

Runtime Safety and Reach-avoid Prediction of Stochastic Systems via Observation-aware Barrier Functions

Shenghua Feng^{1,2}, Jie An^{1,2}, Fanjiang Xu^{1,2}

¹National Key Laboratory of Space Integrated Information System,
Institute of Software Chinese Academy of Sciences, Beijing, China

²University of Chinese Academy of Sciences, Beijing, China

{fengshenghua, anjie, fanjiang}@iscas.ac.cn

Abstract

Stochastic dynamical systems have emerged as fundamental models across numerous application domains, providing powerful mathematical representations for capturing uncertain system behavior. In this paper, we address the problem of runtime safety and reach-avoid probability prediction for discrete-time stochastic systems with online observations, i.e., estimating the probability that the system satisfies a given safety or reach-avoid specification. Unlike traditional approaches that rely solely on offline models, we propose a framework that incorporates real-time observations to dynamically refine probability estimates for safety and reach-avoid events. By introducing observation-aware barrier functions, our method adaptively updates probability bounds as new observations are collected, combining efficient offline computation with online backward iteration. This approach enables rigorous and responsive prediction of safety and reach-avoid probabilities under uncertainty. In addition to the theoretical guarantees, experimental results on benchmark systems demonstrate the practical effectiveness of the proposed method.

Full version — <http://arxiv.org/abs/2511.09192>

Introduction

Stochastic dynamical systems provide robust mathematical frameworks for modeling real-world phenomena under uncertainty. These systems – including Markov decision processes, probabilistic graphical models, and stochastic hybrid automata – are pivotal in various fields, such as reinforcement learning, control theory, physics, signal processing, cryptography, finance, biology, and neuroscience (Bertsekas and Shreve 1996; Steele 2001; Allen 2010). Ensuring reliability and safety in stochastic systems is a significant challenge, especially as these systems operate in increasingly complex and uncertain environments.

Safety and reach-avoid properties form the cornerstone of trustworthy stochastic system operations. Safety probability estimation quantifies the likelihood that a system trajectory remains outside unsafe regions during execution, whereas reach-avoid estimation assesses the probability of successfully reaching a target region without encountering unsafe

states. These estimations are vital for safety-critical control, autonomous systems, robotics, and real-time decision-making (Bertsekas and Shreve 1996; Paul and Baschnagel 2013), where precise risk assessments and robust guarantees are indispensable.

Traditional approaches for safety and reach-avoid predictions, such as stochastic barrier functions, rely on offline computations and predefined uncertainty models to establish probabilistic guarantees (Prajna, Jadbabaie, and Pappas 2004; Feng et al. 2020a; Lechner et al. 2022; Žikelić et al. 2023). These offline methodologies, however, fail to capitalize on real-time information like system observations or environmental dynamics, often leading to conservative or outdated predictions.

In this paper, we introduce a novel framework for runtime safety and reach-avoid prediction in discrete-time stochastic systems. Our method integrates real-time discrete observations gathered during system execution, enabling dynamic refinement of probability estimates. We utilize observation-aware barrier functions – extensions of classical barrier certificates that adaptively update probability bounds in response to new observational data – allowing our approach to rigorously and effectively reflect the evolving state of the system. The proposed framework adopts a hybrid offline-online computational strategy. The offline phase involves the efficient synthesis of barrier functions through semidefinite programming techniques, addressing the computationally intensive aspects of the prediction process beforehand. This preparation greatly reduces the online computational burden. Subsequently, during system operation, the online phase leverages rapid backward iteration updates whenever new discrete observations become available. These updates dynamically recalibrate the barrier functions, swiftly refining the safety and reach-avoid probability estimates to align with the most current state information.

Contributions. Our main contributions are as follows:

- Developing a framework for runtime safety and reach-avoid probability prediction using observation-aware barrier functions to incorporate online observations.
- Providing theoretical guarantees on predicted probability bounds and introducing efficient runtime prediction algorithms.
- Demonstrating significant improvements in adaptivity

and accuracy over traditional methods through experiments on benchmark systems, underscoring the effectiveness of observation-aware verification in reliable stochastic control.

Additional materials, benchmarks, source code, and detailed proofs are provided in the full version of this paper (Feng, An, and Xu 2025).

Problem Formulation

Let \mathbb{R} , $\mathbb{R}_{\geq 0}$, and \mathbb{N} be the reals, nonnegative reals, and natural numbers, respectively. We consider a discrete-time stochastic dynamical system defined by:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \omega_t), \quad t \in \mathbb{N}, \quad (1)$$

where $\mathbf{x}_t \in \mathcal{X} \subseteq \mathbb{R}^n$ denotes the state of the system at time t , $\mathbf{u}_t \in \mathcal{U} \subseteq \mathbb{R}^m$ is the control input, and $\omega_t \in \mathcal{N} \subseteq \mathbb{R}^q$ represents a random disturbance. The disturbance sequence $\{\omega_t\}_{t \in \mathbb{N}}$ is assumed to be independent and identically distributed (i.i.d.) with a known probability distribution. The control input is determined by a policy $\pi : \mathcal{X} \rightarrow \mathcal{U}$, which is a measurable mapping assigning a state \mathbf{x}_t to the control $\mathbf{u}_t = \pi(\mathbf{x}_t)$. The transition function f can be nonlinear, fully characterizing the system dynamics.

Given an initial state \mathbf{x}_0 and a control policy π , a trajectory of the system is represented by the sequence $(\mathbf{x}_0, \mathbf{x}_1, \dots)$ satisfying $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \pi(\mathbf{x}_t), \omega_t)$, with ω_t drawn randomly according to its distribution. The resulting trajectory is denoted by $\{X_t^{\mathbf{x}_0}\}_{t \in \mathbb{N}}$. This induces a probability measure $\mathbb{P}_{\mathbf{x}_0}^\pi$ over all possible trajectories, which we will reference without superscripts or subscripts when the context is clear.

Safety and Reach-Avoid Probability Estimation. Consider the discrete-time stochastic system (1) with a given control policy π , an initial set $I \subseteq \mathbb{R}^n$, and an unsafe set $U \subseteq \mathbb{R}^n$. The *safety probability estimation* problem seeks to determine a tight lower bound λ on the probability that the system's trajectory never enters the unsafe set U . Formally, the goal is to find a nontrivial λ such that

$$\mathbb{P}_{\mathbf{x}_0}(X_n \notin U \text{ for all } n \in \mathbb{N}) \geq \lambda, \quad (2)$$

for all initial states $\mathbf{x}_0 \in I$. In the *reach-avoid* scenario, an additional target set $T \subseteq \mathbb{R}^n$ is specified. The objective is then to estimate a lower bound λ on the probability that the system reaches the target set T while avoiding the unsafe set U up to that point. Specifically, the reach-avoid probability estimation problem is to find a nontrivial λ such that

$$\mathbb{P}_{\mathbf{x}_0}(\text{RA}(U, T)) \geq \lambda, \quad (3)$$

for all initial states $\mathbf{x}_0 \in I$, where $\text{RA}(U, T)$ denotes the event that the trajectory reaches T before entering U :

$$\text{RA}(U, T) := \{\exists n \in \mathbb{N} : (X_n \in T) \wedge (\forall i < n : X_i \notin U)\}.$$

Traditional methods addressing these problems frequently employ *stochastic barrier functions* to establish safety and reach-avoid guarantees. However, these methods typically rely on offline computations based on prior system knowledge and statistical models of uncertainty, lacking adaptability to real-time observations. Such offline analyses fail to

fully exploit valuable runtime data that could significantly refine safety assessments.

In practical scenarios, discrete-time observations become available at runtime, revealing that the system state resides within certain subsets O_i at discrete time instances t_i . These runtime observations provide critical updates regarding system behavior and environmental interactions, potentially altering the probabilities associated with safety or reach-avoid conditions. Consequently, traditional stochastic estimation methods must be adapted to dynamically integrate this observational data. Motivated by this challenge, we introduce the problem of *runtime safety and reach-avoid prediction*, where real-time observations are leveraged to yield more accurate and responsive safety predictions.

Runtime Safety and Reach-avoid Prediction. Consider again the stochastic system (1) with control policy π , unsafe set U , initial set I , and a sequence of observation times $\{t_i\}_{i \in \mathbb{N}}$ with $t_0 = 0 < t_1 < t_2 < \dots$. At each observation time t_i , the system state is observed to belong to a subset $O_i \subseteq \mathbb{R}^n$. The task is to dynamically compute or update the probability that, conditioned on the system being in O_i at each observation time t_i , the trajectory remains safe (or satisfies the reach-avoid property) from that point onward. Formally, given any finite observation sequence $\{(t_i, O_i)\}_{i=1}^k$, the objective is to find a nontrivial λ such that

$$\mathbb{P}_{\mathbf{x}_0}(X_n \notin U \text{ for all } n \in \mathbb{N} \mid X_{t_i} \in O_i \text{ for } i \leq k) \geq \lambda \quad (4)$$

or, for the reach-avoid case:

$$\mathbb{P}_{\mathbf{x}_0}(\text{RA}(U, T) \mid X_{t_i} \in O_i \text{ for } i \leq k) \geq \lambda, \quad (5)$$

where probability estimates are updated iteratively as new observations become available at each observation time.

Remark 1. For well-posedness, we assume the initial set I and target set T do not intersect the unsafe set U , and each observation O_i is disjoint from both U and T .

Remark 2. As new observations are incorporated, the conditional safety or reach-avoid probability may increase or decrease, depending on observations. Therefore, the probability does not necessarily change monotonically, reflecting the adaptive and data-driven nature of our framework.

Throughout the process of incorporating observations, we implicitly assume that the system has not entered the unsafe set U (or reached the target set, in the reach-avoid setting) at any prior time. This assumption is both natural and practically justified, as entering the unsafe region or reaching the target is typically a detectable event in real systems. Once the system is observed to have entered U or reached the target, further prediction or updating of safety or reach-avoid probabilities is no longer necessary.

Theoretical Results: Observation-aware Barrier Functions

In this section, we develop a theoretical framework for estimating safety and reach-avoid probabilities conditioned on runtime observations. We show that runtime safety and reach-avoid predication can be systematically characterized

using the proposed observation-aware barrier functions. We first address runtime safety estimation in detail, and then briefly discuss the reach-avoid case. Proof sketches for the main results are provided below, with complete proofs given in the full version.

For runtime safety probability estimation, we leverage Bayes' theorem to reformulate the conditional safety probability in terms of joint and marginal probabilities involving both the system trajectory and the observation sequence. Given a finite sequence of observations $\{(t_i, O_i)\}_{i=1}^k$, the conditional safety probability can be written as

$$\begin{aligned} & \mathbb{P}_{\mathbf{x}_0}(X_n \notin U \text{ for all } n \in \mathbb{N} \mid X_{t_i} \in O_i \text{ for } i \leq k) \\ &= 1 - \mathbb{P}_{\mathbf{x}_0}(\exists n, X_n \in U \mid X_{t_i} \in O_i \text{ for } i \leq k) \\ &= 1 - \frac{\mathbb{P}_{\mathbf{x}_0}((\exists n, X_n \in U) \wedge (X_{t_i} \in O_i, \forall i \leq k))}{\mathbb{P}_{\mathbf{x}_0}(X_{t_i} \in O_i, \forall i \leq k)}. \end{aligned}$$

This formulation reveals that bounding the conditional safety probability reduces to estimating two terms: the probability of observing both a safety violation and the given observation sequence (the numerator), and the probability of observing the sequence alone (the denominator). If we can compute a lower bound q for the denominator and an upper bound p for the numerator, we immediately obtain

$$\mathbb{P}_{\mathbf{x}_0}(X_n \notin U \text{ for all } n \in \mathbb{N} \mid X_{t_i} \in O_i \text{ for } i \leq k) \geq 1 - \frac{p}{q}.$$

This reduces the original problem to estimating two manageable probability bounds.

To facilitate this, we introduce *observation-aware barrier functions* (OBFs) and *observation-aware safety barrier functions* (OSBFs), which serve as core analytical tools for bounding the denominator and numerator, respectively, in the conditional probability formulation above. These functions extend classical barrier function techniques by explicitly incorporating information from discrete-time observations, enabling more accurate and adaptive safety prediction in the presence of runtime data.

Observation-aware Barrier Functions. We define an *observation-aware barrier function* (OBF) to be a function $B : \mathbb{N}_{\leq(t_k+1)} \times \mathcal{X} \rightarrow \mathbb{R}$, where $\mathbb{N}_{\leq(t_k+1)} := \{0, 1, \dots, t_k + 1\}$ denotes the discrete time indices from the initial time up to one step after the last observation. The OBF assigns values to time-state pairs and generalizes classical barrier functions to estimate the probability of observing a specified sequence of events, independent of safety requirements. The construction of OBFs enables lower bounds on observation probabilities to be propagated and updated as new information is gathered during system execution.

Intuitively, the value of an OBF is required to remain within $[0, 1]$ everywhere (*Probability condition*), to be lower bounded by q at the initial state (*Initial condition*), and to reach 1 at the terminal time after the last observation (*Terminal condition*). Along system trajectories, the OBF must not decrease in expectation before and at each observation time, with conditions further refined depending on whether the current state satisfies the observation at time t (*Safe before observation* and *Observation-aware increase*). These properties ensure that, if the system consistently matches all observations while remaining safe, the OBF can only increase,

thereby certifying a rigorous lower bound on the probability of realizing the entire observation sequence.

Definition 1 (Observation barrier functions). *Let $I \subseteq \mathbb{R}^n$ and $U \subseteq \mathbb{R}^n$ be the initial set and unsafe set, respectively, and let q be the probability threshold. Given an observation sequence $\{(t_i, O_i)\}_{i=1}^k$, a function $B : \mathbb{N}_{\leq(t_k+1)} \times \mathcal{X} \rightarrow \mathbb{R}$ is said to be an observation barrier function (OBF) with respect to $I, U, \{(t_i, O_i)\}$, and q if it satisfies:*

1. *Probability condition:* $0 \leq B(t, \mathbf{x}) \leq 1$ for all $(t, \mathbf{x}) \in \mathbb{N}_{\leq(t_k+1)} \times \mathcal{X}$;
2. *Initial condition:* $B(0, \mathbf{x}) \geq q$ for all $\mathbf{x} \in I$;
3. *Terminal condition:* $B(t_k + 1, \mathbf{x}) = 1$ for all $\mathbf{x} \in \mathcal{X}$;
4. *Safe before observation:* for all $t \in \{0, 1, \dots, t_k\}$,

$$[\mathbf{x} \in \mathcal{X} \setminus U] \cdot \mathbb{E}_{\omega \sim d}[B(t+1, f(\mathbf{x}, \pi(\mathbf{x}), \omega_t))] \geq B(t, \mathbf{x}).$$
5. *Observation-aware increase:* for all $t \in \{t_1, t_2, \dots, t_k\}$,

$$[\mathbf{x} \in O_t] \cdot \mathbb{E}_{\omega \sim d}[B(t+1, f(\mathbf{x}, \pi(\mathbf{x}), \omega_t))] \geq B(t, \mathbf{x}).$$

Remark 3. *The safe before observation condition in Definition 1 explicitly reflects the assumption that the system has not entered the unsafe set before each observation time. Consequently, all probabilities in this framework are conditioned on prior safety, aligning with practical scenarios where entering the unsafe region is immediately observable and thus further predictions become unnecessary.*

The following theorem formalizes the probabilistic guarantee provided by the existence of an observation-aware barrier function. Specifically, it asserts that if such a function can be constructed, then the probability of realizing the entire prescribed observation sequence is lower bounded by the threshold q .

Theorem 1. *Suppose there exists an observation-aware barrier function $B(t, \mathbf{x})$ for the system with initial set I , unsafe set U , observation sequence $\{(t_i, O_i)\}_{i=1}^k$, and threshold q . Then, for any $x_0 \in I$,*

$$\mathbb{P}_{x_0}(X_{t_i} \in O_i, \forall i \leq k) \geq q.$$

Proof sketch. The main idea is to construct a submartingale based on the OBF, such that its expected value at time $t_k + 1$ exactly equals the probability of realizing the prescribed observation sequence. Intuitively, a submartingale is a stochastic process whose expected value does not decrease over time; it captures the notion that, as the system evolves, the likelihood of satisfying the observation constraints cannot decrease unexpectedly. This property is crucial for establishing lower bounds on observation probabilities. Specifically, for the process

$$Y_n := \prod_{i < n} [X_i \in (\mathcal{X} \setminus U)] \prod_{t_i < n} [X_{t_i} \in O_{t_i}] \cdot B(n, X_n),$$

the OBF conditions guarantee that Y_n is a submartingale. By the submartingale property and the OBF initial condition, we have $q \leq B(0, x_0) \leq \mathbb{E}[Y_{t_k+1}]$, which exactly equals the desired observation probability. This establishes the lower bound. \square

Observation-aware Safety Barrier Functions. While the observation-aware barrier function provides a rigorous lower bound on the probability of realizing a specified sequence of observations, runtime safety prediction also requires quantifying the probability of safety violations. To address this, we introduce the notion of an *observation-aware safety barrier function* (OSBF).

Definition 2 (Observation-aware safety barrier functions). *Let $I \subseteq \mathbb{R}^n$ be the initial set, $U \subseteq \mathbb{R}^n$ the unsafe set, and let p be the probability threshold. Given an observation sequence $\{(t_i, O_i)\}_{i=1}^k$, a function $V : \mathbb{N} \times \mathcal{X} \rightarrow \mathbb{R}$ is said to be an observation-aware safety barrier function (OSBF) with respect to $U, I, \{(t_i, O_i)\}$, and p if it satisfies:*

1. *Nonnegativity:* $V(t, \mathbf{x}) \geq 0$ for all $(t, \mathbf{x}) \in \mathbb{N} \times \mathcal{X}$.
2. *Initial condition:* $V(0, \mathbf{x}) \leq p$ for each $\mathbf{x} \in I$.
3. *Safety condition:* $V(t, \mathbf{x}) \geq 1$ for $t \geq t_k + 1$ and $\mathbf{x} \in U$.
4. *Safe before observation:* for $t \in \{0, 1, \dots, t_k\}$,

$$[\mathbf{x} \in \mathcal{X} \setminus U] \cdot \mathbb{E}_{\omega \sim d}[V(t+1, f(\mathbf{x}, \pi(\mathbf{x}), \omega))] \leq V(t, \mathbf{x}).$$

5. *Observation-aware decrease:* for $t \in \{t_1, \dots, t_k\}$,

$$[\mathbf{x} \in O_t] \cdot \mathbb{E}_{\omega \sim d}[V(t+1, f(\mathbf{x}, \pi(\mathbf{x}), \omega))] \leq V(t, \mathbf{x}).$$

6. *Expected decrease after last observation:* for $t \geq t_k + 1$, and $\mathbf{x} \in \mathcal{X} \setminus U$,

$$\mathbb{E}_{\omega \sim d}[V(t+1, f(\mathbf{x}, \pi(\mathbf{x}), \omega))] \leq V(t, \mathbf{x}).$$

Remark 4. *Compared to the OBF, the OSBF includes an additional condition—the expected decrease after the last observation (cond 6). This condition is essential for certifying the system’s safety beyond the observation horizon.*

Intuitively, an OSBF must be nonnegative (*Nonnegativity*), bounded above at the initial state (*Initial condition*), and bounded below on unsafe states after the last observation (*Safety condition*). Its value must not increase in expectation before and at each observation time, with specific conditions depending on whether the state matches the most recent observation (*Safe before observation* and *Observation-aware decrease*). After the final observation, a standard expected decrease applies (*Expected decrease*). Collectively, these properties ensure that if the system stays safe and satisfies all observations, the OSBF cannot increase unexpectedly, and any safety violation is flagged when the function exceeds a certain threshold. The following theorem formalizes this guarantee.

Theorem 2. *Suppose there exists an OSBF $V(t, \mathbf{x})$ for the system with initial set I , unsafe set U , observation sequence $\{(t_i, O_i)\}_{i=1}^k$, and threshold p . Then, for any $\mathbf{x}_0 \in I$,*

$$\mathbb{P}_{\mathbf{x}_0}((\exists n, X_n \in U) \wedge (X_{t_i} \in O_i, \forall i \leq k)) \leq p.$$

Proof sketch. The complete proof is provided in the Appendix. The argument is similar to the previous result: we construct a supermartingale based on the OSBF, whose expected value at the stopping time for entering the unsafe set captures the safety violation probability. The result follows from the optional stopping theorem and the OSBF conditions. \square

Together, the OSF and the OSBF provide a compositional approach to lower bounding the conditional safety probability. By separately certifying a bound for the joint probability of safety and observation events (via the OSBF) and a bound for the probability of the observation sequence (via the OBF), we can combine these results through the Bayesian reformulation established earlier. The following theorem formalizes this compositional guarantee:

Theorem 3. *Suppose the notations and assumptions above hold, and there exist an OSBF $V(t, \mathbf{x})$ and an OBF $B(t, \mathbf{x})$ for the system with initial set I , unsafe set U , observation sequence $\{(t_i, O_i)\}_{i=1}^k$, and thresholds p and q , respectively. Then, for any $\mathbf{x}_0 \in I$,*

$$\mathbb{P}_{\mathbf{x}_0}(X_n \notin U \text{ for all } n \in \mathbb{N} \mid X_{t_i} \in O_i \text{ for } i \leq k) \geq 1 - \frac{p}{q}.$$

Remark 5. *The tightness of the obtained bound can be quantified by evaluating how well the computed barrier functions satisfy the barrier conditions; see the full version of this paper for the detailed formulation.*

The Reach-avoid Case. The *reach-avoid* scenario can be treated analogously to the safety case, with the additional assumption that the system will eventually enter $U \cup T$ with probability one. Under this assumption, the conditional reach-avoid probability can be expressed as

$$\begin{aligned} & \mathbb{P}_{\mathbf{x}_0}(\text{RA}(U, T) \mid X_{t_i} \in O_i \text{ for } i \leq k) \\ &= 1 - \mathbb{P}_{\mathbf{x}_0}(\text{RA}(T, U) \mid X_{t_i} \in O_i \text{ for } i \leq k) \\ &= 1 - \frac{\mathbb{P}_{\mathbf{x}_0}(\text{RA}(T, U) \wedge (X_{t_i} \in O_i, \forall i \leq k))}{\mathbb{P}_{\mathbf{x}_0}(X_{t_i} \in O_i, \forall i \leq k)}. \end{aligned}$$

The second equality holds because the system is guaranteed to enter either U or T eventually. Here, $\text{RA}(T, U)$ denotes the event that the trajectory reaches U before entering T :

$$\text{RA}(T, U) := \{\exists n : X_n \in U, \forall j < n, X_j \notin T\}.$$

Therefore, the reach-avoid probability estimation reduces to obtaining a lower bound for the denominator and an upper bound for the numerator. Notably, the denominator coincides with that in the safety case, so only the numerator requires a new estimate.

Remark 6. *Rather than directly lower bounding the conditional reach-avoid probability, we focus on upper bounding $\mathbb{P}_{\mathbf{x}_0}(\text{RA}(T, U) \mid X_{t_i} \in O_i, \forall i \leq k)$. This approach is preferable because upper bounds are typically easier to obtain, and the denominator estimate from the safety case can be reused.*

To upper bound the numerator above, we introduce the notion of an *observation-aware reach-avoid barrier function* (ORBF), which generalizes the observation-aware safety barrier function to the reach-avoid setting.

Definition 3 (Observation-aware reach-avoid barrier functions). *Let $I \subseteq \mathbb{R}^n$ be the initial set, $U \subseteq \mathbb{R}^n$ the unsafe set, T the target set, and let p be the probability threshold. Given an observation sequence $\{(t_i, O_i)\}_{i=1}^k$, a function $V : \mathbb{N} \times \mathcal{X} \rightarrow \mathbb{R}$ is said to be an observation-aware reach-avoid barrier function (ORBF) with respect to $U, T, I, \{(t_i, O_i)\}$, and p if it satisfies conditions 1–5 in Definition 2, and further satisfies:*

6. *Expected decrease after last observation (RA case): for $t \geq t_k + 1$ and $\mathbf{x} \in \mathcal{X} \setminus (U \cup T)$,*

$$\mathbb{E}_{\omega \sim d}[V(t+1, f(\mathbf{x}, \pi(\mathbf{x}), \omega))] \leq V(t, \mathbf{x}).$$

Remark 7. *The ORBF differs from the OSBF in that its final expected decrease condition applies to all $\mathbf{x} \in \mathcal{X} \setminus (U \cup T)$, reflecting that safety requirements cease once the target set T is reached.*

The ORBF provides a rigorous characterization of the barrier property needed to upper bound the probability of violating the reach-avoid objective, given the observation constraints. The main probabilistic guarantees are formalized as follows:

Theorem 4. *If there exists an ORBF $V(t, \mathbf{x})$ for the system with initial set I , unsafe set U , target set T , observation sequence $\{(t_i, O_i)\}_{i=1}^k$, and threshold p , then for any $\mathbf{x}_0 \in I$,*

$$\mathbb{P}_{\mathbf{x}_0}(\text{RA}(T, U) \wedge (X_{t_i} \in O_i, \forall i \leq k)) \leq p.$$

Combining this result with the previously established lower bound on the probability of observing the sequence, we obtain the following conditional reach-avoid probability guarantee:

Theorem 5. *Suppose the notations and assumptions above hold, and the system enters $U \cup T$ with probability one. If there exist an ORBF $V(t, \mathbf{x})$ and an OSBF $B(t, \mathbf{x})$ for the system with initial set I , unsafe set U , target set T , observation sequence $\{(t_i, O_i)\}_{i=1}^k$, and thresholds p and q respectively, then for any $\mathbf{x}_0 \in I$,*

$$\mathbb{P}_{\mathbf{x}_0}(\text{RA}(U, T) \mid X_{t_i} \in O_i, \forall i \leq k) \geq 1 - \frac{p}{q}.$$

Remark 8. *The almost-sure reachability assumption – that the system eventually enters $U \cup T$ with probability one – can often be established using standard techniques such as stochastic ranking functions (Chatterjee, Fu, and Goharshady 2016). See also (Majumdar and Sathiyarayanan 2025) for results on the termination of stochastic systems.*

Algorithm for Runtime Safety and Reach-avoid Prediction

In this section, we present practical algorithms for runtime safety and reach-avoid prediction based on observation-aware barrier functions. Our approach combines offline polynomial optimization and online backward iteration, enabling efficient updates of probability bounds upon receiving new observations.

The complete procedure is summarized in Algorithm 1. In the offline phase, we first synthesize a barrier function $v(\mathbf{x})$ satisfying the observation-independent conditions (conditions 1, 3, and 6 for OSBF and ORBF), which provides precomputed values for future time steps beyond the latest observation (lines 1–6). In the online phase, each time a new observation is received, we iteratively update the OBF, OSBF, or ORBF via backward computation (lines 7–17). Specifically, at each step, barrier values are updated using expectations over successor states, enforcing observation-aware and safety-related constraints. This backward iterative update dynamically refines the probability bounds in

Algorithm 1 Runtime Safety and Reach-avoid Prediction

Require: System dynamics f , control policy π , initial set I , unsafe set U , (target set T for reach-avoid), runtime observations

Ensure: Probability lower bound for system safety or reach-avoid

```

1: ▷ Offline synthesis
2: if safety case then
3:   Synthesize  $v(\mathbf{x})$  satisfying conds. (1), (3), (6) in Definition 2
4: else if reach-avoid case then
5:   Verify the system enters  $U \cup T$  almost surely
6:   Synthesize  $v(\mathbf{x})$  satisfying conds. (1), (3), (6) in Definition 3
7: ▷ Online update
8: repeat
9:   if new observation  $O_k$  at time  $t_k$  then
10:    ▷ Observation sequence:  $\{(t_i, O_i)\}_{i=1}^k$ 
11:     $q, B(t, \mathbf{x}) \leftarrow \text{GET-OBF}(\{(t_i, O_i)\}_{i=1}^k)$ 
12:    if safety case then
13:       $p, V(t, \mathbf{x}) \leftarrow \text{GET-OSBF}(\{(t_i, O_i)\}_{i=1}^k, v(\mathbf{x}))$ 
14:    else if reach-avoid case then
15:       $p, V(t, \mathbf{x}) \leftarrow \text{GET-ORBF}(\{(t_i, O_i)\}_{i=1}^k, v(\mathbf{x}))$ 
16:      Update bound:  $1 - p/q$ 
17: until no new observation

```

response to new runtime data, thus ensuring rigorous and adaptive runtime safety and reach-avoid prediction. The detailed computation procedures for OBF, OSBF, and ORBF are provided below.

Computing OBF. A tight OBF can be constructed via backward iteration as shown in Algorithm 2, procedure GET-OBF. Starting from the terminal time (where $B(t_k + 1, \mathbf{x}) = 1$ for all \mathbf{x}), the OBF is recursively computed backward in time. At each step, the value is updated by the expected value over successor states, ensuring the safe-before-observation condition is satisfied. At observation times, the OBF is set to zero outside the observed set to enforce the observation-aware increase condition. This procedure ensures all conditions in Definition 1 are satisfied. Moreover, by maximizing the value at each step subject to these constraints, the method yields the largest possible OBF and, consequently, the tightest lower bound for the observation probability.

Computing OSBF and ORBF. For OSBF construction, we assume $V(t, \mathbf{x})$ is time-invariant after the last observation, i.e., $V(t, \mathbf{x}) = v(\mathbf{x})$ for all $t \geq t_k + 1$. Under this assumption, the offline computation of $v(\mathbf{x})$ only needs to satisfy the nonnegativity, safety, and expected decrease constraints, independent of runtime observations. This corresponds to lines 1–3 in Algorithm 1. Specifically, the offline synthesis of $v(\mathbf{x})$ involves solving:

1. $v(\mathbf{x}) \geq 0$, for all $\mathbf{x} \in \mathcal{X}$ (Nonnegativity);
2. $v(\mathbf{x}) \geq 1$, for all $\mathbf{x} \in U$ (Safety condition);
3. $\mathbb{E}_{\omega \sim d}[v(f(\mathbf{x}, \pi(\mathbf{x}), \omega))] \leq v(\mathbf{x})$, for all $\mathbf{x} \in (\mathcal{X} \setminus U)$ (Expected decrease).

When f , π , and v are polynomials, these can be encoded as sum-of-squares (SOS) constraints and efficiently solved

Algorithm 2 Calculating OBF, OSBF, and ORBF

Require: System dynamics f , policy π , initial set I , unsafe set U , (target set T for reach-avoid).

```

1:  $\triangleright$  Backward calculation of OBF ◁
2: procedure GET-OBF( $\{(t_i, O_i)\}_{i=1}^k$ )
3:    $B(t_k + 1, x) \leftarrow 1$  for all  $x \in \mathcal{X}$   $\triangleright$  Terminal condition
4:   for  $t = t_k$  down to 0 do
5:     if  $t$  is an observation time then
6:        $B(t, x)$ 
7:        $\leftarrow \begin{cases} \mathbb{E}[B(t+1, f(x, \pi(x), \omega_t))] & \text{if } x \in O_t \\ 0 & \text{else} \end{cases}$ 
8:     else
9:        $B(t, x)$ 
10:       $\leftarrow \begin{cases} \mathbb{E}[B(t+1, f(x, \pi(x), \omega_t))] & \text{if } x \in \mathcal{X} \setminus U \\ 0 & \text{else} \end{cases}$ 
11:    $q \leftarrow \min_{x \in I} B(0, x)$   $\triangleright$  Lower bound at initial set
12:   return  $q, B$ 

11:  $\triangleright$  Backward calculation of OSBF with offline  $v(x)$  ◁
12: procedure GET-OSBF( $\{(t_i, O_i)\}_{i=1}^k, v(x)$ )
13:    $V(t_k + 1, x) \leftarrow v(x)$  for all  $x \in \mathcal{X}$   $\triangleright$  Terminal condition
14:   for  $t = t_k$  down to 0 do
15:     if  $t$  is an observation time then
16:        $V(t, x)$ 
17:        $\leftarrow \begin{cases} \mathbb{E}[V(t+1, f(x, \pi(x), \omega_t))] & \text{if } x \in O_t \\ 0 & \text{else} \end{cases}$ 
18:     else
19:        $V(t, x)$ 
20:       $\leftarrow \begin{cases} \mathbb{E}[V(t+1, f(x, \pi(x), \omega_t))] & \text{if } x \in \mathcal{X} \setminus U \\ 0 & \text{else} \end{cases}$ 
21:    $p \leftarrow \max_{x \in I} V(0, x)$   $\triangleright$  Upper bound at initial set
22:   return  $p, V$ 

21:  $\triangleright$  ORBF shares the same procedure as OSBF, but with dif- ◁
    ferent  $v(x)$ 
22: procedure GET-ORBF( $\{(t_i, O_i)\}_{i=1}^k, v(x)$ )
23:   return GET-OSBF( $\{(t_i, O_i)\}_{i=1}^k, v(x)$ )
  
```

with semidefinite programming, e.g., using MOSEK. See the appendix for formulation details.

Once $v(x)$ is synthesized, $V(t, x)$ for $t = 0, \dots, t_k$ is computed via backward iteration similar to OBF. At observation times, $V(t, x)$ is set to zero outside the observed set (to enforce the observation-aware decrease), while at other times, it is updated via the expectation over successor states, subject to the safe-before-observation condition.

The synthesis of ORBF follows the same structure as OSBF, with only the final expected decrease condition modified for the reach-avoid setting. Once $v(x)$ is obtained, the ORBF is constructed by backward iteration, as above.

Experiments

To demonstrate the effectiveness and applicability of our runtime safety and reach-avoid prediction framework, we implemented the proposed algorithms in Python 3.13 (for backward iteration) and MATLAB R2025a interfaced with

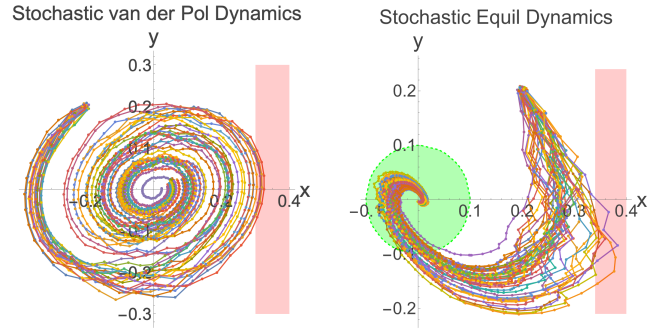


Figure 1: Visualization of the stochastic Van der Pol system (safety) and Equil system (reach-avoid). The red region denotes the unsafe set; the green region denotes the target set.

YALMIP (Löfberg 2004) and MOSEK (Andersen, Roos, and Terlaky 2003) (for offline barrier synthesis). All experiments were conducted on a 2.60 GHz Intel Core i9-13905H laptop with 32 GB RAM, running 64-bit Windows 11.

Benchmarks and Experiment Setup. We evaluated our framework on a set of polynomial benchmarks frequently used in the control literature. These include the Van der Pol oscillator (Kanamaru 2007), a classical nonlinear system commonly employed for verifying stochastic techniques due to its rich nonlinear dynamics, and the Equil system (Prajna, Jadbabaie, and Pappas 2007), a variant of the Duffing oscillator extensively studied in control theory. The stochastic trajectories of these two representative benchmarks are illustrated in Fig. 1. Further details about each benchmark are provided in the appendix. For each benchmark, we conducted runtime safety or reach-avoid predictions under different observation scenarios: no observation (purely offline prediction) and incremental online updates with up to five discrete-time observations.

Experimental Results and Analysis. The experimental results are summarized in Table 1. The experimental results clearly demonstrate the advantages and impact of incorporating runtime observations into safety and reach-avoid prediction:

- *Computation Efficiency:* The initial offline safety or reach-avoid estimation, computed without runtime observations, typically incurs higher computational costs due to semidefinite optimization. In contrast, subsequent online predictions that incorporate runtime observations exhibit shorter computation times. This confirms that our online iterative updating scheme efficiently leverages the precomputed barriers, enabling rapid predictions suitable for real-time applications.
- *Probability Refinement and Adaptation:* Runtime observations led to notable changes in the estimated safety and reach-avoid probabilities. As experiments show, additional runtime observations do not guarantee monotonically improving safety predictions. Depending on the observed states, predictions either enhanced the system's safety confidence (e.g., the osc benchmark) or, con-

Benchmark	No observation		2 observations		3 observations		4 observations		5 observations	
	time (off.)	prob.	time (on.)	prob.	time (on.)	prob.	time (on.)	prob.	time (on.)	prob.
arch	2.39s	0.813	0.01s	0.768	0.03s	0.780	0.38s	0.769	0.51s	0.574
descent	0.78s	0.704	0.01s	0.626	0.15s	0.655	0.31s	0.502	1.12s	0.124
osc	1.49s	0.436	0.01s	0.532	0.01s	0.470	0.03s	0.850	0.62s	0.873
vanderpol-1	1.57s	0.343	0.01s	0.380	0.06s	0.251	0.17s	0.336	0.23s	0.227
vanderpol-2	1.78s	0.158	0.01s	0.169	0.06s	0.224	0.15s	0.406	0.21s	0.567
liederivative	6.80s	0.246	0.01s	0.232	0.15s	0.174	1.26s	0.157	1.76s	0.056
equil	1.60s	0.357	0.01s	0.274	0.03s	0.239	0.43s	0.194	3.72s	0.087
lyapunov	8.63s	0.226	0.05s	0.283	0.50s	0.279	0.85s	0.352	2.50s	0.308
lotka	14.96s	0.504	0.02s	0.708	0.42s	0.784	0.61s	0.907	5.62s	0.963

Table 1: Experimental results for runtime safety and reach-avoid probability prediction. *time (off)* denotes offline computation time without observations; *time (on)* denotes online prediction time with observations; *prob* is the probability lower bound for safety or reach-avoid.

versely, markedly reduced the estimated safety (e.g., descent benchmark). This reflects the adaptive nature of our method, as the predictions dynamically align with actual system trajectories and observations rather than solely relying on initial conservative estimates.

- *Practical Implications:* The experiments underline the importance of runtime information: without observations, estimates remain conservative or overly optimistic, potentially misleading safety assessments. By incorporating discrete-time observations, the framework enables adaptive and realistic safety predictions, crucial for decision-making in safety-critical stochastic systems.

Overall, our experimental evaluations validate that the proposed runtime prediction framework effectively integrates online observations, efficiently computes updated predictions, and provides adaptively refined probability bounds essential for practical safety-critical applications.

Related Work

Verification of Deterministic Systems. The technique presented in this paper falls within the realm of barrier-based approaches. In their seminal work (Prajna, Jadbabaie, and Pappas 2004), Prajna proposed the concept of barrier certificates to encode inductive invariants that witness safety (or dually, reachability) of deterministic dynamical systems over an unbounded-time horizon. Since then, significant efforts have been dedicated to developing more relaxed forms of barrier-certificate conditions that still admit efficient synthesis, thereby leading to exponential-type barrier certificates (Kong et al. 2013), Darboux-type barrier certificates (Zeng et al. 2016), general barrier certificates (Dai et al. 2017), and vector barrier certificates (Sogokon et al. 2018). Similar barrier conditions have been utilized to verify systems with control inputs (Xu et al. 2015; Ames et al. 2016) and disturbances (Wang et al. 2017) against various Linear Temporal Logic (LTL) (Vardi 2005) properties.

Verification of Stochastic Systems. Barrier-based methods have also been extended to verifying stochastic systems (Feng et al. 2020b; Lechner et al. 2022; Jagtap, Soud-

jani, and Zamani 2020; Žikelić et al. 2023). There are also various alternative methods to reason about discrete-time stochastic dynamics, including techniques based on sampling (Henriques et al. 2012; Legay, Sedwards, and Traonouez 2014), dynamic programming (Abate et al. 2008; Summers and Lygeros 2010), Markov abstractions (Lahijanian, Andersson, and Belta 2015), probabilistic model checking (Baier and Katoen 2008; Kwiatkowska 2003) (for finite-state models), and various forms of value iteration (Baier et al. 2017; Hartmanns and Kaminski 2020; Quatmann and Katoen 2018) for determining reachability probabilities in Markov models.

Runtime Monitoring and Verification. Runtime verification (Bartocci et al. 2018) involves dynamically monitoring a system during execution to verify that it satisfies its specified properties. Typically, properties are specified using temporal logic (Deshmukh et al. 2017), system signals are continuously monitored, and robustness metrics quantify how strongly a signal satisfies or violates the specification (Raman et al. 2015; Su et al. 2025). Recently, learning-based runtime monitoring (Yu, Žikelić, and Henzinger 2025; Dawson, Gao, and Fan 2023) has attracted significant attention, as it enables scalable and adaptive monitoring in high-dimensional, uncertain environments.

Conclusion

We proposed a runtime prediction framework for safety and reach-avoid probabilities in discrete-time stochastic systems, effectively integrating real-time observations through observation-aware barrier functions. Our approach, combining offline computations with online updates, provides rigorous yet adaptive probability estimates, validated by experimental results on standard benchmarks. A promising avenue for future research is to leverage these runtime predictions for dynamically modifying control policies, thus enhancing real-time system safety and performance.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments. This work is supported by the National Natural Science Foundation of China (Grants Nos. 62502475, W2511064, 62032024), and by the ISCAS Basic Research (Grant No. ISCAS-JCZD-202406).

References

- Abate, A.; Prandini, M.; Lygeros, J.; and Sastry, S. 2008. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica*, 44(11): 2724–2734.
- Allen, L. J. 2010. *An introduction to stochastic processes with applications to biology*. CRC press.
- Ames, A. D.; Xu, X.; Grizzle, J. W.; and Tabuada, P. 2016. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8): 3861–3876.
- Andersen, E. D.; Roos, C.; and Terlaky, T. 2003. On implementing a primal-dual interior-point method for conic quadratic optimization. *Math. Program.*, 95(2): 249–277.
- Baier, C.; and Katoen, J.-P. 2008. *Principles of Model Checking*. MIT press.
- Baier, C.; Klein, J.; Leuschner, L.; Parker, D.; and Wunderlich, S. 2017. Ensuring the Reliability of Your Model Checker: Interval Iteration for Markov Decision Processes. In *CAV (II)*, volume 10426 of *LNCS*, 160–180. Springer.
- Bartocci, E.; Falcone, Y.; Francalanza, A.; and Reger, G. 2018. Introduction to runtime verification. In *Lectures on Runtime Verification: Introductory and Advanced Topics*, 1–33. Springer.
- Bertsekas, D.; and Shreve, S. E. 1996. *Stochastic optimal control: The discrete-time case*, volume 5. Athena Scientific.
- Chatterjee, K.; Fu, H.; and Goharshady, A. K. 2016. Termination analysis of probabilistic programs through Positivstellensatz's. In *International Conference on Computer Aided Verification*, 3–22. Springer.
- Dai, L.; Gan, T.; Xia, B.; and Zhan, N. 2017. Barrier certificates revisited. *J. Symb. Comput.*, 80: 62–86.
- Dawson, C.; Gao, S.; and Fan, C. 2023. Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control. *IEEE Transactions on Robotics*, 39(3): 1749–1767.
- Deshmukh, J. V.; Donzé, A.; Ghosh, S.; Jin, X.; Juniwal, G.; and Seshia, S. A. 2017. Robust online monitoring of signal temporal logic. *Formal Methods in System Design*, 51(1): 5–30.
- Feng, S.; An, J.; and Xu, F. 2025. Runtime Safety and Reach-avoid Prediction of Stochastic Systems via Observation-aware Barrier Functions. arXiv:2511.09192.
- Feng, S.; Chen, M.; Xue, B.; Sankaranarayanan, S.; and Zhan, N. 2020a. Unbounded-time safety verification of stochastic differential dynamics. In *International Conference on Computer Aided Verification*, 327–348. Springer.
- Feng, S.; Chen, M.; Xue, B.; Sankaranarayanan, S.; and Zhan, N. 2020b. Unbounded-Time Safety Verification of Stochastic Differential Dynamics. In *CAV (II)*, volume 12225 of *LNCS*, 327–348. Springer.
- Hartmanns, A.; and Kaminski, B. L. 2020. Optimistic Value Iteration. In *CAV (II)*, volume 12225 of *LNCS*, 488–511. Springer.
- Henriques, D.; Martins, J. G.; Zuliani, P.; Platzer, A.; and Clarke, E. M. 2012. Statistical Model Checking for Markov Decision Processes. In *QEST*, 84–93. IEEE Computer Society.
- Jagtap, P.; Soudjani, S.; and Zamani, M. 2020. Formal synthesis of stochastic systems via control barrier certificates. *IEEE Transactions on Automatic Control*, 66(7): 3097–3110.
- Kanamaru, T. 2007. Van der Pol oscillator. *Scholarpedia*, 2(1): 2202.
- Kong, H.; He, F.; Song, X.; Hung, W. N. N.; and Gu, M. 2013. Exponential-condition-based barrier certificate generation for safety verification of hybrid systems. In *CAV*, volume 8044 of *LNCS*, 242–257. Springer.
- Kwiatkowska, M. Z. 2003. Model Checking for Probability and Time: From Theory to Practice. In *LICS*, 351. IEEE Computer Society.
- Lahijanjan, M.; Andersson, S. B.; and Belta, C. 2015. Formal Verification and Synthesis for Discrete-Time Stochastic Systems. *IEEE Trans. Autom. Control.*, 60(8): 2031–2045.
- Lechner, M.; Zikelic, D.; Chatterjee, K.; and Henzinger, T. A. 2022. Stability verification in stochastic control systems via neural network supermartingales. In *Proceedings of the aaai conference on artificial intelligence*, volume 36, 7326–7336.
- Legay, A.; Sedwards, S.; and Traonouez, L. 2014. Scalable Verification of Markov Decision Processes. In *SEFM Workshops*, volume 8938 of *LNCS*, 350–362. Springer.
- Löfberg, J. 2004. YALMIP: A toolbox for modeling and optimization in MATLAB. In *CACSD'04*, 284–289.
- Majumdar, R.; and Sathiyarayanan, V. 2025. Sound and complete proof rules for probabilistic termination. *Proceedings of the ACM on Programming Languages*, 9(POPL): 1871–1902.
- Paul, W.; and Baschnagel, J. 2013. *Stochastic Processes: From Physics to Finance*. Springer.
- Prajna, S.; Jadbabaie, A.; and Pappas, G. J. 2004. Stochastic safety verification using barrier certificates. In *CDC'04*, volume 1, 929–934. IEEE.
- Prajna, S.; Jadbabaie, A.; and Pappas, G. J. 2007. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8): 1415–1428.
- Quatmann, T.; and Katoen, J. 2018. Sound Value Iteration. In *CAV (I)*, volume 10981 of *LNCS*, 643–661. Springer.
- Raman, V.; Donzé, A.; Sadigh, D.; Murray, R. M.; and Seshia, S. A. 2015. Reactive synthesis from signal temporal

- logic specifications. In *Proceedings of the 18th international conference on hybrid systems: Computation and control*, 239–248.
- Sogokon, A.; Ghorbal, K.; Tan, Y. K.; and Platzer, A. 2018. Vector Barrier Certificates and Comparison Systems. In *FM'18*, volume 10951 of *Lecture Notes in Computer Science*, 418–437. Springer.
- Steele, J. M. 2001. *Stochastic calculus and financial applications*, volume 1. Springer.
- Su, H.; Shankar, S.; Pinisetty, S.; Roop, P. S.; and Zhan, N. 2025. Runtime enforcement of CPS against signal temporal logic. In *Proceedings of the 28th ACM International Conference on Hybrid Systems: Computation and Control*, 1–11.
- Summers, S.; and Lygeros, J. 2010. Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem. *Autom.*, 46(12): 1951–1961.
- Vardi, M. Y. 2005. An automata-theoretic approach to linear temporal logic. In *Logics for concurrency: structure versus automata*, 238–266. Springer.
- Wang, Q.; Li, Y.; Xia, B.; and Zhan, N. 2017. Generating semi-algebraic invariants for non-autonomous polynomial hybrid systems. *J. Syst. Sci. Complex.*, 30(1): 234–252.
- Xu, X.; Tabuada, P.; Grizzle, J. W.; and Ames, A. D. 2015. Robustness of control barrier functions for safety critical control. In *ADHS*, volume 48, 54–61. Elsevier.
- Yu, E.; Žikelić, D.; and Henzinger, T. A. 2025. Neural Control and Certificate Repair via Runtime Monitoring. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 26409–26417.
- Zeng, X.; Lin, W.; Yang, Z.; Chen, X.; and Wang, L. 2016. Darboux-type barrier certificates for safety verification of nonlinear hybrid systems. In *EMSOFT*, 1–10. ACM.
- Žikelić, D.; Lechner, M.; Henzinger, T. A.; and Chatterjee, K. 2023. Learning control policies for stochastic systems with reach-avoid guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 11926–11935.