020.

[312] Kok-Lim Alvin Yau, Yung-Wey Chong, Xiumei Fan, Celimuge Wu, Yasir Saleem, and Phei-Ching Lim. Reinforcement learning models and algorithms for diabetes management. *IEEE Access*, 11:28391–28415, 2023.

[313] Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. Reinforcement learning in healthcare: A survey. *ACM Computing Surveys (CSUR)*, 55(1):1–36, 2021.

[314] Chao Yu, Jiming Liu, and Hongyi Zhao. Inverse reinforcement learning for intelligent mechanical ventilation and sedative dosing in intensive care units. *BMC medical informatics and decision making*, 19(Suppl 2):57, 2019.

[315] Kun-Hsing Yu, Andrew L Beam, and Isaac S Kohane. Artificial intelligence in healthcare. *Nature biomedical engineering*, 2(10):719–731, 2018.

[316] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

[317] Tom Zahavy, Nir Ben-Zrihem, and Shie Mannor. Graying the black box: Understanding dqns. In *International conference on machine learning*, pages 1899–1908. PMLR, 2016.

[318] Amber E Zelvelder, Marcus Westberg, and Kary Främling. Assessing explainability in reinforcement learning. In *Explainable and Transparent AI and Multi-Agent Systems: Third International Workshop, EXTRAAMAS 2021, Virtual Event, May 3–7, 2021, Revised Selected Papers 3*, pages 223–240. Springer, 2021.

[319] Yan Zeng, Ruichu Cai, Fuchun Sun, Libo Huang, and Zhifeng Hao. A survey on causal reinforcement learning. *arXiv preprint arXiv:2302.05209*, 2023.

[320] Junzhe Zhang. Designing optimal dynamic treatment regimes: A causal reinforcement learning approach. In *International conference on machine learning*, pages 11012–11022. PMLR, 2020.

[321] Ke Zhang, Jun Zhang, Peidong Xu, Tianlu Gao, and Wenzhong Gao. A multi-hierarchical interpretable method for drl-based dispatching control in power systems. *International Journal of Electrical Power & Energy Systems*, 152:109240, 2023.

[322] Kristine Zhang, Henry Wang, Jianzhun Du, Brian Chu, Aldo Robles Arévalo, Ryan Kindle, Leo Anthony Celi, and Finale Doshi-Velez. An interpretable rl framework for pre-deployment modeling in icu hypotension management. *NPJ Digital Medicine*, 5(1):173, 2022.

[323] Guangxiang Zhu, Jianhao Wang, Zhizhou Ren, Zichuan Lin, and Chongjie Zhang. Object-oriented dynamics learning through multi-level abstraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6989–6998, 2020.

[324] He Zhu, Zikang Xiong, Stephen Magill, and Suresh Jagannathan. An inductive synthesis framework for verifiable reinforcement learning. In *Proceedings of the 40th ACM SIGPLAN conference on programming language design and implementation*, pages 686–701, 2019.

[325] Taiyu Zhu, Kezhi Li, Pau Herrero, and Pantelis Georgiou. Basal glucose control in type 1 diabetes using deep reinforcement learning: An in silico validation. *IEEE Journal of Biomedical and Health Informatics*, 25(4):1223–1232, 2020.

[326] Taiyu Zhu, Kezhi Li, Pau Herrero, and Pantelis Georgiou. Deep learning for diabetes: a systematic review. *IEEE Journal of Biomedical and Health Informatics*, 25(7):2744–2757, 2020.

[327] Yue-ting Zhuang, Fei Wu, Chun Chen, and Yun-he Pan. Challenges and opportunities: from big data to knowledge in ai 2.0. *Frontiers of Information Technology & Electronic Engineering*, 18:3–14, 2017.

[328] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.