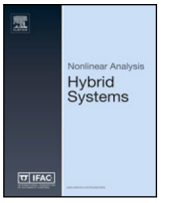
Nonlinear Analysis: Hybrid Systems 57 (2025) 101606

Contents lists available at ScienceDirect

Nonlinear Analysis: Hybrid Systems

journal homepage: www.elsevier.com/locate/nahs



Conformal quantitative predictive monitoring of stochastic systems with conditional validity

Francesca Cairoli [a],*, Tom Kuipers [b], Luca Bortolussi [a], Nicola Paoletti [b]

[a] University of Trieste, Department of Mathematics, Informatics and Geosciences, Italy
[b] King's College London, Department of Informatics, United Kingdom

ARTICLE INFO

Keywords:
Predictive monitoring
Stochastic process
Conformal quantile regression
Multi-agent
Conditional validity

ABSTRACT

We consider the problem of predictive monitoring (PM), i.e., predicting at runtime the satisfaction of a desired property from the current system's state. Due to its relevance for runtime safety assurance and online control, PM methods need to be efficient to enable timely interventions against predicted violations, while providing correctness guarantees. We introduce *quantitative predictive monitoring (QPM)*, a PM method to support stochastic processes and rich specifications given in Signal Temporal Logic (STL). QPM provides a quantitative measure of satisfaction of some property ϕ by predicting its quantitative (a.k.a. robust) STL semantics, either spatial or temporal. QPM derives prediction intervals that are highly efficient to compute and with probabilistic guarantees, in that the intervals cover with arbitrary probability the STL robustness values relative to the stochastic evolution of the system. To do so, we take a machine-learning approach and leverage recent advances in conformal inference for quantile regression, thereby avoiding expensive Monte Carlo simulations at runtime to estimate the intervals. We also show how our monitors can be combined in a compositional manner to handle composite formulas, without retraining the predictors or sacrificing the guarantees. We further equip QPM with techniques to ensure conditional validity of the prediction intervals, i.e., such that the probabilistic guarantees hold relative to any state of the system (or any satisfaction value), thereby significantly enhancing the consistency and reliability of the resulting monitor. We demonstrate the effectiveness and scalability of QPM over a benchmark of five discrete-time stochastic processes with varying degrees of complexity, including a stochastic multi-agent system.

1. Introduction

Predictive monitoring (PM) [1–3] is the problem of predicting at runtime the satisfaction of a certain requirement from the current system's state. Unlike traditional monitoring [4], PM has the potential to detect failures before they occur, thereby enabling preemptive countermeasures, such as switching to a fail-safe mode [5]. To enable effective deployment at runtime, PM methods must be efficient and respond quickly to prevent any system failure in time.

Performing model checking at run-time would provide a precise solution to the PM problem (precise up to the accuracy of the system's model), but such a solution is computationally expensive in general unless the model is trivial or fully deterministic. In

* Corresponding author.
E-mail address: francesca.cairoli@units.it (F. Cairoli).

https://doi.org/10.1016/j.nahs.2025.101606
Received 23 February 2024; Received in revised form 30 January 2025; Accepted 18 April 2025
Available online 14 May 2025

particular, when the system is stochastic, a statistical model checking solution [6] would require simulating at runtime a typically large number of Monte-Carlo trajectories to achieve desired error levels.[1]

For this reason, several approximate PM techniques based on machine learning have been recently proposed (see e.g. [7–11]), including the so-called Neural Predictive Monitoring (NPM) method [1–3,12]. In NPM in particular, the predictive monitor is a neural network classifier trained using data generated through a model checker to predict for any system state whether or not the state satisfies some reachability property. To improve the reliability of the reachability predictions, NPM relies on conformal prediction [13,14] to produce prediction regions with guaranteed coverage and derive uncertainty measures used to infer whether a particular prediction can be trusted.

This paper builds on [15] that introduces a predictive monitoring method for stochastic processes and Signal Temporal Logic (STL) [16] specifications, referred to as *quantitative predictive monitoring (QPM)*. QPM predicts the quantitative semantics R_ϕ of an STL formula ϕ (a.k.a. the STL robustness) for a general class of stochastic systems and offers probabilistic validity guarantees. QPM is inspired by NPM, but it addresses two significant limitations of the latter, which support only Boolean reachability specifications (as opposed to the full spectrum of STL properties and their quantitative interpretation) and cannot adequately deal with stochastic dynamics.

By supporting arbitrary STL specifications, QPM gains expressiveness in the type of requirements that can be monitored and thus in the type of violations that can be detected. In particular, by predicting STL robustness values, QPM provides key quantitative information on the degree of property satisfaction, unlike existing predictive monitoring approaches that can only predict Boolean satisfaction. STL robustness predictions can be meaningfully used to determine the extent of any corrective actions and enable efficient online model predictive control for STL [17]: for instance, depending on the specification ϕ, if QPM predicts a high value of R_ϕ, then little or no intervention might be needed, while a low R_ϕ value might require a more substantial or quicker intervention to steer the system back to safety. In this paper, the STL robustness is considered both in its spatial and temporal formulations.

A major challenge when dealing with stochastic processes is that every state induces a distribution of robustness values (conditional on the state) relative to the future stochastic evolution of the system. In general, such distribution is analytically intractable[2] and an accurate empirical estimate of such conditional distribution can be very expensive to obtain, potentially requiring a high number of Monte-Carlo simulations.

Our approach overcomes this computational bottleneck by deriving monitors that are able to directly predict some relevant quantiles of the conditional STL robustness distribution. Such quantiles have a two-fold purpose. First, they provide a *measure of risk* [18,19]: for instance, if the 10%-quantile of R_ϕ is zero, then "only" 10% of the system's future trajectories will violate ϕ (i.e., lead to a negative R_ϕ), which, depending on the application, can be interpreted as a low-risk scenario. Second, and most crucially, it allows us to derive *prediction intervals* that cover a certain mass of probability. For small α, if the $(\alpha/2, 1-\alpha/2)$-quantile interval for R_ϕ is entirely above (below) zero, then we can be fairly confident that the system will evolve into satisfying (violating) ϕ with high probability. On the other hand, intervals straddling the zero denote states whose future trajectories may or may not violate the property, which can be thus regarded as uncertain.

To predict the quantiles reliably, our approach builds on *Conformalized Quantile Regression (CQR)* [20], a recent conformal inference technique that produces statistically valid prediction intervals on top of quantile regression models. Crucially, this method provides us with marginal probabilistic guarantees, in that, for an arbitrary error probability $\alpha \in (0,1)$, the resulting interval for R_ϕ is guaranteed to cover the true STL robustness value with probability at least $1-\alpha$. With marginal guarantees, we mean that the coverage probability is satisfied on average w.r.t. the distribution of the stochastic system's trajectories. In this paper, we go beyond the marginal case and study techniques to obtain prediction intervals with guaranteed conditional coverage [21,22], meaning that guarantees are met at each system state, and not only on average. Such guarantees are akin to those of statistical model checking, with the key difference that CQR prediction intervals are extremely fast to compute and, thus, can be used at runtime.

By default, our QPM approach derives a monitor for a fixed choice of STL property ϕ. On one hand, this is advantageous in that the monitor is tailored and hence, highly accurate at predicting R_ϕ, but on the other hand, it lacks flexibility. To this purpose, we further show how monitors trained for different properties ϕ_1 and ϕ_2 can be combined in a modular way to monitor Boolean combinations of ϕ_1 and ϕ_2 without retraining the underlying models and preserving the desired probabilistic guarantees.

Contributions. This paper extends [15], where we introduce QPM as a predictive monitoring method that supports stochastic processes and the quantitative spatial semantics of arbitrary STL specifications. In [15], we show how QPM provides prediction intervals for STL robustness with marginal probabilistic guarantees. We demonstrate its compositionality by showing that composite formulas can be monitored by combining the QPM monitors of the subformulas, and we evaluate the effectiveness and scalability of QPM on several stochastic processes. In this paper, we introduce the following *novel contributions*:

(i) We extend the statistical guarantees beyond marginal coverage. In particular, we introduce *input-conditional guarantees* using a conformalized localization technique [21,22] that reweights calibration samples based on a measure of proximity with the considered test input, so that every test state is associated with a prediction interval with quasi-conditional guarantees, meaning that guarantees hold within a defined neighborhood of the conditioning input.

[1] Numerical/symbolic probabilistic model checking techniques typically are at least as expensive as statistical methods and can be applied only to a restricted class of models.

[2] With simple stochastic models and simple properties, it may become tractable but still impractical to analyze at runtime.

(ii) We also extend the marginal guarantees of [15] to better address scenarios with unbalanced datasets, where, e.g., trajectories satisfying the specification (a.k.a. positive trajectories) outnumber those violating it (negative trajectories). In particular, we provide conformalized prediction intervals with *sign-conditional coverage guarantees*, meaning that there is an equal probability of positive samples being covered as there is for negative ones. Since failures are rare to observe, this extension enhances the reliability of our QPM in detecting safety hazards.
(iii) We combine the methods (i) and (ii) to obtain prediction intervals that offer both sign- and input-conditional guarantees.
(iv) We derive monitors not just for STL space robustness but also for *STL time robustness* [16,23], which quantifies how much a signal can be perturbed in time (as opposed to space) before affecting its Boolean satisfaction value. This allows us to monitor the imminence of failures, thus providing crucial insights into the available time for an intervention.
(v) We introduce a new *multi-agent* case study, a multi-particle environment where we monitor collision-avoidance properties from partial observations of the state space.

In predictive monitoring applications, the importance of conditional guarantees cannot be overstated, as they significantly improve the monitor's accuracy, reliability, and consistency. These stricter guarantees are particularly important in contexts where predictions drive critical decision-making, risk management, and process optimization. Relying solely on marginal guarantees could lead to catastrophic errors for a small proportion of system states — errors that, instead, a conditionally valid monitor can prevent. Furthermore, it is common for real-world datasets to display imbalances, where marginal guarantees often fail to adequately cover less prevalent groups. This issue is particularly pronounced in predictive monitoring, where failures may be rare. Implementing class-balanced guarantees effectively addresses this challenge, ensuring the same coverage levels regardless of the satisfaction value.

2. Problem statement

We illustrate the predictive monitoring problem we target with QPM, after introducing background on stochastic processes and Signal Temporal Logic.

2.1. Stochastic processes

The systems we consider can be modeled as stochastic processes. A stochastic process is defined as a collection of random variables indexed by some index set T. These random variables are defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is the sample space, $\mathcal{F}$ is the σ-algebra and $\mathbb{P}$ is the probability measure. We can denote the stochastic process as $\{\mathbf{S}(t,\omega), t \in T\}$. A random variable $\mathbf{S}(t,\omega)$ in the collection is thus a function of the time variable $t \in T$ and the sample variable $\omega \in \Omega$. In our application, the index set is countable and represents *discrete time* $T = \{0, 1, \dots\}$. Each random variable in the collection takes values in a space $S \subseteq \mathbb{R}^n$, the *state space* of dimension n, that should be measurable. A discrete-time step makes the stochastic process move from index i to index $i+1$. Given a stochastic process $\{\mathbf{S}(t,\omega) : t \in T\}$, then for any point $\omega \in \Omega$, the mapping $\mathbf{S}(\cdot,\omega) : T \to S$, is called a *realization*, or a *sample trajectory* of the stochastic process $\{\mathbf{S}(t,\omega) : t \in T\}$. We assume that the dynamics of the system are Markovian. This assumption is not strict as most systems of interest – Markov chains, stochastic hybrid systems (without non-determinism), and stochastic difference equations – are Markovian or can be made so by augmenting the state space.

2.2. Signal temporal logic

System requirements can be expressed via Signal Temporal Logic (STL) [16,24], which enables the specification of properties of dense-time, real-valued signals, and the automatic generation of monitors for testing properties on individual trajectories. The rationale of STL is to transform real-valued signals into Boolean ones, using formulae built on the following *STL syntax*:

$$\phi := true \mid \mu \mid \neg\phi \mid \phi \wedge \phi \mid \phi \; U_I \; \phi,$$

where $I \subseteq \mathbb{T}$ is a temporal interval, either bounded, $I = [a,b]$, or unbounded, $I = [a,+\infty)$, for any $0 \le a < b$. Atomic propositions μ are (non-linear) inequalities on the states of a signal $\vec{s}$ at a time t, $\mu = (g(\vec{s}(t)) > 0)$, where $g : S \to \mathbb{R}$ and $\vec{s}(t)$ is a state in S. From this essential syntax, it is easy to define other operators, used to abbreviate the syntax in an STL formula: $false := \neg true$, $\phi \vee \psi := \neg(\neg\phi \wedge \neg\psi)$, $F_I := true \; U_I \phi$ and $G_I := \neg F_I \neg\phi$. Monitoring the satisfaction of a formula is done recursively by leveraging the tree structure of the STL formula. The satisfaction relation is defined as follows.

$$\begin{aligned}
(\vec{s},t) \vDash \mu &\Leftrightarrow g(\vec{s}(t)) > 0 \\
(\vec{s},t) \vDash \neg\phi &\Leftrightarrow \neg((\vec{s},t) \vDash \phi) \\
(\vec{s},t) \vDash \phi_1 \wedge \phi_2 &\Leftrightarrow (\vec{s},t) \vDash \phi_1 \wedge (\vec{s},t) \vDash \phi_2 \\
(\vec{s},t) \vDash \phi_1 U_{[a,b]} \phi_2 &\Leftrightarrow \exists\, t' \in [t+a, t+b] \text{ s.t. } (\vec{s},t') \vDash \phi_2 \wedge \\
&\qquad\qquad \forall\, t'' \in [t,t'],\ (\vec{s},t'') \vDash \phi_1
\end{aligned}$$

Given a formula ϕ and a signal $\vec{s}$ over a bounded time interval we can define the Boolean satisfaction signal as $\chi^\phi(\vec{s},t) = 1$ if $(\vec{s},t) \vDash \phi$ and $\chi^\phi(\vec{s},t) = 0$ otherwise.

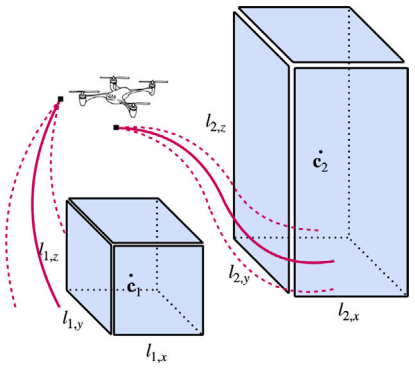


Fig. 1. Drone flying with stochastic dynamics. The STL requirement is to avoid collision with the nearby buildings.

Quantitative semantics. The robustness of a trajectory quantifies the level of satisfaction w.r.t. ϕ. Positive robustness means that the property is satisfied, whereas negative robustness means that the property is violated. Let h denote the maximum allowable signal length. Robustness is denoted as a function $R_\phi : S^h \times T \to \mathbb{R}$ that maps a given signal $\vec{s}$, a formula ϕ and a time t to some real value, $R_\phi(\vec{s}, t) \in \mathbb{R}$. It measures the maximum perturbation that can be applied to the signal without changing its truth value w.r.t. ϕ. In particular, we distinguish between *space robustness*, which deals with perturbations in the space dimension, and *time robustness*, which deals with perturbations in time.

Similarly to the Boolean semantics, the quantitative semantics of a formula ϕ over a signal $\vec{s}$ is defined recursively over the tree structure of the STL formula.

Space robustness is defined as a function $C_\phi : S^h \times T \to \mathbb{R}$ such that:

$$
\begin{aligned}
C_\mu(\vec{s}, t) &= g(\vec{s}(t)) \\
C_{\neg\phi}(\vec{s}, t) &= -C_\phi(\vec{s}, t) \\
C_{\phi_1 \wedge \phi_2}(\vec{s}, t) &= \min(C_{\phi_1}(\vec{s}, t), C_{\phi_2}(\vec{s}, t)) \\
C_{\phi_1 U_{[a,b]} \phi_2}(\vec{s}, t) &= \sup_{t' \in [t+a, t+b]} \Big(\min\big(C_{\phi_2}(\vec{s}, t'), \inf_{t'' \in [t, t']} C_{\phi_1}(\vec{s}, t'')\big)\Big).
\end{aligned}
$$

Similarly, *time robustness* captures the effect on the satisfaction of shifting the signal in time. The (right) time robustness of an STL formula ϕ with respect to a trace $\vec{s}$ at time t is defined by:

$$
Q^+_\phi(\vec{s}, t) = \chi^\phi(\vec{s}, t) \cdot \max\{d \geq 0 \ s.t. \ \forall t' \in [t, t+d], \ \chi^\phi(\vec{s}, t') = \chi^\phi(\vec{s}, t)\}.
$$

While space robustness is most common, our QPM approach can support any other kind of STL quantitative semantics, e.g., based on a combined space–time robustness [16] or resiliency [25]. Hereafter, we represent a generic STL monitor, encompassing either spatial or temporal quantitative satisfaction, as $R_\phi \in \{C_\phi, Q^+_\phi\}$.[3] The sign of R_ϕ indicates the Boolean satisfaction of signal $\vec{s}$, and in particular, we have $R_\phi(\vec{s}, t) > 0 \Rightarrow (\vec{s}, t) \vDash \phi$ and $R_\phi(\vec{s}, t) < 0 \Rightarrow (\vec{s}, t) \nvDash \phi$ [16,23].

Running example. Let us consider a drone flying at a constant velocity in a three-dimensional space (see Fig. 1). Given the system's current state s, a probabilistic controller regulates the yaw angle to avoid obstacles, e.g. buildings. The avoid property can be easily expressed as an STL formula:

$$
\phi_{avoid} := G_{[0,H]}\big(\ (\|s - \mathbf{c}_1\|_\infty > \mathbf{l}_1) \wedge (\|s - \mathbf{c}_2\|_\infty > \mathbf{l}_2) \ \big),
$$

where $\mathbf{c}_i$, $\mathbf{l}_i$ denote respectively the center and the edges of obstacle $i \in \{1, 2\}$ and H is the time horizon. Fig. 1 shows the evolution of stochastic dynamics for two different initial states. The dashed lines denote the upper and lower quantiles, whereas the solid line denotes the median of the distribution over the trajectory space.

2.3. Quantitative predictive monitoring

Given a stochastic process $\mathbf{S} = \{\mathbf{S}(t, \omega), t \in T\}$, an STL requirement ϕ and a state $s_0 \in S$, the robustness over future evolutions of the system starting from s_0 is stochastically distributed according to the conditional distribution

$$
p\left(R_\phi(\vec{s}, 0) \mid \vec{s}(0) = s_0\right),
$$

[3] Time-robustness assumes discrete values, here we define R_ϕ as a real-valued function considering that $\mathbb{Z} \subset \mathbb{R}$.

where $\vec{s}$ is a random signal given by the sequence of random variables $\vec{\mathbf{S}} := (\mathbf{S}(0,\cdot), \mathbf{S}(1,\cdot), \dots, \mathbf{S}(H-1,\cdot))$. For time k, $\mathbf{S}(k,\cdot)$ denotes the random variable corresponding to the state at time k. This conditional distribution captures the distribution of the STL robustness values for trajectories of length H starting in s_0.

We now present the quantitative predictive monitoring problem. Our goal is to construct prediction intervals that, with high probability, contain the true STL robustness value of future stochastic trajectories. We formalize this by learning a *monitoring function* $I : S \to \mathcal{I}(\mathbb{R})$, where $\mathcal{I}(\mathbb{R})$ denotes the set of all real-valued intervals, that maps each system state to an interval over robustness values. Given an error probability $\alpha \in (0,1)$, we require that the intervals produced by I achieve a coverage probability of at least $1-\alpha$. This coverage probability is computed through a two-step process: first, we sample an initial state s from $\mathbf{S}(0,\cdot)$, then we sample trajectories from the conditional distribution $p(\vec{s}|\vec{s}(0)=s)$. By averaging over these sampled trajectories, we obtain *marginal* coverage guarantees for our prediction intervals. A formal statement of the problem is given below.

Problem 1 (*Quantitative Predictive Monitoring*)**.**

Given a discrete-time stochastic process $\mathbf{S} = \{\mathbf{S}(t,\omega), t \in T\}$ over a state space S, temporal horizon H, an error probability $\alpha \in (0,1)$ and an STL formula ϕ, derive a monitoring function I producing regions that satisfy

$$\mathbb{P}_{\substack{s\sim\mathbf{S}(0,\cdot)\\ \vec{s}\sim p(\vec{s}|\vec{s}(0)=s)}}\Big(R_\phi(\vec{s},0) \in I\big(s\big)\Big) \geq 1-\alpha. \tag{1}$$

We will solve Problem 1 as a quantile regression problem. This boils down to learning, for a generic state s_0, an upper and a lower quantile of the random variable $R_\phi(\vec{s},0)$ induced by $p(\vec{s}|\vec{s}(0)=s_0)$.

Sign-conditional guarantees. Our observations can be divided into two classes, positive and negative, based on the sign of the robustness, i.e., on the value of the Boolean satisfaction of the property. We may be interested in obtaining coverage guarantees over every class, i.e., we may require that the prediction intervals are $1-\alpha$ accurate both when observations are positive and also when they are negative. Mathematically, sign-conditional guarantees are formulated as follows:

$$\mathbb{P}_{\substack{s\sim\mathbf{S}(0,\cdot)\\ \vec{s}\sim p(\vec{s}|\vec{s}(0)=s)}}\Big(R_\phi(\vec{s},0) \in I\big(\vec{s}(0)\big) \;\Big|\; R_\phi(\vec{s},0) \geq 0\Big) \geq 1-\alpha. \tag{2}$$

and

$$\mathbb{P}_{\substack{s\sim\mathbf{S}(0,\cdot)\\ \vec{s}\sim p(\vec{s}|\vec{s}(0)=s)}}\Big(R_\phi(\vec{s},0) \in I\big(\vec{s}(0)\big) \;\Big|\; R_\phi(\vec{s},0) < 0\Big) \geq 1-\alpha. \tag{3}$$

Input-conditional guarantees. Both (1), (2) and (3) provide a notion of marginal coverage, where coverage guarantees hold on average over the distribution of the stochastic process trajectories. Below, we formulate input-conditional guarantees, which are stronger as they instead require the same level of coverage for every initial state $s_* \sim \mathbf{S}(0,\cdot)$:

$$\mathbb{P}_{\vec{s}\sim p(\vec{s}|\vec{s}(0)=s_*)}\Big(R_\phi(\vec{s},0) \in I\big(\vec{s}(0)\big) \;\Big|\; \vec{s}(0)=s_*\Big) \geq 1-\alpha. \tag{4}$$

Running example (continued). Given a randomly selected current position $s_0 \sim \mathbf{S}(0,\cdot)$, the drone evolves according to the conditional distribution $p(\vec{s} \mid \vec{s}(0)=s_0)$. The monitoring function I produces an interval $I(s_0) \subseteq \mathbb{R}$. This interval either covers — value 1 — or not — value 0 — the true robustness value $R_{\phi_{avoid}}(\vec{s}\,',0)$ for a trajectory $\vec{s}\,' \sim p(\vec{s} \mid \vec{s}(0)=s_0)$. For a given $\alpha \in (0,1)$, the average of these coverage values over multiple initial states and over trajectories sampled starting from these states must be higher than $1-\alpha$.

In sign conditional guarantees, the same property holds averaging only w.r.t. trajectories leading respectively to positive or negative robustness values. Both classes must reach average coverage values higher than $1-\alpha$.

In input-specific guarantees, we average over trajectories starting from the same position. The average of the resulting coverage values must be higher than $1-\alpha$ for every possible initial position of the drone.

3. Background on conformal prediction

Consider a generic supervised learning setting where X denotes the input space, Y the target space, and $Z = X \times Y$. Let $\mathcal{Z}$ be the data-generating distribution, i.e., the distribution of the points $(x,y) \in Z$. We assume that the target y of a point $(x,y) \in Z$ is the result of the application of a function $f^* : X \to Y$, typically unknown or very expensive to evaluate. The goal of a supervised learning algorithm is to find a function $f : X \to Y$ that, from a finite set of observations, learns to behave as similarly as possible to f^* over the entire input space. For the sake of clarity, we start by showing conformal prediction approaches for deterministic predictors and then move to present Conformalized Quantile Regression (CQR) [20], an approach to handle probabilistic ones.

Conformal prediction (CP) associates measures of reliability with any traditional supervised learning problem, either regression or classification [13,14]. CP enriches point-wise predictions with *prediction regions with guaranteed marginal validity*.

We start by considering a dataset $Z_c = \{z_1, \dots, z_{N_c}\}$ of N_c independent and identically distributed pairs sampled from the unknown distribution $\mathcal{Z}$. The idea of CP is to construct a prediction region by "inverting" a suitable hypothesis test: given a test point x and a tentative target value y', we *exclude* y' from the prediction region only if it is unlikely that y' is the true value for x. The test statistic is given by a so-called *nonconformity function (NCF)* $\Delta : Z \to \mathbb{R}$, which, given a predictor f and a point $z=(x,y)$, measures the deviation between the true value y and the corresponding prediction $f(x)$. In this sense, Δ can be viewed as a generalized residual function. From a practical viewpoint, the NCF distribution $\mathbb{P}_{(x,y)\sim\mathcal{Z}}(\Delta(x,y))$ cannot be derived in an analytical form and thus we use an empirical approximation derived using a sample Z_c of $\mathcal{Z}$. This approach is called *inductive CP* [26] and Z_c

is referred to as *calibration set*. Given an error probability $\alpha \in (0,1)$, under the assumption that $(x,y),(x_1,y_1),\dots,(x_{N_c},y_{N_c})$ are i.i.d. or at least exchangeable, CP builds a probabilistically valid prediction region $B_\alpha : \mathbb{R}^{N_c} \to \mathbb{R}$ such that

$$\mathbb{P}\left(\Delta(x,y) \le B_\alpha(\beta_1,\dots,\beta_{N_c})\right) \ge 1-\alpha, \tag{5}$$

where $\beta_i = \Delta(x_i,y_i)$. From this bound over nonconformity scores, we can build a *prediction region* over target values, $\Gamma^\alpha(x;Z_c)$, by including all targets y' whose NCF values are likely to follow the NCF distribution of the true targets, i.e., have a nonconformity score lower than $B_\alpha(\beta_1,\dots,\beta_{N_c})$:

$$\Gamma^\alpha(x;Z_c) := \left\{ y' \in Y \,:\, \Delta(x,y') \le B_\alpha(\beta_1,\dots,\beta_{N_c}) \right\}. \tag{6}$$

This prediction region is guaranteed to contain the true (unknown) value y with confidence $1-\alpha$:

$$\mathbb{P}(y \in \Gamma^\alpha(x;Z_c)) \ge 1-\alpha. \tag{7}$$

We stress that the above guarantees are marginal, meaning averaged over the test and calibration data. They hold when (x,y) is exchangeable w.r.t. calibration data Z_c, i.e. when the joint probability of $(x_1,y_1),\dots,(x_{N_c},y_{N_c}),(x,y)$ is invariant to permutations.

A natural choice to derive the upper bound B_α from the calibration nonconformity scores $\beta_1,\dots\beta_{N_c}$ is the quantile function, i.e., $B_\alpha(\beta_1,\dots,\beta_{N_c}) := Q(1-\alpha;\mathcal{F}_c)$. More precisely, given $\alpha \in (0,1)$, $Q(1-\alpha;\mathcal{F}_c)$ denotes the $(1-\alpha)$-th quantile over the empirical distribution $\mathcal{F}_c = \frac{1}{N_c+1}\left(\sum_{z_i \in Z_c} \delta_{\beta_i} + \delta_\infty\right)$, with δ_{β_i} being the Dirac distribution centered at $\beta_i = \Delta(x_i,y_i)$ and ∞ is added as a correction to obtain finite sample guarantees. For a generic real-valued random variable T, the quantile function Q is formally defined as $Q(1-\alpha;T) = \inf\{t \in \mathbb{R} \mid \mathbb{P}(T \le t) \ge 1-\alpha\}$.

Validity and efficiency. CP performance is measured via two quantities: 1) *validity* (or *coverage*), i.e. the empirical error rate observed on a test sample, which should be as close as possible to the significance level α, and 2) *efficiency*, i.e. the size of the prediction regions, which should be small in order to have informative (i.e., non-trivial) regions. CP-based prediction regions are automatically valid, whereas the efficiency depends on the chosen nonconformity function and on the accuracy of the underlying model.

3.1. Conformal prediction for regression

In regression problems, we have a continuous target space $Y \subseteq \mathbb{R}^n$. The inductive CP algorithm is divided into an offline phase, executed only once, and an online phase, executed for every test point x_*. In the offline phase (steps 1–3 below), we train the classifier f and construct the calibration distribution, i.e., the empirical approximation of the NCF distribution. In the online phase (steps 4–5), we derive the prediction region for x_* using the computed regressor and distribution.

1. Draw sample Z' of $\mathcal{Z}$. Split Z' into training Z_t and calibration set Z_c.
2. Train regressor f using Z_t. Use f to define an NCF Δ.
3. Construct the empirical distribution $\mathcal{F}_c = \frac{1}{N_c+1}\left(\sum_{z_i \in Z_c} \delta_{\beta_i} + \delta_\infty\right)$ of calibration scores by computing, for each $z_i \in Z_c$, the NCF score $\beta_i = \Delta(z_i)$.
4. Identify the critical value $B_\alpha(\beta_1,\dots,\beta_{N_c}) := Q(1-\alpha;\mathcal{F}_c)$ of the calibration distribution, i.e. its empirical $(1-\alpha)$-quantile, or the $\lfloor \alpha \cdot (|Z_c|+1) \rfloor$-th largest calibration score.
5. Return the prediction region

$$\Gamma^\alpha(x_*;Z_c) = \left\{ y' \in Y \,:\, \Delta(x_*,y') \le Q(1-\alpha;\mathcal{F}_c) \right\}. \tag{8}$$

A natural NCF in regression is the norm of the difference between the real and the predicted target value, i.e., $\Delta(x) = \|y - f(x)\|$. In such a setting, the prediction region becomes

$$\Gamma^\alpha(x_*;Z_c) = [f(x_*) - Q(1-\alpha;\mathcal{F}_c),\, f(x_*) + Q(1-\alpha;\mathcal{F}_c)].$$

Notice that such prediction intervals have the same width ($Q(1-\alpha;\mathcal{F}_c)$) for all inputs.

Predictive uncertainty. A CP-based prediction region provides a set of plausible predictions with statistical guarantees and, as such, also captures the uncertainty about the prediction. The size of the prediction region is determined by the chosen significance level α. Specifically, from Eq. (8) we can see that, for levels $\alpha_1 \ge \alpha_2$, the corresponding prediction regions are such that $\Gamma^{\alpha_1} \subseteq \Gamma^{\alpha_2}$, as a smaller α yields a larger critical value $B_\alpha(\beta_1,\dots,\beta_{N_c})$.

3.2. Conformalized quantile regression

Let us now consider the stochastic setting with a probabilistic function mapping an input $x \in X$ into a distribution over the target space Y.

Quantile regression. The aim of conditional Quantile Regression (QR) is to estimate a given quantile of such a distribution over Y conditional on an input $x \in X$. Let $F(y'|x = x') := \mathbb{P}(y \leq y'|x = x')$ be the conditional distribution function of y given x. Then, the α-th conditional quantile function is defined as

$$q_\alpha(x') := \inf\{y' \in \mathbb{R} \mid F(y'|x = x') \geq \alpha\}. \tag{9}$$

Given an error probability α, we consider lower and upper quantiles w.r.t. $\alpha_{lo} = \alpha/2$ and $\alpha_{hi} = 1 - \alpha/2$, respectively. We define the desired prediction interval as $PI(x) := [q_{\alpha_{lo}}(x), q_{\alpha_{hi}}(x)]$. By construction, this interval satisfies

$$\mathbb{P}\big(y \in PI(x)\big) \geq 1 - \alpha. \tag{10}$$

Since the prediction interval depends on the input, the length of the interval is not fixed in general and changes at different values of x. QR infers such prediction interval from the data. In particular, estimating the quantiles can be expressed as approximating the quantile function, and thus, it can be framed as an optimization problem. In a nutshell, the idea is to propose a parametric function $f(x;\theta)$ as a candidate approximator for $q_\alpha(x)$ and then optimize over its parameters θ so that it closely resembles the quantile function. The optimization problem can be formally expressed as finding the optimal parameters $\hat{\theta}$ such that

$$\hat{\theta} = \arg\min_\theta \left[\frac{1}{n}\sum_i \mathcal{L}_\alpha(Y_i, f(X_i;\theta)) + \mathcal{G}(\theta)\right],$$

where $\mathcal{L}_\alpha$ is the check or pinball loss, defined as

$$\mathcal{L}_\alpha(y, \hat{y}) = \alpha \max(y - \hat{y}, 0) + (1 - \alpha)\max(\hat{y} - y, 0), \tag{11}$$

and $\mathcal{G}$ is an optional regularization term. In this work, we use deep neural networks (NN) as parametric approximators $f(\cdot,\theta)$ of the quantile function and train them using Eq. (11) as loss function. Once trained, the output of $f(x;\hat{\theta})$ is a prediction $\hat{q}_\alpha(x)$ of the conditional α-th quantile.

In general, each quantile requires the training of a different neural network. However, one could also train a single multi-output NN that learns to approximate multiple quantile functions at the same time. The multi-output objective function is obtained by averaging over the respective losses. In this work, we decide to simultaneously learn $q_{\alpha_{lo}}$, $q_{0.5}$ (the median) and $q_{\alpha_{hi}}$, where for a choice of $\alpha \in (0, 0.5)$, $\alpha_{lo} = \alpha$ and $\alpha_{hi} = 1 - \alpha$. The loss then becomes

$$\mathcal{L}_\alpha(y, \hat{y}) = \tfrac{1}{3} \cdot (\mathcal{L}_{\alpha_{lo}}(y, \hat{y}) + \mathcal{L}_{0.5}(y, \hat{y}) + \mathcal{L}_{\alpha_{hi}}(y, \hat{y})).$$

In general, a single NN could simultaneously learn even a larger number of quantiles.

Conformalized quantile regression. The goal of Conformalized Quantile Regression (CQR) is to adjust the QR prediction interval so that it is guaranteed to contain the $(1 - \alpha)$ mass of probability, i.e. to satisfy (10). As for CP, we divide the dataset Z' in a training set Z_t and a calibration set Z_c. We train the QR $f(\cdot;\theta)$ over Z_t and on Z_c we compute the nonconformity scores as

$$E_i := \max\{\hat{q}_{\alpha_{lo}}(x_i) - y_i, y_i - \hat{q}_{\alpha_{hi}}(x_i) \mid (x_i, y_i) \in Z_c\}. \tag{12}$$

In our notation, $\hat{q}_{\alpha_{lo}}(x)$ and $\hat{q}_{\alpha_{hi}}(x)$ denote the two outputs of $f(x;\hat{\theta})$. For this choice of nonconformity score and for an error probability $\alpha \in (0, 1)$, the conformalized prediction interval is defined as

$$CPI_\alpha(x_*; Z_c) = [\hat{q}_{\alpha_{lo}}(x_*) - \tau, \hat{q}_{\alpha_{hi}}(x_*) + \tau],$$

where τ is the $(1 - \alpha)$-quantile of the distribution $\mathcal{F}_c := \frac{1}{|Z_c|+1}\big(\sum_{z_i \in Z_c} \delta_{E_i} + \delta_\infty\big)$ of $\{E_i : z_i \in Z_c\} \cup \{\infty\}$, i.e., $\tau = Q(1 - \alpha; \mathcal{F}_c)$.

In the following, we will abbreviate with *PI* a (non-calibrated) QR prediction interval and with *CPI* a (calibrated) conformalized prediction interval.

Remark 1. This nonconformity function, and thus τ, can be negative, and thus, the conformalized prediction interval can be tighter than the original prediction interval. This means that the CPI can be more efficient than the PI, where the efficiency is the average width of the prediction intervals over a test set. The CPI has guaranteed coverage (the PI does not), meaning that, under the exchangeability assumption between (x, y) and Z_c, it holds that

$$\mathbb{P}(y \in CPI_\alpha(x; Z_c)) \geq 1 - \alpha.$$

Remark 2. In principle, one could use traditional CP for regression (see Section 3.1) to obtain valid prediction intervals for the stochastic case. The main advantage of CQR is that it produces CPIs that are adaptive to heteroscedasticity, i.e., they account for the fact that the variability in the output may be affected by the value of the input. On the contrary, intervals produced by CP for regression have fixed sizes and, hence, do not account for heteroscedasticity. Moreover, the PI of CP for regression would be relative to the conditional mean of the STL robustness instead of the conditional quantile range that we are interested in.

Class-conditional guarantees. One might require prediction intervals that maintain similar error rates across various subsets of the data [27]. Suppose, for instance, that observations (x, y) can be categorized in a discrete and finite set of classes $\{1, \dots, G\}$ corresponding to a finite. We could ask for *class-balanced* guarantees of coverage:

$$\mathbb{P}(y \in CPI_\alpha(x; Z_c) \mid (x, y) \in g) \geq 1 - \alpha \tag{13}$$

for every class $g \in \{1, \dots, G\}$. This solution typically relies on splitting the calibration set Z_c^ϕ in G sub-calibration set and using only the group-specific calibration scores to recalibrate the prediction interval. Within the QPM framework, classes are determined by the Boolean satisfaction value, namely, the sign of R_ϕ. As a result, we require guarantees that are conditional on the sign. Once again, these guarantees hold under the class-specific exchangeability assumption between test and calibration data.

Conditional coverage. We may require prediction intervals that capture the uncertainty of a specific input point. However, the guarantees obtained so far are averaged over the data distribution, and the critical value τ used to "conformalize" the predictions is constant and does not adapt to the predictive uncertainty of a single data point. This kind of adaptivity is typically formalized by the following conditional coverage [28] requirement:

$$\mathbb{P}(y \in CPI_\alpha(x, Z_c) \mid x = x_*) \geq 1 - \alpha, \text{ for all } x_* \in X. \tag{14}$$

That is, for every value of the input x, we seek to return a prediction interval with $1 - \alpha$ coverage over the outputs of x. However, distribution-free conditional coverage is impossible to achieve with a finite sample [28,29], and hence, we adopt a localization-based strategy to achieve approximate conditional validity, as described below. Despite the conditional guarantees, we still require (x, y) and Z_c to be exchangeable.

Quasi-conditional coverage. In conformal inference, each calibration point contributes equally to the critical value τ, without considering its proximity to the test input x_*. To achieve approximate conditional coverage, however, it is essential to give more importance to points that are close to x_* and less importance to those further away. This intuition is at the core of the so-called *localized CP* approach [21,22].

A *localizer* is a function $L : X \times X \to [0, 1]$ such that, for any pair of inputs $x_i, x_j \in X$, $L(x_i, x_j)$ grows as the distance between x_i and $x_j \in X$ shrinks. For example, one could define $L(x_i, x_j) = \exp(-\|x_i - x_j\|)$. In general, L must be defined so that, for every $x_i \in X$, $L(x_i, x_i) = 1$. We denote with $L_* = L(x_*, \cdot) : X \to [0, 1]$ the localizer relative to test point x_*. The localizer is used to reweight the calibration distribution so that calibration points close to x_* have a higher probability, resulting in the distribution

$$\mathcal{F}_* := \frac{1}{|Z_c|+1}\Big(\sum_{z_i \in Z_c} w_*^i \cdot \delta_{E_i} + w_* \cdot \delta_\infty \Big),$$

where $w_*^i := \frac{L_*(x_i)}{L_*(x_*)+\sum_{z_i \in Z_c} L(x_i)}$ and $w_* := \frac{L_*(x_*)}{L_*(x_*)+\sum_{z_i \in Z_c} L(x_i)}$. In general, in the following, w_j^i denotes the weight associated to the calibration score z_j when localized w.r.t. a point x_i.

However, this reweighting breaks the exchangeability assumption, as permutations of calibration and test points no longer hold equal probability, meaning that it may compromise marginal validity. To address this problem and uphold finite sample coverage guarantees, localized CP provides a criterion to select the error level $\tilde{\alpha}$ used to derive the critical value $\tau^* = Q(1-\tilde{\alpha}; \mathcal{F}_*)$ and construct the quasi-conditional prediction region at x_*. For a calibration point $z_j = (x_j, y_j) \in Z_c$, let

$$\mathcal{F}_j^* := \frac{1}{|Z_c|+1}\Big(\sum_{z_i \in Z_c} w_j^i \cdot \delta_{E_i} + w_j^* \cdot \delta_\tau^* \Big)$$

be distribution of $\{E_i : z_i \in Z_c\} \cup \{\tau^*\}$ reweighted around x_j, and let

$$\mathcal{F}_j^0 := \frac{1}{|Z_c|+1}\Big(\sum_{z_i \in Z_c} w_j^i \cdot \delta_{E_i} + w_j^* \cdot \delta_0 \Big)$$

be the reweighted distribution of $\{E_i : z_i \in Z_c\} \cup \{0\}$. Let $\tau_{j1}^* := Q\big(1 - \tilde{\alpha}; \mathcal{F}_j^*\big)$, and $\tau_{j2}^* := Q\big(1 - \tilde{\alpha}; \mathcal{F}_j^0\big)$. The criterion consists of selecting $\tilde{\alpha}$ as the highest value in the range $[0, \alpha]$ such that the following two conditions are met

$$\frac{1}{|Z_c|+1}\left(\sum_{z_i \in Z_c} \mathbb{1}_{\{E_i \leq \tau_{i1}^*\}} \right) \geq 1 - \alpha \text{ and } \frac{1}{|Z_c|+1}\left(\sum_{z_i \in Z_c} \mathbb{1}_{\{E_i \leq \tau_{i2}^*\}} \right) \geq 1 - \alpha, \tag{15}$$

or $\tau^* = \infty$. The resulting prediction interval

$$CPI_\alpha^L(x_*; Z_c) = [\hat{q}_{\alpha_{lo}}(x_*) - \tau^*, \hat{q}_{\alpha_{hi}}(x_*) + \tau^*], \tag{16}$$

can be proven to satisfy marginal validity [21] and provides the following quasi-conditional guarantees: given x_* it holds that

$$\mathbb{P}(y \in CPI_\alpha^L(x; Z_c) \mid x = x_*) \geq 1 - \alpha. \tag{17}$$

The search for $\tilde{\alpha}$ has to be carried out for each test input, potentially impacting computational efficiency due to its dependence on the calibration set size. Specifically, the matrix of weights w_j^i needs to be computed once for every test point, leading to a computational complexity that scales quadratically with the number of calibration points, $n := |Z_c|$. Additionally, verifying the condition requires, for every candidate $\tilde{\alpha}$, the computation of n quantiles from sorted arrays, which is linear in n. Therefore, the overall complexity for each test point is $O(n^2 + n^2 \cdot k)$ where k is the number of candidate values for $\tilde{\alpha}$. An efficient implementation (see [22]) may reduce the cost to $O(n \cdot \log n)$. However, this process typically does not present a significant computational challenge for moderately sized calibration sets, given that the condition is often satisfied long before all k candidate values of $\tilde{\alpha}$ are explored.

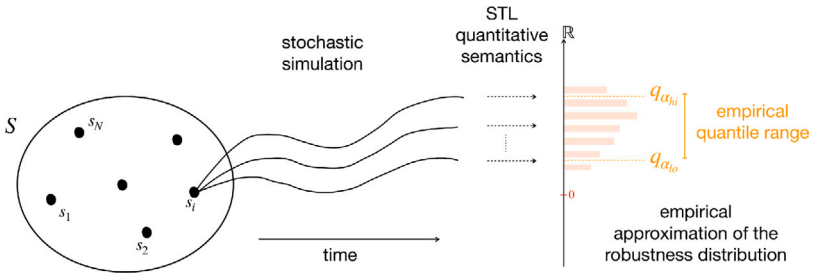


Fig. 2. Diagram illustrating the generation of the dataset.

4. Quantitative predictive monitoring

We present a method to solve Problem 1.[4] For a given a discrete-time stochastic process $\mathbf{S} = \{\mathbf{S}(t,\omega), t \in T\}$ over state space S and a state $s \sim \mathbf{S}(k,\cdot)$ at time $k \in T$, the stochastic evolution (bounded by horizon H) of the system starting at s can be described by the conditional distribution

$$p(\vec{s} \mid \vec{s}(k) = s),$$

where $\vec{s} = (\vec{s}(k), \dots, \vec{s}(k+H)) \in S^H$ is the random trajectory of length H starting at time t_k, $\vec{s}(i) = \mathbf{S}(i,\cdot)$ for any $i \in T$. To ease the notation, in the rest of the presentation, we denote the system's current state with $k = 0$.

The quantitative STL semantic, either spatial or temporal, inherits the stochasticity from the system's dynamics. For an STL property ϕ, we denote with $\mathcal{R}_\phi(s,0)$ the random variable denoting the STL robustness value relative to ϕ of trajectories starting from $s \in S$ at time 0, i.e.,

$$\mathcal{R}_\phi(s,0) \sim p(R_\phi(\vec{s},0) \mid \vec{s}(0) = s).$$

Our conformal solution to the QPM aims at finding, for each state $s \in S$ of the system, a prediction region that covers a certain probability mass of the STL robustness distribution $\mathcal{R}_\phi(s,0)$, see Problem 1. In simpler terms, we aim to monitor how safe the system is relative to the unknown and stochastic future evolution from its current state s. In this way, one can intervene preemptively on the system in order to prevent any failures. However, the distribution of $\mathcal{R}_\phi(s,0)$ is impossible to compute exactly and efficiently. Thus, we resort to Conformalized Quantile Regression (CQR), introduced in Section 3.2, to compute a prediction interval that, for each state s, is guaranteed to cover a desired level $(1-\alpha)$ of the probability mass for the distribution of robustness values $\mathcal{R}_\phi(s,0)$.

In short, our solution consists of four steps, detailed below: *dataset generation*, *QR training*, *NCF scores computation* and *inference*. Note that only the last step, which is by far the quickest, is performed online, the others are performed offline and hence, do not affect runtime performance.

Dataset generation. In this step, we collect data for training the QR function and constructing the calibration set. To do so, we perform Monte-Carlo simulations of the process in order to obtain an empirical approximation of $\mathcal{R}_\phi(s,0)$. In particular, we randomly sample N states $s_1, \dots, s_N \sim \mathbf{S}(\cdot,\cdot)$. Then, for each state s_i, we simulate M trajectories of length H, $\vec{s}_i^1, \dots, \vec{s}_i^M$ where $\vec{s}_i^j$ is a realization of $p(\vec{s} \mid \vec{s}(0) = s_i)$, and compute the robustness value $R_\phi(\vec{s}_i^j, 0)$ of each of these trajectories. We note that $\{R_\phi(\vec{s}_i^j,0)\}_{j=1}^M$ is an empirical approximation of $\mathcal{R}_\phi(s_i,0)$. The dataset is thus defined as

$$Z^\phi = \left\{ \left(s_i, R_\phi(\vec{s}_i^j, 0)\right); i = 1, \dots, N;\ j = 1, \dots, M \right\}. \tag{18}$$

Fig. 2 shows an overview of the steps needed to generate the dataset. The generation of the test set Z^ϕ_{test} is very similar to that of Z^ϕ. The main difference is in that the number of trajectories that we simulate from each state s is much larger than M, i.e., $M_{test} \gg M$. This allows us to obtain a highly accurate empirical approximation of the distribution of $\mathcal{R}_\phi(s)$, which we use as the ground-truth baseline in our experimental evaluation.[5]

QR training and residuals computation. We divide the dataset Z^ϕ into a training set Z^ϕ_t and a calibration set Z^ϕ_c. We then use Z^ϕ_t to train a QR that learns how to map states s into three quantiles

$$f(s;\hat{\theta}) = \{\hat{q}_{\alpha_{lo}}(s), \hat{q}_{0.5}(s), \hat{q}_{\alpha_{hi}}(s)\},$$

where $\alpha_{lo} = \alpha/2$ and $\alpha_{hi} = 1 - \alpha/2$. In order to better reconstruct the shape of the target distribution, we also predict the median quantile, $q_{0.5}(s)$. We then apply CQR, i.e. we compute the residuals of the QR over Z^ϕ_c, as in (12), and find the critical residual value τ as depicted in Fig. 3.

[4] When the context is clear, we will use the term "Quantitative Predictive Monitoring" to refer to both the problem and the solution method.

[5] In the limit of infinite sample size, the empirical approximation approaches the true distribution.

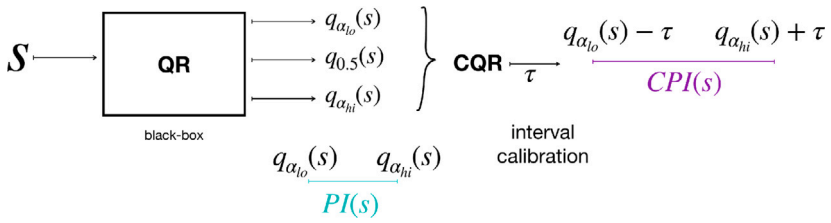


Fig. 3. Overview of the conformal quantitative predictive monitoring technique.

Inference. For a test state s_*, this step involves predicting the relevant quantiles, $\hat{q}_{\alpha_{lo}}(s_*)$ and $\hat{q}_{\alpha_{hi}}(s_*)$, using the QR predictor and correct the resulting interval using the critical residual value τ (see Fig. 3). The calibrated interval returned at inference time becomes

$$CPI_\alpha(s_*; Z_c^\phi) := [\hat{q}_{\alpha_{lo}}(s_*) - \tau, \hat{q}_{\alpha_{hi}}(s_*) + \tau] \subseteq \mathbb{R} \tag{19}$$

and provides the marginal coverage guarantees stated in Eq. (1) with a minor caveat as stated in the remark below.

Remark 3. CP guarantees depend on the random sampling of the calibration set. While the expected coverage across different calibration sets is guaranteed to be at least $1-\alpha$, the actual coverage for any specific calibration set will generally deviate from this value. However, we can control the magnitude of these coverage fluctuations by choosing a sufficiently large calibration set. In particular, the distribution of coverage is known to have the following analytic form [27,28]

$$\mathbb{P}_{\substack{s\sim \mathbf{S}(0,\cdot) \\ \vec{s}\sim p(\vec{s}|\vec{s}(0)=s)}} \left(R_\phi(\vec{s},0) \in CPI_\alpha(s; Z_c^\phi) \right) \sim \text{Beta}(N_c + 1 - l, l),$$

where $l = \lfloor (N_c+1)\cdot\alpha \rfloor$. In our experiments, $\alpha = 0.1$ and the calibration set size goes from 50,000 for low-dimensional case studies to 200,000 for high-dimensional ones, see Section 5. This leads to marginal coverage between 0.896 and 0.903 with 99% confidence in the low-dimensional case and between 0.898 and 0.902 in the high-dimensional case. These fluctuations are arguably minor and can be tightened by introducing more calibration points.

4.1. Conditional validity

System failures are rare to observe, and so, the datasets used to train our predictive monitors are often unbalanced in that they contain more examples of satisfying signals (positive samples) than violating signals (negative samples). In this setting, marginal coverage does not guarantee adequate coverage of both classes of signals. Suppose, for instance, that positive samples have frequency $1-\alpha$, whereas negative samples have frequency α, and prediction intervals always cover positive samples but never negative ones. Then, these prediction intervals meet the $1-\alpha$ nominal coverage in a marginal sense but fail to cover negative instances. Sign-conditional coverage would imply that the prediction intervals cover the output at least $1-\alpha$ of the time in both classes. Similarly, input-conditional coverage is a very strong property that states the probability of the prediction interval needs to be greater than $1-\alpha$ for each input, i.e., for any subset of the data [27]. Localized conformal predictions offer local coverage guarantees.

1. Sign-conditional coverage. The goal here is to provide calibrated prediction intervals where both positive and negative samples are ensured equal coverage. To achieve this, the calibration set Z_c^ϕ is partitioned into its positive $Z_c^{\phi^{(+)}}$ and negative $Z_c^{\phi^{(-)}}$ parts, i.e., in the subsets containing only calibration points with positive and negative outputs, respectively. The prediction interval $PI(s)$ induces two separate calibration distributions of nonconformity scores from which we extract the two $(1-\alpha)$-quantiles, τ^+ and τ^- respectively. The prediction interval $PI(s)$ undergoes a sign-specific calibration obtaining $\overline{CPI}_\alpha^+(s; Z_c^{\phi^{(+)}}) = CPI_\alpha^+(s, Z_c^{\phi^{(+)}}) \cap \mathbb{R}^+$, where $CPI_\alpha^+(s; Z_c^{\phi^{(+)}})$ is defined as in (19), guaranteed to cover $1-\alpha$ of the positive values of $R_\phi(\vec{s},0)$:

$$\mathbb{P}\Big(R_\phi(\vec{s},0) \in \overline{CPI}_\alpha^+(\vec{s}(0); Z_c^{\phi^{(+)}}) \Big| \; R_\phi(\vec{s},0) \geq 0 \Big) \geq 1-\alpha \tag{20}$$

and $\overline{CPI}_\alpha^-(x; Z_c^{\phi^{(-)}}) = CPI_\alpha^-(x; Z_c^{\phi^{(-)}}) \cap \mathbb{R}^-$ is guaranteed to cover $1-\alpha$ of the negative values of $R_\phi(\vec{s},0)$:

$$\mathbb{P}\Big(R_\phi(\vec{s},0) \in \overline{CPI}_\alpha^-(\vec{s}(0); Z_c^{\phi^{(-)}}) \Big| \; R_\phi(\vec{s},0) < 0 \Big) \geq 1-\alpha. \tag{21}$$

The union $CPI_\alpha(s; Z_c^\phi) := \overline{CPI}_\alpha^-(s; Z_c^{\phi^{(-)}}) \cup \overline{CPI}_\alpha^+(s; Z_c^{\phi^{(+)}})$ of the two recalibrated interval is guaranteed to cover $1-\alpha$ of all the values of $R_\phi(\vec{s},0)$ for $\vec{s}(0) = s$:

$$\mathbb{P}\Big(R_\phi(\vec{s},0) \in CPI_\alpha(\vec{s}(0); Z_c^\phi) \Big) \geq 1-\alpha. \tag{22}$$

The proof is provided in Appendix C.

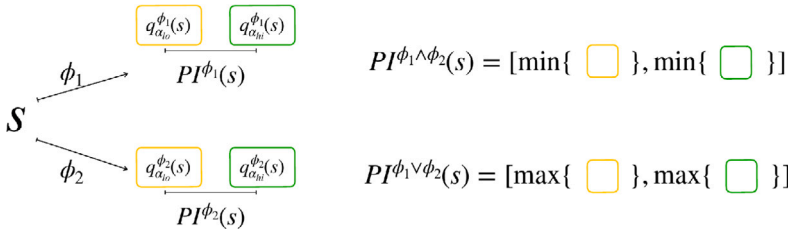


Fig. 4. Overview of the combination of the conformal quantitative predictive monitors for two different properties ϕ_1 and ϕ_2.

2. Localized conformal prediction. We build the localizer function $L_* = L(s_*)$ (introduced in Remark 2, *Quasi-conditional coverage*) as a k-nearest-neighbors function, where $L_*(s) = 1$ if s is among the k nearest neighbors of s_* and 0 otherwise. Each sample in the calibration set is weighted according to L_*, resulting in a state-specific calibration of the prediction interval $CPI^L(s_*)$ as in (16) resulting in quasi-conditional guarantees

$$\mathbb{P}\Big(R_\phi(\vec{s}, 0) \in CPI^L_\alpha(s_*; Z_c^{\phi(L_*)}) \;\Big|\; \vec{s}(0) = s_*\Big) \geq 1 - \alpha. \tag{23}$$

Remark 4. It is straightforward to combine the two approaches and obtain prediction intervals that offer both sign and input conditional guarantees.

4.2. Composing monitors for composite properties

The above-described QPM approach is end-to-end, because it directly predicts the robustness of a fixed property ϕ. On one hand, this is advantageous in that the monitor is tailored and, hence, highly accurate at predicting R_ϕ, but on the other hand, it lacks flexibility. Thus, it becomes natural to ask oneself whether and how two predictive monitors trained for distinct properties ϕ_1 and ϕ_2 can be combined in a compositional manner to construct a monitor for a Boolean combination of the two properties, $\phi_1 \wedge \phi_2$ or $\phi_1 \vee \phi_2$, in such a way that the resulting monitor meets the desired probabilistic guarantees and does not require additional data collection or training.

Here, we propose two solutions to combine monitors for logical conjunction and disjunction. Handling the negation of a property, $\neg\phi$, is rather simple as we can take the interval with opposite signs. Therefore, we cover the basic Boolean logic. In what follows, we will use a superscript ϕ_i with $i = 1, 2$ to distinguish predictors, datasets, critical values, and prediction regions of the two properties. To ease the notation, in the following, we denote the prediction intervals as $CPI^\phi(\cdot) := CPI_\alpha(\cdot; Z_c^\phi)$, i.e., we drop the dependence on α and Z_c^ϕ.

1. Union of intervals. The first, simple solution considers the *union* of the two property-specific prediction intervals. Mathematically,

$$CPI^{\phi_{1,2}} := CPI^{\phi_1} \cup CPI^{\phi_2}.$$

We now prove that $CPI^{\phi_{1,2}}$ has a guaranteed coverage of $(1 - \alpha)$, the same of CPI^{ϕ_1} and CPI^{ϕ_2}, for both $\phi_1 \wedge \phi_2$ and $\phi_1 \vee \phi_2$.

Proposition 1. *Given two properties ϕ_1 and ϕ_2, let CPI^{ϕ_i} be the monitor that maps each state $s \in S$ into a prediction interval for $\mathcal{R}_{\phi_i}(s)$ with coverage $1 - \alpha$. Then, the interval $CPI^{\phi_{12}} := CPI^{\phi_1} \cup CPI^{\phi_2}$ is a valid prediction interval for both $\mathcal{R}_{\phi_1 \wedge \phi_2}$ and $\mathcal{R}_{\phi_1 \vee \phi_2}$, i.e.,*

$$\mathbb{P}\left(R_{\phi_1 \wedge \phi_2}(s) \in CPI^{\phi_{12}}(s)\right) \geq 1 - \alpha \tag{24}$$

and

$$\mathbb{P}\left(R_{\phi_1 \vee \phi_2}(s) \in CPI^{\phi_{12}}(s)\right) \geq 1 - \alpha \tag{25}$$

for every state $s \in S$.

The proof is provided in Appendix B.

2. Calibrated interval arithmetic. The second solution leverages the fact that CP (and CQR) can provide valid intervals on top of any predictor. Hence, we build a monitor for the composite property by first combining the predictors of the individual properties and then re-calibrating such obtained predictors. In particular, for $\phi_1 \wedge \phi_2$ we define for a state $s \in S$ the prediction interval as

$$PI^{\phi_1 \wedge \phi_2}(s) := \left[\min\left(q^{\phi_1}_{\alpha_{lo}}(s), q^{\phi_2}_{\alpha_{lo}}(s)\right), \min\left(q^{\phi_1}_{\alpha_{hi}}(s), q^{\phi_2}_{\alpha_{hi}}(s)\right)\right].$$

See Fig. 4 for a schematic visualization. We then compute the residuals of $PI^{\phi_1 \wedge \phi_2}$ over the calibration set, obtaining the critical value $\tau^{\phi_1 \wedge \phi_2}$ and the conformalized prediction interval

$$CPI^{\phi_1 \wedge \phi_2}(s) := \left[\min\left(q^{\phi_1}_{\alpha_{lo}}(s), q^{\phi_2}_{\alpha_{lo}}(s)\right) - \tau^{\phi_1 \wedge \phi_2},\right.$$

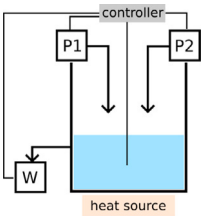


Fig. 5. Scheme of the heated tank system whose components can undergo exponentially distributed failures.

$$\min\left(q^{\phi_1}_{\alpha_{hi}}(s), q^{\phi_2}_{\alpha_{hi}}(s)\right) + \tau^{\phi_1 \wedge \phi_2}\Big],$$

which has a guaranteed coverage of $1-\alpha$.

The procedure is the same for property $\phi_1 \vee \phi_2$, with the only difference that the prediction interval is given by

$$PI^{\phi_1 \vee \phi_2}(s) := \left[\max\left(q^{\phi_1}_{\alpha_{lo}}(s), q^{\phi_2}_{\alpha_{lo}}(s)\right), \max\left(q^{\phi_1}_{\alpha_{hi}}(s), q^{\phi_2}_{\alpha_{hi}}(s)\right)\right].$$

Solution (2) is expected to produce tighter intervals compared to solution (1).[6] On the other hand, solution (2) requires the generation of the respective calibration set, $Z_c^{\phi_1 \wedge \phi_2}$ or $Z_c^{\phi_1 \vee \phi_2}$, which solution (1) does not need.

We remark that such compositionality is defined only for Boolean connectives and not for temporal operators.

5. Experimental results

We experimentally evaluate the proposed QPM over a benchmark of four discrete-time stochastic processes with varying degrees of complexity.

5.1. Case studies

The first two case studies are selected from the stochastic hybrid benchmarks presented in the ARCH-COMP competitions [30]. We then consider two stochastic hybrid systems with modular nature where the complexity can arbitrarily grow, making them suitable to test the scalability of the proposed solution. Finally, we consider a stochastic multi-agent system. See the Supplementary Material (SM) for more details about the case studies presented below.

Automated Anaesthesia Delivery (AAD) [30]: 3-dimensional discrete-time stochastic process with state $s(t) = (q(t), v(t))$ evolving according to linear dynamics with a controller and a Gaussian disturbance. The property to be monitored is: $\phi = F_{[0,t_H]}(v_1 < v_2)$, where $Safe = [1, 6; 0, 10; 0, 10]$, $H = 60$ min, $\Delta t = 20$ s. The controller is defined as follows: $q = 7$ if $v_1 < 3.5$ and $q = 3.5$ if $v_1 \geq 3.5$.

Heated Tank (HT): a stochastic hybrid system composed of a tank containing a liquid whose level is influenced by two inflow pumps P_1 and P_2 and an outflow valve W, all managed by a controller C (see Fig. 5). The liquid should absorb and transport the heat from a heat source. The discrete state space is five-dimensional. The continuous state space is two-dimensional and composed of a state $v(t) = (v_{height}(t), v_{temp}(t))$ representing the liquid height and the liquid temperature. The general model adopted for this heated tank system is that of piece-wise deterministic Markov processes, where the continuous part evolves according to switching differential equations. The switching is managed by the controller and by two exponentially distributed failure events. The property to be monitored is: the liquid level is always in the safe range $Safe = [H_{dryout}, H_{overflow}]$ and the temperature is never too high and it can be expressed by the STL property $\phi = G_{[t_0,t_H]}\big((v_{height} \in Safe) \wedge (v_{temp} \leq T_{overheat})\big)$.

Multi-Room Heating (MRH) [31–33]: discrete-time stochastic hybrid system modeling the temperature evolution in a building with h rooms. In each room is located a heater that switches between *on* and *off* depending on the current temperature in the room. The state of the system is hybrid $S = Q \times V$: the discrete state represents the status of the h heaters, $Q = \{on, off\}^h$, and the continuous state represents the temperatures in the h rooms, $V = \mathbb{R}^h$. The average temperature evolves according to a stochastic difference equation that depends on the rooms' layout (see Fig. 6 (left)). The room-specific requirement to be monitored is: the temperature v_i of room i always stays in a desirable range $[V_i^{lb}, V_i^{ub}]$ and it can be expressed by the STL property $\phi_i = G_{[0,t_H]}(V_i^{lb} \leq v_i \leq V_i^{ub})$.

Gene Regulatory Network (GRN) [34,35]: discrete-time stochastic hybrid system modeling a genetic regulatory network composed of h genes that produce respectively h proteins that repress each other in a cyclic fashion (see Fig. 6 (right)). Each gene could either be *on*, actively producing its protein, or *off*, expression is deactivated when a repressing protein binds to the gene. The state is thus hybrid $S = Q \times V$: the state of the genes is discrete, $Q = \{on, off\}^h$, whereas the proteins count is continuous, $V = \mathbb{N}^h$. Protein counts change deterministically according to a linear differential equation, whereas transitions between modes follow a Markov jump process. The gene-specific requirement to monitor is: the count v_i of protein i stabilize under a certain threshold V_i^{ub} and it can be expressed by the STL property $\phi_i = G_{[t_{H/2},t_H]}(v_i \leq V_i^{ub})$.

[6] In particular, the PI of solution (2) is strictly tighter than the CPI of solution (1).

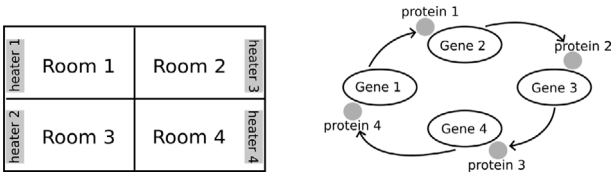


Fig. 6. Schematic visualization of the layout of the two modular case studies: multi-room heating (left) and gene regulatory network (right).

Multi Particle Environment (MPE) [36]: discrete-time stochastic multi-agent reinforcement learning system composed of N_a agents, one of which is an adversarial agent denoted a_e, and M_l landmarks. Each good agent aims to minimize its distance to a predetermined goal landmark, l_{goal}, whilst avoiding the other agents, particularly the adversary. The objective of the adversary is to create as much distance as possible between all agents and the goal (which is not known by the adversary a priori). The state is $s(t) = ((x_i(t), v_i(t))_{i=1}^{N_a}, l_1, \dots, l_{M_l})$. In each step, where x and v denote the 2D position and velocity of each agent, respectively. In each step, each agent i receives a noisy and partial observation of the global state and operates according to deep neural network policies trained with Deep Q-Networks [37] where the possible actions are: *do nothing, left, right, down, up*. The system dynamics are described by second-order motion laws with a degree of actuation noise applied. The (global) property to be monitored is one of collision avoidance: all agents (including the adversary) maintain a safe distance d_s throughout the trajectory. It is described by the STL property: $\phi = G_{[0,H]}\left(\wedge_{i \neq j} \|pos_i - pos_j\| > d_s\right)$.

5.2. Evaluation metrics

We want our method to be capable of working at runtime in safety-critical applications, which translates into the need for high reliability and high computational efficiency in producing the predictive intervals and calibrating them. We emphasize that the time required to train the quantile regressor and to compute the calibration score τ does not affect its runtime efficiency, as it is performed in advance (offline) only once. The localized version is an exception as the computation of τ^* is state-dependent, and thus, it has to be performed online. Once trained, the time needed to have a prediction interval for the current state is simply the time needed to evaluate a neural network with a rather simple architecture. This time is almost negligible (in the order of microseconds on GPU). In addition, we do not want an over-conservative predictor, as an unnecessarily large interval would reduce the effectiveness of our QPM. Keeping that in mind, we introduce some relevant metrics to evaluate the performances of our QPM.

Accuracy. For each test state s_i, we derive the empirical quantiles $q^i_{\alpha/2}$ and $q^i_{1-\alpha/2}$ from samples $R_\phi(\vec{s}_i^{\,1}, 0), \dots, R_\phi(\vec{s}_i^{\,M}, 0)$, where $\vec{s}_i^{\,j}(0) = s_i$, and use these to label the state as *safe*, *unsafe*, or *risky*: if the trajectories starting from s_i satisfy ϕ with probability above $1 - \alpha/2$ (i.e., $q^i_{\alpha/2} > 0$), then we label s_i as $\ell_i = +1$ (safe); if the probability of satisfaction rather than violation here is below $\alpha/2$ (i.e., $q^i_{1-\alpha/2} < 0$), then we label s_i as $\ell_i = -1$ (unsafe); otherwise, we use label $\ell_i = 0$ (risky). We then compare the sign of the predicted interval with the sign ℓ of the empirical quantile range over the test set. In simpler words, the latter can be interpreted as a classification accuracy over the sign of the prediction intervals and the dataset labels ℓ. If both the real and the estimated interval are risky, i.e. if they both straddle zero, then the prediction is correct. If the signs of the two intervals are opposite, the prediction is *wrong*. On the other hand, if the state is either safe or unsafe and it is predicted as risky, we label the prediction as *uncertain*. Moreover, we analyze the percentage of *false positive* (FP) errors, the most dangerous ones as they compromise the safety of the system. A false positive occurs when the state is either unsafe or risky but is predicted to be safe.

Coverage and efficiency. We experimentally check that the guaranteed validity of CQR is empirically met in the test evaluation. The efficiency represents the average width of the prediction intervals over the test set. In general, the larger the prediction interval, the more conservative the CQR predictions. If the prediction intervals over the robustness values are always very large, we have little information about the satisfaction of the property ϕ. However, the predictive efficiency must be compared with the width of the empirical quantile range (EQR), i.e. the interval that contains $(1 - \alpha)$ of the simulated robustness values. We can thus measure the conservativeness as the difference in width between the predicted efficiency and the EQR width. We also compare the coverage and the efficiency of QR against those of CQR.

Overall, we aim to reach high accuracy in the reconstruction of the robustness distribution, which translates into a high percentage of correct predictions, i.e. intervals with agreeing signs. However, it is even more important to reduce the number of wrong predictions, especially false positives, as they can compromise the system's safety. We rather have more uncertain predictions but avoid erroneous ones. We check if the statistical guarantees about the coverage are met empirically by measuring the robustness of how many trajectories fall inside the predicted interval. Let us remark that the coverage and the percentage of correct predictions denote two different quantities and that the efficiency is not a percentage but the average width of the prediction interval measuring how conservative the predictor is compared to the empirical efficiency (EQR width).

Conditional coverage. To evaluate conditional coverage, we measure, for each state $s_i \in Z^{\phi}_{test}$, the empirical coverage $Cov(s_i)$, i.e. how many of the robustness values $R_{\phi}(\vec{s}_i^{\,1}, 0), \dots, R_{\phi}(\vec{s}_i^{\,M_{test}}, 0)$ fall inside the calibrated prediction interval $CPI_{\alpha}(s_i; Z^{\phi}_c)$:

$$Cov(s_i) := \frac{1}{M_{test}} \sum_{j=1}^{M_{test}} \mathbb{1}_{\{R_{\phi}(\vec{s}_i^{\,j}, 0) \in CPI_{\alpha}(s_i; Z^{\phi}_c)\}}. \tag{26}$$

We could then examine the distribution of these coverage rates over the test set and how this distribution changes with different types of guarantees, such as marginal, sign-conditional, and quasi-conditional. The anticipated behavior can be described as follows: in the case of marginal guarantees, the distribution of input-conditional coverage is expected to have a mean value around $1 - \alpha$ with a potentially large variance. For quasi-input conditional guarantees, this distribution should also average around $1 - \alpha$, but with substantially lower variance. Similarly, one can assess the coverage rate distributions for positive and negative samples independently. If the average values of these distributions deviate from $1 - \alpha$ it indicates a lack of sign-conditional coverage. In the class-balanced approach, the average coverage rate for both categories should approximate $1 - \alpha$.

5.3. Experiments

The workflow can be divided into steps: (1) define the model of the stochastic process, (2) generate the synthetic datasets Z^{ϕ} (simulate and compute STL robustness), (3) train the QR, (4) compute the calibration score τ (obtaining CQR prediction intervals), (5) evaluate both the QR and the CQR on a test set Z^{ϕ}_{test}.

Experimental settings. The entire pipeline is implemented in Python. The `pcheck` library[7] is used to quantify the spatial robustness of STL formula for a specific trajectory. Time robustness has been computed from the STL signals provided by the `pcheck` library. The neural networks of the quantile regressors are trained with PyTorch [38]. The experiments were conducted on a virtual machine with a 32-core Processor, 64 GB of RAM and an NVIDIA A100 GPU with 20 GB VRAM, and 8 VCPU. Our implementation is available at https://github.com/francescacairoli/Conformal_QPM.git.

Datasets. For every model, we generate a dataset Z^{ϕ} with a size that increases with the dimensionality of the model. In general, the chosen datasets are not too large. In particular, being n the model's dimension, we sample $N_{train} = n \cdot 1000$ states for the training set, $N_{cal} = n \cdot 500$ states for the calibration set and $N_{test} = n \cdot 100$ states for the test set. From each state, we simulate $M = 50$ trajectories for the training and for the calibration set and $M_{test} = 500$ trajectories for the test set. Data are scaled to the interval $[-1, 1]$ to avoid sensitivity to different scales.

Training details. In all the experiments, we use the same neural network architecture. The network comprises 3 hidden layers with 20 neurons each. We use the Adam optimizer with a learning rate of 0.0005 and a dropout rate of 0.1 applied on each layer. We use a LeakyReLU activation function with slope 0.01 and train the network for 500 epochs over batches of size 512.

Offline costs. The offline overhead consists of the time needed to generate the dataset and the time needed to train the neural network. The cost of the latter depends on the chosen architecture and on the dataset size. In our experiments, it ranges from 30 minutes to 6 hours for larger models. On the other hand, the time needed to generate the dataset depends on the time needed to simulate the trajectories and label them. These quantities are influenced by many factors: the chosen horizon H, the complexity of the STL property, and the complexity of the stochastic dynamics. For instance, the time needed to generate a dataset of $2000 \times 50 = 100K$ observations for the $GRN2$ model is double (1 hour) compared to the $MRH2$ model (30 minutes), even if they have the same dimensionality. The dataset generation time explodes easily around 20 hour for $GRN6$ and $MRH8$. For the MPE case study, the data generation time is approximately 15 h for 3000 initial states and 200 trajectories per each state. The MPE datasets in particular was generated using 100 threads on a machine with 136 cores, 512GB of RAM.

Performance evaluation. We choose in all the experiments $\alpha = 0.1$, so that $\alpha_{lo} = 0.05$ and $\alpha_{hi} = 0.95$. The results are presented with the following structure. For each case study and for each property we train the QR and then recalibrate the predicted interval by means of CQR. For each case study, we also analyze the sign-conditional and the quasi-input conditional coverage. For the modular case studies, i.e. MRH and GRN, we consider properties regarding each individual component, i.e. for each room and for each gene, respectively. To analyze how the performance scales with the dimensionality of the underlying system, we test our QPM over different configurations of the modular system. In particular, we consider a multi-room heating (MRH) system with respectively 2, 4, and 8 rooms and a gene regulatory net (GRN) system with 2, 4, and 6 genes.

Results. Fig. 7, 9, 11, 13, 15 and 17 (and D.20, D.22) show a comparison of the predicted intervals, both PI (non-calibrated, cyan) and CPI (calibrated, blue), over 40 states randomly sampled from the test set and compare them with the empirical samples (orange) collected in the test set. Intervals above zero mean that the state is safe, intervals below mean that the state is unsafe, whereas intervals that straddle zero mean that the state is risky. Empirical samples that lie outside the EQR are denoted in a lighter orange. Results are compared against the prediction intervals provided by the sign-balanced interval (CB-CPI, top figure) and the interval obtained using a kNN-localizer with $k = 10$ (kNN-CPI, bottom figure) — both denoted with a purple interval.

[7] https://github.com/simonesilvetti/pcheck

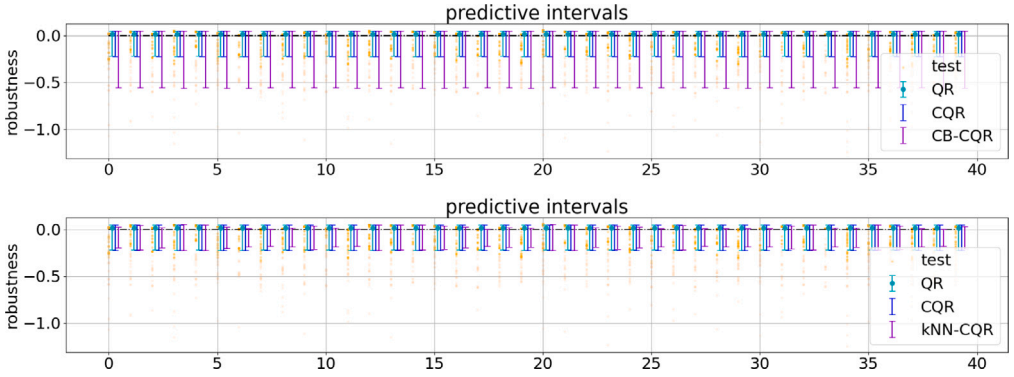


Fig. 7. Visualization of prediction intervals – cyan for QR and blue for CQR – over 40 randomly selected states of the HT model. Purple intervals denote either the sign-balanced (top) or the localized (bottom) prediction intervals.

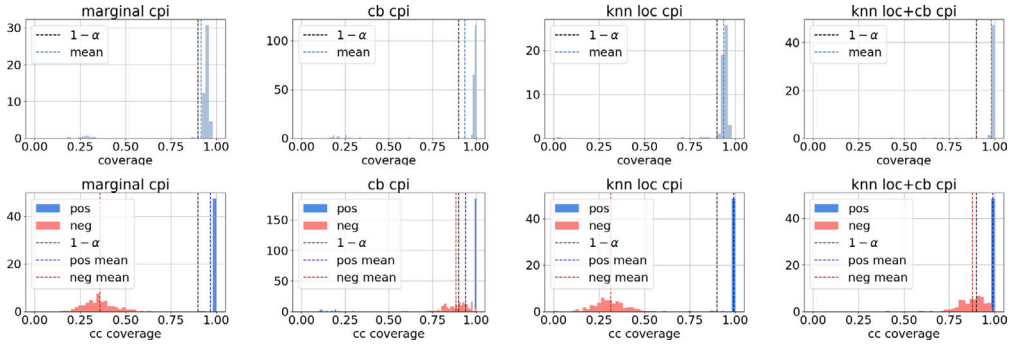


Fig. 8. Visualization of coverage distributions, either marginal (top line) or sign-conditional (bottom line), for the HT model.

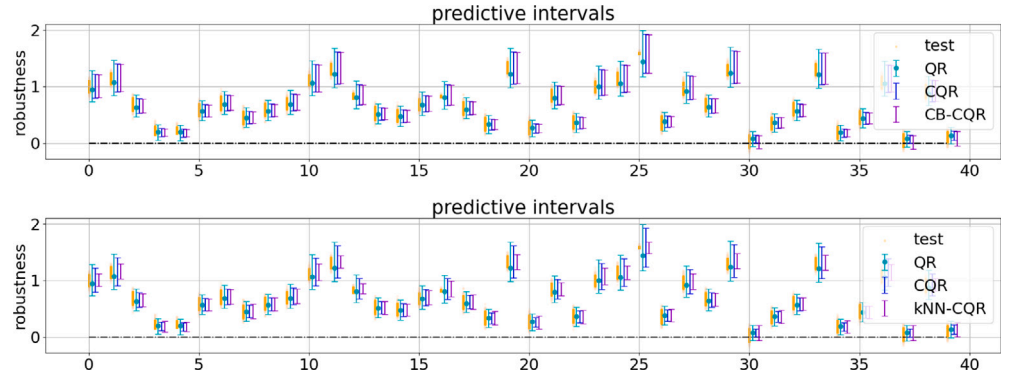


Fig. 9. Visualization of prediction intervals – cyan for QR and blue for CQR – over 40 randomly selected states of the AAD model. Purple intervals denote either the sign-balanced (top) or the localized (bottom) prediction intervals.

Fig. 8, 10, 12, 14, 16 and 18 (and D.21, D.23) show a comparison of the distribution of coverage rates over the test set (as described in (26)). Histograms in the top row show the empirical distribution of coverage rates for the CPI (first column), the CB-CPI (second column), the kNN-CPI (third column) and the CB-kNN-CPI (fourth column). Similarly, histograms in the bottom row show the empirical distribution of coverage for positive (blue) and negative (red) samples. Dashed vertical lines show the average coverage, which is compared against the nominal $1-\alpha$ value in black. We notice how CPI provides highly unbalanced sign-conditional coverage, whereas CB-CPI significantly alleviates the problem. On the other hand, we notice how the localized prediction interval kNN-CPI offers coverage distributions more centered around the nominal value $1-\alpha$, drastically reducing the number of outliers. However, as for CPI, the sign-conditional coverages of kNN-CPI are unbalanced. To get the best out of the two approaches, we combine them (as in Remark 4) and obtain a sign and input conditional prediction intervals, whose results are shown in the last column (CB-kNN-CPI).

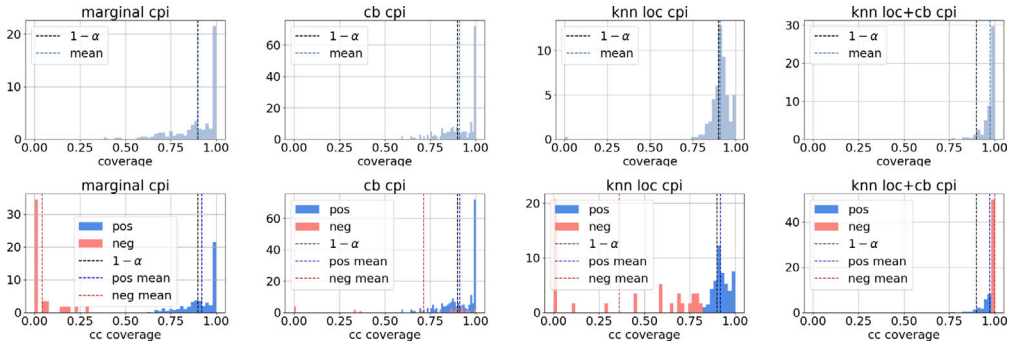


Fig. 10. Visualization of coverage distributions, either marginal (top line) or sign-conditional (bottom line), for the AAD model.

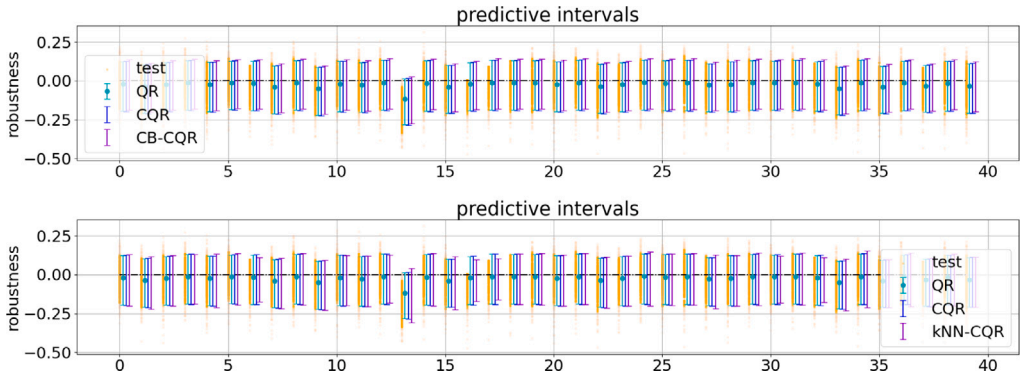


Fig. 11. Visualization of prediction intervals – cyan for QR and blue for CQR – over 40 randomly selected states of the MRH2 model (property over room 1). Purple intervals denote either the sign-balanced (top) or the localized (bottom) prediction intervals.

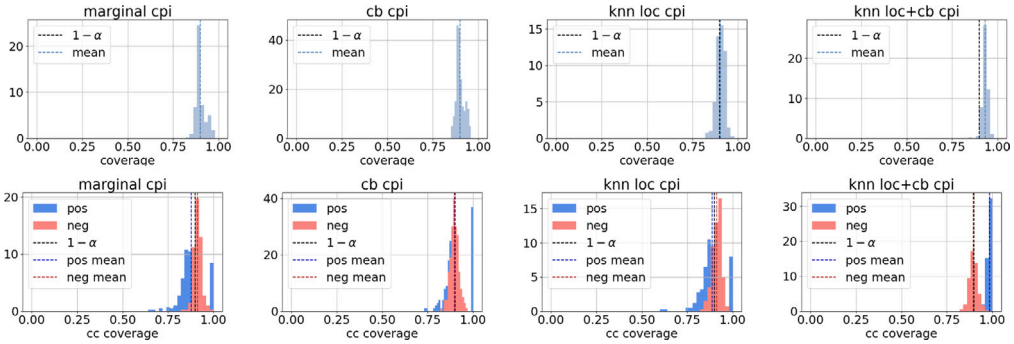


Fig. 12. Visualization of coverage distributions, either marginal (top line) or sign-conditional (bottom line), for the MRH2 model (property over room 1).

Tables 1–5 (and E.6–E.8c) provide a detailed summary of the performance and of the generalization capabilities of our QPM method. In particular, they compare the accuracy (rate of correct, uncertain, wrong and false positive predictions), the coverage and the efficiency of our solution, in all its variations, over all the different case studies.

We observe that the percentage of wrong and FP predictions is extremely low in all the experimental configurations (always lower than 5%). The percentage of uncertain prediction intervals is, in general, very low (lower than 10%). Some of the most complex case studies make an exception and present a high percentage of uncertain predictions (around 30% of the test predictions). This happens because most of the empirical quantile ranges in the test set have one of the two extremes very close to zero. Therefore, even if numerical results may seem to suggest that we have a poor reconstruction, in reality, we are successfully capturing the fact that most of the states are on the verge of violating the property. Similarly, the sign-balanced solutions often present a higher percentage of uncertain predictions. This happens in case studies with highly unbalanced datasets, where the sign-conditional guarantees come at the cost of more conservative prediction intervals, as reflected by the higher values of efficiency.

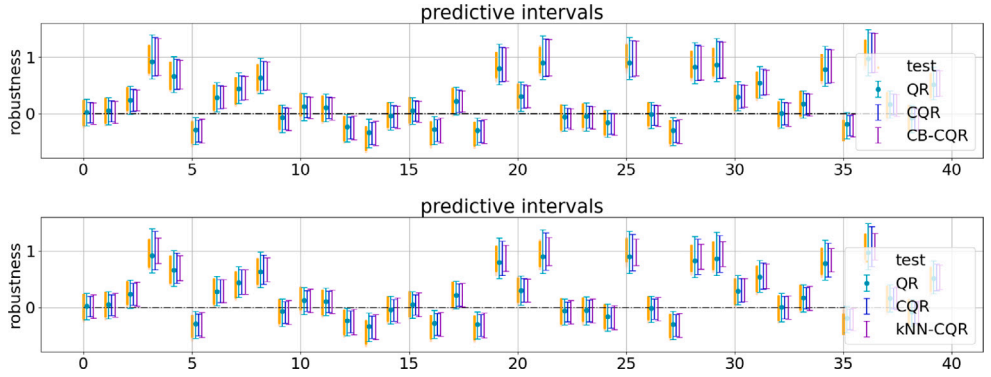


Fig. 13. Visualization of prediction intervals – cyan for QR and blue for CQR – over 40 randomly selected states of the GRN2 model (property over gene 2). Purple intervals denote either the sign-balanced (top) or the localized (bottom) prediction intervals.

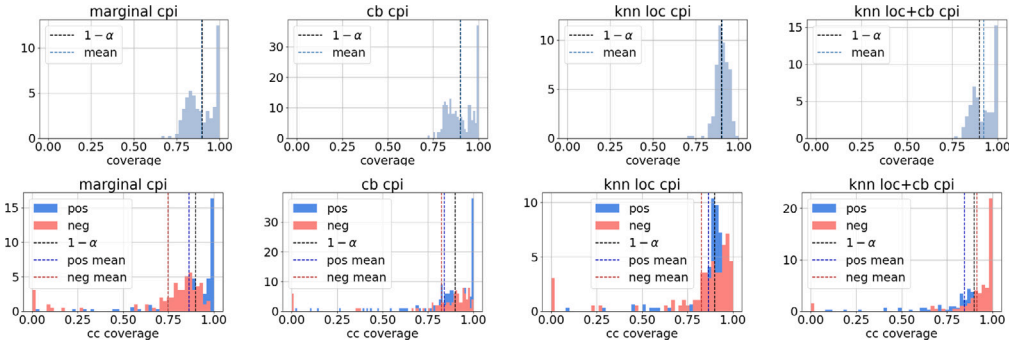


Fig. 14. Visualization of coverage distributions, either marginal (top line) or sign-conditional (bottom line), for the GRN2 model (property over gene 2).

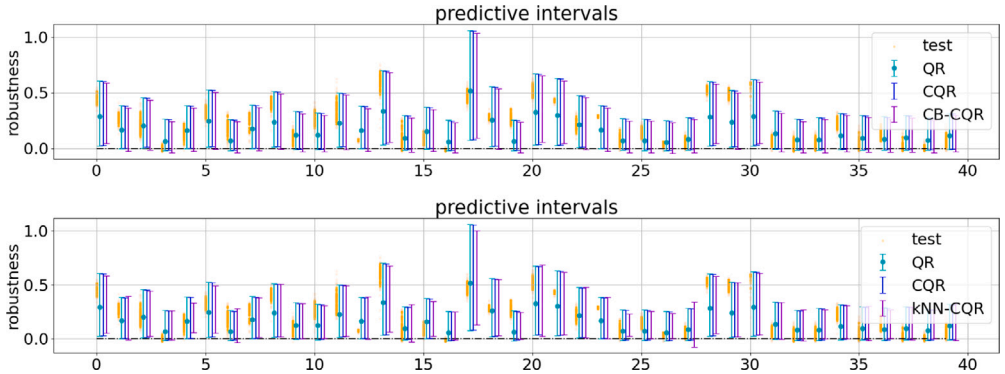


Fig. 15. Visualization of prediction intervals – cyan for QR and blue for CQR – over 40 randomly selected states of the multi-agent model MPE2 (2 agents and 1 adversary) with **space robusteness**. Purple intervals denote either the sign-balanced (top) or the localized (bottom) prediction intervals.

From the same tables, we can also experimentally observe that CQR always meets the statistical guarantees: when QR is over-conservative, CQR refines the predicted interval (negative τ). When QR is over-confident, CQR enlarges the predicted interval (positive τ). The efficiencies of the PI and CPI predicted intervals are compared to the width of the empirical quantile range computed over the test set (*EQR width* column). Both QR and CQR produces intervals that are on average 20% more conservative than the EQR. However, CQR tends to be tighter in easier case studies (QR is on average 23% more conservative than EQR against the 13% of CQR), and instead, it tends to be more conservative on more complex case studies (QR on average is 17% more conservative than EQR against the 27% of CQR). Recall that the efficiency is the width of the interval, and so lower values are, in general, more desirable. We also notice that the input-conditional recalibration improves the efficiency (kNN-CQR is, on average, only 10% more conservative than EQR over easier case studies), whereas sign-conditional recalibration results in a significant loss in efficiency (CB-CQR is on average almost 30% more conservative than EQR over easier case studies). See Fig. 9 for an intuitive visualization of such

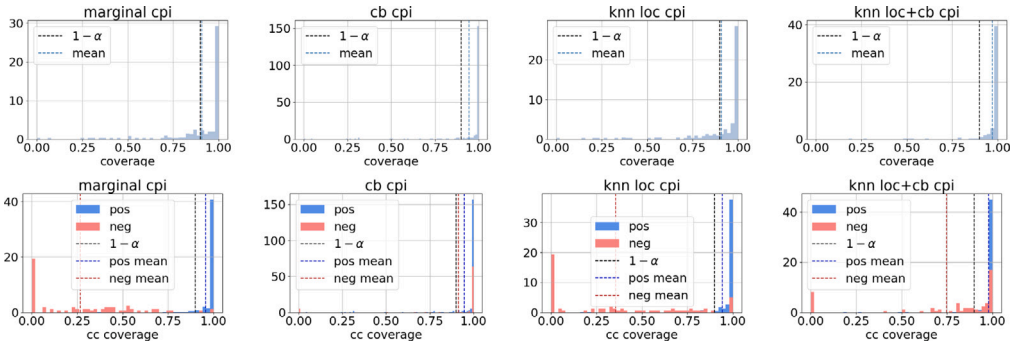


Fig. 16. Visualization of coverage distributions, either marginal (top line) or sign-conditional (bottom line), for the multi-agent system with 2 agents and 1 adversary (MPE2)with **space robusteness**.

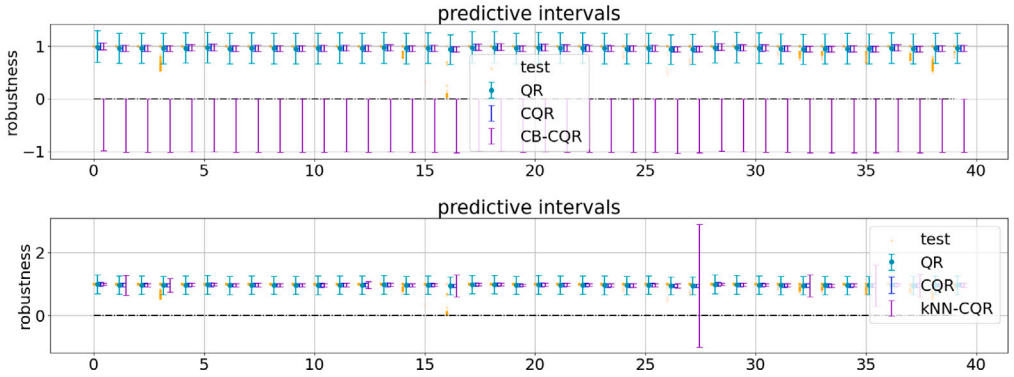


Fig. 17. Visualization of prediction intervals – cyan for QR and blue for CQR – over 40 randomly selected states of the multi-agent model MPE2 (2 agents and 1 adversary) with **time robusteness**. Purple intervals denote either the sign-balanced (top) or the localized (bottom) prediction intervals.

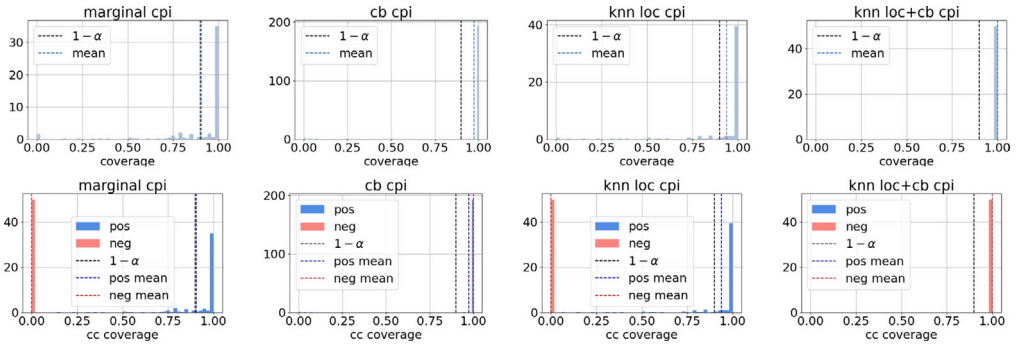


Fig. 18. Visualization of coverage distributions, either marginal (top line) or sign-conditional (bottom line), for the multi-agent system with 2 agents and 1 adversary (MPE2) with **time robusteness**.

behavior. Localization enables tighter prediction intervals in regions presenting good reconstruction, resorting to broader intervals only in poorly reconstructed regions. Hence, it is possible to observe an improved efficiency w.r.t. CQR, especially over simple case studies. On the other hand, the over-conservative behavior of sign-balanced prediction intervals is foreseeable, as most datasets consist of a limited amount of negative samples that often lie outside the prediction interval. This leads to a large recalibration score for the lower bound, τ^-, as observed in Fig. 7, and thus in a high-efficiency value.

The modular experiments, $MRH2, MRH4, MRH8$, $GRN2, GRN4$ and $GRN6$, allow us to analyze the compositionality of QPM over the conjunction of properties (each property is room- or gene-specific). Due to space limitations, we focus only on the conjunction ($\wedge$ operator). The same analysis can be done for the disjunction ($\vee$ operator). In Tables 3–5 (and E.6–E.8c in Appendix E), we compare the performances of four different approaches: QR and CQR trained over the training set $Z_t^{\phi_1 \wedge \phi_2}$ and calibrated over $Z_c^{\phi_1 \wedge \phi_2}$, the minimum w.r.t. interval arithmetic (MIN in tables) of the PI intervals calibrated once again over $Z_c^{\phi_1 \wedge \phi_2}$ (solution (2) presented in Section 4.2), and finally the union (UNION in tables) of the CPI intervals (solution (1) presented in Section 4.2). Results

Table 1
Correct, uncertain, wrong and coverage represent percentage values (%). *EQR width* denotes the average width of the empirical quantile range over the test set.

AAD	Correct	Uncertain	Wrong	FP	Coverage	Efficiency	EQR width
QR	97.50	2.50	0.00	0.00	98.19	0.4484	0.1942
CQR	96.50	0.00	3.50	3.50	89.79	0.3199	
CB-CQR	93.50	6.50	0.00	0.00	91.08	0.3185	
kNN-CQR	98.50	1.00	0.50	0.50	90.88	0.2413	
HT	**correct**	**uncertain**	**wrong**	FP	**coverage**	**efficiency**	EQR width
QR	99.00	1.00	0.00	0.00	89.79	0.2682	0.2583
CQR	99.00	1.00	0.00	0.00	91.61	0.2732	
CB-CQR	99.00	1.00	0.00	0.00	93.47	0.6067	
kNN-CQR	99.00	1.00	0.00	0.00	93.53	0.2292	

Table 2
Correct, uncertain, wrong and coverage represent percentage values (%). *EQR width* denotes the average width of the empirical quantile range over the test set.

MPE2-space	Correct	Uncertain	Wrong	FP	Coverage	Efficiency	EQR width
QR	62.50	36.00	1.50	1.50	93.16	0.4163	0.1299
CQR	69.00	27.00	4.00	4.00	90.80	0.4059	
CB-CQR	39.50	60.00	0.50	0.50	94.44	0.4082	
kNN-CQR	68.50	26.00	5.50	5.50	89.59	0.3929	
MPE2-time	**correct**	**uncertain**	**wrong**	FP	**coverage**	**efficiency**	EQR width
QR	100.00	0.00	0.00	0.00	96.07	0.5809	0.0959
CQR	100.00	0.00	0.00	0.00	90.64	0.1161	
CB-CQR	0.00	100.00	0.00	0.00	97.40	2.0333	
kNN-CQR	97.00	3.00	0.00	0.00	93.73	0.3600	

confirm our intuition that the UNION approach produces over-conservative predictions, i.e. intervals with high coverage and high efficiency. The MIN approach, on the other hand, provides better coverage (less conservative) and better efficiency (comparable to that of CQR). In particular, the CQR approach produces intervals that are on average 14% more conservative than the EQR, MIN's intervals are on average 4% more conservative than the EQR, and UNION's intervals are on average 14% more conservative, i.e. the intervals are wider. The same compositional approach is applied to sign-balanced and localized prediction intervals. Results resemble the unconditional ones. The same analysis is also performed for prediction intervals calibrated to obtain sign and input conditional guarantees and their union or their minimum. The results, shown in the same tables as before, are comparable.

Multi-agency and time robustness. The experimental setup for the multi-agent scenarios is structured as follows: the first scenario, referred to as MPE2, involves three agents, including one adversary, while the second scenario, MPE3, comprises four agents with one acting as the adversary. The QPM operates under conditions of partial observability, basing its predictions of future robustness solely on the current positions of each agent, without considering their velocities or the positions of landmarks. Therefore, the input dimension is $2 \cdot N_a$, with N_a denoting the total number of agents. The QPM predicts either the spatial STL robustness (Figs. 15–16 and Figs. D.20–D.21 in Appendix D) or the temporal STL robustness (Figs. 17–18 and Figs. D.22–D.23 in Appendix D). Our approach considers stochastic processes defined over a discrete time, with time robustness quantifying how robust a trajectory, $\vec{s}$, is w.r.t. temporal shifts. This essentially measures the number of time steps required to alter the Boolean satisfaction value of $\vec{s}$, making time robustness and its quantiles discrete functions. Although designing a quantile regressor with a discrete output space could enhance the learning, we choose to apply the same QPM architecture used for spatial robustness to the temporal context for consistency. Time robustness values were normalized to the range $[-1, 1]$. Results related to time robustness (Fig. 17 18 and D.22 D.23 and Tables 2 and E.6 in Appendix E) illustrate that both marginal and localized conformal recalibration significantly improve the efficiency of the quantile regressor. However, the pronounced scarcity of samples with negative time robustness results in excessively conservative prediction intervals under sign-balanced guarantees. More specifically, Fig. 17 and D.22 show how the two sign-specific calibration intervals, $\overline{CPI}^{+}$ and $\overline{CPI}^{-}$, are disjoint, with $\overline{CPI}^{-}$ compensating for the coverage of under-represented negative samples. Although these intervals might not have immediate practical applications, they are invaluable for highlighting significant imbalances in data representation across different classes.

Sequential data. All the results presented so far consider a dataset of states independently sampled from a distribution S. By doing so, we aim to quantify the generalization capabilities of our QPM. However, we are also interested in applying our QPM at runtime to systems that are evolving in time. States will thus have a temporal dependency, meaning that we lose the exchangeability requirement behind the theoretical validity of the CPI. Fig. 19 shows the performance of QPM applied to such sequential data. In particular, we randomly sampled a trajectory of length 20 and applied the QPM to each state in the trajectory. We observe that QPM well captures the evolution of the robustness values but it tends to be either slightly over-confident or slightly over-conservative. This behavior is typical of CP when the data-generating distribution at test time differs from the one used during the training and the calibration phase.

Table 3
Multi room heating results with two rooms on the top and Gene Regulatory Network results with two genes on the bottom. Correct, uncertain, wrong and coverage represent percentage values (%). *EQR width* denotes the average width of the empirical quantile range over the test set.

MRH2	Correct	Uncertain	Wrong	FP	Coverage	Efficiency	EQR width
Room 1							
QR	99.50	0.00	0.50	0.00	89.41	0.3154	0.3167
CQR	99.50	0.00	0.50	0.00	89.95	0.3208	
CB-CQR	99.50	0.00	0.50	0.00	89.96	0.3183	
kNN-CQR	99.50	0.00	0.50	0.00	90.34	0.3227	
Room 2							
QR	91.50	8.00	0.50	0.50	95.82	0.2585	0.1644
CQR	94.00	0.50	5.50	5.50	90.08	0.2020	
CB-CQR	77.50	2.30	0.00	0.00	92.54	0.2260	
kNN-CQR	98.00	0.50	1.50	1.50	89.23	0.1911	
Room 1 $\wedge$ Room 2							
QR	99.50	0.50	0.00	0.0	89.77	0.3078	0.3026
CQR	99.50	0.50	0.00	0.0	89.77	0.3122	
CB-CQR	99.50	0.50	0.00	0.0	90.12	0.3096	
kNN-CQR	99.50	0.50	0.00	0.0	90.67	0.3152	
MIN	99.50	0.50	0.00	0.0	90.27	0.3117	
UNION	–	–	–	–	92.25	0.2968	
CB-MIN	99.50	0.50	0.00	0.0	90.08	0.3093	
CB-UNION	–	–	–	–	91.85	0.4786	
kNN-MIN	99.50	0.50	0.00	0.0	90.64	0.3140	
kNN-UNION	–	–	–	–	92.47	0.2864	

GRN2	Correct	Uncertain	Wrong	FP	Coverage	Efficiency	EQR width
Gene 1							
QR	94.50	5.00	0.50	0.50	94.14	0.8516	0.7978
CQR	94.50	3.00	2.50	2.50	89.32	0.8045	
CB-CQR	95.50	3.00	1.50	1.50	89.44	0.8071	
kNN-CQR	95.50	2.00	2.50	2.50	89.81	0.8084	
Gene 2							
QR	94.50	5.50	0.00	0.00	98.12	0.5665	0.4229
CQR	97.00	1.00	2.00	0.00	89.48	0.4602	
CB-CQR	96.50	1.00	2.50	0.0	90.03	0.4602	
kNN-CQR	97.50	1.50	1.00	0.00	90.26	0.4316	
Gene 1 $\wedge$ Gene 2							
QR	85.00	14.50	0.50	0.50	93.44	0.6283	0.5267
CQR	88.50	10.00	1.50	0.50	93.44	0.5642	
CB-CQR	88.00	11.00	1.00	0.50	89.25	0.5632	
kNN-CQR	91.00	8.00	1.00	0.50	90.12	0.5626	
MIN	88.00	9.50	2.50	2.00	89.69	0.6008	
UNION	–	–	–	–	92.10	0.5483	
CB-MIN	89.00	8.50	2.50	2.00	89.97	0.5952	
CB-UNION	–	–	–	–	93.32	1.0044	
kNN-MIN	88.00	9.50	2.50	2.50	89.94	0.5946	
kNN-UNION	–	–	–	–	93.53	0.5418	

5.4. Discussion

In summary, we can see that the proposed QPM is effective in tackling the scalability issues surrounding the predictive problem for stochastic processes. In this regard, we test such scalability over a complex multi-agent scenario, increasing the number of agents in the system. The solution is inevitably approximate, as an exact solution is unfeasible even for extremely simple stochastic processes. In this sense, the statistical guarantees our method provides play a crucial role when dealing with safety-critical applications. We observe that conditional validity generally ensures more reliable guarantees of coverage. In many cases, sign-conditional guarantees result in over-conservative prediction intervals as a means to compensate for the scarcity of negative observation. On the other hand, input-conditional guarantees often boost efficiency by allocating uncertainty more strategically. We show that QPM performs reasonably well also at runtime with sequentially generated data. We also show how the proposed methodology is not strictly related to the chosen measure of robustness, and one could interchangeably decide to focus on a notion of either space or time robustness and apply the same QPM methodology.

Table 4
Multi room heating results with four rooms (MRH4). Correct, uncertain, wrong and coverage represent percentage values (%). *EQR width* denotes the average width of the empirical quantile range over the test set.

MRH4	Correct	Uncertain	Wrong	FP	Coverage	Efficiency	EQR width
Room 1							
QR	100.00	0.00	0.00	0.00	90.72	0.2962	0.2782
CQR	100.00	0.00	0.00	0.00	90.27	0.2929	
CB-CQR	100.00	0.00	0.00	0.00	90.24	0.3065	
kNN-CQR	100.00	0.00	0.00	0.00	90.60	0.2949	
Room 2							
QR	95.25	2.75	2.00	2.00	94.05	0.2151	0.1540
CQR	96.00	0.25	3.75	3.75	90.66	0.1830	
CB-CQR	72.75	27.25	0.00	0.00	92.23	0.2214	
kNN-CQR	96.75	0.00	3.25	3.25	87.86	0.1795	
Room 3							
QR	96.50	1.50	2.00	2.00	93.71	0.2176	0.1704
CQR	96.75	0.00	3.25	3.25	89.46	0.1893	
CB-CQR	82.00	18.00	0.00	0.00	92.18	0.2113	
kNN-CQR	97.50	0.00	2.50	2.00	87.96	0.1885	
Room 4							
QR	100.00	0.00	0.00	0.00	92.08	0.2987	0.2766
CQR	100.00	0.00	0.00	0.00	89.75	0.2788	
CB-CQR	1.00	99.00	0.00	0.00	95.49	0.4556	
kNN-CQR	100.00	0.00	0.00	0.00	89.84	0.2800	
Room 1 $\wedge$ Room 2							
QR	100.00	0.00	0.00	0.00	90.91	0.2832	0.2547
CQR	100.00	0.00	0.00	0.00	90.91	0.2795	
CB-CQR	100.00	0.00	0.00	0.00	90.79	0.2945	
kNN-CQR	100.00	0.00	0.00	0.00	90.81	0.2824	
MIN	100.00	0.00	0.00	0.00	90.41	0.2747	
UNION	–	–	–	–	93.11	0.2692	
CB-MIN	100.00	0.00	0.00	0.00	90.63	0.2875	
CB-UNION	–	–	–	–	94.03	0.4520	
kNN-MIN	100.00	0.00	0.00	0.00	90.71	0.2741	
kNN-UNION	–	–	–	–	93.17	0.2692	
Room 1 $\wedge$ Room 3							
QR	100.00	0.00	0.00	0.00	89.67	0.2851	0.2583
CQR	100.00	0.00	0.00	0.00	89.67	0.2903	
CB-QR	100.00	0.00	0.00	0.00	90.21	0.3018	
kNN-CQR	100.00	0.00	0.00	0.00	90.46	0.2912	
MIN	100.00	0.00	0.00	0.00	90.17	0.2764	
UNION	–	–	–	–	93.96	0.2850	
CB-MIN	100.00	0.00	0.00	0.00	90.63	0.2875	
CB-UNION	–	–	–	–	94.03	0.4520	
kNN-MIN	100.00	0.00	0.00	0.00	90.71	0.2741	
kNN-UNION	–	–	–	–	93.17	0.2692	
Room 1 $\wedge$ Room 4							
QR	100.00	0.00	0.00	0.00	92.10	0.2983	0.2760
CQR	100.00	0.00	0.00	0.00	92.10	0.2783	
CB-QR	100.00	0.00	0.00	0.00	95.47	0.4555	
kNN-CQR	1.00	99.00	0.00	0.00	89.77	0.2786	
MIN	100.00	0.00	0.00	0.00	89.78	0.2784	
UNION	–	–	–	–	91.25	0.3209	
CB-MIN	1.00	99.00	0.00	0.00	95.48	0.4554	
CB-UNION	–	–	–	–	95.51	0.6008	
kNN-MIN	100.00	0.00	0.00	0.00	89.77	0.27486	
kNN-UNION	–	–	–	–	91.30	0.3218	

6. Related work

Learning-based approaches to runtime PM have been recently proposed, including the so-called Neural Predictive Monitoring (NPM) method [1–3,12,39,40]. In NPM, neural networks are used to infer the Boolean satisfaction of a property and conformal prediction (CP) are used to provide statistical guarantees. However, NPM is restricted to reachability properties. NPM has been extended to support some source of stochasticity in the system: in [3] they allow partial observability and noisy observations, whereas in [40], the system dynamics are stochastic but the monitor only evaluates the Boolean satisfaction of some quantile

Table 5
Gene Regulatory Network results with four genes on the right. Correct, uncertain, wrong and coverage represent percentage values (%). *EQR width* denotes the average width of the empirical quantile range over the test set.

GRN4	Correct	Uncertain	Wrong	FP	Coverage	Efficiency	EQR width
Gene 1							
QR	95.50	3.50	1.00	1.00	93.62	0.5627	0.5062
CQR	97.25	1.25	1.50	1.50	89.87	0.5051	
CB-CQR	91.25	8.25	0.50	0.50	90.14	0.5393	
kNN-CQR	97.00	1.75	1.25	1.25	89.38	0.5033	
Gene 2							
QR	88.75	9.75	1.50	1.50	93.80	0.5124	0.4359
CQR	91.25	7.00	1.75	1.75	89.85	0.4700	
CB-CQR	88.00	10.50	1.50	1.50	89.76	0.4786	
kNN-CQR	90.25	7.75	2.00	2.00	90.39	0.4671	
Gene 3							
QR	100.00	0.00	0.00	0.00	96.94	0.3734	0.2620
CQR	100.00	0.00	0.00	0.00	89.92	0.2993	
CB-CQR	0.00	100.00	0.00	0.00	94.72	inf	
kNN-CQR	100.00	0.00	0.00	0.00	90.45	0.2967	
Gene 4							
QR	94.50	4.50	1.00	1.00	84.18	0.3757	0.2825
CQR	91.75	7.75	0.50	0.50	89.71	0.4175	
CB-CQR	91.75	7.00	1.25	1.25	89.75	0.4097	
kNN-CQR	95.50	3.50	1.00	1.00	90.55	0.3950	
Gene 1 $\wedge$ Gene 2							
QR	86.00	10.50	3.50	3.25	89.94	0.5337	0.4723
CQR	86.00	10.50	3.50	3.25	89.94	0.5387	
CB-CQR	85.50	12.25	2.25	1.75	89.71	0.5492	
kNN-CQR	88.00	9.00	3.00	3.00	89.07	0.5241	
MIN	88.75	8.25	3.00	3.00	89.36	0.4775	
UNION	–	–	–	–	90.55	0.5339	
CB-MIN	86.75	11.25	2.00	2.00	89.41	0.4946	
CB-UNION	–	–	–	–	92.50	0.8428	
kNN-MIN	89.00	7.75	3.25	3.25	89.15	0.4716	
kNN-UNION	–	–	–	–	91.11	0.5319	
Gene 1 $\wedge$ Gene 3							
QR	94.25	4.00	1.75	1.50	89.88	0.5503	0.4763
CQR	94.25	4.00	1.75	1.50	89.88	0.5489	
CB-CQR	89.75	9.50	0.75	0.75	90.10	0.5822	
kNN-CQR	93.50	4.00	2.50	2.50	89.17	0.5428	
MIN	97.25	1.25	1.50	1.50	90.04	0.4749	
UNION	–	–	–	–	90.78	0.4650	
CB-MIN	91.25	8.25	0.50	0.50	90.25	0.5090	
CB-UNION	–	–	–	–	100.00	inf	
kNN-MIN	97.50	1.50	1.00	1.00	89.54	0.4717	
kNN-UNION	–	–	–	–	90.80	0.4638	
Gene 1 $\wedge$ Gene 4							
QR	87.25	8.75	4.00	2.75	89.38	0.4873	0.4133
CQR	87.00	9.00	4.00	2.75	89.38	0.4932	
CB-CQR	85.75	11.25	3.00	1.50	89.22	0.4927	
kNN-CQR	89.75	6.50	3.75	2.25	89.61	0.4993	
MIN	93.00	5.00	2.00	2.00	89.54	0.4274	
UNION	–	–	–	–	92.03	0.5036	
CB-MIN	92.50	5.75	1.75	1.75	89.56	0.4365	
CB-UNION	–	–	–	–	92.53	0.8116	
kNN-MIN	93.25	5.00	1.75	1.75	89.27	0.4233	
kNN-UNION	–	–	–	–	91.86	0.4923	

trajectories, providing a limited understanding of the safety level of the current state. Predictive monitoring under partial observability is also analyzed in [9], where the authors combine Bayesian state estimation with pre-computed reach sets to reduce the runtime overhead. While their reachability bounds are certified, no correctness guarantees can be established for the estimation step.

Other learning-based approaches for reachability prediction of stochastic systems include [8,41–45]. Of these, [8] develop techniques to detect potential prediction errors using neural network verification methods [46]. These verification methods, however, do not scale well on large models and support only specific classes of neural networks. In [42], the authors introduce

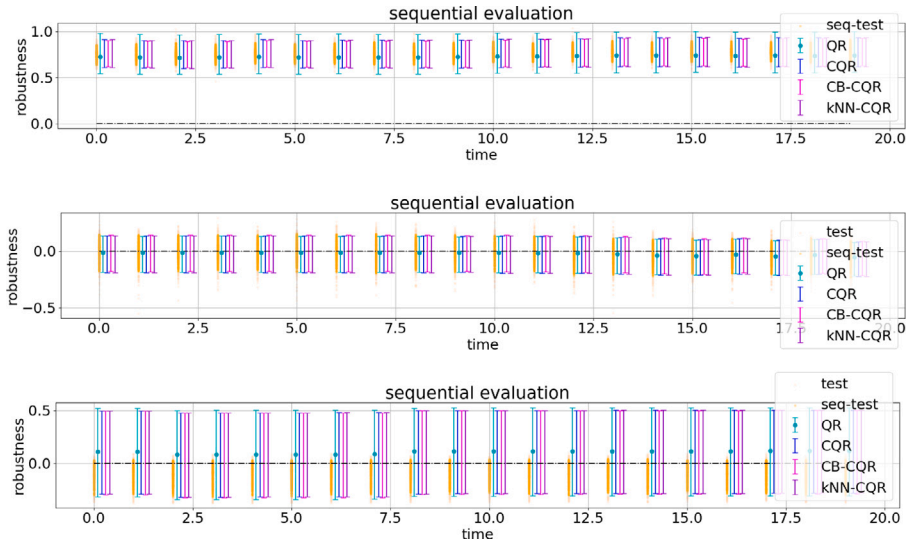


Fig. 19. Visualization of prediction intervals – cyan for PI, blue for CPI, magenta for CB-CPI and violet for kNN-CPI – for a runtime evaluation of the AAD (top), the MRH2 (middle) and the GRN2 (bottom) system.

an efficient approximate Bayesian monitor for the satisfaction of STL properties for stochastic systems. However, this work focuses, once again, on Boolean satisfaction.

Various learning-based PM approaches for temporal logic properties [7,10,23,47–49] have been recently proposed. In particular, [7] provide (like we do) guaranteed prediction intervals, but (unlike our method), they are limited to ARMA/ARIMA models. Ma et al. [10] use uncertainty quantification with Bayesian RNNs to provide confidence guarantees. However, these models are, by nature, not well-calibrated (i.e., the model uncertainty does not reflect the observed one [50]), making the resulting guarantees not theoretically valid.

An alternative method for localized conformal predictions, introduced in [51], involves employing kernel density estimation on training samples to learn the cumulative density function of the calibration score conditional on the input and use its predictions to adapt the non-conformity scores. The resulting prediction intervals are calibrated on a point-by-point basis and provide quasi-conditional guarantees, while maintaining the principle of exchangeability between calibration and test samples.

Recent advances in predictive monitoring have yielded sophisticated techniques capable of handling stochastic processes and signal temporal logic specifications with quantitative semantics [52–55]. For a comprehensive overview of these methodologies and their theoretical foundations, we refer readers to the detailed survey presented in [56].

We contribute to the state of the art by developing a quantitative predictive monitor that offers good scalability, provides statistical guarantees (both marginal and conditional), and supports stochastic systems with rich STL-based requirements. Another important novelty is to monitor and predict the quantitative, rather than the Boolean, STL satisfaction of a requirement.

7. Conclusions

We presented quantitative predictive monitoring (QPM), a technique to reliably monitor the evolution of a stochastic system at runtime. In particular, given a requirement expressed as an STL formula, QPM quantifies how robustly this requirement is satisfied both in space and in time by means of a range of STL robustness values. This interval undergoes a principled recalibration that guarantees a desired level of coverage, i.e. the interval covers the exact STL robustness values with a given confidence. By carefully designing the recalibration strategy, we can achieve either marginal or conditional coverage guarantees. The proposed technique avoids expensive Monte-Carlo simulations at runtime by leveraging conformalized quantile regression. The resulting method has very little overhead during runtime execution.

Our experimental evaluation demonstrates that we reach a high accuracy and low rate of prediction errors, and the statistical guarantees are always met. The conformal approach also adjusts the width of the interval w.r.t. desired confidence level, reducing the number of over-conservative predictions. These results support the claim of having an efficient and reliable QPM.

In future work, we will investigate a possible dynamics-aware approach to inference. The latter should aim at limiting the inference only to an estimate of the system manifold, i.e., the state space region likely to be visited by the evolving stochastic process.

CRediT authorship contribution statement

Francesca Cairoli: Writing – original draft, Methodology, Investigation, Conceptualization. **Tom Kuipers:** Data curation. **Luca Bortolussi:** Supervision. **Nicola Paoletti:** Writing – review & editing, Supervision, Methodology, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been partially supported by the PNRR project iNEST (Interconnected North-Est Innovation Ecosystem) funded by the European Union Next-GenerationEU (Piano Nazionale di Ripresa e Resilienza (PNRR) – Missione 4 Componente 2, Investimento 1.5 – D.D. 1058 23/06/2022, ECS_00000043) and by the "REXASI-PRO" H-EU project, call HORIZON-CL4-2021-HUMAN-01-01, Grant agreement ID: 101070028.

Appendix A. Case studies

The first two case studies are selected from the stochastic hybrid benchmarks presented in the ARCH-COMP competitions [30]. We then consider two stochastic hybrid systems with modular nature where the complexity can arbitrarily grow, making them suitable to test the scalability of the proposed solution.

Automated Anaesthesia Delivery (AAD) [30]: three dimensional discrete-time stochastic process with state $s(t) = (q(t), v(t))$ evolving according to the following dynamics:

$$v(t_{k+1}) = A \cdot v(t_k) + b \cdot q(t_k) + w(t_k),$$

where A, b are patient-specific parameters and there is a Gaussian disturbance $w(t) \sim \mathcal{N}(0, 10^{-3}\mathbb{I}_3)$. The properties to be monitored are about the safety of the system, $\phi_1 = G_{[0,t_H]}(v \in Safe)$, $\phi_2 = F_{[0,t_H]}(v_1 < v_2)$ where $Safe = [1, 6; 0, 10; 0, 10]$, $H = 60$ min, $\Delta t = 20$ s. The controller is defined as follows: $q = 7$ if $v_1 < 3.5$ and $q = 3.5$ if $v_1 \geq 3.5$.

Heated Tank (HT): system composed of a tank T containing a liquid whose level is influenced by two inflow pumps P_1 and P_2 and a outflow valve W, all managed by a controller C. The liquid should absorb and transport the heat from a heat source. The discrete state space is five-dimensional composed of states q:

$$q(t) = (q_{P_1}(t), q_{P_2}(t), q_W(t), q_C(t), q_T(t)).$$

The general model adopted for this heated tank system is that of piece-wise deterministic Markov processes. The continuous state space is two-dimensional composed of a state $v(t) = (v_{height}(t), v_{temp}(t))$ representing the liquid height and the liquid temperature. The continuous part evolves according to the following switching differential equations:

$$\dot{v}_H(t) = (q_{P_1}(t) + q_{P_2}(t) - q_W(t)) \cdot g$$
$$\dot{v}_T(t) = \frac{((\chi_{P_1}(t) + \chi_{P_2}(t)) \cdot (T_{in} - v_{temp}(t)) \cdot g + E_{in})}{v_{height}(t)},$$

where q_i indicates if unit $i \in \{P_1, P_2, W\}$ is working 1 or not 0, g is the flow parameter, $T_{in} = 15\,\mathrm{deg}\,C$ is the inflow temperature and $E_{in} = 1\frac{\mathrm{deg}\,Cm}{h}$ is the heat source parameter. Each unit can be in one of the four discrete modes $Q_U = \{On, Off, StuckOn, StuckOff\}$. The switching from On to Off and vice-versa is managed by the controller. In addition to this switching, there are two exponentially distributed failure switching possibilities: from On to $StuckOn$ and from Off to $StuckOff$ with rates $\lambda_{P_1} = 1/219$, $\lambda_{P_2} = 1/175$ and $\lambda_W = 1/320$. The controller mode q_C switches between the following configurations $Q_C = \{Normal, Increase, Decrease\}$. If $Normal$ the controller does not try to influence the pumps and valve. If $Increase$ the controller aims to increase the height of the liquid in the tank, by switching both pumps on, and by switching the valve off. If $Decrease$ the controller aims to decrease the height of the liquid in the tank, by switching both pumps off, and by switching the valve on. The switching by the controller has no effect on unit i if it is in failure mode $StuckOn$ or $StuckOff$. If $v_{height}(t) \leq H_{low}$ then the controller switches from $Normal$ or $Decrease$ to $Increase$. If $v_{height}(t) \geq H_{high}$ the controller switches from $Normal$ or $Increase$ to $Decrease$. $H_{low} = 6$ m, $H_{high} = 8$ m. The property to monitor is $\phi = G_{[t_0,t_H]}\big((H_{dryout} \leq v_{height} \leq H_{overflow}) \wedge (v_{temp} \leq T_{overheat})\big)$.

Multi-Room Heating (MRH) [31–33]: discete-time stochastic hybrid system modeling the temperature evolution in a building with h rooms. In each room is located a heater that switches between *on* and *off* depending on the current temperature in the room. The state of the system is hybrid $S = Q \times V$: the discrete state represents the status of the h heaters, $Q = \{on, off\}^h$, and the continuous state represents the temperatures in the h rooms, $V = \mathbb{R}^h$. The average temperature evolves according to the following stochastic difference equation:

$$v_i(t_{k+1}) = v_i(t_k) + b_i(x_a - v_i(t_k)) +$$
$$+ \sum_{i \neq j} a_{ij}(v_j(t_k) - v_i(t_k)) + c_i \mathbb{1}_{Q_i}(q(t_k)) + w_i(t_k),$$

where x_a is the ambient temperature (constant for the entire building) and $\mathbb{1}_{Q_i}(\cdot)$ is the indicator function of set $Q_i := \{(q_1, \ldots, q_h) \in Q : q_i = on\}$. The quantities b_i, a_{ij} and c_i are non-negative constants representing respectively the average heat transfer rate from room i to the ambient (b_i) and to room $j \neq i$ (a_{ij}) and the heat rate supplied to room i by the heater in room i (c_i). a_{ij} also reflects the rooms layout, e.g. $a_{ij} = 0$ if room i and j are not adjacent. The overall complexity of the stochastic hybrid system is determined

by the number of rooms. The room-specific requirement to monitor is: the temperature v_i of room i always stays in a desirable range $[V_i^{lb}, V_i^{ub}]$ and it can be expressed by the STL property $\phi_i = G_{[0,t_H]}(V_i^{lb} \leq v_i \leq V_i^{ub})$.

Gene Regulatory Network(GRN) [34,35]: discrete-time stochastic hybrid system modeling a genetic regulatory network composed of h genes that produces respectively h proteins that repress each other in a cyclic fashion. Each gene could either be *on*, actively producing its protein, or *off*, expression is deactivate when a repressing protein binds to the gene. The state is thus hybrid $S = Q \times V$: the state of the genes is discrete, $Q = \{on, off\}^h$, whereas the proteins count is continuous, $V = \mathbb{N}^h$. Protein counts change deterministically according to a linear differential equation, whereas transition between modes follow a Markov jump process. The change of protein i has the form $a_i - b_i v_i$ if $q_i = off$ and $c_i - d_i v_i$ if $q_i = on$, where a_i and c_i are the production rates of protein i if the gene is inactive or active, respectively. The respective degradation rate constants are given by b_i and d_i. The binding rate between a protein $i-1$ and a gene i is expressed as $k_b \cdot v_{i-1} q_i$, whereas the unbinding rate is constant k_u. The shooting time for this reactions is exponentially distributed. The complexity of the stochastic hybrid system is determined by the number of genes. The gene-specific requirement to monitor is: the count v_i of protein i stabilize under a certain threshold V_i^{ub} and it can be expressed by the STL property $\phi_i = G_{[t_{H/2}, t_H]}(v_i \leq V_i^{ub})$.

Multi Particle Environment (MPE) [36]: discrete-time stochastic multi-agent reinforcement learning (MARL) system composed of N agents – one of which is an adversarial agent denoted a_e – and M *landmarks*. Each *good* agent aims to minimize its distance to a predetermined *goal* landmark, l_{goal}, whilst avoiding the other agents, particularly the adversary. The objective of the adversary is to create as much distance as possible between all agents and the goal (which is not known by the adversary a priori). The stochastic process this system follows is described by the following equation:

$$\mathbf{S}(1) \sim P_{init}(\cdot);\ \mathbf{S}(t+1) \sim P(\cdot \mid \mathbf{S}(t), \mathbf{A}(t));$$
$$A_i(t) \sim \pi_i(\cdot \mid Y_i(t));\ Y_i(t) = o_i(\mathbf{S}(t)) + W_i^{sensor}$$

where capital letters denote random variables, lowercase as concrete realizations of those variables, and bold letters denote vectors of random variables/realizations. The state space $S \subseteq \mathbb{R}^{2(2 \cdot k + M)}$ describes the (two-dimensional) positions and velocities of the k agents, and the positions of the M landmarks and is given as $s(t) = ((x_i(t), v_i(t))_{i=1}^N, l_1, \ldots, l_M)$. The initial state of the world at time 0 is sampled from an initial distribution $P_{init}(\cdot)$. In each step, each agent a_i receives an noisy observation (subset) of the global state given as: $y_i(t) = (x_i(t), v_i(t), (x_j(t))_{j \neq i}, l_1, \ldots, l_M) + w_i^{sensor}$ and operates according to deep neural network policies trained with Deep Q-Networks [37], $\pi : Y \to \Delta^A$, where A is defined as the actions *{do nothing, left, right, down, up}* and Δ^A is the probability simplex over A. All subsequent states are derived from the conditional distribution of the current state, the collection of actions taken by every agent, $\mathbf{A}$, and the actuation noise applied at each step for each agent. The system dynamics/transition kernel P are described by as second-order motion laws determined by the current state, the performed action and some degree of actuation noise applied. Specifically for each agent, a_i, it can be defined as:

$$X_i(t+1) = X_i(t) + V_i(t) + W_i^{act};\ V_i(t+1) = V_i(t) + U_i(t),$$

where X_i and V_i are the position and velocity of agent i, W_i^{act} a white Gaussian noise term modeling actuation noise, and U_i is the control input, which is determined from the (discrete) action A_i as follows:

$$U_i(t) = c_u \cdot \begin{cases} [0,1]^T & \text{if } A_i(t) = \text{up} \\ [0,-1]^T & \text{if } A_i(t) = \text{down} \\ [1,0]^T & \text{if } A_i(t) = \text{right} \\ [-1,0]^T & \text{if } A_i(t) = \text{left} \\ [0,0]^T & \text{otherwise} \end{cases}$$

where $c_u > 0$ is a parameter determining the intensity of the acceleration. The (global) property to be monitored is one of collision-avoidance: all agents (including the adversary) maintain a safe distance d_s throughout the trajectory. It is described by the STL property: $\phi = G_{[0,H]}\left(\wedge_{a_i \neq a_j} d\left(a_i, a_j\right) > d_s\right)$.

Appendix B. Guarantees of the union

Proposition. Given two properties ϕ_1 and ϕ_2, let CPI^{ϕ_i} be the monitor that maps each state $s \in S$ into a prediction interval for $\mathcal{R}_{\phi_i}(s)$ with coverage $1-\alpha$. Then, the interval $CPI^{\phi_{12}} := CPI^{\phi_1} \cup CPI^{\phi_2}$ is a valid prediction interval for both $\mathcal{R}_{\phi_1 \wedge \phi_2}$ and $\mathcal{R}_{\phi_1 \vee \phi_2}$, i.e.,

$$\mathbb{P}\left(R_{\phi_1 \wedge \phi_2}(s) \in CPI^{\phi_{12}}(s)\right) \geq 1 - \alpha \tag{B.1}$$

and

$$\mathbb{P}\left(R_{\phi_1 \vee \phi_2}(s) \in CPI^{\phi_{12}}(s)\right) \geq 1 - \alpha \tag{B.2}$$

for every state $s \in S$.

Proof. We know that, for $i \in \{1,2\}$ and for each state $s \in S$, $\mathbb{P}\Big(R_{\phi_i}(s) \in CPI^{\phi_i}(s)\Big) \geq 1-\alpha$. We also know that, by the STL quantitative semantics,

$$\mathcal{R}_{\phi_1 \wedge \phi_2}(s) = \min\Big(\mathcal{R}_{\phi_1}(s), \mathcal{R}_{\phi_2}(s)\Big).$$

The following inequalities hold:

$$\begin{aligned}
&\mathbb{P}\Big(\min\big(\mathcal{R}_{\phi_1}(s), \mathcal{R}_{\phi_2}(s)\big) \in CPI^{\phi_{12}}(s)\Big) = \\
&\mathbb{P}\big(\mathcal{R}_{\phi_1}(s) \in CPI^{\phi_{12}}(s)\big) + \mathbb{P}\big(\mathcal{R}_{\phi_2}(s) \in CPI^{\phi_{12}}(s)\big) \\
&- \mathbb{P}\big(\mathcal{R}_{\phi_1}(s) \in CPI^{\phi_{12}}(s), \mathcal{R}_{\phi_2}(s) \in CPI^{\phi_{12}}(s)\big).
\end{aligned}$$

The first two addends are higher than $1-\alpha$ since $CPI^{\phi_i}(s) \subseteq CPI^{\phi_{12}}(s)$ for every $i \in \{1,2\}$. The last addend is always lower than $\mathbb{P}\Big(\min\big(\mathcal{R}_{\phi_1}(s), \mathcal{R}_{\phi_1}(s)\big)\Big)$. Thus the inequality becomes:

$$2\mathbb{P}\Big(\min\big(\mathcal{R}_{\phi_1}(s), \mathcal{R}_{\phi_2}(s)\big) \in CPI^{\phi_{12}}(s)\Big) \geq 2(1-\alpha),$$

which proves the proposition. The proof for the $\phi_1 \vee \phi_2$ is very similar, the main difference is that

$$\mathcal{R}_{\phi_\vee \phi_2}(s) = \max\Big(\mathcal{R}_{\phi_1}(s), \mathcal{R}_{\phi_1}(s)\Big). \quad \square$$

Appendix C. Sign-conditional guarantees

To achieve sign-conditional guarantees the calibration set Z_c^{ϕ} is split in its positive $Z_c^{\phi(+)}$ and negative $Z_c^{\phi(-)}$ part. The prediction interval $PI(s)$ induces two separate calibration distributions of nonconformity scores from which we extract the two $(1-\alpha)$-quantiles, τ^+ and τ^- respectively. In other words, τ^+ is the $\lfloor \alpha \cdot (|Z_c^{\phi(+)}|+1) \rfloor$-th largest calibration score (same for τ^- w.r.t. $Z_c^{\phi(-)}$). To ease the notation, in the following, we denote the prediction intervals as $CPI(\cdot) := CPI_\alpha(\cdot; Z_c^{\phi})$, i.e., we drop the dependence on α and Z_c^{ϕ}. The prediction interval $PI(s)$ undergoes a sign-specific calibration obtaining $\overline{CPI}^{+}(s) = CPI^{+}(s) \cap \mathbb{R}^+$, where $CPI^{+}(s)$ is defined as in (19), guaranteed to cover $1-\alpha$ of the positive values of $R_\phi(\vec{s},0)$:

$$\mathbb{P}\Big(R_\phi(\vec{s},0) \in \overline{CPI}^{+}(\vec{s}(0)) | R_\phi(\vec{s},0) \geq 0\Big) \geq 1-\alpha \tag{C.1}$$

and $\overline{CPI}^{-}(x) = CPI^{-}(x) \cap \mathbb{R}^-$ is guaranteed to cover $1-\alpha$ of the negative values of $R_\phi(\vec{s},0)$:

$$\mathbb{P}\Big(R_\phi(\vec{s},0) \in \overline{CPI}^{-}(\vec{s}(0)) | R_\phi(\vec{s},0) < 0\Big) \geq 1-\alpha \tag{C.2}$$

The union $CPI(s) := \overline{CPI}^{-}(s) \cup \overline{CPI}^{+}(s)$ of the two recalibrated interval is guaranteed to cover $1-\alpha$ of all the values of $R_\phi(\vec{s},0)$ for $\vec{s}(0) = s$ (to ease the notation this remark is omitted in what follows):

$$\begin{aligned}
\mathbb{P}\Big(R_\phi(\vec{s},0) \in CPI(s)\Big) &= \mathbb{P}\Big(R_\phi(\vec{s},0) \in CPI(s) | R_\phi(\vec{s},0) \geq 0\Big)\mathbb{P}(R_\phi(\vec{s},0) \geq 0) + \\
&+ \mathbb{P}\Big(R_\phi(\vec{s},0) \in CPI(s) | R_\phi(\vec{s},0) < 0\Big)\mathbb{P}(R_\phi(\vec{s},0) < 0) \\
&\geq (1-\alpha)\Big(\mathbb{P}(R_\phi(\vec{s},0) \geq 0) + \mathbb{P}(R_\phi(\vec{s},0) < 0)\Big) = (1-\alpha).
\end{aligned} \tag{C.3}$$

Appendix D. Additional plots of results

See Figs. D.20–D.23

Appendix E. Additional tables of results

See Tables E.6, E.7a, E.7b and E.8a–E.8c

Data availability

Data and code are available online.

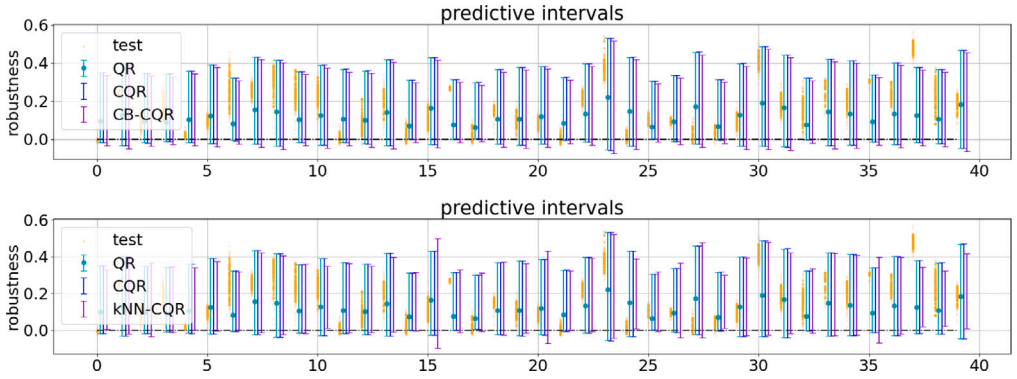


Fig. D.20. Visualization of prediction intervals – cyan for QR and blue for CQR – over 40 randomly selected states of the multi-agent model MPE3 (3 agents and 1 adversary) with **space robustness**. Purple intervals denote either the sign-balanced (top) or the localized (bottom) prediction intervals.

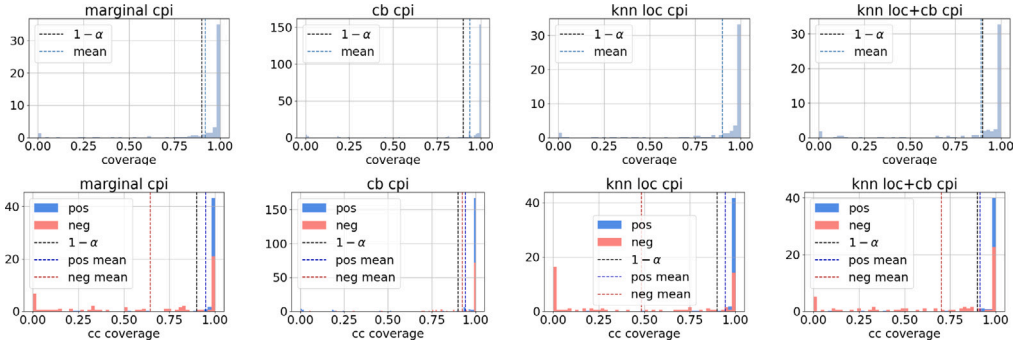


Fig. D.21. Visualization of coverage distributions, either marginal (top line) or sign-conditional (bottom line), for the multi-agent system with 3 agents and 1 adversary (MPE3) with **space robustness**.

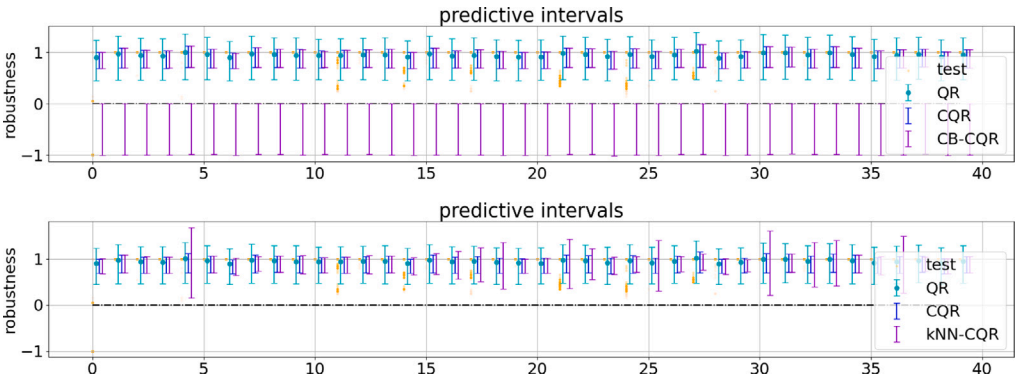


Fig. D.22. Visualization of prediction intervals – cyan for QR and blue for CQR – over 40 randomly selected states of the multi-agent model MPE3 (3 agents and 1 adversary) with **time robustness**. Purple intervals denote either the sign-balanced (top) or the localized (bottom) prediction intervals.

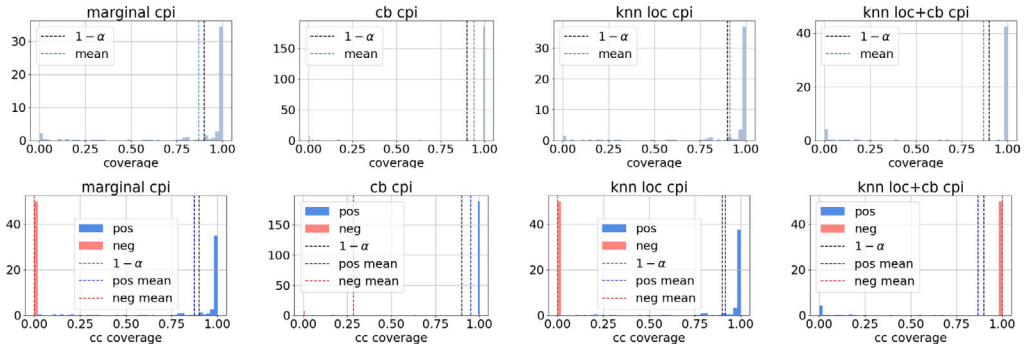


Fig. D.23. Visualization of coverage distributions, either marginal (top line) or sign-conditional (bottom line), for the multi-agent system with 3 agents and 1 adversary (MPE3) with **time robustness**.

Table E.6
Correct, uncertain, wrong and coverage represent percentage values (%). *EQR width* denotes the average width of the empirical quantile range over the test set.

MPE3-space	Correct	Uncertain	Wrong	FP	Coverage	Efficiency	EQR width
QR	24.00	73.00	3.00	3.00	91.62	0.3933	0.1099
CQR	24.00	73.50	2.50	2.50	91.78	0.3943	
CB-CQR	21.50	78.50	0.00	0.00	93.67	0.3967	
kNN-CQR	38.00	59.50	2.50	2.50	89.84	0.3899	
MPE3-time	**correct**	**uncertain**	**wrong**	FP	**coverage**	**efficiency**	EQR width
QR	98.00	0.00	2.00	20.00	93.48	0.8236	0.0959
CQR	98.00	0.00	2.00	2.00	89.91	0.3618	
CB-CQR	1.00	99.00	0.00	0.00	93.98	2.0529	
kNN-CQR	98.00	0.00	2.00	2.00	91.19	0.5417	

Table E.7a
Gene regulatory network with six genes (GRN6). Correct, uncertain, wrong ∧ coverage represent percentage values (%). *EQR width* denotes the average width of the empirical quantile range over the test set.

GRN6	Correct	Uncertain	Wrong	FP	Coverage	Efficiency	EQR width
Gene 1							
QR	93.17	6.83	0.00	0.00	94.53	0.4370	0.3772
CQR	95.50	4.17	0.33	0.33	89.98	0.3981	
CB-CQR	88.67	11.33	0.00	0.00	90.04	0.4170	
kNN-CQR	94.83	4.67	0.50	0.50	89.93	0.3995	
Gene 2							
QR	91.67	7.33	1.00	1.00	92.15	0.4217	0.3502
CQR	92.83	6.00	1.17	1.17	89.35	0.3928	
CB-CQR	84.83	14.33	0.83	0.83	89.38	0.4148	
kNN-CQR	93.00	5.67	1.33	1.33	89.66	0.3944	
Gene 3							
QR	89.83	9.33	0.83	0.83	93.98	0.3954	0.3237
CQR	90.50	8.67	0.83	0.83	89.56	0.3619	
CB-CQR	88.67	10.67	0.67	0.67	89.41	0.3688	
kNN-CQR	90.50	8.67	0.83	0.83	90.01	0.3627	
Gene 4							
QR	87.33	11.17	1.50	1.50	94.30.11	0.3785	0.3000
CQR	88.67	9.33	2.00	2.00	89.67	0.3140	
CB-CQR	88.33	10.33	1.33	1.33	89.93	0.3228	
kNN-CQR	88.67	9.66	1.67	1.67	89.77	0.3152	
Gene 5							
QR	100.00	0.00	0.00	0.00	89.08	0.2970	0.2032
CQR	100.00	0.00	0.00	0.00	89.55	0.2997	
CB-CQR	0.00	100.00	0.00	0.00	91.74	inf	
kNN-CQR	100.00	0.00	0.00	0.00	90.04	0.2843	

(*continued on next page*)

Table E.7a (*continued*).

Gene 6							
QR	93.00	6.50	0.50	0.50	84.92	0.3439	0.2745
CQR	90.83	8.67	0.50	0.50	90.18	0.3711	
CB-CQR	91.83	7.67	0.50	0.50	89.89	0.3637	
kNN-CQR	93.33	6.33	0.33	0.33	90.36	0.3675	

Table E.7b
Gene regulatory network with six genes (GRN6). Combination of single-gene properties. Correct, uncertain, wrong ∧ coverage represent percentage values (%). *EQR width* denotes the average width of the empirical quantile range over the test set.

GRN6	Correct	Uncertain	Wrong	FP	Coverage	Efficiency	EQR width
Gene 1 ∧ Gene 2							
QR	88.17	9.17	2.67	2.50	89.22	0.4422	0.3834
CQR	87.83	9.67	2.50	2.50	89.22	0.4440	
CB-CQR	83.33	15.33	1.33	0.83	89.60	0.4591	
kNN-CQR	88.17	7.50	4.33	3.50	87.10	0.4308	
MIN	90.33	8.50	1.17	1.17	89.49	0.3983	
UNION	–	–	–	–	90.37	0.4536	
CB-MIN	82.83	16.33	0.83	0.83	89.62	0.4221	
CB-UNION	–	–	–	–	91.85	0.7047	
kNN-MIN	91.17	6.67	2.17	2.17	87.95	0.3937	
kNN-UNION	–	–	–	–	90.77	0.4518	
Gene 1 ∧ Gene 3							
QR	87.17	11.17	1.67	1.67	88.08	0.4190	0.3522
CQR	86.17	12.50	1.33	1.33	88.08	0.4358	
CB-CQR	84.67	13.33	2.00	1.50	89.61	0.4385	
kNN-CQR	87.00	10.33	2.67	1.67	86.97	0.4204	
MIN	88.50	10.50	1.00	1.00	89.43	0.3669	
UNION	–	–	–	–	91.29	0.4297	
CB-MIN	84.33	15.17	0.50	0.50	89.44	0.3833	
CB-UNION	–	–	–	–	92.34	0.6708	
kNN-MIN	88.17	10.67	1.17	1.17	88.68	0.3668	
kNN-UNION	–	–	–	–	90.99	0.4268	
Gene 1 ∧ Gene 4							
QR	88.67	9.17	2.17	1.67	89.69	0.4069	0.3373
CQR	89.00	9.33	1.67	1.67	89.69	0.4119	
CB-CQR	86.33	12.17	1.50	1.00	90.29	0.4178	
kNN-CQR	88.17	9.33	2.50	2.33	88.03	0.3978	
MIN	87.17	11.33	1.50	1.50	90.29	0.3507	
UNION	–	–	–	–	90.82	0.4049	
CB-MIN	83.67	15.50	0.83	0.83	90.41	0.3654	
CB-UNION	–	–	–	–	92.18	0.6316	
kNN-MIN	87.50	10.67	1.83	1.83	89.15	0.3457	
kNN-UNION	–	–	–	–	90.46	0.4027	
Gene 1 ∧ Gene 5							
QR	94.50	4.00	1.50	1.33	88.09	0.4240	0.3552
CQR	93.83	5.83	0.33	1.33	88.09	0.4431	
CB-CQR	88.50	10.50	1.00	0.50	89.82	0.4509	
kNN-CQR	92.00	5.83	2.17	0.02	87.91	0.4283	
MIN	95.50	4.17	0.33	0.33	89.86	0.3721	
UNION	–	–	–	–	90.92	0.4003	
CB-MIN	88.67	11.33	0.00	0.00	89.86	0.3907	
CB-UNION	–	–	–	–	100.00	inf	
kNN-MIN	95.00	4.33	0.67	0.67	88.85	0.3692	
kNN-UNION	–	–	–	–	90.21	0.3913	

(*continued on next page*)

Table E.7b (*continued*).

Gene 1 ∧ Gene 6							
QR	86.33	11.00	2.67	1.83	88.27	0.4215	0.3466
CQR	85.83	11.50	2.67	1.83	88.27	0.4301	
CB-CQR	85.33	13.00	1.67	0.83	89.74	0.4349	
kNN-CQR	88.17	9.17	2.67	1.83	86.84	0.4193	
MIN	91.50	8.00	0.50	0.17	89.50	0.3710	
UNION	–	–	–	–	93.15	0.4297	
CB-MIN	90.17	9.50	0.33	0.00	89.62	0.3771	
CB-UNION	–	–	–	–	93.40	0.6866	
kNN-MIN	91.67	7.67	0.67	0.33	88.85	0.3652	
kNN-UNION	–	–	–	–	92.96	0.4243	

Table E.8a
Multi room heating results with 8 rooms. Correct, uncertain, wrong and coverage represent percentage values (%). *EQR width* denotes the average width of the empirical quantile range over the test set.

MRH8	Correct	Uncertain	Wrong	FP	Coverage	Efficiency	EQR width
Room 1							
QR	68.25	31.75	0.00	0.00	90.59	0.2320	0.2241
CQR	68.25	31.75	0.00	0.00	90.62	0.2332	
CB-CQR	68.25	31.75	0.00	0.00	91.98	0.3039	
kNN-CQR	62.25	25.00	12.75	12.75	90.47	0.2338	
Room 2							
QR	98.25	0.88	0.87	0.87	75.89	0.1769	0.1314
CQR	88.13	11.75	0.12	0.12	99.42	0.3184	
CB-CQR	85.00	15.00	0.00	0.00	99.45	0.3217	
kNN-CQR	89.38	10.50	0.12	0.12	98.53	0.3064	
Room 3							
QR	99.00	0.00	1.00	1.00	84.60	0.1663	0.1222
CQR	96.13	3.87	0.00	0.00	98.47	0.2684	
CB-CQR	79.25	20.75	0.00	0.00	98.79	0.2920	
kNN-CQR	96.00	3.25	0.75	0.75	92.80	0.2437	
Room 4							
QR	100.00	0.00	0.00	0.00	89.30	0.2591	0.2641
CQR	100.00	0.00	0.00	0.00	90.01	0.2646	
CB-CQR	0.75	99.25	0.00	0.00	94.04	0.4254	
kNN-CQR	100.00	0.00	0.00	0.00	90.13	0.2663	
Room 5							
QR	100.00	0.00	0.00	0.00	88.34	0.2522	0.2364
CQR	100.00	0.00	0.00	0.00	88.51	0.2533	
CB-CQR	0.00	100.00	0.00	0.00	97.96	inf	
kNN-CQR	100.00	0.00	0.00	0.00	88.53	0.2540	
Room 6							
QR	98.13	0.00	1.87	1.87	90.41	0.1841	0.1423
CQR	98.13	0.12	1.75	1.75	95.45	0.2123	
CB-CQR	80.13	19.87	0.00	0.00	95.81	0.2323	
kNN-CQR	98.13	0.12	1.75	1.75	94.08	0.2053	
Room 7							
QR	99.00	0.00	1.00	1.00	90.11	0.1728	0.1126
CQR	99.00	0.00	1.00	1.00	89.56	0.1696	
CB-CQR	85.63	14.37	0.00	0.00	94.80	0.1839	
kNN-CQR	99.00	0.00	1.00	1.00	87.93	0.1640	
Room 8							
QR	99.88	0.12	0.00	0.00	90.52	0.2525	0.2422
CQR	99.88	0.12	0.00	0.00	90.72	0.2540	
CB-CQR	99.88	0.12	0.00	0.00	90.73	0.2571	
kNN-CQR	99.88	0.12	0.00	0.00	90.69	0.2549	

Table E.8b
Multi-Room Heating with eight rooms (MRH8). Combination of single-room properties. Correct, uncertain, wrong ∧ coverage represent percentage values (%). *EQR width* denotes the average width of the empirical quantile range over the test set.

MRH8	Correct	Uncertain	Wrong	FP	Coverage	Efficiency	EQR width
Room 1 ∧ Room 2							
QR	100.00	0.00	0.00	0.00	90.91	0.2832	0.2547
CQR	100.00	0.00	0.00	0.00	90.91	0.2795	
CB-CQR	100.00	0.00	0.00	0.00	90.79	0.2945	
kNN-CQR	100.00	0.00	0.00	0.00	90.81	0.2824	
MIN	69.25	30.75	0.00	0.00	95.40	0.2585	
UNION	–	–	–	–	94.88	0.2890	
CB-MIN	69.25	30.75	0.00	0.00	97.38	0.3117	
CB-UNION	–	–	–	–	98.93	0.5403	
kNN-MIN	59.88	21.37	18.75	18.75	90.14	0.2248	
kNN-UNION	–	–	–	–	93.73	0.2611	
Room 1 ∧ Room 3							
QR	69.50	30.38	0.12	0.00	88.61	0.2083	0.1920
CQR	69.63	30.38	0.00	0.00	88.61	0.2438	
CB-CQR	69.63	30.37	0.00	0.00	96.03	0.3038	
kNN-CQR	64.25	23.87	11.87	11.75	87.49	0.2183	
MIN	60.13	8.25	31.62	31.62	90.34	0.2093	
UNION	–	–	–	–	94.19	0.2717	
CB-MIN	69.68	30.37	0.00	0.00	93.56	0.2875	
CB-UNION	–	–	–	–	97.31	0.5283	
kNN-MIN	55.00	16.38	28.62	28.62	88.36	0.2094	
kNN-UNION	–	–	–	–	91.64	0.2388	
Room 1 ∧ Room 4							
QR	100.00	0.00	0.00	0.00	89.28	0.2590	0.2640
CQR	100.00	0.00	0.00	0.00	89.28	0.2646	
CB-CQR	2.00	98.00	0.00	0.00	94.04	0.4228	
kNN-CQR	100.0	0.00	0.00	0.00	89.41	0.2636	
MIN	100.00	0.00	0.00	0.00	90.01	0.2645	
UNION	–	–	–	–	90.05	0.2816	
CB-MIN	0.75	99.25	0.00	0.00	94.03	0.4253	
CB-UNION	–	–	–	–	94.04	0.5968	
kNN-MIN	100.00	0.00	0.00	0.00	89.42	0.2637	
kNN-UNION	–	–	–	–	89.46	0.2810	
Room 1 ∧ Room 5							
QR	100.00	0.00	0.00	0.00	88.14	0.2524	0.2364
CQR	100.00	0.00	0.00	0.00	88.14	0.2540	
CB-CQR	0.00	100.00	0.00	0.00	97.98	inf	
kNN-CQR	100.00	0.00	0.00	0.00	87.77	0.2526	
MIN	69.00	31.00	0.00	0.00	91.43	0.2143	
UNION	–	–	–	–	93.91	0.2468	
CB-MIN	69.00	31.00	0.00	0.00	94.31	0.2880	
CB-UNION	–	–	–	–	96.82	0.4704	
kNN-MIN	57.75	19.13	23.12	23.12	90.18	0.2127	
kNN-UNION	–	–	–	–	92.87	0.2402	

Table E.8c
MRH8: caption as in E.8b.

MRH8	Correct	Uncertain	Wrong	FP	Coverage	Efficiency	EQR width
Room 1 ∧ Room 6							
QR	69.75	28.88	1.37	1.37	88.21	0.2109	0.1964
CQR	69.00	31.00	0.00	0.00	88.21	0.2492	
CB-CQR	69.00	31.00	0.00	0.00	95.84	0.3049	
kNN-CQR	65.38	24.12	10.50	10.50	89.40	0.2291	
MIN	69.00	31.00	0.00	0.00	91.43	0.2143	
UNION	–	–	–	–	93.91	0.2468	
CB-MIN	69.00	31.00	0.00	0.00	94.31	0.2880	
CB-UNION	–	–	–	–	96.82	0.4704	
kNN-MIN	57.75	19.13	23.12	23.12	90.18	0.2127	
kNN-UNION	–	–	–	–	92.87	0.2402	
Room 1 ∧ Room 7							
QR	69.38	30.25	0.37	0.37	86.67	0.2078	0.1960
CQR	69.00	31.00	0.00	0.00	86.67	0.2452	
CB-CQR	68.88	31.00	0.12	0.00	95.58	0.3033	
kNN-CQR	62.13	27.62	10.25	10.25	90.21	0.2290	
MIN	67.88	17.87	14.25	14.25	90.51	0.2109	
UNION	–	–	–	–	92.33	0.2283	
CB-MIN	69.00	31.00	0.00	0.00	93.57	0.2884	
CB-UNION	–	–	–	–	94.84	0.4400	
kNN-MIN	57.38	16.75	25.87	25.87	89.22	0.2118	
kNN-UNION	–	–	–	–	90.92	0.2218	
Room 1 ∧ Room 8							
QR	99.63	0.12	0.25	0.00	89.63	0.2283	0.2203
CQR	99.75	0.12	0.12	0.00	89.63	0.2339	
CB-CQR	99.88	0.00	0.12	0.00	90.55	0.2353	
kNN-CQR	99.63	0.25	0.12	0.00	89.65	0.2328	
MIN	100.00	0.00	0.00	0.00	90.52	0.2264	
UNION	–	–	–	–	94.69	0.2491	
CB-MIN	99.88	0.00	0.12	0.00	90.60	0.2307	
CB-UNION	–	–	–	–	95.65	0.3417	
kNN-MIN	99.88	0.125	0.00	0.00	90.08	0.2269	
kNN-UNION	–	–	–	–	94.24	0.2473	

References

[1] L. Bortolussi, F. Cairoli, N. Paoletti, S.A. Smolka, S.D. Stoller, Neural predictive monitoring, in: International Conference on Runtime Verification, Springer, 2019, pp. 129–147.

[2] L. Bortolussi, F. Cairoli, N. Paoletti, S.A. Smolka, S.D. Stoller, Neural predictive monitoring and a comparison of frequentist and bayesian approaches, Int. J. Softw. Tools Technol. Transf. 23 (4) (2021) 615–640.

[3] F. Cairoli, L. Bortolussi, N. Paoletti, Neural predictive monitoring under partial observability, in: Runtime Verification: 21st International Conference, RV 2021, Virtual Event, October 2021 11–14, Proceedings 21, Springer, 2021, pp. 121–141.

[4] E. Bartocci, J. Deshmukh, A. Donzé, G. Fainekos, O. Maler, D. Ničković, S. Sankaranarayanan, Specification-based monitoring of cyber–physical systems: a survey on theory, tools and applications, in: Lectures on Runtime Verification, Springer, 2018, pp. 135–175.

[5] D.T. Phan, R. Grosu, N. Jansen, N. Paoletti, S.A. Smolka, S.D. Stoller, Neural simplex architecture, in: NASA Formal Methods Symposium, Springer, 2020, pp. 97–114.

[6] H.L. Younes, M. Kwiatkowska, G. Norman, D. Parker, Numerical vs. statistical probabilistic model checking, Int. J. Softw. Tools Technol. Transf. 8 (3) (2006) 216–228.

[7] X. Qin, J.V. Deshmukh, Predictive monitoring for signal temporal logic with probabilistic guarantees, in: Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control, ACM, 2019, pp. 266–267.

[8] E. Yel, T.J. Carpenter, C. Di Franco, R. Ivanov, Y. Kantaros, I. Lee, J. Weimer, N. Bezzo, Assured runtime monitoring and planning: Toward verification of neural networks for safe autonomous operations, IEEE Robot. Autom. Mag. 27 (2) (2020) 102–116.

[9] Y. Chou, H. Yoon, S. Sankaranarayanan, Predictive runtime monitoring of vehicle models using bayesian estimation and reachability analysis, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2020, pp. 2111–2118.

[10] M. Ma, J. Stankovic, E. Bartocci, L. Feng, Predictive monitoring with logic-calibrated uncertainty for cyber–physical systems, ACM Trans. Embed. Comput. Syst. (TECS) 20 (5s) (2021) 1–25.

[11] L. Bortolussi, F. Cairoli, G. Carbone, P. Pulcini, Scalable stochastic parametric verification with stochastic variational smoothed model checking, in: International Conference on Runtime Verification, Springer, 2023, pp. 45–65.

[12] F. Cairoli, L. Bortolussi, N. Paoletti, Learning-based approaches to predictive monitoring with conformal statistical guarantees, in: International Conference on Runtime Verification, Springer, 2023, pp. 461–487.

[13] V. Balasubramanian, S.-S. Ho, V. Vovk, Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications, Newnes, 2014.

[14] V. Vovk, A. Gammerman, G. Shafer, Algorithmic Learning in a Random World, Springer Science & Business Media, 2005.

[15] F. Cairoli, N. Paoletti, L. Bortolussi, Conformal quantitative predictive monitoring of stl requirements for stochastic processes, in: Proceedings of the 26th ACM International Conference on Hybrid Systems: Computation and Control, 2023, pp. 1–11.

[16] A. Donzé, O. Maler, Robust satisfaction of temporal logic over real-valued signals, in: Proceedings of International Conference on Formal Modeling and Analysis of Timed Systems, Springer, Klosterneuburg, Austria, 2010, pp. 92–106, http://dx.doi.org/10.1007/978-3-642-15297-9_9.

[17] V. Raman, A. Donzé, M. Maasoumy, R.M. Murray, A. Sangiovanni-Vincentelli, S.A. Seshia, Model predictive control with signal temporal logic specifications, in: Proceedings of the IEEE Conference on Decision and Control, IEEE, Los Angeles, USA, 2014, pp. 81–87, http://dx.doi.org/10.1109/CDC.2014.7039363.
[18] A. Majumdar, M. Pavone, How should a robot assess risk? towards an axiomatic theory of risk in robotics, in: Robotics Research, Springer, 2020, pp. 75–84.
[19] L. Lindemann, A. Rodionova, G. Pappas, Temporal robustness of stochastic signals, in: 25th ACM International Conference on Hybrid Systems: Computation and Control, 2022, pp. 1–11.
[20] Y. Romano, E. Patterson, E.J. Candès, Conformalized quantile regression, 2019, arXiv preprint arXiv:1905.03222.
[21] L. Guan, Conformal prediction with localization, 2019, arXiv preprint arXiv:1908.08558.
[22] L. Guan, Localized conformal prediction: A generalized inference framework for conformal prediction, Biometrika 110 (1) (2023) 33–50.
[23] A. Rodionova, L. Lindemann, M. Morari, G.J. Pappas, Time-robust control for stl specifications, in: 2021 60th IEEE Conference on Decision and Control, CDC, IEEE, 2021, pp. 572–579.
[24] O. Maler, D. Nickovic, Monitoring temporal properties of continuous signals, in: Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, Springer, 2004, pp. 152–166.
[25] H. Chen, S. Lin, S.A. Smolka, N. Paoletti, An STL-based formulation of resilience in cyber–physical systems, in: International Conference on Formal Modeling and Analysis of Timed Systems, Springer, 2022, pp. 117–135.
[26] H. Papadopoulos, Inductive conformal prediction: Theory and application to neural networks, in: Tools in Artificial Intelligence, vol. 18, InTech, 2008, pp. 315–330.
[27] A.N. Angelopoulos, S. Bates, A gentle introduction to conformal prediction and distribution-free uncertainty quantification, 2021, arXiv preprint arXiv:2107.07511.
[28] V. Vovk, Conditional validity of inductive conformal predictors, in: Asian Conference on Machine Learning, PMLR, 2012, pp. 475–490.
[29] J. Lei, L. Wasserman, Distribution-free prediction bands for non-parametric regression, J. R. Stat. Soc. Ser. B Stat. Methodol. 76 (1) (2014) 71–96.
[30] A. Abate, H. Blom, M. Bouissou, N. Cauchi, H. Chraibi, J. Delicaris, S. Haesaert, A. Hartmanns, M. Khaled, A. Lavaei, et al., Arch-comp21 category report: Stochastic models, in: 8th International Workshop on Applied Verification of Continuous and Hybrid Systems, ARCH 2021, EasyChair, 2021, pp. 55–89.
[31] A. Abate, J.-P. Katoen, J. Lygeros, M. Prandini, Approximate model checking of stochastic hybrid systems, Eur. J. Control 16 (6) (2010) 624–641.
[32] A. Fehnker, F. Ivančić, Benchmarks for hybrid systems verification, in: International Workshop on Hybrid Systems: Computation and Control, Springer, 2004, pp. 326–341.
[33] R. Malhame, C.-Y. Chong, Electric load model synthesis by diffusion approximation of a high-order hybrid-state stochastic system, IEEE Trans. Autom. Control 30 (9) (1985) 854–860.
[34] E. Bartocci, L. Bortolussi, L. Nenzi, G. Sanguinetti, System design of stochastic models using robustness of temporal properties, Theoret. Comput. Sci. 587 (2015) 3–25.
[35] P. Kurasov, A. Lück, D. Mugnolo, V. Wolf, Stochastic hybrid models of gene regulatory networks–a pde approach, Math. Biosci. 305 (2018) 170–177.
[36] J. Terry, B. Black, N. Grammel, M. Jayakumar, A. Hari, R. Sullivan, L.S. Santos, C. Dieffendahl, C. Horsch, R. Perez-Vicente, et al., Pettingzoo: Gym for multi-agent reinforcement learning, Adv. Neural Inf. Process. Syst. 34 (2021) 15032–15043.
[37] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M.A. Riedmiller, Playing atari with deep reinforcement learning, 2013, CoRR abs/1312.5602. arXiv:1312.5602.
[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, 2019, pp. 8024–8035.
[39] D. Phan, N. Paoletti, T. Zhang, R. Grosu, S.A. Smolka, S.D. Stoller, Neural state classification for hybrid systems, in: International Symposium on Automated Technology for Verification and Analysis, Springer, 2018, pp. 422–440.
[40] F. Cairoli, N. Paoletti, L. Bortolussi, Neural predictive monitoring for collective adaptive systems, in: Leveraging Applications of Formal Methods, Verification and Validation. Adaptation and Learning: 11th International Symposium, ISoLA 2022, Rhodes, Greece, October 22–30, 2022 Proceedings, Part III, Springer-Verlag, 2022, pp. 30–46.
[41] L. Bortolussi, D. Milios, G. Sanguinetti, Smoothed model checking for uncertain continuous-time Markov chains, Inform. and Comput. 247 (2016) 235–253.
[42] L. Bortolussi, F. Cairoli, G. Carbone, P. Pulcini, Stochastic variational smoothed model checking, 2022, arXiv preprint arXiv:2205.05398.
[43] B. Djeridane, J. Lygeros, Neural approximation of PDE solutions: An application to reachability computations, in: Proceedings of the 45th IEEE Conference on Decision and Control, IEEE, 2006, pp. 3034–3039.
[44] V.R. Royo, D. Fridovich-Keil, S. Herbert, C.J. Tomlin, Classification-based approximate reachability with guarantees applied to safe trajectory tracking, 2018, arXiv preprint arXiv:1803.03237.
[45] W. Granig, S. Jakšić, H. Lewitschnig, C. Mateis, D. Ničković, Weakness monitors for fail-aware systems, in: International Conference on Formal Modeling and Analysis of Timed Systems, Springer, 2020, pp. 283–299.
[46] R. Ivanov, J. Weimer, R. Alur, G.J. Pappas, I. Lee, Verisig: verifying safety properties of hybrid systems with neural network controllers, in: Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control, 2019, pp. 169–178.
[47] X. Qin, J.V. Deshmukh, Clairvoyant monitoring for signal temporal logic, in: International Conference on Formal Modeling and Analysis of Timed Systems, Springer, 2020, pp. 178–195.
[48] H. Yoon, S. Sankaranarayanan, Predictive runtime monitoring for mobile robots using logic-based bayesian intent inference, in: 2021 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2021, pp. 8565–8571.
[49] X. Yu, W. Dong, X. Yin, S. Li, Model predictive monitoring of dynamic systems for signal temporal logic specifications, 2022, arXiv preprint arXiv:2209.12493.
[50] V. Kuleshov, N. Fenner, S. Ermon, Accurate uncertainties for deep learning using calibrated regression, in: International Conference on Machine Learning, PMLR, 2018, pp. 2796–2804.
[51] X. Han, Z. Tang, J. Ghosh, Q. Liu, Split localized conformal prediction, 2022, arXiv preprint arXiv:2206.13092.
[52] L. Lindemann, X. Qin, J.V. Deshmukh, G.J. Pappas, Conformal prediction for stl runtime verification, in: Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023), 2023, pp. 142–153.
[53] Y. Zhao, B. Hoxha, G. Fainekos, J.V. Deshmukh, L. Lindemann, Robust conformal prediction for stl runtime verification under distribution shift, in: 2024 ACM/IEEE 15th International Conference on Cyber-Physical Systems, ICCPS, IEEE, 2024, pp. 169–179.
[54] L. Lindemann, M. Cleaveland, G. Shim, G.J. Pappas, Safe planning in dynamic environments using conformal prediction, IEEE Robot. Autom. Lett. (2023).
[55] E.E. Vlahakis, L. Lindemann, P. Sopasakis, D.V. Dimarogonas, Probabilistic tube-based control synthesis of stochastic multi-agent systems under signal temporal logic, 2024, arXiv preprint arXiv:2405.02827.
[56] L. Lindemann, Y. Zhao, X. Yu, G.J. Pappas, J.V. Deshmukh, Formal verification and control with conformal prediction, 2024, arXiv preprint arXiv:2409.00536.