Delft University of Technology

Formal Verification of Unknown Dynamical Systems Via Gaussian Process Regression

Skovbekk, John; Laurenti, Luca; Frew, Eric; Lahijanian, Morteza

DOI
10.1109/TAC.2025.3532812

Publication date
2025

Document Version
Final published version

Published in
IEEE Transactions on Automatic Control

Citation (APA)
Skovbekk, J., Laurenti, L., Frew, E., & Lahijanian, M. (2025). Formal Verification of Unknown Dynamical Systems Via Gaussian Process Regression. *IEEE Transactions on Automatic Control*, *70*(8), 4960-4975. https://doi.org/10.1109/TAC.2025.3532812

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Green Open Access added to TU Delft Institutional Repository
as part of the Taverne amendment.

More information about this copyright law amendment
can be found at https://www.openaccess.nl.

Otherwise as indicated in the copyright section:
the publisher is the copyright holder of this work and the
author uses the Dutch legislation to make this work public.

Formal Verification of Unknown Dynamical Systems via Gaussian Process Regression

John Skovbekk, *Student Member, IEEE*, Luca Laurenti, *Member, IEEE*, Eric Frew, and Morteza Lahijanian, *Member, IEEE*

***Abstract*—Leveraging autonomous systems in safety-critical scenarios requires verifying their behaviors in the presence of uncertainties and black-box components that influence the system dynamics. In this work, we develop a framework for verifying discrete-time dynamical systems with unmodeled dynamics and noisy measurements against temporal logic specifications from an input–output dataset. The verification framework employs Gaussian process (GP) regression to learn the unknown dynamics from the dataset and abstracts the continuous-space system as a finite-state, uncertain Markov decision process (MDP). This abstraction relies on space discretization and transition probability intervals that capture the uncertainty due to the error in GP regression by using reproducible kernel Hilbert space analysis as well as the uncertainty induced by discretization. The framework utilizes existing model checking tools for verification of the uncertain MDP abstraction against a given temporal logic specification. We establish the correctness of extending the verification results on the abstraction created from noisy measurements to the underlying system. We show that the computational complexity of the framework is polynomial in the size of the dataset and discrete abstraction. The complexity analysis illustrates a tradeoff between the quality of the verification results and the computational burden to handle larger datasets and finer abstractions. Finally, we demonstrate the efficacy of our learning and verification framework on several case studies with linear, nonlinear, and switched dynamical systems.**

***Index Terms*—Bayesian inference, data-driven certification, data-driven modeling, formal logic, formal verification, Gaussian processes (GPs), Markov decision processes (MDPs).**

I. Introduction

RECENT advances in technology have led to a rapid growth of autonomous systems operating in *safety-critical* domains. Examples include self-driving vehicles, unmanned aircraft, and surgical robotics. As these systems are given such delicate roles, it is *essential* to provide guarantees on their performance. To address this need, formal verification offers a powerful framework with rigorous analysis techniques [1], [2], which are traditionally model-based. An accurate dynamics model for an autonomous system, however, may be unavailable due to, e.g., black-box components, or so complex that existing verification tools cannot handle. To deal with such shortcomings, machine learning offers capable methods that can identify models solely from data. While eliminating the need for an accurate model, these learning methods often lack quantified guarantees with respect to the latent system [3]. The gap between model-based and data-driven approaches is in fact the key challenge in verifiable autonomy. This work focuses on closing this gap by developing a data-driven verification method that can provide formal guarantees for systems with unmodeled dynamics.

Formal verification of continuous control systems has been widely studied, e.g., [4], [5], [6], [7], [8], [9], [10]. These methods are typically based on model checking algorithms [1], [2], which check whether a finite-state model satisfies a given specification. The specification language is usually a form of temporal logic, which provides rich expressivity. Specifically, probabilistic *linear temporal logic* (LTL) and *probabilistic computation tree logic* (PCTL) are used to define specifications for stochastic systems [1], [2]. To bridge the gap between continuous and discrete domains, a finite-state *abstraction* is constructed. This takes the form of a finite Markov process if the latent system is stochastic [10], [11], [12], and the resulting frameworks admit strong formal guarantees. Nevertheless, these frameworks are model-based and cannot be employed for analysis of systems with unknown models.

Machine learning has emerged as a powerful tool for learning unknown functions from data. In particular, Gaussian process (GP) regression is becoming widespread to learn dynamical systems due to its predictive power, uncertainty quantification, and ease of use [13], [14]. By conditioning a prior GP on a dataset, GP regression returns a posterior distribution that predicts the system dynamics via the application of the Bayes rule. However, the main challenge in using GP regression (and machine learning in general) for verification in the context of safety-critical applications is the need to formally quantify the error in the learning process and propagate it in the verification pipeline. Existing data-driven approaches often lack the required formalism and/or are limited to linear systems.

Received 19 June 2024; revised 25 October 2024 and 20 December 2024; accepted 7 January 2025. Date of publication 22 January 2025; date of current version 30 July 2025. This work was supported in part by NSF under Grant 2039062 and in part by NSF Center for Unmanned Aircraft Systems under Grant IIP-1650468. Recommended by Associate Editor Z. Shu. *(Corresponding author: John Skovbekk.)*

John Skovbekk, Eric Frew, and Morteza Lahijanian are with the Smead Aerospace Engineering Sciences Department, University of Colorado, Boulder, CO 80305 USA (e-mail: john.skovbekk@colorado.edu; eric.frew@colorado.edu; morteza.lahijanian@colorado.edu).

Luca Laurenti is with the Delft Center for Systems and Control, TU Delft, 2628 CD Delft, The Netherlands (e-mail: l.laurenti@tudelft.nl).

Digital Object Identifier 10.1109/TAC.2025.3532812

1558-2523 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: TU Delft Library. Downloaded on August 01,2025 at 13:58:20 UTC from IEEE Xplore. Restrictions apply.

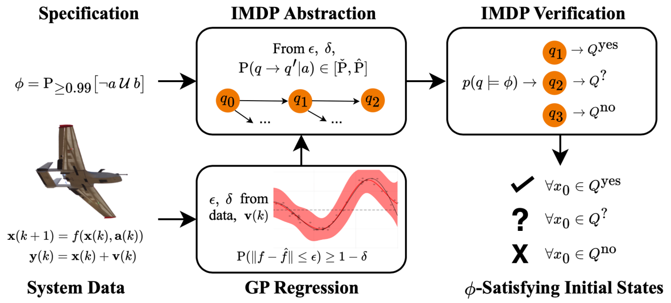

Specification
IMDP Abstraction
IMDP Verification
$\phi = \mathrm{P}_{\geq 0.99}[\neg a\, \mathcal{U}\, b]$
From ϵ, δ,
$\mathrm{P}(q \to q'|a) \in [\check{\mathrm{P}}, \hat{\mathrm{P}}]$
q_0
q_1
q_2
...
...
$p(q \models \phi) \to$
q_1 $\to Q^{\mathrm{yes}}$
q_2 $\to Q^{?}$
q_3 $\to Q^{\mathrm{no}}$
ϵ, δ from data, $\mathbf{v}(k)$
$\mathrm{P}(\|f - \hat{f}\| \leq \epsilon) \geq 1 - \delta$
$\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{a}(k))$
$\mathbf{y}(k) = \mathbf{x}(k) + \mathbf{v}(k)$
✓ $\forall x_0 \in Q^{\mathrm{yes}}$
? $\forall x_0 \in Q^{?}$
X $\forall x_0 \in Q^{\mathrm{no}}$
System Data
GP Regression
ϕ-Satisfying Initial States

Fig. 1. Overview of our learning-based, formal verification framework.

This article introduces an end-to-end data-driven verification framework that enables the use of off-the-shelf model-based verification tools and provides the required formalism to extend the guarantees to the latent system. An overview of the framework is shown in Fig. 1. Given a noisy input–output dataset of the latent system and a temporal logic specification (LTL or PCTL formula), the framework produces verification results in the form of three sets of initial states: Q^{yes}, $Q^{?}$, and Q^{no}, from which the system is guaranteed to fully, possibly, and never satisfy the specification, respectively.

Beginning with the dataset, the framework learns the dynamics via GP regression and formally quantifies the distance between the learned model and the latent system. For this step, we build on the GP regression error analysis in [15] and derive formal bounds for the constants in the error term. Specifically, by relying on the relation between reproducing kernel Hilbert spaces (RKHS) and kernel functions, we establish upper bounds for the RKHS constant of a function and the information gain. Using these bounds, the framework then constructs a finite-state, uncertain Markov decision process (MDP) abstraction that includes probabilistic bounds for every possible behavior of the system. Next, it uses an existing model checking tool to verify the abstraction against a given PCTL or LTL formula. Here, we specifically focus on PCTL formulas since a model checking tool for uncertain MDPs is readily available [9]. Finally, the framework closes the verification loop by mapping the resulting guarantees to the latent system.

The main contribution of this work is a *data-driven formal verification framework for unknown dynamical systems*. The novelties include the following.

1) Formalization of the GP regression error bound in [15] by deriving bounds for its challenging constants.
2) An abstraction procedure including derivation of optimal transition probability bounds.
3) A proof of correctness of the final verification results for the latent system.
4) An illustration of the framework on several case studies with linear, nonlinear, and switched dynamical systems as well as empirical analysis of parameter trade-offs.

A. Related Work

Recently, there has been a surge in applying formal verification methods to dynamic systems with machine learning components [16], [17]. We begin with a brief overview of model checking for verification, and then discuss its application to learning-based approaches for dynamic systems.

Verification of continuous-space systems via model checking requires constructing a finite abstraction with a simulation [11], [18]. An appropriate and widely-used abstraction model for continuous stochastic systems is the *interval-valued MDP* (IMDP) [9], [11], [19], [20], [21]. The advantage of IMDPs is their capability to incorporate multiple sources of uncertainty as well as the availability of their model checking tools with PCTL and LTL specifications [9], [22], [23]. For a system with unknown dynamics and non-Gaussian measurement noise, however, it is not clear how to construct such an abstraction. In this work, we address this problem by studying a method for formalization of all the sources of uncertainties to enable IMDP abstraction construction from data with formal guarantees.

Data-driven verification methods can be categorized to linear and nonlinear assumptions on the latent system dynamics. Existing work on linear systems include methods to verify parameterized, time-invariant dynamics subject to performance specifications [24], [25], [26], [27], to perform system identification with statistical guarantees [28], and to monitor the safety of stochastic linear systems with unmodeled errors [29]. For nonlinear systems, existing methods are largely based on sampling the latent system and using statistical bounding to generate safety or reachability guarantees [30], [31], [32], [33], [34], [35], [36]. Specifically, the DryVR framework relies on learning discrepancy functions for an unknown, deterministic system [30], and NeuReach learns a neural-network for computing probabilistic reachable sets by sampling deterministic system trajectories [34]. Recently, jointly performing reinforcement learning and formal verification can guarantee safety and stability for parameterized polynomial systems from sampled trajectories [37]. Those results, however, are limited to reachability property analysis from the sampled initial conditions, and many consider noiseless measurements of the system.

For complex objectives beyond safety and reachability, existing approaches construct finite-state abstractions from observed system data [38], [39], [40], [41], [42]. These methods require rigorous error quantification of the data-driven models to guarantee the correctness of the verification results. Such errors can be quantified with deterministic bounds [41], *probably-approximately correct* bounds [27], [33], [34], [35], [39], [43], or Bayesian probabilistic bounds [38], [40]. Among these techniques, PAC-based methods provide straightforward results, but they rely heavily on the number of samples and primarily do not consider measurement noise to the best of the authors' knowledge. In addition, unlike Bayesian methods, the guarantees they provide are conditioned on a confidence that is not rooted in a probability measure on the system. In this work, we consider unknown nonlinear systems with measurement noise, and our approach is Bayesian-based and provides guarantees based on well-defined probability measure over system trajectories.

To quantify probabilistic uncertainty, GP regression is widely employed due to its universal approximation property and mature theory [13]. For unknown dynamic systems, GP regression has been used to learn maximal invariant sets in reinforcement learning [44], [45], [46], [47], for runtime control and safety

Authorized licensed use limited to: TU Delft Library. Downloaded on August 01,2025 at 13:58:20 UTC from IEEE Xplore. Restrictions apply.

monitoring of dynamic systems [29], [48], and to construct barrier certificates for verifying system safety [49], [50]. A limitation of GP regression method is that the correctness of the predicted distribution is conditioned on that fact that the latent function is drawn from the prior process, which may be too restrictive as an assumption for various applications [14]. Furthermore, it only supports Gaussian additive noise [13]. These limitations can be relaxed if the latent function lies in the RKHS defined by the GP kernel. Then, the noise can be non-Gaussian, and more importantly, the GP error can be quantified probabilistically [15], [41], [47], [49], [50] for arbitrary latent continuous functions. A closely-related work uses deterministic RKHS-based GP errors to construct a deterministic transition system and verify against LTL specifications when there is no measurement noise [41]. Nevertheless, that and many other existing works that use the RKHS setting rely on ad-hoc parameter approximations of the error term, which, strictly speaking, revokes the correctness of the resulting guarantees. In this work, we also employ RKHS-based approach but provide a formal method for determining the error parameters, enabling hard guarantees.

II. Problem Formulation

We consider a discrete-time controlled process given by

$$\begin{aligned}\mathbf{x}(k+1) &= f(\mathbf{x}(k), \mathbf{a}(k))\\ \mathbf{y}(k) &= \mathbf{x}(k) + \mathbf{v}(k)\end{aligned} \tag{1}$$

$$k \in \mathbb{N}, \quad \mathbf{x}(k) \in \mathbb{R}^n, \quad \mathbf{a}(k) \in A, \quad \mathbf{y}(k) \in \mathbb{R}^n$$

where $A = \{a_1, \ldots, a_{|A|}\}$ is a finite set of actions, $f : \mathbb{R}^n \times A \to \mathbb{R}^n$ is a (possibly nonlinear) function that is *unknown*, and $\mathbf{y}$ is a measurement of $\mathbf{x}$ with noise $\mathbf{v}(k)$, a random variable with probability density function $p_{\mathbf{v}}$. We assume that $p_{\mathbf{v}}$ is a stationary and *conditionally R-sub-Gaussian* distribution[1] and the R parameter of the distribution is known, but the knowledge of the explicit form of the distribution is not necessary. Intuitively, $\mathbf{y}$ is a stochastic process whose behavior depends on the latent process $\mathbf{x}$, which itself is driven by actions from A.

The evolution of (1) is described using trajectories of the states and measurements. A finite *state trajectory* up to step k is a sequence of state-action pairs denoted by $\omega^k_{\mathbf{x}} = x_0 \xrightarrow{a_0} x_1 \xrightarrow{a_1} \ldots \xrightarrow{a_{k-1}} x_k$. The state trajectory $\omega^k_{\mathbf{x}}$ induces a measurement (or observation) trajectory $\omega^k_{\mathbf{y}} = y_0 \xrightarrow{a_0} y_1 \xrightarrow{a_1} \ldots \xrightarrow{a_{k-1}} y_k$ via the measurement noise process. We denote infinite-length state and measurement trajectories by $\omega_{\mathbf{x}}$ and $\omega_{\mathbf{y}}$ and the set of all state and measurement trajectories by Ω_x and Ω_y, respectively. Further, we use $\omega_{\mathbf{x}}(i)$ and $\omega_{\mathbf{y}}(i)$ to denote the ith elements of a state and measurement trajectory, respectively.

A. Probability Measure

We assume that the action at time k is chosen by (unknown) *control strategy* $\pi_{\mathbf{y}} : \Omega^k_y \to A$ based on the measurement trajectory up to time k. Note that, even though $\pi_{\mathbf{y}}$ is deterministic, given a finite state trajectory, the next action chosen by $\pi_{\mathbf{y}}$ is stochastic due to the noise process $\mathbf{v}$, which induces a probability distribution over the trajectories of the latent process $\mathbf{x}$. In particular, given a fixed initial condition $x_0 \in \mathbb{R}^n$ and a strategy $\pi_{\mathbf{y}}$, we denote with $\omega^{k,\pi_{\mathbf{y}},x_0}_{y,v_0,\ldots,v_k}$ the measurement trajectory with noise at time $0 \leq i \leq k$ fixed to v_i and relative to the inducing state trajectory Ω^k_x such that $\Omega^k_x(0) = x_0$.[2] We can then define the probability space $(\Omega^k_x, \mathcal{B}(\Omega^k_x), \mathrm{P})$, where $\mathcal{B}(\Omega^k_x)$ is the σ-algebra on Ω^k_x generated by the product topology, and P is a probability measure on the sets in $\mathcal{B}(\Omega^k_x)$ such that for a set $X \subset \mathbb{R}^n$ [52]

$$\mathrm{P}(\omega_{\mathbf{x}}(0) \in X) = \mathbf{1}_X(x_0)$$

$$\mathrm{P}(\omega_{\mathbf{x}}(k) \in X \mid \omega_{\mathbf{x}}(0) = x_0) = \int \ldots \int T\left(X \mid x, \pi_{\mathbf{y}}(\omega^{k-1,\pi_{\mathbf{y}},x_0}_{y,v_0,\ldots,v_{k-1}})\right) p_{\mathbf{v}}(v_{k-1}) \ldots p_{\mathbf{v}}(v_0) dv_{k-1} \ldots dv_0$$

where

$$\mathbf{1}_X(x) = \begin{cases} 1, & \text{if } x \in X \\ 0, & \text{otherwise} \end{cases}$$

is the indicator function, and

$$T(X \mid x, a) = \mathrm{P}(\omega_{\mathbf{x}}(k) \in X \mid \omega_{\mathbf{x}}(k-1) = x, \mathbf{a}_{k-1} = a) \tag{2}$$

is the transition kernel that defines the single step probabilities for (1). In what follows, we call $t(\bar{x} \mid x, a)$ the density function associated to the transition kernel, i.e.,

$$T(X \mid x, a) = \int_X t(\bar{x} \mid x, a) d\bar{x}.$$

Intuitively, P is defined by marginalizing T over all possible observation trajectories. Note that as the initial condition x_0 is fixed, the marginalization is over the distribution of the noise up to time $k-1$. We remark that P is also well defined for $k = \infty$ by the *Ionescu–Tulcea extension theorem* [53].

The above definition of P implies that if f were known, the transition probabilities of (1) are Markov and deterministic once the action is known. However, we stress that, even in the case where a is known, the exact computation of the above probabilities is infeasible in general as f is unknown. Next, we introduce continuity assumptions that allow us to estimate f and consequently compute bounds on the transition kernel T.

B. Continuity Assumption

Taking function f as completely unknown leads to an ill-posed problem. Hence, we assume f belongs to a RKHS, which is a Hilbert space of functions that lie in the span of a positive-definite kernel function. This is a standard assumption [14], [15] that constrains f to be a well-behaved analytical function on a compact (closed and bounded) set.

Assumption 1 (RKHS Continuity): For a compact set $W \subset \mathbb{R}^n$, let $\kappa : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be a given kernel and $\mathcal{H}_\kappa(W)$ the RKHS of functions over W corresponding to κ [14]. Let the

[1] Conditional R-sub-Gaussian distributions are those whose tail decay at least as fast as the tail of normal distribution with variance R^2 conditioned on the filtration up to the previous step. This class of distributions includes the Gaussian distribution itself and distributions with bounded support [51].

[2] Note that Ω^k_x is uniquely defined by its initial state x_0 and the value of the noise at the various time steps, i.e, $v_0, \ldots, v_k$.

Authorized licensed use limited to: TU Delft Library. Downloaded on August 01,2025 at 13:58:20 UTC from IEEE Xplore. Restrictions apply.

RKHS-norm of f be $\|f\|_\kappa$ (defined in Lemma 2 in Appendix). Then, for each $a \in A$ and $i \in \{1, \ldots, n\}$, $f^{(i)}(\cdot, a) \in \mathcal{H}_\kappa(W)$ and for a constant $B_i > 0$, $\|f^{(i)}(\cdot, a)\|_\kappa \leq B_i$, where $f^{(i)}$ is the ith component of f.

Although Assumption 1 limits f to a class of analytical functions, it is not overly restrictive as we can choose κ such that $\mathcal{H}_\kappa$ is dense in the space of continuous functions [54]. For instance, this holds for the widely-used squared exponential kernel function [13]. This assumption allows us to use GP regression to estimate f and leverage error results as we detail in Section III-A.

C. PCTL Specifications

We are interested in the behavior of the latent process $\mathbf{x}$ as defined in (1) over regions of interest $\mathcal{R} = \{r_1, \ldots, r_{|\mathcal{R}|}\}$ with $r_i \subset \mathbb{R}^n$. For example, these regions may indicate target sets that should be visited or unsafe sets that must be avoided both in the sense of physical obstacles and state constraints (e.g., velocity limits). In order to define properties over $\mathcal{R}$, for a given state x and region r_i, we define an atomic proposition $\mathfrak{p}_i$ to be true ($\top$) if $x \in r_i$, and otherwise false ($\perp$). The set of atomic propositions is given by $AP = \{\mathfrak{p}_1, \ldots, \mathfrak{p}_{|\mathcal{R}|}\}$, and the label function $L : \mathbb{R}^n \to 2^{AP}$ returns the set of atomic propositions that are true at each state.

PCTL [2] is a formal language that allows for the expression of complex behaviors of stochastic systems.

We begin by defining the syntax and semantics of PCTL specifications.

Definition 1 (PCTL Syntax): Formulas in PCTL are recursively defined over the set of atomic propositions AP in the following manner:

$$\text{State Formula} \quad \phi := \top \mid \mathfrak{p} \mid \neg\phi \mid \phi \wedge \phi \mid \mathrm{P}_{\bowtie p}[\psi]$$

$$\text{Path Formula} \quad \psi := \mathcal{X}\phi \mid \phi\,\mathcal{U}^{\leq k}\phi \mid \phi\,\mathcal{U}\phi$$

where $\mathfrak{p} \in AP$, $\neg$ is the negation operator, $\wedge$ is the conjunction operator, $\mathrm{P}_{\bowtie p}$ is the probabilistic operator, $\bowtie \in \{\leq, <, \geq, >\}$ is a relation placeholder, and $p \in [0, 1]$. The temporal operators are the $\mathcal{X}$ (Next), $\mathcal{U}^{\leq k}$ (Bounded-Until) with respect to time step $k \in \mathbb{N}$, and $\mathcal{U}$ (Until).

Definition 2 (PCTL Semantics): The satisfaction relation $\models$ is defined inductively as follows, for state formulas:

1) $x \models \top$ for all $x \in \mathbb{R}^n$;
2) $x \models \mathfrak{p} \iff \mathfrak{p} \in L(x)$;
3) $x \models (\phi_1 \wedge \phi_2) \iff (x \models \phi_1) \wedge (x \models \phi_2)$;
4) $x \models \neg\phi \iff x \not\models \phi$;
5) $x \models \mathrm{P}_{\bowtie p}[\psi] \iff p^x(\psi) \bowtie p$, where $p^x(\psi)$ is the probability that all infinite trajectories initialized at x satisfy ψ.

For a state trajectory $\omega_{\mathbf{x}}$, the satisfaction relation $\models$ for path formulas is defined as follows:

1) $\omega_{\mathbf{x}} \models \mathcal{X}\phi \iff \omega_{\mathbf{x}}(1) \models \phi$;
2) $\omega_{\mathbf{x}} \models \phi_1\mathcal{U}^{\leq k}\phi_2 \iff \exists i \leq k \quad \text{s.t.} \quad \omega_{\mathbf{x}}(i) \models \phi_2 \wedge \omega_{\mathbf{x}}(j) \models \phi_1 \ \forall j \in [0, i)$;
3) $\omega_{\mathbf{x}} \models \phi_1\mathcal{U}\phi_2 \iff \exists i \geq 0$ s.t. $\omega_{\mathbf{x}}(i) \models \phi_2 \wedge \omega_{\mathbf{x}}(j) \models \phi_1 \ \forall j \in [0, i)$.

The common operators bounded eventually $\mathcal{F}^{\leq k}$ and eventually $\mathcal{F}$ are defined, respectively, as $\mathrm{P}_{\bowtie p}[\mathcal{F}^{\leq k}\phi] \equiv \mathrm{P}_{\bowtie p}[\top\mathcal{U}^{\leq k}\phi]$, and $\mathrm{P}_{\bowtie p}[\mathcal{F}\phi] \equiv \mathrm{P}_{\bowtie p}[\top\mathcal{U}\phi]$. The globally operators $\mathcal{G}^{\leq k}$ and $\mathcal{G}$ are defined as $\mathrm{P}_{\bowtie p}[\mathcal{G}^{\leq k}\phi] \equiv \mathrm{P}_{\bar{\bowtie} 1-p}[\mathcal{F}^{\leq k}\neg\phi]$, and $\mathrm{P}_{\bowtie p}[\mathcal{G}\phi] \equiv \mathrm{P}_{\bar{\bowtie} 1-p}[\mathcal{F}\neg\phi]$, where $\bar{\bowtie}$ indicates the opposite relation, i.e, $\bar{<} \equiv >$, $\bar{\leq} \equiv \geq$, $\bar{\geq} \equiv \leq$, and $\bar{>} \equiv <$.

As an example the property, *"the probability of reaching the target by while avoiding unsafe areas, and with probability at least 0.95 of eventually reaching the secondary target if the breach area is entered, is at least 0.95"* can be expressed with the PCTL formula

$$\phi = \mathrm{P}_{\geq 0.95}\left[(\neg\mathfrak{p}_{\text{unsafe}} \wedge (\mathfrak{p}_{\text{breach}} \right.$$
$$\left. \implies \mathrm{P}_{\geq 0.95}[\mathcal{F}\mathfrak{p}_{\text{secondary}}]))\ \mathcal{U}\ \mathfrak{p}_{\text{target}}\right].$$

D. Problem Statement

The verification problem asks whether (1) satisfies a PCTL formula ϕ under all control strategies. It can also be posed as finding the set of initial states in a compact set $X \subset \mathbb{R}^n$, from which (1) satisfies ϕ.

In lieu of an analytical form of f, we assume to have a dataset $\mathrm{D} = \{(x_i, a_i, y_i)\}_{i=1}^d$ generated by (1) where y_i is the noisy measurement of $f(x_i, a_i)$, and $d \in \mathbb{N}^+$. Using this dataset, we can infer f and reason about the trajectories of (1) (via Assumption 1). The formal statement of the problem considered in this work is as follows.

Problem 1: Let $X \subset \mathbb{R}^n$ be a compact set, $\mathcal{R}$ a set of regions of interest, and AP its corresponding set of atomic propositions (as defined in Section II-C). Given a dataset D generated by (1) and a PCTL formula ϕ defined over AP, determine the initial states $X_0 \subset X$ from which (1) satisfies ϕ without leaving X.

Our approach to Problem 1 is based on constructing an IMDP abstraction (formally defined in Section III-C) of the latent process $\mathbf{x}$ using GP regression and checking if the abstraction satisfies ϕ. We begin by estimating the unknown dynamics with GP regression and, by virtue of Assumption 1, we leverage existing probabilistic error bounds on this estimate. We present upper bounds on the constants required to compute this probabilistic error and derive bounds on the transition kernel in (2) over a discretization of the space X. Then, we construct the abstraction of (1) in the form of an uncertain (interval-valued) MDP, which accounts for the uncertainties in the regression of f and the discretization of X. Finally, we perform model checking on the abstraction to get sound bounds on the probability that $\mathbf{x}$ satisfies ϕ from any initial state.

Remark 1: We remark that our IMDP abstraction construction is general, and hence, can be used for the verification of (1) against properties in other specification languages, such as LTL. We specifically focus on PCTL properties mainly because of the availability of its model checking tool for IMDPs [9].

III. Preliminaries

A. GP Regression

GP regression [13] aims to estimate an unknown function $\mathrm{f} : \mathbb{R}^n \to \mathbb{R}$ from a dataset $\mathrm{D} = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^d$ where $\mathrm{y}_i = \mathrm{f}(\mathrm{x}_i) + \mathrm{v}_i$ and v_i is a sample of a normal distribution with zero mean and $\sigma_{\mathbf{v}}^2$ variance, denoted by $\mathrm{v}_i \sim \mathcal{N}(0, \sigma_{\mathbf{v}}^2)$. The standard

Authorized licensed use limited to: TU Delft Library. Downloaded on August 01,2025 at 13:58:20 UTC from IEEE Xplore. Restrictions apply.

assumption is that f is a sample from a prior GP with zero-valued mean function and kernel function κ_0.[3]

Let X and Y be ordered vectors with all points in D such that $X_i = \mathrm{x}_i$ and $Y_i = \mathrm{y}_i$. Further, let K denote the matrix with element $K_{i,j} = \kappa_0(\mathrm{x}_i, \mathrm{x}_j)$, $k(\mathrm{x}, X)$ the vector such that $k_i(\mathrm{x}, X) = \kappa_0(\mathrm{x}, X_i)$, and $k(X, \mathrm{x})$ defined accordingly. Predictions at a new input point x are given by the conditional distribution of the prior at x given D, which is still Gaussian and with mean μ_{D} and variance σ^2_{D} given by

$$\mu_{\mathrm{D}}(\mathrm{x}) = k(\mathrm{x}, X)^T \left(K + \sigma_{\mathrm{v}}^2 I\right)^{-1} Y \tag{3}$$

$$\sigma^2_{\mathrm{D}}(\mathrm{x}) = \kappa_0(\mathrm{x}, \mathrm{x}) - k(\mathrm{x}, X)^T \left(K + \sigma_{\mathrm{v}}^2 I\right)^{-1} k(X, \mathrm{x}) \tag{4}$$

where I is the identity matrix of size d.

As mentioned previously, the noise in Process (1) is sub-Gaussian, so the standard assumption that f is a sample from the prior GP cannot be used. Hence, we cannot directly use (3) and (4) to bound the latent function. Rather, as common in [14], we rely on the relationship between GP regression and the RKHS.

B. Error Quantification

The reliance on a positive-definite kernel function is the basis for relating a GP with kernel κ with an RKHS $\mathcal{H}_\kappa$. For universal kernels (such as the squared exponential function), the associated RKHS is dense in the continuous functions on any compact set [55].

Given Assumption 1, the following proposition bounds the GP learning error when f is a function in $\mathcal{H}_\kappa$, without posing any distributional assumption on f.

Proposition 1 ([15], Theorem 2): Let W be a compact set, $\delta \in (0, 1]$, and D be a given dataset generated by f with cardinality $|\mathrm{D}| = d$. Further, let $\Gamma > 0$ be a bound on the maximum information gain of κ, i.e., $\gamma^d_\kappa \leq \Gamma$, and $B > 0$ such that $\|\mathrm{f}\|_\kappa \leq B$. Assume that v is R-sub-Gaussian and that μ_{D} and σ_{D} are found by setting $\sigma^2_{\mathrm{v}} = 1 + 2/d$ and using (3) and (4). Define $\beta(\delta) = B + R\sqrt{2(\Gamma + 1 + \log 1/\delta)}$. Then, it holds that

$$\mathrm{P}\left(\forall \mathrm{x} \in W,\ |\mu_{\mathrm{D}}(\mathrm{x}) - \mathrm{f}(\mathrm{x})| \leq \beta(\delta)\sigma_{\mathrm{D}}(\mathrm{x})\right) \geq 1 - \delta. \tag{5}$$

Proposition 1 assumes that the noise on the measurements is R-sub-Gaussian, which is more general than Gaussian noise. Nevertheless, the probability bound (5) depends on bounding the constants $\|\mathrm{f}\|_\kappa$ and γ^d_κ which are described as *challenging to compute* in the literature [56] and are often bounded according to heuristics without guarantees [46], [49]. In this work, we derive formal upper bounds for each of these terms in Section IV-B2.

C. Interval Markov Decision Processes

An IMDP is a generalization of a MDP, where the transitions under each state-action pair are defined by probability intervals [57].

Definition 3 (IMDP): An IMDP is a tuple $\mathcal{I} = (Q, A, \check{\mathrm{P}}, \hat{\mathrm{P}}, AP, L)$ where

1) Q is a finite set of states,
2) A is a finite set of actions, where $A(q)$ is the set of available actions at state $q \in Q$,
3) $\check{\mathrm{P}} : Q \times A \times Q \to [0, 1]$ is a function, where $\check{\mathrm{P}}(q, a, q')$ defines the lower bound of the transition probability from state $q \in Q$ to state $q' \in Q$ under action $a \in A(q)$,
4) $\hat{\mathrm{P}} : Q \times A \times Q \to [0, 1]$ is a function, where $\hat{\mathrm{P}}(q, a, q')$ defines the upper bound of the transition probability from state q to state q' under action $a \in A(q)$,
5) AP is a finite set of atomic propositions, and
6) $L : Q \to 2^{AP}$ is a labeling function that assigns to each state q possibly several elements of AP.

For all $q, q' \in Q$ and $a \in A(q)$, it holds that $\check{\mathrm{P}}(q, a, q') \leq \hat{\mathrm{P}}(q, a, q')$ and

$$\sum_{q' \in Q} \check{\mathrm{P}}(q, a, q') \leq 1 \leq \sum_{q' \in Q} \hat{\mathrm{P}}(q, a, q').$$

Let $\mathcal{D}(Q)$ denote the set of probability distributions over Q. Given $q \in Q$ and $a \in A(q)$, we call $\mathfrak{t}^a_q \in \mathcal{D}(Q)$ a *feasible distribution* over states reachable from q under a if the transition probabilities respect the intervals defined for each possible successor state q', i.e., $\check{\mathrm{P}}(q, a, q') \leq \mathfrak{t}^a_q(q') \leq \hat{\mathrm{P}}(q, a, q')$. We denote the set of all feasible distributions for state q and action a by $\mathfrak{T}^a_q$.

A path ω of an IMDP is a sequence of state-action pairs $\omega = q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} \ldots$ such that $a_i \in A(q_i)$ and $\hat{\mathrm{P}}(q_i, a_i, q_{i+1}) > 0$ (i.e., transitioning is possible) for all $i \in \mathbb{N}$. We denote the last state of a finite path ω^{fin} by $last(\omega^{\mathrm{fin}})$ and the set of all finite and infinite paths by $Paths^{\mathrm{fin}}$ and $Paths$, respectively. Actions taken by the IMDP are determined by a choice of strategy π which is defined below.

Definition 4 (Strategy): A strategy π of an IMDP $\mathcal{I}$ is a function $\pi : Paths^{\mathrm{fin}} \to A$ that maps a finite path ω^{fin} of $\mathcal{I}$ onto an action in A. The set of all strategies is denoted by Π.

Once an action is chosen according to a strategy, a feasible distribution needs to be chosen from $\mathfrak{T}^a_q$ to enable a transition to the next state. This task falls on the adversary function θ as defined below.

Definition 5 (Adversary): Given an IMDP $\mathcal{I}$, an adversary is a function $\theta : Paths^{\mathrm{fin}} \times A \to \mathcal{D}(Q)$ that, for each finite path $\omega^{\mathrm{fin}} \in Paths^{\mathrm{fin}}$ and action $a \in A(\mathrm{last}(\omega^{\mathrm{fin}}))$, chooses a feasible distribution $\mathfrak{t}^a_q \in \mathfrak{T}^a_{\mathrm{last}(\omega^{\mathrm{fin}})}$. The set of all adversaries is denoted by Θ.

Once a strategy is selected, the IMDP becomes an interval Markov chain. Further, choosing an adversary results in a standard Markov chain. Hence, given a strategy and an adversary, a probability measure can be defined on the paths of the IMDP via the probability of paths on the resulting Markov chain [9].

IV. IMDP Abstraction

To solve Problem 1, we begin by constructing a finite abstraction of (1) in the form of an IMDP that captures the state evolution of the system under known actions. This involves partitioning the set X into discrete regions and determining the transition probability intervals between each pair of discrete regions under each action to account for the uncertainty due to the learning and discretization processes.

[3] Extensions with nonzero mean are a trivial generalization [13].

Authorized licensed use limited to: TU Delft Library. Downloaded on August 01,2025 at 13:58:20 UTC from IEEE Xplore. Restrictions apply.

A. IMDP States and Actions

In the first step of the abstraction, the set X is discretized into a finite set of nonoverlapping regions Q_X such that $\bigcup_{q\in Q_X} q = X$. The discretization must maintain consistent labelling with the regions of interest $\mathcal{R} = \{r_i\}_{i=1}^{|\mathcal{R}|}$ where $r_i \subseteq X$, i.e., (with an abuse of the notation of L)

$$L(q) = L(r_i) \quad \text{iff} \quad L(x) = L(r_i) \quad \forall x \in q.$$

To ensure this consistency, the regions of interest are used as the foundation of the discretization. Then, we define

$$Q = Q_X \cup \{q_{\bar{X}}\}, \quad \text{where} \quad q_{\bar{X}} = \mathbb{R}^n \setminus X.$$

The regions in Q are associated with the states of the IMDP, and with an abuse of notation, $q \in Q$ indicates both a state of the IMDP and the associated discrete region. Finally, the IMDP action space is set to the action set A of (1).

B. Transition Probability Bounds

The IMDP transition probability intervals are pivotal to abstracting (1) correctly. For all $q, q' \in Q$ and action $a \in A$, the intervals should bound the true transition kernel

$$\check{\mathrm{P}}(q, a, q') \leq \min_{x\in q} T(q' \mid x, a)$$

$$\hat{\mathrm{P}}(q, a, q') \geq \max_{x\in q} T(q' \mid x, a).$$

The bounds on the transition kernel T must account for the uncertainties due to unknown dynamics and space discretization. We use GP regression and account for regression errors to derive these bounds for every (q, a, q') tuple.

1) IMDP State Images and Regression Error: We perform GP regression using the given dataset D and analyze the evolution of each IMDP state under the learned dynamics. In addition, the associated learning error is quantified using Proposition 1.

Each output component of the dynamics under an action is estimated by a separate GP, making a total of $n\,|A|$ regressions. Let $\mu^{(i)}_{a,\mathrm{D}}(x)$ and $\sigma^{(i)}_{a,\mathrm{D}}(x)$, respectively, denote the posterior mean and covariance functions for the ith output component under action a obtained via GP regression. These GPs are used to evolve q under action a, defined by image

$$Im(q, a) = \left\{\mu^{(i)}_{a,\mathrm{D}}(x) \mid x \in q,\ i \in [1, n]\right\}$$

which describes the evolution of all $x \in q$.

The *worst-case* regression error is

$$e^{(i)}(q, a) = \sup_{x\in q} |\mu^{(i)}_{a,\mathrm{D}}(x) - f^{(i)}(x, a)|$$

which provides an error upper-bound for all continuous states $x \in q$. To use Proposition 1 for probabilistic reasoning on this worst-case error, the supremum of the posterior covariance in q denoted by

$$\overline{\sigma}^{(i)}_{a,\mathrm{D}}(q) = \sup_{x\in q} \sigma^{(i)}_{a,\mathrm{D}}(x)$$

is used. Both the posterior image and covariance supremum can be calculated for each IMDP state using GP interval bounding as in [58]. Then, Proposition 1 is applied with a scalar $\epsilon^{(i)} \geq 0$

$$\mathrm{P}\left(e^{(i)}(q, a) \leq \epsilon^{(i)}\right) \geq 1 - \delta \tag{6}$$

where δ satisfies $\epsilon^{(i)} = \beta(\delta)\overline{\sigma}^{(i)}_{a,\mathrm{D}}(q)$.

2) Bounds on the RKHS Parameters: The computation of $\mathrm{P}(e^{(i)}(q, a) \leq \epsilon^{(i)})$ via Proposition 1 requires two constants: the RKHS norm $\|f^{(i)}\|_\kappa$ and information gain γ^d_κ. In this section, we provide formal upper bounds for these constants.

Proposition 2 (RKHS Norm Bound): Let $f^{(i)} \in \mathcal{H}$ where $\mathcal{H}$ is the RKHS defined by the kernel function κ. Then

$$\|f^{(i)}\|_\kappa \leq \frac{\sup_{x\in X} |f^{(i)}(x)|}{\inf_{x,x'\in X} \kappa(x, x')^{\frac{1}{2}}} \tag{7}$$

for all $x, x' \in X$.

The proof is provided in the Appendix, which relies on the relationship between the RKHS norm and the kernel function.

Proposition 2 bounds the RKHS norm by the quotient of the supremum of the function and the infimum of the kernel function on a compact set. Note that (7) requires the square root of the kernel in the denominator. Hence, so long the chosen kernel is a positive function (e.g., squared exponential), we can compute a finite upper bound for the RKHS norm. As X is a finite-dimensional compact set, it can be straightforward to calculate bounds as in the following remark.

Remark 2: An upper bound for the numerator of (7) can be estimated using the Lipschitz constant $L_{f^{(i)}}$ of $f^{(i)}$ (or an upper bound of it). That is, the supremum in the numerator can be bounded by

$$\sup_{x\in X} |f^{(i)}(x)| \leq |f^{(i)}(x')| + L_{f^{(i)}}\mathrm{diam}(X)$$

where x' is any point in X, $\mathrm{diam}(X) := \sup_{x,x'\in X} \|x - x'\|$ is the diameter of X. The term $|f^{(i)}(x')|$ can be bounded by sampling at a single point x' if the measurement noise has bounded support or can be set using a known equilibrium point in X, i.e., a point that satisfies $f(x') = x'$ and using the components therein. The Lipschitz constant itself can be estimated from data with statistical guarantees [59], [60].

The information gain γ^d_κ can be bounded using the of size of the dataset and the GP hyperparameters as in the following proposition.

Proposition 3 (Information Gain Bound): Let d be the size of the dataset D and σ_{v} the parameter used for GP regression. Then, the maximum information gain is bounded by

$$\gamma^d_\kappa \leq d \log(1 + \sigma_{\mathrm{v}}^{-2} s)$$

where $s = \sup_{x\in X} \kappa(x, x)$.

The proof is provided in the Appendix and relies on the parameters for regression and positive-definiteness of κ to employ Hadamard's inequality. While this bound on γ^d_κ is practical, it can be further improved by using more detailed knowledge of the kernel function, such as its spectral properties [61].

3) Transitions Between States in Q_X: To reason about transitions using the images of regions, the following notations are used to indicate the expansion and reduction of a region and the intersection of two regions.

Definition 6 (Region Expansion and Reduction): Given a compact set (region) $q \subset \mathbb{R}^n$ and a set of n scalars $c = \{c_1, \dots, c_n\}$, where $c_i \ge 0$, the expansion of q by c is defined as

$$\overline{q}(c) = \{x \in \mathbb{R}^n \mid \exists x_q \in q \;\; s.t. \;\; |x_q^{(i)} - \quad x^{(i)}| \le c_i \\ \forall i = \{1, \dots, n\}\}$$

and the reduction of q by c is

$$\underline{q}(c) = \{x_q \in q \mid \forall x_{\partial q} \in \partial q, \;\; |x_q^{(i)} - x_{\partial q}^{(i)}| > c_i \\ \forall i = \{1, \dots, n\}\}$$

where ∂q is the boundary of q.

The intersection between the expanded and reduced states indicate the possibility of transitioning between the states. This intersection function is defined as

$$\mathbf{1}_V(W) = \begin{cases} 1, & \text{if } V \cap W \neq \emptyset \\ 0, & \text{otherwise} \end{cases}$$

for sets V and W.

The following theorem presents bounds for the transition probabilities of Process (1) that account for both the regression error and induced discretization uncertainties using Proposition 1 and sound approximations using the preceding definitions.

Theorem 1: Let $q, q' \in Q_X$ and ϵ, $\epsilon' \in \mathbb{R}^n$ be nonnegative vectors. For any action $a \in A$, the transition kernel is bounded by

$$\min_{x \in q} T(q' \mid x, a) \\ \ge \left(1 - \mathbf{1}_{X \setminus \underline{q}'(\epsilon)}\left(Im(q,a)\right)\right) \prod_{i=1}^{n} \mathrm{P}\left(e^{(i)}(q,a) \le \epsilon^{(i)}\right)$$

and

$$\max_{x \in q} T(q' \mid x, a) \\ \le 1 - \prod_{i=1}^{n} \mathrm{P}\left(e^{(i)}(q,a) \le \epsilon'^{(i)}\right) \left(1 - \mathbf{1}_{\overline{q}'(\epsilon')}\left(Im(q,a)\right)\right).$$

Proof: The proof of Theorem 1 relies on bounding the probability of transitioning from q to q' conditioned on the learning error ϵ (ϵ') given by (6), and using the expanded (reduced) image of q with ϵ (ϵ') to find a point in q that minimizes (maximizes) this bound. We provide the proof for the upper bound on T, and the lower bound follows analogously.

Let q' denote subsequent states of q, and suppose $\omega_{\mathbf{x}}(k) \in q$ for an arbitrary timestep k. The probability of transitioning to q' under action a is upper-bounded beginning with transition kernel and using the law of total probability conditioned on the learning error

$$\max_{x \in q} \mathrm{P}(\omega_{\mathbf{x}}(k+1) \in q' \mid \omega_{\mathbf{x}}(k) = x, a) \\ = \max_{x \in q} \left(\mathrm{P}(\omega_{\mathbf{x}}(k+1) \in q' \wedge e(q,a) \le \epsilon \mid \omega_{\mathbf{x}}(k) = x)\right. \\ \left. + \mathrm{P}(\omega_{\mathbf{x}}(k+1) \in q' \wedge e(q,a) > \epsilon \mid \omega_{\mathbf{x}}(k) = x)\right) \\ = \max_{x \in q} \left(\mathrm{P}(\omega_{\mathbf{x}}(k+1) \in q' \mid e(q,a) \le \epsilon, \omega_{\mathbf{x}}(k) = x)\right. \\ \times \mathrm{P}(e(q,a) \le \epsilon) \\ \left. + \mathrm{P}(\omega_{\mathbf{x}}(k+1) \in q' \mid e(q,a) > \epsilon, \omega_{\mathbf{x}}(k) = x) \mathrm{P}(e(q,a) > \epsilon)\right).$$

Next, $\mathrm{P}(\omega_{\mathbf{x}}(k+1) \in q' \mid e(q,a) \le \epsilon, \omega_{\mathbf{x}}(k) = x)$ is upper bounded using the intersection indicator, $\mathrm{P}(\omega_{\mathbf{x}}(k+1) \in q' \mid e(q,a) > \epsilon, \omega_{\mathbf{x}}(k) = x)$ is upper bounded by one, and $\mathrm{P}(e(q,a) > \epsilon)$ is replaced by the equivalent $1 - \mathrm{P}(e(q,a) \le \epsilon)$

$$\le \max_{x \in q} \left(\mathbf{1}_{\overline{q}'}\left(Im(q,a)\right) \mathrm{P}(e(q,a) \le \epsilon)\right. \\ \left. + 1 \cdot (1 - \mathrm{P}(e(q,a) \le \epsilon))\right) \\ = \max_{x \in q} \left(\mathbf{1}_{\overline{q}'}\left(Im(q,a)\right) \prod_{i=1}^{n} \mathrm{P}(e^{(i)}(q,a) \le \epsilon_i)\right. \\ \left. + 1 - \prod_{i=1}^{n} \mathrm{P}(e^{(i)}(q,a) \le \epsilon_i)\right)$$

where the last inequality is due to fact that the components of $\mathbf{v}$ are mutually independent. Rearranging these terms, we obtain the upper bound on T. □

The width of the transition probability intervals in Theorem 1 relies on the choices of ϵ and ϵ'. The following proposition provides the optimal values for both ϵ and ϵ'.

Proposition 4: Let ∂W denote the boundary of a compact set $W \subset \mathbb{R}^n$. The distance between the upper and lower bounds in Theorem 1 is minimized if $\epsilon' = \epsilon$ and ϵ is chosen such that for each $i \in [1, n]$

$$\epsilon^{(i)} = \inf_{x \in \partial Im(q,a),\, x' \in \partial q'} |x^{(i)} - x'^{(i)}|.$$

The proof for Proposition 4 is provided in the Appendix.

The intuition for this proposition is as follows. Consider the image $Im(q,a)$ and the intersection $Im(q,a) \cap q'$. There are three possible outcomes; 1) the intersection is empty, 2) the intersection is nonempty but not equal to $Im(q,a)$, and 3) the intersection is equal to $Im(q,a)$. By examining Theorem 1, it is clear that if (2) is true, then we get trivial transition probability bounds of [0,1]. In this case, the choice of ϵ does not matter. If (1) is true, then the bounds are $[0, \hat{\mathrm{P}}(q,a,q')]$ and we have an opportunity to get a nontrivial upper bound. The best upper bound is achieved by choosing large ϵ to capture more regression error, but small enough to keep the indicator function zero. This is achieved by choosing ϵ according to Proposition 4, which corresponds to minimizing the L1 norm of $x - x'$. Finally, if (3) is true, then the bounds are $[\check{\mathrm{P}}(q,a,q'), 1]$ and the approach is similar.

4) Transitions to $q_{\bar{X}}$: Transition intervals to the unsafe state $q_{\bar{X}}$ are the complement of the transitions to the full set X, or

$$\check{\mathrm{P}}(q,a,q_{\bar{X}}) = 1 - \hat{\mathrm{P}}(q,a,X)$$
$$\hat{\mathrm{P}}(q,a,q_{\bar{X}}) = 1 - \check{\mathrm{P}}(q,a,X)$$

where $\check{\mathrm{P}}(q,a,X)$ and $\hat{\mathrm{P}}(q,a,X)$ are calculated with Theorem 1. Finally, the unsafe state has an enforced absorbing property where $\check{\mathrm{P}}(q_{\bar{X}},a,q_{\bar{X}}) = \hat{\mathrm{P}}(q_{\bar{X}},a,q_{\bar{X}}) = 1$.

The IMDP abstraction of Process (1) incorporates the uncertainty due to the measurement noise, the uncertainty of f, and the

Authorized licensed use limited to: TU Delft Library. Downloaded on August 01,2025 at 13:58:20 UTC from IEEE Xplore. Restrictions apply.

induced error from discretizing the continuous state space. The width of the transition intervals quantifies the conservativeness of the IMDP abstraction with respect to the underlying system. In the worst-case, a transition interval of [0,1] indicates that the transition is possible but provides no meaningful information about the true transition probability. Likewise, smaller intervals indicate the abstraction better models the underlying system.

V. Verification

In this section, IMDP verification against PCTL specifications is summarized, and the correctness of the verification results on the IMDP abstraction is established via Theorem 2.

A. IMDP Verification

PCTL model checking of an IMDP is a well established procedure [9]. For completeness, we present a summary of it here. Specifically, we focus on the probabilistic operator $\mathrm{P}_{\bowtie p}[\psi]$, where ψ includes the bounded until ($\mathcal{U}^{\leq k}$) or unbounded until ($\mathcal{U}$) operator, since the procedure for the next ($\mathcal{X}$) operator is analogous [9]. In what follows, ψ is assumed to be a path formula with $\mathcal{U}^{\leq k}$, which becomes $\mathcal{U}$ when $k = \infty$.

For an initial state $q \in Q$ and PCTL path formula ψ, let Q^0 and Q^1 be the set of states from which the probability of satisfying ψ is 0 and 1, respectively. These sets are determined simply by the labels of the states in Q. Further, let $\check{p}^k(q)$ and $\hat{p}^k(q)$ be, respectively, the lower-bound and upper-bound probabilities that the paths initialized at q satisfy ψ in k steps. These bounds are defined recursively by

$$\check{p}^k(q) = \begin{cases} 1, & \text{if } q \in Q^1 \\ 0, & \text{if } q \in Q^0 \\ 0, & \text{if } q \notin (Q^0 \cup Q^1) \wedge k = 0 \\ \min\limits_{a} \min\limits_{\theta_q^a} \sum_{q'} \theta_q^a(q')\check{p}^{k-1}(q'), & \text{otherwise} \end{cases} \tag{8}$$

$$\hat{p}^k(q) = \begin{cases} 1, & \text{if } q \in Q^1 \\ 0, & \text{if } q \in Q^0 \\ 0, & \text{if } q \notin (Q^0 \cup Q^1) \wedge k = 0 \\ \max\limits_{a} \max\limits_{\theta_q^a} \sum_{q'} \theta_q^a(q')\hat{p}^{k-1}(q'), & \text{otherwise.} \end{cases} \tag{9}$$

In short, the procedure exactly finds the minimizing and maximizing adversaries with respect to the equations above for each state-action pair, and then determines the minimizing and maximizing action for each state. It is guaranteed that each probability bound converges to a fix value in a finite number of steps. Hence, this procedure can be used for both bounded until ($\mathcal{U}^{\leq k}$) and unbounded until ($\mathcal{U}$) path formulas. See [9] for more details. The final result is the probability interval

$$[\check{p}^k(q), \hat{p}^k(q)] \subseteq [0, 1]$$

of satisfying ψ within k time steps for each IMDP state $q \in Q$. Similar to the transition intervals, the widths of the satisfaction probability intervals are a measure of the conservativeness of the verification results.

Given the PCTL formula $\phi = \mathrm{P}_{\bowtie p}[\psi]$, the satisfaction intervals are used to classify states as belonging to Q^{yes}, Q^{no}, or $Q^{?}$, i.e., those states that satisfy, violate and possibly satisfy ϕ. For example, when the relation $\bowtie$ in formula ϕ is $>$ and p is the threshold value inherent to ϕ

$$q \in \begin{cases} Q^{\text{yes}}, & \text{if } \check{p}^k(q) > p \\ Q^{\text{no}}, & \text{if } \hat{p}^k(q) \leq p \\ Q^{?}, & \text{otherwise.} \end{cases}$$

Thus, Q^{yes} consist of the states that satisfy ϕ, and Q^{no} consists of the states that do no satisfy ϕ for every choice of adversary and action. $Q^{?}$ is the set of indeterminant states for which no guarantees can be made with respect to ϕ due to large uncertainty.

B. Verification Extension and Correctness

The final task is to extend the IMDP verification results to Process (1) even though the IMDP does not model the measurement noise on the system. This is possible by observing that the value iteration procedures in (8) and (9) are solved by finding the *extreme* actions at each state. All other actions have outcomes that lie in the satisfaction probability intervals. The following theorem asserts that these intervals, defined for q, bound the probability that all paths initialized at $x \in q$ satisfy ψ.

Theorem 2: Let $q \in Q$ be both a region in X and a state of the IMDP abstraction constructed on dataset D, and $\check{p}(q)$ and $\hat{p}(q)$ be the lower- and upper-bound probabilities of satisfying ψ from q computed by the procedure in Section V-A. Further, let $x \in q$ be a point and $\mathrm{P}(\omega_{\mathbf{x}} \models \psi \mid \omega_{\mathbf{x}}(0) = x, \pi_{\mathbf{y}})$ be the probability that all paths initialized at x satisfy ψ under strategy $\pi_{\mathbf{y}}$ given D. Then, it holds that

$$\mathrm{P}(\omega_{\mathbf{x}} \models \psi \mid \omega_{\mathbf{x}}(0) = x, \pi_{\mathbf{y}}) \geq \check{p}(q)$$
$$\mathrm{P}(\omega_{\mathbf{x}} \models \psi \mid \omega_{\mathbf{x}}(0) = x, \pi_{\mathbf{y}}) \leq \hat{p}(q)$$

for every strategy $\pi_{\mathbf{y}}$.

Proof: We present the proof of the lower bound; the proof for the upper bound is analogous. Furthermore, in what follows for the sake of a simpler notation, we assume that $\pi_{\mathbf{y}}$ is stationary, i.e., $\pi_{\mathbf{y}} : \mathbb{R}^n \to A$. The case of time-dependent strategies follows similarly. The following lemma shows a value iteration that computes the probability of satisfying ψ from initial state x under $\pi_{\mathbf{y}}$ in k steps.

Lemma 1: Let Q^0 and Q^1 be sets of discrete regions that satisfy ψ with probability 0 and 1, respectively, and $X^0 = \cup_{q \in Q^0} q$ and $X^1 = \cup_{q \in Q^1} q$. Further, let $V_k^{\pi_{\mathbf{y}}} : \mathbb{R}^n \to \mathbb{R}$ be defined recursively as

$$V_k^{\pi_{\mathbf{y}}}(x) = \begin{cases} 1, & \text{if } x \in X^1 \\ 0, & \text{if } x \in X^0 \\ 0, & \text{if } x \notin (X^0 \cup X^1) \wedge k = 0 \\ \mathbf{E}_{v \sim p_{\mathbf{v}},\, \bar{x} \sim T(\cdot \mid x, \pi_{\mathbf{y}}(x+v))}[V_{k-1}(\bar{x})], & \text{otherwise.} \end{cases}$$

Then, it holds that $V_k^{\pi_{\mathbf{y}}}(x) = \mathrm{P}(\omega_{\mathbf{x}}^k \models \psi \mid \omega_{\mathbf{x}}(0) = x, \pi_{\mathbf{y}})$.

The proof of Lemma 1 is in the Appendix. Hence, to conclude the proof of Theorem 2, it suffices to show that for $x \in q$, $V_k^{\pi_{\mathbf{y}}}(x) \geq \check{p}^k(q)$ for every $k \geq 0$ and every $\pi_{\mathbf{y}}$. This can be shown by induction similarly to [62, Thm. 4.1]. The base case is $k = 0$, where it immediately becomes clear that

$$V_0^{\pi_{\mathbf{y}}}(x) = \mathbf{1}_{X^1}(x) = \mathbf{1}_{Q^1}(q) = \check{p}^0(q).$$

Authorized licensed use limited to: TU Delft Library. Downloaded on August 01,2025 at 13:58:20 UTC from IEEE Xplore. Restrictions apply.

For the induction step we need to show that under the assumption that for every $q \in Q$

$$\min_{x\in q} V_{k-1}^{\pi_{\mathbf{y}}}(x) \geq \check{p}^{k-1}(q)$$

holds, then $V_k^{\pi_{\mathbf{y}}}(x) \geq \check{p}^k(q)$. Assume for simplicity $x \notin (X^0 \cup X^1)$, then we have

$$\begin{aligned}
V_k^{\pi_{\mathbf{y}}}(x) &= \mathbf{E}_{v\sim p_{\mathbf{v}}, \bar{x}\sim T(\cdot|x,\pi_{\mathbf{y}}(x+v))}[V_{k-1}(\bar{x})]\\
&= \int\int V_{k-1}^{\pi_{\mathbf{y}}}(\bar{x}) t(\bar{x} \mid x, \pi_{\mathbf{y}}(x+v)) p_{\mathbf{v}}(v) d\bar{x} dv\\
&= \int \left(\sum_{q\in Q} \int_q V_{k-1}^{\pi_{\mathbf{y}}}(\bar{x}) t(\bar{x} \mid x, \pi_{\mathbf{y}}(x+v)) d\bar{x}\right) p_{\mathbf{v}}(v) dv\\
&\geq \int \left(\sum_{q\in Q} \min_{x\in q} V_{k-1}^{\pi_{\mathbf{y}}}(x) \int_q t(\bar{x} \mid x, \pi_{\mathbf{y}}(x+v)) d\bar{x}\right) p_{\mathbf{v}}(v) dv\\
&\geq \int \sum_{q\in Q} V_{k-1}^{\pi_{\mathbf{y}}}(q) T(q \mid x, \pi_{\mathbf{y}}(x+v)) p_{\mathbf{v}}(v) dv\\
&\geq \min_{a\in A} \sum_{q\in Q} \check{p}^{k-1}(q) T(q \mid x, a)\\
&\geq \min_a \min_{\theta_q^a} \sum_{q'} \theta_q^a(q') \check{p}^{k-1}(q')\\
&= \check{p}^k(q)
\end{aligned}$$

where the last inequality follows by Theorem 1 while the second to last inequality follows by the induction assumption and by the observation that the noise distribution $p_{\mathbf{v}}$ only affects the choice of the action. □

Theorem 2 extends the IMDP verification guarantees to (1), even though (1) is unknown a priori and observed with noise. We provide empirical validation of Theorem 2 in our case studies in Section VI by comparing the verification results of a known and learned system.

C. End-to-End Algorithm

Here, we provide an overview of the entire verification framework in Algorithm 3 and its complexity. The algorithm takes dataset D, set $X \subset \mathbb{R}^n$, and specification formula ϕ, and returns three sets of initial states Q^{yes}, Q^{no}, and $Q^{?}$, from which (1) is guaranteed to satisfy, violate, and possibly satisfy ϕ respectively. Assuming a uniform discretization of X, the number of states in the abstraction is $|Q| = 2^n + 1$.

First, GP regression is performed $n\,|A|$ times in Line 1 of Algorithm 3. In general, each GP regression is $\mathcal{O}(d^3)$, where d is the number of input points in the dataset. GP regression can reach a complexity closer to $\mathcal{O}(d^{2.6})$ through improved Cholesky factorization, e.g., [63], although in practice many GP regression toolboxes are fast enough. As this is repeated for each action and output component, the total complexity of performing all regressions is $\mathcal{O}(d^3\, n\, |A|)$.

Next, the GPs are used to find the image and posterior covariance supremum of region in the discretization in Algorithm 1. We perform this computation with a branch-and-bound optimization procedure introduced in [58] until one of two termination criteria are reached: a maximum search depth T, or a minimum distance between the bounds (e.g., 10^{-3}). In our case studies, we observe that the algorithm usually terminates long before the maximum search depth is reached. The complexity of this operation is $\mathcal{O}(2^T\, d^2\, n\, |Q|\, |A|)$, where T allows a tradeoff between accuracy and computation time. Hence, for a fixed T, the computations in Lines 3 and 4 are polynomial in size of the dataset d, dimensions n, abstraction $|Q|$, action set $|A|$. We note that the branch-and-bound method is exponential in T, and $|Q|$ can be exponential in the size of the dimensions if, e.g., a uniform discretization is used.

Algorithm 1: Bound State Images and Errors.

```
Input: Posterior GPs $\{\mathcal{GP}^a\}$, actions $A$, IMDP states $Q$
Output: Image and error bounds $Im$, $e$
1: $Im \leftarrow \{\}, e \leftarrow \{\}$
2: for $q \in Q, a \in A$ do
3:     $Im(q,a) \leftarrow$ OverapproximateImage$(q, \mathcal{GP}^a)$
4:     $e(q,a) \leftarrow$ BoundError$(q, \mathcal{GP}^a)$
   return $Im$, $e$
```

Algorithm 2: Calculate Transition Intervals.

```
Input: Image and error bounds $Im$, $e$, actions $A$, IMDP
  states $Q$
Output: Transition interval matrices $\check{\mathrm{P}}$, $\hat{\mathrm{P}}$
1: $\check{\mathrm{P}} \leftarrow \{\}, \hat{\mathrm{P}} \leftarrow \{\}$
2: for $(q,q') \in Q \times Q, a \in A$ do
3:     $\epsilon \leftarrow$ Proposition 4
4:     $\check{\mathrm{P}}(q,a,q'), \hat{\mathrm{P}}(q,a,q') \leftarrow$ Theorem 1 using $Im, e$
   return$\check{\mathrm{P}}$, $\hat{\mathrm{P}}$
```

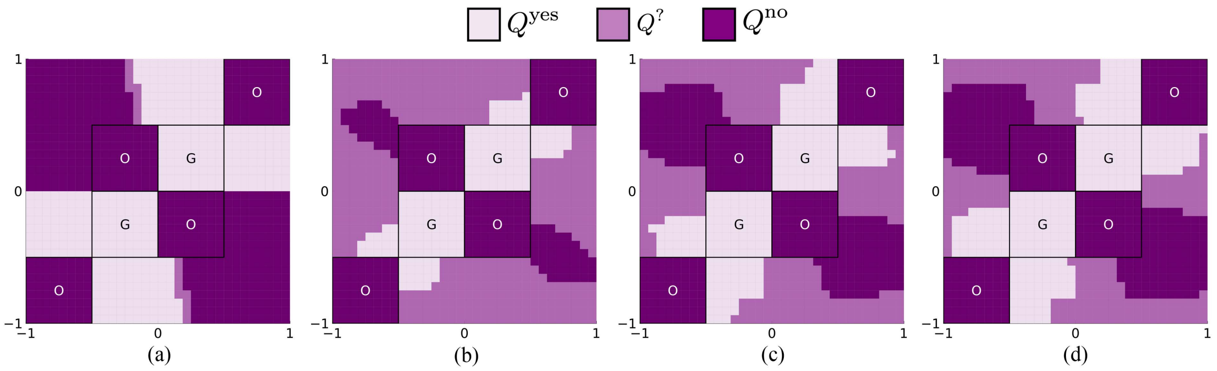
The abstraction is completed by calculating the transition probability intervals between each pair of discrete states under each action using Theorem 1 and Proposition 4 (see Lines 2–4) in Algorithm 2. The calculation of the transition probability intervals between every state-action state pair is $\mathcal{O}(|Q|^2\, |A|) = \mathcal{O}(2^{2n}\, |A|)$, which is standard for IMDP construction.

Finally, we use an existing tool for verifying the abstraction IMDP against specification formula ϕ with its inherent operator $\bowtie$ and threshold value p (see Line 6). The correctness of the obtained results is guaranteed by Theorem 2. PCTL model checking of IMDP is polynomial in the size of the state-action pairs and length of the PCTL formula [9]. Hence, the introduction of a learned component to abstraction-based verification does not fundamentally make the problem harder, as the exponential influence of the discretization of the state-space (2^{2n}) is still dominant.

VI. Case Studies

We demonstrate the efficacy of our data-driven verification framework through several case studies. We begin with verifying a single-mode linear system and compare it with a barrier function approach. Then, we show the influence of different parameters on the quality of the verification results and

Authorized licensed use limited to: TU Delft Library. Downloaded on August 01,2025 at 13:58:20 UTC from IEEE Xplore. Restrictions apply.



(a) (b) (c) (d)

Fig. 2. Unbounded ($k = \infty$) verification results for the known system and learned system using increasingly large datasets subject to $\phi = \mathrm{P}_{\geq 0.95}[\neg O \, \mathcal{U} \, G]$. (a) Known system. (b) 100 data points. (c) 500 data points. (d) 2000 data points.

Algorithm 3: Learning, Abstraction, and Verification.

```
Input: Data D, set $X$, spec. $\phi$, $AP$, label fcn. $L$
Output: Initial states $X_0$ that satisfy $\phi$
$\{\mathcal{GP}^a\} \leftarrow$ GP regressions for each $a$ using D
$Q \leftarrow$Partition$(X) \cup \{q_{\bar{X}}\}$
$Im, e \leftarrow$ Algorithm 1
$\check{\mathrm{P}}, \hat{\mathrm{P}} \leftarrow$ Algorithm 2
$\mathcal{I} \leftarrow$AssembleIMDP$(Q, A, \check{\mathrm{P}}, \hat{\mathrm{P}}, AP, L)$
$Q^{\text{yes}}, Q^{\text{no}}, Q^{?} \leftarrow$VerifyIMDP$(\mathcal{I}, \phi)$
```

computation times. Finally, we demonstrate the efficacy of our approach on switching and nonlinear systems.

For each demonstration, we specify the compact set X and use a uniform-grid discretization of X of cells with side length Δ. The synthetic datasets are generated by uniformly sampling states from X, propagating through f, and adding Gaussian noise with zero mean and 0.01 standard deviation. The squared-exponential kernel with length scale 1 and scale factor 1 and the zero mean function are used as the priors for GP regression, and we selected an upper-bound for the numerator for each example to employ Proposition 7. An implementation of our tool is available online.[4]

A. Linear System

First, we perform a verification of an unknown linear system with dynamics defined on $\mathbb{R}^2$

$$\mathbf{x}(k+1) = \begin{bmatrix} 0.4 & 0.1 \\ 0.0 & 0.5 \end{bmatrix} \mathbf{x}(k), \quad \mathbf{y}(k) = \mathbf{x}(k) + \mathbf{v}(k)$$

and regions of interest with labels G and O as shown in Fig. 2. The PCTL formula is $\phi = \mathrm{P}_{\geq 0.95}[\neg O \, \mathcal{U} \, G]$, which states that a path initializing at an initial state does not visit O until G is reached with a probability of at least 95%. The verification result using the known dynamics is shown in Fig. 2(a), which provides a basis for comparing the results of the learning-based approach. This baseline shows the initial states that belong to Q^{yes}, Q^{no}, and the indeterminate set $Q^{?}$. Some states belong to $Q^{?}$ due to the discretization, and subsets of $Q^{?}$ either satisfy or violate ϕ. Fig. 2(a) asserts that most discrete states ideally belong to either Q^{yes} or Q^{no}.

The verification results of the unknown system with various dataset sizes are shown in Fig. 2(b) to (d). With 100 data points, the learning error dominates the quality of the verification result as compared to the baseline. Nevertheless, even with such a small dataset (used to learn f), the framework is able to identify a subset of states belonging to Q^{yes} and Q^{no} as shown in Fig. 2(b). Fig. 2(c) and (d) shows these regions growing when using 500 and 2000 datapoints, respectively. The set of indeterminate states, $Q^{?}$, begins to converge to the baseline result but the rate at which it does slows with larger datasets. The computation times for the datasets with 100, 500, and 2000 points were 2.9, 10.3, and 140 s, respectively.

Compared to the abstraction built using the known system, the learning-based abstractions are 48, 24, and 16 times more conservative, respectively, in terms of average transition interval widths. In the limit of infinite data, the error could be driven to zero even with imperfect RKHS constant approximations, which would cause the learning-based results to converge to the known results in Fig. 2(a). Additional uncertainty reduction depends on refining the states in $Q^{?}$ through further discretization at the expense of additional memory and computational time. Below, we examine the tradeoffs between discretization resolution with the quality of the final results and the effect on total computation time.

1) Comparison With Barrier Functions:

We compare our approach with a barrier method based on sum-of-squares (SOS) polynomials with barrier degree eight [50], [64]. Thus, the barrier is defined on $\mathbb{R}^2$, the initial set is one of $q \in Q$, and the unsafe set is the union of $\mathbb{R}^2 \setminus X$ and O. The key limitations of the barrier method are threefold. First, barrier functions are generated with respect to an initial set, so the generation of the barrier is repeated for every discrete state. Second, barrier functions are unable to generate meaningful results when the initial state is adjacent to an unsafe state due to existing methods finding only smooth barriers (such as a polynomial function). Finally, the barrier methods use the uncertainty from Proposition 4 as a confidence wrapper, meaning the results hold with a confidence. In this case study, we set the confidence to 95%.

The methods are compared first using the true system, and then using 500 datapoints to learn the system. The state color in Fig. 3 indicates satisfaction using our approach, and the state

[4] [Online]. Available: https://github.com/aria-systems-group/TransitionIntervals.jl

Authorized licensed use limited to: TU Delft Library. Downloaded on August 01,2025 at 13:58:20 UTC from IEEE Xplore. Restrictions apply.

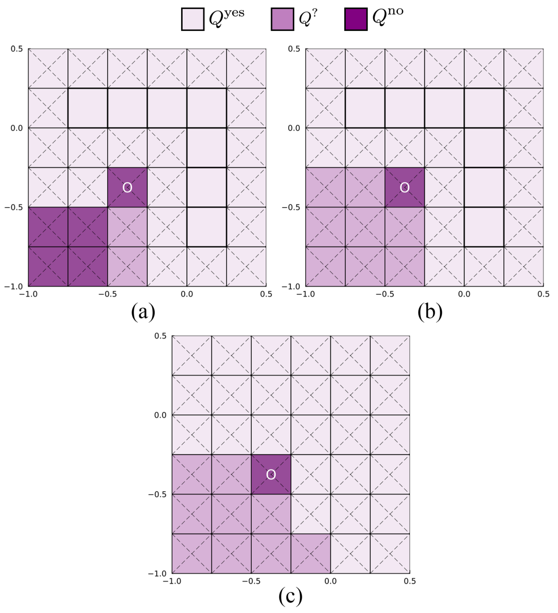


Fig. 3. Verification using our approach (colors) and SOS barriers (outlines) for the known and learned systems. Bold outlines indicate a satisfying barrier is found for that state, while dashed Xs indicate the barrier does not satisfy safety. (a) Known system, $k = 1$. (b) Learned system, $k = 1$. (c) Learned system, $k = 5$.

outlines indicate those using the barrier method. Our approach provided more safety guarantees in both the known and data-driven settings, especially beyond one step. After five steps, the barrier-based method cannot guarantee satisfaction anywhere. The complete runtimes for the learned results are 0.82 seconds for our approach and 12 s for the barrier method. Our method overcomes each of the limitations of the barrier approach, while running ten times faster.

B. Parameter Considerations

The verification results on a linear system and computation times are compared for varying the learning error and discretization parameters, ϵ and Δ, respectively, and the efficacy of using Proposition 4 to choose ϵ is highlighted. The unknown linear system is given by

$$\mathbf{x}(k+1) = \begin{bmatrix} 0.8 & 0.5 \\ 0.0 & 0.5 \end{bmatrix} \mathbf{x}(k), \quad \mathbf{y}(k) = \mathbf{x}(k) + \mathbf{v}(k).$$

The specification is the safety formula $\mathrm{P}_{\geq 0.95}[\mathcal{G}\, X]$, i.e., the probability of remaining within X is at least 95%.

We used 200 datapoints for the verification of this system. To compare the effect of ϵ, we manually selected values to be applied for every transition and compared the results with using the criterion in Proposition 4. The discretization fineness is studied by varying Δ to produce abstractions consisting of 17 states up to 10 001 states.

Fig. 4 compares the average satisfaction probability interval sizes defined by

$$\bar{p} = \frac{1}{|Q|} \sum_{q \in Q} \hat{p}(q) - \check{p}(q)$$

where a smaller value indicates more certain (less-conservative) intervals, i.e., upper bound and lower bound probabilities are similar. As the uniform choice of ϵ approaches 0.15, the average probability interval size decreases although this trend reverses as ϵ decreases further. This is due to ϵ becoming too small to get nonzero regression error probabilities from Proposition 1. However, by applying Proposition 4, we obtain an optimal value for ϵ for every region pair, resulting in the smallest interval averages $\bar{p}$ for every discretization.

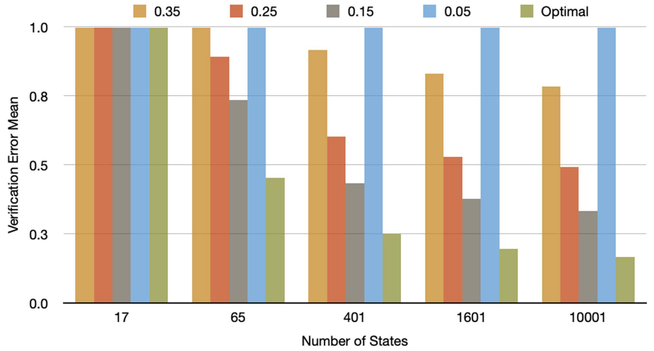


Fig. 4. Effect of the size of discretization and ϵ on the average satisfaction probability interval size (verification error).

TABLE I
COMPONENT COMPUTATION TIMES FOR AN INCREASING NUMBER OF STATES

States	Computation Time (s)		
	Disc. + Images	Transitions	Verification
65	0.202	0.000 691	0.001 04
401	1.3	0.0191	0.000 358
1,601	5.39	0.309	0.001 27
10,001	32.3	12.2	0.008 44
40,001	135.0	204.0	0.0858
62,501	208.0	516.0	0.151

Components include discretization and image over-approximations, transition probability interval calculations, and final verification times in seconds.

Note that there is an asymptotic decay in the average probability interval size as the discretization gets finer. This however comes at a higher computation cost. The total computation times for framework components are shown in Table I. The most demanding component involves discretizing X and determining the image over-approximation and error upper bound of each state as discussed in Section IV-B. The transition probability interval calculations involve checking the intersections between the image over-approximations and their respective target states. The unbounded verification procedure is fast relative to the time it takes to construct the IMDP abstraction.

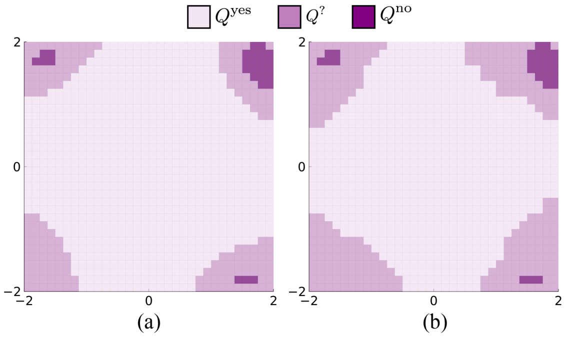
C. Switched System

We extend the previous example to demonstrate the verification of a system with multiple actions. The unknown system $f(\mathbf{x}(k), a_i) = A_i \mathbf{x}(k)$ has two actions $A = \{a_1, a_2\}$ where

$$A_1 = \begin{bmatrix} 0.8 & 0.5 \\ 0.0 & 0.5 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0.5 & 0 \\ -0.5 & 0.8 \end{bmatrix}.$$

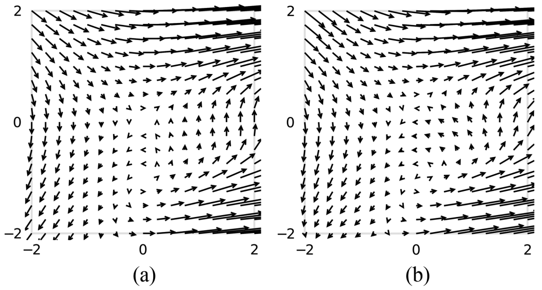
Verification was performed subject to $\phi = \mathrm{P}_{\geq 0.95}[\mathcal{G}^{\leq k}\, X]$ (the probability of remaining in X within the k time steps is at least

Authorized licensed use limited to: TU Delft Library. Downloaded on August 01,2025 at 13:58:20 UTC from IEEE Xplore. Restrictions apply.



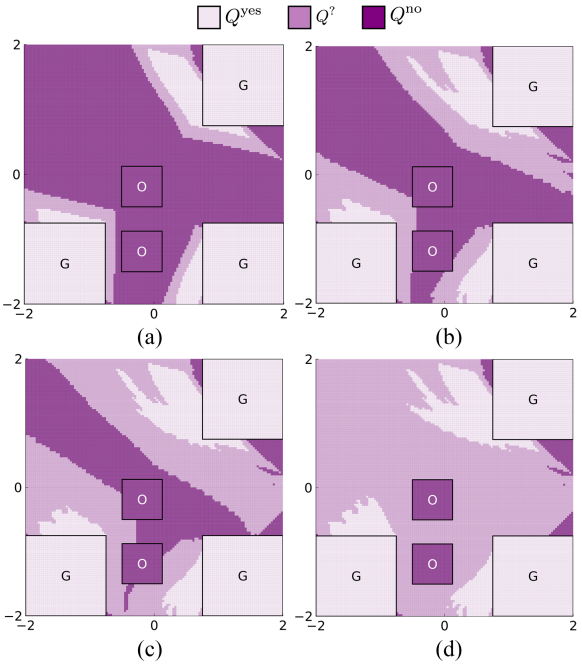
(a) (b)

Fig. 5. Verification results for the switched system subject to $\phi = \mathrm{P}_{\geq 0.95}[\mathcal{G}^{\leq k} X]$. (a) $k = 1$. (b) $k = \infty$.



(a) (b)

Fig. 6. Learned vector field for the autonomous nonlinear system. (a) True vector field. (b) Learned vector field.



(a) (b) (c) (d)

Fig. 7. Verification results for the nonlinear system subject to $\phi = \mathrm{P}_{\geq 0.95}[O\,\mathcal{U}^{\leq k} G]$. (a) $k = 1$. (b) $k = 2$. (c) $k = 3$. (d) $k = \infty$.

95%) using 400 datapoints for each action to construct the GPs. The $k = 1$ and $k = \infty$ results, in Fig. 5(a) and (b), respectively, show the regions where the system satisfies ϕ regardless of which action is selected. The framework can handle a system with an arbitrary number of actions so long there is data available for each action.

D. Autonomous Nonlinear System

We perform verification on the unknown nonlinear system

$$f(\mathbf{x}(k)) = \mathbf{x}(k) + 0.2 \begin{bmatrix} \mathbf{x}^{(2)}(k) + (\mathbf{x}^{(2)}(k))^2 e^{\mathbf{x}^{(1)}(k)} \\ \mathbf{x}^{(1)}(k) \end{bmatrix}$$

with the vector field shown in Fig. 6(a). The system is unstable about its equilibrium points at (0,0) and $(0, -1)$ and slows as it approaches them. The flow enters the set X in the upper-left quadrant, and exits in the others. This nonlinear system could represent a closed-loop control system with dangerous operating conditions near the equilibria that should be avoided.

A 2000-point dataset is used for GP regression with the learned vector fields shown in Fig. 6(b). The specification $\phi = \mathrm{P}_{\geq 0.95}[\neg O\,\mathcal{U}^{\leq k} G]$ commands to avoid O and remain within X until G is reached within k steps. With $\Delta = 0.03125$, the uniform discretization yields 16 384 discrete states, which culminated into an end-to-end verification runtime of 38 minutes. The majority of this time involved computing the images and error bounds for each discrete state, which is the only parallelized part of the framework.

For $k = 1$ shown in Fig. 7(a), Q^{yes} consists of the G regions and nearby states that transition to the G regions with high probability. The size of the $Q^{?}$ buffer between Q^{yes} and Q^{no} provides a qualitative measure of the uncertainty embedded in the abstraction. As k increases, Q^{yes} modestly grows while Q^{no} shrinks. The $k = \infty$ result shown in Fig. 7(d) indicates where guarantees of satisfying or violating ϕ can be made over an unbounded horizon. There are many more states in Q^{yes} as compared to even the $k = 3$ result. The increase of $Q^{?}$ from $k = 1$ to $k = \infty$ is due to the propagating learning and discretization uncertainty over longer horizons. The discretization resolution plays an important role in the size of $Q^{?}$, as the uncertain regions are easily induced by discrete states with possible self-transitions, which is mitigated with a finer discretization.

E. 3-D Closed-Loop Dubin's Car

Finally, we consider a discrete-time form of the Dubin's car model

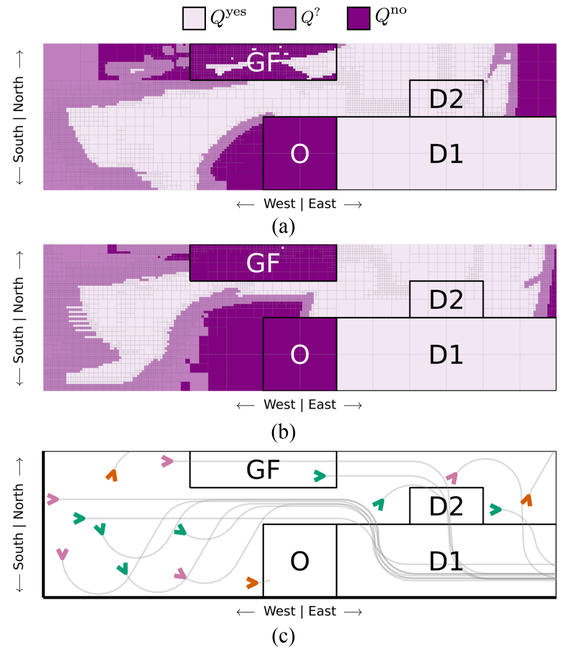
$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \Delta T \begin{bmatrix} \operatorname{sinc}(\tilde{u}) v \cos(\theta + \tilde{u}) \\ \operatorname{sinc}(\tilde{u}) v \sin(\theta + \tilde{u}) \\ u \end{bmatrix}$$

where $x \in [-1, 3]$ and $y \in [0, 1]$ are the position, $\theta \in [-0.6, 3.8]$ is the heading angle, u controls the change in heading angle θ, $\tilde{u} = 0.5 u \Delta T$, v is the constant speed, and ΔT is the time discretization parameter.

The system models an aircraft that evolves in the environment shown in Fig. 8(c) that is equipped with a feedback controller π designed to satisfy the following specification: "avoid obstacle O and reach a goal area $D1$; if the aircraft enters the geofenced area GF, it has to enter area $D2$ before reaching $D1$." The PCTL formula for verification is

$$\phi = \mathrm{P}_{\geq 0.95}\left[(\neg O \wedge (GF \Rightarrow \mathrm{P}_{\geq 0.95}[\mathcal{F}\, D2]))\ \mathcal{U}\, D1\right].$$

Authorized licensed use limited to: TU Delft Library. Downloaded on August 01,2025 at 13:58:20 UTC from IEEE Xplore. Restrictions apply.



(a)

(b)

(c)

Fig. 8. Verification results for the Dubin's car system at different initial headings subject to $\mathsf{P}_{\geq 0.95}[(\neg O \wedge (GF \Rightarrow \mathsf{P}_{\geq 0.95}[\mathcal{F}\, D2]))\,\mathcal{U}\, D1]$. (a) $\theta_0 = \pi/2$ →. (b) $\theta_0 = 9\pi/10$ →. (c) Trajectories with initial heading colored to indicate prior satisfaction (>), violation (>), or indeterminant (>) guarantees according to the framework.

To verify the (unknown) system, a dataset consisting of 100 000 datapoints was generated. To increase the speed of the framework with this large dataset, GP regression was performed *locally*, which has been shown to be improve efficiency in higher dimensions [65]. For each discrete region, the nearest 1000 datapoints according to a suitable $SE(2)$ metric are used for local dynamics modelling. In the same vein, local RKHS constants are computed for each region. An automatic refinement procedure was employed to repeatedly refine the subset of $Q^{?}$ that may reach Q^{yes}, with 12 refinement steps in total taking 35 hours. With the initial discretization consisting of 280 states, 21.4% and 17.5% of all states by volume belong to Q^{yes} and Q^{no}, respectively. After the 12 refinement steps, the discretization consists of 309 092 states with 49.3% and 26.4% belonging to Q^{yes} and Q^{no}, respectively.

Fig. 8 presents the verification results in the x and y (North and East) dimensions for two initial headings θ_0, although results were generated for all headings within $[-0.6, 3.8]$ radians. Fig. 8(c) shows trajectories of the system with controller π with indicated initial headings that are colored according to their performance guarantee. The classification of each initial state as satisfying or violating the specification intuitively follows a pattern depending on the initial heading. When this heading is eastward, as seen in Fig. 8(a), the controller can steer the system around the obstacle O if there is enough clearance. However, when the initial region is South–Eastward, shown in Fig. 8(b), the size of Q^{no} around the obstacle grows as there is not enough time to avoid the obstacle.

While the controller was designed with the implication $GF \Rightarrow \mathsf{P}_{\geq 0.95}[\mathcal{F}\, D2]$ in mind, the verification procedure identifies shortcomings in the controller performance. For example, when the initial heading is south-eastward [see Fig. 8(b)] or the system is initialized in GF, in most regions there is no guarantee that the system reaches $D2$ with at least 95% probability. Of the two trajectories that pass through GF with $\theta_0 = \pi/2$ in Fig. 8(c), one is classified as satisfying ϕ and the other classified as indeterminant. This is due to the eastern regions of GF in Fig. 8(a) having a high probability of reaching $D2$, which satisfies the implication. The western portion of GF, in addition to the majority of GF in the other case, does not have a minimum probability that satisfies this relation. However, this does not preclude the system from actually achieving the task as seen in Fig. 8(c), since the maximum probability of reaching $D2$ is above zero but not guaranteed to be at least 95%. In fact, further refinement of the abstraction to reduce the transition uncertainty may result in greater portions of GF satisfying the requirement of reaching $D2$.

VII. Conclusion

We present a verification framework for unknown dynamic systems with measurement noise subject to PCTL specifications. Through learning with GP regression and sound IMDP abstraction, we can achieve guarantees on satisfying (or violating) complex properties when the system dynamics are a priori unknown. The maturation of this framework will address the challenges of scaling to higher dimensions, which include the reliance on larger datasets and more discrete states in the abstraction, which can lead to the state-explosion dilemma.

Extensions to the framework could allow for synthesis with measurement noise, general and nonlinear measurement models, and the scaling issues of GP regression and the bounding of state images and posterior covariance suprema. While effective for GP regression, our interval-based abstraction method may be applied to other types of learning-based approaches provided sound probabilistic errors are available.

Appendix A
Proof of Proposition 2

Proof: The proof relies on the following lemma, which gives equivalent conditions involving a function f in an RKHS $\mathcal{H}$, its RKHS norm $\|f\|_\kappa$, and the associated kernel κ.

Lemma 2 (Theorem 3.11 from [66]): Let $\mathcal{H}$ be an RKHS on X with reproducing kernel κ and let $\mathrm{f} : X \to \mathbb{R}$ be a function. Then, the following are equivalent:

1) $\mathrm{f} \in \mathcal{H}$;
2) there exists a constant $c \geq 0$ such that, for every finite subset $F = \{x_1, \ldots x_n\} \subseteq X$, there exists a function $h \in \mathcal{H}$ with $\|h\|_\kappa \leq c$ and $\mathrm{f}(x_i) = h(x_i)$ for all $i = 1, \ldots n$;
3) there exists a constant $c \geq 0$ such that the function $c^2\kappa(x, x') - \mathrm{f}(x)\mathrm{f}(x')$ is a kernel function.

Moreover, if $\mathrm{f} \in \mathcal{H}$ then $\|\mathrm{f}\|_\kappa$ is the least c that satisfies the inequalities in 2 and 3.

Let $\mathrm{f} : X \to \mathbb{R}$ reside in the RKHS $\mathcal{H}$ defined by κ. Note that any c that satisfies 3 in Lemma 2 also satisfies 2. due to the assumption that $\mathrm{f} \in \mathcal{H}$ and choosing $h = \mathrm{f}$.

A kernel function κ is positive semidefinite if and only if, for all finite subsets $\{x_1, \ldots x_n\} \subset X$ and real vectors $\{b_1, \ldots b_n\} \in \mathbb{R}^n$, $\sum_i \sum_j \kappa(x_i, x_j) b_i b_j \geq 0$ which the given

Authorized licensed use limited to: TU Delft Library. Downloaded on August 01,2025 at 13:58:20 UTC from IEEE Xplore. Restrictions apply.

kernel satisfies. By Lemma 2, there exists a constant $c \geq 0$ such that

$$\begin{aligned}
&\sum_i \sum_j (c^2 \kappa(x_i, x_j) - \mathrm{f}(x_i)\mathrm{f}(x_j))b_i b_j \\
&= \sum_i \sum_j \kappa(x_i, x_j) b_i b_j - c^{-2}\mathrm{f}(x_i)\mathrm{f}(x_j) b_i b_j \\
&= \sum_i \sum_j \kappa(x_i, x_j) b_i b_j - \sum_i \sum_j c^{-2}\mathrm{f}(x_i)\mathrm{f}(x_j) b_i b_j \geq 0
\end{aligned}$$

which leads to

$$\sum_i \sum_j \kappa(x_i, x_j) b_i b_j \geq \sum_i \sum_j c^{-2}\mathrm{f}(x_i)\mathrm{f}(x_j) b_i b_j.$$

Assume $\kappa(x, x') \geq 0$. Then choose $c \geq 0$ such that, $\forall i, j$

$$\kappa(x_i, x_j) \geq |c^{-2}\mathrm{f}(x_i)\mathrm{f}(x_j)|$$
$$c^2 \geq \frac{|\mathrm{f}(x_i)\mathrm{f}(x_j)|}{\kappa(x_i, x_j)}$$

which obligates the choice

$$c^2 = \sup_{x, x' \in X} \frac{|\mathrm{f}(x)\mathrm{f}(x')|}{\kappa(x, x')} \leq \frac{\sup_{x \in X} f(x)^2}{\inf_{x, x' \in X} \kappa(x, x')}.$$

By Lemma 2, the right-hand side is an upper bound of $\|\mathrm{f}\|_\kappa^2$. □

Appendix B
Proof of Proposition 3

Proof: The proof relies on Hadamard's determinant bounding inequality and $s = \sup_{x \in X} \kappa(x, x)$. Let K denote the kernel matrix from d input points and kernel κ. Then, the maximum information gain is bounded by

$$\begin{aligned}
\gamma_\kappa^d &:= \max_K \log |I + \sigma^{-2} K| \leq \max_K \log \prod_{i=1}^d |1 + \sigma^{-2} K_{i,i}| \\
&\leq \log \prod_{i=1}^d |1 + \sigma^{-2} s| = \sum_{i=1}^d \log(1 + \sigma^{-2} s) \\
&= d \log(1 + \sigma^{-2} s)
\end{aligned}$$

where $|\cdot|$ is the matrix determinant function. □

Appendix C
Proof of Proposition 4

Proof: Let $\mathrm{P}(\epsilon_i) = (1 - \delta_i(\epsilon_i))$ denote the CDF of the probabilistic regression error. In order to maximize the upper bound in Theorem 1, the indicator function should return 1 and the product with $\mathrm{P}(\epsilon_i)$ should be maximizing. The indicator function returns 1 if $Im(q, a) \cap \underline{q}' = Im(q, a)$. The most q' can shrink is

$$\inf_{a \in \partial Im(q,a), b \in \partial q'} |a - b|.$$

Since $\mathrm{P}(\epsilon_i)$ is nondecreasing, the choice of ϵ is maximizing. The proof for the minimizing case is similar. □

Appendix D
Proof of Lemma 1

Proof: Assume, for simplicity and w.l.o.g., that $\pi_{\mathbf{y}}$ is stationary. Then $V_k^{\pi_{\mathbf{y}}} : \mathbb{R}^n \to \mathbb{R}$ is defined recursively as

$$V_k^{\pi_{\mathbf{y}}}(x) = \begin{cases} 1, & \text{if } x \in X^1 \\ 0, & \text{if } x \in X^0 \\ 0, & \text{if } x \notin (X^0 \cup X^1) \wedge k = 0 \\ \mathbf{E}_{v \sim p_{\mathbf{v}}, \bar{x} \sim T(\cdot | x, \pi_{\mathbf{y}}(x+v))}[V_{k-1}(\bar{x})], & \text{otherwise.} \end{cases}$$

To prove Lemma 1, we need to show that

$$V_k^{\pi_{\mathbf{y}}}(x) = \mathrm{P}(\omega_{\mathbf{x}}^k \models \psi \mid \omega_{\mathbf{x}}(0) = x, \pi_{\mathbf{y}})$$

that is the probability that a path of length k of Process (1) initialized at x is equal to $V_k^{\pi_{\mathbf{y}}}(x)$.

The proof is by induction over the length of the path. The base case is

$$\mathrm{P}(\omega_{\mathbf{x}}^0 \models \psi \mid \omega_{\mathbf{x}}^0(0) = x, \pi_{\mathbf{y}}) = \mathbf{1}_{X^1}(x) = V_0^{\pi_{\mathbf{y}}}(x).$$

Then, to conclude the proof we need to show that under the assumption that for all $x \in \mathbb{R}^n$

$$\mathrm{P}(\omega_{\mathbf{x}}^{k-1} \models \psi \mid \omega_{\mathbf{x}}(0) = x) = V_{k-1}^{\pi_{\mathbf{y}}}(x),$$

it holds that for all $x \in \mathbb{R}^n$

$$\mathrm{P}(\omega_{\mathbf{x}}^k \models \psi \mid \omega_{\mathbf{x}}(0) = x, \pi_{\mathbf{y}}) = V_k^{\pi_{\mathbf{y}}}(x).$$

Call $\bar{X} = X \setminus (X^0 \cup X^1)$, that is the complement of set $X^0 \cup X^1$, and define notation

$$\omega_{\mathbf{x}}^k([i, k-1]) \in X := \forall j \in [i, k-1], \omega_{\mathbf{x}}^k(j) \in X.$$

Assume w.l.o.g. that $x \in \bar{X}$. The final part of the proof is

$$\begin{aligned}
&\mathrm{P}(\omega_{\mathbf{x}}^k \models \psi \mid \omega_{\mathbf{x}}(0) = x, \pi_{\mathbf{y}}) \\
&= \sum_{i=1}^k \mathrm{P}(\omega_{\mathbf{x}}^k(i) \in X^1 \wedge \omega_{\mathbf{x}}^k([1, i-1]) \in \bar{X} \mid \omega_{\mathbf{x}}^k(0) = x, \pi_{\mathbf{y}}) \\
&= \mathrm{P}(\omega_{\mathbf{x}}^k(1) \in X^1 \mid \omega_{\mathbf{x}}^k(0) = x, \pi_{\mathbf{y}}) \\
&\quad + \sum_{i=2}^k \mathrm{P}(\omega_{\mathbf{x}}^k(i) \in X^1 \wedge \omega_{\mathbf{x}}^k([1, i-1]) \in \bar{X} \mid \omega_{\mathbf{x}}^k(0) = x, \pi_{\mathbf{y}}) \\
&= \iint_{X^1} t(\bar{x} \mid x, \pi_{\mathbf{y}}(x+v)) p_{\mathbf{v}}(v) d\bar{x} dv \\
&\quad + \sum_{i=2}^k \iint_{\bar{X}} \mathrm{P}(\omega_{\mathbf{x}}^k(i) \in X^1 \wedge \omega_{\mathbf{x}}^k([2, i-1]) \in \bar{X} \mid \\
&\qquad\qquad \omega_{\mathbf{x}}^k(1) = \bar{x}, \pi_{\mathbf{y}})\, t(\bar{x} \mid x, \pi_{\mathbf{y}}(x+v)) p_{\mathbf{v}}(v) d\bar{x} dv \\
&= \iint_{X^1} t(\bar{x} \mid x, \pi_{\mathbf{y}}(x+v)) p_{\mathbf{v}}(v) d\bar{x} dv \\
&\quad + \sum_{i=1}^{k-1} \iint_{\bar{X}} \mathrm{P}\left(\omega_{\mathbf{x}}^{k-1}(i) \in X^1 \wedge \omega_{\mathbf{x}}^{k-1}([1, i-1]) \in \bar{X} \mid \right. \\
&\qquad\qquad \left. \omega_{\mathbf{x}}^{k-1}(0) = \bar{x}, \pi_{\mathbf{y}}\right)\ t(\bar{x} \mid x, \pi_{\mathbf{y}}(x+v)) p_{\mathbf{v}}(v) d\bar{x} dv \\
&= \iint \left(\mathbf{1}_{X^1}(\bar{x}) + \mathbf{1}_{\bar{X}}(\bar{x})\, \mathrm{P}(\omega_{\mathbf{x}}^{k-1} \models \psi \mid \omega_{\mathbf{x}}(0) = \bar{x}, \pi_{\mathbf{y}})\right)
\end{aligned}$$

Authorized licensed use limited to: TU Delft Library. Downloaded on August 01,2025 at 13:58:20 UTC from IEEE Xplore. Restrictions apply.

$$
\begin{aligned}
&\quad t(\bar{x} \mid x, \pi_{\mathbf{y}}(x+v)) p_{\mathbf{v}}(v) d\bar{x} dv \\
&= \iint \mathrm{P}(\omega_{\mathbf{x}}^{k-1} \models \psi \mid \omega_{\mathbf{x}}(0) = \bar{x}, \pi_{\mathbf{y}}) \\
&\quad t(\bar{x} \mid x, \pi_{\mathbf{y}}(x+v)) p_{\mathbf{v}}(v) d\bar{x} dv \\
&= V_k^{\pi_{\mathbf{y}}}(x)
\end{aligned}
$$

where in the third equality we marginalized over the event $\omega_{\mathbf{x}}^k(1) = \bar{x}$ and use the definition of conditional probability, and in the fourth equality we use that for $i < k$ and $x \in \bar{X}$

$$
\begin{aligned}
&\mathrm{P}(\omega_{\mathbf{x}}^{k-1}(i) \in X^1 \wedge \omega_{\mathbf{x}}^{k-1}([1, i-1]) \in \bar{X} \mid \omega_{\mathbf{x}}^{k-1}(0) = \bar{x}, \pi_{\mathbf{y}}) \\
&\qquad = \mathrm{P}(\omega_{\mathbf{x}}^{k}(i+1) \in X^1 \wedge \omega_{\mathbf{x}}^{k}([2, i+1-1]) \in \bar{X} \mid \\
&\qquad\quad \omega_{\mathbf{x}}^{k}(1) = \bar{x}, \pi_{\mathbf{y}})
\end{aligned}
$$

which holds because of the Markov property. □

References

[1] E. M. Clarke, O. Grumberg, and D. Peleg, *Model Checking*. Cambridge, MA, USA: MIT Press, 1999.

[2] C. Baier and J.-P. Katoen, *Principles of Model Checking*. Cambridge, MA, USA: MIT Press, 2008.

[3] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Mach. Learn.*, vol. 110, no. 3, pp. 457–506, 2021.

[4] P. Tabuada, *Verification and Control of Hybrid Systems: A Symbolic Approach*. Berlin, Germany: Springer Science & Business Media, 2009.

[5] C. Belta, B. Yordanov, and E. A. Gol, *Formal Methods for Discrete-Time Dynamical Systems*, vol. 89. Berlin, Germany: Springer, 2017.

[6] L. Doyen, G. Frehse, G. J. Pappas, and A. Platzer, "Verification of hybrid systems," in *Handbook of Model Checking*, Berlin, Germany: Springer, 2018, pp. 1047–1110.

[7] H. J. Kushner and P. Dupuis, *Numerical Methods for Stochastic Control Problems in Continuous Time*. New York, NY, USA: Springer-Verlag, 2001.

[8] S. E. Z. Soudjani, C. Gevaerts, and A. Abate, "FAUST2: Formal abstractions of uncountable-State stochastic processes," in *Proc. Int. Conf. Tools Algorithms Construction Anal. Syst.*, 2015, pp. 272–286.

[9] M. Lahijanian, S. B. Andersson, and C. Belta, "Formal verification and synthesis for discrete-time stochastic systems," *IEEE Trans. Autom. Control*, vol. 60, no. 8, pp. 2031–2045, Aug. 2015.

[10] L. Laurenti, M. Lahijanian, A. Abate, L. Cardelli, and M. Kwiatkowska, "Formal and efficient synthesis for continuous-time linear stochastic hybrid processes," *IEEE Trans. Autom. Control*, vol. 66, no. 1, pp. 17–32, Jan. 2021.

[11] M. Lahijanian, S. B. Andersson, and C. Belta, "Approximate Markovian abstractions for linear stochastic systems," in *Proc. 51st IEEE Conf. Decis. Control*, Dec. 2012, pp. 5966–5971.

[12] N. Cauchi, L. Laurenti, M. Lahijanian, A. Abate, M. Kwiatkowska, and L. Cardelli, "Efficiency through uncertainty: Scalable formal synthesis for stochastic hybrid systems," in *Proc. 22nd ACM Int. Conf. Hybrid Syst.: Comput. Control*, 2019, pp. 240–251.

[13] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*, Berlin, Germany: Springer, 2003, pp. 63–71.

[14] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for Gaussian process optimization in the bandit setting," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3250–3265, May 2012.

[15] S. R. Chowdhury and A. Gopalan, "On kernelized multi-armed bandits," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 844–853.

[16] H. Khosrowjerdi and K. Meinke, "Learning-based testing for autonomous systems using spatial and temporal requirements," in *Proc. 1st Int. Workshop Mach. Learn. Softw. Eng. Symbiosis*, 2018, pp. 6–15.

[17] W. Xiang et al., "Verification for machine learning, autonomy, and neural networks survey," 2018, *arXiv:1810.01989*.

[18] A. Girard and G. J. Pappas, "Approximation metrics for discrete and continuous systems," *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 782–798, May 2007.

[19] R. Luna, M. Lahijanian, M. Moll, and L. E. Kavraki, "Asymptotically optimal stochastic motion planning with temporal goals," in *Proc. 11th Int. Workshop Algorithmic Found. Robot.*, Istanbul, Turkey, Aug. 2014, pp. 335–352.

[20] R. Luna, M. Lahijanian, M. Moll, and L. E. Kavraki, "Fast stochastic motion planning with optimality guarantees using local policy reconfiguration," in *Proc. IEEE Conf. Robot. Automat.*, Hong Kong, May 2014, pp. 3013–3019.

[21] R. Luna, M. Lahijanian, M. Moll, and L. E. Kavraki, "Optimal and efficient stochastic motion planning in partially-known environments," in *Proc. 28th AAAI Conf. Artif. Intell.*, Quebec City, Canada, Jul. 2014, pp. 2549–2555. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8457

[22] E. M. Hahn, V. Hashemi, H. Hermanns, M. Lahijanian, and A. Turrini, "Multi-objective robust strategy synthesis for interval Markov decision processes," in *Proc. Int. Conf. Quantitative Eval. SysTems*, Berlin, Germany, Sep. 2017, pp. 207–223.

[23] E. M. Hahn, V. Hashemi, H. Hermanns, M. Lahijanian, and A. Turrini, "Interval Markov decision processes with multiple objectives: From robust strategies to Pareto curves," *ACM Trans. Model. Comput. Simul.*, vol. 29, no. 4, pp. 1–31, 2019.

[24] S. Haesaert, P. M. Van den Hof, and A. Abate, "Data-driven and model-based verification via Bayesian identification and reachability analysis," *Automatica*, vol. 79, pp. 115–126, 2017.

[25] J. Kenanian, A. Balkan, R. M. Jungers, and P. Tabuada, "Data driven stability analysis of black-box switched linear systems," *Automatica*, vol. 109, 2019, Art. no. 108533.

[26] A. Salamati, S. Soudjani, and M. Zamani, "Data-driven verification of stochastic linear systems with signal temporal logic constraints," *Automatica*, vol. 131, 2021, Art. no. 109781.

[27] T. S. Badings, A. Abate, N. Jansen, D. Parker, H. A. Poonawala, and M. Stoelinga, "Sampling-based robust control of autonomous systems with non-Gaussian noise," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 9669–9678.

[28] S. Dean, H. Mania, N. Matni, B. Recht, and S. Tu, "On the sample complexity of the linear quadratic regulator," *Found. Comput. Math.*, vol. 20, no. 4, pp. 633–679, 2020.

[29] H. Yoon, Y. Chou, X. Chen, E. Frew, and S. Sankaranarayanan, "Predictive runtime monitoring for linear stochastic systems and applications to geofence enforcement for UAVs," in *Proc. Int. Conf. Runtime Verification*, 2019, pp. 349–367.

[30] C. Fan, B. Qi, S. Mitra, and M. Viswanathan, "DryVR: Data-driven verification and compositional reasoning for automotive systems," in *Proc. Int. Conf. Comput. Aided Verification*, 2017, pp. 441–461.

[31] Z. Wang and R. M. Jungers, "Scenario-based set invariance verification for black-box nonlinear systems," *IEEE Contr. Syst. Lett.*, vol. 5, no. 1, pp. 193–198, Jan. 2021.

[32] S. Herbert, J. J. Choi, S. Sanjeev, M. Gibson, K. Sreenath, and C. J. Tomlin, "Scalable learning of safety guarantees for autonomous systems using Hamilton–Jacobi reachability," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2021, pp. 5914–5920.

[33] N. Musavi, D. Sun, S. Mitra, G. Dullerud, and S. Shakkottai, "HooVer: A framework for verification and parameter synthesis in stochastic systems using optimistic optimization," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, 2021, pp. 923–930.

[34] D. Sun and S. Mitra, "NeuReach: Learning reachability functions from simulations," in *Proc. 28th Int. Conf. Tools Algorithms Construction Anal. Syst.*, Munich, Germany, 2022, pp. 322–337.

[35] A. Nejati, A. Lavaei, P. Jagtap, S. Soudjani, and M. Zamani, "Formal verification of unknown discrete- and continuous-time systems: A data-driven approach," *IEEE Trans. Autom. Control*, vol. 68, no. 5, pp. 3011–3024, May 2023.

[36] A. Alanwar, A. Koch, F. Allgöwer, and K. H. Johansson, "Data-driven reachability analysis from noisy data," *IEEE Trans. Autom. Control*, vol. 68, no. 5, pp. 3054–3069, May 2023.

[37] Y. Wang et al., "Joint differentiable optimization and verification for certified reinforcement learning," in *Proc. ACM/IEEE 14th Int. Conf. Cyber-Phys. Syst.*, 2023, pp. 132–141.

[38] J. Jackson, L. Laurenti, E. Frew, and M. Lahijanian, "Safety verification of unknown dynamical systems via Gaussian process regression," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, 2020, pp. 860–866.

[39] Z. Jin, M. Khajenejad, and S. Z. Yong, "Data-driven abstraction and model invalidation for unknown systems with bounded Jacobians," *IEEE Control Syst. Lett.*, vol. 6, pp. 3421–3426, Jun. 2022.

[40] J. Jiang, Y. Zhao, and S. Coogan, "Safe learning for uncertainty-aware planning via interval MDP abstraction," *IEEE Control Syst. Lett.*, vol. 6, pp. 2641–2646, May 2022.

Authorized licensed use limited to: TU Delft Library. Downloaded on August 01,2025 at 13:58:20 UTC from IEEE Xplore. Restrictions apply.

[41] K. Hashimoto, A. Saoud, M. Kishida, T. Ushio, and D. V. Dimarogonas, "Learning-based symbolic abstractions for nonlinear control systems," *Automatica*, vol. 146, 2022, Art. no. 110646.

[42] A. Peruffo and M. Mazo, "Data-driven abstractions with probabilistic guarantees for linear PETC systems," *IEEE Control Syst. Lett.*, vol. 7, pp. 115–120, June 2022.

[43] B. Xue, M. Zhang, A. Easwaran, and Q. Li, "PAC model checking of black-box continuous-time dynamical systems," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 39, no. 11, pp. 3944–3955, Nov. 2020.

[44] A. K. Akametalu, S. Kaynama, J. F. Fisac, M. N. Zeilinger, J. H. Gillula, and C. J. Tomlin, "Reachability-based safe learning with Gaussian processes," in *Proc. IEEE 53rd Annu. Conf. Decis. Control (CDC)*, Los Angeles, CA, USA, 2014, pp. 1424–1431.

[45] Y. Sui, A. Gotovos, J. W. Burdick, and A. Krause, "Safe exploration for optimization with Gaussian processes," in *Proc. Mach. Learn. Res.*, 2015, pp. 997–1005.

[46] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause, "Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, 2016, pp. 4661–4666.

[47] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 908–918.

[48] M. K. Helwa, A. Heins, and A. P. Schoellig, "Provably robust learning-based approach for high-accuracy tracking control of Lagrangian systems," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1587–1594, Apr. 2019.

[49] P. Jagtap, G. J. Pappas, and M. Zamani, "Control barrier functions for unknown nonlinear systems using Gaussian processes," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, 2020, pp. 3699–3704.

[50] R. Wajid, A. U. Awan, and M. Zamani, "Formal synthesis of safety controllers for unknown stochastic control systems using Gaussian process learning," in *Proc. 4th Annu. Learn. Dyn. Control Conf.*, 2022, pp. 624–636.

[51] P. Massart, *Concentration Inequalities and Model Selection*, vol. 6. Berlin, Germany: Springer, 2007.

[52] D. P. Bertsekas and S. Shreve, *Stochastic Optimal Control: The Discrete-Time Case*. Nashua, NH, USA: Athena Scientific, 2004.

[53] A. Abate, F. Redig, and I. Tkachev, "On the effect of perturbation of conditional probabilities in total variation," *Statist. Probability Lett.*, vol. 88, pp. 1–8, 2014.

[54] I. Steinwart, "On the influence of the kernel on the consistency of support vector machines," *J. Mach. Learn. Res.*, vol. 2, pp. 67–93, 2001.

[55] C. A. Micchelli, Y. Xu, and H. Zhang, "Universal kernels," *J. Mach. Learn. Res.*, vol. 7, pp. 2651–2667, 2006.

[56] A. Lederer, J. Umlauft, and S. Hirche, "Uniform error bounds for Gaussian process regression with application to safe control," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 657–667.

[57] R. Givan, S. Leach, and T. Dean, "Bounded-parameter Markov decision processes," *Artif. Intell.*, vol. 122, no. 1/2, pp. 71–109, 2000.

[58] A. Blaas, A. Patane, L. Laurenti, L. Cardelli, M. Kwiatkowska, and S. Roberts, "Adversarial robustness guarantees for classification with Gaussian processes," in *Proc. 23rd Int. Conf. Artif. Intell. Statist.*, 2020, pp. 3372–3382.

[59] C. Knuth, G. Chou, N. Ozay, and D. Berenson, "Planning with learned dynamics: Probabilistic guarantees on safety and reachability via Lipschitz constants," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5129–5136, Jul. 2021.

[60] A. Chakrabarty, D. K. Jha, G. T. Buzzard, Y. Wang, and K. G. Vamvoudakis, "Safe approximate dynamic programming via kernelized Lipschitz estimation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 405–419, Jan. 2021.

[61] S. Vakili, K. Khezeli, and V. Picheny, "On information gain and regret bounds in Gaussian process bandits," in *Proc. 24th Int. Conf. Artif. Intell. Statist.*, 2021, pp. 82–90.

[62] G. Delimpaltadakis, L. Laurenti, and M. Mazo Jr, "Abstracting the sampling behaviour of stochastic linear periodic event-triggered control systems," in *Proc. 60th IEEE Conf. Decis. Control*, 2021, pp. 1287–1294.

[63] C. Camarero, "Simple, fast and practicable algorithms for Cholesky, LU and QR decomposition using fast rectangular matrix multiplication," 2018, *arXiv:1812.02056*.

[64] P. Jagtap, S. Soudjani, and M. Zamani, "Formal synthesis of stochastic systems via control barrier certificates," *IEEE Trans. Autom. Control*, vol. 66, no. 7, pp. 3097–3110, Jul. 2021.

[65] J. Jackson, L. Laurenti, E. Frew, and M. Lahijanian, "Synergistic offline-online control synthesis via local Gaussian process regression," in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, 2021, pp. 2232–2239.

[66] V. I. Paulsen and M. Raghupathi, *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*, vol. 152. Cambridge, U.K.: Cambridge Univ. Press, 2016.

John Skovbekk (Student Member, IEEE) received the bachelor's degree in aerospace engineering and mechanics, and the master's degree in science from the University of Minnesota, Twin Cities, Minneapolis, MN, USA, in 2015 and 2018, respectively, and the Ph.D. degree in aerospace engineering sciences from the Ann and H.J. Smead Department of Aerospace Engineering Sciences from the University of Colorado, Boulder, CO, USA.

His research focuses on the intersection of formal control methods with data-driven modeling to enhance the safety and efficacy of robotics.

Luca Laurenti (Member, IEEE) received the Ph.D. degree in computer science from the Department of Computer Science, University of Oxford, Oxford, U.K.

He is a tenure-track Assistant Professor with the Delft Center for Systems and Control, TU Delft, Delft, The Netherlands, and Co-Director of the HERALD Delft AI Lab, Delft. He was a Member of the Trinity College, Oxford. He has a background in stochastic systems, control theory, formal methods, and artificial intelligence. His research focuses on developing data-driven systems provably robust to interactions with a dynamic and uncertain world.

Eric Frew received the B.S degree. in mechanical engineering from Cornell University, NY, USA, in 1995, and the M.S and Ph.D. degrees in aeronautics and astronautics from Stanford University, Stanford, CA, USA, in 1996 and 2003, respectively.

He is currently the Center Director with the National Science Foundation Industry University Cooperative Research Center (IUCRC) for Autonomous Air Mobility and Sensing (CAAMS), and a Professor with the Ann and H.J. Smead Aerospace Engineering Sciences Department,e University of Colorado, Boulder, CO, USA. He has been designing and deploying autonomous robotic systems for more than twenty five years His research interests include autonomous flight of heterogeneous uncrewed aircraft systems; distributed information-gathering by mobile robots; miniature self-deploying systems; and guidance and control of uncrewed aircraft in complex atmospheric phenomena.

Dr. Frew has successfully deployed autonomous robots in a wide variety of scenarios.

Morteza Lahijanian (Member, IEEE) received the B.S. degree in bioengineering from the University of California, Berkeley, Berkeley, CA, USA, in 2005, and the Ph.D. degree in mechanical engineering from the Boston University, Boston, MA, USA, in 2013.

He is an Assistant Professor with the Aerospace Engineering Sciences Department, an Affiliated Faculty with the Computer Science Department and Robotics program, and the Director of the Assured, Reliable, and Interactive Autonomous (ARIA) Systems group, University of Colorado Boulder, Boulder, CO, USA. He was a Research Scientist with the Department of Computer Science, University of Oxford, Oxford, U.K. He was a Postdoctoral Scholar in computer science with Rice University, Houston, TX, USA. His research interests include control theory, stochastic hybrid systems, formal methods, machine learning, and game theory with applications in robotics, particularly, motion planning, strategy synthesis, formal verification, and human–robot interaction.

Dr. Lahijanian was the recipient of the RIO Faculty Fellowship, Outstanding Junior Faculty Award, Ella Mae Lawrence R. Quarles Physical Science Achievement Award, Jack White Engineering Physics Award, and NSF GK-12 Fellowship.

Authorized licensed use limited to: TU Delft Library. Downloaded on August 01,2025 at 13:58:20 UTC from IEEE Xplore. Restrictions apply.