# Data-Driven Abstractions via Binary-Tree Gaussian Processes for Formal Verification

**Oliver Schön** * **Shammakh Naseer** * **Ben Wooding** *
**Sadegh Soudjani** **

* *School of Computing, Newcastle University, UK*
** *Max Planck Institute for Software Systems, Germany*

**Abstract:** To advance formal verification of stochastic systems against temporal logic requirements for handling unknown dynamics, researchers have been designing data-driven approaches inspired by breakthroughs in the underlying machine learning techniques. As one promising research direction, abstraction-based solutions based on Gaussian process (GP) regression have become popular for their ability to learn a representation of the latent system from data with a quantified error. Results obtained based on this model are then translated to the true system via various methods. In a recent publication, GPs using a so-called binary-tree kernel have demonstrated a polynomial speedup w.r.t. the size of the data compared to their vanilla version, outcompeting all existing sparse GP approximations. Incidentally, the resulting binary-tree Gaussian process (BTGP) is characteristic for its piecewise-constant posterior mean and covariance functions, naturally abstracting the input space into discrete partitions. In this paper, we leverage this natural abstraction of the BTGP for formal verification, eliminating the need for cumbersome abstraction and error quantification procedures. We show that the BTGP allows us to construct an interval Markov chain model of the unknown system with a speedup that is polynomial w.r.t. the size of the abstraction compared to alternative approaches. We provide a delocalized error quantification via a unified formula even when the true dynamics do not live in the function space of the BTGP. This allows us to compute upper and lower bounds on the probability of satisfying reachability specifications that are robust to both aleatoric and epistemic uncertainties.

*Keywords:* Stochastic Systems, Formal Verification, Gaussian Processes, Uncertain Systems, Error Quantification, Machine Learning, System Identification

## 1. INTRODUCTION

Developing the control software for safety-critical cyber-physical systems such as power grids, autonomous vehicles, medical implants, and robotic systems has raised the need for formal approaches to guarantee the systems will behave as expected, and not lead to catastrophe. Designing controllers for such systems is especially challenging for their stochasticity and heterogeneous nature, i.e., the components of the system evolve over both discrete and continuous spaces. Previous approaches were mainly based on physics-based models of the underlying systems. To meet the increasing need for more scalable and uncertainty-robust techniques, data-driven approaches govern the contemporary landscape of active research. One powerful offspring that is being explored is *Gaussian process* (GP) regression (Rasmussen and Williams, 2018). GPs use a Bayesian paradigm to learning unknown functions from observations, that is particularly valuable for safety-critical applications as the statistical approximation error is naturally quantified.

In this work, we consider GPs based on the so-called *binary-tree* (BT) *kernel* (Cohen et al., 2022b), a discontinuous kernel that has been shown to reduce the computational burden of GP regression significantly, increasing its scalability to higher dimensional systems and larger data sets. In comparison to alternative approaches to constructing scalable GPs (see, e.g., Quinonero-Candela and Rasmussen (2005)), the BT kernel achieves GP regression in linear time – a polynomial improvement. The resulting *binary-tree Gaussian process* (BTGP) is characteristic for its piecewise-constant posterior mean and covariance functions, resulting in a natural partitioning of its input space. The main idea of this paper is to exploit this direct learning of an abstract representation of the unknown data-generating system for formal verification.

Formal verification of known stochastic systems against temporal logic properties has been well studied (Baier and Katoen, 2008). For settings where the system dynamics are uncertain – termed *epistemic uncertainty* — and specifications have infinite horizons, most available abstraction-based approaches can be divided into two categories. Approaches based on statistical relations replace the original system with an estimated surrogate model that does not capture the epistemic uncertainty, generally a variant of a *Markov chain* (MC). Results obtained on this model hence rely on quantifying the probabilistic deviation to the true system to transfer guarantees (Abate et al., 2008; Haesaert et al., 2018). For example, *probabilistic coupling relations* can be used to transfer guarantees obtained on a parameterized surrogate model of the unknown system

back to the original latent system by quantifying the expected *parametric uncertainty* (Schön et al., 2023). Verification is conducted using a robust version of dynamic programming (Haesaert and Soudjani, 2020). Establishing such coupling relations, however, requires the computation of local error parameters quantifying the error between the true system and its surrogate model for every discrete partition, deeming their application to high-dimensional systems computationally excruciating. In an effort to avoid the construction of surrogate models requiring such types of relations, approaches based on *interval Markov chains* (IMCs) incorporate both stochasticity and epistemic uncertainty (Badings et al., 2023; Lavaei et al., 2022; Jackson et al., 2020). The IMCs constructed based on upper and lower bounds on the transition probabilities, learned, e.g., using GP regression (Jackson et al., 2020), can be subsequently verified via the *interval iteration algorithm* (Haddad and Monmege, 2018). Whilst this alleviates the need for constructing additional coupling relations, the computations required to construct and verify the IMC are substantially more involving than the corresponding computations on the MC in the coupling-based approach.

In this paper, we make an effort to overcome this compromise. We use BTGP regression to learn the unknown dynamics of a discrete-time continuous-space stochastic system and show that the probability bounds of the resulting IMC are considerably easier to compute by obtaining transition probability distributions from the BTGP that are naturally piecewise-constant. Whilst prior IMC-based approaches require the optimization of both probability bounds to identify the worst-case probabilities for every transition, this change allows us to perform only two integrations for each transition – one for either bound – and obtain equivalent robustness results. Hence, we get a speedup for the construction of the abstraction that is polynomial w.r.t. the size of the abstraction. With this, we make the following contributions.

- We provide a data-driven approach to verify uncertain stochastic systems defined over continuous spaces against infinite-horizon specifications based on an IMC generated via BTGP regression. In comparison to existing approaches, the IMC is constructed without the need for the identification of the worst-case bounds for every partition. This yields a computational speedup over IMC abstraction that is polynomial in the number of partitions.
- We obtain statistical guarantees on the accuracy of the BTGP model that combine available measure-theoretic results with an error inflicted by the approximation error of the BTGP even if the true dynamics function does not live in the BT kernels RKHS.
- We show how a discontinuous kernel can be used to generate a finite abstraction of the latent model and how this abstraction can be efficiently verified.
- We give closed-form expressions for the function space generated by the BT kernel.

**Related work:** We give a brief review of relevant related approaches on data-driven formal verification and synthesis of uncertain stochastic systems.

The works by Badings et al. (2023) and Lavaei et al. (2022) address epistemic uncertainty in stochastic systems by abstracting the system to an interval *Markov decision process* (MDP). While prior is limited to finite horizons and linear systems, latter requires solving a scenario optimization problem. Jackson et al. (2020) and Jiang et al. (2022) have used GPs instead of scenario optimization. In a related kernel-based setting, Thorpe et al. (2022) use conditional mean embeddings (CMEs) to embed the conditional distribution of a non-Markovian random trajectory into an RKHS. Temporal logic control of systems captured by stochastic neural network dynamic models (NNDM) is addressed by Adams et al. (2022) through translating the NNDMs to IMDPs. The starting point, however, is the NNDM and no guarantees on the correctness of the learned model w.r.t. the data-generating system are considered, rendering the approach only robust from a model-based perspective. Capturing epistemic uncertainty using interval MDPs relies on the assumption of state-wise independent epistemic uncertainty, leading to inconsistent and overly conservative results. In contrast, Schön et al. (2023) capture epistemic uncertainty explicitly using parametric MDPs and probabilistic coupling relations to the latent true system. Furthermore, their results are applicable to a wider range of specifications and distributions. The neural abstractions studied by Abate et al. (2022) rely on a counterexample-producing scheme via SMT solvers. A contraction-guided adaptive partitioning approach is introduced by Harapanahalli et al. (2023a).

Latest work in the area of abstraction-free approaches are based on constructing a set of feasible models from data and finding a common control barrier certificate (CBC) (Cohen et al., 2022a; Lopez and Slotine, 2023). Salamati et al. (2024) find CBCs using robust scenario optimization. Whilst this necessitates them to sample transitions between pairs of partitions multiple times, they also require knowledge of the Lipschitz constants of several intermediate functions. The work by Wajid et al. (2022) addresses the problem of finding CBCs using GPs. Whilst CBCs are often deemed more scalable than abstraction-based solutions, finding a valid CBC for systems with non-control-affine and non-polynomial dynamics is generally hard. Even more, CBCs for specifications beyond simple reachability yields, e.g., sequential reachability problems (Jagtap et al., 2020), greatly impeding their applicability. Further work on model-free reinforcement learning studies synthesizing robust temporal logic controllers without constructing an explicit model of the system (Hasanbeig et al., 2019; Kazemi and Soudjani, 2020; Lavaei et al., 2020). An approach based on computing reachable sets leverages random set theory to obtain infinite-sample guarantees is provided by Lew and Pavone (2021) without analyzing finite-sample convergence rates. Similarly, Cao et al. (2022) construct reachable sets efficiently using GPs under the assumption of known bounds on the unknown dynamics components. There exist a body of work on verifying models composed of neural networks, e.g., (Harapanahalli et al., 2023b; Jafarpour et al., 2023). However, similar as mentioned before, no guarantees of the correctness of the data-driven model w.r.t. the data-generating system are provided.

The remainder of the manuscript is structured as follows. After providing preliminary definitions and the problem statement in Sec. 2, we dive into system identification

using BTGP regression in Sec. 3. Here, we provide our results disseminating the BTGP and its associated statistical error bound. In Sec. 4, we demonstrate how the BTGP allows us to generate an IMC model efficiently and provide a suitable verification algorithm. In Sec. 5, we showcase the presented approach on a nonlinear case study.

## 2. PRELIMINARIES AND PROBLEM STATEMENT

Let $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, $\mathbb{N}$, and $\mathbb{B}$ be the sets of reals, non-negative reals, positive integers, and binary numbers. Let $s \in \mathbb{B}^q$ be a binary string of length $|s| := q$. We denote its substring prefix of length $l \in \mathbb{N}$ as $[s]_{\leq l}$. Let $I_N$ denote the $N \times N$ identity matrix. The transpose of a vector or matrix $A$ is indicated as $A^\top$. For some topological space $\mathbb{X}$, let $X_N := [x_1, \ldots, x_N]^\top$ be a column vector with $x_i \in \mathbb{X}$. To save space, we may use a compact notation and write he column vector as $X_N := [x_i]_{i=1}^N$. We denote the element-wise evaluation of a function $f : \mathbb{X} \to \mathbb{Y}$ on $X_N$ as $f(X_N) := [f(x_i)]_{i=1}^N$. Similarly, we may write $A = [a_{ij}]_{i,j=1}^N$ to denote a matrix with its elements. Trace of such a matrix is denoted by $\mathrm{tr}(A) := \sum_i a_{ii}$.

**Probability theory:** A *probability space* is a tuple $(\mathbb{X}, \mathcal{B}(\mathbb{X}), p)$ equipped with a sample space $\mathbb{X}$, a Borel $\sigma$-algebra $\mathcal{B}(\mathbb{X})$ defined over $\mathbb{X}$, i.e., the smallest $\sigma$-algebra containing open subsets of $\mathbb{X}$, and a probability measure $p$, which has realizations $x \sim p(\,\cdot\,)$. In this work, we restrict our attention to Polish sample spaces (Bogachev, 2007). A *probability measure* $p$ on a measurable space $(\mathbb{X}, \mathcal{B}(\mathbb{X}))$ is a map $p : \mathcal{B}(\mathbb{X}) \to [0, 1]$ such that for all countable collections $\{A_i\}_{i=1}^\infty$ of pairwise disjoint sets in $\mathcal{B}(\mathbb{X})$ it holds that $p(\bigcup_i A_i) = \sum_i p(A_i)$ and $p(\mathbb{X}) = 1$. For two measurable spaces $(\mathbb{X}, \mathcal{B}(\mathbb{X}))$ and $(\mathbb{Y}, \mathcal{B}(\mathbb{Y}))$, a *probability kernel* is a mapping $\mathbf{p} : \mathbb{X} \times \mathcal{B}(\mathbb{Y}) \to [0, 1]$ such that $\mathbf{p}(x, \,\cdot\,) : \mathcal{B}(\mathbb{Y}) \to [0, 1]$ is a probability measure for all $x \in \mathbb{X}$, and $\mathbf{p}(\,\cdot\,, Y) : \mathbb{X} \to [0, 1]$ is measurable for all $Y \in \mathcal{B}(\mathbb{Y})$. A probability kernel associates to each point $x \in \mathbb{X}$ a measure, also denoted by $\mathbf{p}(\,\cdot\,|x)$. Given $n \in \mathbb{N}$, the (Gaussian) normal probability measure with mean $\mu \in \mathbb{R}^n$ and covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$ is denoted as $\mathcal{N}(dx|\mu, \Sigma)$.

**RKHS theory:** A positive-definite, symmetric function $k : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ is called a *kernel* (note the distinction from *probability kernels*) if for all $N \in \mathbb{N}$ we have $\sum_{i=1}^N \sum_{j=1}^N a_i a_j k(x_i, x_j) \geq 0$ for all $x_1, \ldots, x_N \in \mathbb{X}$ and $a_1, \ldots, a_N \in \mathbb{R}$. Note that a positive-definite kernel gives rise to a positive-definite *Gram matrix* $K := [k(x_i, x_j)]_{i,j=1}^N$. Given a kernel $k$ on a non-empty set $\mathbb{X}$, there exists a corresponding unique *reproducing kernel Hilbert space* (RKHS) $\mathcal{H}_k$ of functions. $\mathcal{H}_k$ is equipped with an inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_k}$ with the *reproducing property* such that for any function $f : \mathbb{X} \to \mathbb{R}$, $f \in \mathcal{H}_k$ and $x \in \mathbb{X}$ we have $f(x) = \langle f, k(\,\cdot\,, x) \rangle_{\mathcal{H}_k}$ where $k(\,\cdot\,, x) : \mathbb{X} \to \mathcal{H}_k$ is a real-valued function for which $k(x', x) = \langle k(\,\cdot\,, x'), k(\,\cdot\,, x) \rangle_{\mathcal{H}_k}$ for all $x, x' \in \mathbb{X}$. The inner product induces the norm $||f||_{\mathcal{H}_k} := \sqrt{\langle f, f \rangle_{\mathcal{H}_k}}$. A kernel $k$ is called *translation-invariant* if it can be expressed as a function of distance between points $x, x' \in \mathbb{X}$, i.e., there exists some function $\Delta : \mathbb{X} \to \mathbb{R}_{\geq 0}$ such that $k(x, x') = \Delta(x - x')$.

**Dynamical systems:** Consider a system $\mathbf{M}$ with a state space $\mathbb{X} \subset \mathbb{R}^n$ of dimension $n \in \mathbb{N}$ and dynamics

$$\mathbf{M} : \; x_{t+1} = f(x_t) + v_t, \quad v_t \sim \mathcal{N}(\,\cdot\,|0, \sigma_v^2 I_n), \qquad (1)$$

where $x_t \in \mathbb{X}$ denotes the system state at the time instance $t \in \mathbb{N} \cup \{0\}$. The *independent, identically distributed* (i.i.d.) process noise $v_t$ is zero-mean Gaussian with covariance matrix $\sigma_v^2 I_n$. The transition function $f : \mathbb{X} \to \mathbb{X}$ is unknown and we assume that it lives in the RKHS of some translation-invariant kernel $k$, i.e., $f \in \mathcal{H}_k$. The covariance matrix of the process noise is known.

**Reachability specifications:** Given a target set $\mathcal{T} \subset \mathbb{X}$, the system satisfies the reachability specification $\psi_\mathcal{T}$ with probability at least $p_\psi \in [0, 1]$, if the probability that the trajectories stay in $\mathbb{X}$ until eventually reaching the target set $\mathcal{T}$ is greater than or equal to $\psi_\mathcal{T}$. This is denoted by $\mathbb{P}(\mathbf{M} \models \psi_\mathcal{T}) \geq p_\psi$. For the purpose of this work, we limit ourselves to reachability specifications, but the provided verification algorithm can be applied to any linear temporal logic specification by adding a preprocessing step on the graph structure of the constructed abstract model (Baier and Katoen, 2008; Haddad and Monmege, 2018).

**Problem statement:** Let $\mathcal{D}_N := \{x_i, y_i\}_{i=1}^N \subset \mathbb{X} \times \mathbb{X}$ be a set of $N \in \mathbb{N}$ data samples from the latent true system in (1) such that $y_i := f(x_i) + v_i$ are output samples with i.i.d. noise $v_i \sim \mathcal{N}(\,\cdot\,|0, \sigma_v^2 I_n)$. We may write $\mathcal{D}_N = (X_N, Y_N)$, where $X_N := [x_1, \ldots, x_N]$ and $Y_N := [y_1, \ldots, y_N]^\top$.

*Problem 1.* Let data $\mathcal{D}_N$ from the system in (1) with unknown function $f \in \mathcal{H}_k$ be given. For an infinite-horizon reachability specification $\psi_\mathcal{T}$, verify that the system satisfies $\psi_\mathcal{T}$ with probability at least $p_\psi \in [0, 1]$.

To address this problem, we construct a function estimator $\hat{f} : \mathbb{X} \to \mathbb{X}$ based on $\mathcal{D}_N$ that approximates the unknown function $f \in \mathcal{H}_k$ in (1). In particular, we specify a kernel $\hat{k}$ characterizing the function class of $\hat{f} \in \mathcal{H}_{\hat{k}}$. To be able to address a wealth of functions $f$ and perform the function approximation in a function space $\mathcal{H}_{\hat{k}} \not\supset \mathcal{H}_k$, we raise the following assumption.

*Assumption 1.* Let the unknown function $f \in \mathcal{H}_k$ and the function estimator $\hat{f} \in \mathcal{H}_{\hat{k}}$, with known kernels $k, \hat{k} : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ and complexity bound $B \geq ||f||_{\mathcal{H}_k} \geq 0$ .

Moving forward, we use '$\hat{}$' to denote symbols referring to the estimated model.

*Remark 1.* We have raised Assumption 1 to allow for a greater flexibility in choosing the kernel $\hat{k}$ used for estimation. In particular, we will use a discontinuous kernel. To still be able to address more arbitrary functions, such as continuous functions living in the RKHS of, e.g., the squared exponential kernel, we need to quantify the error introduced by constructing an estimator in a different RKHS.

## 3. SYSTEM IDENTIFICATION USING BINARY-TREE GAUSSIAN PROCESSES

### 3.1 Gaussian Process Regression

A *Gaussian process* (GP) is a parameter-free regression model that allows the user to learn input-output mappings of the form $\hat{f}_d : \mathbb{X} \to \mathbb{R}$, $d \in \{1, \ldots, n\}$, from empirical data $\mathcal{D}_N = \{x_i, y_i\}_{i=1}^N$, $y_i := [y_{i,1}, \ldots, y_{i,n}]^\top$, such that the prediction errors $\sum_{i=1}^N ||y_{i,d} - \hat{f}_d(x_i)||$ are minimized for all dimensions $d$ (Rasmussen and Williams, 2018). Recall

that, as specified in (1), we assume that the process noise on the samples $y_i$ is dimension-wise independent, which allows us to use individual GPs for each output dimension. This is without loss of generality, and the results of this paper are applicable to non-diagonal covariance matrices by applying a linear transformation on the data gathered from the system. In the following, we may drop the subscript $d$ indicating the dimension for simplicity.

As an infinite-dimensional generalization of the multivariate normal distribution, GPs are particularly popular for their ability to capture statistical information about the latent mapping. The following definition is analogous to Definition 2.2 by Kanagawa et al. (2018).

*Definition 1.* (Gaussian process (GP)). Let $\hat{k} : \mathbb{X}\times\mathbb{X} \to \mathbb{R}$ be a (symmetric, positive-definite) *kernel* or *covariance function* and $\hat{m}_d : \mathbb{X} \to \mathbb{R}$. A random function $\hat{f}_d : \mathbb{X} \to \mathbb{R}$ is a *GP*, denoted by $\hat{f}_d \sim \mathcal{GP}(\hat{m}_d, \hat{k})$, if for any finite $X_N := [x_1, \dots, x_N]$, $x_i \in \mathbb{X}$, $N \in \mathbb{N}$, the vector $\hat{f}_{d,N} := [\hat{f}_d(x_i)]_{i=1}^N$ has the distribution

$$\hat{f}_{d,N} \sim \mathcal{N}(\cdot\,|\hat{m}_{d,N}, \hat{K}_N),$$

with $\hat{m}_{d,N} := [\hat{m}_d(x_i)]_{i=1}^N$ and Gram matrix $\hat{K}_N := [\hat{k}(x_i, x_j)]_{i,j=1}^N$.

The goal of GP regression is to generate predictions of the output at test points $x \in \mathbb{X}$ that are not captured by the data $\mathcal{D}_N$. Even though we do not assume that the unknown data-generating function $f$ is a sample from the GP with kernel $\hat{k}$ (see Assumption 1), we can construct a so-called *posterior GP* and obtain a probabilistic bound on its prediction error (e.g., via Theorem 2 by Chowdhury and Gopalan (2017) or the results by Fiedler et al. (2021)).

*Definition 2.* (Posterior GP). Let $\mathcal{D}_N = (X_N, Y_N)$ from the system (1). Then, the *posterior* distribution of $\hat{f}_d(x)$ given the data $\mathcal{D}_N$ is $\hat{f}_d(x) \sim \mathcal{N}(\cdot\,|\hat{\mu}_{d,N}(x), \hat{\sigma}^2_N(x))$ is characterized by the posterior mean and covariance

$$\hat{\mu}_{d,N}(x) := \hat{k}_N(x)^\top [\hat{K}_N + \sigma_v^2 I_N]^{-1} Y_{d,N}, \text{ and} \tag{2}$$

$$\hat{\sigma}^2_N(x) := \hat{k}(x,x) - \hat{k}_N(x)^\top [\hat{K}_N + \sigma_v^2 I_N]^{-1} \hat{k}_N(x), \tag{3}$$

with $\hat{k}_N(x) := [\hat{k}(x_i, x)]_{i=1}^N$ and $Y_{d,N}$ being the $d^{\text{th}}$ column of $Y_N$.

Based on Definition 2, predictions of the next state of the system (1) for the current state $x \in \mathbb{X}$ can be generated via $\hat{y}_d(x) \sim \mathcal{N}(\cdot\,|\hat{\mu}_{d,N}(x), \hat{\sigma}^2_N(x) + \sigma_v^2)$. We establish a probabilistic bound on the prediction error in Subsec. 3.3 after introducing a specific type of GP in the upcoming subsection.

### *3.2 Binary-Tree Gaussian Processes*

We focus on GPs based on the *binary-tree* (BT) kernel, which is defined on a finite space of binary strings and wields a range of practical advantages that renders it particularly interesting for formal approaches (Cohen et al., 2022b). Let us define a map $\Psi_q : \mathbb{X} \to \mathbb{B}^q$ that encodes every state $x \in \mathbb{X}$ from the continuous space $\mathbb{X}$ as a binary string $s \in \mathbb{B}^q$ in the finite space of binary strings of length $q$. Note that this implies a partitioning of the space $\mathbb{X}$ into $2^q$ disjoint sets, thus $q \in \mathbb{N}$ is called the *bit-depth* or *precision*.

In the following, we denote the set of states represented by a string $s \in \mathbb{B}^q$ as $\Xi_q(s) \subset \mathbb{X}$, i.e., we have $\Psi_q(x) = s$ for all $x \in \Xi_q(s)$. It follows that the state space is the union of the partitions, i.e., $\mathbb{X} = \bigcup_{s\in\mathbb{B}^q} \Xi_q(s)$ for any precision $q \in \mathbb{N}$. For a set $A \subset \mathbb{X}$, we may write $\Psi_q(A) \subset \mathbb{B}^q$ to denote its projection onto $\mathbb{B}^q$. To simplify the notation, we may write $\Psi_i(x) := [\Psi_q(x)]_{\leq i}$, $x \in \mathbb{X}$, $i \in \mathbb{N}$, i.e., the operator that takes the first $i$ elements of the binary string $\Psi_q(x)$. With this, we define the BT kernel formally.

*Definition 3.* (Binary-tree (BT) kernel). Given a map $\Psi_q : \mathbb{X} \to \mathbb{B}^q$, the *BT kernel* $\hat{k}_q : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ of *bit-depth* or *precision* $q \in \mathbb{N}$ is defined as

$$\hat{k}_q(x, x') := \textstyle\sum_{i=1}^q w_i \mathbb{1}\big(\Psi_i(x) = \Psi_i(x')\big), \tag{4}$$

with weight coefficients $w_i \in \mathbb{R}$ such that $\sum_{i=1}^q w_i = 1$.

Intuitively, the BT kernel $\hat{k}_q$ assigns its covariance mass based on whether points $x, x' \in \mathbb{X}$ fall into the same partitions. The final covariance value is a weighted sum of this coincidence on all discretisation levels up to precision $q$. It stands out that, in contrast to well-known alternatives such as the *squared-exponential* or *Matérn* kernel, the BT kernel is discontinuous since it is translation-*variant*. Hence, it produces discontinuous functional mappings when used in the context of GP regression. Fig. 1 depicts the piecewise-constant posterior mean $\hat{\mu}_N$ and double standard deviation $\hat{\mu}_N \pm 2\hat{\sigma}_N$ as functions of state of a GP utilizing a BT kernel of precision $q = 2$. Moving forward, we will call such a GP a *binary-tree Gaussian process* (BTGP), which yields inherently discrete abstractions/representations of the underlying system. Furthermore, we may decide to write them as explicit functions of $s \in \mathbb{B}^q$, i.e., $\hat{\mu}_N(s)$ and $\hat{\sigma}^2_N(s)$. We note that although function samples from the BTGP (or any GP in general) do not live in the RKHS of the associated kernel almost surely, the resulting posterior mean function does (Kanagawa et al., 2018, Sec. 4).

*Remark 2.* The BT kernel violates the conditions for a continuous RKHS (Berlinet and Thomas-Agnan, 2011, Theorem 17, condition a)). Despite being discontinuous, the BT kernel is still a valid positive-definite kernel as shown in Proposition 1 by Cohen et al. (2022b) and therefore by the Moore-Aronszajn theorem acts as a reproducing kernel for a unique RKHS (Berlinet and Thomas-Agnan, 2011, Theorem 3), which contains only discontinuous functions.

*Remark 3.* An implication of using the BT kernel of precision $q$ in GP regression is that the resulting Gram matrix $\hat{K}_N$ will be capped at a size of at most $2^q \times 2^q$. This is since $M \in \mathbb{N}$ data samples $\{x_i, y_i\}_{i=1}^N$ can be replaced by a modified dataset $\{x_s, y_s\}_{s\in\mathbb{B}^q}$, where $x_s \in \Xi_q(s)$ is a representative state and $y_s = \frac{1}{|A(s)|}\sum_{i\in A(s)} y_i$ with $A(s) := \{i \,|\, x_i \in \Xi_q(s)\}$ (averaging over $y_i$'s with $x_i$ being in the same partition set).

The following theorem establishes a closed-form representation of the functions living in the RKHS of the BT kernel.

*Theorem 1.* (BT function space). Let $\hat{k}_q : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ be the BT kernel in (4). Then, its associated RKHS contains functions $f \in \mathcal{H}_{\hat{k}_q}$ that can be represented in the form of

$$f(x) = \textstyle\sum_{i=1}^q \sqrt{w_i} \sum_{s\in\mathbb{B}^i} y_s \mathbb{1}\big(x \in \Xi_i(s)\big), \tag{5}$$

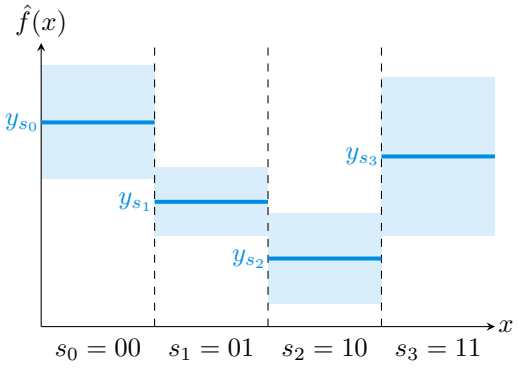


Fig. 1. The BTGP contains functions that are piecewise constant over the partitions of $\mathbb{X}$. The posterior mean $\hat{\mu}_N$ and double standard deviation $\hat{\mu}_N \pm 2\hat{\sigma}_N$ are displayed as a piecewise-constant graph and an area plot in light gray, respectively. For this example with chosen precision $q = 2$, the posterior mean takes values $\hat{\mu}_N \in \{y_{s_0}, \dots, y_{s_3}\}$.

for some $y_s \in \mathbb{R}$. The resulting finite dimensional RKHS $\mathcal{H}_{\hat{k}_q}$ (of dimension $\sum_{i=1}^{q} 2^i$) is characterized by the piecewise-constant feature map

$$\hat{\phi}_q(x) := \left\{ \sqrt{w_i} \mathbb{1}(x \in \Xi_i(s)) \,|\, s \in \mathbb{B}^i, i = 1, \dots, q \right\}. \quad (6)$$

A measure for the complexity of the functions in a given RKHS is the associated RKHS norm. For the BT kernel, a bound on the RKHS norm can be computed as follows.

*Theorem 2.* (BT RKHS norm bound). Let $\hat{k}_q : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ be the BT kernel in (4) and $\mathcal{H}_{\hat{k}_q}$ its associated RKHS. For all functions $f \in \mathcal{H}_{\hat{k}_q}$ of the form (5), we have that

$$||f||_{\mathcal{H}_{\hat{k}_q}} \leq \sqrt{\textstyle\sum_{i=1}^{q} w_i \sum_{s \in \mathbb{B}^i} y_s^2}.$$

### *3.3 Statistical Error Bound*

To use GP regression in the context of formal approaches, we must quantify the error between the unknown data-generating function $f \in \mathcal{H}_k$ in (1) and the GP posterior mean $\hat{\mu}_N \in \mathcal{H}_{\hat{k}_q}$ in (2). To be able to address a wealth of functions $f$ beyond those discontinuous functions living in the RKHS $\mathcal{H}_{\hat{k}_q}$ associated with the BT kernel, we leverage Assumption 1 to quantify the additional error introduced by the erroneous function space. The following theorem takes inspiration from Theorem 2.1 by Hsu et al. (2012) and Proposition 2 by Fiedler et al. (2021), where latter is itself an adaptation of Theorem 2 by Chowdhury and Gopalan (2017). As mentioned before, the output dimensions – indexed by $d$ – are modeled each by one GP.

*Theorem 3.* (Piecewise-constant error bound). Consider the BT kernel $\hat{k}_q$ in (4) with precision $q \in \mathbb{N}$ and the associated RKHS $\mathcal{H}_{\hat{k}_q}$. Let the unknown function $f \in \mathcal{H}_k$ with a known translation-invariant kernel $k$ that has a constant $c$ satisfying $c^2 \geq k(x,x) \geq 0$ for all $x \in \mathbb{X}$, and a complexity bound $B \geq ||f||_{\mathcal{H}_k} \geq 0$ (Assumption 1). Given data $\mathcal{D}_N = (X_N, Y_N)$ and a constant $\delta \in (0,1)$, we have for each dimension $d \in \{1, \dots, n\}$ for the corresponding BTGP posterior mean $\hat{\mu}_{N,d}(s) \in \mathcal{H}_{\hat{k}_q}$ in (2) that

$$\mathbb{P}\Big( \forall s \in \mathbb{B}^q : \sup_{x \in \Xi(s)} \left| \hat{\mu}_{N,d}(s) - f_d(x) \right| \leq \varepsilon_d(s) \Big) \geq 1 - \delta, \quad (7)$$

with the error $\varepsilon_d(s) := \varepsilon_{d,1}(s) + \varepsilon_{d,2}(s) + \varepsilon_{d,3}(s)$ given by

$$\varepsilon_{d,1}(s) := \min \Big\{ \big|\big|\hat{C}_N^{-1} \hat{k}_N(s)\big|\big| \sqrt{N + 2\left(N \log \tfrac{1}{\delta}\right)^{\frac{1}{2}} + 2 \log \tfrac{1}{\delta}},$$
$$\tfrac{\hat{\sigma}_{N,d}(s)}{\sigma_v} \sqrt{\operatorname{tr}(\Sigma) + 2\left(\operatorname{tr}(\Sigma^2) \log \tfrac{1}{\delta}\right)^{\frac{1}{2}} + 2||\Sigma|| \log \tfrac{1}{\delta}} \Big\},$$
$$\varepsilon_{d,2}(s) := B\big(2\left|c^2 - k_{x_s}^+(s)\right|\big)^{\frac{1}{2}},$$
$$\varepsilon_{d,3}(s) := B\left|\hat{k}_N(s)^\top \hat{C}_N^{-1}\Big(K_N \hat{C}_N^{-1} \hat{k}_N(s) - 2k_N(x_s)\Big) + c^2\right|^{\frac{1}{2}},$$

where $\hat{C}_N := (\hat{K}_N + \sigma_v^2 I_N)$, $\hat{k}_N(s) := [k(x_i, \Psi_q(s))]_{i=1}^N$, the posterior covariance $\hat{\sigma}_{N,d}^2(s)$ in (3), $\Sigma := \hat{K}_N \hat{C}_N^{-1}$, $k_{x_s}^+(s) := \sup_{x \in \Xi(s)} k(x_s, x)$ for some fixed set of representative points $\{x_s\}_{s \in \mathbb{B}^q} \subset \mathbb{X}$, $K_N := [k(x_i, x_j)]_{i,j=1}^N$, and $k_N(x) := [k(x_i, x)]_{i=1}^N$.

*Remark 4.* (Approximating a continuous kernel). The performance benefits of GP regression via the BT kernel stem from the fact that it constitutes a series of subsequent partitioning of the input space such that in every layer the precision of the previous partitioning is refined (or kept the same). For each layer, functional mass represented by the corresponding weight coefficient is applied whenever two points fall into the same partition. As shown in Theorem 1 by Cohen et al. (2022b), this allows for an efficient computation of the Gram matrix and its inverse (computations for the inversion have a complexity for every precision level $i := 1, \dots, q$ of $O(N)$ instead of $O(iN)$, which overall leads to $O(qN)$).

*Remark 5.* The above theorem establishes a finite-sample bound on the error between the posterior mean of the BTGP and the actual unknown dynamics $f$. The complexity bound $B$ on $||f||_{\mathcal{H}_k}$ enables us to provide guarantees even when $f$ does not live in the RKHS of the BTGP, $\mathcal{H}_{\hat{k}_q}$, e.g., if $f$ is continuous. This is called the 'misspecified case'. In comparison to Fiedler et al. (2021), we quantify the introduced error $\varepsilon_d$ without computing a global upper bound on the error between the two kernels, which would be at its maximum in our case as $\hat{k}_q$ does generally not approximate $k$. The first term in the minimization of $\varepsilon_{d,1}$ is equivalent to prior works and captures the statistical error of the properly specified case, which is generally very conservative. We provide an alternative second term in an effort to reduce this error by exploiting the low-dimensional BT kernel Gram matrices. Nevertheless, note that whilst the error bounds $(\varepsilon_{2,d}, \varepsilon_{3,d})$ can be tightened by increasing the precision $q$, this is not the case for $\varepsilon_{d,1}$.

Determining the error bound $\varepsilon_{d,3}$ requires us to compute Gram matrices via both $k$ and $\hat{k}_q$, reducing the scalability benefits of the BTGP. It is interesting to study the quantification of $\varepsilon_d$ without the need for constructing $K_N$.

One might be tempted to use the BT kernel in Definition 3, or an analogously defined alternative kernel bearing the same properties, to approximate a continuous kernel such as the squared-exponential kernel for the limit case $q \to \infty$, allowing for a simplification of Theorem 3. However, such a kernel does not exist. In fact, let $\hat{k}_q(x, x')$ be such a kernel with precision $q \in \mathbb{N}$ and consider $x \in \mathbb{X}$ to be fixed and $x' \in \mathbb{X}$ variable. Then, for $q < q' \in \mathbb{N}$ the kernels $\hat{k}_q$ and $\hat{k}_{q'}$ define two discrete distance metrics where the exponential increase in the level of precision of $\hat{k}_{q'}$ compared to $\hat{k}_q$ is

focused solely on the neighborhood of $x$. Hence, escalating the level of precision merely increases the smoothness of the covariance in the neighborhood of $x$ and has no effect on the quantification elsewhere.

In contrast to methods such as probabilistic coupling relations (Schön et al., 2023) where the error between the true system and the discrete abstraction is captured by computing local parameters bounding the local deviations, the error bound $\varepsilon_d(s)$ in Theorem 3 summarizes the error in a formula that similarly yields local deviations when evaluated for individual partitions $s \in \mathbb{B}^q$.

## 4. VERIFICATION APPROACH

The verification approach we consider is based on the construction of a finite-state abstraction of the dynamical system in (1) via the BTGP introduced in Sec. 3.2. For the learned BTGP, let $\mathbb{B}^q$ be a finite set of binary strings representing the states of this abstraction. Recall that the BTGP defines a map $\Psi_q : \mathbb{X} \to \mathbb{B}^q$ that maps every state $x \in \mathbb{X}$ from the continuous state space $\mathbb{X}$ to a string $s \in \mathbb{B}^q$ associated with a partition of $\mathbb{X}$ and a representative state $x_s \in \mathbb{X}$. To capture the probabilistic ambiguity w.r.t. transitions between discrete partitions, recent approaches for verifying infinite-horizon specifications are based on *interval Markov chains* (IMCs), which can be learned from data via GP regression (Badings et al., 2023; Lavaei et al., 2022; Jackson et al., 2020). In this manuscript, we follow a similar approach, however, we perform GP regression using the BT kernel introduced in Sec. 3.2, that allows us to reduce significantly the computational effort in constructing an IMC from data.

*Definition 4.* (Interval Markov chain (IMC)). An IMC is a tuple $\widehat{\mathbf{M}} := (\mathbb{S}, s_{\text{init}}, \underline{\mathbf{t}}, \bar{\mathbf{t}})$, with a finite state space $\mathbb{S}$ of discrete states $s \in \mathbb{S}$; initial state $s_{\text{init}} \in \mathbb{S}$; and probability bounds $\underline{\mathbf{t}}, \bar{\mathbf{t}} : \mathbb{S} \times \mathbb{S} \to [0, 1]$ describing lower and upper bounds on the transition probability.

### 4.1 Computing the Abstract IMC via BTGP

The abstract IMC $\widehat{\mathbf{M}} := (\mathbb{S}, s_{\text{init}}, \underline{\mathbf{t}}, \bar{\mathbf{t}})$ is constructed with the state space $\mathbb{S} := \mathbb{B}^q$, and the initial state $s_{\text{init}} := \Psi_q(x_{\text{init}})$ with $x_{\text{init}}$ being the initial state of the system $\mathbf{M}$. Recall that we use one BTGP to model the dynamics for each output dimension. We hence obtain $n$ BTGPs $\hat{f}_d \sim \mathcal{GP}(\hat{\mu}_{N,d}, \hat{\sigma}^2_{N,d})$, $d \in \{1, \ldots, n\}$, where we use the same precision $q \in \mathbb{N}$ to generate a common partitioning $\mathbb{S} = \Psi_q(\mathbb{X})$. Computing the discrete transition kernels $\underline{\mathbf{t}}, \bar{\mathbf{t}}$ from the constructed BTGPs is comparatively straightforward due to the piecewise-constant nature of the posterior mean and covariance functions of the BTGP. In other words, for every partition $s \in \mathbb{S}$, the consecutive state (for each dimension $d \in \{1, \ldots, n\}$) is distributed according to a Gaussian distribution whose mean is uncertain in the form of $\hat{\mu}_{N,d}(s) + \epsilon_d(s)$ with error $\epsilon_d(s) \in E_d(s) := [-\varepsilon_d(s), \varepsilon_d(s)]$ obtained via the corresponding stochastic learning errors (Theorem 3). The Gaussian distribution will have the covariance $\hat{\sigma}^2_{N,d}(s)$ (see Fig. 2 bottom). We obtain lower and upper bounds on the probability of transitioning from $s$ to any $s' \in \mathbb{S}$ as

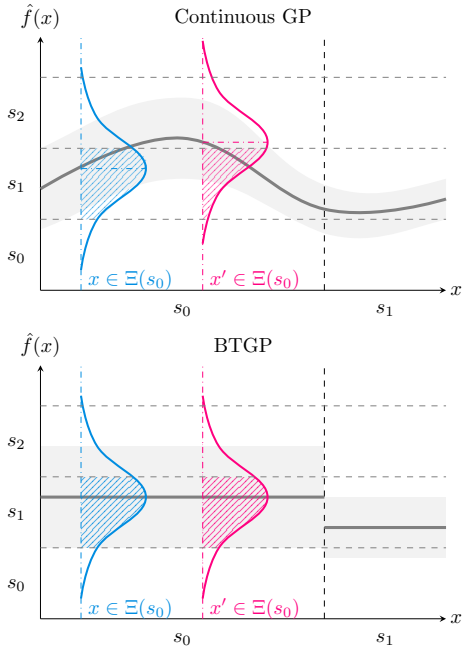


Fig. 2. Computing transition probabilities from a continuous GP (top) vs. from a BTGP (bottom).

$$\underline{\mathbf{t}}(s'|s) = \min_{\epsilon \in E(s)} g(s, \epsilon) \quad \text{and} \quad \bar{\mathbf{t}}(s'|s) = \max_{\epsilon \in E(s)} g(s, \epsilon), \tag{8}$$

$$g(s, \epsilon) := \int_{x' \in \Xi(s')} \prod_{d=1}^{n} \mathcal{N}(dx'_d | \hat{\mu}_{N,d}(s) + \epsilon_d, \hat{\sigma}^2_{N,d}(s)), \tag{9}$$

with $E(s) := \prod_{d=1}^{n} E_d(s)$, $\epsilon = [\epsilon_d]_{d=1}^n$, and $x' = [x'_d]_{d=1}^n$. If the partitions of the BTGP are selected to be rectangular, the integration in (9) and subsequent optimizations in (8) can be done separately for each dimension. In comparison, for a kernel that is not piecewise-constant, e.g., the squared-exponential kernel in Fig. 2 (top), obtaining the bounds requires optimizing with respect to $\epsilon$ the function

$$g(s, \epsilon) = \underset{x \in \Xi(s)}{\mathsf{optimize}} \int_{x' \in \Xi(s')} \prod_{d=1}^{n} \mathcal{N}(dx'_d | \hat{\mu}_{N,d}(x) + \epsilon_d(x), \hat{\sigma}^2_{N,d}(x)), \tag{10}$$

with $\mathsf{optimize} \in \{\max, \min\}$ for each pair $(s, s')$. This shows that the BTGP eliminates the additional inner optimization, thus significantly simplifying the computational burden associated with constructing the IMC to one integration per bound per transition.

### 4.2 Verification Algorithm

In this section, we adapt the *interval iteration* from the work by Haddad and Monmege (2018) as given in Alg. 1, which is based on the iterative update of two *value functions* forming an envelope or *interval* around the true probability of satisfying the specification starting from some initial partition.

*Proposition 1.* (Value convergence). For a given IMC $\widehat{\mathbf{M}} := (\mathbb{S}, s_{\text{init}}, \underline{\mathbf{t}}, \bar{\mathbf{t}})$ and an infinite-horizon reachability specification $\psi_{\mathcal{T}}$, let $V^* : \mathbb{S} \to [0, 1]$ be the *optimal value function* such that $\mathbb{P}(\widehat{\mathbf{M}} \vDash \psi_{\mathcal{T}}) = V^*(s_{\text{init}})$. Then, the interval iteration in Alg. 1 converges to $V^*$ as $i \to \infty$. Furthermore, for every iteration $i \in \mathbb{N} \cup \{0\}$ we have $\underline{V}_i(s_{\text{init}}) \leq V^*(s_{\text{init}}) \leq \bar{V}_i(s_{\text{init}})$.

**Algorithm 1** Interval iteration algorithm

**Input:** IMC $\widehat{\mathbf{M}} = (\$, s_{\text{init}}, \underline{\mathbf{t}}, \overline{\mathbf{t}})$, specification $\psi_{\mathcal{T}}$, stopping threshold $\nu > 0$

1: **for** $s \in \$$ **do**
2: $\quad \overline{r}(s) \leftarrow \min\Big\{\sum_{s' \in \Psi_q(\mathcal{T})} \overline{\mathbf{t}}(s'|s),\, 1\Big\}$
3: $\quad \underline{r}(s) \leftarrow \sum_{s' \in \Psi_q(\mathcal{T})} \underline{\mathbf{t}}(s'|s)$
4: $\quad \overline{l}(s) \leftarrow 1 - \sum_{s' \in \$} \underline{\mathbf{t}}(s'|s)$
5: $\quad \underline{l}(s) \leftarrow \max\big\{0,\, 1 - \sum_{s' \in \$} \overline{\mathbf{t}}(s'|s)\big\}$
6: **end for**
7: $i \leftarrow 0$, $\overline{V}_0 \leftarrow 1$, $\underline{V}_0 \leftarrow 0$
8: **while** $\|\overline{V}_i - \underline{V}_i\|_\infty > \nu$ **do**
9: $\quad r, \mathbf{t} \leftarrow$ solve min LP in (11)
10: $\quad$ **for** $s \in \$$ **do**
11: $\quad\quad \overline{V}_{i+1}(s) \leftarrow r(s) + \sum_{s' \in \$ \setminus \Psi_q(\mathcal{T})} \mathbf{t}(s'|s) \overline{V}_i(s')$
12: $\quad\quad \underline{V}_{i+1}(s) \leftarrow r(s) + \sum_{s' \in \$ \setminus \Psi_q(\mathcal{T})} \mathbf{t}(s'|s) \underline{V}_i(s')$
13: $\quad$ **end for**
14: $\quad i \leftarrow i+1$, $\overline{V}_i \leftarrow \overline{V}_{i+1}$, $\underline{V}_i \leftarrow \underline{V}_{i+1}$
15: **end while**
16: $V_{\min} \leftarrow \underline{V}_i$ ▷ Take underapproximation
17: $i \leftarrow 0$, $\overline{V}_0 \leftarrow 1$, $\underline{V}_0 \leftarrow 0$
18: **while** $\|\overline{V}_i - \underline{V}_i\|_\infty > \nu$ **do**
19: $\quad r, \mathbf{t} \leftarrow$ solve max LP in (11)
20: $\quad$ **for** $s \in \$$ **do**
21: $\quad\quad \overline{V}_{i+1}(s) \leftarrow r(s) + \sum_{s' \in \$ \setminus \Psi_q(\mathcal{T})} \mathbf{t}(s'|s) \overline{V}_i(s')$
22: $\quad\quad \underline{V}_{i+1}(s) \leftarrow r(s) + \sum_{s' \in \$ \setminus \Psi_q(\mathcal{T})} \mathbf{t}(s'|s) \underline{V}_i(s')$
23: $\quad$ **end for**
24: $\quad i \leftarrow i+1$, $\overline{V}_i \leftarrow \overline{V}_{i+1}$, $\underline{V}_i \leftarrow \underline{V}_{i+1}$
25: **end while**
26: $V_{\max} \leftarrow \overline{V}_i$ ▷ Take overapproximation

**Output:** $V_{\min}, V_{\max}$

Before giving a summary of Alg. 1, let us define the following linear program (LP), where $\mathsf{optimize} \in \{\max, \min\}$:

$$\begin{aligned} \underset{r,\mathbf{t}}{\mathsf{optimize}} \quad & r(s) + \textstyle\sum_{s' \in \$ \setminus \Psi_q(\mathcal{T})} \mathbf{t}(s'|s) V_i(s'), \\ \text{s.t.} \quad & \underline{\mathbf{t}}(s'|s) \le \mathbf{t}(s'|s) \le \overline{\mathbf{t}}(s'|s), \\ & \underline{r}(s) \le r(s) \le \overline{r}(s), \\ & \underline{l}(s) \le l(s) \le \overline{l}(s), \\ & r(s) + l(s) + \textstyle\sum_{s' \in \$ \setminus \Psi_q(\mathcal{T})} \mathbf{t}(s'|s) = 1. \end{aligned} \tag{11}$$

In lines 1 to 4 of Alg. 1, we compute functional lower and upper bounds $(\underline{r}, \overline{r})$ on the *reward* – the probability of reaching the projected target region $\Psi_q(\mathcal{T}) \subset \$$ in one time-step starting from the state $s \in \$$. Analogously, $(\overline{l}, \underline{l})$ represent functional bounds on the *loss* – the probability of leaving the state space $\$$. In line 5, the overapproximation $\overline{V}_0$ and underapproximation $\underline{V}_0$ of the *value function* – the probability of *eventually* reaching the target starting from partition $s$ – are initialized. Until convergence, we repeat the following two steps. First, for every $s, s' \in \$$, we minimize the LP in (11) in line 7 to find a feasible reward $r$ and transition probabilities $\mathbf{t}(s'|s)$ that capture the worst-case behavior of the system. Then, we update $\overline{V}_0$ and $\underline{V}_0$ via the *Bellman equation* in lines 9 and 10. The while-loop converges when the maximal difference between the over and underapproximation is less than the predefined closeness bound $\nu$. After convergence, the underapproximation $\underline{V}_i$ is taken as the lower bound $V_{\min}$ on the value function in line 14. The same procedure is repeated to obtain an overapproximation $V_{\max}$ in lines 15 to 24, where the LP is maximized in line 17. From Proposition 1 we get that the true latent value $V^*$ lies between the obtained upper and lower bounds.

Following from Theorem 3 and Proposition 1, we provide a robust satisfaction probability for the latent true system in (1) satisfying a reachability specification $\psi_{\mathcal{T}}$.

*Theorem 4.* (Robust satisfaction probability). Consider the setup of Theorem 3. Let $\hat{f}_d \sim \mathcal{GP}(\hat{\mu}_{N,d}, \hat{\sigma}^2_{N,d})$, $d \in \{1, \ldots, n\}$ be the posterior BTGPs based on data $\mathcal{D}_N$ from the system $\mathbf{M}$ in (1). Let $\widehat{\mathbf{M}}$ be the corresponding IMC constructed as outlined in (8). Then, we have for an infinite-horizon reachability specification $\psi_{\mathcal{T}}$ and $\mathbf{M}$ initialized at any state $x_{\text{init}} \in \mathbb{X}$ that

$$V_{\min}(s_{\text{init}}) \le \mathbb{P}(\mathbf{M} \vDash \psi_{\mathcal{T}}) \le V_{\max}(s_{\text{init}}),$$

where $s_{\text{init}} := \Psi_q(x_{\text{init}})$ and $V_{\min}$, $V_{\max}$ are the lower and upper bounds on the value function obtained via Alg. 1.

## 5. CASE STUDY

Consider the discrete-time nonlinear system

$$\begin{bmatrix} x_{1,t+1} \\ x_{2,t+1} \end{bmatrix} = \begin{bmatrix} x_{1,t} - \tau_s x_{1,t} + 0.5\tau_s \sin(x_{2,t}) \\ x_{2,t} - \tau_s x_{2,t} + 0.5\tau_s \sin(x_{1,t}) \end{bmatrix} + v_t,$$

evolving on the continuous state space $\mathbb{X} = [-10, 10]^2 \subset \mathbb{R}^2$ with $\tau_s = 0.5$ and $v_t \sim \mathcal{N}(\cdot\,|0, \sigma_v^2 I_2)$. The goal is to compute a lower bound on the probability of the system reaching the target set $\mathcal{T} = [-3, 3]^2 \subset \mathbb{X}$ based on $N = 5000$ observations $\mathcal{D}_N$ from the system generated with $\sigma_v = 3.16$. We construct two BTGPs – one for each output dimension – with a precision of $q = 12$. Training and evaluation of the BTGPs at the representative states takes 13 seconds on an Apple MacBook Pro M1. To compute an error bound via Theorems 3, we choose the confidence bound $\delta := 0.2$, squared-exponential kernels $k_d(x, x') := c_d^2 \exp(-\frac{1}{2}(x - x')^\top M_d^{-1}(x - x'))$ for $d \in \{1, 2\}$ with hyperparameters $c_1 = 12$, $c_2 = 7$, and $M_d = \text{diag}(l_d^2)$ where $l_1 = [4000, 2500]$ and $l_2 = [500, 2000]$, and dimension-wise complexity bounds $B_1 = 0.015$, $B_2 = 0.006$. For $\{x_s\}_{s \in \mathbb{B}^q}$, we choose the center points of the partitions. Due to its piecewise-constant nature, generating the IMC via (8) from the BTGPs and error bounds takes less than 3 seconds. In comparison, abstraction of a comparable continuous GP via (10) takes around 8 hours. The IMC is subsequently verified via Alg. 1, which converges after 9 seconds ($\nu = 10^{-8}$). The resulting lower bound on the probability of reaching $\mathcal{T}$ is depicted as a function of the initial state in Fig. 3.

## 6. CONCLUSION

In this paper, we showcased how to use binary-tree Gaussian processes (BTGPs) for learning model representations from data that are naturally ideal for finite-state abstraction and verification due to the piecewise-constant nature of the employed binary tree kernel. We also formulated the error bounds in the learning, which were used to generate the abstraction as an interval Markov chain. We are currently working on improving the computation of the error bounds and benchmarking the proposed approach against alternative data-driven methods.

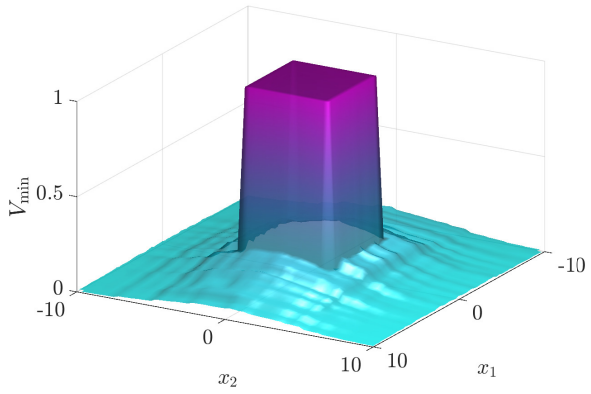


Fig. 3. Lower bound on the infinite-horizon reachability probability as a function of the initial state.

## REFERENCES


Abate, A., D'Innocenzo, A., Di Benedetto, M.D., and Sastry, S.S. (2008). Markov set-chains as abstractions of stochastic hybrid systems. In *International Workshop on Hybrid Systems: Computation and Control*, 1–15. Springer.

Abate, A., Edwards, A., and Giacobbe, M. (2022). Neural abstractions. *NeurIPS*.

Adams, S., Lahijanian, M., and Laurenti, L. (2022). Formal control synthesis for stochastic neural network dynamic models. *L-CSS*.

Badings, T., Romao, L., Abate, A., and Jansen, N. (2023). Probabilities are not enough: Formal controller synthesis for stochastic dynamical models with epistemic uncertainty. In *AAAI*.

Baier, C. and Katoen, J.P. (2008). *Principles Of Model Checking*, volume 950. MIT Press.

Berlinet, A. and Thomas-Agnan, C. (2011). *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media.

Bogachev, V.I. (2007). *Measure theory*. Springer Science & Business Media.

Cao, M.E., Bloch, M., and Coogan, S. (2022). Efficient learning of hyperrectangular invariant sets using Gaussian processes. *OJ-CSYS*.

Chowdhury, S.R. and Gopalan, A. (2017). On kernelized multi-armed bandits. In *ICML*.

Cohen, M.H., Belta, C., and Tron, R. (2022a). Robust control barrier functions for nonlinear control systems with uncertainty: A duality-based approach. In *CDC*.

Cohen, M.K., Daulton, S., and Osborne, M.A. (2022b). Log-linear-time Gaussian processes using binary tree kernels. *NeurIPS*.

Fiedler, C., Scherer, C.W., and Trimpe, S. (2021). Practical and rigorous uncertainty bounds for Gaussian process regression. In *AAAI*.

Haddad, S. and Monmege, B. (2018). Interval iteration algorithm for MDPs and IMDPs. *TCS*.

Haesaert, S., Nilsson, P., Vasile, C., Thakker, R., Aghamohammadi, A., Ames, A., and Murray, R. (2018). Temporal logic control of POMDPs via label-based stochastic simulation relations. *ADHS*.

Haesaert, S. and Soudjani, S. (2020). Robust dynamic programming for temporal logic control of stochastic systems. *TAC*.

Harapanahalli, A., Jafarpour, S., and Coogan, S. (2023a). Contraction-guided adaptive partitioning for reachability analysis of neural network controlled systems. *arXiv:2304.03671*.

Harapanahalli, A., Jafarpour, S., and Coogan, S. (2023b). Forward invariance in neural network controlled systems. *L-CSS*.

Hasanbeig, M., Abate, A., and Kroening, D. (2019). Logically-constrained neural fitted Q-iteration. In *AAMAS*.

Hsu, D., Kakade, S.M., and Zhang, T. (2012). A tail inequality for quadratic forms of subgaussian random vectors. *ECP*.

Jackson, J., Laurenti, L., Frew, E., and Lahijanian, M. (2020). Safety verification of unknown dynamical systems via Gaussian process regression. In *CDC*.

Jafarpour, S., Harapanahalli, A., and Coogan, S. (2023). Interval reachability of nonlinear dynamical systems with neural network controllers. In *L4DC*.

Jagtap, P., Soudjani, S., and Zamani, M. (2020). Formal synthesis of stochastic systems via control barrier certificates. *TAC*.

Jiang, J., Zhao, Y., and Coogan, S. (2022). Safe learning for uncertainty-aware planning via interval MDP abstraction. *L-CSS*.

Kanagawa, M., Hennig, P., Sejdinovic, D., and Sriperumbudur, B.K. (2018). Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv:1807.02582*.

Kazemi, M. and Soudjani, S. (2020). Formal policy synthesis for continuous-state systems via reinforcement learning. In *IFM*.

Lavaei, A., Somenzi, F., Soudjani, S., Trivedi, A., and Zamani, M. (2020). Formal controller synthesis for continuous-space MDPs via model-free reinforcement learning. In *ICCPS*.

Lavaei, A., Soudjani, S., Frazzoli, E., and Zamani, M. (2022). Constructing MDP abstractions using data with formal guarantees. *L-CSS*.

Lew, T. and Pavone, M. (2021). Sampling-based reachability analysis: A random set theory approach with adversarial sampling. In *CoRL*.

Lopez, B.T. and Slotine, J.J.E. (2023). Unmatched control barrier functions: Certainty equivalence adaptive safety. In *ACC*.

Quinonero-Candela, J. and Rasmussen, C.E. (2005). A unifying view of sparse approximate Gaussian process regression. *JMLR*.

Rasmussen, C.E. and Williams, C.K.I. (2018). *Gaussian Processes for Machine Learning*. The MIT Press.

Salamati, A., Lavaei, A., Soudjani, S., and Zamani, M. (2024). Data-driven verification and synthesis of stochastic systems via barrier certificates. *Automatica*.

Schön, O., van Huijgevoort, B., Haesaert, S., and Soudjani, S. (2023). Bayesian approach to temporal logic control of uncertain systems. *arXiv:2304.07428*.

Thorpe, A., Lew, T., Oishi, M., and Pavone, M. (2022). Data-driven chance constrained control using kernel distribution embeddings. In *L4DC*.

Wajid, R., Awan, A.U., and Zamani, M. (2022). Formal synthesis of safety controllers for unknown stochastic control systems using Gaussian process learning. In *L4DC*.

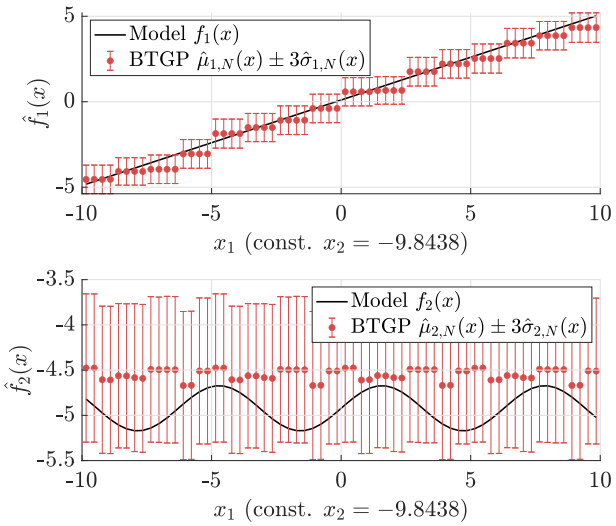


Fig. A.1. BTGP model fit $\hat{\mu}_{d,N}(x) \pm 3\hat{\sigma}_{d,N}(x)$ versus $f_d(x)$ (top: $d = 1$, bottom: $d = 2$).

## Appendix A. ADDITIONAL FIGURES ON THE CASE STUDY

Fig. A.1 illustrates the fit of the BTGPs for the system in the case study (Sec. 5). The top and bottom plots show, respectively for dimensions $d = 1, 2$, the posterior mean and triple standard deviation of the BTGPs as well as the actual model response for reference.

## Appendix B. PROOF OF THEOREM 1

We can write the BT kernel in Eq. (4) as the sum

$$\hat{k}_q(x, x') = \sum_{i=1}^{q} \kappa_i(x, x'), \tag{B.1}$$

of kernels $\kappa_i(x, x') := \phi_i(x)^\top \phi_i(x)$ with feature maps

$$\phi_i(x) := \sqrt{w_i}\,[\mathbb{1}(x \in \Xi_i(s))]_{s \in \mathbb{B}^i},$$

where we expanded each $i^{\text{th}}$ summand of $\hat{k}_q$ w.r.t. all possible binary substrings of length $i$. Note that the binary substrings $s \in \mathbb{B}^i$ in $\kappa_i : \mathbb{X} \times \mathbb{X} \to [0, 1]$ are unique and $\kappa_i$ is clearly positive-definite. The associated RKHS $\mathcal{H}_{\kappa_i}$ is of dimension $2^i$. Since finite dimensional spaces are automatically complete, $\mathcal{H}_{\kappa_i}$ is simply the span of the functions $\kappa_i(x, \cdot)$, $\forall x \in \mathbb{X}$. All functions $f_i \in \mathcal{H}_{\kappa_i}$ are hence linear combinations of the form

$$f_i(x) = \sum_{j=1}^{m} \alpha_j \kappa_i(x, x_j) = \sum_{j=1}^{m} \alpha_j \phi_i(x)^\top \phi_i(x_j),$$

for arbitrary $m \in \mathbb{N}$, $\alpha_j \in \mathbb{R}$, and $x_j \in \mathbb{X}$ (Berlinet and Thomas-Agnan, 2011). Note that $\phi_i(x_j)$ only places functional weight on the partition $s = \Psi_i(x_j)$. Therefore, $f_i \in \mathcal{H}_{\kappa_i}$ is always of the form

$$f_i(x) = y_i^\top \phi_i(x),$$

with $y_i \in \mathbb{R}^{2^i}$. Since the BT kernel is a weighted sum of kernels $\kappa_i$ as shown in (B.1), we obtain

$$f(x) = y^\top \hat{\phi}_q(x),$$

with $y^\top := [y_1^\top, \ldots, y_q^\top]$ and feature map $\hat{\phi}_q(x)^\top := [\phi_1(x)^\top, \ldots, \phi_q(x)^\top]$ via the kernel sum rule (Berlinet and Thomas-Agnan, 2011, Theorem 5), which is equivalent to (5). Note that $\hat{k}_q(x, x') = \langle \hat{\phi}_q(x), \hat{\phi}_q(x) \rangle_{\mathcal{H}_k}$ and $\mathcal{H}_{\hat{k}_q}$ is $(\sum_{i=1}^{q} 2^i)$-dimensional.

## Appendix C. PROOF OF THEOREM 2

We can write the BT kernel in (4) as the sum

$$\hat{k}_q(x, x') = \sum_{i=1}^{q} w_i \sum_{s \in \mathbb{B}^i} \kappa_s(x, x'),$$

of (positive-definite) kernels

$$\kappa_s(x, x') := \mathbb{1}(\Psi_{|s|}(x) = s)\mathbb{1}(\Psi_{|s|}(x') = s).$$

Following similar reasoning as in the proof of Theorem 1, we find

$$\begin{aligned}\mathcal{H}_{\kappa_s} &= \left\{ f_s \,|\, f_s = y_s \mathbb{1}(\Psi_{|s|}(\,\cdot\,) = s),\, y_s \in \mathbb{R} \right\}, \\ &= \left\{ f_s \,|\, f_s = y_s \mathbb{1}(\Psi_{|s|}(\,\cdot\,) = s)\mathbb{1}(s_{\leq |s|} = s),\, y_s \in \mathbb{R} \right\}, \\ &= \left\{ f_s \,|\, f_s = y_s \kappa_s(\,\cdot\,, x_s),\, y_s \in \mathbb{R},\, x_s \in \Xi_{|s|}(s) \right\}.\end{aligned}$$

We compute the norm of $f_s \in \mathcal{H}_{\kappa_s}$ given $\kappa_s$ via

$$||f_s||^2_{\mathcal{H}_{\kappa_s}} \equiv \langle f_s, f_s \rangle_{\mathcal{H}_{\kappa_s}} = y_s^2 \langle \kappa_s(\,\cdot\,, x_s), \kappa_s(\,\cdot\,, x_s) \rangle_{\mathcal{H}_{\kappa_s}} = y_s^2,$$

where we used $\kappa_s(x_s, x_s) = 1$. From Aronszajn's sums of kernels theorem (Berlinet and Thomas-Agnan, 2011, Theorem 5) we have $\forall f \in \mathcal{H}_{\hat{k}_q}$

$$||f||^2_{\mathcal{H}_{\hat{k}_q}} \leq \sum_{i=1}^{q} w_i \sum_{s \in \mathbb{B}^i} ||f_s||^2_{\mathcal{H}_{\kappa_s}},$$

concluding the proof.

## Appendix D. PROOF OF THEOREM 3

Similar to the proof of Theorem 2 by Chowdhury and Gopalan (2017), we start by splitting the error bound into one term related to the noise and one to the statistical approximation error. For the data $\mathcal{D}_N = (X_N, Y_N)$, let $Y_d := [y_{1,d}, \ldots, y_{N,d}]^\top$ and recognize that we have $Y_d \equiv F_d + V_d$, with $F_d := [f_d(x_1), \ldots, f_d(x_N)]^\top$ and $V_d := [v_{1,d}, \ldots, v_{N,d}]^\top$. Via Cauchy-Schwarz we obtain

$$\begin{aligned}&\left|\hat{\mu}_{N,d}(s) - f_d(x)\right| \\ &\quad \leq \left|\hat{k}_N(s)^\top \hat{C}_N^{-1} V_d\right| + \left|\hat{k}_N(s)^\top \hat{C}_N^{-1} F_d - f_d(x)\right|.\end{aligned} \tag{D.1}$$

where we abbreviate $\hat{C}_N^{-1} := [\hat{K}_N + \sigma_v^2 I_N]^{-1}$. We bound the noise term following similar steps to Proposition 2 by Fiedler et al. (2021) to get

$$\left|\hat{k}_N(s)^\top \hat{C}_N^{-1} V_d\right| \leq \left|\left|\hat{C}_N^{-1} \hat{k}_N(s)\right|\right| \sqrt{N + 2\left(N \log \tfrac{1}{\delta}\right)^{\frac{1}{2}} + 2 \log \tfrac{1}{\delta}},$$

which yields the first term in the minimization of $\varepsilon_{d,1}(s)$. As an alternative bound, we can follow the same steps as in the proof of Theorem 2 by Chowdhury and Gopalan (2017) to derive that

$$|\hat{k}_N(s)^\top \hat{C}_N^{-1} V_d| \leq \sigma_v^{-1} \hat{\sigma}_{N,d}(s) \sqrt{V_d^\top \hat{K}_N \hat{C}_N^{-1} V_d}.$$

From (Hsu et al., 2012, Theorem 2.1) we have that

$$\mathbb{P}\Big( ||\Sigma V_d||^2 \leq \operatorname{tr}(\Sigma) + 2\left(\operatorname{tr}(\Sigma^2) t\right)^{\frac{1}{2}} + 2||\Sigma|| t \Big) \geq 1 - e^{-t},$$

where $\Sigma := \hat{K}_N \hat{C}_N^{-1}$. Setting $t := \log\left(\frac{1}{\delta}\right)$ yields the second term in the minimization of $\varepsilon_{d,1}(s)$. Depending on the setting, one or the other bound will provide a less conservative error for $\varepsilon_{d,1}(s)$.

Next, we select some fixed $x_s \in \mathbb{X}$ for all $s \in \mathbb{S}$ such that $\bar{f}_d(s) := f_d(x_s)$. Expanding the second term in (D.1) with $\bar{f}_d(s)$ and Cauchy-Schwarz yields for every $s \in \mathbb{S}$ that for all $x \in \Xi(s)$

$$\begin{aligned}&\left|\hat{k}_N(s)^\top \hat{C}_N^{-1} F_d - f_d(x)\right| \\ &\quad \leq \left|\hat{k}_N(s)^\top \hat{C}_N^{-1} F_d - \bar{f}_d(s)\right| + \left|\bar{f}_d(s) - f_d(x)\right|.\end{aligned} \tag{D.2}$$

The latter term is bounded from above via

$$\begin{aligned}\left|\bar{f}_d(s) - f_d(x)\right| &= \left|(\phi(x_s) - \phi(x))^\top f_d\right|, \\ &\leq \left|\left|\phi(x_s) - \phi(x)\right|\right|_{\mathcal{H}_k} \left|\left|f_d\right|\right|_{\mathcal{H}_k}, \\ &= \left|k(x_s, \bar{x}_s) - 2k(x_s, x) + k(x, x)\right|^{\frac{1}{2}} \left|\left|f_d\right|\right|_{\mathcal{H}_k}, \\ &\leq \varepsilon_{d,2}(s).\end{aligned}$$

Let $\Phi_N := [\phi(x_1), \ldots, \phi(x_N)]^\top$. With this and Cauchy-Schwarz, the prior term in (D.2) yields

$$\begin{aligned}&\left|\hat{k}_N(s)^\top \hat{C}_N^{-1} F_d - \bar{f}_d(s)\right| \\ &\qquad \leq \left|\left|\hat{k}_N(s)^\top \hat{C}_N^{-1} \Phi_N - \phi(x_s)^\top\right|\right|_{\mathcal{H}_k} \left|\left|f_d\right|\right|_{\mathcal{H}_k},\end{aligned}$$

which reduces to $\varepsilon_{d,3}(s)$, concluding the proof.