International Journal on Software Tools for Technology Transfer
https://doi.org/10.1007/s10009-023-00704-3

EXPLANATION PARADIGMS LEVERAGING ANALYTIC INTUITION

Special Section: Introducing Explanation Paradigms Leveraging Analytic Intuition

Decision-making under uncertainty: beyond probabilities

Challenges and perspectives

Thom Badings[1] · Thiago D. Simão[1] · Marnix Suilen[1] · Nils Jansen[1]

Accepted: 4 April 2023

Abstract

This position paper reflects on the state-of-the-art in decision-making under uncertainty. A classical assumption is that probabilities can sufficiently capture all uncertainty in a system. In this paper, the focus is on the uncertainty that goes beyond this classical interpretation, particularly by employing a clear distinction between aleatoric and epistemic uncertainty. The paper features an overview of Markov decision processes (MDPs) and extensions to account for partial observability and adversarial behavior. These models sufficiently capture aleatoric uncertainty, but fail to account for epistemic uncertainty robustly. Consequently, we present a thorough overview of so-called uncertainty models that exhibit uncertainty in a more robust interpretation. We show several solution techniques for both discrete and continuous models, ranging from formal verification, over control-based abstractions, to reinforcement learning. As an integral part of this paper, we list and discuss several key challenges that arise when dealing with rich types of uncertainty in a model-based fashion.

Keywords Decision-making under uncertainty · Markov decision process · Partially observable Markov decision process · Formal abstractions · Reinforcement learning · Epistemic uncertainty · Aleatoric uncertainty

1 Introduction

Artificial intelligence (AI) enters our everyday life, often in critical domains such as health, defense, energy, or transportation. AI systems have to make intelligent decisions within such domains that are often safety-critical, yet, at the same time, have to deal with the inherent uncertainty that arises in the real world. This position paper reflects on a particular branch of AI, called *decision-making under uncertainty* [86].

How does uncertainty affect AI decision-making? We discuss the concept of uncertainty beyond its generic use. Generally, uncertainty has been "largely related to the lack of predictability of some major events or stakes, or a lack of data" [11]. To name a few, there is uncertainty (1) in technological, social, environmental, or financial factors in the business literature [139], (2) in greenhouse gas emissions and concentrations for climate modeling [67], (3) about sensor imprecision and lossy communication channels in robotics [153], and (4) on the expected responses of a human operator in decision support systems [86]. The level and type of uncertainty affect the capabilities of AI systems to make intelligent decisions [6, 86]. A deterministic environment implies perfect information, and each decision has a single outcome. The real world, however, is uncertain. Let us give a small example [164]. A robot perceives its environment and potential obstacles through a noisy sensor. A naive way to deal with this uncertainty is to assume the sensor data is always correct. Because of the imperfect measurements, the robot may, at some point, make a disastrous decision. Alternatively, the robot may use Bayesian reasoning [60, 64]: the probability that the sensor reading is correct is used to update the belief about the robot's environment. Over time, the confidence in the position of the obstacles will grow. We distinguish *aleatoric* and *epistemic uncertainty* [140]. Aleatoric uncertainty is intrinsic to the environment and quantifies unknowns, for instance, partial observability due to measurement noise. Epistemic uncertainty indicates a lack of knowledge and is reducible by collecting more data. For example, by making more measurements, the robot can estimate the level of noise of its sensor more accurately.

How to capture uncertainty within a model? State-of-the-art approaches use models, in particular Markov decision pro-

✉ T. Badings

[1] Department of Software Science, Radboud University, Nijmegen, The Netherlands

Published online: 30 May 2023

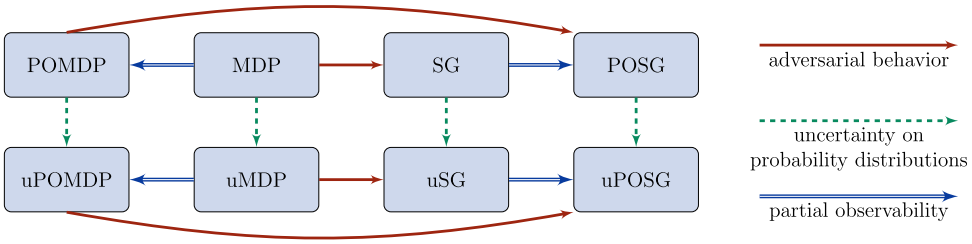


Fig. 1 A family of closely related uncertainty models that we cover in this paper. Adversarial behavior increases from left to right. The left and right columns are partially observable models. Finally, the bottom row shows models that (in addition to probabilistic and adversarial behavior) account for uncertainty in probability distributions

cesses (MDPs), to capture sequential decision-making problems for agents operating in uncertain environments [119]. Sensor limitations may lead to partial observability about the system's current state, giving rise to partially observable Markov decision processes (POMDPs) [82]. MDPs augmented with a model of adversarial behavior are stochastic games (SGs) [45]. Their partially observable counterpart is a POSG [35, 70]. Finally, all of these models have continuous counterparts, which are often formalized as dynamical models [12, 31].

Precise probabilities are not enough. The likelihood of uncertain events, such as a message loss in communication channels or specific responses by human operators, may only be an estimate from data. The models introduced above capture uncertainty in the form of precise probabilities—either in their transition dynamics or in their observation models. However, such point estimates of probabilities from data carry the risk of statistical errors. Moreover, the optimal policies for agents are usually highly sensitive to small perturbations in transition probabilities, leading to suboptimal outcomes such as a deterioration in performance [68, 100]. Uncertainty models remove this assumption by incorporating uncertainty sets of probabilities. In the literature, uncertain MDPs (uMDPs) use, for example, probability intervals or likelihood functions [66, 77, 108, 118, 162, 163, 165]. Similar extensions exist for uncertain POMDPs (uPOMDPs), where uncertainty may also affect the observation model [32, 33, 52, 76, 143]. To the best of our knowledge, there is no prior work on uncertain POSGs (uPOSGs). Figure 1 shows a family of the uncertainty models that we are interested in, capturing different types of uncertainty and their relation to each other. The three different types of arrows indicate the addition of (1) adversarial behavior, (2) uncertainty on probability distributions, and (3) partial observability from one model to another.

Different solutions across the research areas. We focus on decision-making scenarios that can sufficiently be described by uncertainty models.[1] A general problem is then to synthesize a policy for such a model that satisfies a certain goal. Such a goal may, for instance, refer to maximizing a reward measure or satisfying a (formal) specification in temporal logic [115]. This policy synthesis problem is the subject of active research throughout different areas: AI, formal verification, optimization, and control theory.

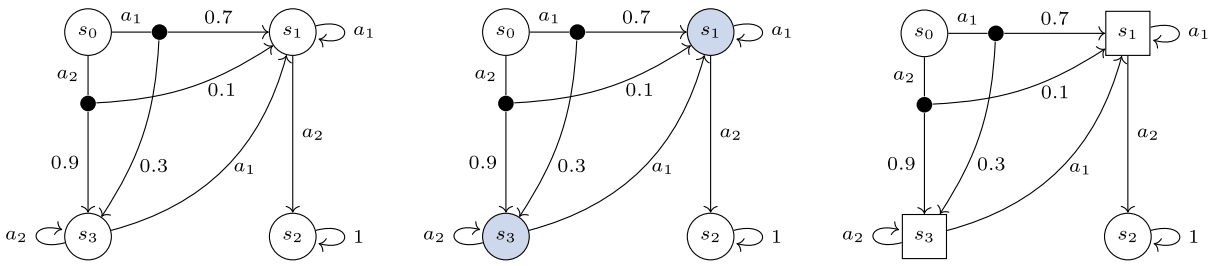
Challenges and perspectives. In this paper, we provide an overview of techniques for decision-making under uncertainty that stem from reinforcement learning (RL) [146], model checking [23, 49], systems and control [168], and convex optimization [30]. We highlight and discuss various assumptions and challenges that are central to these techniques, such as prior knowledge, data availability, theoretical complexity, and the guarantees that are possible in the various settings. For example, settings that exhibit strict safety requirements require decisions that are verifiably robust against uncertainty [139]. Such considerations require precise knowledge about the nature of uncertainty.

We structure this paper as follows. In Sect. 2, we highlight various types of uncertainty models and their properties. In Sect. 3, we describe state-of-the-art planning approaches to solve them against different kinds of specifications. In Sect. 4, we detail recent progress on dealing with uncertainty in realistic, continuous spaces, and in Sect. 5, we discuss various approaches in reinforcement learning that deal with uncertainty. Finally, in Sect. 6, we discuss a number of important challenges to this research area and provide an outlook on potential future work and directions.

2 Modeling under uncertainty

Decision-making under uncertainty from a model-based perspective classically revolves around Markov decision processes (MDPs) [119]. An MDP is defined by a tuple (S, s_i, A, P), where S is a set of states, $s_i \in S$ is the initial

[1] We do not assume per se that a model is available.



(a) An MDP. (b) A POMDP. (c) An SG.

Fig. 2 Examples of a classical MDP, POMDP, and SG

state, A is a set of actions, and $P\colon S \times A \to \mathit{Distr}(S)$ is the probabilistic transition function that maps each enabled state-action pair to a probability distribution over successor states. The probabilistic transition function may be partial, reflecting that not every action is necessarily enabled in every state. An example of an MDP can be seen in Fig. 2a.

A policy (also called scheduler, strategy, or controller) resolves the non-determinism of an MDP. Formally, a finite-memory policy is a function $\pi\colon (S \times A)^* \times S \to \mathit{Distr}(A)$ that maps sequences of states and actions to a distribution over actions. If the policy accounts for only a single state, i.e., it is of the form $\pi\colon S \to \mathit{Distr}(A)$, it is called memoryless. A policy is deterministic if it maps each state to a single action, i.e., $\pi\colon S \to A$.

MDPs can be extended with a reward function $R\colon S \times A \to \mathbb{R}$, assigning a real-valued reward to each state-action pair. Let r_t be the reward collected at time t when following policy π, and $\gamma \in (0,1]$ a discount factor. We refer to the accumulated (discounted) rewards under π and γ as the *return* $G = \sum_t \gamma^t r_t$. Then, the goal is to find a policy π that maximizes the expected return:

$$\mathit{arg\,max}_\pi \mathbb{E}_\pi [G]\,. \tag{1}$$

In this paper, we primarily focus on temporal logic objectives [115]. For temporal logic objectives, the goal is to find a policy that maximizes the probability with which a temporal logic formula φ is satisfied:

$$\mathit{arg\,max}_\pi \mathbb{P}_\pi [\varphi],$$

where $\mathbb{P}_\pi$ is the probability measure of the Markov chain induced by the MDP with policy π (see, e.g., [23] for details). We particularly employ reachability ($\varphi = \Diamond T$) and reach-avoid ($\varphi = \neg B \,\mathsf{U}\, T$) objectives or their time-bounded analogue, where T is a set of target states, and B is a set of "bad" states to be avoided. Computing policies that optimize for reachability or expected reward is decidable in polynomial time, and 2EXPTIME-complete for general temporal logic specifications [23].

Example 2.1
For the MDP given in Fig. 2a, an optimal memoryless deterministic policy for eventually reaching s_2 with probability 1 is, for instance, choosing a_1 in s_0 and s_3, and a_2 otherwise. □

2.1 Partial observability

Partially observable MDPs (POMDPs) are a common extension of MDPs to account for limited information in the decision-making problem [82]. Formally, a POMDP is a tuple (S, s_i, A, P, Z, O), where (S, s_i, A, P) forms an MDP, Z is a set of observations, and $O\colon S \times A \to \mathit{Distr}(Z)$ is the probabilistic observation function. An example POMDP with state-based observations represented by shaded states is presented in Fig. 2b.

A POMDP is equivalent to a fully-observable, infinite-state MDP, called the *belief MDP*. Each state of this MDP represents a belief: a probability distribution over the (finite) states of the POMDP that summarizes the history of all observations and actions so far. Upon taking an action and receiving an observation, the current belief can be updated to a new belief via the standard belief update function [82].

A policy in a POMDP is a policy in the belief MDP. That is, a function that maps beliefs to actions, $\pi\colon \mathit{Distr}(S) \to A$. Alternatively, we may also consider only a part of the full history. Then, π is of the form $\pi\colon (Z \times A)^* \times Z \to \mathit{Distr}(A)$, and is called a *finite-memory* policy. Where computing optimal policies in MDPs is decidable, and even in polynomial time for expected reward or reachability properties [23], it is undecidable in POMDPs [98]. Restricting to finite-memory policies renders the problem decidable, but the resulting policies may be sub-optimal. Randomizing over the actions may be used to trade off memory size. Already computing a memoryless randomized policy, i.e., of type $\pi\colon Z \to \mathit{Distr}(A)$, is NP-hard in POMDPs [159].

Example 2.2
For the POMDP in Fig. 2b, an optimal policy for reaching state s_2 exists, but requires either finite-memory or randomization. The key problem is that an agent needs to distinguish

between states s_1 and s_3, since in s_1 action a_2 is the optimal choice, and in s_3 the agent should choose a_1. By (for instance, uniformly) randomizing over action a_1 and a_2 when the observation is "blue", the agent will eventually reach s_2 with probability 1. □

Most POMDP methods rely on the reduction to a belief MDP to then perform value iteration [82, 137], policy iteration [69, 101], or point-based methods [114, 142, 160]. Alternatively, approaches exploit a reduction to an optimization problem [7, 81], or employing recurrent neural networks as policy representation [37–39, 71].

2.2 Adversarial behavior

Besides partial observability, we may also extend MDPs with one (or multiple) adversaries, effectively defining a stochastic game (SG). In a two-player stochastic game, the set of states is partitioned into two parts, and each player may control the actions in their states.

Example 2.3
A two-player SG is shown in Fig. 2c, where the shape of the states (squares and circles) indicates which player the state belongs to. In this SG, the square player can prevent the game from reaching s_2 by always choosing a_1 in their state s_1. Hence, there is no winning policy for the circle player when starting in s_0. □

Efficient implementations exist, for instance, as part of the model checking tool PRISM-GAMES [90]. Such a stochastic game may also be made partially observable, yielding a partially observable stochastic game (POSG). Due to the generality of POSGs, they cover numerous application areas such as robotics [87], cybersecurity [74], and air-traffic control [129]. However, computing a reward-optimal policy for an agent in a POSG, for instance using dynamic programming, is notoriously hard [70]. Approximate methods deal with small settings, while realistic problems remain largely intractable [57, 73, 89].

2.3 Classifying uncertainty

Uncertainty is often classified into two classes, namely aleatoric and epistemic uncertainty [61, 75, 145]. Distinguishing aleatoric from epistemic uncertainty is identified as a key challenge towards trustworthy AI [151].

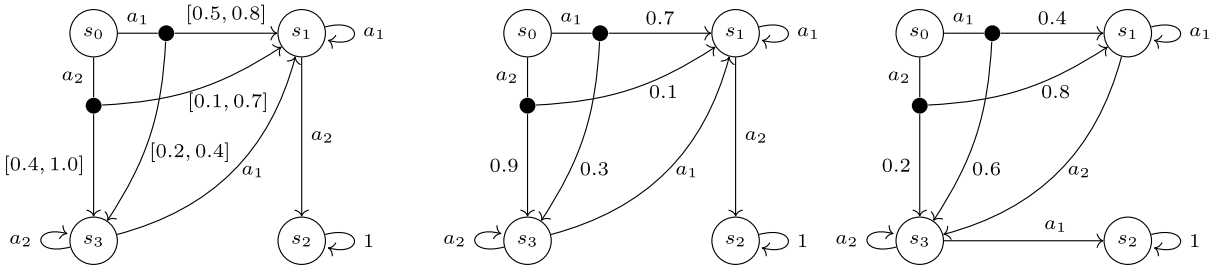
Aleatoric uncertainty. Aleatoric uncertainty (also called statistical uncertainty) describes the natural variability and randomness of processes. Consider, for example, the action of accelerating an autonomous car by a fixed force. The car will not reach the same velocity every time that we repeat this action, due to random and complicated effects that cannot be determined sufficiently accurately. Aleatoric uncertainty is captured by probability distributions over the outcomes of actions and can thus be naturally modeled by the transition probabilities of MDPs. Similarly, aleatoric uncertainty about measurement processes can be captured by the probabilistic observation function of a POMDP. Aleatoric uncertainty is irreducible in the sense that it is not realistically possible (what is "realistic" may boil down to a philosophical debate) to gather the additional knowledge needed to eliminate the randomness.

Epistemic uncertainty. By contrast, epistemic uncertainty (also called systematic uncertainty) is caused by a systemic lack of knowledge, and can thus be reduced by gathering more knowledge about the system [138]. Take, for example, an autonomous car whose mass is only known to lie between 950 – 1050 kg, i.e., there is epistemic uncertainty about the mass of the car. The mass clearly affects the acceleration of the car in response to a certain input to the engine. However, without any further information about the likelihood of certain values for the mass, there is no logical justification for taking a stochastic perspective to reason about the probability that the car behaves in a certain way. Note that if such likelihoods are known, epistemic uncertainty may still be captured by probabilistic models, as is commonly done in Bayesian approaches [64]. Epistemic uncertainty can be reduced by collecting more data. For example, we may improve our knowledge about the mass of the car by collecting more accurate measurements of its weight.

Mixed uncertainty types. Besides aleatoric and epistemic uncertainty in pure form, mixtures between these two uncertainty types also exist. In fact, these mixed uncertainties are of huge importance for the uncertainty models that we will introduce in Sect. 3. Consider, for example, a system whose underlying model is an MDP, but the transition probabilities are only known to lie in a particular set. Thus, there is epistemic uncertainty (which we may reduce by, e.g., sampling the MDP) about the aleatoric uncertainty (the probabilistic transitions of the MDP). In Sect. 3, we will discuss several ways of dealing with such mixtures between aleatoric and epistemic uncertainty.

3 Planning under uncertainty

The classical models for decision-making under uncertainty are MDPs and POMDPs, and SGs in multi-agent settings. These models deal with uncertainty in the aleatoric form by using probability distributions on the outcomes of actions. In this section, we extend the notion of uncertainty in these models in various ways, particularly by adding uncertainty



(a) A uMDP. (b) A MEMDP with two environments.

Fig. 3 Extensions of an MDP with continuous uncertainty (uMDP) and discrete uncertainty (MEMDP)

of the epistemic form. We discuss how to deal with these additional uncertainties in the policy synthesis problem and how to learn (and possibly reduce) the degree of uncertainty from data.

3.1 Sets of (PO)MDPs

An *uncertain MDP* (uMDP; also known as *robust MDP*) is an MDP where the probability distributions over successor states at each state-action pair are replaced by a set of possible distributions [108, 162]. An uncertain MDP can be viewed as a set $\mathcal{M}$ of (uncountably many) standard MDPs M. Consequently, we write $M \in \mathcal{M}$ for an MDP M that is contained in the uMDP $\mathcal{M}$.

If we assume there exists one true MDP within this set, then uMDPs can be seen as a layer of epistemic uncertainty on top of the transition probabilities of the true model, which can be reduced by gathering information. Additionally, uMDPs are a form of stochastic game where at each state one player chooses the actions, and the adversary chooses the probability distribution.

The most common way to define uMDPs is by replacing the individual transition probabilities with probability intervals. In that case, the uMDP is also called an interval MDP (iMDP), and the uncertainty set at a state-action pair is defined as a *convex polytope* constructed by intersecting the Cartesian product of the intervals with the set of all possible distributions over the successor states. Such a uMDP is illustrated in Fig. 3a. Alternative forms of uncertainty sets have also been considered, most notably convex uncertainties [118], such as ellipsoidal [27] and L_1-distance based sets, most commonly used in reinforcement learning [78].

A common goal in a uMDP $\mathcal{M}$ is to compute a policy that maximizes the expected return under the worst-case instance of the uncertainty, typically denoted as a max-min problem:

$$arg\,max_\pi \min_{M \in \mathcal{M}} \mathbb{E}_\pi^M [G], \tag{2}$$

or, in the case of a temporal logic formula φ:

$$arg\,max_\pi \min_{M \in \mathcal{M}} \mathbb{P}_\pi^M [\varphi]. \tag{3}$$

Computing such policies can be done via (robust) dynamic programming [108, 163] or convex optimization [118].

Related to this is the notion of optimism in the face of uncertainty [106], which is typically used as an exploration strategy in reinforcement learning. Instead of choosing the worst-case model M, we now choose the best-case model M by also maximizing over the set of models $\mathcal{M}$, that is, a max-max problem. If the goal of the decision-maker is to minimize, we may alternatively speak of min-max and min-min problems, respectively. Similar to standard MDPs, computing such policies for simple reachability or expected return specifications can be done in polynomial time [162], provided the uncertainty set is convex (as mentioned above) and that the probability distribution of each state-action pair is independent of the others, also known as the *rectangularity* assumption (which we discuss in more detail below).

Example 3.1
In our example uMDP in Fig. 3a, when the agent chooses a_1 in s_0, the worst-case probability to go to s_1 is 0.6, as this is the lowest probability in the interval [0.5,0.8] that can add up to one with a probability (0.4) from the other transition interval [0.2,0.4]. Similarly, the optimistic probability here is 0.8. □

Uncertain POMDPs. Uncertain MDPs may also be extended with partial observability, in the same way extending MDPs to POMDPs works, effectively defining *uncertain POMDPs* (uPOMDPs) [143]. The standard decision-making problem in a uPOMDP is again the max-min (or min-max) problem, except that we are again restricted to (finite-memory) observation-based policies. Solution methods rely on a belief-based approach that minimizes over the uncertainty during the belief update [109], or convex optimization [52, 143]. To the best of our knowledge, no complexity results for uPOMDPs exist, though clearly standard POMDPs are included in uPOMDPs, hence problems cannot be easier.

Discrete model uncertainty. Uncertain MDPs form a continuous set of MDPs that vary only in their transition proba-

bilities. Analogously, we may also consider a *discrete* set of MDPs. A multiple-environment MDP (MEMDP) is a finite set of MDPs that share the same state and action spaces, and only differ in their transition functions [120]. In particular, these transition functions are not required to have the same support, meaning that each MDP in the MEMDP may have a different underlying graph.

Example 3.2
An example of a MEMDP is shown in Fig. 3b. The two environments not only differ in the transition probabilities on their shared transitions, but also in whether s_2 is directly reachable from s_1 or s_2. Thus, both MDPs in the MEMDP have a different underlying graph. Similar to the POMDP in Fig. 2b, this example MEMDP also shows the need for memory or randomization in the policy, as the agent does not know in which of the two s_1 states it is, and thus needs to (uniformly) randomize between a_1 and a_2 to eventually reach s_2 regardless of which environment the agent operates in. □

MEMDPs have been studied extensively and under many different names, among which hidden-model MDPs [41] and POMDP-lite [44]. Indeed, as that last alternative name suggests, MEMDPs have a strong connection to POMDPs. In fact, every MEMDP can be transformed into a POMDP by introducing a latent variable for the environment index into the state space [42], and many POMDP examples from the literature (such as the famous Tiger Problem [82]) are actually MEMDPs [44]. Solution methods for MEMDPs typically rely on casting the problem as a POMDP and then using POMDP solutions methods. Yet, MEMDPs form an interesting class of models on their own as computing policies that satisfy almost-sure parity objectives, which is undecidable for POMDPs [43], is decidable for MEMDPs [120].

Assumptions and limitations. One key underlying assumption typically used in uncertain (PO)MDPs is that all models in the set have the same topology. Concretely, this assumption ensures that while there is uncertainty about with which exact probability a transition will occur, it is known whether the transition is possible (with probability > 0) or not (with probability 0). Solution methods for both uMDPs and uPOMDPs, such as [52, 118, 143, 162, 163], rely on this assumption. Another assumption commonly made is the rectangular assumption, which states that the choice of distribution in the uncertainty set at one state-action pair is independent of the choice of distribution in any other state-action pair. This assumption is also key to efficient solution methods. Indeed, reachability or expected return objectives in uMDPs with rectangular uncertainty can be solved in polynomial time, whereas solving uMDPs with non-rectangular uncertainty is NP-hard [162]. Finally, there are multiple (semantic) interpretations of such uncertain models. The first one assumes that there is one true model within the set that is selected non-deterministically at the start, also referred to as a *stationary uncertainty model*. The other interpretation is that at every step (i.e., action choice) one of the models is chosen by an adversary, known as a *time-varying uncertainty model* [108].

3.2 Learning models and uncertainty sets

A fundamental question that arises is where the models, and, in particular, the uncertainty sets discussed above, come from. Clearly, a standard MDP could be learned from data by estimating the probabilities of the transition function via maximum likelihood estimation, i.e., fractions of empirical occurrences in some data set. Such estimates naturally introduce statistical errors, especially when the data set is small. A natural application of uncertainty sets and uMDPs presents itself here: we over-approximate the MDP we try to learn by a uMDP that (ideally) contains the actual MDP.

PAC learning. Probably approximately correct (PAC) learning of MDPs typically aims to learn a concrete MDP by deriving point estimates from data, and then extending these point estimates to intervals by including error margins that follow from concentration inequalities such as Hoeffding's inequality [72]. The resulting model is a uMDP with a probabilistic correctness guarantee on each individual transition. By distributing the confidence over all transitions, the PAC guarantee can be extended to the entire model, and, as a result, also to the optimal value of Eq. (2) and (3). This latter approach is used in, e.g., PAC statistical model checking [14]. Hoeffding's inequality provides an upper bound on the probability that a point estimate of a random variable deviates from its expected value by more than a certain value, but this upper bound is typically very conservative in practice. Furthermore, Hoeffding's inequality relies on independent and identically distributed (i.i.d.) sampling from a fixed distribution. Thus, Hoeffding's inequality cannot be applied to cases where the underlying model that is being learned may shift between distributions.

Model learning. Active automata learning, or model learning [157], typically makes no assumptions regarding the state space or the topology of the model. Instead, model learning infers the state space and the topology from observations by iteratively expanding a set of states. Model learning techniques for MDPs use point estimates of probabilities and make the assumption that the underlying MDP is deterministic, to uniquely identify states [149, 150].

Learning under distributional drift. The learning techniques discussed above rely on the fact that there is one fixed, true

model that generates the data used in the learning process. This assumption may not always be realistic. Probability distributions may suddenly change, for example due to hardware failures [169], or slowly drift due to deterioration of components. So-called *sliding window* (also called *receding horizon*) approaches try to deal with these cases [46, 62]. In such approaches, older data is deemed less valuable and is ignored if it falls outside a predefined time window. Recently, linearly updating intervals were suggested as an effective approach to deal with changing environments [144]. This method provides a flexible Bayesian framework that iteratively updates a uMDP in accordance with new data. While not providing formal guarantees in terms of correctness, the approach performs well in empirical evaluations and can easily adapt to distributional shifts by updating the uncertainty model accordingly.

4 Continuous control under uncertainty

Having explored a broad family of discrete Markov models, we now shift our attention to continuous state and action models. While such continuous models can often be expressed as infinite or continuous MDPs, it is generally more convenient to formalize models as a dynamical model (we focus on the discrete-time case) [12, 31]. While dynamical models form the continuous analog of MDPs and POMDPs, dynamical models generally exhibit more structure and smoothness in their transition (and observation) functions across the state and action spaces. Formally, a dynamical model is characterized by a (deterministic) state transition function (also called kernel) $f\colon \mathbb{R}^n \times \mathcal{U} \times \mathbb{R}^p \to \mathbb{R}^n$ that maps the current state $x_k \in \mathbb{R}^n$, a control input (i.e., an action) $u_k \in \mathcal{U} \subseteq \mathbb{R}^m$, and a vector of disturbances $w_k \in \mathbb{R}^p$ to a successor state $x_{k+1} \in \mathbb{R}^n$. To account for partial observability and sensor imprecision, we may define a separate observation model $g\colon \mathbb{R}^n \times \mathbb{R}^q \to \mathbb{R}^r$ that is independent of the state transition model, and which maps the state $x_k \in \mathbb{R}^n$ and another vector of disturbances $v_k \in \mathbb{R}^q$ to an observation $y_k \in \mathbb{R}^d$. The dynamical model is time-invariant if the functions f and g do not change with the time step $k \in \mathbb{N}$, yielding the pair of equations

$$x_{k+1} = f(x_k, u_k, w_k), \tag{4a}$$

$$y_k = g(x_k, v_k). \tag{4b}$$

We deliberately leave the mechanism by which the disturbances w_k and v_k are determined unspecified. As we shall see, depending on this mechanism, the disturbance may reflect various types of uncertainty, including set-bounded uncertain parameters and stochastic noise terms. If w_k and v_k are precisely known for each time step k the dynamical model is deterministic, and if $x_k = y_k$ for each k, the model is fully observable.

Linear dynamical models. One important class of dynamical models concerns state transition and observation functions f and g that are linear in their arguments. In such a linear dynamical model, also called linear time-invariant (LTI) system if the functions f and g are time-invariant, the successor state x_{k+1} and the observation y_k are computed as linear combinations of their respective arguments:

$$x_{k+1} = Ax_k + Bu_k + w_k, \tag{5a}$$

$$y_k = Cx_k + v_k, \tag{5b}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{d\times n}$ are matrices of appropriate size. Linear dynamical models find important applications in many research areas, including control theory [154], power system modeling [126], mechanical engineering [10], and signal processing [93].

Example 4.1
The position p_k and velocity v_k of a drone moving along a straight line can be modeled as a linear dynamical model with a 2-dimensional state $x_k = [p_k, v_k]^\top$ and dynamics defined as

$$x_{k+1} = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} \tau^2 \\ \tau \end{bmatrix} u_k + w_k, \tag{6}$$

where $u_k \in \mathcal{U} = [\underline{u}, \overline{u}]$ is the force applied to the drone at time step $k \in \mathbb{N}$, $\tau > 0$ is the discretization time, and w_k is the disturbance vector. Now assume that we have access to noisy measurements of only the position, but not the velocity, of the drone. We model this through the observation model as

$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k + v_k, \tag{7}$$

where v_k is the measurement disturbance vector. □

4.1 Capturing uncertainty in dynamical models

Like Markov models, dynamical models can be used to capture various sources of uncertainty, including stochastic noise, set-bounded disturbances, and partial and/or limited observability of the system's state.

Stochastic uncertainty in dynamical models. We can capture stochastic uncertainty in dynamical models by respectively defining the disturbances w_k and v_k to be stochastic processes. The term w_k affects the state transitions and is typically called *process noise*, whereas v_k affects the observations and is called *measurement noise*. When analyzing dynamical models with stochastic noise, the typical goal is to reason over the probability that the system generates certain state trajectories (analogous to reasoning over probability distributions in MDPs).

Example 4.2
For the drone model in Example 4.1, we can account for stochastic factors in the environment (e.g., the influence of the wind) by defining w_k as a Gaussian (or any other) distribution, i.e., $w_k = \mathcal{N}(\mu_{w_k}, \Sigma_{w_k})$, where μ_{w_k} and Σ_{w_k} are the mean and covariance matrix. Similarly, we can account for normally distributed measurement errors by defining $v_k = \mathcal{N}(\mu_{v_k}, \Sigma_{v_k})$. □

Set-bounded disturbances in dynamical models. Recall from Sect. 2 that in some cases it is unrealistic to employ a probabilistic (stochastic) model for the uncertainty. Instead, to capture uncertainty in a dynamical model for which no likelihoods of each possible outcome are known, we can define $w_k \in \mathcal{W}$ or $v_k \in \mathcal{V}$ to be unknown yet bounded disturbances, where $\mathcal{W}$ and $\mathcal{V}$ are uncertainty sets. To achieve computational tractability, the uncertainty sets $\mathcal{W}$ and $\mathcal{V}$ are typically convex (hyperrectangles, in the simplest case). In the linear dynamical model in Eq. (5a)–(5b), we can additionally make the matrices A, B, and C dependent on additional set-bounded parameters, see, e.g., [22]. We typically take a robust approach [26], meaning that we aim to generate a solution that is valid *for all* values of the disturbances or the uncertain parameters in their domain. When we take a robust approach and assume that the value of the disturbance can take on any value in its set, then the outcome of a control input is nondeterministic.

Example 4.3
We modify the dynamics in Example 4.1 to explicitly account for the weight $m > 0$ of the drone:

$$x_{k+1} = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} \frac{\tau^2}{m} \\ \frac{\tau}{m} \end{bmatrix} u_k + w_k, \tag{8}$$

i.e., the larger the weight, the higher the force needed to change the state of the drone. Assume that the weight is only known to lie in a certain interval, $m \in [\underline{m}, \overline{m}]$. Contrary to Example 4.2, we do not have information about the likelihood of each value for the mass in the interval $[\underline{m}, \overline{m}]$, so employing a probabilistic model is unrealistic. Instead, we aim to generate a controller that performs robustly against any values $m \in [\underline{m}, \overline{m}]$. □

Partial observability in dynamical models. A clear separation of the transition and observation model enables us to capture partial observability, as with POMDPs. The features of the observation y_k reflect quantities relating to the system that is observed from the outside, while x_k models the internal state of the system. The state x_k and observation y_k may not contain the same features, nor do they need to have the same dimension.

Partial observability does not necessarily mean that the dynamical model is not observable in control-theoretic terms. Roughly speaking, a dynamical model is said to be observable if its internal state x_k can be reconstructed from a series of outputs $y_1, y_2, \ldots$ only [15]. For example, the model in Example 4.1 is still observable, since two consecutive measurements y_k, y_{k+1} will also reveal the velocity of the drone. If a dynamical model is not observable, then there exist state trajectories $x_1, \ldots, x_k$ that cannot be distinguished from their produced outputs $y_1, \ldots, y_{k-1}$ only.

4.2 Expressing aleatoric and epistemic uncertainty

We now discuss how to use stochastic noise, set-bounded disturbances, and partial/limited observability to express aleatoric and epistemic uncertainty in dynamical models.

Aleatoric uncertainty. Recall from Sect. 2 that aleatoric uncertainty is characterized by probability distributions over the outcomes of actions. Thus, aleatoric uncertainty about the state transitions and observations of a dynamical model is naturally modeled by stochastic process and measurement noise, analogous to the transition probabilities in an MDP. Doing so, we can reason probabilistically over the paths generated by the dynamical model under different values of the aleatoric uncertainty. In principle, however, it is also possible to deal with aleatoric uncertainty from a robust perspective. For example, if the support of the distribution underlying the aleatoric uncertainty is bounded, we can also capture the uncertainty as a set-bounded disturbance. As such, we can enforce robustness against all possible outcomes. Robust approaches may be preferred with respect to safety constraints, but can also be significantly more conservative than probabilistic approaches.

Epistemic uncertainty. In principle, we can also reason probabilistically over epistemic uncertainty, as long as a prior distribution over the values for the uncertain parameter is known, as is common in Bayesian approaches [64]. Recall, however, that epistemic uncertainty is not always associated with such a distribution over possible outcomes, such as for the autonomous car from Sect. 2, whose mass is only known to lie in a certain interval. In the absence of a prior distribution for the likelihood of each value for the mass, it is common to model epistemic uncertainty in the form of set-bounded disturbances and take a robust approach [26]. Dealing with epistemic uncertainty in dynamical models from a robust perspective is analogous to the max-min (or min-max) problem for u(PO)MDPs.

4.3 Decision-making for dynamical models

The objective in decision-making for dynamical models under uncertainty is analogous to those for discrete MDPs and

POMDPs. The general synthesis problem is to compute a (feedback[2]) policy π such that the probability of satisfying a temporal logic formula is maximized (or, as with some methods, is above some predefined threshold). Policies for dynamical models are typically deterministic; that is, they map to a single control input rather than a distribution over inputs. In what follows, we present a non-exhaustive overview of approaches that can be used to solve the synthesis problem under various types of uncertainty.

Only stochastic uncertainty. In this case, the disturbances w_k and v_k are both stochastic processes. A common assumption to ensure computational tractability of the synthesis problem is that this stochastic process follows a Gaussian distribution [111]. One such classical setting is linear-quadratic-Gaussian (LQG) control [8], which considers a linear dynamical model with Gaussian noise and with a quadratic cost function in which case a closed-form solution exists for the optimal feedback controller. However, richer specifications (such as temporal logic formulae) do not admit algorithmic or closed-form solutions in general [28].

One popular approach to synthesizing controllers that provably satisfy temporal logic formulae is to create a discrete abstraction of the dynamical model in the form of an MDP [5, 91, 94, 141]. Under an appropriate simulation relation [65], guarantees about the satisfaction of a temporal logic formula on the abstract model carry over to the continuous system. Various approaches formalize discrete abstractions as uMDPs or interval MDPs. For example, the tool StocHy [40] synthesizes policies for stochastic hybrid systems by creating discrete abstractions that capture abstraction errors in the probability intervals of an iMDP. Similarly, [18, 19] use abstractions to synthesize certifiably safe controllers for dynamical models with stochastic uncertainty of unknown probability distribution about the state transition model. By sampling the stochastic noise of unknown distribution, [18, 19] compute PAC bounds on the transition probabilities of MDP abstractions of dynamical models, thus formalizing these abstract models as iMDPs.

Only set-bounded uncertainty. The synthesis problem for dynamical models with set-bounded disturbances has mostly been studied at the intersection of control theory and formal methods [25]. In particular, various approaches create discrete abstractions of such dynamical models in the form of deterministic finite transition systems on which temporal logic formulae are easily verified [99, 147]. Generally, safety objectives can be verified by over-approximating the set of reachable states under any possible value of the disturbance about which there exists uncertainty, while reachability objectives can be verified by under-approximations [121]. Besides abstraction, various approaches use optimization, such as [59], which synthesizes controllers for reach-avoid specifications on linear models with bounded disturbances.

Stochastic and set-bounded uncertainty. Decision-making and the synthesis problem for dynamical models with both stochastic and set-bounded uncertainty are largely understudied. The problem is that purely probabilistic approaches are only able to deal with stochastic uncertainty about the state transition and observation model, while deterministic reachability-based approaches only address set-bounded uncertainty about these models. For stability specifications, the problem has recently been considered from a control-theoretic approach by [103]. However, to provide guarantees about temporal logic specifications, abstractions into richer models, such as uncertain MDPs, are needed. This approach is taken by [95], who learn MDP abstractions with uncertain transition probabilities of dynamical models with discrete control input sets from data. Moreover, the recent paper [22] synthesizes provably correct controllers for dynamical models with stochastic (aleatoric) and set-bounded (epistemic) uncertainty, by generating interval MDP abstractions that simultaneously capture both types of uncertainty about the model dynamics.

The partially observable case. Decision-making for partially observable dynamical models typically relies on a recursive state estimator. Such a state estimator maintains a belief over the continuous state space based on previous observations and the available model of the dynamical model. The classical state estimator for linear dynamical models is the Kalman filter, which assumes Gaussian process and measurement noise, and also represents the belief as a Gaussian distribution over states [83, 117]. For linear dynamical models with additive Gaussian noise, the Kalman filter is an optimal state estimator in the minimum mean-square-error sense, i.e., its estimate is the least uncertain of any filter, given the same history of information. Kalman filters have been used by [21] to synthesize controllers that satisfy reach-avoid specifications for partially observable linear dynamical models by generating iMDP abstractions.

Another widely used state estimator is the particle filter, which is especially used for dynamical models with non-linear dynamics and non-Gaussian noise [153]. While the Kalman filter maintains the belief as a Gaussian distribution, the particle filter maintains the belief as a set of so-called particles [97, 130]. Intuitively, these particles are hypothesis states that are recursively propagated through the dynamical model by means of simulation methods. By weighing the particles after each simulation step based on their likelihood of being an accurate state estimate, the particle filter recursively improves the quality of the belief.

[2] The word feedback denotes that the policy takes the (current) state into account when computing a control input.

5 Reinforcement learning under uncertainty

In the previous sections, we have seen how to reason about uncertainty in sequential decision-making when the MDP that models the system is known, and when this model exhibits additional uncertainty. When the dynamics of the MDP are unknown, we may resort to reinforcement learning (RL) algorithms, which can compute policies through experiences [146]. In this case, we typically see the problem as a sequence of interactions between an agent and an environment, as Fig. 4 demonstrates. In each episode, the agent performs a sequence of actions, and each action yields a corresponding reward.

An RL agent must explore the environment to find a policy that yields the maximum expected return.[3] As the agent collects experiences, it can update its policy. A classical example is the Q-learning algorithm [161], which learns action-values $Q(s,a)$ that indicate the value of executing action a in state s. An RL agent typically requires some form of exploration, and the Q-learning algorithm follows an ϵ-greedy policy. Upon visiting a state s_t at time step t, the agent takes with probability $1-\epsilon$ an action a_t that is chosen greedily according to the current value estimates, and with probability ϵ samples a random action:

$$a_t = \begin{cases} \arg\max_{a \in A} Q(s_t, a) & \text{if } \sim [0,1] > \epsilon \\ \sim U(A) & \text{otherwise,} \end{cases}$$

where U denotes a uniform distribution. After executing action a_t in state s_t the agent receives a reward r_t and observes the next state s_{t+1}, so it updates the state action value:

$$Q(s_t, a_t) \leftarrow (1-\alpha) Q(s_t, a_t) + \alpha \left[r_t + \gamma \max_{a' \in A} Q(s_{t+1}, a') \right],$$

where α is a learning rate.

Considerable advances have been made in RL by applying function approximation to estimate the action value or to represent the agent's policy [102, 131].

Following this simple but powerful framework, RL has shown promising results [132]. Nevertheless, it is still challenging to employ such methods in real-world applications [56]. Since RL typically makes no assumption about the environment, the agent often relies on random exploration to learn a policy in a trial-and-error fashion. However, naive exploration, such as the ϵ-greedy exploration used in Q-learning, may require excessively many interactions with the environment, and such randomized exploration can be detrimental for real-world applications, since it may lead to undesirable outcomes.

[3] Recall from Sect. 2 that the expected return commonly refers to the expected accumulated reward.

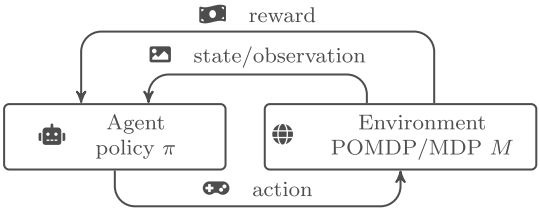


Fig. 4 An agent interacting with its environment

A model-based approach can help us improve the safety and sample efficiency of RL algorithms [104]. One of the key challenges, in this case, is to distinguish aleatoric from epistemic uncertainty. In other words, we want to learn a model from experiences (i.e., reducing epistemic uncertainty) that faithfully captures its stochastic nature (the aleatoric uncertainty). Reasoning about these uncertainties may allow an agent to perform reliably and improve its exploration [50]. For example, an optimistic agent explores regions of the environment with high epistemic uncertainty to improve its sample efficiency [78], while a pessimistic agent may avoid regions with high aleatoric uncertainty to reduce the variance of the returns [54].

In this section, we review how different areas of RL deal with aleatoric and epistemic uncertainty. First, we discuss robust approaches, which aim to ensure that a reasonable performance is always met. Then, we discuss the Bayesian setting, which captures uncertainty via explicit distributions over the underlying (true) model. Finally, we discuss the offline setting, where the uncertainty is irreducible beyond a certain point due to the limited data available.

5.1 Robust RL

A major advantage of reasoning about different types of uncertainty is that we are able to make decisions that are more robust against potential variations and changes in the environment [105]. This is one of the main lines of research in safe RL, where one tries to ensure the agent always maintains a reasonable performance [63]. Such approaches are particularly suitable for situations where data collection is expensive and risky.

To achieve such a goal under aleatoric uncertainty, we can change the objective of the RL agent. Considering that executing a policy π in an MDP induces a distribution over the return G, we may choose to optimize other criteria instead of the mean of the return (Eq. (1)). For instance, we may penalize the variance of the return [54]. We can also aim to maximize the worst-case return [51] or the tail of the return distribution [47], which can be formalized by the conditional value at risk (CVaR) [123]. The α-CVaR can be seen as the mean return of the α trajectories with a lower return.

Robustness can also make an RL agent more reliable in the constrained setting, where the environment is modeled

by a constrained MDP [4]. In this setting, the agent observes, besides the reward, an extra signal, called the cost, that must be kept under a predefined threshold.[4] This cost signal is often used to explicitly model safety requirements, which allows an engineer to easily specify the behavior expected from the agent [84, 127]. In the typical constrained RL setting, the goal of the agent is to maximize the expected return while keeping the expectation of the cost-return (the accumulated cost in an episode) under the given threshold [1]. To bound the cost-return of the worse trajectories, we may constrain the CVaR to remain under the threshold instead of the expectation [166, 167]. From an epistemic uncertainty perspective, we can consider the worst-case expected return of a uMDP. In this case, the RL agent keeps track of a uMDP, and it can compute a policy using a pessimistic (max-min) approach (Eq. (2)).

We remark that the use of a worst-case or adversarial approach may lead to overly conservative policies. In this case, approaches such as the optimization of the CVaR may provide mechanisms for a finer balance between the risks and performance. For instance, we may choose $\alpha = 1$ to recover a risk-neutral approach, while by setting α closer to 0, we get a worst-case perspective.

In deep RL, there are different approaches to make a policy more robust, such as increasing the policy's entropy [58], or using adversarial training, which can generate policies more robust against observation perturbations [112] or actuator perturbations [148].

In cases where certain (catastrophic) events must be avoided, a robust approach may be insufficient to describe the user's preferences. Recently, a number of approaches from the formal methods community consider a so-called shield that blocks certain actions that carry the risk of violating a given safety property [3, 79]. These approaches have also been extended to deep RL and partially observable environments, showcasing the robustness of the obtained policies as well as an improvement of the convergence rate [36].

5.2 Bayesian RL

In many applications, we already have some data or some prior knowledge from an expert, which may be used to infer a distribution over the underlying MDP. This distribution can be represented by a distribution over the parameters of the MDP. Such a distribution can be seen as a prior, which yields a Bayesian-Adaptive MDP (BAMDP) [64, 158], where the state space is augmented with a belief over the underlying MDP. Thus, as the agent interacts with the environment, the belief over the underlying model is updated.

BAMDPs may be used to devise efficient exploration strategies. In theory, a BAMDP can be described as a POMDP [55] where the unknown parameters of the underlying MDP are seen as hidden continuous variables. This allows us to find an optimal trade-off between exploration and exploitation. However, solving these POMDPs is infeasible due to their excessive size, since we must keep a belief over the distribution of each unknown parameter of the underlying MDP. To make the problem more tractable, we may consider other types of prior knowledge. For example, we may assume the system is modeled by a factored MDP, where the state of the MDP is described by a set of features, and the dynamics of the features can be compactly represented by a dynamic Bayesian network (DBN) [29]. In this case, we can assume a prior over the structure of the DBN [125].

In the case of partial observability, a Bayesian approach has also been considered, modeling the problem as a Bayes-Adaptive POMDP [124]. Similarly to the MDP setting, we can also exploit the structure of the underlying system to find more scalable algorithms [85].

Naturally, there are intersections between Bayesian and robust RL. For instance, a Bayesian approach can be used to construct uncertainty sets tighter than the usual norms, resulting in less conservative policies [128]. As another example, we can change the objective of the BAMDP to maximize the CVaR of the return instead of the expectation [122].

Bayesian methods have also been used in deep RL. For example, to track the uncertainty around the action values and improve the exploration of deep RL methods [16] or to reduce the variance of the returns [53]. Furthermore, in constrained RL a Bayesian world model has been used to allow an agent to explore the environment optimistically with respect to the reward function and pessimistically with respect to the safety constraints [13].

5.3 Offline RL

In offline RL, the agent only has access to historical data previously collected [96]. We call the decision mechanism used to collect such data the behavior policy. Offline RL poses a particular challenge since the agent does not receive any feedback from the environment, making it susceptible to overestimation errors [88]. Moreover, restricted data renders the handling of uncertainty a major challenge for offline RL, as it impairs the ability of the agent of reducing its epistemic uncertainty [155]. In online RL, the agent has the ability to reduce the epistemic uncertainty by interacting with the environment. In offline RL, this ability largely depends on the quantity and coverage of the data available [155]. Two main approaches exist to mitigate such issues [80]. First, we may constrain the new policy to stay close to the behavior policy, and second, we may penalize uncertain parts of the state space. Such approaches may lead to sufficient robustness against epistemic uncertainty.

[4] Notice that the cost has a semantic difference from a negative reward, so it cannot be easily combined with the reward into a scalarized reward.

To evaluate the reliability of offline RL algorithms, we can compare the performance of the policy computed with the performance of the behavior policy. A reliable algorithm[5] has a high probability of returning a policy that outperforms the behavior policy [152]. To achieve that goal, we may augment the reward function of the estimated model to penalize states that are less present in the data [113]. Alternatively, we can bootstrap the behavior policy in states with fewer visits [92, 107]. In this setting, we can also exploit the structure of towards higher sample efficiency [134, 135]. Finally, we can use an estimate of the behavior policy to reliably compute new policies when the behavior policy is unknown [133]. All of the above methods assume a fully observable environment (i.e., MDP). Recent work extended [92] to partially observable environments (POMDPs) under certain assumptions [136].

Finally, we can also consider risk-averse methods in offline settings. For instance, we may compute policies maximizing the CVaR instead of the expected return [156], or the use of robust MDPs [110].

6 Challenges and perspectives

In this section, we discuss important challenges to the research directions discussed above. In particular, we identify and summarize six key challenges and provide an outlook on potential future research directions.

Challenge 1: Mixing uncertainty types

Classical models for decision-making often focus on one particular type of uncertainty while making strong assumptions about others. Developing decision-making approaches with models that faithfully and efficiently reason over different (and possibly dependent) types of uncertainty is crucial for developing reliable AI systems.

For example, recall from Sect. 3.1 the assumption for uMDPs that the underlying graph is known, i.e., the uncertainty is continuous over the transition probabilities only. MEMDPs lift this assumption by allowing for different underlying graphs, but these models are still understudied to date. More generally, we wish to study richer types of uncertainty sets that are capable of combining continuous and discrete uncertainty types.

Another assumption discussed in Sect. 3.1 is the rectangular assumption for uMDPs, which states that uncertainties between state-action pairs are independent [162]. This assumption allows for tractable solution methods, but is unrealistic in many practical scenarios, making solutions more conservative. Thus, we believe that lifting such assumptions while preserving tractability is key to improving the quality of solution methods.

In continuous-state and -action models, most research has considered models with either aleatoric or epistemic uncertainty, but not with both types at the same time. One recent exception is the work in [22], but the resulting abstraction method is computationally expensive. Thus, we see potential for developing more efficient methods that are able to faithfully reason over mixed uncertainty types.

Challenge 2: Sensitivity analysis in uncertainty models

A natural question in all uncertainty models is from where these uncertainty sets originate. While we have discussed a number of approaches for learning uncertainty sets, see for instance Sect. 3.2 and the approaches in [14, 144], there is still an abundance of open research questions in this domain. For example, assume that we are learning an MDP by interacting with an environment in an RL setting, and we formalize the learned model as a uMDP. By interacting further with the environment, we may naturally reduce the size of the uncertainty sets, thus reducing the epistemic uncertainty. To facilitate this learning process, an important question is what policy we should use to explore the environment. An optimal exploration policy should, for instance, maximizes the improvement in the worst-case expected return in Eq. (2). To find that policy, we essentially wish to perform a *sensitivity analysis* on the constraints that define the uncertainty sets of the uMDP. Intuitively, this allows us to answer questions such as: *"When sampling transition X once more, what change can we expect in the uncertainty set associated with that transition in the uMDP?"* Similarly, starting from a concrete MDP, we can ask ourselves: *"How robust can we make this model (by arbitrarily adding uncertainty in transition probabilities) while still satisfying some property of interest?"* Developing principled and rigorous methods that can be used to answer such questions is a promising direction for future research.

Challenge 3: Incorporating prior knowledge

Another aspect is how to incorporate prior knowledge in uncertainty models. For instance, we might be able to query experts [9] or ask for demonstrations [116]. Such prior knowledge naturally gives rise to a distribution over models (similar to the Bayesian-Adaptive approaches discussed in Sect. 5.2) rather than a family of models (as is done with an uMDP). Similarly, other papers have considered prior distributions over MDPs [17] and CTMCs [20]. A common problem is then to obtain a solution *"that is robust against (for example) 99% probability mass of the distribution."* Such an approach generally yields less conservative solutions than purely robust approaches, but determining what 1% of the distribution should be disregarded can be extremely difficult [34]. Thus,

[5] In the literature, such approaches are referred to as *safe policy improvement*.

a key challenge is how to exploit prior distributions over models to obtain solutions that are less conservative but still carry rigorous robustness guarantees.

Challenge 4: High-dimensional state and action spaces

Dealing with high-dimensional states and actions has been identified as a critical challenge in RL [56]. Generally, the state space explosion is a well-known problem in formal verification [48], also referred to as the curse of dimensionality [24]. Naturally, this challenge is relevant to all listed approaches for uncertainty models in this paper. In particular, many approaches for verifying dynamical models against complex temporal logic specifications employ finite abstractions. Naive abstractions are inherently subject to exponential complexity in the dimension of the continuous state and the resolution of the partitioning. To mitigate complexity issues, adaptive discretization procedures [141] and iterative abstraction refinement schemes [18, 19] have been developed. Despite these advances, applying abstraction techniques to high-dimensional models (e.g., above 6-dimensional state spaces) and specifications that require fine-grained partitions remains challenging. One potential direction is to leverage efficient tools from motion and path planning to compute candidate policies for the desired specification on the dynamical model. By generating a finite abstraction of only the portion of the continuous state space that is relevant under the candidate policy, one can then verify in advance whether the specification is indeed satisfied.

Challenge 5: Adapting to changing distributions

As we mentioned before, in many scenarios, the dynamics of the environment are not stationary and may change in different ways. For instance, the components of a robot degrade over its lifetime. Thus, a policy that was optimal initially might become sub-optimal as the motors of the robot lose efficiency. Similar phenomena may happen after long periods of use, as the motors of the robot start overheating. In practice, the dynamics of this system are drifting. There are also cases where the dynamics of the system change suddenly. For example, an autonomous vehicle might need to adapt quickly to new conditions when it starts raining. Furthermore, in a multi-agent setting, the environment becomes non-stationary due to the (potentially adversary) behavior of other agents. In this case, as the remaining agents change their behaviors, the dynamics of the environment change accordingly from the perspective of the ego agent.

Using a model-based perspective with uncertainty models can be helpful in detecting such changes in the environment, and might allow the agent to quickly adapt to the new dynamics without having to compute a new policy from scratch [2]. For instance, if we have learned an uMDP that does not agree with the dynamics of the latest trajectories, we might consider enlarging the uncertainty set. A particular challenge in this situation is to distinguish the aleatoric and epistemic uncertainty. A key question is then: *"How many times must the agent observe an unlikely trajectory to conclude that the dynamics of the environment have changed?"*

Similarly, approaches for decision-making under uncertainty that rely on sampling techniques, e.g., [14, 19], generally require the underlying stochastic process to be i.i.d. Dropping these (and related) assumptions is an important challenge for further research.

Challenge 6: Partial observability

Finally, addressing all of the above challenges under partial observability is another challenge on its own. As we have seen, POMDPs and POSGs, as well as dynamical models with partial observability, have been widely studied so far. While there has been significant progress in the last years on solving such models, there are still major scalability issues. For example, problem settings with additional uncertainty, particularly epistemic uncertainty, are significantly understudied. A few exceptions exist, see Sect. 3.1 and, e.g., the approaches in [52, 143]. Yet, both the theoretical and practical implications, in particular for POSGs, of adding another type of uncertainty are, to the best of our knowledge, not known so far. Developing rigorous and tractable methods for decision-making in such partially observable models with additional uncertainty remains an open challenge.

7 Conclusion

This paper has provided an overview of various formal models that exhibit different types of uncertainty. We have highlighted the most common solution approaches, identified some of their shortcomings, and concluded by presenting a number of key challenges regarding decision-making under uncertainty. We sincerely hope this paper can inspire future research in this important direction.

Funding This work was funded by the ERC Starting Grant 101077178 (DEUCE), and the NWO grants NWA.1160.18.238 (PrimaVera) and OCENW.KLEIN.187 (Provably Correct Policies for Uncertain POMDPs).

References

1. Achiam, J., Held, D., Tamar, A., Abbeel, P.: Constrained policy optimization. In: ICML. Proceedings of Machine Learning Research, vol. 70, pp. 22–31. PMLR, mlr.press (2017)
2. Alegre, L.N., Bazzan, A.L.C., da Silva, B.C.: Minimum-delay adaptation in non-stationary reinforcement learning via online high-confidence change-point detection. In: AAMAS, pp. 97–105. ACM, New York (2021)
3. Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., Topcu, U.: Safe reinforcement learning via shielding. In: AAAI, pp. 2669–2678. AAAI Press, Menlo Park (2018)
4. Altman, E.: Constrained Markov Decision Processes: Stochastic Modeling. Routledge, London (1999)
5. Alur, R., Henzinger, T.A., Lafferriere, G., Pappas, G.J.: Discrete abstractions of hybrid systems. Proc. IEEE **88**(7), 971–984 (2000)
6. Amato, C.: Decision-making under uncertainty in multi-agent and multi-robot systems: Planning and learning. In: IJCAI, pp. 5662–5666 (2018). ijcai.org
7. Amato, C., Bernstein, D.S., Zilberstein, S.: Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs. Auton. Agents Multi-Agent Syst. **21**(3), 293–320 (2010)
8. Anderson, B.D., Moore, J.B.: Optimal control: linear quadratic methods. Courier Corporation, Mineola, New York (2007)
9. Andrés, I., de Barros, L.N., Mauá, D.D., Simão, T.D.: When a robot reaches out for human help. In: IBERAMIA. Lecture Notes in Computer Science, vol. 11238, pp. 277–289. Springer, Berlin (2018)
10. Antsaklis, P.J., Michel, A.N.: Linear Systems. Birkhäuser, Basel (2006)
11. Argote, L.: Input uncertainty and organizational coordination in hospital emergency units. Administrative science quarterly, 420–434 (1982)
12. Arrowsmith, D.K., Place, C.M., Place, C., et al.: An introduction to dynamical systems. Cambridge University Press, Cambridge (1990)
13. As, Y., Usmanova, I., Curi, S., Krause, A.: Constrained policy optimization via bayesian world models. In: ICLR (2022). OpenReview.net
14. Ashok, P., Kretínský, J., Weininger, M.: PAC statistical model checking for markov decision processes and stochastic games. In: CAV (1). Lecture Notes in Computer Science, vol. 11561, pp. 497–519. Springer, Berlin (2019).
15. Åström, K.J., Murray, R.M.: Feedback systems: an introduction for scientists and engineers. Princeton University Press, Princeton (2010)
16. Azizzadenesheli, K., Brunskill, E., Anandkumar, A.: Efficient exploration through bayesian deep Q-networks. In: ITA, pp. 1–9. IEEE, ieee.org (2018)
17. Badings, T., Cubuktepe, M., Jansen, N., Junges, S., Katoen, J.P., Topcu, U.: Scenario-based verification of uncertain parametric MDPs. International Journal on Software Tools for Technology Transfer, 1–17 (2022)
18. Badings, T., Romao, L., Abate, A., Parker, D., Poonawala, H.A., Stoelinga, M., Jansen, N.: Robust Control for Dynamical Systems with Non-Gaussian Noise via Formal Abstractions. J. Artif. Intell. Res. **76**, 341–391 (2023)
19. Badings, T.S., Abate, A., Jansen, N., Parker, D., Poonawala, H.A., Stoelinga, M.: Sampling-based robust control of autonomous systems with non-gaussian noise. In: AAAI, pp. 9669–9678. AAAI Press, Menlo Park (2022)
20. Badings, T.S., Jansen, N., Junges, S., Stoelinga, M., Volk, M.: Sampling-Based Verification of CTMCs with Uncertain Rates. Preprint arXiv:2205.08300 (2022)
21. Badings, T.S., Jansen, N., Poonawala, H.A., Stoelinga, M.: Filter-based abstractions with correctness guarantees for planning under uncertainty. Preprint arXiv:2103.02398 (2021)
22. Badings, T.S., Romao, L., Abate, A., Jansen, N.: Probabilities are not enough: Formal controller synthesis for stochastic dynamical models with epistemic uncertainty. In: AAAI (2023)
23. Baier, C., Katoen, J.: Principles of model checking. MIT Press, Cambridge (2008)
24. Bellman, R.: Dynamic programming. Science **153**(3731), 34–37 (1966)
25. Belta, C., Yordanov, B., Gol, E.A.: Formal methods for discrete-time dynamical systems, vol. 15. Springer, Berlin (2017)
26. Ben-Tal, A., Ghaoui, L.E., Nemirovski, A.: Robust Optimization. Princeton Series in Applied Mathematics, vol. 28. Princeton University Press, Princeton (2009)
27. Bertsimas, D., Brown, D.B., Caramanis, C.: Theory and applications of robust optimization. SIAM Rev. **53**(3), 464–501 (2011)
28. Blondel, V.D., Tsitsiklis, J.N.: A survey of computational complexity results in systems and control. Autom. **36**(9), 1249–1274 (2000)
29. Boutilier, C., Dearden, R., Goldszmidt, M.: Stochastic dynamic programming with factored representations. Artif. Intell. **121**(1–2), 49–107 (2000)
30. Boyd, S.P., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2014)
31. Brin, M., Stuck, G.: Introduction to dynamical systems. Cambridge University Press, Cambridge (2002)
32. Bry, A., Roy, N.: Rapidly-exploring random belief trees for motion planning under uncertainty. In: ICRA, pp. 723–730. IEEE, ieee.org (2011)
33. Burns, B., Brock, O.: Sampling-based motion planning with sensing uncertainty. In: ICRA, pp. 3313–3318. IEEE, ieee.org (2007)
34. Campi, M.C., Garatti, S.: Introduction to the scenario approach. SIAM, Philadelphia (2018)
35. Carr, S., Jansen, N., Bharadwaj, S., Spaan, M.T.J., Topcu, U.: Safe policies for factored partially observable stochastic games. In: Robotics: Science and Systems (2021)
36. Carr, S., Jansen, N., Junges, S., Topcu, U.: Safe reinforcement learning via shielding under partial observability. In: AAAI (2023)
37. Carr, S., Jansen, N., Topcu, U.: Verifiable rnn-based policies for pomdps under temporal logic constraints. In: IJCAI, pp. 4121–4127 (2020). ijcai.org
38. Carr, S., Jansen, N., Topcu, U.: Task-aware verifiable rnn-based policies for partially observable markov decision processes. J. Artif. Intell. Res. **72**, 819–847 (2021)
39. Carr, S., Jansen, N., Wimmer, R., Serban, A.C., Becker, B., Topcu, U.: Counterexample-guided strategy improvement for pomdps using recurrent neural networks. In: IJCAI, pp. 5532–5539 (2019). ijcai.org
40. Cauchi, N., Abate, A.: Stochy: Automated verification and synthesis of stochastic processes. In: TACAS (2). Lecture Notes in Computer Science, vol. 11428, pp. 247–264. Springer, Berlin (2019)
41. Chades, I., Carwardine, J., Martin, T.G., Nicol, S., Sabbadin, R., Buffet, O.: MOMDPs: A Solution for Modelling Adaptive Management Problems. In: AAAI, pp. 267–273. AAAI Press, Menlo Park (2012)
42. Chatterjee, K., Chmelík, M., Karkhanis, D., Novotný, P., Royer, A.: Multiple-environment markov decision processes: Efficient analysis and applications. In: ICAPS, pp. 48–56. AAAI Press, Menlo Park (2020)
43. Chatterjee, K., Chmelik, M., Tracol, M.: What is decidable about partially observable markov decision processes with ω-regular objectives. J. Comput. Syst. Sci. **82**(5), 878–911 (2016)
44. Chen, M., Frazzoli, E., Hsu, D., Lee, W.S.: POMDP-lite for robust robot planning under uncertainty. In: ICRA, pp. 5427–5433. IEEE, ieee.org (2016)

45. Chen, T., Forejt, V., Kwiatkowska, M.Z., Parker, D., Simaitis, A.: Prism-games: A model checker for stochastic multi-player games. In: TACAS. LNCS, vol. 7795, pp. 185–191. Springer, Berlin (2013)
46. Cheung, W.C., Simchi-Levi, D., Zhu, R.: Reinforcement learning for non-stationary markov decision processes: The blessing of (more) optimism. In: ICML. Proceedings of Machine Learning Research, vol. 119, pp. 1843–1854. PMLR, mlr.press (2020)
47. Chow, Y., Ghavamzadeh, M., Janson, L., Pavone, M.: Risk-constrained reinforcement learning with percentile risk criteria. J. Mach. Learn. Res. **18**, 167:1–167:51 (2018)
48. Clarke, E.M.: Model checking – my 27-year quest to overcome the state explosion problem. In: NASA Formal Methods, NASA Conference Proceedings, vol. NASA/CP–2009–215407, p. 1 (2009F)
49. Clarke, E.M., Henzinger, T.A., Veith, H., Bloem, R.: Handbook of Model Checking. Springer, Berlin (2018)
50. Clements, W.R., Robaglia, B., Delft, B.V., Slaoui, R.B., Toth, S.: Estimating risk and uncertainty in deep reinforcement learning. Preprint arXiv:1905.09638 (2019)
51. Coraluppi, S.P., Marcus, S.I.: Risk-sensitive and minimax control of discrete-time, finite-state markov decision processes. Autom. **35**(2), 301–309 (1999)
52. Cubuktepe, M., Jansen, N., Junges, S., Marandi, A., Suilen, M., Topcu, U.: Robust Finite-State Controllers for Uncertain POMDPs. In: AAAI, pp. 11792–11800. AAAI Press, Menlo Park (2021)
53. Depeweg, S., Hernández-Lobato, J.M., Doshi-Velez, F., Udluft, S.: Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In: ICML. Proceedings of Machine Learning Research, vol. 80, pp. 1192–1201. PMLR, mlr.press (2018)
54. Di Castro, D., Tamar, A., Mannor, S.: Policy gradients with variance related risk criteria. In: ICML. icml.cc / Omnipress, Madison (2012)
55. Duff, M.O.: Optimal learning: Computational procedures for Bayes-adaptive Markov decision processes. Ph.D. thesis, University of Massachusetts Amherst (2002)
56. Dulac-Arnold, G., Levine, N., Mankowitz, D.J., Li, J., Paduraru, C., Gowal, S., Hester, T.: Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. Mach. Learn. **110**(9), 2419–2468 (2021)
57. Emery-Montemerlo, R., Gordon, G.J., Schneider, J.G., Thrun, S.: Approximate solutions for partially observable stochastic games with common payoffs. In: AAMAS, pp. 136–143. IEEE Comput. Soc., Los Alamitos (2004)
58. Eysenbach, B., Levine, S.: Maximum entropy RL (provably) solves some robust RL problems. In: ICLR (2022). OpenReview.net
59. Fan, C., Qin, Z., Mathur, U., Ning, Q., Mitra, S., Viswanathan, M.: Controller synthesis for linear system with reach-avoid specifications. IEEE Trans. Autom. Control **67**(4), 1713–1727 (2022)
60. Fisac, J.F., Akametalu, A.K., Zeilinger, M.N., Kaynama, S., Gillula, J.H., Tomlin, C.J.: A general safety framework for learning-based control in uncertain robotic systems. IEEE Trans. Autom. Control **64**(7), 2737–2752 (2019)
61. Fox, C.R., Ülkümen, G.: Distinguishing two dimensions of uncertainty. Fox, Craig R. and Gülden Ülkümen (2011), "Distinguishing Two Dimensions of Uncertainty". In: Brun, W., Kirkebøen, G., Montgomery, H. (eds.) Essays in Judgment and Decision Making. Universitetsforlaget, Oslo (2011)
62. Gajane, P., Ortner, R., Auer, P.: A sliding-window algorithm for markov decision processes with arbitrarily changing rewards and transitions. Preprint arXiv:1805.10066 (2018)
63. García, J., Fernández, F.: A comprehensive survey on safe reinforcement learning. J. Mach. Learn. Res. **16**, 1437–1480 (2015)
64. Ghavamzadeh, M., Mannor, S., Pineau, J., Tamar, A.: Bayesian Reinforcement Learning: A Survey. Found. Trends Mach. Learn. **8**(5–6), 359–483 (2015)
65. Girard, A., Pappas, G.J.: Approximation metrics for discrete and continuous systems. IEEE Trans. Autom. Control **52**(5), 782–798 (2007)
66. Givan, R., Leach, S.M., Dean, T.L.: Bounded-parameter markov decision processes. Artif. Intell. **122**(1–2), 71–109 (2000)
67. Goodess, C.M., Hall, J., Best, M., Betts, R., Cabantous, L., Jones, P.D., Kilsby, C.G., Pearman, A., Wallace, C.: Climate scenarios and decision making under uncertainty. Built Environ. **33**(1), 10–30 (2007)
68. Goyal, V., Grand-Clement, J.: Robust Markov Decision Process: Beyond Rectangularity (2020)
69. Hansen, E.A.: An Improved Policy Iteration Algorithm for Partially Observable MDPs. In: NIPS, pp. 1015–1021. MIT Press, Cambridge (1997)
70. Hansen, E.A., Bernstein, D.S., Zilberstein, S.: Dynamic programming for partially observable stochastic games. In: AAAI, pp. 709–715. AAAI Press / The MIT Press, Menlo Park / Cambridge (2004)
71. Hausknecht, M.J., Stone, P.: Deep recurrent q-learning for partially observable mdps. In: AAAI Fall Symposia, pp. 29–37. AAAI Press, Menlo Park (2015)
72. Hoeffding, W.: Probability Inequalities for Sums of Bounded Random Variables. J. Am. Stat. Assoc. **58**(301), 13–30 (1963)
73. Horák, K., Bosanský, B., Pechoucek, M.: Heuristic Search Value Iteration for One-Sided Partially Observable Stochastic Games. In: AAAI, pp. 558–564. AAAI Press, Menlo Park (2017)
74. Horák, K., Zhu, Q., Bosanský, B.: Manipulating Adversary's Belief: A Dynamic Game Approach to Deception by Design for Proactive Network Security. In: GameSec. LNCS, vol. 10575, pp. 273–294. Springer, Berlin (2017)
75. Hüllermeier, E., Waegeman, W.: Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. Mach. Learn. **110**(3), 457–506 (2021)
76. Itoh, H., Nakamura, K.: Partially observable Markov decision processes with imprecise parameters. Artif. Intell. **171**(8), 453–490 (2007)
77. Jaeger, M., Bacci, G., Bacci, G., Larsen, K.G., Jensen, P.G.: Approximating euclidean by imprecise markov decision processes. In: ISoLA (1). Lecture Notes in Computer Science, vol. 12476, pp. 275–289. Springer, Berlin (2020)
78. Jaksch, T., Ortner, R., Auer, P.: Near-optimal regret bounds for reinforcement learning. J. Mach. Learn. Res. **11**, 1563–1600 (2010)
79. Jansen, N., Könighofer, B., Junges, S., Serban, A., Bloem, R.: Safe reinforcement learning using probabilistic shields (invited paper). In: CONCUR, LIPIcs, vol. 171, pp. 3:1–3:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Wadern (2020)
80. Jin, Y., Yang, Z., Wang, Z.: Is pessimism provably efficient for offline RL? In: ICML. Proceedings of Machine Learning Research, vol. 139, pp. 5084–5096. PMLR, mlr.press (2021)
81. Junges, S., Jansen, N., Wimmer, R., Quatmann, T., Winterer, L., Katoen, J., Becker, B.: Finite-State Controllers of POMDPs using Parameter Synthesis. In: UAI, pp. 519–529. AUAI Press, auai.org (2018)
82. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artif. Intell. **101**(1–2), 99–134 (1998)
83. Kalman, R.E.: A new approach to linear filtering and prediction problems. J. Fluids Eng. **82**(1), 35–45 (1960)
84. Kamran, D., Simão, T.D., Yang, Q., Ponnambalam, C.T., Fischer, J., Spaan, M.T.J., Lauer, M.: A modern perspective on safe automated driving for different traffic dynamics using constrained reinforcement learning. In: ITSC, pp. 4017–4023. IEEE, ieee.org (2022)

85. Katt, S., Oliehoek, F.A., Amato, C.: Bayesian Reinforcement Learning in Factored POMDPs. In: AAMAS, pp. 7–15. IFAAMAS, ifaamas.org (2019)
86. Kochenderfer, M.J.: Decision Making Under Uncertainty: Theory and Application. MIT Press, Cambridge (2015)
87. Kress-Gazit, H., Fainekos, G.E., Pappas, G.J.: Temporal-Logic-Based Reactive Mission and Motion Planning. IEEE Trans. Robot. **25**(6), 1370–1381 (2009)
88. Kumar, A., Zhou, A., Tucker, G., Levine, S.: Conservative q-learning for offline reinforcement learning. In: NeurIPS (2020)
89. Kumar, A., Zilberstein, S.: Dynamic Programming Approximations for Partially Observable Stochastic Games. In: FLAIRS Conference. AAAI Press, Menlo Park (2009)
90. Kwiatkowska, M., Norman, G., Parker, D., Santos, G.: Prism-games 3.0: Stochastic game verification with concurrency, equilibria and time. In: CAV. Lecture Notes in Computer Science, vol. 2, pp. 475–487. Springer, Berlin (2020). 12225
91. Lahijanian, M., Andersson, S.B., Belta, C.: Formal verification and synthesis for discrete-time stochastic systems. IEEE Trans. Autom. Control **60**(8), 2031–2045 (2015)
92. Laroche, R., Trichelair, P., des Combes, R.T.: Safe policy improvement with baseline bootstrapping. In: ICML. Proceedings of Machine Learning Research, vol. 97, pp. 3652–3661. PMLR, mlr.press (2019)
93. Lathi, B.P., Green, R.A.: Signal processing and linear systems, vol. 2. Oxford University Press, Oxford (1998)
94. Lavaei, A., Soudjani, S., Abate, A., Zamani, M.: Automated verification and synthesis of stochastic hybrid systems: A survey. Preprint arXiv:2101.07491 (2021)
95. Lavaei, A., Soudjani, S., Frazzoli, E., Zamani, M.: Constructing MDP Abstractions Using Data with Formal Guarantees. arXiv e-prints pp. arXiv–2206 (2022)
96. Levine, S., Kumar, A., Tucker, G., Fu, J.: Offline reinforcement learning: Tutorial, review, and perspectives on open problems. Preprint arXiv:2005.01643 (2020)
97. Liu, J.S., Chen, R.: Sequential monte carlo methods for dynamic systems. J. Am. Stat. Assoc. **93**(443), 1032–1044 (1998)
98. Madani, O., Hanks, S., Condon, A.: On the undecidability of probabilistic planning and related stochastic optimization problems. Artif. Intell. **147**(1–2), 5–34 (2003)
99. Mallik, K., Schmuck, A., Soudjani, S., Majumdar, R.: Compositional synthesis of finite-state abstractions. IEEE Trans. Autom. Control **64**(6), 2629–2636 (2019)
100. Mannor, S., Simester, D., Sun, P., Tsitsiklis, J.N.: Bias and Variance Approximation in Value Function Estimates. Manag. Sci. **53**(2), 308–322 (2007)
101. Meuleau, N., Kim, K., Kaelbling, L.P., Cassandra, A.R.: Solving POMDPs by Searching the Space of Finite Policies. In: UAI, pp. 417–426. Morgan Kaufmann, San Mateo (1999)
102. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M.A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nat. **518**(7540), 529–533 (2015)
103. Modares, H.: Data-driven safe control of linear systems under epistemic and aleatory uncertainties. Preprint arXiv:2202.04495 (2022)
104. Moerland, T.M., Broekens, J., Jonker, C.M.: Model-based reinforcement learning: A survey. Preprint arXiv:2006.16712 (2020)
105. Moos, J., Hansel, K., Abdulsamad, H., Stark, S., Clever, D., Peters, J.: Robust reinforcement learning: A review of foundations and recent advances. Mach. Learn. Knowl. Extr. **4**(1), 276–315 (2022)
106. Munos, R.: From bandits to monte-carlo tree search: The optimistic principle applied to optimization and planning. Found. Trends Mach. Learn. **7**(1), 1–129 (2014)
107. Nadjahi, K., Laroche, R., des Combes R.T.: Safe policy improvement with soft baseline bootstrapping. In: ECML/PKDD. Lecture Notes in Computer Science, vol. 3, pp. 53–68. Springer, Berlin (2019). 11908
108. Nilim, A., Ghaoui, L.E.: Robust control of markov decision processes with uncertain transition matrices. Oper. Res. **53**(5), 780–798 (2005)
109. Osogami, T.: Robust partially observable markov decision process. In: ICML. JMLR Workshop and Conference Proceedings, vol. 37, pp. 106–115 (2015). JMLR.org
110. Panaganti, K., Xu, Z., Kalathil, D., Ghavamzadeh, M.: Robust reinforcement learning using offline data. Preprint arXiv:2208.05129 (2022)
111. Park, S., Serpedin, E., Qaraqe, K.A.: Gaussian assumption: The least favorable but the most useful [lecture notes]. IEEE Signal Process. Mag. **30**(3), 183–186 (2013)
112. Pattanaik, A., Tang, Z., Liu, S., Bommannan, G., Chowdhary, G.: Robust deep reinforcement learning with adversarial attacks. Preprint arXiv:1712.03632 (2017)
113. Petrik, M., Ghavamzadeh, M., Chow, Y.: Safe policy improvement by minimizing robust baseline regret. In: NIPS, pp. 2298–2306 (2016)
114. Pineau, J., Gordon, G.J., Thrun, S.: Point-based value iteration: An anytime algorithm for pomdps. In: IJCAI, pp. 1025–1032. Morgan Kaufmann, San Mateo (2003)
115. Pnueli, A.: The temporal logic of programs. In: FOCS, pp. 46–57. IEEE Comput. Soc., Los Alamitos (1977)
116. Ponnambalam, C.T., Oliehoek, F.A., Spaan, M.T.J.: Abstraction-guided policy recovery from expert demonstrations. In: ICAPS, pp. 560–568. AAAI Press, Menlo Park (2021)
117. Prentice, S., Roy, N.: The belief roadmap: Efficient planning in linear pomdps by factoring the covariance. In: ISRR. Springer Tracts in Advanced Robotics, vol. 66, pp. 293–305. Springer, Berlin (2007)
118. Puggelli, A., Li, W., Sangiovanni-Vincentelli, A.L., Seshia, S.A.: Polynomial-time verification of PCTL properties of mdps with convex uncertainties. In: CAV. Lecture Notes in Computer Science, vol. 8044, pp. 527–542. Springer, Berlin (2013)
119. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley Series in Probability and Statistics. Wiley, New York (1994)
120. Raskin, J., Sankur, O.: Multiple-environment markov decision processes. In: FSTTCS. LIPIcs, vol. 29, pp. 531–543 Schloss Dagstuhl - Leibniz-Zentrum für Informatik, ??? (2014)
121. Reissig, G., Weber, A., Rungger, M.: Feedback refinement relations for the synthesis of symbolic controllers. IEEE Trans. Autom. Control **62**(4), 1781–1796 (2017)
122. Rigter, M., Lacerda, B., Hawes, N.: Risk-averse bayes-adaptive reinforcement learning. In: NeurIPS, pp. 1142–1154 (2021)
123. Rockafellar, R.T., Uryasev, S.: Optimization of conditional value-at-risk. J. Risk **2**(3), 21–41 (2000)
124. Ross, S., Chaib-draa, B., Pineau, J.: Bayes-Adaptive POMDPs. In: NIPS, pp. 1225–1232. Curran Associates, Red Hook (2007)
125. Ross, S., Pineau, J.: Model-based bayesian reinforcement learning in large structured domains. In: UAI, pp. 476–483. AUAI Press, auai.org (2008)
126. Rostampour, V., Badings, T.S., Scherpen, J.: Demand flexibility management for buildings-to-grid integration with uncertain generation. Energies **13**(24), 6532 (2020)
127. Roy, J., Girgis, R., Romoff, J., Bacon, P., Pal, C.J.: Direct Behavior Specification via Constrained Reinforcement Learning. In: ICML. Proceedings of Machine Learning Research, vol. 162, pp. 18828–18843. PMLR, mlr.press (2022)
128. Russel, R.H., Petrik, M.: Beyond Confidence Regions: Tight Bayesian Ambiguity Sets for Robust MDPs. In: NeurIPS, pp. 7047–7056 (2019)

129. Russell, S.J., Norvig, P.: Artificial Intelligence - A Modern Approach, Third International Edition. Pearson Education, Upper Saddle River (2010)
130. Sarkar, P.: Sequential monte carlo methods in practice. Technometrics **45**(1), 106 (2003)
131. Schulman, J., Moritz, P., Levine, S., Jordan, M.I., Abbeel, P.: High-dimensional continuous control using generalized advantage estimation. In: ICLR (Poster) (2016)
132. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T.P., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D.: Mastering the game of go with deep neural networks and tree search. Nat. **529**(7587), 484–489 (2016)
133. Simão, T.D., Laroche, R., Tachet des Combes, R.: Safe Policy Improvement with an Estimated Baseline Policy. In: AAMAS, pp. 1269–1277. IFAAMAS, ifaamas.org (2020)
134. Simão, T.D., Spaan, M.T.J.: Safe policy improvement with baseline bootstrapping in factored environments. In: AAAI, pp. 4967–4974. AAAI Press, Menlo Park (2019)
135. Simão, T.D., Spaan, M.T.J.: Structure learning for safe policy improvement. In: IJCAI, pp. 3453–3459 (2019). ijcai.org
136. Simão, T.D., Suilen, M., Jansen, N.: Safe Policy Improvement for POMDPs via Finite-State Controllers In: AAAI (2023). Preprint arXiv:2301.04939
137. Smallwood, R.D., Sondik, E.J.: The optimal control of partially observable markov processes over a finite horizon. Oper. Res. **21**(5), 1071–1088 (1973)
138. Smith, R.C.: Uncertainty quantification: theory, implementation, and applications, vol. 12. SIAM, Philadelphia (2013)
139. Sniazhko, S.: Uncertainty in decision-making: A review of the international business literature. Cogent Bus. Manag. **6**(1), 1650692 (2019)
140. Soize, C.: Uncertainty quantification. Springer, Berlin (2017)
141. Soudjani, S.E.Z., Abate, A.: Adaptive and sequential gridding procedures for the abstraction and verification of stochastic processes. SIAM J. Appl. Dyn. Syst. **12**(2), 921–956 (2013)
142. Spaan, M.T.J., Vlassis, N.: Perseus: Randomized Point-based Value Iteration for POMDPs. J. Artif. Intell. Res. **24**, 195–220 (2005)
143. Suilen, M., Jansen, N., Cubuktepe, M., Topcu, U.: Robust Policy Synthesis for Uncertain POMDPs via Convex Optimization. In: IJCAI, pp. 4113–4120 (2020). ijcai.org
144. Suilen, M., Simão, T.D., Parker, D., Jansen, N.: Robust anytime learning of markov decision processes. Preprint arXiv:2205.15827 (2022)
145. Sullivan, T.J.: Introduction to uncertainty quantification, vol. 63. Springer, Berlin (2015)
146. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT press, Cambridge (2018)
147. Tabuada, P.: Verification and Control of Hybrid Systems - A Symbolic Approach. Springer, Berlin (2009)
148. Tan, K.L., Esfandiari, Y., Lee, X.Y., Aakanksha, S.S.: Robustifying reinforcement learning agents via action space adversarial training. In: ACC, pp. 3959–3964. IEEE, ieee.org (2020)
149. Tappler, M., Aichernig, B.K., Bacci, G., Eichlseder, M., Larsen, K.G.: L^*-based learning of markov decision processes (extended version). Form. Asp. Comput. **33**(4–5), 575–615 (2021)
150. Tappler, M., Muskardin, E., Aichernig, B.K., Pill, I.: Active model learning of stochastic reactive systems. In: SEFM. Lecture Notes in Computer Science, vol. 13085, pp. 481–500. Springer, Berlin (2021)
151. Thiebes, S., Lins, S., Sunyaev, A.: Trustworthy artificial intelligence. Electron. Mark. **31**(2), 447–464 (2021)
152. Thomas, P.S., Theocharous, G., Ghavamzadeh, M.: High Confidence Policy Improvement. In: ICML. JMLR Workshop and Conference Proceedings, vol. 37, pp. 2380–2388 (2015). JMLR.org
153. Thrun, S., Burgard, W., Fox, D.: Probabilistic robotics. Intelligent robotics and autonomous agents. MIT Press, Cambridge (2005)
154. Trentelman, H.L., Stoorvogel, A.A., Hautus, M.: Control theory for linear systems. Springer, Berlin (2012)
155. Uehara, M., Sun, W.: Pessimistic model-based offline reinforcement learning under partial coverage. In: ICLR (2022). OpenReview.net
156. Urpí, N.A., Curi, S., Krause, A.: Risk-averse offline reinforcement learning. In: ICLR (2021). OpenReview.net
157. Vaandrager, F.W.: Model learning. Commun. ACM **60**(2), 86–95 (2017)
158. Vlassis, N., Ghavamzadeh, M., Mannor, S., Poupart, P.: Bayesian reinforcement learning. In: Wiering, M.A., van Otterlo, M. (eds.) Reinforcement Learning, Adaptation, Learning, and Optimization, vol. 12, pp. 359–386. Springer, Berlin (2012)
159. Vlassis, N., Littman, M.L., Barber, D.: On the Computational Complexity of Stochastic Controller Optimization in POMDPs. ACM Trans. Comput. Theory **4**(4), 12:1–12:8 (2012)
160. Walraven, E., Spaan, M.T.J.: Point-based value iteration for finite-horizon pomdps. J. Artif. Intell. Res. **65**, 307–341 (2019)
161. Watkins, C.J.C.H.: Learning from delayed rewards. King's College, Cambridge United Kingdom (1989). Ph.D. thesis
162. Wiesemann, W., Kuhn, D., Sim, M.: Distributionally robust convex optimization. Oper. Res. **62**(6), 1358–1376 (2014)
163. Wolff, E.M., Topcu, U., Murray, R.M.: Robust control of uncertain markov decision processes with temporal logic specifications. In: CDC, pp. 3372–3379. IEEE, ieee.org (2012)
164. Wooldridge, M.: The Road to Conscious Machines: The Story of AI. Penguin, Baltimore (2020)
165. Xu, H., Mannor, S.: Distributionally Robust Markov Decision Processes. Math. Oper. Res. **37**(2), 288–300 (2012)
166. Yang, Q., Simão, T.D., Tindemans, S.H., Spaan, M.T.: Safety-constrained reinforcement learning with a distributional safety critic. Machine Learning, 1–29 (2022)
167. Yang, Q., Simão, T.D., Tindemans, S.H., Spaan, M.T.J.: WCSAC: Worst-Case Soft Actor Critic for Safety-Constrained Reinforcement Learning. In: AAAI, pp. 10639–10646. AAAI Press, Menlo Park (2021)
168. Zak, S.H.: Systems and control, vol. 198. Oxford University Press, New York (2003)
169. Zhao, X., Calinescu, R., Gerasimou, S., Robu, V., Flynn, D.: Interval change-point detection for runtime probabilistic model checking. In: ASE, pp. 163–174. IEEE, ieee.org (2020)