Attribution-based Explanations for Markov Decision Processes

Paul Kobialka[1], **Andrea Pferscher**[1], **Francesco Leofante**[2], **Erika Ábrahám**[3], **Silvia Lizeth Tapia Tarifa**[1] and **Einar Broch Johnsen**[1]
[1]University of Oslo, Norway
[2]Imperial College London, United Kingdom
[3]RWTH Aachen University, Germany
{paulkob, sltarifa, einarj}@ifi.uio.no, f.leofante@imperial.ac.uk,
abraham@cs.rwth-aachen.de

Abstract

Attribution techniques explain the outcome of an AI model by assigning a numerical score to its inputs. So far, these techniques have mainly focused on attributing importance to static input features at a single point in time, and thus fail to generalize to sequential decision-making settings. This paper fills this gap by introducing techniques to generate attribution-based explanations for Markov Decision Processes (MDPs). We give a formal characterization of what attributions should represent in MDPs, focusing on explanations that assign importance scores to both individual states and execution paths. We show how importance scores can be computed by leveraging techniques for strategy synthesis, enabling the efficient computation of these scores despite the non-determinism inherent in an MDP. We evaluate our approach on five case-studies, demonstrating its utility in providing interpretable insights into the logic of sequential decision-making agents.

1 Introduction

Let us consider a loan application process, in which clients applying for a loan interact with a bank to discuss their eligibility and receive quotes on possible loans. While established techniques from Process Mining and Machine Learning can be used to automate process monitoring and decision-making (see, e.g., [Teinemaa *et al.*, 2019; Leo *et al.*, 2019; Shi *et al.*, 2022; Montevechi *et al.*, 2024]), the overall procedure remains inherently semi-structured. A successful application involves a sequence of interdependent steps, e.g., consultations and document submissions. Each step influences the probability of a successful outcome. Because the impact of these steps is often uncertain and state-dependent, the process can be viewed as a series of state transitions guided by both agent decisions and external variables.

Such stochastic, sequential decision-making tasks can be formally represented using Markov Decision Processes (MDPs) (e.g., [Baier and Katoen, 2008]). The application is then modeled as a system where transitions between process steps depend on the current state and specific actions, capturing the probabilistic nature of the problem. While MDPs effectively capture possible decisions, they lack native explainability mechanisms to communicate the significance of specific states (or sequences thereof) to a human observer.

Attribution-based explanations offer a potential solution to this problem by assigning importance scores to process elements based on their contribution to the final outcome. For a bank consultant or an applicant, such methods could provide intuitive answers to questions such as: "*Which specific steps in this application process were most critical in securing the loan approval?*". Unfortunately, existing attribution techniques are primarily designed for static, one-step prediction tasks [Bodria *et al.*, 2023; Molnar, 2020] and fail to transfer to complex sequential decision-making settings.

Contributions. To bridge this gap, we introduce a novel framework for attribution-based explanations specifically designed for decision-making in MDPs. Unlike existing methods, our approach accounts for the long-term impact of transitions by evaluating the importance of a state relative to a specific strategy (or the set of all possible strategies) for reaching a goal state. Specifically, we define an attribution measure that quantifies state importance based on the reachability probabilities of optimal strategies, ensuring that explanations reflect the most efficient paths to reach a goal. We further extend this notion from individual states to execution paths and propose two optimization approaches to efficiently compute importance scores. Finally, we evaluate our approach on five case studies, demonstrating its feasibility on complex sequential decision-making tasks modeled by MDPs, with thousands of states and tens of thousands of transitions.

Outline. Section 2 reviews related work on attribution-based explanations and Sect. 3 presents background on MDPs, optimization and attribution-based explanations in Explainable AI. Our attribution-based explanations for MDPs are formalized in Sect. 4, and our encoding for computing them is presented in Sect. 5. We evaluate the performance and stability of our encoding in Sect. 6 and conclude in Sect. 7.

2 Related Work

In the context of attribution-based explanations, Shapley values [Shapley, 1953] have emerged as a dominant theoretical framework. Originally developed to determine fair contributions in cooperative games, they have been widely adapted

to machine learning to quantify the impact of input features on model outputs [Lundberg and Lee, 2017]. Closer to our work, [Baier *et al.*, 2021] leveraged Shapley values to propose a notion of attribution that captures entire paths in parametric Markov Chains and, later, extended this notion to general transition systems by transforming them into "one-shot" cooperative games [Baier *et al.*, 2025]. Finally, [Triantafyllou *et al.*, 2021] proposed several attribution techniques, including the Shapley value and Banzhaf index [Banzhaf, 1965], to assign blame to actors in a sequential cooperative multi-agent setting. These works, however, either do not deal with MDPs, or abstract from the MDP structure and do not leverage reasoning about strategies to determine importance scores.

This paper approaches the problem from a different angle and exploits the structure of MDPs to provide explanations for sequential decision making in MDPs. To this aim, we devise novel methods to compute attribution scores based on strategy synthesis. While strategies have been used to provide explanations for MDPs [Kobialka *et al.*, 2025], their work targeted counterfactual explanations — not attribution-based explanations — and is as such complementary to this work.

3 Preliminaries

For a finite set X, let $\mathit{Distr}(X)$ be the set of all *distributions over* X, which are functions $\mu : X \to [0,1]$ with $\sum_{x\in X} \mu(x)=1$; we set $\mathit{supp}(\mu) = \{x \in X \mid \mu(x)>0\}$.

3.1 Markov Decision Processes

A *Markov Decision Process* (MDP) $\mathcal{M}$ is a tuple $\langle S, A, s_0, \delta\rangle$ with a finite non-empty set S of states, a finite set A of actions, an initial state $s_0 \in S$, and a partial transition function $\delta : S \times A \rightharpoonup \mathit{Distr}(S)$. An action $a \in A$ is *enabled* in state $s \in S$, if $\delta(s,a)$ is defined. Let $A(s)$ be the set of enabled actions in state s; we require that $A(s) \neq \emptyset$ for all $s \in S$. Transition probabilities $\delta(s,a)(s')$ are abbreviated $\delta(s,a,s')$.

Let $\mathcal{M} = \langle S, A, s_0, \delta\rangle$ be an MDP. An *infinite path* is an infinite sequence $\tau = s_0, a_0, s_1, \ldots$ of alternating states $s_i \in S$ and actions $a_i \in A(s_i)$ with $\delta(s_i, a_i, s_{i+1}) > 0$ for all $i \geq 0$. A *finite path* is a non-empty finite prefix $\tau = s_0, \ldots, s_n$ of an infinite path; τ is *simple* if $s_i \neq s_j$ for all $0 \leq i < j \leq n$, and its *cylinder set* $\mathit{Cyl}(\tau)$ consists of all infinite paths extending τ. Let $\mathit{Paths}_{\mathcal{M}}(s)$ and $\mathit{Paths}^{\mathit{fin}}_{\mathcal{M}}(s)$ be the sets of all infinite resp. finite paths of $\mathcal{M}$ starting in $s \in S$. A state $t \in S$ is *reachable* from $s \in S$, if a finite path starts in s and ends in t; let $\overleftarrow{R}(t)$ be the set of all states from which t is reachable.

Strategies (or *schedulers*) resolve nondeterminism in MDPs by deciding which actions to take. We consider *stochastic memoryless* strategies $\sigma : S \to \mathit{Distr}(A)$ with $\mathit{supp}(\sigma(s)) \subseteq A(s)$ for all $s \in S$, forming the set $\Sigma_{\mathcal{M}}$. We abbreviate strategy probabilities $\sigma(s)(a)$ as $\sigma(s,a)$, and call σ *deterministic* if $|\mathit{supp}(\sigma(s))| = 1$ for any $s \in S$. *Finite memory* strategies extend memoryless ones with a finite set M of memory modes, differentiating how the current state is reached. Finite memory strategies for an MDP with states S can be simulated by memoryless strategies on an MDP with state space $S \times M$. Therefore, we use the memoryless notation also for finite memory strategies.

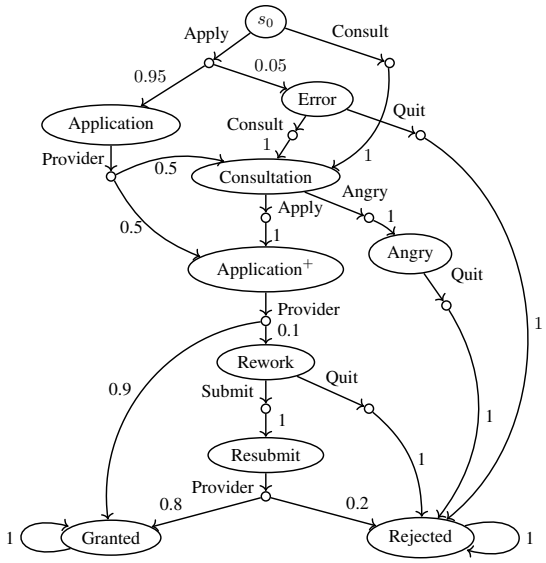


Figure 1: An MDP model of a loan application process. The bank can only perform the Provider action, which is purely stochastic; thus, the applicant has full control over non-deterministic choices.

For a strategy σ of $\mathcal{M} = \langle S, A, s_0, \delta\rangle$, its *induced Markov Chain* is $\mathcal{M}^\sigma = \langle S, s_0, \delta^\sigma\rangle$, where $\delta^\sigma \colon S \times S \to [0,1]$ with $\delta^\sigma(s,s') = \sum_{a\in A} \sigma(s)(a) \cdot \delta(s,a,s')$ for all $s, s' \in S$. With $\mathcal{D} = \mathcal{M}^\sigma$ we associate the *probability space* $(\mathit{Paths}_{\mathcal{D}}(s_0), \{\bigcup_{\tau\in R} \mathit{Cyl}(\tau) \mid R \subseteq \mathit{Paths}^{\mathit{fin}}_{\mathcal{D}}(s_0)\}, \Pr_{\mathcal{D}}(s_0))$, where the probability of the cylinder set of a finite path $\tau = s_0 \ldots s_n$ is $\Pr_{\mathcal{D}}(s_0)(\mathit{Cyl}(\tau)) = \prod_{i=0}^{n-1} \delta^\sigma(s_i, s_{i+1})$. For a state t, let $\Sigma^t_{\mathcal{M}}$ be the set of all strategies $\sigma \in \Sigma_{\mathcal{M}}$ with a positive probability of reaching t in $\mathcal{M}^\sigma$.

Example 1. *The MDP in Figure 1 models a loan application process. Starting in s_0, the applicant decides between taking a consultation or directly submitting an application. If applying directly, the application is error-free with a chance of 95%, and the bank decides whether a consultation is needed or the application can be considered directly. If an error occurred when applying, the Error state is reached and a consultation is needed. In the consultation, the user either applies again, or they become angry and quit the application process. In the former case, the submitted application is evaluated by the bank, and may be reworked one more time before a final decision is made.*

3.2 Non-linear Optimization

Mixed Integer Quadratically Constrained Quadratic Problems (MIQCQPs) [Billionnet *et al.*, 2016] are non-linear optimization problems, where both objective function and constraints are quadratic functions over real and integer variables. Formally, MIQCQPs are defined as:

$$\begin{array}{lll} \min & f_0(x) & \\ \text{subject to} & f_i(x) \leq b_i & \textit{for } i = 1, \ldots, m, \\ & 0 \leq x_j \leq u_j & \textit{for } x_j \in V_{\mathbb{Z}} \cup V_{\mathbb{R}}, \\ & x_j \in \mathbb{Z} & \textit{for } x_j \in V_{\mathbb{Z}}, \\ & x_j \in \mathbb{R} & \textit{for } x_j \in V_{\mathbb{R}}, \end{array}$$

using integer-valued ($V_{\mathbb{Z}}$) and real-valued ($V_{\mathbb{R}}$) variables $x = (x_1, \ldots, x_n)$ ($n \in \mathbb{N}$) in the functions $f_i(x) = x^T Q_i x + c_i^T x$ with symmetric matrices $Q_i \in \mathbb{R}^{n \times n}$ and constants $c_i \in \mathbb{R}^n$ for all $i \in \{0, \ldots, m\}$ ($m \in \mathbb{N}$). Upper bounds align with the domain of the constrained variable, i.e., $u_j \in \mathbb{Z}$ for $x_j \in V_{\mathbb{Z}}$, and $u_j \in \mathbb{R}$ for $x_j \in V_{\mathbb{R}}$. MIQCQP are generally non-convex and thus hard to solve [Billionnet *et al.*, 2016; Garey and Johnson, 1979].

We consider *hierarchical* objectives [Anandalingam and Friesz, 1992], such that the objective function $f_0(x)$ maps to a tuple and higher-indexed objectives are optimized under optimally-solved lower-indexed objectives, i.e., $f_0(x)_1$ is optimized with the additional constraint that $f_0(x)_0$ is optimal.

3.3 Attribution-based Explanations

An attribution-based explanation explains the outcome of an AI model by assigning a numerical *score* to its inputs. Intuitively, these *attribution scores* represent the degree to which each input influences the observed result. In Machine Learning, attribution methods such as LIME, SHAP and gradient-scoring [Ribeiro *et al.*, 2016; Lundberg and Lee, 2017; Selvaraju *et al.*, 2017; Sundararajan and Najmi, 2020] assign a score to each input feature (e.g., pixels, words) to reflect their contribution towards an output (e.g., a classification); see also the survey [Molnar, 2020]. Although well-suited for ML, these methods are generally not applicable for MDPs because they attribute importance to static input features at a single point in time, failing to account for crucial factors in sequential decision-making, such as sequential dependencies, long-term rewards, and stochastic state transitions.

4 Attribution-based Explanations for MDPs

As a first contribution, we give a formal characterization of attributions for MDPs, establishing a quantitative basis for measuring the importance of states and paths, such that questions about their influence in reaching a target can be answered.

LTL notation. In the sequel, we will use notation from Linear Temporal Logic (LTL) [Pnueli, 1977], which we briefly recall. Based on a set of atomic propositions, e.g., a set S of state names, LTL formulae φ are inductively defined by the abstract grammar $\varphi ::= \top \mid s \mid \varphi \wedge \varphi \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi \,\mathbf{U}\, \varphi$ with $s \in S$. The formula $\bigcirc\varphi$ expresses that φ holds in the *next* state, $\varphi_1 \,\mathbf{U}\, \varphi_2$ that φ_1 holds *until* φ_2 gets true, and $\Diamond\varphi$ that φ will *eventually* hold (i.e., $\top \,\mathbf{U}\, \varphi$). For an MDP $\mathcal{M}$, a strategy $\sigma \in \Sigma_{\mathcal{M}}$, and an LTL formula φ, $Pr_{\mathcal{M}^\sigma}(\varphi) = Pr_{\mathcal{M}^\sigma}\{\tau \in \mathit{Paths}_{\mathcal{M}}(s_0) \mid \tau \models \varphi\}$ [Baier and Katoen, 2008].

4.1 Our formal attribution framework

As our first contribution, in order to provide a comprehensive view of importance, we formalize four attribution problems. We first address the importance of individual states (**P1**) and paths (**P3**) under a fixed strategy, capturing the realization of a specific behavior. We then generalize these notions to quantify the importance of states (**P2**) and paths (**P4**) across all strategies, revealing the global importance of these elements within the MDP. We begin with quantifying the importance of an individual state within the context of a fixed strategy.

P1

Consider an MDP $\mathcal{M} = \langle S, A, s_0, \delta \rangle$, a target state $t \in S$, and a strategy $\sigma \in \Sigma^t_{\mathcal{M}}$. *How important is a given state $s \in S$ for reaching t in $\mathcal{M}^\sigma$?*

Intuitively, a state $s \in S$ is *important* under strategy σ for reaching state $t \in S$, if the induced Markov Chain $\mathcal{M}^\sigma$ has a high probability of visiting s before reaching t. We denote this probability by $\Pr_{\mathcal{M}^\sigma}(s_0, s, t) \coloneqq \Pr_{\mathcal{M}^\sigma}(\Diamond t \wedge (\neg t \,\mathbf{U}\, s))$, to capture the probability of eventually reaching t and visiting s before reaching t. If $\mathcal{M}$ is loop-free and s is never reached after t on any simple path, then $P_{\mathcal{M}^\sigma}(s_0, s, t)$ reduces to $P_{\mathcal{M}^\sigma}(\Diamond t \wedge \Diamond s)$.

Given the above, we define the importance of a state s for reaching t in MDP $\mathcal{M}$ under strategy σ as:

$$\mathrm{imp}^{s_0,t}_{\mathcal{M}^\sigma}(s) \coloneqq \frac{\Pr_{\mathcal{M}^\sigma}(s_0, s, t)}{\Pr_{\mathcal{M}^\sigma}(\Diamond t)}. \tag{1}$$

To ensure that our importance measure is well-defined, we require that $\sigma \in \Sigma^t_{\mathcal{M}}$ (i.e., the probability of reaching t under σ is positive). Note that $\mathrm{imp}^{s_0,t}_{\mathcal{M}^\sigma}(s) \in [0, 1]$ for any $s \in S$ and $\sigma \in \Sigma^t_{\mathcal{M}}$; $\mathrm{imp}^{s_0,t}_{\mathcal{M}^\sigma}(s) = 0$ indicates that s is never visited before reaching t, and $\mathrm{imp}^{s_0,t}_{\mathcal{M}^\sigma}(s) = 1$ indicates that s is always visited before reaching t. Thus the initial state s_0 and the target state t receive an importance of 1 under all strategies, as they are guaranteed to be visited. However, the importance of the other states depends on the strategy at hand.

P2

Consider an MDP $\mathcal{M} = \langle S, A, s_0, \delta \rangle$ and a target state t. *How important is a given state $s \in S$ for reaching t?*

In contrast to **P1**, no strategy is assumed in **P2**. Here, the importance of a state $s \in S$ depends on whether s is important for reaching t globally. Intuitively, if s is never visited before reaching t under any strategy, then s is not important. In fact, visiting s might even be detrimental for reaching t, and thus receive an importance score of 0. As strategies can maximize and minimize the probability of visiting s before reaching t, we express importance for **P2** as a tuple:

$$\mathrm{imp}^{s_0,t}_{\mathcal{M}}(s) \coloneqq \big(\min_{\sigma \in \Sigma^t_{\mathcal{M}}} \mathrm{imp}^{s_0,t}_{\mathcal{M}^\sigma}(s), \max_{\sigma \in \Sigma^t_{\mathcal{M}}} \mathrm{imp}^{s_0,t}_{\mathcal{M}^\sigma}(s) \big).$$

Obviously, importance scores between the lower and upper bound can be realized by mixing the minimizing and maximizing strategies. However, strategies optimizing $\mathrm{imp}^{s_0,t}_{\mathcal{M}^\sigma}(s)$ might require up to one bit of memory to determine whether s was visited before reaching t. If lower and upper bounds align, the probability to visit s before t equals for all $\sigma \in \Sigma^t_{\mathcal{M}}$.

In **P1**, the notion of importance normalizes the probability of visiting s before t by dividing by the total probability of reaching t. In addition, we also define *absolute importance* without normalization as the tuple $\big(\min_{\sigma \in \Sigma^t_{\mathcal{M}}} \Pr_{\mathcal{M}^\sigma}(s_0, s, t), \max_{\sigma \in \Sigma^t_{\mathcal{M}}} \Pr_{\mathcal{M}^\sigma}(s_0, s, t)\big)$ of minimal and maximal probabilities of reaching t via s.

Available model checking tools apply *value iteration* to monotonically decrease resp. increase the probability of reaching t by local strategy modifications, adapting decisions at individual states. However, the importance of a state is defined as a fraction of *two probabilities* defined by the *same*

strategy, and as such it cannot be directly computed by available tools. Intuitively, value iteration can be applied to both the nominator and the denominator in isolation, but monotonicity cannot be assured for their fraction. See the extended version [Kobialka *et al.*, 2026] for an illustrating example.

We now extend the importance of states for reaching a state t, to the importance of paths, where several states need to be visited before reaching t.

P3

Consider an MDP $\mathcal{M} = \langle S, A, s_0, \delta \rangle$, a target state t, a simple path $\tau = \langle s_0, a_0, \dots, s_n \rangle$ with $s_i \neq t$ for $0 \leq i \leq n$ and $s_n \in \overleftarrow{R}(t)$, and a strategy σ. *How important is τ for reaching t under σ?*

Intuitively, a path τ is important for reaching t under strategy σ, if the induced Markov Chain $\mathcal{M}^\sigma$ is likely to follow τ before reaching t.

For a path τ of the above form, we express the probability of reaching t after following τ as $\Pr_{\mathcal{M}^\sigma}(\tau, t) \coloneqq \Pr_{\mathcal{M}^\sigma}(\Diamond t \wedge (\neg t \,\mathbf{U} \bigcirc(s_1 \wedge (\cdots \wedge (\bigcirc s_n) \dots))))$, which simplifies to $\Pr_{\mathcal{M}^\sigma}(\Diamond t \wedge \bigcirc(s_1 \wedge (\cdots \wedge (\bigcirc s_n) \dots)))$.

The definition of the importance of a state in Equation (1) is extended to paths as follows. For a strategy σ of the MDP $\mathcal{M}$, the importance of τ for reaching t is defined as:

$$\mathrm{imp}^{s_0,t}_{\mathcal{M}^\sigma}(\tau) \coloneqq \frac{\Pr_{\mathcal{M}^\sigma}(\tau, t)}{\Pr_{\mathcal{M}^\sigma}(\Diamond t)}.$$

Though we could allow paths visiting t or with $s_n \notin \overleftarrow{R}(t)$, the importance score would then default to 0. Therefore, we constrain τ not to visit t and end in some $s_n \in \overleftarrow{R}(t)$.

In the absence of a fixed strategy, the importance of path τ is lower and upper bounded, expressing the global importance of τ for reaching t.

P4

Consider an MDP $\mathcal{M} = \langle S, A, s_0, \delta \rangle$, a target state t, and a simple path $\tau = \langle s_0, a_0, \dots, s_n \rangle$, with $s_i \neq t$ for $0 \leq i \leq n$ and $s_n \in \overleftarrow{R}(t)$. *How important is τ for reaching t?*

As **P4** abstracts from the given strategy in **P3**, we define the importance of a path as a tuple of lower and upper bounds on the importance of τ for reaching t under any strategy:

$$\mathrm{imp}^{s_0,t}_{\mathcal{M}}(\tau) \coloneqq \big(\min_{\sigma \in \Sigma^\tau_{\mathcal{M}}} \mathrm{imp}^{s_0,t}_{\mathcal{M}^\sigma}(\tau), \max_{\sigma \in \Sigma^\tau_{\mathcal{M}}} \mathrm{imp}^{s_0,t}_{\mathcal{M}^\sigma}(\tau) \big),$$

where the set $\Sigma^\tau_{\mathcal{M}}$ of strategies from $\Sigma^t_{\mathcal{M}}$ follows the action choices in τ, i.e., $\Sigma^\tau_{\mathcal{M}} = \{\sigma \in \Sigma^t_{\mathcal{M}} \mid \sigma(s_i) = a_i \text{ for } 0 \leq i < n\}$. We restrict the strategies over which the importance of a path is optimized to $\Sigma^\tau_{\mathcal{M}}$, as the lower bound is trivially realizable by not following τ.

We conclude this section with Example 2 to illustrate our definitions on the loan application from Example 1.

Example 2. *A bank employee restructuring the loan application process from Example 1 wants to determine the importance of the consultation state for a successful application. Our attribution framework would show that the importance of Application$^+$ is lower and upper bounded by one, as every successful application must visit that state,* $\mathrm{imp}^{s_0,\mathit{Granted}}_{\mathcal{M}}(\mathit{Application}^+) = (1, 1)$. *The importance of Angry, which is detrimental for a successful application, is lower and upper bounded by* 0*,* $\mathrm{imp}^{s_0,\mathit{Granted}}_{\mathcal{M}}(\mathit{Angry}) = (0, 0)$. *The importance of the path $\tau = \langle s_0, \mathit{Apply}, \mathit{Application} \rangle$ is lower bounded by* 0.9*, and upper bounded by* 1*:* $\mathrm{imp}^{s_0,\mathit{Granted}}_{\mathcal{M}}(\tau) = (0.9, 1)$.

Such importance bounds provide a formal yet interpretable measure of how specific states and paths contribute to – or hinder – the reachability of a target. In the following section, we detail our novel approach used to compute these intervals.

5 Computing Importance

After defining importance, in this section we propose a framework to compute its value. While our method is applicable to both states and paths, the following derivations focus on states, and we discuss the generalization to paths at the end of this section. We introduce three encodings: (i) QP, a non-linear encoding optimizing importance over all strategies, (ii) QP*, a non-linear encoding optimizing over reachability-optimal strategies, and (iii) LP*, a linear encoding optimizing importance over reachability-optimal strategies. Theoretical results connect the encodings: Lemma 1 establishes that QP is computationally difficult and Theorem 1 proves its correctness; Lemma 3 enables the cheaper linear encoding LP*.

Assume an MDP $\mathcal{M}^* = \langle S^*, A, s_0, \delta^* \rangle$ and let $\hat{s} \in S^* \setminus \{s_0\}$ be the state of interest for reaching state $t \in S^*$. Strategies optimizing $\mathrm{imp}^{s_0,t}_{\mathcal{M}^*}(\hat{s})$ require 1 bit of memory to remember whether $\hat{s}$ was visited. An initial preprocessing step introduces this memory, analogously to the memory construction in LTL model checking [Baier and Katoen, 2008].

Preprocessing. We construct the MDP $\mathcal{M} = \langle S, A, s_0^\perp, \delta \rangle$ with two copies $s^\top$ and $s^\perp$ for each state $s \in S^*$: $S = \{s^\top, s^\perp \mid s \in S^*\}$. Index $\top$ indicates that $\hat{s}$ was visited and $\perp$ that $\hat{s}$ was not visited. Accordingly, $\delta(s_1^{\bowtie_1}, a, s_2^{\bowtie_2})$ is $\delta^*(s_1, a, s_2)$ if $s_2^{\bowtie_2} = \hat{s}^\top$ or ($\bowtie_1 = \bowtie_2$ and $s_2 \neq \hat{s}$), and 0 otherwise, for all $s_1, s_2 \in S^*$, $a \in A$, and $\bowtie_1, \bowtie_2 \in \{\top, \perp\}$.

Encoding as Quadratic Program (QP). Based on the preprocessed MDP $\mathcal{M} = \langle S, A, s_0, \delta \rangle$, Fig. 2 shows the encoding of the lower bound $\min_{\sigma \in \Sigma^t_{\mathcal{M}^*}} \mathrm{imp}^{s_0,t}_{\mathcal{M}^{*\sigma}}(\hat{s})$ on the importance of $\hat{s}$ as the solution to a nonlinear optimization problem. QP uses three types of variables: (1) $p_{sa} \in [0, 1]$ stores the probability of choosing action $a \in A$ in state $s \in S$ by the computed strategy, (2) $p_{s,t^\bowtie} \in [0, 1]$ stores the probability of reaching state $t^\top$ resp. $t^\perp$ under the computed strategy, and (3) $\tau_s \in \mathbb{R}$ allows to encode weight-decreasing paths to terminal states to enforce smallest fixedpoints. Note that $p_{s_0^\perp, t^\perp} + p_{s_0^\perp, t^\top}$ is the probability of reaching t from s_0 in the non-preprocessed MDP, as all paths either visit or bypass $\hat{s}$.

Eq. (2) minimizes an objective function that encodes the importance of $\hat{s}$; maximizing corresponds to minimizing the objective multiplied by -1. Constraint 3 ensures that the computed strategy is well-defined (i.e., the probabilities sum up to 1). Constraints 4–5 encode default values for reaching state t, and Constraint 6 ensures a positive probability of reaching t for an arbitrarily small ϵ. Constraint 7 encodes the

$$\min \frac{p_{s_0^\perp,t^\top}}{p_{s_0^\perp,t^\perp} + p_{s_0^\perp,t^\top}} \quad (2)$$

subject to :

$$\forall s \in S_1 := S \setminus \{t^\perp, t^\top\} : \quad \sum_{a \in A(s)} p_{sa} = 1 \quad (3)$$

$$p_{t^\perp,t^\perp} = p_{t^\top,t^\top} = 1 \quad (4)$$

$$p_{t^\perp,t^\top} = p_{t^\top,t^\perp} = 0 \quad (5)$$

$$p_{s_0^\perp,t^\perp} + p_{s_0^\perp,t^\top} \geq \epsilon \quad (6)$$

$$\forall s \in S_1, \bowtie \in \{\top, \perp\} : p_{s,t^\bowtie} = \sum_{a \in A(s)} \sum_{s' \in S} p_{sa} \cdot \delta(s,a,s') \cdot p_{s',t^\bowtie} \quad (7)$$

$$\forall s \in S \setminus T : \tau_s + 1 \leq \sum_{a \in A(s)} \sum_{s' \in S} p_{sa} \cdot \delta(s,a,s') \cdot \tau_{s'} \quad (8)$$

Figure 2: Encoding QP for optimizing the importance of $\hat{s} \in S$ in the preprocessed MDP $\mathcal{M}$ as a nonlinear optimization problem.

Bellman optimality condition. Constraint 8 encodes a state ordering [Wimmer *et al.*, 2012] to ensure that bottom strongly connected components not containing t reach t with probability 0, where the set of terminal states $T \subset S$ contains all absorbing states where the only available action is a self-loop. Constraint 5 is not required for correctness but can reduce the solving time. Constraint 7 can be restricted to states reaching $t^\top$ or $t^\perp$, setting $p_{s,t^\bowtie}$ to 0 for states not reaching $t^\bowtie$. All values in p_{sa} can be transformed into a strategy *with memory* on the non-preprocessed MDP, introducing $\bowtie \in \{\top, \perp\}$ as a memory mode to remember whether $\hat{s}$ was visited.

Let P denote the optimization problem defined in Fig. 2. The Bellman-optimality encoding in Constraint 7 is non-convex, preventing polynomial-time solutions:

Lemma 1. *P is in general non-convex.*

Proof sketch. An optimization problem is convex, if its objective and all constraints are convex [Boyd and Vandenberghe, 2004]. As Constraint 7 is in general non-convex, see e.g. [Kobialka *et al.*, 2025], the theorem follows. □

Lemma 2. *Assume a solution ν to P, which maps each variable v to value $\nu(v)$. Further, let σ be the strategy defined by $\sigma(s,a) = \nu(p_{sa})$, for all $s \in S$ and $a \in A$. Then, the value of the objective function equals $\min_{\sigma \in \Sigma^t_{\mathcal{M}^*}} \mathrm{imp}^{s_0,t}_{\mathcal{M}^{*\sigma}}(\hat{s})$.*

The following theorem states that the encoding facefully captures the importance of $\hat{s}$.

Theorem 1 (Soundness and Completeness)**.** *P is feasible iff the importance of state $\hat{s}$ is well defined.*

The proofs for Lemma 2 and Theorem 1 are contained in the extended version [Kobialka *et al.*, 2026].

Importance for optimal reachability. In practice, we are typically interested in the importance of $\hat{s}$ under strategies that maximize the probability of reaching t. Then we can restrict the importance computation to strategies from $\Sigma^* = \{\sigma \in \Sigma_{\mathcal{M}^*} \mid P_{\mathcal{M}^{*\sigma}}(\Diamond t) \geq P_{\mathcal{M}^{*\sigma'}}(\Diamond t) \text{ for all } \sigma' \in \Sigma_{\mathcal{M}^*}\}$. This yields a constant value for the denominator in Eq. (2), and the problem reduces to minimizing $p_{s_0^\perp,t^\top}$.

When only considering optimal reachability, QP can be further simplified as reachability probabilities from strongly connected components can be lower-bounded directly. We extend the problem to multi-objective optimization, minimizing the sum of reachability probabilities (under the maximizing schedulers from Σ^*) before optimizing for importance:

$$\min \big(\big(\sum_{s \in S, \bowtie \in \{\top,\perp\}} p_{s,t^\bowtie} \big), \quad p_{s_0^\perp,t^\top} \big).$$

We further simplify computations by fixing the optimal reachability probability p^* and lower-bounding the reachability of $t^\top$ and $t^\perp$ over all possible actions to replace Constraint 8:

$$p_{s_0^\perp,t^\perp} + p_{s_0^\perp,t^\top} = p^*$$

$$\forall s \in S_1, a \in A : \sum_{\bowtie \in \{\top,\perp\}} p_{s,t^\bowtie} \geq \sum_{\substack{s' \in S, \\ \bowtie \in \{\top,\perp\}}} \delta(s,a,s') \cdot p_{s',t^\bowtie}$$

This encoding, denoted QP*, does not guarantee that the computed strategy is loop-free, but ensures that the assigned probability corresponds to actually realizable probabilities. The strategy can always be repaired to satisfy a state ordering.

We show that restricting strategies to reachability-optimal strategies simplifies the problem of computing importance:

Lemma 3. *For an MDP $\mathcal{M}$, target state t, state $\hat{s} \in S$ and simple path τ, there exists a deterministic strategy optimizing $\mathrm{imp}^{s_0,t}_{\mathcal{M}^\sigma}(\hat{s})$ and $\mathrm{imp}^{s_0,t}_{\mathcal{M}^\sigma}(\tau)$ over all strategies in Σ^*.*

Lemma 3 is proven in [Kobialka *et al.*, 2026]. It follows that QP* can be adapted to a mixed-integer linear program, denoted as LP*, without quadratic components, in which the variables p_{sa} are turned into binary variables (i.e., from $\{0,1\}$). The optimization objective only optimizes for the importance: $\min p_{s_0^\perp,t^\top}$. Constraints 7 and 8 are replaced by:

$$p_{s_0^\perp,t^\perp} + p_{s_0^\perp,t^\top} = p^* \quad (9)$$

$$\forall s \in S_1, a \in A(s), \bowtie \in \{\top, \perp\} : \quad (10)$$

$$p_{s,t^\bowtie} \leq \sum_{s' \in S} \delta(s,a,s') \cdot p_{s',t^\bowtie} + (1 - p_{sa})$$

$$\forall s \in S_1 \setminus T, a \in A(s) : \quad (11)$$

$$\tau_s + 1 \leq \sum_{s' \in S} \delta(s,a,s') \cdot \tau_{s'} + (M - M \cdot p_{sa})$$

Constraint 9 ensures that optimal reachability is realized. Constraint 10 adapts the Bellman optimality equations to discrete actions; only for the chosen action in each state is the constraint not trivially satisfied. To realize the optimal reachability, those constraints must be satisfied with equation. Constraint 11 ensures that a weight-decreasing path exists under the computed strategy, where M is a large constant.

Remark. All the presented encodings can be extended to compute the importance of a simple path $\tau = \langle s_0, a_0, \ldots, s_n \rangle$, with $s_i \neq t$. The importance of τ can be expressed as the importance of s_n weighted with the probability of observing τ. By partially assigning $p_{s_i a_i} = 1$ to maximize the probability of following τ and then solving for the importance of s_n, the importance of τ is the result multiplied with $\prod \delta^*(s_i, a_i, s_{i+1})$. The optimization problem is only well defined, and thus feasbile, for strategies reaching t with positive probability, i.e., it is ill defined if $s_n \notin \overleftarrow{R}(t)$.

6 Experimental Evaluation

We now show that our approach is able to compute importance scores efficiently across five case-studies. *GrepS* records a programming skill evaluation service where customers (1) sign-up, (2) solve programming tasks, and (3) release their test results [Kobialka *et al.*, 2024]. *BPIC12* [van Dongen, 2012] and *BPIC17* [van Dongen, 2017] are datasets for loan applications released by the IEEE Task Force on Process Mining.[1] *MSSD* is the *Music Streaming Sessions Dataset* [Brost *et al.*, 2019] released by Spotify (*Mini* version with $10\,000$ listening sessions). *Epidemic* [Kazemi *et al.*, 2025] is a synthetic benchmark on vaccination strategies.

We used passive automata learning [Mao *et al.*, 2016; Muskardin *et al.*, 2022] to learn MDPs from the traces, after applying standard preprocessing techniques. For MSSD, we construct models that vary in the number of traces; e.g., MSSD10 includes 10% of the traces. For Epidemic, we vary the maximal population size; e.g., E6 considers a maximum population size of 6. The number of states is significantly larger in the MSSD models than in the other models: MSSD10 contains on average 36 times more states than BPIC12 or BPIC17, and has a 70 times higher maximum of outgoing transitions. Epidemic has a similar number of states as MSSD, but the number of transitions is 6 times greater than that of MSSD. However, the models are less connected, the largest Epidemic model has a maximum of 48 outgoing transitions, and MSSD100 of over $5\,600$. Each model contains a unique target state t that represents success. For example, in BPIC12 and BPIC17, the loan is granted to the user in t.

For Greps, BPIC12, and BPIC17, we compute the importance of every state in the model. For the MSSD models, we sample 10 states per model. For the Epidemic models, we use the initial state proposed by [Kazemi *et al.*, 2025] to compute the importance of the number of available vaccinations for reaching a state without unvaccinated individuals.

All experiments were performed on a workstation with $128\,\mathrm{GB}$ memory and a 64-core AMD EPYC processor. Our implementation and results are online.[2] Python 3.10.12 is used to interface with the Gurobi[3] (v. 11.0.3) solver. Each run of the solver is restricted to $4\,\mathrm{GB}$ of memory and 1 hour, in all experiments we set $M = 10^{16}$ and $\epsilon = 10^{-4}$.

Numerical results. We compute strategies optimizing the importance of states from all models, demonstrating that attribution-based explanations can be computed for models with up to $10\,000$ of states and transitions. Additionally, we highlight differences between the three presented encodings.

Table 1 compares the averaged computation times and results for Greps, BPIC12, and BPIC17. For each model, we compute the lower and upper bounded importance for every state using all three encodings. Encoding (Enc.) denotes the used encoding: *QP* denotes the quadratic program optimizing over $\Sigma_{\mathcal{M}}$, *QP** the quadratic encoding optimizing over reachability optimal strategies $\Sigma^*_{\mathcal{M}}$, and *LP** the linear program. Opt. denotes the number of problems solved optimally, T.O. denotes the timeouts, and Σ the absolute number of problems. The remaining columns describe the mean, standard deviation, minimum, and maximum over all solution times in seconds. While QP* and LP* can solve all inputs without timeout, LP* solves all instances within one second.

Model	Enc.	Mean(t)	Std(t)	Min(t)	Max(t)	Opt.	T.O.	Σ
Greps	**QP**	0.06	0.06	0.00	0.64	128	0	128
	QP*	0.17	0.13	0.00	0.73	128	0	128
	LP*	0.03	0.03	0.00	0.19	128	0	128
BPIC12	**QP**	1616.91	1419.14	0.02	T.O.	192	66	258
	QP*	2.95	11.95	0.01	102.30	258	0	258
	LP*	0.02	0.02	0.00	0.11	258	0	258
BPIC17	**QP**	23.28	209.62	0.03	T.O.	315	1	316
	QP*	2.66	5.25	0.02	37.49	316	0	316
	LP*	0.07	0.05	0.01	0.77	316	0	316

Table 1: Computing importance for GrepS and BPIC: averaged runtimes in seconds

Model	Enc.	Mean(t)	Std(t)	Min(t)	Max(t)	Opt.	T.O.	Σ
E6	**QP***	0.65	0.78	0.01	2.52	26	0	26
	LP*	0.02	0.02	0.00	0.07	26	0	26
E7	**QP***	2.28	3.17	0.01	12.10	30	0	30
	LP*	0.01	0.00	0.00	0.01	30	0	30
E8	**QP***	648.31	1387.09	0.01	T.O.	28	6	34
	LP*	0.01	0.00	0.00	0.01	34	0	34
E9	**QP***	687.61	1403.24	0.02	T.O.	31	7	38
	LP*	0.01	0.00	0.00	0.01	38	0	38
E10	**QP***	1399.82	1751.83	0.02	T.O.	26	16	42
	LP*	0.01	0.00	0.01	0.02	42	0	42
E15	**QP***	1742.02	1813.74	0.08	T.O.	32	30	62
	LP*	0.04	0.02	0.01	0.08	62	0	62
E19	**QP***	1753.99	1811.01	0.16	T.O.	40	38	78
	LP*	0.05	0.03	0.02	0.11	78	0	78

Table 2: Computing importance for Epidemic: averaged runtimes in seconds.

Table 2 shows runtimes for Epidemic with population sizes up to 19. For population sizes larger than 19, the transition probabilities can become very small, i.e., they require more than 10 digits of precision to avoid rounding to 0. Based on the results from BPIC12 and BPIC17, we omitted encoding QP as the Epidemic models are significantly larger. Until E8, encoding QP* can be solved within the one-hour timeout, encoding LP* can be solved within fractions of a second for all models. Comparing E7 with E8 reveals the growing complexity induced by the population size, the number of states increase from 962 to 1379, and transitions from 5644 to 9029. All timed-out instances involved states with sufficient vaccinations for reaching t, and were well connected in the MDP.

After confirming that computing strategies to minimize, respectively maximize, importance takes comparably long, we restricted the MSSD experiments to minimization. Figure 3 presents the averaged runtime results for the MSSD models on the left-hand y-axis and the number of time-outs on the right-hand y-axis. Most strategies were computed within a one-hour time constraint; the $4\,\mathrm{GB}$ memory constraint was reached in three MSSD10 problems. While MSSD10–40 re-

[1] https://www.tf-pm.org/competitions-awards/bpi-challenge

[2] https://github.com/explainableMDPs/mdp_attributive

[3] https://www.gurobi.com/

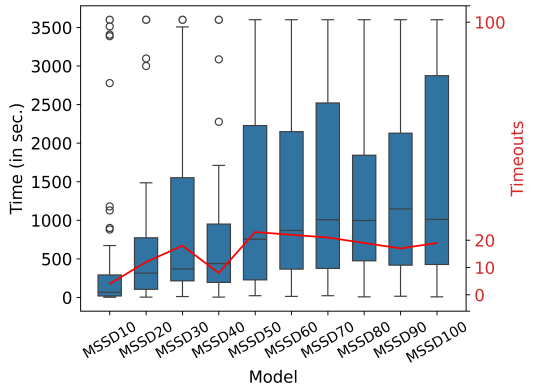


Figure 3: Summarized MSSD runtime results in seconds with timeouts for computing lower-bounded importance using encoding LP*.

ported few timeouts, from MSSD50, up to 23 runs timed out.

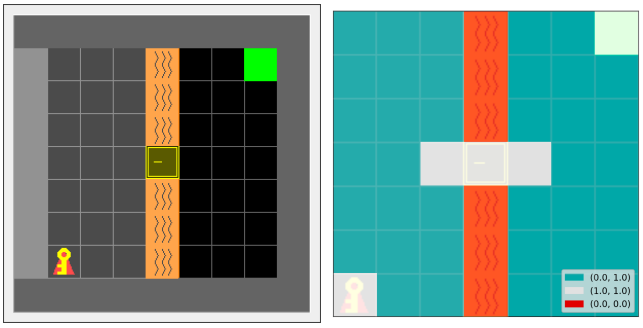
Interpretability of our scores. Attribution-based explanations can support end-users in gaining insights into the decision process by highlighting the most crucial steps. However, to be useful, attribution scores need to be understandable. Indeed, we observe that our technique is most informative when bounds on attribution scores are tight. In such cases, our attribution-based explanations lend themselves to a textual representation that summarizes the most important states, respectively paths, with high lower bounds, and most detrimental states, respectively paths, with low upper bounds. An excerpt of a textual representation of the most influential and detrimental states for receiving a loan in BPIC12 is:

```
States indispensable for a loan application include:
 A_CreateApplication, O_Accepted, and A_Pending.
States detrimental for a loan application include:
 SUPER LONG calls regarding incomplete information.
```

Additionally, process representations where states and paths are colored with heat maps according to their importance can communicate attribution-based explanations well, as demonstrated in Example 3 on a gridworld example.

Example 3. *Figure 4a shows a deterministic gridworld where the agent (red triangle) has to pick up a key, required to cross a river of deadly lava to reach the green target state. This gridworld can be modeled as an MDP where each state corresponds to an agent position, the target state is the state encoding the agent at the green tile position, and touching the lava leads to unsuccessful end states. The actions correspond to moving the agent in the 4-neighborhood, and picking up the key. The key is required to pass through the door, but no "open" action is needed. Without having the key, it is not possible to move on the door-field.*

Our importance scores enable answering questions such as: how important is visiting state s for reaching the target? *(**P2**), or,* how important is following path τ for reaching the target? *(**P4**). The importance of any state is depicted in Fig. 4b via a heat map overlay. There exists a strategy that leads the agent via any non-lava state to the target state, and there exists no strategy that leads the agent via any lava state to the green state. However, there exists no successful strategy where the state across the lava receives an importance with a lower bound of less than* 1.



(a) Gridworld Example. (b) Gridworld Heat Map Overlay.

Figure 4: Gridworld example, where the agent has to reach the green tile, while avoiding the lava tiles. The heat map on the right visualizes the importance from the corresponding MDP and highlights that most states are optional. The three white colored transfer-states are indispensable, and the six red states should be avoided.

Threats to validity. The main threat to validity in our experiments is numerical instability, related to the chosen constants for M and ϵ. Due to its size, M can introduce numerical inaccuracies as MDPs' transition probabilities might become very small, spanning a large range of values in the model. However, the differences between the LP* and QP* encodings are small: on average, results differ by less than $4 \cdot 10^{-4}$. A table showing the differences is included in [Kobialka *et al.*, 2026]. In summary, attributive explanations can be computed efficiently for complex MDPs. Notably, attributions are possible for strategies with optimal reachability in MDPs with up to 10 000 states and transitions.

7 Conclusion

This paper introduces attribution-based explanations for Markov Decision Processes to address the limitations of static feature importance in sequential decision-making. We provide a formal characterization for attribution-based explanations in this context, including the assignment of importance scores to individual states as well as execution paths in the model. We also propose an optimization approach for computing these scores and demonstrate its scalability across a wide range of sequential decision-making problems.

Our work opens several interesting directions for future work. First, we plan to investigate extensions of our framework towards computing importance of subpaths as opposed to full paths (from initial to goal states) considered here. To this aim, we will investigate a formal grammar capable of identifying semantically interesting segments within an execution. Second, we intend to explore the application of our techniques to Deep Reinforcement Learning. This will require novel abstraction methods to map latent representations learned by neural networks onto our formal attribution framework, bridging the gap between symbolic reasoning and neural decision-making. Finally, we believe it would be important to conduct user studies to evaluate the interpretability of our attribution scheme and assess how effectively these explanations help end users understand system behavior.

References

[Anandalingam and Friesz, 1992] Gnana Anandalingam and Terry L Friesz. Hierarchical optimization: An introduction. *Annals of operations research*, 34(1):1–11, 1992.

[Baier and Katoen, 2008] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT press, 2008.

[Baier *et al.*, 2021] Christel Baier, Florian Funke, and Rupak Majumdar. Responsibility attribution in parameterized Markovian models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11734–11743. AAAI Press, 2021.

[Baier *et al.*, 2025] Christel Baier, Sascha Klüppelholz, and Johannes Lehmann. Responsibility in actor-based systems. In *Rebeca for Actor Analysis in Action: Essays Dedicated to Marjan Sirjani on the Occasion of Her 60th Birthday*, pages 44–69. Springer, 2025.

[Banzhaf, 1965] J.F. Banzhaf. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19(2):317–343, 1965.

[Billionnet *et al.*, 2016] Alain Billionnet, Sourour Elloumi, and Amélie Lambert. Exact quadratic convex reformulations of mixed-integer quadratically constrained problems. *Mathematical Programming*, 158(1):235–266, 2016.

[Bodria *et al.*, 2023] Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi, and Salvatore Rinzivillo. Benchmarking and survey of explanation methods for black box models. *Data Mining and Knowledge Discovery*, 37(5):1719–1778, 2023.

[Boyd and Vandenberghe, 2004] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

[Brost *et al.*, 2019] Brian Brost, Rishabh Mehrotra, and Tristan Jehan. The music streaming sessions dataset. In *The World Wide Web Conference*, pages 2594–2600, 2019.

[Garey and Johnson, 1979] Michael R Garey and David S Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[Kazemi *et al.*, 2025] Milad Kazemi, Jessica Lally, Ekaterina Tishchenko, Hana Chockler, and Nicola Paoletti. Counterfactual influence in Markov decision processes. In *Causal Learning and Reasoning*, pages 792–817. PMLR, 2025.

[Kobialka *et al.*, 2024] Paul Kobialka, S Lizeth Tapia Tarifa, Gunnar R Bergersen, and Einar Broch Johnsen. User journey games: Automating user-centric analysis. *Software and Systems Modeling*, 23(3):605–624, 2024.

[Kobialka *et al.*, 2025] Paul Kobialka, Lina Gerlach, Francesco Leofante, Erika Ábrahám, Silvia Lizeth Tapia Tarifa, and Einar Broch Johnsen. Counterfactual strategies for Markov decision processes. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pages 412–420. ijcai.org, 2025.

[Kobialka *et al.*, 2026] Paul Kobialka, Andrea Pferscher, Francesco Leofante, Erika Ábrahám, Silvia Lizeth Tapia Tarifa, and Einar Broch Johnsen. Attribution-based explanations for markov decision processes. *arXiv preprint arXiv:2605.09780*, 2026.

[Leo *et al.*, 2019] Martin Leo, Suneel Sharma, and Koilakuntla Maddulety. Machine learning in banking risk management: A literature review. *Risks*, 7(1):29, 2019.

[Lundberg and Lee, 2017] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4765–4774, 2017.

[Mao *et al.*, 2016] Hua Mao, Yingke Chen, Manfred Jaeger, Thomas D Nielsen, Kim G Larsen, and Brian Nielsen. Learning deterministic probabilistic automata from a model checking perspective. *Machine Learning*, 105(2):255–299, 2016.

[Molnar, 2020] Christoph Molnar. *Interpretable Machine Learning*. Lulu. com, 2020.

[Montevechi *et al.*, 2024] André Aoun Montevechi, Rafael de Carvalho Miranda, André Luiz Medeiros, and José Arnaldo Barra Montevechi. Advancing credit risk modelling with machine learning: A comprehensive review of the state-of-the-art. *Engineering Applications of Artificial Intelligence*, 137:109082, 2024.

[Muskardin *et al.*, 2022] Edi Muskardin, Bernhard K. Aichernig, Ingo Pill, Andrea Pferscher, and Martin Tappler. AALpy: An active automata learning library. *Innov. Syst. Softw. Eng.*, 18(3):417–426, 2022.

[Pnueli, 1977] Amir Pnueli. The temporal logic of programs. In *18th annual symposium on foundations of computer science (sfcs 1977)*, pages 46–57. ieee, 1977.

[Ribeiro *et al.*, 2016] Marco Túlio 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, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144. ACM, 2016.

[Selvaraju *et al.*, 2017] 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 *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 618–626. IEEE Computer Society, 2017.

[Shapley, 1953] L.S. Shapley. A value for n-person games. 1953.

[Shi *et al.*, 2022] Si Shi, Rita Tse, Wuman Luo, Stefano D'Addona, and Giovanni Pau. Machine learning-driven credit risk: a systemic review. *Neural Computing and Applications*, 34(17):14327–14339, 2022.

[Sundararajan and Najmi, 2020] Mukund Sundararajan and Amir Najmi. The many Shapley values for model explanation. In *International conference on machine learning*, pages 9269–9278. PMLR, 2020.

[Teinemaa *et al.*, 2019] Irene Teinemaa, Marlon Dumas, Marcello La Rosa, and Fabrizio Maria Maggi. Outcome-oriented predictive process monitoring: Review and benchmark. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(2):1–57, 2019.

[Triantafyllou *et al.*, 2021] Stelios Triantafyllou, Adish Singla, and Goran Radanovic. On blame attribution for accountable multi-agent sequential decision making. *Advances in Neural Information Processing Systems*, 34:15774–15786, 2021.

[van Dongen, 2012] Boudewijn van Dongen. BPI Challenge, 2012.

[van Dongen, 2017] Boudewijn van Dongen. BPI Challenge, 2017.

[Wimmer *et al.*, 2012] Ralf Wimmer, Nils Jansen, Erika Ábrahám, Bernd Becker, and Joost-Pieter Katoen. Minimal critical subsystems for discrete-time Markov models. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 299–314. Springer, 2012.