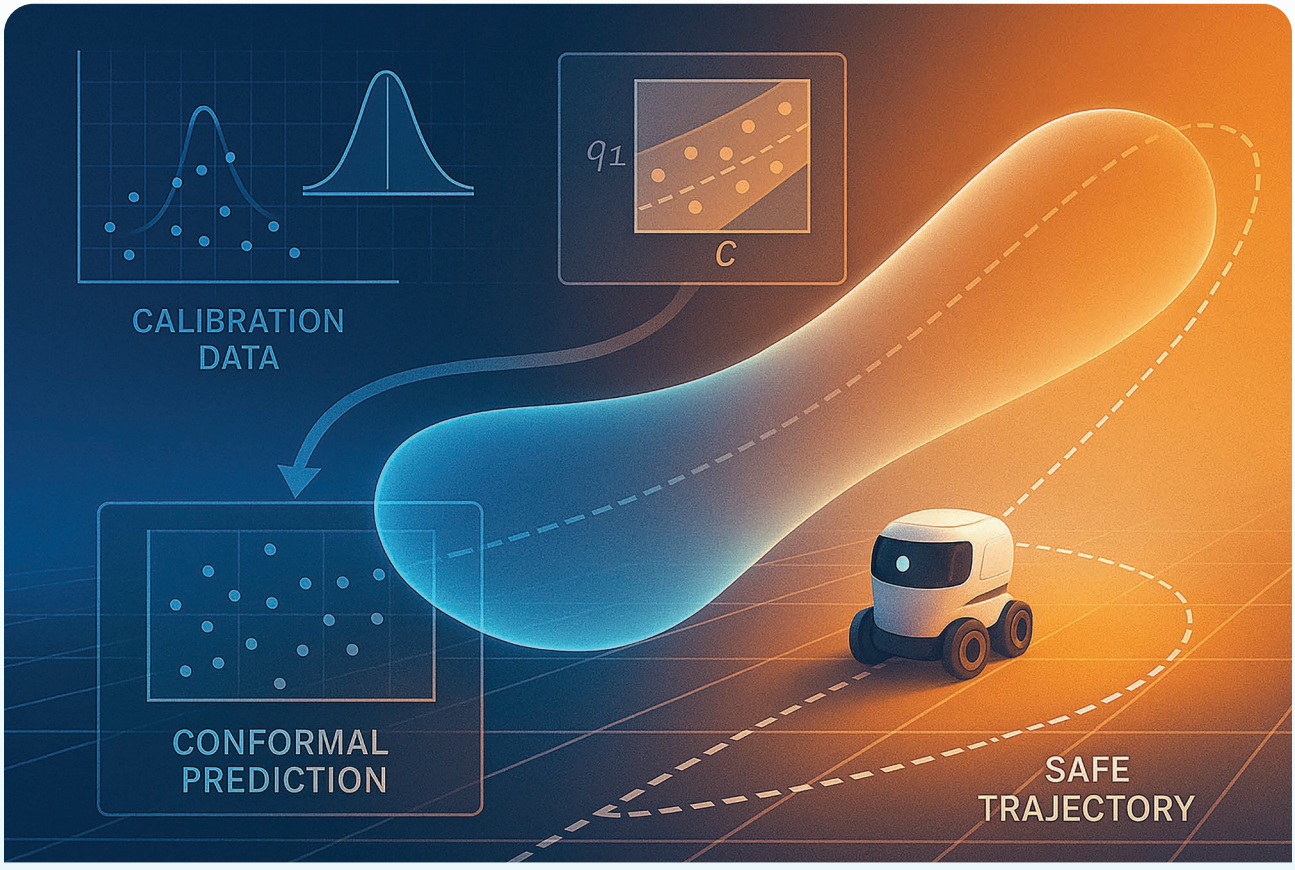
Formal Verification and Control With Conformal Prediction

PRACTICAL SAFETY GUARANTEES FOR AUTONOMOUS SYSTEMS

LARS LINDEMANN, YIQI ZHAO, XINYI YU, GEORGE J. PAPPAS, and JYOTIRMOY V. DESHMUKH



Confidence meets control — verifiably safe systems with conformal prediction.

INTRODUCTION

The design of autonomous systems, which are becoming increasingly learning enabled, has attracted much attention within the research community. Research in this area promises to enable many future technologies, such as autonomous driving, intelligent transportation, and robotics. In recent years, great progress has been made in the design of learning-enabled components (LECs), for example, with neural networks for perception tasks, such as object detection [1], [2], localization and state estimation [3], [4], and trajectory prediction [5], [6], [7]; for decision-making tasks, such as motion and behavior planning [8], [9]; and for low-level control [10], [11], [12]. However, the integration of LECs into safety-critical autonomous systems is limited by their fragility and can

Digital Object Identifier 10.1109/MCS.2025.3611545
Date of current version: 20 November 2025

1066-033X © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies.
Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

result in unsafe behavior, for example, inaccurate and nonrobust object detectors in self-driving cars. The fragility of LECs is the result of highly nonconvex learning problems, distribution shifts from training to the deployment domain, and a lack of model robustness [13], [14]. Unfortunately, these safety challenges are further amplified by the complexity of modern autonomous systems that operate in uncertain and dynamic environments, where traditional approaches for localization and mapping may fail to provide guarantees, for example, simultaneous localization and mapping techniques [4], [15] or Kalman/particle filters [16], [17], [18].

For these reasons, we require verifiable frameworks for the integration of LECs into the perception–action loops of autonomous systems, ultimately resulting in safe learning-enabled autonomous systems (LEASs). Toward this goal, this survey focuses on designing *formal verification and control algorithms for LEASs with practical safety guarantees using conformal prediction* (see "Summary"). By practical, we mean that these algorithms 1) are applicable and scale to complex LEASs, 2) come with formal guarantees, and 3) are easy to understand and extend, even for a novice in the field.

Conformal prediction is a lightweight statistical tool for uncertainty quantification, originally introduced by Shafer and Vovk in [19] and [20], and has recently attracted attention within the machine learning community; for example, see [21] and [22] for up-to-date tutorials. In this survey, we use conformal prediction for designing safe controllers, for integrating online monitors into the perception–action loops of LEASs, and for performing offline verification (see Figure 1).

The main objective behind conformal prediction and uncertainty quantification can be summarized as follows.

Uncertainty quantification with conformal prediction: Given $K+1$ independent and identically distributed (or exchangeable) random variables $R^{(0)}, R^{(1)}, \ldots, R^{(K)}$, conformal prediction aims to quantify the uncertainty of $R^{(0)}$ based on $R^{(1)}, \ldots, R^{(K)}$. Formally, given a failure probability $\delta \in (0,1)$, we want to construct a probabilistically valid prediction region $C : \mathbb{R}^K \to \mathbb{R}$ so that

$$\text{Prob}\,(R^{(0)} \leq C(R^{(1)}, \ldots, R^{(K)})) \geq 1-\delta. \tag{1}$$

The variable $R^{(i)}$ for $i \in \{0, \ldots, K\}$ is usually referred to as the nonconformity score. In regression, it may be defined as the prediction error $R^{(i)} := \| y^{(i)} - \mu(u^{(i)}) \|$, where a predictor μ attempts to predict an output $y^{(i)}$ based on an input $u^{(i)}$. A small (large) nonconformity score thus indicates a good (poor) predictive model. A larger part of our discussion will be on constructing informative nonconformity scores for verification and control. In practice, with (1) we aim to find a probabilistic upper bound of a test datapoint $R^{(0)}$ from a calibration dataset $R^{(1)}, \ldots, R^{(K)}$. Conformal prediction provides a simple and efficient procedure to compute an upper bound $C(R^{(1)}, \ldots, R^{(K)})$ that is valid with a confidence of $1-\delta$. For simplicity, we will often simply use C in the remainder and omit the function arguments $R^{(1)}, \ldots, R^{(K)}$.

Goals

In this survey, we leverage the conformal prediction guarantees in (1) to achieve the following four objectives:

1) designing verification algorithms for LECs to reason about input–output properties of LECs
2) designing control algorithms for LEASs that come with probabilistic safety guarantees
3) designing offline verification algorithms for LEASs to verify safety requirements of LEASs
4) designing online verification algorithms for LEASs that predict system failures with high confidence.

Before starting with the main parts of this survey article, we refer the reader to "Spectrum of Formal Verification and Control Techniques," in which we formally distinguish

Summary

We present recent advances in formal verification and control for autonomous systems with practical safety guarantees enabled by conformal prediction, a statistical tool for uncertainty quantification. This survey is particularly motivated by LEASs, where the complexity of LECs poses a major bottleneck for applying traditional model-based verification and control techniques. To address this challenge, we advocate for conformal prediction as a lightweight alternative and demonstrate its use in formal verification, systems and control, and robotics. Conformal prediction is appealing because of its simplicity (it is easy to understand, implement, and adapt), generality (it requires no assumptions on learned models and underlying data distributions), and efficiency (it is real-time capable and accurate).

This survey provides an accessible introduction to conformal prediction for nonexperts interested in applying it to autonomy problems. We particularly show how conformal prediction can be used for formal verification of LECs and the design of safe control as well as offline and online verification algorithms for LEASs. We present these techniques within a unifying framework that addresses the complexity of LEASs. Our exposition spans simple specifications, such as robot navigation tasks, to complex mission requirements expressed in temporal logic. Throughout the survey, we contrast conformal prediction with other statistical techniques, including scenario optimization and probably approximately correct-Bayes theory, highlighting advantages and limitations for verification and control. Finally, we outline open problems and promising directions for future research.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

offline and online approaches to verification and control and discuss computational bottlenecks.

Related Survey Articles

Survey articles on the topic of conformal prediction were presented in [19], [21], and [22]. However, the material in these articles is presented through the lens of a statistician and without the reference to the use of conformal prediction in formal verification and control that we use.

There exist various survey articles at the intersection of machine learning, formal verification, and control that are related to the topic of this survey. The authors of [33] survey

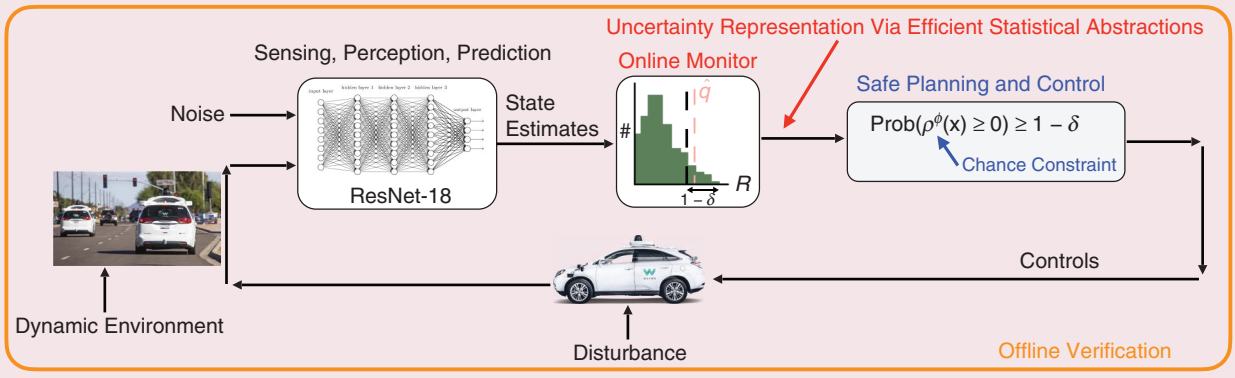


FIGURE 1 Formal verification and control of LEASs. In this survey, we show how to design offline verification algorithms and how to integrate online monitors obtained via efficient statistical abstractions into the control design.

Spectrum of Formal Verification and Control Techniques

The control community distinguishes between open-loop and closed-loop controllers. An open-loop controller computes a sequence of control inputs offline before the system is deployed. On the other hand, a closed-loop (or feedback) controller computes control inputs from online information while the system is operating. Open-loop controllers are usually not used for systems that require the controller to be robust against uncertainty and noise. Instead, a combination of offline (often referred to as planning) and online control is used [S1], [S2], [S3]. At each control level, the designer has to carefully trade off the *computational complexity* of the control problem to be solved with the *model fidelity* (abstract versus concrete physical models) as well as the *task complexity* (high-level specifications versus motion primitives). In safety-critical control, another tradeoff arises between the type of *formal safety guarantee* one is interested in (for example, deterministic, probabilistic, or robust) and the model fidelity as well as task complexity. Another distinction can be made between model-based control design, for example, model predictive control [23], [24], [25] or robust control [S4], and data-driven control techniques, such as data-driven model predictive control [11], [S5] or scenario optimization [26], [27]. In many ways, however, these lines are (and have always been) blurred. See [S6] and [S7] for more references and a detailed discussion.

Although it is less familiar to the control community, but well known within the formal methods community, and one can make similar analogies in verification. Broadly speaking, verification aims at checking the correctness of a system against

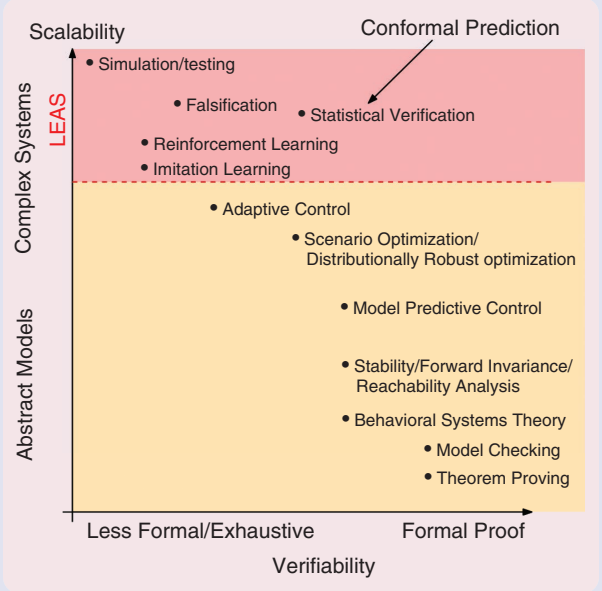


FIGURE S1 Overview of selected formal verification and control techniques.

a system specification expressed in a mathematical logic, for example, in predicate or temporal logics (see "LEC Specifications in Predicate Logic" and "LEAS Specifications in STL"). Offline verification (which we simply refer to as verification in the remainder) aims at verifying the system as a whole by exhaustively checking all possible system behaviors. For this

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

statistical methods for the system identification of linear systems with finite sample guarantees. Utilizing such system identification methods, the same authors present a survey article on statistical learning theory for control design in [34]. In contrast to [33] and [34], however, our focus is more broadly on formal verification and control design of complex LEASs. This includes reasoning over LECs, for example, for perception and prediction and dynamic and uncertain environments, and it requires the use of formal specification languages and design techniques. In this regard, we point the reader to the brief survey in [35], in which the need for formal methods in verifiable autonomy is stressed and grand challenges are discussed. As in [33] and [34], we are interested in obtaining finite sample guarantees, but here by using conformal prediction, which can provide both marginal and calibration conditional coverage guarantees (details follow in the next section). Also related to our article is the survey in [36], which reviews learning-based Hamilton–Jacobi reachability techniques for safe control design. By contrast, here we are concerned about dealing with the complexity of LECs within LEASs and the associated verification and control challenges. The authors of [37] provided a detailed survey of formal control synthesis techniques for safety-critical autonomous systems, however, without a focus on using statistical methods for LEAS as we do. The survey article in [38] focuses on learning-enabled predictive online verification techniques using conformal prediction, similar to the section "Online Verification of LEASs with Conformal Prediction" in this survey. Our focus is again broader in scope, considering offline and online formal verification as well as control design techniques for LEASs. Additionally, the online predictive verification techniques presented here are different in flavor than the ones surveyed in [38] (the details are in the section "Online Verification of LEASs with Conformal Prediction"). Finally, we mention [39] and [40], which are survey articles on simulation-based approaches for autonomous system verification. We note that [39] provides a short survey advocating for a data-centric approach as we do, but that we provide an in-depth treatment of the topic with a strong focus on techniques and algorithms, along with providing formal guarantees via the use of statistical tools. In contrast to [40], which is similar in scope to what we present, the reader will find a different and complementary set of techniques and

purpose, automated verification tools were developed, such as model checking [S8], [S9] or theorem proving [S10], [S11]. Motivated by the computational challenges that these methods face, especially when applied to LEASs, statistical verification techniques were proposed as an efficient alternative that allows checking a $1-\delta$ fraction of the system behaviors [S12], [254]. In online verification, by contrast, one has already observed a partial execution of the system, for example, a partial trajectory of a car navigating traffic, and our task is to check if all possible future behaviors satisfy the specification [28], [29], [30]. Online verification is often also referred to as runtime verification. In online verification, the verification answer can be inconclusive, for example, neither satisfied nor violated. Predictive online verification algorithms instead predict future system behaviors and can thus provide verification answers more reliably and quickly [31], [32]. Ultimately, as for control design, a combination of offline and online verification should be used in practice.

In Figure S1, we present an incomplete (and potentially biased) set of verification and control techniques. We list these techniques according to their scalability and verifiability properties. We specifically delineate between techniques that can (or have been) applied to LEASs. Conformal prediction falls into this category.

REFERENCES

[S1] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas, "Symbolic planning and control of robot motion [Grand Challenges of Robotics]," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 61–70, Mar. 2007, doi: 10.1109/MRA.2007.339624.

[S2] N. Matni, A. D. Ames, and J. C. Doyle, "A quantitative framework for layered multirate control: Toward a theory of control architecture," *IEEE Control Syst. Mag.*, vol. 44, no. 3, pp. 52–94, Jun. 2024, doi: 10.1109/MCS.2024.3382388.

[S3] J.-P. Laumond, *Robot Motion Planning and Control*, vol. 229. Berlin, Heidelberg, Germany: Springer-Verlag, 1998.

[S4] K. Zhou and J. C. Doyle, *Essentials of Robust Control*. Upper Saddle River, NJ, USA: Prentice Hall, 1998.

[S5] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Trans. Autom. Control*, vol. 66, no. 4, pp. 1702–1717, Apr. 2021, doi: 10.1109/TAC.2020.3000182.

[S6] F. Dörfler, "Data-driven control: Part one of two: A special issue sampling from a vast and dynamic landscape," *IEEE Control Syst. Mag.*, vol. 43, no. 5, pp. 24–27, Oct. 2023, doi: 10.1109/MCS.2023.3291624.

[S7] F. Dörfler, "Data-driven control: Part two of two: Hot take: Why not go with models?" *IEEE Control Syst. Mag.*, vol. 43, no. 6, pp. 27–31, Dec. 2023, doi: 10.1109/MCS.2023.3310302.

[S8] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen, "Model checking continuous-time Markov chains by transient analysis," in *Proc. Int. Conf. Comput. Aided Verification*, Berlin, Heidelberg, Germany: Springer-Verlag, 2000, pp. 358–372.

[S9] E. M. Clarke, "Model checking," in *Proc. Int. Conf. Found. Softw. Technol. Theor. Comput. Sci.*, Berlin, Heidelberg, Germany: Springer-Verlag, 1997, pp. 54–56.

[S10] Y. Shoukry, P. Nuzzo, A. L. Sangiovanni-Vincentelli, S. A. Seshia, G. J. Pappas, and P. Tabuada, "SMC: Satisfiability modulo convex optimization," in *Proc. 20th Int. Conf. Hybrid Syst.: Comput. Control (HSCC)*, 2017, pp. 19–28.

[S11] M. Sheeran, S. Singh, and G. Stålmarck, "Checking safety properties using induction and a sat-solver," in *Proc. Int. Conf. Formal Methods Comput.-Aided Des.*, Berlin, Heidelberg, Germany: Springer-Verlag, 2000, pp. 127–144.

[S12] H. L. S. Younes and R. G. Simmons, "Statistical probabilistic model checking with a focus on time-bounded properties," *Inf. Comput.*, vol. 204, no. 9, pp. 1368–1409, 2006, doi: 10.1016/j.ic.2006.05.002.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

Empirical Quantiles as Linear Programs

Following [S13], we can overapproximate the empirical quantile of the nonconformity scores $R^{(1)}, \ldots, R^{(K)}$ at level $1-\delta$ by the solution of the following linear program:

$$q^* := \operatorname*{argmin}_{q} \ \sum_{i=1}^{K} \left((1-\delta)e_i^+ + \delta e_i^-\right) \tag{S1a}$$

$$\text{subject to} \quad e_i^+ - e_i^- = R^{(i)} - q, i = 1, \ldots, K \tag{S1b}$$

$$e_i^+, e_i^- \geq 0, i = 1, \ldots, K. \tag{S1c}$$

The intuition for (S1) is as follows. The variables e_i^+ and e_i^- together with the constraints (S1b) and (S1c) represent deviations of $R^{(i)}$ from the variable q from above and below. The objective function (S1a) penalizes points above q with weight $1-\delta$ and points below q with weight δ. Minimizing this weighted sum of deviations forces the solution to "push" just enough nonconformity scores $R^{(i)}$ above and below q to enforce that $\text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}) \leq q^*$.

From Lemma 1 and (2), we recall next that ∞ has to appear in the empirical quantile $\text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, \infty)$. However, in practice, we cannot encode ∞ in the linear program (S1). Nonetheless, note that it holds that

$$\underset{1-\delta}{\text{Quantile}}(R^{(1)}, \ldots, R^{(K)}, \infty) = \underset{(1+1/K)(1-\delta)}{\text{Quantile}}(R^{(1)}, \ldots, R^{(K)})$$

if $(1+1/K)(1-\delta) \in (0,1)$ so that we can simply replace $1-\delta$ by $(1+1/K)(1-\delta)$ in (S1).[a] If we do exactly this, we obtain $\text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, \infty) \leq q^*$. Indeed, one can show that equality, that is, $\text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, \infty) = q^*$, holds if $(1-\delta)K \notin \mathbb{N}$; see [45] for a proof.

This linear program naturally lends itself to an optimization approach when we would like to optimize the empirical quantile (and thus the upper bound C) over a parameter, for example, when we would like to minimize $\text{Quantile}_{1-\delta}(R_a^{(1)}, \ldots, R_a^{(K)}, \infty)$ over a design parameter a that the nonconformity scores $R_a^{(1)}, \ldots, R_a^{(K)}$ depend on. A practical example would be when we aim to minimize the size of prediction regions; see the section "Control Synthesis for LEASs With Conformal Prediction."

REFERENCE

[S13] R. Koenker and G. Bassett Jr, "Regression quantiles," *Econometrica: J. Econometric Soc.*, vol. 46, no. 1, pp. 33–50, 1978, doi: 10.2307/1913643.

[a]Alternatively, note that $\text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, M) = \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, \infty)$ for a sufficiently large constant $M \geq \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, \infty)$ so that we could instead use M in the linear program (S1).

algorithms in our survey article, for example, conformal prediction guarantees and compositional verification and control (recall Figure 1).

Unlike the coverage guarantees in (1) that we pursue here, risk metrics have appeared as an alternative to capture distinct properties of a random variable, for example, the tail behavior of a random variable that encodes the loss of a system. The use of risk metrics in robotics was advocated for in [41], and their use in verification and control was surveyed in [42] and [43].

CONFORMAL PREDICTION IN A NUTSHELL

Let $R^{(0)}, \ldots, R^{(K)}$ be $K+1$ independent and identically distributed random variables that follow a distribution $\mathcal{R}$, that is, those that are such that $R^{(0)}, \ldots, R^{(K)} \sim \mathcal{R}$. We note that the theory of conformal prediction also holds, and in fact was originally invented, for exchangeable random variables $R^{(0)}, \ldots, R^{(K)}$.[1] For simplicity, and as not needed in this survey article, we avoid explicitly stating the underlying probability space of $R^{(0)}, \ldots, R^{(K)}$.

Vanilla Conformal Prediction

Based on a simple quantile argument, we can compute a probabilistic prediction region as in (1); for example, we can compute an upper bound C for $R^{(0)}$ from calibration data $R^{(1)}, \ldots, R^{(K)}$ that holds with high confidence. We proceed as follows: for a given failure probability $\delta \in (0,1)$, we obtain the upper bound $C : \mathbb{R}^K \to \mathbb{R}$ as

$$C(R^{(1)}, \ldots, R^{(K)}) := \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, \infty) \tag{2}$$

which is the $(1-\delta)$th quantile [formally defined in (S2)] over the empirical distribution of the values $R^{(1)}, \ldots, R^{(K)}$ and ∞, where ∞ is added as a correction to obtain finite sample guarantees. While the definition of the $(1-\delta)$th quantile $\text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, \infty)$ in (S2) appears difficult to parse at first, it turns out that we can efficiently compute $\text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, \infty)$. Therefore, define $R^{(K+1)} := \infty$ and assume, without loss of generality, that $R^{(1)}, \ldots, R^{(K)}$ are sorted in nondecreasing order, that is, by reordering. We now obtain $C = R^{(p)}$ with $p := \lceil (K+1)(1-\delta) \rceil$, where $\lceil \cdot \rceil$ is the ceiling function; that is, C is the pth smallest nonconformity score. Effectively, the quantile computation in (2) amounts to computing the pth-order statistic of $R^{(1)}, \ldots, R^{(K)}$ and is thus related to order statistics [44]. This computation can be formed efficiently using standard sorting algorithms, which usually have $O(K \log K)$ time complexity. For our purposes, the strength of this result lies in the freedom of choosing a suitable nonconformity score—we will encounter various examples in this survey. In "Empirical Quantiles as Linear Programs," we further illustrate that the empirical quantile can be approximated by a linear program. This will be useful later in the case where the nonconformity scores are parameterized by a free parameter over which we want

[1]The random variables $R^{(0)}, \ldots, R^{(K)}$, are exchangeable if the joint distribution of $R^{(0)}, \ldots, R^{(K)}$ is equivalent to the joint distribution of $R^{(\sigma(0))}, \ldots, R^{(\sigma(K))}$ for any permutation σ on $\{0, \ldots, K\}$. It is easy to see that independent and identically distributed random variables are exchangeable, but not vice versa.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

Proof: Conformal Prediction

We define the $(1-\delta)$th quantile of a random variable Z as

$$\underset{1-\delta}{\text{Quantile}}(Z) := \inf\{z \in \mathbb{R} \,|\, \text{Prob}(Z \le z) \ge 1-\delta\}.$$

We then define the $(1-\delta)$th quantile of $R^{(0)}, R^{(1)}, \ldots, R^{(K)}$ via the random variable $Z_0 \sim \sum_{i=0}^{K} \delta_{R^{(i)}}/(K+1)$, where $\delta_{R^{(i)}}$ is the Dirac distribution centered at $R^{(i)}$. With a slight abuse of notation, let hence the $(1-\delta)$th quantile of $R^{(0)}, R^{(1)}, \ldots, R^{(K)}$ be

$$\underset{1-\delta}{\text{Quantile}}(R^{(0)}, R^{(1)}, \ldots, R^{(K)}) := \underset{1-\delta}{\text{Quantile}}(Z_0).$$

Similarly, let the $(1-\delta)$th quantile of $R^{(1)}, \ldots, R^{(K)}, \infty$ be

$$\underset{1-\delta}{\text{Quantile}}(R^{(1)}, \ldots, R^{(K)}, \infty) := \underset{1-\delta}{\text{Quantile}}(Z_\infty) \tag{S2}$$

with $Z_\infty \sim (\sum_{i=1}^{K} \delta_{R^{(i)}} + \delta_\infty)/(K+1)$.

FACT 1

Following these definitions, note the following. If

$$R^{(0)} > \underset{1-\delta}{\text{Quantile}}(R^{(0)}, R^{(1)}, \ldots, R^{(K)}) =: q$$

then we can change $R^{(0)}$ to arbitrary values without changing $\text{Quantile}_{1-\delta}(R^{(0)}, R^{(1)}, \ldots, R^{(K)})$ as long as $R^{(0)} > q$; that is, for $M > q$ we have that $\text{Quantile}_{1-\delta}(R^{(0)}, R^{(1)}, \ldots, R^{(K)}) = \text{Quantile}_{1-\delta}(M, R^{(1)}, \ldots, R^{(K)})$. This implies that $R^{(0)} > \text{Quantile}_{1-\delta}(R^{(0)}, R^{(1)}, \ldots, R^{(K)})$ is equivalent to $R^{(0)} > \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, \infty)$. Consequently, it holds that

$$R^{(0)} \le \underset{1-\delta}{\text{Quantile}}(R^{(1)}, \ldots, R^{(K)}, \infty) \Leftrightarrow R^{(0)} \le \underset{1-\delta}{\text{Quantile}}(R^{(0)}, R^{(1)}, \ldots, R^{(K)}).$$

FACT 2

Let z take one of the values of $R^{(0)}, R^{(1)}, \ldots, R^{(K)}$, and let the pth-order statistic correspond to this value.[a] Then, it holds that $\text{Prob}(Z_0 \le z) = p_z/(K+1)$. By definition, we know that $\text{Quantile}_{1-\delta}(Z_0)$ is equivalent to the minimum value z from $R^{(0)}, R^{(1)}, \ldots, R^{(K)}$ such that $\text{Prob}(Z_0 \le z) \ge 1-\delta$. This implies that p_z is such that $p_z/(K+1) \ge 1-\delta$, and since we also desire the minimum p_z, that $p_z = \lceil (K+1)(1-\delta) \rceil$. Hence, we know that $R^{(0)} \le \text{Quantile}_{1-\delta}(R^{(0)}, R^{(1)}, \ldots, R^{(K)})$ is equivalent to $R^{(0)}$ being among the $\lceil (K+1)(1-\delta) \rceil$th smallest values of $R^{(0)}, R^{(1)}, \ldots, R^{(K)}$.

PUTTING IT ALL TOGETHER

From Facts 1 and 2, it follows that $R^{(0)} \le \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, \infty)$ is equivalent to $R^{(0)}$ being among the $\lceil (K+1)(1-\delta) \rceil$th smallest values of $R^{(0)}, R^{(1)}, \ldots, R^{(K)}$. As $R^{(0)}, R^{(1)}, \ldots, R^{(K)}$ are independent and identically distributed, the latter (and thus the former) holds with a probability of at least $\lceil (K+1)(1-\delta) \rceil / (K+1) \ge 1-\delta$.

[a] If there are ties, let p_z be the largest possible value; for example, let $p_z := 3$ for $z := 5$ and $R^{(1)} := 1$, $R^{(2)} := R^{(3)} := 5$, $R^{(4)} := 7$.

to optimize. Let us summarize our results so far before providing more explanations and insights.

Lemma 1 (Marginal Coverage [19], [20])

Let $R^{(0)}, \ldots, R^{(K)}$ be $K+1$ independent and identically distributed random variables. Then, it holds that the probabilistic prediction region in (1) is valid by the choice of C in (2).

For the interested reader, we provide a proof of this result in "Proof: Conformal Prediction." This proof is inspired and mainly taken from [46]. We remark that it can further be shown that the probabilistic prediction region is tight; that is, we have the following:

$$\text{Prob}(R^{(0)} \le C(R^{(1)}, \ldots, R^{(K)})) \le 1 - \delta + \frac{1}{K+1}$$

under mild assumptions on $R^{(0)}, R^{(1)}, \ldots, R^{(K)}$, that is, that the joint distribution of $R^{(0)}, R^{(1)}, \ldots, R^{(K)}$ is continuous [47].

How many calibration data are needed? We may now ask how many data are needed to be able to compute a nontrivial upper bound C with (1) and (2). If $\lceil (K+1)(1-\delta) \rceil > K$, we note that C attains the value of a trivial and uninformative upper bound $C = \infty$.[2] To obtain a nontrivial upper bound, we hence require that $\lceil (K+1)(1-\delta) \rceil \le K$. From here, it is easy to see that

$$K \ge \frac{1-\delta}{\delta}$$

provides such a lower bound on the number of data required to obtain a nontrivial upper bound C.

Marginal Coverage Guarantees

We note that the guarantees in (1) are "marginal"[3] over the randomness in the test and calibration datapoints $R^{(0)}, R^{(1)}, \ldots, R^{(K)}$ as opposed to being conditional on the calibration datapoints $R^{(1)}, \ldots, R^{(K)}$. In other words, $\text{Prob}(\cdot)$ captures randomness in the draw over all random variables $R^{(0)}, R^{(1)}, \ldots, R^{(K)}$. Formally, the probability measure $\text{Prob}(\cdot)$ in (1) is a $K+1$ product measure of the probability measure associated with $\mathcal{R}$; see [48].[4]

[2] Note that computing $\text{Quantile}_{1-\delta}(R^{(0)}, \ldots, R^{(K)}, \infty)$ is equivalent to selecting the $p := \lceil (K+1)(1-\delta) \rceil$th smallest value among $R^{(1)}, \ldots, R^{(K+1)}$, where $R^{(K+1)} = \infty$. Correspondingly, if $p > K$, then the pth smallest value among $R^{(1)}, \ldots, R^{(K+1)}$ is $R^{(K+1)} = \infty$.

[3] We are using the standard terminology here from the conformal prediction community, not to be confused with a marginal distribution of a set of random variables.

[4] We choose a simplified notation here to not overcomplicate the matter and make the survey accessible to a larger readership. Specifically, we use the notation $\text{Prob}(\cdot)$ in a liberal way throughout the article, where $\text{Prob}(\cdot)$ can capture randomness over either the test data $R^{(0)} \sim \mathcal{R}$ or both test and calibration data $R^{(0)}, R^{(1)}, \ldots, R^{(K)} \sim \mathcal{R}$. The only exception to this rule is when we discuss calibration conditional guarantees, where we use $\text{Prob}(\cdot)$ to capture randomness over calibration data $R^{(1)}, \ldots, R^{(K)} \sim \mathcal{R}$.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

How can we now empirically validate marginal guarantees of the form (1) to show that the theoretical guarantees of conformal prediction indeed align with experimental observations? We do so by performing the following experiment N times: we sample one test datapoint $R^{(0)}$ and K calibration datapoints $R^{(1)}, \ldots, R^{(K)}$ from $\mathcal{R}$. For each experiment, we check if $R^{(0)} \leq C(R^{(1)}, \ldots, R^{(K)})$ is satisfied. We then compute the empirical coverage, which is the ratio between the number of times $R^{(0)} \leq C(R^{(1)}, \ldots, R^{(K)})$ is satisfied and N. As $N \to \infty$, we expect (and will observe) that the empirical coverage converges to $1-\delta$. Formally, for $R_n^{(0)}, R_n^{(1)}, \ldots, R_n^{(K)} \sim \mathcal{R}$ with $n \in \{1, \ldots, N\}$, we compute the empirical coverage as

$$EC := \sum_{n=1}^{N} \frac{\mathbb{1}(R_n^{(0)} \leq C_n)}{N} \tag{3}$$

with $C_n := C(R_n^{(1)}, \ldots, R_n^{(K)})$ denoting the nth experiment and where $\mathbb{1}(R_n^{(0)} \leq C_n)$ is the indicator function that evaluates to 1 if $R_n^{(0)} \leq C_n$ and 0 otherwise.[5]

Let us think of a practical example to get some intuition of how we can interpret the marginal guarantees in (1). Assume that we have designed a control algorithm for a robot and that $R^{(0)}, R^{(1)}, \ldots, R^{(K)}$ denote the performance of the robot over $K+1$ independent runs under the designed controller. In this sense, (1) gives us guarantees for the control algorithm that hold on average over the $K+1$ runs rather than for an individual run of the robot.[6] This is a common characteristic of statistical techniques with finite sample guarantees, and similar (yet slightly different) interpretations can be found in other statistical (learning) techniques that provide calibration conditional coverage guarantees, which we discuss next.

Calibration Conditional Coverage Guarantees

In the aforementioned robot example, we would ideally like to obtain conditional guarantees of the form $\text{Prob}(R^{(0)} \leq C(R^{(1)}, \ldots, R^{(K)}) \,|\, R^{(1)}, \ldots, R^{(K)})$ that tell us more about an individual run of the robot.[7] It is known that such conditional guarantees cannot be obtained in this setting without additional assumptions. However, it is known that this conditional probability is a random variable that follows a beta distribution centered around $1-\delta$ if the distribution of $\mathcal{R}$ is continuous, with decreasing variance as K increases. See [49] and [21, Section 3.2] for details or [50] for newer results and a summary. Formally, if the distribution of $\mathcal{R}$ is continuous, then

$$\text{Prob}(R^{(0)} \leq C(R^{(1)}, \ldots, R^{(K)}) \,|\, R^{(1)}, \ldots, R^{(K)}) \sim \text{Beta}(\beta_1, \beta_2) \tag{4}$$

where $\text{Beta}(\beta_1, \beta_2)$ is a beta distribution with parameters $\beta_1 := K+1-\lfloor (K+1)\delta \rfloor$ and $\beta_2 := \lfloor (K+1)\delta \rfloor$. This result tells us that, by increasing the size of the calibration data K, we can reduce the variance in our conditional coverage.

How can we empirically validate conditional guarantees of the form (4)? We do so by performing the following experiment N times: we sample J test datapoints $R_1^{(0)}, \ldots, R_J^{(0)}$ and K calibration datapoints $R^{(1)}, \ldots, R^{(K)}$ from $\mathcal{R}$. For each experiment, we compute the ratio between the number of times $R_1^{(0)}, \ldots, R_J^{(0)}$ satisfying $R_j^{(0)} \leq C(R^{(1)}, \ldots, R^{(K)})$ and J. We then plot the histogram of these ratios of the N experiments, which will resemble $\text{Beta}(\beta_1, \beta_2)$ as $N, J \to \infty$. Formally, for $R_{nj}^{(0)}, R_n^{(1)}, \ldots, R_n^{(K)} \sim \mathcal{R}$ with $n \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, J\}$, compute the conditional empirical coverage as

$$CEC_n := \sum_{j=1}^{J} \frac{\mathbb{1}(R_{nj}^{(0)} \leq C_n)}{J} \tag{5}$$

and plot the histogram of CEC_n over the N experiments. For a more detailed discussion of the effects of increasing/decreasing the parameters K, N, and J, we refer the reader to [21, Appendix C].

This was a lot already! In "Robot Navigation Under Sensor Uncertainty," we illustrate everything that we have learned so far in a simple academic example. For the controls engineer, this example provides a first connection between conformal prediction and dynamical systems.

How to Select Nonconformity Scores

In the previous example, we selected a particular nonconformity score to quantify sensor uncertainty. Indeed, the choice of the nonconformity score is essential in obtaining accurate uncertainty estimates. While this choice gives us engineers great flexibility, it also raises the question of how to select it optimally to solve a given problem. Unfortunately, there is no general rule on how to do so. What we present in this survey are choices of nonconformity scores that have worked well and have appeared in the literature. We will provide literature pointers whenever needed.

Full and Split Conformal Prediction

There are two prominent variants of conformal prediction that differ in the way in which the nonconformity score is set up. In this survey, we will only use split conformal prediction. Nonetheless, we next briefly explain full conformal prediction and illustrate its computational challenges in practice. Therefore, assume that we are given a dataset of input–output pairs $(u^{(i)}, y^{(i)}) \sim \mathcal{D}$ for $i \in \{0, \ldots, L+K\}$, where $L, K > 0$. We assume scalar inputs and outputs for simplicity.

In split conformal prediction, we split the dataset into a training dataset of size L and a calibration dataset of size K. Specifically, we 1) train/construct a predictor $\mu : \mathbb{R} \to \mathbb{R}$ using only the training data $(u^{(i)}, y^{(i)})$, where $i \in \{K+1, \ldots, K+L\}$, and 2) define a nonconformity score $R^{(i)}$ that measures the prediction error between the predictor μ and the calibration

[5] To compute the empirical coverage (3) in practice [similar to the conditional empirical coverage (5)], we have to observe the realizations of $R_n^{(0)}, R_n^{(1)}, \ldots, R_n^{(K)} \sim \mathcal{R}$. for each $n \in \{1, \ldots, N\}$.

[6] Specifically, we can obtain an equivalent interpretation as $\text{Prob}(R^{(0)} \leq C(R^{(1)}, \ldots, R^{(K)})) = \mathbb{E}(\mathbb{1}(R^{(0)} \leq C(R^{(1)}, \ldots, R^{(K)})))$.

[7] The nonconformity scores $R^{(0)}, R^{(1)}, \ldots, R^{(K)}$ are generally treated as random variables that we would like to reason about. With the statement $\text{Prob}(R^{(0)} \leq C(R^{(1)}, \ldots, R^{(K)}) \,|\, R^{(1)}, \ldots, R^{(K)})$ we condition over observed realizations of $R^{(1)}, \ldots, R^{(K)}$.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

data $(u^{(i)}, y^{(i)})$, where $i \in \{1, \ldots, K\}$. For instance, a naive choice could be

$$R^{(i)} := |y^{(i)} - \mu(u^{(i)})|.$$

From the dataset splitting, it is easy to see that the nonconformity scores $R^{(i)}$ computed over $i \in \{0, \ldots, K\}$ are independent and identically distributed so that we obtain $\text{Prob}(R^{(0)} \leq C(R^{(1)}, \ldots, R^{(K)})) \geq 1 - \delta$ by Lemma 1. This guarantee can be rewritten in the form

$$\text{Prob}(y^{(0)} \in \mathcal{Y}(u^{(0)})) \geq 1 - \delta$$

where the prediction set $\mathcal{Y}(u^{(0)}) \subseteq R$ is defined as

$$\mathcal{Y}(u^{(0)}) := \{y \in \mathbb{R} \,||y - \mu(u^{(0)})| \leq C(R^{(1)}, \ldots, R^{(K)})\}.$$

In other words, the set $\mathcal{Y}(u^{(0)})$ contains the test output $y^{(0)}$ with a probability no less than $1 - \delta$. Additionally, note that the set $\mathcal{Y}(u^{(0)})$ is easy to compute in closed form.

Full conformal prediction, on the other hand, is computationally more expensive but does not require dataset splitting. For now, fix the pair $(u, y) \in \mathbb{R}^2$ and train/construct a predictor $\mu_{(u,y)} : \mathbb{R} \to \mathbb{R}$ using the augmented dataset consisting of the pair (u, y) and the full dataset $(u^{(i)}, y^{(i)})$, where $i \in \{1, \ldots, K + L\}$.[8] Similar to what we did before, define the nonconformity scores

$$R^{(i)}_{(u,y)} := |y^{(i)} - \mu_{(u,y)}(u^{(i)})|.$$

Let us now assume that $u = u^{(0)}$ and $y = y^{(0)}$. Then, note that $R^{(i)}_{(u^{(0)},y^{(0)})}$ computed over $i \in \{0, \ldots, K + L\}$ are not independent and identically distributed. Luckily, from the symmetry in $R^{(i)}_{(u^{(0)},y^{(0)})}$, one can check that $R^{(i)}_{(u^{(0)},y^{(0)})}$ are exchangeable so that we can still invoke Lemma 1; that is,

$$\text{Prob}(R^{(0)}_{(u^{(0)},y^{(0)})} \leq C(R^{(1)}_{(u^{(0)},y^{(0)})}, \ldots, R^{(K)}_{(u^{(0)},y^{(0)})})) \geq 1 - \delta.$$

This guarantee can again be rewritten in the form

$$\text{Prob}(y^{(0)} \in \mathcal{Y}(u^{(0)})) \geq 1 - \delta$$

where the prediction set $\mathcal{Y}(u^{(0)}) \subseteq \mathbb{R}$ is now defined as

$$\mathcal{Y}(u^{(0)}) := \left\{y \in \mathbb{R} \,||y - \mu_{(u^{(0)},y)}(u^{(0)})| \leq C\left(R^{(1)}_{(u^{(0)},y)}, \ldots, R^{(K)}_{(u^{(0)},y)}\right)\right\}.$$

Importantly, the set $\mathcal{Y}(u^{(0)})$ is no longer easy to compute as it requires us to train/construct a new predictor $\mu_{(u^{(0)},y)}$ for each potential output $y \in \mathbb{R}$. This is why, in this survey, we focus on split conformal prediction. The interested reader can learn more about this in [51].

[8]We here assume that the training of $\mu_{(u,y)}$ is deterministic and a symmetric function of the training dataset; that is, swapping datapoints within the training dataset does not change $\mu_{(u,y)}$.

Different Variants of Conformal Prediction

Various extensions of the vanilla conformal prediction framework were presented over the last two decades. While it is not within the scope of this article to review all of these extensions, we next present a selection of results that we believe are of great interest to the reader.

Calibration Conditional Conformal Prediction

As we noted before, calibration conditional guarantees of the form $\text{Prob}(R^{(0)} \leq C(R^{(1)}, \ldots, R^{(K)}) | R^{(1)}, \ldots, R^{(K)})$ cannot be obtained. However, the authors of [49] present a conformal prediction variant that provides calibration conditional coverage guarantees of the form

$$\text{Prob}_K(\text{Prob}(R^{(0)} \leq \bar{C}(R^{(1)}, \ldots, R^{(K)})) \geq 1 - \delta) \geq 1 - \beta \qquad (6)$$

where $\beta \in (0, 1)$ is a user-defined failure probability over the calibration data. We intuitively interpret the statement in (6) as "with a probability no less than $1 - \beta$ over the draw of calibration datapoints $R^{(1)}, \ldots, R^{(K)}$, it holds that $R^{(0)} \leq \bar{C}(R^{(1)}, \ldots, R^{(K)})$ with a probability no less than $1 - \delta$ over the draw of a test datapoint $R^{(0)}$." Consequently, the outer probability measure $\text{Prob}_K(\cdot)$ is defined over the randomness in $R^{(1)}, \ldots, R^{(K)}$, while the inner probability measure $\text{Prob}(\cdot)$ is defined over the randomness in $R^{(0)}$. If $\beta \in (0, 1)$ is chosen to be very small, then one can approximately obtain calibration conditional guarantees. We next summarize how to compute the prediction region $\bar{C} : \mathbb{R}^K \to \mathbb{R}$ in (6) with calibration conditional conformal prediction.

Lemma 2 (Calibration Conditional Coverage [49])

Let $R^{(0)}, \ldots, R^{(K)}$ be $K + 1$ independent and identically distributed random variables. Then, it holds that the probabilistic prediction region in (6) is valid by the choice of

$$\bar{C} := \text{Quantile}_{1 - \delta + \sqrt{\frac{\ln(1/\beta)}{2K}}}(R^{(1)}, \ldots, R^{(K)}, \infty).$$

To summarize this result: if we are interested in calibration conditional guarantees of the form (6), then we can simply use a slightly modified version of the vanilla conformal prediction algorithm. In fact, instead of computing the empirical quantile at a confidence level of $1 - \delta$, we compute the empirical quantile at a corrected confidence level of $1 - \delta + \sqrt{(\ln(1/\beta))/2K}$. We remark that Lemma 2 was derived in [49] using Hoeffding's inequality, a popular concentration inequality. As such, there is a close connection to concentration inequalities (see the section "Other Uncertainty Quantification Techniques" and also see "Calibration Conditional Conformal Prediction and Hoeffding's Inequality" for more details). Nonetheless, this result shows that calibration conditional guarantees can be derived by computing an empirical quantile.

There are two other variants of calibration conformal prediction that do not use Hoeffding's inequality and improve upon Lemma 2. One variant is stated next.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

Robot Navigation Under Sensor Uncertainty

Consider a robot described by discrete-time double-integrator dynamics $x_{t+1} = f(x_t, u_t)$, where $x_t := (p_t, v_t)$ denotes the 2D position p_t and velocity v_t. There are two unknown locations $r_l := (r_{lx}, r_{ly})$ for $l \in \{1, 2\}$ that follow uniform distributions $r_1 \sim \mathcal{U}([1.5, 2.5] \times [0.5, 1])$ and $r_2 \sim \mathcal{U}([2.5, 3.5] \times [4, 4.5])$. To estimate their location, we assume to have noisy sensors $s_l = (s_{lx}, s_{ly}) \sim \mathcal{L}(r_l, 0.025)$, where $\mathcal{L}$ denotes the Laplace distribution. The distributions of r_l and s_l are unknown, but for calibration purposes we assume to have access to K calibration samples from the sensors $(s_1^{(i)}, s_2^{(i)})$ along with ground truth locations $(r_1^{(i)}, r_2^{(i)})$ for $i \in \{1, \ldots, K\}$.

TASK

At test time, the robot should reach the locations $r_1^{(0)}$ and $r_2^{(0)}$ at times 5 and 15, respectively, with a probability no less than $1 - \delta := 0.95$; that is, for precision $\epsilon := 0.6$ we want that

$$\text{Prob}\left(\max\left(\|p_5 - r_1^{(0)}\|, \|p_{15} - r_2^{(0)}\|\right) \le \epsilon\right) \ge 1 - \delta. \tag{S3}$$

Furthermore, the robot should arrive at location $(5, 5)$ with precision 0.2 at time $T := 20$; that is, $\|p_T - (5, 5)^\top\| \le 0.2$.

SENSOR CALIBRATION

We define the nonconformity score

$$R^{(i)} := \max_{l \in \{1,2\}} \left(\|s_l^{(i)} - r_l^{(i)}\|\right).$$

Following Lemma 1, we know that $\text{Prob}\left(\max_{l \in \{1,2\}}(\|s_l^{(0)} - r_l^{(0)}\|) \le C\right) \ge 1 - \delta$, and consequently we also know that

$$\text{Prob}\left(\|s_l^{(0)} - r_l^{(0)}\| \le C, \forall l \in \{1, 2\}\right) \ge 1 - \delta$$

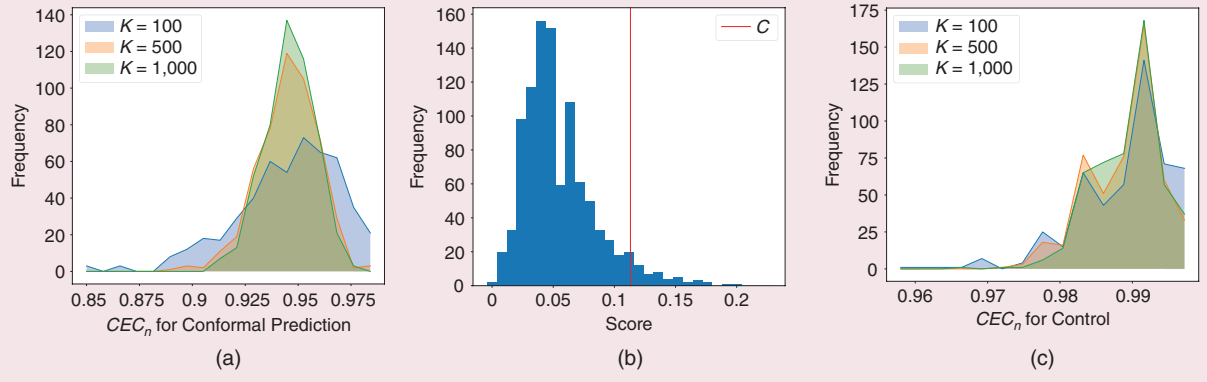


FIGURE S2 Empirical validation for the example here in "Robot Navigation Under Sensor Uncertainty." (a) Histogram of the conditional empirical coverage CEC_n over all N experiments. (b) Histogram of the nonconformity score $R^{(i)}$ over $K := 1{,}000$ calibration datapoints (for one of the N experiments). (c) Histogram of the conditional empirical coverage of (S3) over all N experiments.

Lemma 3 (Calibration Conditional Coverage [49])

Let $R^{(0)}, \ldots, R^{(K)}$ be $K + 1$ independent and identically distributed random variables. Then, it holds that the probabilistic prediction region in (6) is valid by the choice of

$$\bar{C} := \text{Quantile}_{1-\delta+\sqrt{\frac{2\delta\ln(1/\beta)}{K}}+\frac{2\ln(1/\beta)}{K}}\left(R^{(1)}, \ldots, R^{(K)}, \infty\right).$$

This variant, compared to the first, is beneficial for small δ. The last variant requires optimizing over an incomplete beta function to obtain the corrected confidence level that achieves (6). We refer the interested reader to [49] and [51].

Throughout this survey article, we will use conformal prediction with marginal coverage guarantees, as per Lemma 1. However, we emphasize that all proposed algorithms can be modified to provide calibration conditional coverage guarantees; for example, as per Lemma 2. Although the marginal and calibration conditional guarantees in (1) and (6) are semantically different, [49] shows that one can convert (1) into (6), and vice versa.

Extensions of Conformal Prediction

Our exposition was naturally biased by the goals that we pursue with this survey, and we only scratched the surface of the results from almost two decades of active research. For the interested reader, we summarize some of these results that we find important and potentially useful within formal verification and control in additional sidebars. In "Heteroskedasticity and Conformal Prediction," we present different ways in which one can deal with heteroskedasticity in the underlying data distribution.[9] For instance, in the example in "Robot Navigation Under Sensor Uncertainty," one could

[9]Heteroskedasticity happens when the variability of a variable depends on another independent variable.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

for sensor reading $s_l^{(0)}$ and unknown ground truth location $r_l^{(0)}$.

To perform an empirical validation of the previous statement, we conduct $N := 500$ experiments for calibration set sizes of $K \in \{100, 500, 1{,}000\}$. The obtained empirical coverage EC in (3) was 0.96, 0.958, and 0.948, respectively. We also computed the conditional empirical coverage CEC_n in (5) with $J := 500$. We plot the histogram of CEC_n in Figure S2(a), along with the histogram of the nonconformity scores of the calibration data from one experiment in Figure S2(b).

CONTROL SYNTHESIS

We solve the optimization problem

$$\begin{aligned} \min_{u_0,\ldots,u_{T-1}} \quad & \sum_{t=0}^{T-1} 0.4u_t^2 + 0.6\left\| p_T - (5,5)^\top \right\| \\ \text{subject to} \quad & x_{t+1} = f(x_t, u_t), \forall t \in \{1,\ldots,T-1\}, \\ & -1 \le u_t \le 1, \forall t \in \{0,\ldots,T-1\}, \\ & \max\left(\left\| p_5 - s_1^{(0)} \right\|, \left\| p_{15} - s_2^{(0)} \right\|\right) \le \epsilon - C, \\ & \| p_T - (5,5)^\top \| \le 0.2. \end{aligned}$$

Note that a feasible solution ensures that (S3) is satisfied. To see why, note that by the triangle inequality we have that

$$\begin{aligned} \left\| p_5 - r_1^{(0)} \right\| &\le \left\| p_5 - s_1^{(0)} \right\| + \left\| s_1^{(0)} - r_1^{(0)} \right\| \\ &\le \epsilon - C + \max\left(\left\| s_1^{(0)} - r_1^{(0)} \right\|, \left\| s_2^{(0)} - r_2^{(0)} \right\|\right). \end{aligned}$$

The same upper bound can be derived for the expression $\| p_{15} - r_2^{(0)} \|$. Since we know that $\text{Prob}(\max(\| s_1^{(0)} - r_1^{(0)} \|, \| s_2^{(0)} - r_2^{(0)} \|) \le C) \ge 1 - \delta$, we can conclude that (S3) is satisfied. Importantly, we want to remind the reader that the obtained guarantees are marginal; that is, in this case, the probability measure $\text{Prob}(\cdot)$ is defined over the randomness of test and calibration data $\{(r_1^{(i)}, r_2^{(i)}, s_1^{(i)}, s_2^{(i)})\}_{i=0}^{K}$.

For validation of the open-loop control policy, we compute the conditional empirical coverage of the constraint $\max(\| p_5 - r_1^{(0)} \|, \| p_{15} - r_2^{(0)} \|) \le \epsilon$ [in the same way CEC_n is defined in (5)], and we plot the histogram in Figure S2(c). Finally, we show three robot trajectories under the controller in Figure S3.

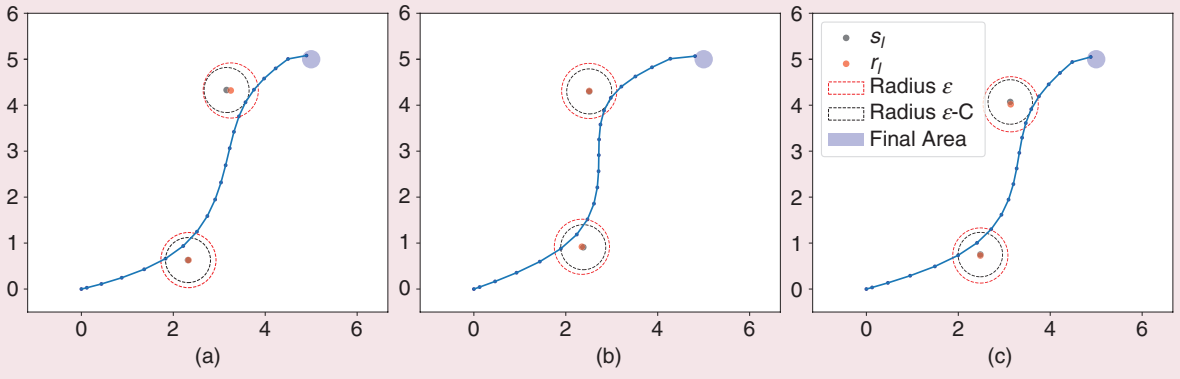


FIGURE S3 Three robot trajectories for this example in "Robot Navigation Under Sensor Uncertainty."

imagine that the variability of the sensor measurements depends on the location of the regions of interest; that is, the farther away a region is from the robot, the larger the measurement error will be.

In "Conformal Prediction Under Distribution Shift," we further discuss how to deal with the case where the datapoints $R^{(0)}, R^{(1)}, \ldots, R^{(K)}$ are not independent and identically distributed (and not exchangeable). There may be various reasons why our data may not be independent and identically distributed in practice; for example, consider the sim-2real gap of a system where the design conditions are different from the test/deployment conditions.

Other Uncertainty Quantification Techniques

There is a plethora of statistical techniques used for formal verification and control. In this section, our aim is to provide some context and show how conformal prediction relates to these techniques. However, exploring connections between these techniques and providing guidelines for when to use which technique are still active research areas; see, for example, [54]. To begin with, we can divide these techniques into two broad categories: techniques for 1) uncertainty quantification, for example, to quantify the approximation error of a neural network, and 2) solving optimization problems under uncertainty, for example, in solving chance-constrained optimization problems (CCOPs).

Techniques for Uncertainty Quantification

Pursuing the same goal as in conformal prediction, concentration inequalities provide bounds for random variables that hold with high probability [55], [56]. Well-known bounds are obtained by Markov's and Chebyshev's inequalities. However, these require knowledge of the random variable's first- and second-order moments, respectively.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

Heteroskedasticity and Conformal Prediction

Consider a setting in which inputs u are drawn from some distribution $\mathcal{D}_u$, for example, $u \sim \mathcal{D}_u$. Consider also outputs y that are drawn from some conditional distribution $\mathcal{D}_{y|u}$, that is, a distribution that depends on input u. We can write more compactly $(y,u) \sim \mathcal{D} := \mathcal{D}_{y|u} \times \mathcal{D}_u$. As a consequence, the variance of the output y may vary greatly as a function of u, a phenomenon referred to as heteroskedasticity. To capture such effects, we desire prediction regions C_u that depend on the sampled input u; that is, we desire $\text{Prob}(y \in C_u \,|\, u) \geq 1-\delta$, where $\text{Prob}(\cdot \,|\, u)$ is a conditional probability. However, such guarantees are known to be impossible to obtain without further assumptions, such as a finite input space [S14]. The good news is that there are efficient algorithms that provide marginal probability guarantees while capturing the effects of heteroskedasticity. *Conformalized quantile regression* (*CQR*) is one such algorithm, originally presented in [S15]. CQR builds on the success of quantile regressors [S16], [S17], here estimators $\hat{q}_{1-\delta}$ of the $1-\delta$ quantile of the random variable y. CQR follows a two-step procedure: 1) finding estimates $\hat{q}_{\delta/2}(u)$ and $\hat{q}_{1-\delta/2}(u)$ for the corresponding quantiles of y, and 2) computing the nonconformity score

$$R^{(i)} := \max\left(\hat{q}_{\delta/2}(u^{(i)}) - y^{(i)}, y^{(i)} - \hat{q}_{1-\delta/2}(u^{(i)})\right)$$

over a calibration dataset $(y^{(i)}, u^{(i)}) \sim \mathcal{D}$ for $i \in \{1,\ldots,K\}$. Then, for a test point $(y^{(0)}, u^{(0)}) \sim \mathcal{D}$ Lemma 1 guarantees

$$\text{Prob}\left(y^{(0)} \in [\hat{q}_{\delta/2}(u^{(i)}) - C, \hat{q}_{1-\delta/2}(u^{(0)}) + C]\right) \geq 1-\delta \qquad \text{(S4)}$$

where $C := \text{Quantile}_{1-\delta}(R^{(1)},\ldots,R^{(K)},\infty)$. Note specifically that the prediction region in (S4) depends on the input.

One other method is *locally adaptive conformal prediction*, in which we first train a predictor μ that predicts an output y from an input u. We then define the nonconformity score $R^{(i)} := \|y^{(i)} - \mu(u^{(i)})\|/\sigma(u^{(i)})$, where $\sigma(u^{(i)}) > 0$ measures the variability in $\|y^{(i)} - \mu(u^{(i)})\|$, for example, a neural network that was trained to estimate $\|y^{(i)} - \mu(u^{(i)})\|$. Lemma 1 directly gives us

$$\text{Prob}\left(\left\|y^{(0)} - \mu(u^{(0)})\right\| \leq C\sigma(u^{(i)})\right) \geq 1-\delta \qquad \text{(S5)}$$

where again $C := \text{Quantile}_{1-\delta}(R^{(1)},\ldots,R^{(K)},\infty)$. If $\sigma(u^{(i)})$ is a good approximator of the prediction error, we expect that the term $C\sigma(u^{(i)})$ in (S5) will capture the effects of heteroskedasticity.

REFERENCES

[S14] R. Foygel Barber, E. J. Candes, A. Ramdas, and R. J. Tibshirani, "The limits of distribution-free conditional predictive inference," *Inf. Inference: J. IMA*, vol. 10, no. 2, pp. 455–482, 2021, doi: 10.1093/imaiai/iaaa017.

[S15] Y. Romano, E. Patterson, and E. Candes, "Conformalized quantile regression," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 3543–3553.

[S16] R. Koenker, *Quantile Regression*, vol. 38. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[S17] L. Hao and D. Q. Naiman, *Quantile Regression*. London, U.K.: Sage, 2007.

Particularly popular are concentration inequalities for sums of independent and identically distributed random variables that do not require knowledge of first- and second-order moments. The motivation here is that the sum of random variables approximates the mean of the random variable if properly normalized, that is, for $\Sigma_{i=1}^{K} S^{(i)}/K$, where $S^{(1)},\ldots,S^{(K)}$ are independent and identically distributed bounded random variables. The Chernoff–Hoeffding inequality provides bounds for sums of Bernoulli random variables, whereas Hoeffding's inequality does so for sums of bounded random variables. We briefly discuss Hoeffding's inequality and show how it relates to the first variant of calibration conditional conformal prediction from Lemma 2 in "Calibration Conditional Conformal Prediction and Hoeffding's Inequality."

Closely related is also the Dvoretzky–Kiefer–Wolfowitz (DKW) inequality, which can provide a bound on the cumulative distribution function of a random variable in terms of an empirically estimated cumulative distribution function (see "Calibration Conditional Conformal Prediction and DKW Inequality"). We note that these concentration inequalities provide calibration conditional guarantees, while split and full conformal prediction provide marginal guarantees.

Finally, the Clopper–Pearson bound (discussed in more detail later) can be used to construct confidence bounds for a sequence of Bernoulli random variables. However, we are primarily interested in real-valued random variables that capture quantitative information, such as robustness.

Techniques for Solving Optimization Problems Under Uncertainty

Scenario optimization was initially proposed to approximately solve semi-infinite optimization problems, such as those arising in robust control [26], [27], [57]. Effectively, scenario optimization solves CCOPs by 1) collecting a finite number of samples from the uncertain parameters of the CCOP, 2) solving an approximate deterministic problem using these samples, and 3) providing statistical guarantees for constraint satisfaction of the original CCOP in the form of calibration conditional coverage guarantees. Scenario optimization mainly applies to CCOPs with convex cost and constraint functions, but has recently been extended to the nonconvex setting [58]. Very recently, the authors of [54] and [59] revealed close connections between scenario optimization and conformal prediction. Specifically, they show how to set up a scenario optimization problem that recovers marginal conformal prediction guarantees.

Another technique for solving CCOPs is distributionally robust optimization [60], [61]. Here, the main idea is to 1) define an ambiguity set (that is, a set of distributions)

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

Conformal Prediction Under Distribution Shift

The conformal prediction guarantees in Lemma 1 rely on the assumption that the random variables $R^{(0)}, \ldots, R^{(K)}$ are independent and identically distributed (or exchangeable). In practice, this assumption may not always hold, which is one of the major challenges. For instance, in autonomy applications, calibration data may only be available in the form of simulation data. While this data may come from high-fidelity photorealistic simulators in many applications, such as in autonomous driving and robotics, reality will always be different from these simulators. Another example is that the system may encounter different scenarios when deployed, for example, traffic scenarios or weather conditions.

One way to address this challenge is by using *robust conformal prediction* [52]. There, one assumes that $R^{(0)}, \ldots, R^{(K)}$ are again independent, but not identically distributed in the sense that $R^{(0)} \sim \mathcal{R}_0$ while $R^{(1)}, \ldots, R^{(K)} \sim \mathcal{R}$, where $\mathcal{R}_0$ is a test distribution. Under the assumption that the calibration and test distributions $\mathcal{R}$ and $\mathcal{R}_0$ are "close," the calibration data can still be used to bound $R^{(0)}$. Specifically, assume that $D_f(\mathcal{R}_0, \mathcal{R}) \leq \epsilon$, where $D_f(\cdot)$ is any f-divergence and $\epsilon \geq 0$. It then holds that

$$\text{Prob}\,(R^{(0)} \leq \tilde{C}) \geq 1 - \delta$$

where the constant $\tilde{C}$ is now instead computed as

$$\tilde{C} := \underset{1-\delta}{\text{Quantile}}\,(R^{(1)}, \ldots, R^{(K)}) \tag{S6}$$

where the tightened confidence level $\tilde{\delta}$ is defined as

$$\tilde{\delta} := 1 - g^{-1}(1 - \delta_n) \tag{S7}$$

and obtained by solving the set of convex optimization problems

$$\begin{aligned}
\delta_n &:= 1 - g\Big(\Big(1 + \frac{1}{K}\Big) g^{-1}(1 - \delta)\Big) \\
g(\beta) &:= \inf\left\{ z \in [0,1] \,|\, \beta f\Big(\frac{z}{\beta}\Big) + (1-\beta) f\Big(\frac{1-z}{1-\beta}\Big) \leq \epsilon \right\} \\
g^{-1}(\tau) &:= \sup\{\beta \in [0,1] \,|\, g(\beta) \leq \tau\}.
\end{aligned}$$

The reader is referred to [52] for more information. Robust conformal prediction provides valid guarantees without being too conservative when the distribution shift is small. However, large distribution shifts can result in conservative prediction regions $\tilde{C}$, potentially even with $\tilde{C} = \infty$, for example, when $1 - \tilde{\delta} = 1$. The authors of [S18] presented a form of robust conformal prediction that uses the Lévy–Prokhorov metric instead of an f-divergence, including the ∞-Wasserstein and the total variation distance as a special case. Recent work in [S19] also considers distribution shifts measured in the 1-Wasserstein distance. Finally, we mention work on adversarially robust conformal prediction [S20], [S21], [S22].

If we instead focus on the input–output setting in "Heteroskedasticity and Conformal Prediction" but with covariate distribution shifts, then we can obtain nonconservative guarantees via *weighted conformal prediction* [46], [S23]. Therefore, consider that $(y^{(i)}, u^{(i)})$ for $i \in \{1, \ldots, K\}$ are drawn from some calibration distribution $\mathcal{D} := \mathcal{D}_{y|u} \times \mathcal{D}_u$, while $(y^{(0)}, u^{(0)})$ is drawn from some test distribution $\mathcal{D}_0 := \mathcal{D}_{y|u} \times \tilde{\mathcal{D}}_u$. Note that the conditional distribution $\mathcal{D}_{y|u}$ remains the same, while only the covariate distribution can shift as $\tilde{\mathcal{D}}_u \neq \mathcal{D}_u$. The idea is that one can use a weighted version of conformal prediction when the covariate likelihood ratios $w(u) := \mathrm{d}p(u)/\mathrm{d}\tilde{p}(u)$ are known, where $p(u)$ and $\tilde{p}(u)$ are the probability density functions arising from $\mathcal{D}_u$ and $\tilde{\mathcal{D}}_u$, respectively. By defining the weight function $\pi(u^{(i)}) := w(u^{(i)}) / (\Sigma_{j=1}^{n} w(u^{(j)}) + w(u^{(i)}))$, we compute the weighted quantile for the nonconformity score $R(u^{(i)}, y^{(i)})$ as

$$C_w := \underset{1-\delta}{\text{Quantile}}\,(\pi(u^{(1)}) R(u^{(1)}, y^{(1)}), \ldots, \pi(u^{(K)}) R(u^{(K)}, y^{(K)}))$$

from which it follows that $\text{Prob}\,(R(u^{(0)}, y^{(0)}) \leq C_w) \geq 1 - \delta$. A generalization of weighted conformal prediction appeared in [S24], enabling conformal prediction for sequential feedback-loop covariate shifts as common in autonomous systems.

An important form of a distribution shift can be observed for time series in which the distribution of the nonconformity score $R^{(t)}$ changes at each time $t \in \{1, 2, \ldots\}$, for example, where $R^{(t)} \sim \mathcal{D}_t$ for some arbitrary distribution $\mathcal{D}_t$. In such cases, we can use a version of *adaptive conformal prediction* for which we present a basic discussion next; see [S25], [S26], [S27], [S28], [S29], and [53] for details. The idea here is to obtain a prediction region $C^{(t+1)}$ adaptively at time t so that $\Sigma_{t=0}^{T-1} \text{Prob}\,(R^{(t+1)} \leq C^{(t+1)})/T$ converges asymptotically to $1 - \delta$ as $T \to \infty$. In fact, this region is now computed as

$$C^{(t+1)} := \underset{1-\delta_{t+1}}{\text{Quantile}}\,(R^{(1)}, \ldots, R^{(t)})$$

where the variable δ_{t+1} is not necessarily equal to δ and instead adapted online. One such choice (following [S26]) can be

$$\delta_{t+1} := \delta_t + \gamma(\delta - e_t) \quad \text{with} \quad e_t := \begin{cases} 0 & \text{if } R^{(t)} \leq C^{(t)} \\ 1 & \text{otherwise} \end{cases}$$

where $\gamma > 0$ is a learning rate. Note that adaptive conformal prediction is the least restrictive, since no assumptions were made on the distributions $\mathcal{D}_t$. As such, we can capture large distribution shifts and rapidly changing distributions. However, this comes at the cost of weaker asymptotic guarantees. Finally, we note that we later obtain prediction regions for time series by instead assuming to have access to trajectory calibration data.

REFERENCES

[S18] L. Aolaritei, M. I. Jordan, Y. Marzouk, Z. O. Wang, and J. Zhu, "Conformal prediction under Levy-Prokhorov distribution shifts: Robustness to local and global perturbations," 2025, *arXiv:2502.14105*.

[S19] R. Xu, C. Chen, Y. Sun, P. Venkitasubramaniam, and S. Xie, "Wasserstein-regularized conformal prediction under general distribution shift," in *Proc. 13th Int. Conf. Learn. Representations*, 2025.

(*Continued*)

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

Conformal Prediction Under Distribution Shift (*Continued*)

[S20] A. Gendler, T.-W. Weng, L. Daniel, and Y. Romano, "Adversarially robust conformal prediction," in *Proc. Int. Conf. Learn. Representations*, 2021.

[S21] S. Ghosh, Y. Shi, T. Belkhouja, Y. Yan, J. Doppa, and B. Jones, "Probabilistically robust conformal prediction," in *Proc. 39th Uncertainty Artif. Intell.*, PMLR, 2023, pp. 681–690.

[S22] L. Jeary, T. Kuipers, M. Hosseini, and N. Paoletti, "Verifiably robust conformal prediction," in *Proc. 38th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2024, pp. 4295–4314.

[S23] J. Jonkers, G. Van Wallendael, L. Duchateau, and S. Van Hoecke, "Conformal predictive systems under covariate shift," in *Proc. 13th Symp. Conformal Probabilistic Prediction Appl.*, 2024, pp. 1–18.

[S24] D. Prinster, S. D. Stanton, A. Liu, and S. Saria, "Conformal validity guarantees exist for any data distribution (and how to find them)," in *Proc. 41st Int. Conf. Mach. Learn. (ICML)*, 2024, pp. 41,086–41,118.

[S25] M. Zaffran, O. Féron, Y. Goude, J. Josse, and A. Dieuleveut, "Adaptive conformal predictions for time series," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2022, pp. 25,834–25,866.

[S26] I. Gibbs and E. Candes, "Adaptive conformal inference under distribution shift," in *Proc. 35th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2021, pp. 1660–1672.

[S27] I. Gibbs and E. J. Candès, "Conformal inference for online prediction with arbitrary distribution shifts," *J. Mach. Learn. Res.*, vol. 25, no. 162, pp. 1–36, 2024.

[S28] A. Angelopoulos, E. Candes, and R. J. Tibshirani, "Conformal PID control for time series prediction," in *Proc. 37th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2024, pp. 23,047–23,074.

[S29] S. Feldman, L. Ringel, S. Bates, and Y. Romano, "Achieving risk control in online learning settings," 2024, *arXiv:2205.09095*.

that contains the underlying distribution of the CCOP with confidence no less than $1-\beta$, and 2) solve a robust version of the CCOP using this ambiguity set. The ambiguity set is usually selected so that solving the robust CCOP is easier than solving the original problem. For instance, one popular way is to draw samples from the uncertain parameters of the CCOP and construct Wasserstein distance sets around them [62]. Similar to scenario optimization, calibration conditional coverage guarantees can be obtained for the constraint satisfaction of the underlying CCOP. To the best of our knowledge, there are currently no known connections between distributionally robust optimization and scenario optimization or conformal prediction. However, the work in [63] presents initial results for using conformal prediction to obtain ambiguity sets that can be used within the distributionally robust optimization framework.

Recent approaches explored the use of conformal prediction for solving CCOPs [45], [64], [65]. The idea here is to 1) replace the chance constraint of the CCOP by a deterministic constraint that involves the empirical quantile over samples from the uncertain parameters of the CCOP, 2) rewrite this deterministic constraint in a more tractable way (for example, via complementarity or mixed-integer linear constraints), and 3) solve this reformulated problem and conformalize the solution using a second set of samples from the uncertain parameters of the CCOP. There are notable differences to scenario optimization and distributionally robust optimization as this approach 1) involves a second set of samples and 2) cannot a priori guarantee constraint satisfaction of the underlying CCOP to a desired confidence level. Nonetheless, such approaches can provide a posteriori guarantees on the constraint satisfaction in the convex as well as in the nonconvex setting. Furthermore, empirical evidence suggests that this approach is more sample efficient than scenario optimization, where the bounds on constraint violation scale with the dimension of the decision variable [64]. More detailed comparisons should be conducted in future work.

Finally, we mention probably approximately correct (PAC)-Bayes theory, which is a technique for solving supervised learning problems, such as classification or regression problems [66], [67], [68], [69]. PAC-Bayes theory considers randomized predictors via distributions over parameters from a family of parameterized predictors. PAC-Bayes theory then derives calibration conditional coverage guarantees on the error of the predictor, for example, by using Hoeffding's inequality. Importantly, starting from a given prior distribution of parameters, PAC-Bayes theory provides error bounds that can be optimized over. The authors of [70] present PAC-Bayes generalization bounds for conformal predictors that provide efficient prediction regions. In the remainder of this survey, we illustrate the benefits of using conformal prediction as a simple, general, and efficient technique for uncertainty quantification applicable to formal verification and control.

LEC VERIFICATION WITH CONFORMAL PREDICTION

In this section, we show how to use conformal prediction for verifying the correctness of LECs, such as for computing the input–output properties of neural networks. The presented techniques are a straightforward application of conformal prediction, and our main goal is to ease the reader into the topic and to illustrate conformal prediction's simplicity and efficiency. The main technical sections, presented hereafter, will then present techniques for control design as well as offline and online verification of LEASs. To set the stage, we use $\mu : \mathbb{R}^{n_u} \to \mathbb{R}^{n_y}$ to denote an LEC, for example, a feedforward neural network. Note that this is the most general description of an LEC possible.

Challenges in LEC Verification

The verification of input–output properties of LECs has been well studied in the computer science community over the past decade. The goal is to verify (or disprove) that the output of an already-trained LEC satisfies an output property ϕ_{out} for all inputs that satisfy an input property ϕ_{in}.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

Calibration Conditional Conformal Prediction and Hoeffding's Inequality

Let $S^{(1)},\dots,S^{(K)}$ be independent and identically distributed random variables that have bounded support. For simplicity, we here assume that $0\le S^{(1)},\dots,S^{(K)}\le 1$. For $X := S^{(1)}+\dots+S^{(K)}$, Hoeffding's inequality tells us that

$$\text{Prob}_K(X-\mathbb{E}(X)\le t)\ge 1-\exp\left(\frac{-2t^2}{K}\right)$$

where $\mathbb{E}(X)$ is the expected value of the random variable X. Note here that the probability measure $\text{Prob}_K(\cdot)$ is again taken over the randomness of $S^{(1)},\dots,S^{(K)}$. To explore connections with conformal prediction, let us again consider a test datapoint $S^{(0)}$ and assume that $S^{(0)},S^{(1)},\dots,S^{(K)}$ are independent and identically distributed. Note also that $\mathbb{E}(X)=\Sigma_{i=1}^{K}\mathbb{E}(S^{(i)})=K\mathbb{E}(S^{(0)})$. If we now set $\beta := \exp(-2t^2/K)$, we then obtain

$$\text{Prob}_K\left(X-K\mathbb{E}(S^{(0)})\le\sqrt{\frac{\ln(1/\beta)K}{2}}\right)\ge 1-\beta.$$

If the random variable $S^{(i)}$ is chosen to be the indicator function of an event related to the nonconformity score $R^{(i)}$, we can draw a connection with calibration conditional conformal prediction in Lemma 2 and (6). Particularly, for the choice of $S^{(i)} := \mathbb{1}(R^{(i)}\le C')$, where C' is some user-defined constant, we obtain $\mathbb{E}(S^{(0)})=\text{Prob}(R^{(0)}\le C')$[a] so that

$$\text{Prob}_K\left(\text{Prob}(R^{(0)}\le C')\ge\frac{X}{K}-\sqrt{\frac{\ln(1/\beta)}{2K}}\right)\ge 1-\beta. \tag{S8}$$

This choice of $S^{(i)}$ that leads to the guarantee in (S8) now reveals a connection with (6). This is not surprising since, as we mentioned before, Lemma 2 was derived using Hoeffding's inequality. To compare (S8) with (6), fix β and C' in (S8) and set $1-\delta := X/K-\sqrt{(\ln(1/\beta))/2K}$ so that $\delta := 1-X/K+\sqrt{(\ln(1/\beta))/2K}$. By plugging this δ into (6), we $\bar{C} := \text{Quantile}_{X/K}(R^{(1)},\dots,R^{(K)},\infty)$. For (S8) and (6) to give similar guarantees, we expect that $C'\approx\bar{C}$. We recall that $\bar{C}=R^{(p)}$, where $p := \lceil (K+1)X/K\rceil$ and $R^{(K+1)} := \infty$. For simplicity, assume that $R^{(1)},\dots,R^{(K)}$ are not the same. Then, close inspection reveals that $R^{(p-1)}\le C'\le\bar{C}=R^{(p)}$.

Finally, we remark that this comparison concerns the first variant of calibration conditional conformal prediction in Lemma 2. A comparison with the variant in Lemma 3 is not immediate.

[a]Note that the expected value of an indicator function of an event is equivalent to the probability of that event.

Calibration Conditional Conformal Prediction and DKW Inequality

Let $R^{(0)},R^{(1)},\dots,R^{(K)}$ again be independent and identically distributed random variables. The DKW inequality [S30], [S31] guarantees that

$$\text{Prob}_K\left(\sup_{r\in\mathbb{R}}F_n(r)-F(r)\le t\right)\ge 1-\exp(-2Kt^2)$$

if $\exp(-2Kt^2)\le 0.5$. Here, $F(r) := \text{Prob}(R^{(0)}\le r)$ is the cumulative distribution function of $R^{(0)}$, while $F_n(r) := \Sigma_{i=1}^{K}\mathbb{1}(R^{(i)}\le r)/K$ is the corresponding empirical cumulative distribution function. Setting $\beta := \exp(-2Kt^2)$ and fixing $r := C'$, we hence get

$$\text{Prob}_K\left(Prob(R^{(0)}\le C')\ge F_n(C')-\sqrt{\frac{\ln(1/\beta)}{2K}}\right)\ge 1-\beta.$$

Noting that $F_n(C')=X/K$, we can make similar arguments as in "Calibration Conditional Conformal Prediction and Hoeffding's Inequality" to establish a connection between calibration conditional conformal prediction and the DKW inequality.

REFERENCES

[S30] A. Dvoretzky, J. Kiefer, and J. Wolfowitz, "Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator," *Ann. Math. Statist.*, vol. 27, no. 3, pp. 642–669, 1956, doi: 10.1214/aoms/1177728174.

[S31] P. Massart, "The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality," *Ann. Probability*, vol. 18, no. 3, pp. 1269–1283, 1990, doi: 10.1214/aop/1176990746.

Formally, we want to verify that $\mu(u)\vDash\phi_{\text{out}}$ holds for all inputs $u\in\mathbb{R}^{n_u}$ that satisfy $u\vDash\phi_{\text{in}}$, where $\vDash$ is the satisfaction operator; that is, $u\vDash\phi_{\text{in}}$ means "the input u satisfies the property ϕ_{in}."[10] Equivalently, we can write

$$\forall u\in\mathbb{R}^{n_u},\ u\vDash\phi_{\text{in}}\Rightarrow\mu(u)\vDash\phi_{\text{out}}. \tag{7}$$

[10]One may be interested in more general properties ϕ that are defined over inputs and outputs of μ simultaneously; for example, one may be interested in verifying $\forall u\in\mathbb{R}^{n_u},(u,\mu(u))\vDash\phi$. However, the if–then form in (7) is expressive enough in practice and is commonly used within the community.

As a concrete example, consider an input set $C_{\text{in}}\subseteq\mathbb{R}^{n_u}$ and an output set $C_{\text{out}}\subseteq\mathbb{R}^{n_y}$ for which we want to verify that $\mu(C_{\text{in}})\subseteq C_{\text{out}}$ with $\mu(C_{\text{in}}) := \{y\in\mathbb{R}^{n_y}\,|\,\exists u\in C_{\text{in}},\mu(u)=y\}$. This specific example points to an important problem regarding the reachability of μ.

Computational Bottlenecks in Model-Based Approaches

To address this goal, model-based approaches (often also referred to as optimization-based approaches) were proposed that use a mathematical description of the LEC, usually with a focus on feedforward neural networks; see [71] and [72] for an

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

overview. However, these model-based verification techniques are challenging to solve as the underlying verification problem is NP-hard. A sound verification algorithm provides a positive answer only if (7) is true, while a complete verification algorithm provides a positive answer if and only if (7) is true. Popular complete verification algorithms are based on mixed-integer linear programming [73], [74], [75], [76] or satisfiable modulo theory solvers [77], [78], [79]. Computationally more efficient sound, but noncomplete, verification algorithms were also proposed, for example, using semidefinite programming [80], [81], bound propagation [82], [83], and abstract interpretation techniques [84], [85], [86]. While complete verification algorithms are subject to scalability challenges in practice, noncomplete methods are usually conservative because of the use of overapproximations and hence restrictive. We have already mentioned that these challenges of model-based approaches will only amplify when attempting to verify LEASs, which use complex LECs.

Efficient Statistical Verification of LECs

In what follows, we apply conformal prediction to navigate the tradeoff between scalability and conservatism while additionally being able to deal with uncertainty, for example, noise affecting the input of an LEC. We will obtain practical algorithms that can compute input–output properties in real time and with high confidence. In doing so, we give up on the soundness of our verification algorithms and instead obtain probabilistically sound verification algorithms. A probabilistically sound verification algorithm for (7) provides a positive answer only if (7) is true with a probability not less than $1-\delta$. We are thus interested in

$$u \sim \mathcal{D}_{\phi_{\text{in}}} \Longrightarrow \text{Prob}\,(\mu(u) \vDash \phi_{\text{out}}) \geq 1-\delta \tag{8}$$

where $\mathcal{D}_{\phi_{\text{in}}}$ is a distribution with support over the input set $C_{\text{in}} := \{u \in \mathbb{R}^{n_u} \,|\, u \vDash \phi_{\text{in}}\}$. Indeed, the data in verification problems are not always deterministic—for example, in image classification or state estimation, where stochasticity has to be taken into account. Verification problems are thus not always expressed in the form of (7) and one instead formulates them in the form of (8).

Remark 1

So far, our exposition has focused solely on computing input–output properties of an LEC μ. However, LECs are usually evaluated on input–output data, for example, to predict states from measurements in state estimation or to predict labels from images in image classification. We note that (8) can easily be modified to capture requirements in such settings. That is, let

$$(u, y) \sim \mathcal{D} \Rightarrow \text{Prob}((\mu(u), y) \vDash \phi) \geq 1-\delta$$

where inputs and outputs $(u, y) \sim \mathcal{D}$ are drawn from some distribution $\mathcal{D}$, which usually arises from an underlying data-generating distribution, and the output property ϕ, which is now defined over the output y and the output of the LEC $\mu(u)$. For instance, an output property $\phi := \|\mu(u) - y\| \leq \epsilon$ expresses that the prediction error $\|\mu(u) - y\|$ is not larger than $\epsilon > 0$.

In verification problems that are not deterministic, for example, when the underlying data are stochastic as in Remark 1, the distribution $\mathcal{D}_{\phi_{\text{in}}}$ arises naturally as part of the problem formulation. However, as mentioned before, we can also obtain probabilistically sound verification algorithms for (7) by instead solving (8) for a carefully constructed distribution $\mathcal{D}_{\phi_{\text{in}}}$, for example, by uniform sampling over C_{in}. When dealing with verification problems of the form (8), we hence advise the reader to carefully investigate when $\text{D}_{\phi_{\text{in}}}$ arises naturally from the problem formulation and when $\mathcal{D}_{\phi_{\text{in}}}$ is constructed. We will encounter both cases, sometimes mixed, in this survey. While the verification problems in (7) and (8) are stated for LECs, we will later extend them to apply to LEASs.

Finally, we emphasize two distinct advantages of such statistical approaches over model-based verification techniques. First, model-based verification techniques face various challenges in the presence of uncertainty (inducing more conservatism and/or computational complexity), while statistical techniques elegantly reason about stochastic uncertainty. Second, model-based techniques are model dependent and often tailored to specific types of neural networks, while statistical techniques apply to any contemporary black-box LEC. We argue that this is an important property when using algorithms from a fast-paced field, such as machine learning.

Probabilistically Sound LEC Verification

We now illustrate how to obtain probabilistically sound verification algorithms for input–output properties of LECs using conformal prediction. The key idea will be to define nonconformity scores that are in accordance with the input–output properties at hand. Throughout this section, we make the following assumption.

Assumption 1

We have independently sampled a calibration dataset of K inputs from the distribution $\mathcal{D}_{\phi_{\text{in}}}$; that is, we have access to samples $u^{(i)} \sim \mathcal{D}_{\phi_{\text{in}}}$ for $i \in \{1, \ldots, K\}$.

We proceed by presenting solutions to problems of reachability and general verification of LECs. Specifically, in the former case, the set $C_{\text{out}} := \{y \in \mathbb{R}^{n_y} | y \vDash \phi_{\text{out}}\}$ is explicitly given (for example, a polytope in $\mathbb{R}^{n_y}$), while in the latter case, only knowledge of ϕ_{out} is available.

Reachability of LECs

Assume that we are given an input set $C_{\text{in}} \subseteq \mathcal{R}^{n_u}$ and an output set $C_{\text{out}} \subseteq \mathbb{R}^{n_y}$. Then, we can directly define the nonconformity score

$$R^{(i)} := \text{dist}\,(\mu(u^{(i)}), C_{\text{out}}) \tag{9}$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

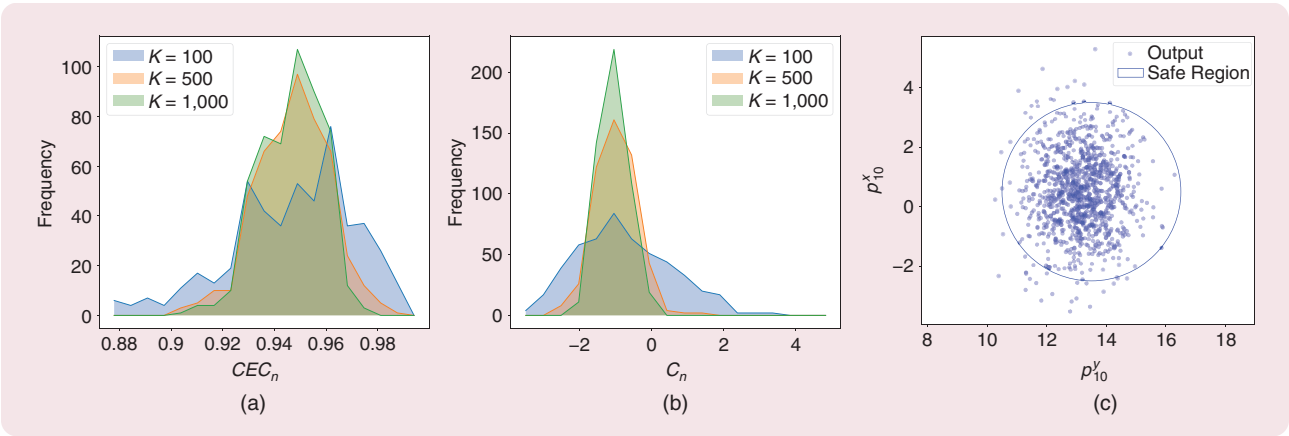


FIGURE 2 Empirical validation for Example 1. (a) Histogram of the conditional empirical coverage CEC_n over all N experiments. (b) Histogram of the bound C_n over all N experiments. (c) Illustration of 1,000 test datapoints.

where we recall that $u^{(i)} \sim \mathcal{D}_{\phi_{\text{in}}}$ for $i \in \{0, 1, \ldots, K\}$, and where $\text{dist}(\mu(u^{(i)}), C_{\text{out}})$ measures the distance between the output $\mu(u^{(i)})$ and the output set C_{out}.[11] As before, the distribution $\mathcal{D}_{\phi_{\text{in}}}$ is assumed to have support over C_{in}, and $\mathcal{D}_{\phi_{\text{in}}}$ can either arise as part of the problem formulation or by sampling the input set C_{in}. By applying Lemma 1, we directly obtain the following result.

Corollary 1

Given an output set $C_{\text{out}} \subseteq \mathbb{R}^{n_y}$ the distribution $\mathcal{D}_{\phi_{\text{in}}}$ with support over an input set $C_{\text{in}} \subseteq \mathbb{R}^{n_u}$, and a test input $u^{(0)} \sim \mathcal{D}_{\phi_{\text{in}}}$, we have that

$$\text{Prob}\left(\text{dist}(\mu(u^{(0)}), C_{\text{out}}) \leq C\right) \geq 1 - \delta$$

where $C := \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, \infty)$ with $R^{(i)}$ being defined in (9).

By the definition of the signed distance function, we obtain a positive verification answer if $C < 0$.[12] In this case, the absolute value of C indicates how robustly the reachability specification is satisfied. Note that this is a sufficient condition, which means that our algorithm is inconclusive if $C \geq 0$. In this case, we can instead define the nonconformity score $R^{(i)} := -\text{dist}(\mu(u^{(i)}), C_{\text{out}})$ with which we can compute C such that $\text{Prob}(-\text{dist}(\mu(u^{(0)}), C_{\text{out}}) \leq C) \geq 1 - \delta$. We now obtain a negative verification answer if $-C > 0$.

We note that the distance $\text{dist}(\mu(u^{(i)}), C_{\text{out}})$ may not be easy to compute if the set C_{out} has a nontrivial geometric structure. If the output set is instead parameterized as $C_{\text{out}} := \{y \in \mathbb{R}^{n_y} | h_{\text{out}}(y) \leq 0\}$ for some function $h_{\text{out}} : \mathbb{R}^{n_y} \to \mathbb{R}$, we can select the nonconformity score $R^{(i)} := h_{\text{out}}(\mu(u^{(i)}))$ and proceed in the same way as done previously. Let us illustrate Corollary 1 with a simple example.

Example 1

Consider a robot described by the initial state $u := (p_0^x, p_0^y, \theta_0, v_0)$, where p_0^x and p_0^y denote the 2D position, θ_0 denotes the angle, and v_0 denotes the translational velocity. The unknown model of the robot follows a discrete-time unicycle dynamics with constant translational and rotational velocity and additive noise on the position following the normal distribution $\mathcal{N}(0, 0.01^2)^2$. We want to predict p_{10}^x and p_{10}^y from the initial state u, where p_{10}^x and p_{10}^y are the robot's position 10 time steps ahead. We thus train a simple feedforward neural network μ to obtain predictions $y := \mu(u) \approx (p_{10}^x, p_{10}^y)$. How can we now check if the model μ gives reasonable predictions? We first fix an input distribution $\mathcal{D}_{\phi_{\text{in}}} := \mathcal{U}([0,1))^2 \times \mathcal{N}(0, 0.1^2) \times \mathcal{TN}(1, 0.1^2)^2$, where $\mathcal{U}$ is the uniform distribution, $\mathcal{N}$ is the normal distribution, and $\mathcal{TN}$ is the normal distribution truncated at [0, 2]. Using prior physical knowledge of the robot, we postulate that the safe set $C_{\text{out}} := \{p := (p^x, p^y) \in \mathbb{R}^2 \mid h_{\text{out}}(p) \leq 0\}$ with $h_{\text{out}}(p) := (p^x - 13.5)^2 + (p^y - 0.5)^2 - 3^2$ contains p_{10}^x and p_{10}^y, for example, $(p_{10}^x, p_{10}^y) \in C_{\text{out}}$. We set the failure probability to $\delta := 0.05$ and use the nonconformity score $R^{(i)} := h_{\text{out}}(\mu(u^{(i)}))$ as discussed before, where $u^{(i)} \sim \mathcal{D}_{\phi_{\text{in}}}$. Following Corollary 1, we know that $\text{Prob}(h_{\text{out}}(\mu(u^{(0)})) \leq C) \geq 1 - \delta$. We conduct $N := 500$ experiments for calibration set sizes of $K \in \{100, 500, 1{,}000\}$. Again, we first verify statistical validity empirically and compute the empirical coverage EC according to (3), as 0.962, 0.964, and 0.96, respectively. We also plot the conditional empirical coverage CEC_n according to (5) with $J := 500$ in Figure 2(a). To check the quality of the learned model μ, we compute $\Sigma_{n=1}^{N} C_n / N$, which is the average of the bound C_n over all N experiments, where we recall from (3) and (5) that $C_n := C(R_n^{(1)}, \ldots, R_n^{(K)})$. For $K \in \{100, 500, 1{,}000\}$, we obtain the values -0.51, -0.75, and -0.81, respectively. We can thus conclude that the model μ reasonably captures the reachable set of the unicycle dynamics, which was postulated as C_{out}, at the confidence level of $1 - \delta$. In Figure 2(b), we also plot the histogram of C_n over all N experiments. Finally, we illustrate 1,000 test datapoints in Figure 2(c).

[11] A common choice to measure the distance between a point y and a set C is the signed distance $\text{dist}(y, C) := \inf_{y' \in C} \|y - y'\|$ if $y \notin C$ and $\text{dist}(y, C) := -\inf_{y' \in \mathbb{R}^{n_y} \setminus C} \|y - y'\|$ otherwise.

[12] Corollary 1 implies $\text{Prob}(C < 0 \implies \mu(u^{(0)}) \in C_{\text{out}}) \geq 1 - \delta$.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

General Verification of LECs

We can verify other input–output properties as long as we can encode them as real-valued functions, that is, any performance function that is defined over the output of the LEC. In the remainder, we focus on the case where ϕ_{out} is expressed in predicate logic over outputs of the LEC μ by its generality. For a survey on formal specifications for the verification of LECs, we refer the reader to [89]. To give a motivating example, consider the specification

$$\begin{aligned}\phi_{\text{out}} := ((\mu_1(u) \geq 0) \implies (\mu_2(u) < 0)) \\ \wedge ((\mu_2(u) \geq 0) \implies (\mu_1(u) < 0))\end{aligned}$$

which expresses mutual exclusivity of the two outputs μ_1 and μ_2 of μ. As we do not want to distract the reader from the main developments of this survey, we refer the reader to "LEC Specifications in Predicate Logic" for more details on how predicate logic specifications over μ can formally be defined. Given a predicate logic specification ϕ_{out} over μ, this sidebar also instructs how one can construct a real-valued performance function $\rho^{\phi_{\text{out}}}: \mathbb{R}^{n_y} \to \mathbb{R}$ that enjoys the following soundness property:

$$\rho^{\phi_{\text{out}}}(\mu(u)) > 0 \implies \mu(u) \vDash \phi_{\text{out}}.$$

We note that larger values of $\rho^{\phi_{out}}(\mu(u))$ are beneficial as they indicate robustness against perturbation in u; see [87] and [88]. For the previous mutual exclusivity property, one obtains

$$\begin{aligned}\rho^{\phi_{\text{out}}}(\mu(u)) := \min(\max(-\mu_1(u), -\mu_2(u)), \\ \max(-\mu_2(u), -\mu_1(u)))\end{aligned}$$

and it can easily be verified that the soundness property holds, which we leave as an exercise for the reader.

To now verify LEC specifications in predicate logic, we can define the nonconformity score

$$R^{(i)} := -\rho^{\phi_{\text{out}}}(\mu(u^{(i)})) \tag{10}$$

from which we get the following result using Lemma 1.

LEC Specifications in Predicate Logic

We begin by defining the syntax of predicate logic over μ as

$$\phi ::= \text{True} \mid h \mid \neg\phi' \mid \phi' \wedge \phi'' \tag{S9}$$

where $h: \mathbb{R}^{n_y} \to \mathbb{R}$ is a constraint function to impose constraints of the form $h(\mu(u)) \geq 0$, $\neg\phi'$ denotes the negation of a predicate logic specification ϕ', and $\phi' \wedge \phi''$ denotes the conjunction of the predicate logic specifications ϕ' and ϕ''. This syntax defines a set of rules according to which predicate logic specifications can be constructed. The notation $::=$ means that the left-hand side in (S9), where ϕ is a free variable, is assigned to be one of the expressions from the right-hand side (separated by vertical bars). In other words, the expressions True and h are predicate logic specifications that we can denote by ϕ' or ϕ''. We then recursively construct new predicate logic specifications using negations and conjunctions. We can further define disjunction, implication, and equivalence operators as

$$\begin{aligned}\phi' \vee \phi'' &:= \neg(\neg\phi' \wedge \neg\phi'') && \text{(disjunction operator)} \\ \phi' \Rightarrow \phi'' &:= \neg\phi' \vee \phi'' && \text{(implication operator)} \\ \phi' \Leftrightarrow \phi'' &:= (\phi' \Rightarrow \phi'') \wedge (\phi'' \Rightarrow \phi') && \text{(equivalence operator).}\end{aligned}$$

To formally determine if a predicate logic specification as defined in (S9) is satisfied, we define the Boolean semantics as

$$\begin{aligned}\mu(u) &\vDash \text{True} && \text{iff holds by definition,} \\ \mu(u) &\vDash h && \text{iff } h(\mu(u)) \geq 0, \\ \mu(u) &\vDash \neg\phi' && \text{iff } \mu(u) \nvDash \phi', \\ \mu(u) &\vDash \phi' \wedge \phi'' && \text{iff } \mu(u) \vDash \phi' \text{ and } \mu(u) \vDash \phi''.\end{aligned}$$

Finally, we can define quantitative semantics $\rho^{\phi}: \mathbb{R}^{n_y} \to \mathbb{R}$ that will reduce checking $\mu(u) \vDash \phi$ to instead checking the positivity of $\rho^{\phi}(\mu(u))$.[a] Formally, we define the quantitative semantics as

$$\begin{aligned}\rho^{\text{True}}(\mu(u)) &:= \infty \\ \rho^{h}(\mu(u)) &:= h(\mu(u)) \\ \rho^{\neg\phi'}(\mu(u)) &:= -\rho^{\phi'}(\mu(u)) \\ \rho^{\phi' \wedge \phi''}(\mu(u)) &:= \min(\rho^{\phi'}(\mu(u)), \rho^{\phi''}(\mu(u))).\end{aligned}$$

It is known that these semantics are sound [87], [88]; that is, that it holds that $\mu(u) \vDash \phi$ holds if $\rho^{\phi}(\mu(u)) > 0$. Predicate logic additionally allows one to include existential ($\exists$) and universal ($\forall$) quantifiers over the input u, which we mention here but omit from our presentation for the sake of simplicity.

To give more intuition, we consider an academic example via the specification $\phi := (h_1 \vee h_2) \wedge \neg h_3$. The conjunction $\wedge$ is the parent operator that connects to the expressions $h_1 \vee h_2$ and $\neg h_3$. Following the semantics, we know that

$$\mu(u) \vDash \phi \text{ iff } \mu(u) \vDash h_1 \vee h_2 \text{ and } \mu(u) \vDash \neg h_3.$$

For $h_1 \vee h_2$ and $\mu(u) \vDash \neg h_3$, we also know that

$$\begin{aligned}\mu(u) \vDash h_1 \vee h_2 &\text{ iff } \mu(u) \vDash h_1 \text{ or } \mu(u) \vDash h_2 \\ \mu(u) \vDash \neg h_3 &\text{ iff } \mu(u) \nvDash h_3.\end{aligned}$$

From this, we conclude that $\mu(u) \vDash \phi$ if and only if

- $h_1(\mu(u)) \geq 0$ or $h_2(\mu(u)) \geq 0$ and
- $h(\mu(u)) < 0$.

[a] In the main text of this survey article, we refer to ρ^{ϕ} also as a performance function.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

Corollary 2

Given an output property ϕ_{out} with sound performance function $\rho^{\phi_{\text{out}}}: \mathbb{R}^{n_y} \to \mathbb{R}$, the distribution $\mathcal{D}_{\phi_{\text{in}}}$ with support over the set $\mathcal{C}_{\text{in}} := \{u \in \mathbb{R}^{n_u} \mid u \vDash \phi_{\text{in}}\} \subseteq \mathbb{R}^{n_u}$ defined by an input property ϕ_{in}, and a test input $u^{(0)} \sim \mathcal{D}_{\phi_{\text{in}}}$, we have

$$\text{Prob}\left(-C \leq \rho^{\phi_{\text{out}}}(\mu(u^{(0)}))\right) \geq 1 - \delta.$$

where $C := \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, \infty)$ with $R^{(i)}$ being defined in (10).

Lemma 1 gives C such that $\text{Prob}\left(-\rho^{\phi_{\text{out}}}(\mu(u^{(0)})) \leq C\right) \geq 1 - \delta$, from which the preceding result is derived by simple manipulation. We then conclude that the algorithm produces a positive verification result if $C < 0$.[13]

Limitations of Statistical Techniques

Corollaries 1 and 2 are powerful and enable verification of arbitrary LECs. However, we want to remind the reader that the coverage guarantees obtained via conformal prediction are marginal in the sense that the probability measure $\text{Prob}(\cdot)$ in Corollaries 1 and 2 is defined jointly over the randomness in the test and calibration datapoints $u^{(i)} \sim \mathcal{D}_{\phi_{\text{in}}}$ for $i \in \{0, \ldots, K\}$. The ideal result, by contrast, would be that the probability measure $\text{Prob}(\cdot)$ is only defined over the randomness in the test datapoint $u^{(0)} \sim \mathcal{D}_{\phi_{\text{in}}}$. This limitation is unavoidable with statistical techniques; for example, it is also present with calibration conditional coverage guarantees obtained via calibration conditional conformal prediction (see the section "Different Variants of Conformal Prediction") or scenario optimization.

More Properties

Other examples of input–output properties are checking probabilistic prediction sets for classification tasks (see [21]) or estimating function properties of μ, such as Lipschitz constants (see [90]), and extreme values, among others. To provide one last example, consider that we want to estimate the extreme value of μ when $n_y := 1$; that is, we want to estimate $\mu^* := \sup_{u \in \mathcal{C}_{\text{in}}} \mu(u)$. By defining the nonconformity score $R^{(i)} := \mu(u^{(i)})$, we can obtain C such that $\text{Prob}(\mu(u^{(0)}) \leq C) \geq 1 - \delta$. So far, our mental model has been to fix a failure probability $\delta \in (0,1)$ and apply conformal prediction. However, this approach would not make much sense here, as we are interested in the largest value of $\mu(u^{(i)})$. We thus reverse our approach and select the smallest $\delta \in (0,1)$ that satisfies $\lceil (K+1)(1-\delta) \rceil \leq K$. This choice, as discussed before, guarantees that $R^{(i)} \neq \infty$ while finding C that is equal to the largest value of $\mu(u^{(i)})$. Our intention with this paragraph was to illustrate that conformal prediction can be a versatile tool. To estimate extreme values, we note that there exist specialized techniques, such as extreme value theory; for example, see [91].

[13] Corollary 2 implies $\text{Prob}(C < 0 \implies \mu(u^{(0)}) \vDash \phi_{\text{out}}) \geq 1 - \delta$.

State Estimation

Before concluding this section, we refer the reader to "Conformalizing the Kalman Filter," which shows how state estimators, such as the Kalman filter, can be verified. We note that the algorithm proposed in the sidebar indeed applies to arbitrary state estimators. Our goal with this example is to show that conformal prediction can complement well-established algorithms within control and estimation theory, such as the Kalman filter, which may not provide guarantees if certain assumptions break.

CONTROL SYNTHESIS FOR LEASS WITH CONFORMAL PREDICTION

On an abstract level, LEASs are dynamical control systems that use learning-enabled controllers, either learned in an end-to-end fashion or as a composition of object detection, state estimation, trajectory prediction, decision making, and low-level feedback control layers. We can model LEASs as discrete-time stochastic dynamical systems

$$z_{t+1} = f(z_t, u_t, v_t) \tag{11a}$$

$$y_t = p(z_t, w_t) \tag{11b}$$

where we follow the convention that $z_t \in \mathbb{R}^{n_z}$, $u_t \in \mathbb{R}^{n_u}$, and $y_t \in \mathbb{R}^{n_y}$ describe the system's state, control input, and measurement at time t, while $v_t \in \mathbb{R}^{n_v}$ and $w_t \in \mathbb{R}^{n_w}$ describe generic disturbances and sensor noise, respectively. We assume that v_t and w_t follow distributions $\mathcal{D}_v$ and $\mathcal{D}_w$, respectively. Assume also that we sampled the initial condition z_0 from a distribution $\mathcal{D}_z$. The functions $f: \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_z}$ and $p: \mathbb{R}^{n_z} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_y}$ model the dynamics and the sensors of the system, respectively. We note that the model in (11) is restricted to discrete-time models but is otherwise fairly general.

The control input u_t is assumed to be a function of past measurements $\{y_s\}_{s=0}^t$, for example,

$$u_t := \pi(\{y_s\}_{s=0}^t) \tag{12}$$

which can reduce to a state feedback controller $\pi(z_t)$ in the simplest case. We recall that the system in (11) is learning enabled when π contains LECs, for example, a reinforcement learning controller or a combination of a model predictive controller with learning-enabled state predictors as we mostly consider in this section.

Objective

Our next goal is to design a learning-enabled controller π that satisfies a given property with high confidence. This is in contrast to later sections on online and offline verification of LEASs with conformal prediction, where π can be

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

Conformalizing the Kalman Filter

Let $z_{t+1} = f(z_t, v_t)$ be a dynamical system, where $z_t \in \mathbb{R}^{n_z}$ and $v_t \in \mathbb{R}^{n_v}$ are the state and the disturbance at time t, respectively. Let $y_t = p(z_t, w_t) \in \mathbb{R}^{n_y}$ be measurements, where $w_t \in \mathbb{R}^{n_w}$ is sensor noise. The functions $f : \mathbb{R}^{n_z} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_z}$ and $p : \mathbb{R}^{n_z} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_y}$ model system dynamics and sensors. We assume that the input $u := (z_0, v_t, w_t)$ of the dynamical system follows a distribution $\mathcal{D}$ from which a trajectory $z := (z_0, z_1, \ldots)$ is generated. For calibration purposes, we assume that we have available K inputs $u^{(i)}$ sampled from $\mathcal{D}$ along with the corresponding trajectories $z^{(i)}$. We note that knowledge of calibration trajectories $z^{(i)}$ is a strong assumption at this point, which we will comment on further in the section "State Estimation and Perception."

State estimators use past and present measurements $\{y_s\}_{s=0}^{t}$ to compute an estimate $\hat{z}_t$ of the state z_t at time t. Popular techniques are based on Kalman or particle filtering (see [18] for an overview). However, recursive state estimators, such as the Kalman filter, are only valid under restrictive assumptions on the system, while the particle filter is computationally prohibitive in practice. Specifically, the Kalman filter is an optimal state estimator that minimizes the expected mean-square estimation error if 1) the functions f and p are linear, and 2) the random variables v_t and w_t are Gaussian. In practice, these assumptions rarely hold, motivating us to conformalize the Kalman filter. Consider, therefore, the nonconformity score

$$R^{(i)} := \max\left(\left\|\hat{z}_1^{(i)} - z_1^{(i)}\right\|, \ldots, \left\|\hat{z}_T^{(i)} - z_T^{(i)}\right\|\right)$$

which we evaluate over the calibration dataset, for example, over ground truth information $z^{(i)}$ for $i \in \{1, \ldots, K\}$. Having defined this nonconformity score, it is easy to see that

$$\text{Prob}\left(\left\|\hat{z}_t^{(0)} - z_t^{(0)}\right\| \le C, \forall t \in \{1, \ldots, T\}\right) \ge 1 - \delta$$

by the choice of $C := \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, \infty)$. We specifically note that the use of the max operator in the definition of the nonconformity score enables us to get simultaneous coverage of $\left\|\hat{z}_t^{(0)} - z_t^{(0)}\right\|$ for all times $t \in \{1, \ldots, T\}$ with high probability. However, using the maximum in this way may let time(s) where the estimation error is large dominate. We will later discuss in the section "Statistical Abstractions of Dynamic Environments" how to address this subtlety and present a normalized version of this nonconformity score.

an arbitrary controller of the form (12) that may not be guaranteed to satisfy the required property, hence motivating us to verify (11) under π.

Autonomous systems often operate in dynamic and uncertain environments. The state z_t of the system in (11) hence consists of physical states $x_t \in \mathbb{R}^{n_x}$ and environment states $e_t \in \mathbb{R}^{n_e}$, for example, describing uncontrollable agents, such as pedestrians or objects of interest. To make this distinction explicit, let us write the LEAS in (11) as

$$\begin{bmatrix} x_{t+1} \\ e_{t+1} \end{bmatrix} = \begin{bmatrix} f_x(x_t, u_t) \\ f_e(e_t, v_t) \end{bmatrix}$$
$$y_t = (x_t, e_t)$$

where we note that we, at least for now, decoupled the x and e subsystems. This assumption will help us to illustrate the main ideas while being able to provide strong guarantees. In the section "Safe Control in Dynamic Environments," we will explain how this assumption can be relaxed. Further note that only the x system is controllable by the input u_t, while only the e subsystem is affected by the disturbance v_t. We assume to know the function $f_x : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$, that is, the system dynamics, while the function $f_e : \mathbb{R}^{n_e} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_e}$ that describes the environment is not known. In fact, we do not need to assume Markovian dynamics for the e subsystem and can instead assume that the sequence $(e_0, \ldots, e_T)$ follows an unknown discrete-time random process, for example, $(e_0, \ldots, e_T) \sim \mathcal{D}_e$. Therefore, we consider the system

$$x_{t+1} = f_x(x_t, u_t) \tag{13a}$$
$$(e_0, \ldots, e_T) \sim \mathcal{D}_e \tag{13}$$
$$y_t = (x_t, e_t) \tag{13c}$$

and assume to have access to trajectories of finite length T from $\mathcal{D}_e$, where $T > 0$ is some finite task horizon.

Assumption 2

We have independently sampled a calibration dataset of K trajectories from the environment in (13b); that is, we have access to samples $e^{(i)} := (e_0^{(i)}, \ldots, e_T^{(i)}) \sim \mathcal{D}_e$ for $i \in \{1, \ldots, K\}$.

Finally, note that we here assumed an ideal sensor with $y_t = (x_t, e_t)$; that is, we have access to x_t and e_t at time t. Later, in the section "State Estimation and Perception," we will relax this assumption and consider dealing with perception and estimation uncertainty.

We are interested in designing safe controllers, for example, controllers that achieve the satisfaction of constraints formulated in terms of x and e. For most of this section, we are interested in solving the following problem.

Problem 1

Given the system in (13), a safety constraint $c : \mathbb{R}^{n_x} \times \mathbb{R}^{n_e} \to \mathbb{R}$ over states x_t and e_t, and a task horizon $T > 0$, design control inputs $u_0, \ldots, u_{T-1}$ such that x_t satisfies c for all times with a probability no less than $1 - \delta$, for example, such that

$$\text{Prob}\left(c(x_t, e_t) \ge 0, \ \forall t \in \{1, \ldots, T\}\right) \ge 1 - \delta.$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

The safety constraint c is enforced pointwise in time. In the section "Temporal Logic-Constrained Control," we will consider a more general setting where we enforce trajectory-wise constraints, for example, temporal logic specifications ϕ defined over the trajectories x and e. We also note that the task horizon T is finite. While this assumption does exclude us from expressing asymptotic properties, such as stability and forward invariance, it still allows for expressing a broad range of properties in practice. Similar assumptions can commonly be found in the learning-based control and formal verification literature.

In what follows, we consider a compositional approach to design π. Indeed, conformal prediction is particularly suited for compositional reasoning and control design, as will become evident in this section. The high-level idea here is to construct efficient statistical abstractions of LECs, for example, those used for state estimation and prediction, which can then be used for control design; recall Figure 1.

Statistical Abstractions of Dynamic Environments

To safely control the system in (13), we need to reliably quantify the future behavior of the environment.

Environment Prediction

At each time t, we need to predict future states $(e_{t+1}, \ldots, e_T)$ from past and present observations $(e_0, \ldots, e_t)$. We denote the predictions of e_τ for future times $\tau > t$ by $\hat{e}_{\tau|t}$. One can think of the predictor at time t as a function $\mu : \mathbb{R}^{(t+1)n_e} \to \mathbb{R}^{(T-t)n_e}$ such that

$$(\hat{e}_{t+1|t}, \ldots, \hat{e}_{T|t}) = \mu(e_0, \ldots, e_t).$$

Popular prediction models are recurrent neural networks [92], long short-term memory networks [5], [93], [94], transformers [95], or more traditional autoregressive moving-average models [96]. These models may achieve varying accuracy on different prediction tasks, which we will account for by quantifying the uncertainty of these predictions via statistical abstractions. Finally, we remark that prediction models have to be trained on datasets that are independent of the calibration dataset from Assumption 2.

The idea for computing statistical abstractions of the environment is to use trajectory predictors and conformal prediction in combination. We hence start by computing predictions of the trajectory predictor over the calibration dataset. Specifically, we compute the predictions

$$(\hat{e}^{(i)}_{t+1|t}, \ldots, \hat{e}^{(i)}_{T|t}) := \mu(e^{(i)}_0, \ldots, e^{(i)}_t)$$

for all times $t \geq 0$ from the calibration data $i \in \{1, \ldots, K\}$.

Uncertainty Representation via Statistical Abstractions

So far, we have been using the terminology of a statistical abstraction rather abstractly. Formally, a statistical abstraction of the environment is a set $E \subseteq \mathbb{R}^{Tn_e}$ such that

$$\text{Prob}((e_1, \ldots, e_T) \in E) \geq 1 - \delta.$$

Ideally, one would like to compute a statistical abstraction E that is minimal according to some quantifiable metric. However, such minimal sets E can generally be difficult to compute and may be nonconvex. Here we are motivated from a practical point of view and having in mind that we want to use these sets downstream.

Once we have obtained such a statistical abstraction, we will be able to use it for control design. For this purpose, we are interested in two slightly different statistical abstractions that enable the design of an open-loop and a closed-loop controller that solve Problem 1 with different information available. For open-loop control, we only have information available at time $t = 0$, and we will compute a statistical abstraction of the form

$$\text{Prob}(\| e_\tau - \hat{e}_{\tau|0} \| \leq C_{\tau|0},\ \forall \tau \in \{1, \ldots, T\}) \geq 1 - \delta \tag{14}$$

where $C_{\tau|0}$ indicates the τ-step-ahead prediction error for predictions $\hat{e}_{\tau|0}$ made at time $t = 0$. For closed-loop control, we use the information available at all times $t \in \{0, \ldots, T-1\}$ to make one-step-ahead predictions $\tau = t + 1$. Specifically, we will compute statistical abstractions of the form

$$\text{Prob}(\| e_{t+1} - \hat{e}_{t+1|t} \| \leq C_{t+1|t},\ \forall t \in \{0, \ldots, T-1\}) \geq 1 - \delta \tag{15}$$

where $C_{t+1|t}$ indicates the one-step-ahead prediction error for predictions $\hat{e}_{t+1|t}$ made at time t.

We note that there are different ways in which we can compute abstractions of the form (14) and (15). We will now present two such approaches: one naive approach that is intuitive but potentially conservative, as presented in a similar form in [97], [98], and another nonconservative approach, as originally presented in [99].

Naive Approach

In this approach, we naively compute a probabilistic prediction region for each error $\|e_\tau - \hat{e}_{\tau|0}\|$ (or $\|e_{t+1} - \hat{e}_{t+1|t}\|$) and combine these into guarantees of the form (14) [or (15)] using Boole's inequality.[14] We define

$$R^{(i)}_{\tau|t} := \| e^{(i)}_\tau - \hat{e}^{(i)}_{\tau|t} \| \tag{16}$$

as the nonconformity score, and then compute

$$C_{\tau|t} := \text{Quantile}_{1-\bar{\delta}}\left(R^{(1)}_{\tau|t}, \ldots, R^{(K)}_{\tau|t}, \infty\right) \tag{17}$$

where $\bar{\delta} := \delta / T$ is a tightened failure probability that is such that $\bar{\delta} \leq \delta$. By using $\bar{\delta}$ instead of δ, we make a

[14]For a countable set of events $A_1, A_2, \ldots$, Boole's inequality states that the probability that at least one of the events is true is upper bounded by the sum of the probabilities of all individual events being true, for example, that $\text{Prob}(\cup_{i=1}^{\infty} A_i) \leq \Sigma_{i=1}^{\infty} \text{Prob}(A_i)$.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

correction that will enable us to obtain guarantees of the form (14) and (15). Let us summarize these results.

Theorem 1

Given a test trajectory $e^{(0)} \sim \mathcal{D}_e$, the statistical abstractions in (14) and (15) are valid by the choice of $C_{\tau|t}$ in (17) with $\bar{\delta} := \delta/T$ and the nonconformity score (16).

Proof

Let us show that the abstraction in (14) is valid by the choice of $C_{\tau|t}$, while the proof for the abstraction in (15) follows similarly. From Lemma 1 and the definition of the nonconformity score $R^{(i)}_{\tau|t}$, it follows immediately that $\text{Prob}(\| e_\tau - \hat{e}_{\tau|t} \| \leq C_{\tau|0}) \geq 1 - \bar{\delta}$ holds for each $\tau \in \{1, \ldots, T\}$ individually. We consequently know that $\text{Prob}(\|e_\tau - \hat{e}_{\tau|0}\| > C_{\tau|0}) \leq \bar{\delta}$. Applying Boole's inequality gives us

$$\text{Prob}(\exists \tau > 0 \text{ subject to } \|e_\tau - \hat{e}_{\tau|0}\| > C_{\tau|0}) \leq \sum_{i=t}^{T} \bar{\delta} = \sum_{i=t}^{T} \frac{\delta}{T} = \delta$$

so that we can finally conclude that

$$\text{Prob}(\|e_\tau - \hat{e}_{\tau|0}\| \leq C_{\tau|0},\ \forall \tau \in \{1, \ldots, T\}) \geq 1 - \delta$$

by which we have shown that (14) holds. □

As a consequence of using the tightened failure probability $\bar{\delta}$ instead of δ, this naive approach requires more data as now $\lceil (K+1)(1-\bar{\delta}) \rceil \leq K$ has to hold to obtain nontrivial $C_{\tau|t}$, for example, $C_{\tau|t}$ that are not ∞. Another consequence is that the statistical abstractions will be conservative since

$$\begin{aligned} &\text{Quantile}_{1-\bar{\delta}}\left(R^{(1)}_{\tau|t}, \ldots, R^{(K)}_{\tau|t}, \infty\right) \\ &\qquad \geq \text{Quantile}_{1-\delta}\left(R^{(1)}_{\tau|t}, \ldots, R^{(K)}_{\tau|t}, \infty\right). \end{aligned}$$

Specifically, we will notice the conservatism when the distribution of $R^{(i)}_{\tau|t}$ has a long tail. This motivates our second approach, as presented next.

Single Nonconformity Score Approach

To avoid using the tightened failure probability $\bar{\delta}$ and combining T conformal prediction instances via Boole's inequality, we define a single nonconformity score that enables us to reason over multiple prediction errors simultaneously. We obtain the open-loop abstraction in (14) by the nonconformity score

$$R^{(i)}_{\text{OL}} := \max_{\tau \in \{1, \ldots, T\}} \alpha_{\tau|0} \left\| e^{(i)}_\tau - \hat{e}^{(i)}_{\tau|0} \right\| \tag{18}$$

where the choice of constants $\alpha_{\tau|0} > 0$ will be crucial to obtain tight abstractions, as we will discuss later. We then proceed by computing

$$C_{\text{OL}} := \text{Quantile}_{1-\bar{\delta}}(R^{(1)}_{\text{OL}}, \ldots, R^{(K)}_{\text{OL}}, \infty). \tag{19}$$

Almost equivalently, we obtain the closed-loop abstraction in (15) by the nonconformity score

$$R^{(i)}_{\text{CL}} := \max_{t \in \{0, \ldots, T-1\}} \alpha_{t+1|t} \left\| e^{(i)}_{t+1} - \hat{e}^{(i)}_{t+1|t} \right\| \tag{20}$$

where we again use constants $\alpha_{t+1|t} > 0$ and then compute

$$C_{\text{CL}} := \text{Quantile}_{1-\bar{\delta}}(R^{(1)}_{\text{CL}}, \ldots, R^{(K)}_{\text{CL}}, \infty). \tag{21}$$

Next, we state the main result up front, again utilizing Lemma 1, which is followed by a discussion of the choice of the constants $\alpha_{\tau|0} > 0$ and $\alpha_{t+1|t} > 0$.

Theorem 2

Given a test trajectory $e^{(0)} \sim \mathcal{D}_e$ and positive constants $\alpha_{\tau|0} > 0$ and $\alpha_{t+1|t} > 0$, the statistical abstractions in (14) and (15) are valid by the choices of

$$\begin{aligned} C_{\tau|0} &:= C_{\text{OL}}/\alpha_{\tau|0}, \\ C_{t+1|t} &:= C_{\text{CL}}/\alpha_{t+1|t} \end{aligned}$$

where C_{OL} and C_{CL} follow from (19) and (21) with the nonconformity scores in (18) and (20), respectively.

Proof

Let us show that the abstraction in (14) is valid by the choice of $C_{\tau|t}$, while the proof for the abstraction in (15) again follows similarly. From Lemma 1 and the definition of the nonconformity score $R^{(i)}_{\text{OL}}$, it immediately follows that

$$\text{Prob}\Big(\max_{\tau \in \{1, \ldots, T\}} \alpha_{\tau|0} \left\| e^{(0)}_\tau - \hat{e}^{(0)}_{\tau|0} \right\| \leq C_{\tau|0}\Big) \geq 1 - \delta$$

which, since $\alpha_{\tau|0} > 0$, is equivalent to

$$\text{Prob}\Big(\left\| e^{(0)}_\tau - \hat{e}^{(0)}_{\tau|0} \right\| \leq C_{\tau|0}/\alpha_{\tau|0},\ \forall \tau \in \{1, \ldots, T\}\Big) \geq 1 - \delta$$

by which we have shown that (14) holds with $C_{\tau|0} := C_{\text{OL}}/\alpha_{\tau|0}$. □

While Theorem 2 holds for arbitrary positive constants $\alpha_{\tau|0}$ and $\alpha_{t+1|t}$, we obtain informative, for example, nonconservative, abstractions only if these constants are computed intelligently. Let us focus on the nonconformity score $R^{(i)}_{\text{OL}}$ in (18) that results in the abstraction (14) to get some intuition. Why do we not simply select $\alpha_{\tau|0} := 1$ for all $\tau \in \{1, \ldots, T\}$? The reason is that the prediction errors $\|e^{(i)}_\tau - \hat{e}^{(i)}_{\tau|0}\|$ have different distributions; for example, the prediction error for $\tau = T$ typically has the largest $1 - \delta$ quantile and would hence dominate the max operator in $R^{(i)}_{\text{OL}}$. In other words, the upper bound $C_{\tau|0}$ from Theorem 2 would be the same for each time $\tau \in \{1, \ldots, T\}$, which could be tight for $\tau = T$ but may be loose for $\tau < T$.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

To overcome this challenge, one simple idea is to select $\alpha_{\tau|0}$ such that it normalizes the prediction error, that is, such that $\alpha_{\tau|0}\|e_\tau^{(i)} - \hat{e}_{\tau|0}^{(i)}\|$ lies within the interval $[0, 1]$. To achieve this (approximately) in practice, we sample an additional dataset of M trajectories from the environment in (13b) so that we have access to $e^{(i)} := (e_0^{(i)}, \ldots, e_T^{(i)}) \sim \mathcal{D}_e$ for $i \in \{K+1, \ldots, K+M\}$.[15] We can now compute the normalization constants for (18) and (20) as

$$\alpha_{\tau|0} := \frac{1}{\max\limits_{i \in \{K+1,\ldots,K+M\}} \left\| e_\tau^{(i)} - \hat{e}_{\tau|0}^{(i)} \right\|} \tag{22a}$$

$$\alpha_{t+1|t} := \frac{1}{\max\limits_{i \in \{K+1,\ldots,K+M\}} \left\| e_{t+1}^{(i)} - \hat{e}_{t+1|t}^{(i)} \right\|}. \tag{22b}$$

We note that the computation of $\alpha_{\tau|0}$ and $\alpha_{t+1|t}$ is highly efficient, but is not necessarily incentivizing any form of optimality. Instead, we can cast the problem of finding the parameters $\alpha_{\tau|0}$ and $\alpha_{t+1|t}$ as an optimization problem in which we minimize the $1-\delta$ quantile over the dataset $e^{(i)} := (e_0^{(i)}, \ldots, e_T^{(i)}) \sim \mathcal{D}_e$ for $i \in \{K+1, \ldots, K+M\}$, following the ideas of [99]. We refer the reader to "Normalization Constants via Mixed-Integer Linear Complementarity Programming" for details. Finally, we provide an example that illustrates the two approaches that we presented to compute statistical abstractions.

Example 2

We construct open-loop statistical abstractions as in (14) for pedestrians modeled by a distribution $e \sim \mathcal{D}_e$ that is implicitly defined by the optimal reciprocal collision avoidance (ORCA) simulator [101]. We consider the current time $t := 0$ and a mission horizon of $T = 20$ time steps, requiring us to make predictions for times $\tau \in \{1, \ldots, 20\}$.[16] First, we train a long short-term memory network that takes social interaction between pedestrians into account; see [7] and [102] for details. In the remainder, we set the failure probability to $\delta := 0.05$ and compute three different types of statistical abstractions using 1) the "naive approach," 2) the "single nonconformity score approach" with normalization constants as in (22a), and 3) the "single nonconformity score approach" with normalization constants as in "Normalization Constants via Mixed-Integer Linear Complementarity Programming." We refer to these three methods as UB, SNSA-CF, and SNSA-LCP, respectively.[17]

Computing Normalization Constants. We first compute the normalization constants for SNSA-CF and SNSA-LCP by using $M := 50$ trajectories sampled from $\mathcal{D}_e$. The average computation times over 500 experiments for SNSA-CF and SNSA-LCP are 0.00016 and 0.08291 s, respectively. While the normalization constants for SNSA-LCP provide tighter statistical abstractions (details and plots are provided next), they require larger computation times. We also remark that the computation times for SNSA-LCP may increase drastically when increasing T or M [recall the optimization problem in (S10)].

Statistical Validation. We conduct $N := 500$ experiments for calibration set sizes of $K := 596$ and test set sizes of $J := 500$. We plot the conditional empirical coverage of $\|e_\tau - \hat{e}_{\tau|0}\| \le C_{\tau|0}$ [defined in the same way CEC_n is defined in (5)] for all three methods in Figure 3(a).[18] As expected, it can be seen that UB is overly conservative, while SNSA-CF and SNSA-LCP provide exact coverage around $1-\delta$ confidence. The average conditional empirical coverage of $\|e_\tau - \hat{e}_{\tau|0}\| \le C_{\tau|0}$ is 0.994, 0.946, and 0.946 for UB, SNSA-CF, and SNSA-LCP, respectively.

Comparing All Three Methods. For one experiment, we illustrate the pedestrian trajectory alongside the predictions and the statistical abstractions in Figure 3(b). In Figure 3(c), we also plot the average value of $\|e_\tau - \hat{e}_{\tau|0}\|$ for each time τ, which can here be interpreted as the volume of the abstraction. We see again that UB has the largest coverage because of the induced conservatism, while SNSA-CF and SNSA-LCP provide valid statistical abstractions with smaller volumes. We also note that SNSA-LCP has the smallest volume, which comes at the cost of increased computation times for the normalization constants (as discussed previously). Finally, we show 2D plots of the statistical abstractions, along with the test trajectories at times $\tau \in \{3, 6, 9\}$ in Figure 4. We again see that SNSA-CF and SNSA-LCP tightly cover a $1-\delta$ fraction of the test data.

Shaping the Geometry of Statistical Abstractions

In both methods (naive and single nonconformity score), we have fixed the spatial shape of the abstraction to $\|e_\tau - \hat{e}_{\tau|t}\|$. In other words, for (14) [and similarly for (15)], the abstraction will be a ball of radius $C_{\tau|0}$ centered around the prediction $\hat{e}_{\tau|0}$. This choice may lead to conservative abstractions when $e_\tau - \hat{e}_{\tau|0}$ is not symmetrically distributed. One such example is when the environment consists of multiple agents, for instance, when $e_t := (e_{t,1}, \ldots, e_{t,N})$ for N different agents, and some agent predictions have more uncertainty than others. Following a similar idea as in (18) and (20), we can define a nonconformity score that uses the maximum prediction error $\|e_{t,n} - \hat{e}_{\tau|t,n}\|$ over all $n \in \{1, \ldots, N\}$ agents [103]. We discuss this approach in more detail at the end of the next section. Another idea was presented in

[15] We remark that this dataset could simply be the dataset that was used to train the predictor μ. However, we emphasize that the dataset cannot be the calibration dataset from Assumption 2, which would result in dependent nonconformity scores.

[16] We assume that 20 time steps of observations are already available as inputs for the predictor, but we keep $t := 0$ nonetheless for notational consistency. We also discretized the original signal at a rate of 8 Hz so that we have a total of 5 s of data, with observations and predictions of 2.5 s each.

[17] UB stands for union bound, while SNSA stands for single nonconformity score approach with extensions CF and LCP for closed-form and linear complementarity programming, respectively.

[18] The computation of $C_{\tau|0}$ for UB follows (17), while the computation of $C_{\tau|0}$ for SNSA-CF and SNSA-LCP follows Theorem 2.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

Normalization Constants via Mixed-Integer Linear Complementarity Programming

Let us focus on the computation of the normalization constants $\alpha_{\tau|0}$; the constants $\alpha_{t+1|t}$ can be computed similarly. We cast the problem of finding $\alpha_{\tau|0}$ as the optimization problem

$$\min_{\alpha_{1|0},\ldots,\alpha_{T|0}} \text{Quantile}_{1-\delta}(R^{(K+1)},\ldots,R^{(K+M)},\infty) \tag{S10a}$$

$$\text{subject to } R^{(i)} = \max_{\tau\in\{1,\ldots,T\}} \alpha_{\tau|0}\left\| e_\tau^{(i)} - \hat{e}_{\tau|0}^{(i)} \right\|, \forall i \in \{K+1,\ldots,K+M\} \tag{S10b}$$

$$\sum_{\tau=1}^{T} \alpha_{\tau|0} = 1 \tag{S10c}$$

$$\alpha_{\tau|0} \geq 0, \forall \tau \in \{1,\ldots,T\}. \tag{S10d}$$

We note that the equality constraint in (S10b) computes the nonconformity score $R^{(i)}$ defined in (18) over the additional dataset $i \in \{K+1,\ldots,K+M\}$, while the cost function in (S10a) minimizes the $1-\delta$ quantile over $R^{(K+1)},\ldots,R^{(K+M)},\infty$. Finally, the inequality constraint in (S10d) enforces positive normalization constants, while the equality constraint in (S10c) enforces a normalization budget, avoiding that $\alpha_{\tau|0} = 0$ for all $\tau \in \{1,\ldots,T\}$, which would be a trivial solution. In fact, this constraint enforces that $\alpha_{\tau|0} > 0$ for all $\tau \in \{1,\ldots,T\}$; see [99, Theorem 1]. The optimization problem in (S10) is always feasible since any set of parameters $\alpha_{1|0},\ldots,\alpha_{T|0}$ that satisfy constraints (S10c) and (S10d) is a feasible solution of (S10).

Given the optimization problem (S10), how can we now solve this problem in practice? Remembering "Empirical Quantiles as Linear Programs," we note that $\text{Quantile}_{1-\delta}(R^{(K+1)},\ldots,R^{(K+M)},\infty)$ in (S10a) can be written as a linear program following (S1), hence making the optimization problem (S10) a bilevel optimization problem. To address this issue, we rewrite the inner optimization problem into a set of equivalent linear complementarity constraints using the Karush–Kuhn–Tucker conditions of the inner optimization problem. Finally, we use a standard reformulation of the max operator into a set of equivalent mixed-integer constraints [100], making the optimization problem (S10) a mixed-integer linear complementarity program. The authors of [99] also showed that the integer constraints can be removed without losing optimality (see [99] for details). We provide a numerical evaluation of the statistical abstraction that is obtained in this way in Example 2.

The presented approach follows a two-step procedure requiring two independent datasets on which we solve the optimization problem in (S10) and apply Theorem 2. Recent work in [70] enables the use of PAC-Bayes theory to obtain generalization bounds without splitting datasets. Theoretical connections between the size of the prediction regions and the generalization properties of the underlying predictor were established in [S32].

REFERENCE

[S32] M. Zecchin, S. Park, O. Simeone, and F. Hellström, "Generalization and informativeness of conformal prediction," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Piscataway, NJ, USA: IEEE Press, 2024, pp. 244–249, doi: 10.1109/ISIT57864.2024.10619313.

[104], where quantile regressors are trained and calibrated for each agent, motivated by the success of conformalized quantile regression (CQR), as discussed in "Heteroskedasticity and Conformal Prediction."

More generally, and not within a multiagent context, we can use learned kernel density estimators, for example, estimators $\hat{p}(e_t)$ of the probability density function of e_t (these estimators may even be conditioned on $e_0,\ldots,e_{t-1}$). We then obtain the statistical abstraction $\{\bar{e}_t \in \mathbb{R}^{n_e} | -\hat{p}(\bar{e}_t) \leq C\}$, where C is the empirical quantile of the nonconformity score $R^{(i)} := -\hat{p}(e_t^{(i)})$; for example, see [105], [106], [107], and [108]. Such approaches capture complex spatial shapes and multimodality in the distribution of e_t. However, computing the set $\{\bar{e}_t \in \mathbb{R}^{n_e} | -\hat{p}(\bar{e}_t) \leq C\}$ can be computationally challenging. Conceptually similar to the work in [99], which we summarized in "Normalization Constants via Mixed-Integer Linear Complementarity Programming," the authors of [109] integrate the shape of the abstraction directly into the loss function when training a neural network. However, this may result in abstractions with complex shapes that are not amenable for downstream formal verification and control. Hence, one may instead be interested in obtaining efficient shapes, for example, by fitting convex shapes, such as ellipsoids or polytopes, to the error $\|e_\tau - \hat{e}_{\tau|t}\|$; see [65]. Other work looked at the problem of multivariate regression; see [110]. More generally, minimum volume abstractions (also referred to as prediction or covering sets) for multimodal distributions were considered in [65], [111], and [112].

Other Techniques for Statistical Abstractions in Dynamic Environments

Constructing statistical abstractions in dynamic environments has received a lot of attention recently. The authors of [113] propose an alternative approach by learning and calibrating copulas, which are cumulative distribution functions of $e_1,\ldots,e_T$ that have a marginal representation of e_t for each $t \in \{1,\ldots,T\}$. The authors of [114] showed how to construct statistical abstractions of dynamical systems with aleatoric and epistemic uncertainty. In [53], adaptive conformal prediction techniques (recall "Conformal Prediction Under Distribution Shift") are combined with nonconformity scores similar to (18) and (20) to obtain abstractions of the form (14) and (15) that adapt to "difficult-to-predict" trajectories. All of the aforementioned approaches require calibration datasets of multiple trajectories as per Assumption 2. This limitation

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

was addressed in [115] within a single trajectory approach under the assumption that samples of e_t converge to a stationary distribution and are "approximately" independent when sufficiently separated in time. To compute statistical abstractions, our exposition has focused on deterministic trajectory predictors μ. Without much change, we can incorporate stochastic trajectory predictors, for example, generative and probabilistic predictors, such as diffusion models, Kalman filters, or decision transformers. The authors of [116] and [117] present a nonconformity score that takes multiple predictions into account by computing the minimum distance between these predictions and the ground truth value. Such an approach can capture multimodal behavior, for example, robots that may turn left or right.

Safe Control in Dynamic Environments

We can now design a control sequence $u_0, \ldots, u_{T-1}$ that solves Problem 1. Motivated by the statistical abstractions in (14) and (15), we define the uncertainty sets

$$\mathcal{B}_{\tau|t} := \{\bar{e}_\tau \in \mathbb{R}^{n_e} \mid \| \bar{e}_\tau - \hat{e}_{\tau|t} \| \leq C_{\tau|t}\}.$$

In the remainder of this article, we follow [98] to design a receding horizon control strategy in which we solve an optimization problem at each time t to obtain the control input u_t.

The Optimization Problem

The optimization problem will use all information available at time t and output a sequence of control inputs $u_{t|t}, \ldots, u_{T-1|t}$ and states $x_{t+1|t}, \ldots, x_{T|t}$. We then apply the first control input $u_t := u_{t|t}$ to the system. Specifically, for a prediction horizon of $H > 0$, consider the optimization problem

$$\min_{(u_{t|t}, \ldots, u_{T-1|t})} J(x,u) \tag{23a}$$

$$\text{subject to } x_{\tau+1|t} = f_x(x_{\tau|t}, u_{\tau|t}), \quad \tau \in \{t, \ldots, T-1\} \tag{23b}$$

$$x_{t|t} = x_t \tag{23c}$$

$$\min_{\bar{e}_\tau \in \mathcal{B}_{\tau|t}} c(x_{\tau|t}, \bar{e}_\tau) \geq 0, \quad \tau \in \{t+1, \ldots, t+H\} \tag{23d}$$

$$u_{\tau|t} \in \mathcal{U}, x_{\tau+1|t} \in \mathcal{X}, \quad \tau \in \{t, \ldots, T-1\} \tag{23e}$$

where $J : \mathbb{R}^{n_x(T+1)} \times \mathbb{R}^{n_u T} \to \mathbb{R}$ is a user-defined cost function that is defined on the state and input trajectories

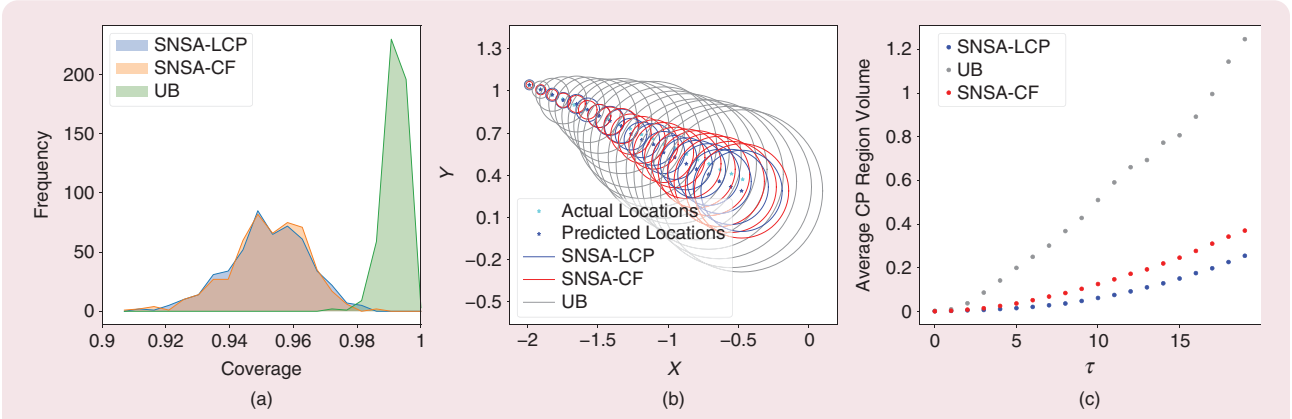


FIGURE 3 Experimental results for Example 2. (a) Histogram of $\|e_\tau - \hat{e}_{\tau|0}\| \leq C_{\tau|0}$, $\forall \tau \in \{1, \ldots, T\}$ over all N experiments. (b) Conformal prediction regions. (c) Average area of the prediction regions. CP: conformal prediction.

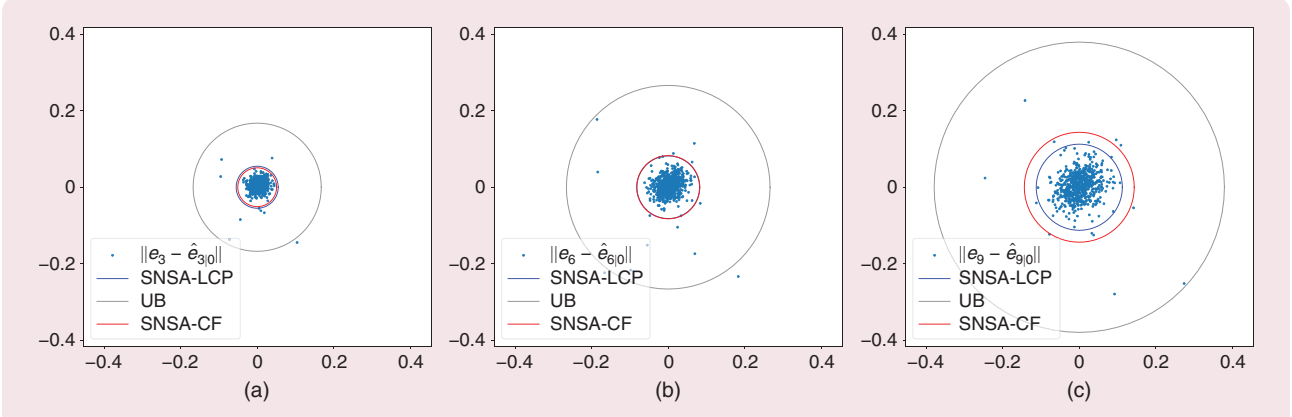


FIGURE 4 Conformal prediction regions and errors for various prediction horizons for Example 2. (a) $\|e_3 - \hat{e}_{3|0}\|$. (b) $\|e_6 - \hat{e}_{6|0}\|$. (c) $\|e_9 - \hat{e}_{9|0}\|$.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

$x := (x_0, \ldots, x_t, x_{t+1|t}, \ldots, x_{T|t})$ and $u := (u_0, \ldots, u_{t-1}, u_{t|t}, \ldots, u_{T-1|t})$, that is, over past and future states and control inputs. The sets $\mathcal{U} \subseteq \mathbb{R}^{n_u}$ and $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ are input and state constraints that we can add to our problem. While the cost function as well as the dynamics, state, and input constraints in (23a), (23b), and (23e) are standard in finite-horizon optimal control problems, the safety constraint in (23d) is new and arises as the dynamic environment e is uncertain. In fact, it is a robust safety constraint that enforces that the constraint c at time τ is enforced for all environment states within the abstraction $\mathcal{B}_{\tau|t}$. This, in turn, will guarantee that $\text{Prob}(c(x_{\tau|t}, e_\tau) \geq 0) \geq 1 - \delta$. However, the robust safety constraint makes the optimization problem (23) a bilevel optimization problem. If the constraint function c is Lipschitz continuous in its second argument, with Lipschitz constant L, we can instead use a computationally more efficient reformulation as

$$c(x_{\tau|t}, \hat{e}_{\tau\,|\,t}) \geq L C_{\tau|t},\ \tau \in \{t+1, \ldots, t+H\}. \tag{24}$$

It is easy to verify that satisfaction of the constraint in (24) is a sufficient condition for satisfaction of the constraint in (23d); see, for example, [98, Theorem 2] for intuition. For convenience, we summarize this optimization problem as

$$\min_{(u_{t|t}, \ldots, u_{T-1|t})} J(x, u) \tag{25a}$$

$$\text{subject to } x_{\tau+1|t} = f_x(x_{\tau|t}, u_{\tau|t}), \qquad \tau \in \{t, \ldots, T-1\} \tag{25b}$$

$$x_{t|t} = x_t \tag{25c}$$

$$c(x_{\tau|t}, \hat{e}_{\tau|t}) \geq L C_{\tau|t}, \qquad \tau \in \{t+1, \ldots, t+H\} \tag{25d}$$

$$u_{\tau|t} \in \mathcal{U}, x_{\tau+1|t} \in \mathcal{X}, \qquad \tau \in \{t, \ldots, T-1\}. \tag{25e}$$

Open-Loop and Closed-Loop Controllers

Having formulated the optimization problems in (23) and (25), we obtain a simple and easy-to-understand open-loop controller at time $t := 0$ if we set $H := T$.

Theorem 3

Given a test trajectory $e^{(0)} \sim \mathcal{D}_e$ and the statistical abstraction (14), then a solution $u_\tau := u_{\tau|0}$ of the optimization problem (23) [or (25) if c is Lipschitz continuous] at time $t := 0$ with $H := T$ is such that

$$\text{Prob}(c(x_t, e_t^{(0)}) \geq 0,\ \forall t \in \{1, \ldots, T\}) \geq 1 - \delta.$$

The proof of this result follows immediately and is intuitive by the construction of the statistical abstractions in (14). While this result provides a solution to Problem 1, Theorem 3 assumes that the optimization problem in (23) [or (25)] is feasible at time $t := 0$. This may not always be the case, especially for long task horizons T, inaccurate trajectory predictors μ, or distributions $\mathcal{D}_e$ with larger variance which may lead to large values of $C_{\tau|0}$, rendering the constraint (23d) [or (25d)] more restrictive. Another direct consequence of this is that the open-loop controller may be conservative. As motivated in the beginning, a better approach is a receding horizon control strategy.

Theorem 4

Given a test trajectory $e^{(0)} \sim \mathcal{D}_e$ and the statistical abstraction (15), then the solutions $u_t := u_{t|t}$ of the optimization problem (23) [or (25) if c is Lipschitz continuous] at times $t \in \{0, \ldots, T-1\}$ with $H > 0$ are such that

$$\text{Prob}(c(x_t, e_t^{(0)}) \geq 0,\ \forall t \in \{1, \ldots, T\}) \geq 1 - \delta.$$

Proof

By assumption, the optimization problem (23) [or (25)] is feasible at each time $t \in \{0, \ldots, T-1\}$ resulting in the control input $u_t := u_{t|t}$. As the control input is recomputed and applied iteratively, only the constraint (23d) [or (25d)] for $\tau = t+1$, for example, one-step ahead, becomes relevant. In fact, from constraint (23d) [or (25d)] and the statistical abstraction (15), it then follows that $\text{Prob}(c(x_{t+1|t}, e_{t+1}) \geq 0, \forall t \in \{0, \ldots, T-1\}) \geq 1 - \delta$. This implies that $\text{Prob}(c(x_{t+1}, e_{t+1}) \geq 0,\ \forall t \in \{0, \ldots, T-1\}) \geq 1 - \delta$, proving the main result. □

The result in Theorem 4 is more subtle than that in Theorem 3 as it only requires statistical abstractions for one-step-ahead predictions, for example, abstractions of the form (15). The intuition of the proof is that the control input u_t is computed and applied to the system iteratively at each time $t \in \{0, \ldots, T-1\}$, enforcing that at the next time $t+1$ the constraint $\text{Prob}(c(x_{t+1}, e_{t+1}) \geq 0) \geq 1 - \delta$ holds. This insight motivates greedy control strategies by selecting small prediction horizons H, as long as the optimization problem remains feasible (more on this next). Additionally, this motivates versions of the optimization problem in (23) [or (25)] that incorporate the knowledge that prediction regions $C_{\tau|t}$ for $\tau > t+1$ will be updated until $\tau = t+1$, which is the time when $C_{\tau|t}$ becomes relevant for obtaining the guarantees in Theorem 4. In other words, instead of the constraint in (23d), we can enforce

$$\min_{\bar{e}_\tau \in \bar{\mathcal{B}}_{\tau|t}} c(x_{\tau|t}, \bar{e}_\tau) \geq 0,\ \tau \in \{t+1, \ldots, t+H\}$$

$$\bar{\mathcal{B}}_{\tau|t} := \{\bar{e}_\tau \in \mathbb{R}^{n_e} \,\|\, \|\bar{e}_\tau - \hat{e}_{\tau|t}\| \leq C_{\tau|\tau-1}\}$$

without losing the guarantees in Theorem 4. Note that here we simply used the uncertainty set $\bar{\mathcal{B}}_{\tau|t}$ instead of $\mathcal{B}_{\tau|t}$ by using $C_{\tau|\tau-1}$ instead of $C_{\tau|t}$. In practice, the set $C_{\tau|\tau-1}$ is expected to be less conservative than $C_{\tau|t}$. Similarly, instead of the constraint in (25d), we can enforce

$$c(x_{\tau|t}, \hat{e}_{\tau|t}) \geq L C_{\tau|\tau-1},\ \tau \in \{t+1, \ldots, t+H\}.$$

Recursive Feasibility

Theorem 4 assumes that the optimization problem (23) [or (25)] is feasible at each time $t \in \{0, \ldots, T-1\}$. Recursive feasibility ensures that a feasible optimization problem at the initial time $t := 0$ implies feasibility of the optimization problem for all future times $t > 0$. Standard receding horizon control approaches with time-invariant state

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

constraints enforce recursive feasibility using terminal constraints and costs (see, for example, [24] and [118]. However, enforcing recursive feasibility of the optimization problem (23) [or (25)] is challenging as the constraint set depends on the a priori unknown predictions $\hat{e}_{\tau|t}$. While deriving a full solution to this issue is an open problem, recursive feasibility can be enforced in this setting for shrinking horizon control schemes; see [119] for details. In a shrinking horizon control scheme, we solve the optimization problem (23) [or (25)] at time $t := 0$ with $H := T$, and then continue solving (23) [or (25)] iteratively at times $t > 0$ by shrinking the task and prediction horizons T and H, respectively. In practice, recursive feasibility can also be achieved by adding slack variables to the constraints in (23d) [or (25d)].

We next illustrate the difference between open-loop and closed-loop controllers with an example taken from [98], and we refer to [98], [103], and [120] for more case studies.

Example 3

The case study in [98] considers a mobile robot, modeled using bicycle dynamics, that has to avoid three pedestrians, modeled by a distribution $e \sim \mathcal{D}_e$ that is implicitly defined by the ORCA simulator [101]. Statistical abstractions of the form (14) and (15) are obtained using a long short-term memory network and the "naive approach" from the section "Statistical Abstractions of Dynamic Environments." For one experiment, Figure 5 shows the results of the open-loop controller (bottom) and the closed-loop controller (top). The statistical abstractions are shown in green and are safely avoided in both cases. As the abstraction in the closed-loop case is updated iteratively, the robot trajectory is less conservative than that in the open-loop case. We refer the reader to [98] for a detailed empirical analysis of the statistical validity of the approach.

Before moving on to the next section, we discuss how the presented approach can be generalized to environments that consist of multiple agents and to the case where there is interaction between the x and e subsystems.

Dealing With Multiple Agents

The environment state e_t may have structure that we can leverage to obtain more accurate and informative prediction sets and to improve efficiency in solving the optimization problem (23). One such case is that in which the environment consists of N uncontrollable dynamic agents, for example, remote-controlled robots or humans. In this case, the environment state is described by the vector $e_t := (e_{t,1}, \ldots, e_{t,N})$, where $e_{t,n}$ denotes the state of agent n. Correspondingly, let us denote by $\hat{e}_{\tau|t,n}$ the prediction of $e_{\tau,n}$, for example, the state of agent n at time τ, made at time t. With minor modification of the nonconformity scores (18) and (20), we define the nonconformity scores

$$R^{(i)}_{\mathrm{OL}} := \max_{(\tau,n)\in\{1,\ldots,T\}\times\{1,\ldots,N\}} \alpha_{\tau|0,n} \,\|\, e^{(i)}_{\tau,n} - \hat{e}^{(i)}_{\tau|0,n} \,\| \tag{26}$$

$$R^{(i)}_{\mathrm{CL}} := \max_{(\tau,n)\in\{0,\ldots,T-1\}\times\{1,\ldots,N\}} \alpha_{t+1|t,n} \,\|\, e^{(i)}_{t+1,n} - \hat{e}^{(i)}_{t+1|t,n} \,\| \tag{27}$$

for the open-loop and the closed-loop controllers. Compared to (18) and (20), we additionally take the maximum over all individual agent prediction errors. The normalization constants $\alpha_{\tau|0,n} > 0$ and $\alpha_{t+1|t,n} > 0$ can be found following the same idea as discussed before. Similarly to Theorem 2, we then obtain the statistical abstractions

$$\begin{aligned}&\mathrm{Prob}\,(\| e_{\tau,n} - \hat{e}_{\tau|0,n} \| \le C_{\tau|0,n},\\ &\qquad \forall (\tau,n) \in \{1,\ldots,T\}\times\{1,\ldots,N\}) \ge 1-\delta\\ &\mathrm{Prob}\,(\| e_{t+1,n} - \hat{e}_{t+1|t,n} \| \le C_{t+1|t,n},\\ &\qquad \forall (t,n) \in \{0,\ldots,T-1\}\times\{1,\ldots,N\}) \ge 1-\delta\end{aligned}$$

where $C_{\tau|0,n} := C_{\mathrm{OL}}/\alpha_{\tau|0,n}$ and $C_{t+1|t,n} := C_{\mathrm{CL}}/\alpha_{t+1|t,n}$ with $C_{\mathrm{OL}} := \mathrm{Quantile}_{1-\delta}(R^{(1)}_{\mathrm{OL}}, \ldots, R^{(K)}_{\mathrm{OL}}, \infty)$ and $C_{\mathrm{CL}} := \mathrm{Quantile}_{1-\delta}(R^{(1)}_{\mathrm{CL}}, \ldots, R^{(K)}_{\mathrm{CL}}, \infty)$ using the nonconformity scores (26) and (27), respectively. Following this construction, we then use the structured uncertainty sets

$$\mathcal{B}_{\tau|t} := \{\bar{e}_\tau \in \mathbb{R}^{n_e} \,|\, \forall n \in \{1,\ldots,N\}, \| \bar{e}_{\tau,n} - \hat{e}_{\tau\,|\,t,n} \| \le C_{\tau|t,n}\}$$

within the optimization problem (23). The structure that is now present in $\mathcal{B}_{\tau|t}$ usually simplifies solving (23), as the constraint c is defined in terms of individual agent states in practice, which allows rewriting the constraint (23d).

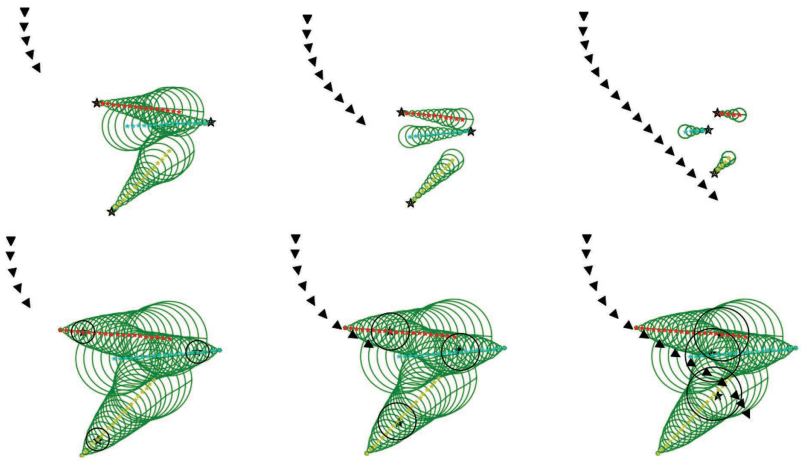


FIGURE 5 The top (bottom) three figures show the closed-loop (open-loop) robot trajectory as black triangles at times 4, 8, and 15. The actual pedestrian trajectories are indicated by black stars. The trajectory predictions are indicated by red, blue, and yellow stars with the corresponding uncertainty sets shown in green. Figure taken from [98].

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

Dealing With Agent Interactions

So far, we have considered the system in (13), where the behavior of the environment e does not depend on the behavior of the control system x and hence not on the control input u. In some applications, however, there may be interaction in the sense that the distribution $\mathcal{D}_e$ of the environment e depends on the trajectory x such that $e \sim \mathcal{D}_e(x)$. For instance, in robot navigation, the environment state e_t may follow the dynamics $e_{t+1} = f_e(e_t, v_t) + g_e(e_t, x_t)$, where the function $g_e : \mathbb{R}^{n_e} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_e}$ models (social) interaction, such as repulsive forces for collision avoidance [121], [122], [123], [124]. In such cases, the interaction term $g_e(e_t, x_t)$ is usually small when the safety constraint c imposes conservative safety margins, such as a self-driving car being forced to take conservative control actions that result in socially acceptable trajectories that do not change the behavior of pedestrians. In these situations, we do not expect our control guarantees to degrade dramatically. Indeed, we can measure the effect of, and even compensate for, interaction-induced shifts in the distribution of $\mathcal{D}_e$. Consider the scenario where calibration trajectories $e^{(1)}, \ldots, e^{(K)}$ are sampled from a nominal distribution $\mathcal{D}_e$, as per Assumption 2, while the test trajectory $e^{(0)}$ is sampled from the distribution $\mathcal{D}_e(x)$ that models interaction with x. For instance, the distribution $\mathcal{D}_e$ may describe the dynamics $e_{t+1} = f_e(e_t, v_t)$, while the distribution $\mathcal{D}_e(x)$ may describe the dynamics $e_{t+1} = f_e(e_t, v_t) + g_e(e_t, x_t)$. For this scenario, the authors of [125] provide a lower bound of $1 - \delta - \tilde{\delta}$ for the statistical coverage, where $\tilde{\delta}$ is a function of, and monotonically increases with, the Kullback–Leibler (KL) divergence between $\mathcal{D}_e$ and $\mathcal{D}_e(x)$. An alternative approach would be to construct robust statistical abstractions when the f-divergence between $\mathcal{D}_e$ and $\mathcal{D}_e(x)$ is upper bounded (or known) by using robust conformal prediction, see "Conformal Prediction Under Distribution Shift." If the interaction-induced distribution shift is difficult to measure, adaptive conformal prediction can be used to design interaction-aware controllers, as for example, in [126], [127], and [128].

Temporal Logic-Constrained Control

So far, we have considered constraint functions c that were enforced pointwise in time. Such constraints can enforce safety, for example, avoiding collisions with other agents at all times. We may, however, be interested in enforcing more complex trajectory-wise safety and performance constraints, such as repeatedly tracking a specific agent or tracking agents that enter forbidden areas. Such constraints can be expressed as temporal logic specifications over the outputs x and e of the LEAS. We here use signal temporal logic (STL), which extends predicate logic, introduced in "LEC Specifications in Predicate Logic," by additionally reasoning over temporal properties. We refer readers not familiar with STL to "LEAS Specifications in STL" for more details and examples. We are now interested in solving the following problem.

Problem 2

Given the system in (13) and an STL specification ϕ over the trajectories of x and e, design control inputs $u_0, \ldots, u_{T-1}$ such that x satisfies ϕ with a probability no less than $1 - \delta$, such that

$$\text{Prob}((x, e) \vDash \phi) \geq 1 - \delta. \tag{28}$$

It was shown in [103] that Problem 2 can be solved in two different ways by using the Boolean semantics or the quantitative semantics associated with the specification ϕ (see "LEAS Specifications in STL" for details). We summarize the Boolean semantics-based control approach next, hereby closely following [103].

Mixed-Integer Linear Encoding

STL specifications can be encoded as constraints within a mixed-integer linear program [129], [130]. The idea is to introduce a binary variable $z_0^\phi \in \{0, 1\}$ and a set of constraints over (x, e) such that $z_0^\phi = 1$ if and only if $(x, e) \vDash \phi$. Since the environment e is stochastic, this encoding cannot be used here. Instead, we introduce a binary variable $\bar{z}_0^\phi \in \{0, 1\}$ and a set of constraints over x and the statistical abstractions such that $\bar{z}_0^\phi = 1$ implies that $\text{Prob}((x, e) \vDash \phi) \geq 1 - \delta$. This encoding provides sufficient conditions, and we remark that necessary conditions cannot be derived here because of the nonuniqueness of statistical abstractions.

Recall that an STL specification ϕ is recursively built from predicates μ using Boolean and temporal operators. We thus derive constraints encoding ϕ recursively.

Predicates

For each predicate μ in ϕ and for each time $\tau \geq 0$, we introduce a binary variable $\bar{z}_{\tau|t}^\mu \in \{0, 1\}$. If $\tau \leq t$, then we have observed the value of e_τ already, and we set $\bar{z}_{\tau|t}^\mu = 1$ if $\mu(x_\tau, e_\tau) \geq 0$. If $\tau > t$, then we would like to constrain $\bar{z}_{\tau|t}^\mu$ such that $\bar{z}_{\tau|t}^\mu = 1$ implies $\text{Prob}(\mu(x_{\tau|t}, e_\tau) \geq 0) \geq 1 - \delta$, while $\bar{z}_{\tau|t}^\mu = 0$ implies $\text{Prob}(\mu(x_{\tau|t}, e_\tau) < 0) \geq 1 - \delta$. We achieve this objective by using the Big M method to define the constraints

$$-\min_{\bar{e} \in \mathcal{B}_{\tau|t}} \mu(x_{\tau|t}, \bar{e}) \leq M(1 - \bar{z}_{\tau|t}^\mu) \tag{29a}$$

$$\max_{\bar{e} \in \mathcal{B}_{\tau|t}} \mu(x_{\tau|t}, \bar{e}) \leq M\bar{z}_{\tau|t}^\mu - \epsilon \tag{29b}$$

where M and ϵ are sufficiently large and small positive constants, respectively (see [100] for details on how to select M and ϵ). The minimization of $\mu(x_{\tau|t}, \bar{e})$ over $\bar{e} \in \mathcal{B}_{\tau|t}$ accounts for all states within the statistical abstraction $\mathcal{B}_{\tau|t}$. Note that the integration of the constraints in (29) into an optimization problem that aims to compute $u_{0|t}, \ldots, u_{T-1|t}$ results in a bilevel optimization problem. The inner optimization problems $\min_{\bar{e} \in \mathcal{B}_{\tau|t}} \mu(x_{\tau|t}, \bar{e})$ and $\max_{\bar{e} \in \mathcal{B}_{\tau|t}} \mu(x_{\tau|t}, \bar{e})$ can be reformulated using its Karush–Kuhn–Tucker conditions when μ is a linear function; see [103] for details.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

LEAS Specifications in STL

Temporal logics, for example, linear temporal logic [S33] or STL [S34], are used to express complex system specifications. STL is a predicate logic equipped with additional temporal operators to reason over the temporal behavior of system trajectories. An STL specification ϕ is constructed from atomic constraint functions $h:\mathbb{R}^{n_z}\to\mathbb{R}$ that impose constraints of the form $h(z_t)\geq 0$. We also define the Boolean predicate $\mu:\mathbb{R}^{n_z}\to\{\text{True},\text{False}\}$ as $\mu(z_t)=\text{True}$ if $h(z_t)\geq 0$ and $\mu(z_t)=\text{False}$ otherwise. We define the syntax of STL over z as

$$\phi ::= \text{True} \,|\, \mu \,|\, \neg\phi' \,|\, \phi' \wedge \phi'' \,|\, \phi' U_I \phi'' \tag{S11}$$

where U_I denotes the until operator with time interval $I\subseteq\mathbb{R}_{\geq 0}$, while the other operators were already discussed in "LEC Specifications in Predicate Logic." Intuitively, the until operator $\phi' U_I \phi''$ encodes that ϕ' has to be true from now on until ϕ'' becomes true at some future time within the time interval I. We can further define eventually and always operators as

$$F_I\phi' := \top\, U_I \phi' \quad \text{(eventually)}$$
$$G_I\phi' := \neg F_I \neg\phi' \quad \text{(always)}.$$

Intuitively, the eventually $F_I\phi'$ (always $G_I\phi'$) operator encodes that ϕ' has to be true at some time (for all future times) within the time interval I. This intuition carries over into the semantics. Formally, to determine if an STL specification as defined in (S11) is satisfied, we define the Boolean semantics

$(z,t)\models \text{True}$ iff holds by definition,
$(z,t)\models \mu$ iff $h(z_t)\geq 0$
$(z,t)\models \neg\phi'$ iff $(z,t)\not\models \phi'$
$(z,t)\models \phi'\wedge\phi''$ iff $(z,t)\models\phi'$ and $(z,t)\models\phi''$
$(z,t)\models \phi' U_I \phi''$ iff $\exists t''\in (t\oplus I)\cap\mathbb{N}$ subject to $(z,t'')\models\phi''$ and $\forall t'\in[t,t'']\cap\mathbb{N}, (z,t')\models\phi'$.

Note that $(z,t)\models\phi$ indicates that the trajectory z satisfies ϕ at time step t. By convention, we use $z\models\phi$ when $t=0$.

To give a simple example, consider the specification

$$\phi := G_{[0,\infty)}\left(F_{I_1}\left\|x-e\right\|\leq\epsilon \wedge \left(\left\|e-A\right\|\leq\epsilon \Rightarrow F_{I_2}\left\|x-e\right\|\leq\epsilon\right)\right)$$

which expresses that the system x should repeatedly (with period I_1) track agent e, while x should immediately (within the time interval I_2) track e whenever e enters the forbidden region A.

We again define quantitative semantics $\rho^\phi(z,t)$ that reduce checking $(z,t)\models\phi$ to checking the positivity of ρ^ϕ. Let

$$\begin{aligned}
\rho^{\top}(z,t) &:= \infty\\
\rho^{\mu}(z,t) &:= h(z_t)\\
\rho^{\neg\phi'}(z,t) &:= -\rho^{\phi'}(z,t)\\
\rho^{\phi'\wedge\phi''}(z,t) &:= \min(\rho^{\phi'}(z,t),\rho^{\phi''}(z,t))\\
\rho^{\phi' U_I \phi''}(z,t) &:= \max_{t''\in(t\oplus I)\cap\mathbb{N}} \min(\rho^{\phi''}(z,t''), \min_{t'\in[t,t'']\cap\mathbb{N}} \rho^{\phi'}(z,t')).
\end{aligned}$$

As before, we use $\rho^\phi(z)$ when $t=0$ by convention.

STL specifications are defined over trajectories z with infinite length. In practice, however, one is often interested in bounded STL specifications, which are STL specifications where the time interval I in the STL syntax is bounded. In this survey, we limit ourselves to bounded STL specifications. To check satisfaction of a bounded STL specification ϕ, for example, to check that $(z,t)\models\phi$, only a signal z of finite length is required. The length of this signal, denoted by T^ϕ, can recursively be computed as

$$\begin{aligned}
T^{\text{True}} = T^{\mu} &:= 0\\
T^{\neg\phi} &:= T^{\phi}\\
T^{\phi'\wedge\phi''} &:= \max(T^{\phi'},T^{\phi''})\\
T^{\phi' U_I \phi''} &:= \max\{I\cap\mathbb{N}\} + \max(T^{\phi'},T^{\phi''}).
\end{aligned}$$

REFERENCES

[S33] A. Pnueli, "The temporal logic of programs," in *Proc. 18th Annu. Symp. Found. Comput. Sci. (SFCS)*, Washington, DC, USA, 1977, pp. 46–57.

[S34] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *Proc. Formal Techn., Model. Anal. Timed Fault-Tolerant Syst.*, Berlin, Heidelberg, Germany: Springer-Verlag, 2004, pp. 152–166.

Boolean and Temporal Operators

Boolean and temporal operators can recursively be encoded using the constraint encoding from [129] and [130], as we illustrate next.

For the negation $\phi := \neg\phi'$, we introduce a binary variable $\bar{z}^\phi_{\tau|t}\in\{0,1\}$ such that $\bar{z}^\phi_{\tau|t}=1$ if and only if $\bar{z}^{\phi'}_{\tau|t}=0$. For this, we simply enforce the constraint

$$\bar{z}^\phi_{\tau|t} = 1-\bar{z}^{\phi'}_{\tau|t}. \tag{30}$$

For the conjunction $\phi := \phi'\wedge\phi''$, we introduce a binary variable $\bar{z}^\phi_{\tau|t}\in\{0,1\}$ such that $\bar{z}^\phi_{\tau|t}=1$ if and only if $\bar{z}^{\phi'}_{\tau|t}=\bar{z}^{\phi'}_{\tau|t}=1$. We achieve this by enforcing the constraints

$$\bar{z}^\phi_{\tau|t} \leq \bar{z}^{\phi'}_{\tau|t} \tag{31a}$$

$$\bar{z}^\phi_{\tau|t} \leq \bar{z}^{\phi''}_{\tau|t'} \tag{31b}$$

$$\bar{z}^\phi_{\tau|t} \geq -1+\bar{z}^{\phi'}_{\tau|t}+\bar{z}^{\phi''}_{\tau|t}. \tag{31c}$$

For the temporal until operator, we follow a similar procedure. Specifically, note that we can rewrite the until operator $\phi=\phi'\mathbf{U}_{[a,b]}\phi''$ as

$$\bar{z}^\phi_{\tau|t} = \bigvee_{\tau'=\tau+a}^{\tau+b}\left(\bar{z}^{\phi''}_{\tau'|t}\wedge\bigwedge_{\tau''=\tau}^{\tau'}\bar{z}^{\phi'}_{\tau''|t}\right). \tag{32}$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

The conjunction and disjunction operators in (32) can then be encoded via the constraints (30) and (31) (for disjunctions, note that $\phi' \vee \phi'' = \neg(\neg\phi' \wedge \neg\phi'')$).

For an STL specification ϕ, this encoding now ensures that $\text{Prob}((x, e) \vDash \phi) \geq 1 - \delta$ if $\bar{z}^{\phi}_{0|t} = 1$. We next summarize this result and refer the reader to [103] for the proof.

Theorem 5

Given an STL specification ϕ over trajectories x and e and a recursive encoding $\bar{z}^{\phi}_{0|t}$ via the constraints in (29)–(32), then $\bar{z}^{\phi}_{0|t} = 1$ implies that $(x, e) \vDash \phi$ holds with a probability no less than $1 - \delta$. For example, it holds that

$$\text{Prob}((x, e) \vDash \phi) \geq 1 - \delta. \tag{33}$$

Control Synthesis

We can now solve Problem 2 by computing a sequence of control inputs $u_{t|t}, \ldots, u_{T-1|t}$ as the solution of the optimization problem

$$\min_{(u_{t|t}, \ldots, u_{T-1|t})} \quad J(x, u) \tag{34a}$$

$$\text{subject to} \quad x_{\tau+1|t} = f_x(x_{\tau|t}, u_{\tau|t}), \quad \tau \in \{t, \ldots, T-1\} \tag{34b}$$

$$x_{t|t} = x_t \tag{34c}$$

$$\bar{z}^{\phi}_{0|t} = 1 \tag{34d}$$

$$\text{encodings (29)–(32)} \tag{34e}$$

$$u_{\tau|t} \in \mathcal{U}, \; x_{\tau+1|t} \in \mathcal{X}, \quad \tau \in \{t, \ldots, T-1\}. \tag{34f}$$

Similar to Theorems 3 and 4, we can now derive open-loop and closed-loop controllers. We summarize these results, which follow from Theorem 5, next.

Theorem 6

Given a test trajectory $e^{(0)} \sim \mathcal{D}_e$ and the statistical abstraction (14), a solution $u_\tau := u_{\tau|0}$ of the optimization problem (34) at time $t := 0$ is such that

$$\text{Prob}((x, e) \vDash \phi) \geq 1 - \delta.$$

Theorem 7

Given a test trajectory $e^{(0)} \sim \mathcal{D}_e$ and the statistical abstraction (24), then the solutions $u_t := u_{t|t}$ of the optimization problem (34) at times $t \in \{0, \ldots, T-1\}$ are such that

$$\text{Prob}((x, e) \vDash \phi) \geq 1 - \delta.$$

We remark that recursive feasibility for the closed-loop controller in Theorem 7 is a more delicate yet interesting problem compared to recursive feasibility in Theorem 4. A detailed discussion is beyond the scope of this survey article, and we hence refer the interested reader to [103].

Control Synthesis Using Quantitative Semantics

In our previous exposition, we solved Problem 2 using the Boolean semantics associated with ϕ. However, Problem 2 can also be solved using the quantitative semantics $\rho^{\phi} : \mathbb{R}^{Tn_x} \times \mathbb{R}^{Tn_e} \to \mathbb{R}$. Indeed, [103] shows that one can design controllers such that $\text{Prob}(\rho^{\phi}(x, e) > 0) \geq 1 - \delta$, by which it follows that Problem 2 is solved. Importantly, one can now choose to maximize $\rho^{\phi}(x, e)$. While controllers obtained this way are more robust, they are at the same time also computationally more demanding.

Finally, we invite the interested reader to find illustrative simulation examples in [103].

State Estimation and Perception

Up until now, we assumed to have knowledge of the state $z_t := (x_t, e_t)$ via perfect sensor measurements $y_t = z_t$. Autonomous systems, however, are complex systems that operate in uncertain environments where the states x_t and e_t cannot be observed directly. These states are usually estimated from noisy sensor measurements $y_t = p(z_t, w_t)$, where $p : \mathbb{R}^{n_z} \times \mathbb{R}^{n_w} \to \mathbb{R}^{n_y}$ is the sensor model and $w_t \sim \mathcal{D}_w$ is sensor noise. A common way of obtaining state estimates is via state estimation algorithms [18]. However, as discussed in "Conformalizing the Kalman Filter," state estimation algorithms are either computationally prohibitive (for example, the particle filter) or only valid under restrictive assumptions on the system (for example, the Kalman filter). This motivated us to conformalize the Kalman filter when applied to systems for which no guarantees can be obtained directly. Many of these classic state estimation algorithms require knowledge of the system dynamics f, for example, the f_x and f_e dynamics of the x and e subsystems, and the sensor model p. However, such knowledge is often difficult to obtain in practice; for example, mathematical models for the motion of pedestrians or for camera sensors are not available. This limitation has further motivated recent work on learning-enabled perception and state estimation algorithms when using complex camera, lidar, and radar sensors [131], [132]. Using ideas similar to those presented in "Conformalizing the Kalman Filter," we can conformalize even such complex perception and state estimation algorithms.

Statistical Perceptual Abstractions

The complexity of learning-enabled perception and state estimation algorithms has led to the introduction of more general statistical abstractions, here broadly referred to as perceptual abstractions. However, there is another reason why one should be interested in more general abstractions than the ones obtained in "Conformalizing the Kalman Filter." To see why, note that the observations $y_t = p(x_t, e_t)$ depend on x_t and hence also on the control inputs $u_0, \ldots, u_{t-1}$.[19] However, if

[19] "Conformalizing the Kalman Filter" assumed access to a trajectory calibration dataset, for example, trajectories of the autonomous system driven by the control inputs $u_0, \ldots, u_T$.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

the control inputs u_0, u_{t-1} depend on the statistical abstraction, which is likely to be the case and the reason why we design an abstraction in the first place, then the nonconformity scores evaluated over the calibration dataset are not independent as before. As a consequence, the statistical abstraction is no longer valid. Another way of thinking about this is that changing the sequence of control inputs leads to a distribution shift in the distribution of the nonconformity score. To address this challenge, various authors introduced different notions of perceptual abstractions; for example, see [133], [134], [135], [136], and [137].

In what follows, let the state $z \in \mathbb{R}^{n_z}$ be unknown. For a perception and state estimation algorithm that computes an estimate $\hat{z}(y)$ of z from a measurement $y = p(z, w)$ where $w \sim \mathcal{D}_w$, a perceptual abstraction is a set-valued function $\mathcal{P} : \mathbb{R}^{n_y} \times \mathbb{R}^{n_z} \to 2^{\mathbb{R}^{n_z}}$ such that $z \in \mathcal{P}(y, \hat{z}(y))$.[20] As can easily be seen, constructing perceptual abstractions is challenging because of the complexity of the system, the sensors, and the learning-enabled perception and state estimation algorithms. This challenge motivates the definition of statistical perceptual abstractions, for which it holds that

$$\text{Prob}\,(z \in \mathcal{P}(y, \hat{z}(y))) \geq 1 - \delta.$$

Such guarantees can be obtained with conformal prediction for compact domains $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$ in the state space when we are able to generously sample multiple calibration datasets from $\mathcal{Z}$. The idea is to construct a grid of $\mathcal{Z}$ and to perform a conformal prediction step for each grid point. Specifically, we construct an ϵ-net $\bar{\mathcal{Z}}$ of $\mathcal{Z}$, where $\epsilon > 0$ is a gridding parameter; i.e., we construct a finite set $\bar{\mathcal{Z}}$ so that for each $z \in \mathcal{Z}$ there exists $z_j \in \bar{\mathcal{Z}}$ such that $\|z - z_j\| \leq \epsilon$. For this purpose, gridding or randomized algorithms can be used that sample from $\mathcal{Z}$ [55]. We can now apply a conformal prediction step for each grid point $z_j \in \bar{\mathcal{Z}}$. That is, we compute the nonconformity score

$$R_j^{(i)} := \|\hat{z}(y_j^{(i)}) - z_j\| \tag{35}$$

over a calibration dataset $y_j^{(i)} := p(z_j, w^{(i)})$ for $i \in \{1, \ldots, K\}$, where $w^{(i)}$ is independently sampled from $\mathcal{D}_w$, for example, $w^{(i)} \sim \mathcal{D}_w$. For a test datapoint $y_j^{(0)} := p(z_j, w^{(0)})$ with $w^{(0)} \sim \mathcal{D}$, we immediately know that

$$\text{Prob}\,(\|\hat{z}(y_j^{(0)}) - z_j\| \leq C_j) \geq 1 - \delta \tag{36}$$

by the choice of $C_j := \text{Quantile}_{1-\delta}(R_j^{(1)}, \ldots, R_j^{(K)}, \infty)$. We note that (36) quantifies uncertainty of the uncertain estimate $\hat{z}(y_j^{(0)})$ as we fixed the state z_j a priori.

To revert this argument and obtain a prediction region for an unknown state $z \in \mathcal{Z}$ from an estimate $\hat{z}(y)$, where $y = p(z, w)$ is a noisy measurement with $w \sim \mathcal{D}_w$, we have to ensure that (36) holds for all states $z \in \mathcal{Z}$ (instead of only $z_j \in \mathcal{Z}$). To do so, the authors of [138] use the properties of the ϵ-net $\bar{\mathcal{Z}}$ along with assumptions on the continuity of the sensor map p and the estimate $\hat{z}$. Specifically, if p and $\hat{z}$ have Lipschitz constants $\mathcal{L}_p$ and $\mathcal{L}_{\hat{z}}$, respectively, then for any state $z \in \mathcal{Z}$ it holds that

$$\text{Prob}(\|\hat{z}(y) - z\| \leq \sup_j C_j + (\mathcal{L}_p \mathcal{L}_{\hat{z}} + 1)\epsilon) \geq 1 - \delta.$$

Variations of statistical perceptual abstractions were obtained in [133], [134], [135], [136], and [137] using similar reasoning via combinations of ϵ-nets and statistical techniques, such as Hoeffding's inequality. Statistical perceptual abstractions enable the design of robust control techniques that take perceptual uncertainty into account. While initial progress was made, as for example, in [136], [137], [138], [139], and [140], the design of controllers with end-to-end safety guarantees is a challenging and open problem. Finally, we note that existing work assumes that ground truth information of z_j is available in the form of calibration data. In practice, such an assumption is limiting unless precise state estimates can be obtained or when realistic data with ground truth information can be collected, for example, from photorealistic simulators. In these cases, the sim2real gap can be addressed using robust uncertainty quantification techniques, such as robust conformal prediction (see "Conformal Prediction Under Distribution Shift").

Beyond State Estimation

So far, we have focused on the design of statistical perceptual abstractions for state estimation and downstream control. However, perception uncertainty can be quantified and incorporated at the task planning level. One line of work considers large language models and conformal prediction for planning in uncertain environments. The authors of [141] use large language models to provide preferences for a set of high-level actions, for example, "go to region A and then B", "go to region B and then A", …. Conformal prediction is here used to constructs sets that contain all actions that conform with the natural language specification of the user with a given confidence. In [142], the same authors explore a setting in which an agent actively explores an uncertain environment to answer a question about the state of the environment. Large vision language models produce actions for exploration, and conformal prediction is used until an answer to the question about the environment is found with a desired confidence. Similarly to Theorem 2, the authors of [141] and [142] construct multistep prediction sets. Importantly, these works assume that the behavior of the environment is independent from the behavior of the control system, similarly to the system considered in (13). In this way, the focus is not directly on computing control inputs but on high-level task

[20]The definition of a perceptual abstraction can be extended to include sequences of measurements instead of one measurement.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

planning. By contrast, the authors of [143] consider a control system that depends on an environment that is drawn from a distribution of potential environments, and a method is proposed to modify an existing perception system using conformal prediction so that a given controller satisfies a desired task.

Related Work: Safe Control Techniques

There is a broad range of safe control techniques, commonly known under the umbrella of constrained control. Popular techniques include model predictive control [23], [24], [25], [144], control barrier functions [145], [146], [147], [148], [149], and funnel control [150], [151], [152], [153], with respective extensions to temporal logic specifications in [154], [155], [156], and [157], and also in [129], [130], and [158]. These control techniques are model based and typically only provide safety guarantees when the system dynamics are simple and known. Model-based controllers are thus hard to design for complex, possibly learning-enabled systems. This has motivated the use of data-driven and machine learning techniques, such as reinforcement learning [159], [160], imitation learning [161], [162], or learned safe controllers [136], [163], [164], [165], even with access to expert demonstrations [12], [166], [167]. While these techniques often demonstrate good performance in practice, they typically lack verifiable safety guarantees.

Toward designing verifiably safe controllers for such complex, often uncertain systems, recent efforts have focused on combining statistical techniques with data-driven control design. While data-driven control techniques are not new within the control field (consider system identification [168] and adaptive control [169], [170]), the integration with statistical techniques enables finite sample guarantees and sample complexity analysis; see [33], [34], and [171] for recent tutorials. As there are many recent developments in this direction, we proceed by first reviewing conformal prediction-based control techniques followed by surveying other statistical control techniques.

There has been a lot of recent work and progress in solving control problems in dynamic environments with conformal prediction. Along the same lines of what we presented in the sections "Statistical Abstractions of Dynamic Environments" and "Safe Control in Dynamic Environments," the authors of [120] design controllers using statistical abstractions via conformalized copulas, as originally presented in [113]. Conceptually similar is also the work in [172], in which an abstraction is built without the use of a trajectory predictor μ. In a different direction, the authors of [173] and [174] consider end-to-end optimization approaches that find statistical abstractions that are optimal for decision making. In a different setting, conformal prediction has been used to design predictive safety filters for reinforcement learning controllers in dynamic environments [175]. The underlying assumption in these works is the existence of an independent and identically distributed dataset as per Assumption 2. In [126], [127], [128], and [176], the authors relax this assumption and instead use adaptive conformal prediction (see "Conformal Prediction Under Distribution Shift") to build statistical abstractions. These abstractions do not come with guarantees of the form (14) [or (15)], but enjoy asymptotic probabilistic guarantees and again enable the design of open-loop and closed-loop controllers. In a similar direction, out-of-distribution detection for dynamic environments and control adaptation is presented in [177]. A version of adaptive conformal prediction, called intermittent quantile tracking, has also been used in the setting of interactive imitation learning [178].

While everything that we presented so far was modular by the use of statistical abstractions, the authors of [179] present conformal decision theory and calibrate decisions directly without constructing abstractions. The main idea is that the controller is parameterized by a risk parameter that is updated via an adaptive conformal prediction update rule. Importantly, this update is applied to the risk parameter, rather than to the quantile, as in adaptive conformal prediction. This method can guarantee that an accumulated loss is asymptotically bounded with probability one. In yet another direction, the works in [180], [181], and [182] propose conformal off-policy evaluation to compute prediction intervals for the value function of a reinforcement learning controller using calibration data gathered under a different controller. This requires dealing with the distribution shift between these two controllers, which is handled via weighted conformal prediction (see "Conformal Prediction Under Distribution Shift"). More recently, conformal prediction was used for the control design of (partially) known or uncertain systems, for example, for nonlinear systems with Lyapunov and barrier functions [183], [184], [185], [186], [187] or for linear systems [64], [188]. We also point the reader to the work in [189], in which robust controllers are designed based on conformal uncertainty estimates of learned system dynamics models. Multiagent control problems with conformal prediction were recently considered in [182], [190], [191], and [192]. Finally, we mention the works in [191], [193], and [194], which use conformal prediction to guarantee the safety of language-instructed and diffusion model-based planners, respectively.

A special instantiation of a dynamic environment is when humans are present. In [195], human intentions are predicted via latent intent prediction algorithms that are subsequently conformalized for interactive planning. The key idea here is to ask the human for help if the uncertainty, computed with conformal prediction, is too large. In a different setting, in [196], adaptive conformal prediction is used to obtain confidence estimates when mapping human inputs to high-dimensional robot actions, for example, in the case of assistive teleoperation of robotic manipulators.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

Before moving to the next section, we briefly summarize other statistical techniques for safe control design. Prominent techniques include the use of Gaussian process-based system modeling for control design (for example, see [197], [198], [199], and [200]) as well as scenario optimization for solving robust or chance-constrained control problems (for example, see [26], [27], [201], [202], and [203]). We also mention that PAC-Bayes theory was used in [204] and [205] for designing controllers that provably generalize to new environments. Finally, we point the reader to control design under risk constraints; for example, see [42], [43], [206], [207], and [208].

OFFLINE VERIFICATION OF LEASs WITH CONFORMAL PREDICTION

Model-based control design techniques usually provide correctness and safety guarantees under specific assumptions on the system. In the section "Control Synthesis for LEASs With Conformal Prediction," we surveyed a hybrid of model-based control techniques with statistical uncertainty quantification techniques. Along the way, we encountered various assumptions that may not, or only approximately, hold in practice, for example, having exact knowledge of the state or encountering dynamic environments whose behavior is independent of the control system. Violation of these assumptions necessitates offline and possibly online verification. See "Spectrum of Formal Verification and Control Techniques" for a distinction between offline and online verification. More generally, learning-enabled controllers may not come with any formal safety guarantees, for example, most reinforcement or imitation learning techniques. We thus present offline and online verification techniques with conformal prediction in the remainder of this survey.

Throughout the following sections, we consider LEASs as in (11) under a controller of the form (12). For convenience, we recall the system as

$$z_{t+1} = f(z_t, u_t, v_t) \tag{37a}$$

$$y_t = p(z_t, w_t) \tag{37b}$$

$$u_t = \pi(\{y_s\}_{s=0}^{t}) \tag{37c}$$

where we adapt the same notation as in the section "Control Synthesis for LEASs With Conformal Prediction." However, as opposed to that section, we now assume that the controller π is fixed and given to us, for example, an arbitrary learning-enabled controller that may not achieve its objective and has to be verified. Importantly, we do not need to limit ourselves to systems of the form (37) as long as we can collect a calibration dataset that consists of system trajectories (this will be formally stated in Assumption 3 next). Such trajectories can typically be collected from a simulator or the real system without having explicit knowledge of (37).

Challenges in LEAS Verification

The analysis of the system in (37) is challenging for two main reasons. First, the sensor map p is hard to model (for example, in the case of cameras), and a description of p is usually not available. In some cases, for example, in legged locomotion or social navigation tasks, we may even have issues identifying a dynamics map f. More crucially though, even if good models of f and p are available, the complexity of the LEAS makes the verification of the LEAS in (37) difficult and computationally challenging.

In the section "Challenges in LEC Verification," we provided a detailed discussion on challenges (specifically, scalability and conservatism) that model-based verification techniques face for LECs. These challenges further amplify when attempting to verify LEASs of the form (37) that iteratively use complex LECs (often even using LECs more complex than simple feedforward neural networks) within their perception–action loops because of compounding errors and increased computational complexity. Indeed, existing model-based verification techniques for LEASs extend complete and noncomplete LEC verification techniques and perform reachability analysis of the LEASs via semidefinite programming [80], [209], [210], Lyapunov/barrier functions [211], [212], and abstractions and hybrid system models [213], [214], [215], [216], [217], among others. While these techniques have been successful at verifying smaller problem instances, such as those for autonomous race cars [216], [218], cruise controller and emergency braking systems [214], [219], autonomous robots [220], or aircraft collision avoidance systems [221], [222], their applicability to real-world LEASs is still limited. Compositional techniques can alleviate the complexity challenges in larger verification problems; for example, see [223]. We emphasize, however, that model-based verification techniques are challenged on resource-constrained platforms and in online verification and control where efficient algorithms are needed, suggesting the use of efficient statistical techniques for probabilistically sound LEAS verification. Such an approach does not even require knowledge of the functions f and p.

Probabilistically Sound LEAS Verification

In the remainder, we verify input–output properties of LEASs using ideas that are conceptually similar to the ideas that we presented for verifying input–output properties of LECs in the section "LEC Verification With Conformal Prediction." Indeed, a LEAS can be thought of as having inputs $(z_0, \{v_t\}_{t=0}^{T-1}, \{w_t\}_{t=0}^{T-1})$ and outputs $z := (z_0, \ldots, z_T)$ with task horizon $T > 0$. In contrast to the section "LEC Verification With Conformal Prediction," we now want to reason over input and output trajectories u and z. By modifying (8), which we stated for LECs previously, we can instead formulate our goal for LEAS verification as

$$(z_0, v, w) \sim \mathcal{D} \implies \text{Prob}(z \vDash \phi_{\text{out}}) \geq 1 - \delta \tag{38}$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

where ϕ_{out} prescribes an output property for the trajectory z of (37). We assume that the output property ϕ_{out} is defined over the first $T+1$ values of z, for example, over $z_0, \ldots, z_T$. In (38), the input distribution is $\mathcal{D} := \mathcal{D}_z \times \mathcal{D}_v^T \times \mathcal{D}_w^T$, where $\mathcal{D}_v^T$ and $\mathcal{D}_w^T$ denote the T-fold probability measure of $\mathcal{D}_v$ and $\mathcal{D}_w$, respectively. Similarly to what we did previously, we assume that $\mathcal{D}$ has support over the input set $C_{\text{in}} := \{(z_0, v, w) \in \mathbb{R}^{n_z + T(n_v+n_w)} | (z_0, v, w) \vDash \phi_{\text{in}}\}$, where ϕ_{in} prescribes an input property for (z_0, v, w). Throughout this section, we make a similar assumption as Assumption 1 in the section "LEC Verification With Conformal Prediction" but now assuming that we have a set of independent trajectories from the system in (37).

Assumption 3

We have independently sampled a calibration dataset of K inputs from the distribution $\mathcal{D}$; that is, we have access to samples $(z_0^{(i)}, \{v_t^{(i)}\}_{t=0}^{T-1}, \{w_t^{(i)}\}_{t=0}^{T-1}) \sim \mathcal{D}$ for $i \in \{1, \ldots, K\}$. We use these to obtain a calibration dataset of K trajectories $z^{(i)} := (z_0^{(i)}, z_1^{(i)}, \ldots, z_T^{(i)})$ via the dynamics in (37).

Reachability of LEASs

Assume that we are given an input set $C_{\text{in}} \subseteq \mathbb{R}^{n_z + T(n_v+n_w)}$ and an output set $C_{\text{out}} \subseteq \mathbb{R}^{n_z(T+1)}$. By a slight abuse of notation, we let $C_{\text{out}}[t] \subseteq \mathbb{R}^{n_z}$ denote the projection of C_{out} onto the reachable set at time t. From here, we can essentially proceed similarly as in the section "Probabilistically Sound LEC Verification" (as summarized in Corollary 1) to verify the reachability of C_{out}, for example, to verify $\text{Prob}(z^{(0)} \in C_{\text{out}}) \geq 1 - \delta$, by defining the nonconformity score

$$R^{(i)} := \min_{t \in \{0, \ldots, T\}} \text{dist}(z_t^{(i)}, C_{\text{out}}[t]) \tag{39}$$

where $\text{dist}(\cdot)$ is a distance function defined in the same way as in the section "Probabilistically Sound LEC Verification" to measure the distance between the output z_t and the output set $C_{\text{out}}[t]$ at time t. By direct application of Lemma 1, we obtain the following result.

Corollary 3

Given an output set $C_{\text{out}} \subseteq \mathbb{R}^{n_z(T+1)}$, the distribution $\mathcal{D}$ with support over an input set $C_{\text{in}} \subseteq \mathbb{R}^{n_z + T(n_v+n_w)}$, and a test trajectory $z^{(0)} := (z_0^{(0)}, z_1^{(0)}, \ldots, z_T^{(0)})$ obtained via the dynamics in (37) from input samples $(z_0^{(0)}, \{v_t^{(0)}\}_{t=0}^{T-1}, \{w_t^{(0)}\}_{t=0}^{T-1}) \sim \mathcal{D}$, we have that

$$\text{Prob}\left(\min_{t \in \{0, \ldots, T\}} \text{dist}(z_t^{(0)}, C_{\text{out}}[t]) \leq C\right) \geq 1 - \delta$$

where $C := \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, \infty)$ with $R^{(i)}$ being defined in (39).

By the definition of the signed distance function and taking the minimum distance to C_{out} over all times $t \in \{0, \ldots, T\}$, we obtain a positive verification answer if $C < 0$. In this case, the absolute value of C again indicates how robustly the reachability specification is satisfied.

We also remark that the reachability analysis performed here was of a qualitative nature by checking if the trajectory $z^{(0)}$ remains within the set C_{out} with high confidence. Previously, in the section "Control Synthesis for LEASs With Conformal Prediction," we showed how to perform a quantitative analysis by computing a minimal set C_{out} that satisfies $\text{Prob}(z^{(0)} \in C_{\text{out}}) \geq 1 - \delta$.

General Verification of LEASs

We would like to verify more general input–output properties of LEASs than reachability, where we are conveniently given the input and output sets C_{in} and C_{out}. Such properties can again be expressed as STL specifications, and we remind the reader of "LEAS Specifications in STL" for an introduction. Similar to the predicate logic specifications in the section "Probabilistically Sound LEC Verification," the sidebar instructs how we can construct a real-valued performance function[21] $\rho^{\phi_{\text{out}}} : \mathbb{R}^{Tn_z} \to \mathbb{R}$ that satisfies the following soundness property:

$$\rho^{\phi_{\text{out}}}(z) > 0 \Longrightarrow z \vDash \phi_{\text{out}}.$$

As before, larger values of $\rho^{\phi_{\text{out}}}(z)$ are beneficial as they indicate robustness against perturbation in z; see [87] and [88].

From here, we can again proceed in the same way as in the section "Probabilistically Sound LEC Verification" (as summarized in Corollary 2) to verify an STL specification ϕ_{out}, for example, to verify $\text{Prob}(-C \leq \rho^{\phi_{\text{out}}}(z^{(0)})) \geq 1 - \delta$, which we briefly summarize next.

Corollary 4

Given an STL specification ϕ_{out} with sound performance function $\rho^{\phi_{\text{out}}} : \mathbb{R}^{n_y} \to \mathbb{R}$, the distribution $\mathcal{D}$, and a test trajectory $z^{(0)} := (z_0^{(0)}, z_1^{(0)}, \ldots, z_T^{(0)})$ obtained via the dynamics in (37) from input samples $(z_0^{(0)}, \{v_t^{(0)}\}_{t=0}^{T-1}, \{w_t^{(0)}\}_{t=0}^{T-1}) \sim \mathcal{D}$, we have

$$\text{Prob}(-C \leq \rho^{\phi_{\text{out}}}(z^{(0)})) \geq 1 - \delta.$$

where $C := \text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, \infty)$ with $R^{(i)} := -\rho^{\phi_{\text{out}}}(z^{(i)})$.

Finally, to illustrate what we learned so far, we present a case study on the verification of a cart–pole system with a learning-enabled feedback controller in "Verifying a Learning-Enabled Cart–Pole System."

Related Work: Statistical Verification Techniques

In the section "Challenges in LEAS Verification," we described computational challenges that arise in the verification of LEASs when using model-based approaches.

[21] The length of the task horizon T can easily be computed from the specification ϕ_{out} as instructed in "LEAS Specifications in STL."

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

Probabilistic and parametric model checking techniques (see [224], [225], [226], [227], and [228] and [229], [230], [231], and [232], respectively), enable the verification of stochastic LEASs but are subject to the same computational challenges. Statistical verification techniques, on the other hand, offer computationally tractable alternatives, for example, via conformal prediction, as demonstrated in the previous sections.

Reachability

Conformal prediction was first used for reachability analysis in [233], with extensions to partially observable systems [234] and distributed systems [235]. The idea here is to conformalize a learned predictor that predicts whether or not the system trajectory, starting from a given initial condition, is safe with respect to an unsafe set. While these

Verifying a Learning-Enabled Cart–Pole System

We want to verify the correctness of a learning-enabled state-feedback controller for a cart–pole system [S35], [S36]. The goal of the controller is to stabilize a pole placed on top of a frictionless cart. The state of the system is $z_t := (z_t^p, z_t^v, z_t^\theta, z_t^\omega) \in \mathbb{R}^4$, where z_t^p, z_t^v, z_t^θ, and z_t^ω indicate the cart position, cart velocity, angle of the pole (with respect to the upright position), and angular velocity of the pole, respectively. We assume that z_0 is uniformly sampled with $z_0 \sim \mathcal{U}((-0.05, 0.05)^4)$. We train a neural network controller [9] that outputs the force applied to the cart, here with a discrete action space: pushing the cart to the left or to the right. This controller can thus be viewed as a bang–bang controller. We simulate the cart–pole system in the OpenAI Gym [S37].

We want to verify that, at all times $t \le T := 228$, the angle of the pole z_t^θ deviates minimally from the upright position, and the cart stays close to the origin. For this specification, we can construct the following sound performance function:[a]

$$\rho^{\phi_{\text{out}}}(z) := \min_{t \in \{0,\dots,T\}} \left(\min\left(\delta_\theta - \left\| z_t^\theta \right\|, \delta_p - \left\| z_t^p \right\| \right) \right)$$

where $\delta_\theta := 0.2$ and $\delta_p := 4.5$ denote permissible thresholds for angle and position displacements, respectively.

We set the failure probability to $\delta := 0.05$ and use the nonconformity score $R^{(i)} := -\rho^{\phi_{\text{out}}}(z^{(i)})$, as discussed before. By this choice, we know that $\text{Prob}(-C \le \rho^{\phi_{\text{out}}}(z^{(0)})) \ge 1 - \delta$. We conduct $N := 500$ experiments for calibration set sizes of $K \in \{100, 300, 500\}$. Again, we first verify statistical validity empirically and compute the empirical coverage EC according to (3) as 0.95, 0.956, and 0.952, respectively. We also plot the conditional empirical coverage CEC_n according to (5) with $J := 400$ in Figure S4(a). We plot the histogram of the nonconformity scores of the calibration data from one experiment in Figure S4(b). To illustrate the verification results, we compute $\Sigma_{n=1}^N C_n/N$, which is the average of the bound C_n over all N experiments, where we recall from (3) and (5) that $C_n := C(R_n^{(1)}, \dots, R_n^{(K)})$. For $K \in \{100, 300, 500\}$, we obtain the values -0.11283, -0.11283, and -0.11277, respectively. We can thus conclude that the system satisfies the specification with a confidence no less than $1 - \delta$. In Figure S4(c), we also plot the histogram of C_n over all N experiments.

[a]The corresponding STL specification is written as $\phi_{\text{out}} \models G_{[0,T]}(\|z^\theta\| \le \delta_\theta \wedge \|z^p\| \le \phi_p)$.

REFERENCES

[S35] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-13, no. 5, pp. 834–846, Sep./Oct. 1983, doi: 10.1109/TSMC.1983.6313077.

[S36] R. V. Florian, "Correct equations for the dynamics of the cart-pole system," 2007. [Online]. Available: https://coneural.org/florian/papers/05_cart_pole.pdf

[S37] G. Brockman et al., "OpenAI gym," 2016, *arXiv:1606.01540*.

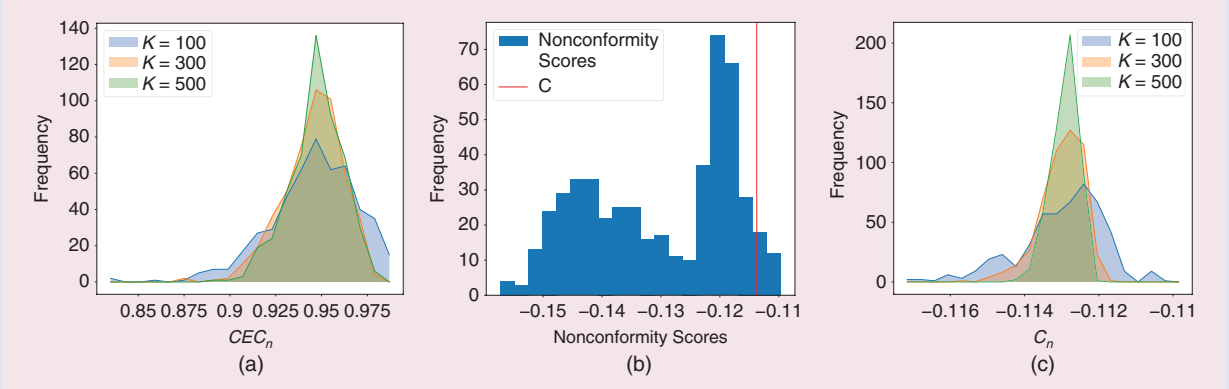


FIGURE S4 Experimental results for "Verifying a Learning-Enabled Cart–Pole System." (a) Histogram of the conditional empirical coverage CEC_n over all N experiments. (b) Histogram of the nonconformity score $R^{(i)}$ over $K := 500$ calibration datapoints (for one of the N experiments). (c) Histogram of the bound C_n over all N experiments.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

We note that techniques for computing probabilistic reachable sets share similarities with techniques for computing statistical abstractions.

techniques provide probabilistic safety guarantees, they do not provide reachable sets that can be used for downstream tasks, such as control synthesis, as described in the section "Control Synthesis for LEASs With Conformal Prediction." The authors of [236], [237], and [238] obtain probabilistic reachable sets by using conformal prediction and reachability analysis techniques for neural network surrogate models of the underlying unknown system. In [239], reachable sets are obtained via the use of Christoffel functions [240] that are subsequently conformalized. Computing probabilistic reachable sets can also be cast as a CCOP that can be solved via scenario optimization [241], [242]. Here, reachable sets are parameterized, and the optimal parameters are found by solving a CCOP. Conformal prediction and scenario optimization were used in [59] to quantify the correctness of learned solutions to Hamilton–Jacobi reachability problems. Finally, we note that techniques for computing probabilistic reachable sets share similarities with techniques for computing statistical abstractions, as we defined in the section "Statistical Abstractions of Dynamic Environments." The difference mainly lies in computational complexity as statistical abstractions for control should be computed efficiently.

General Verification

Conformal prediction was used to verify more general system properties, such as temporal logic specification. The authors of [243] and [244] learn a surrogate model for the performance function $\rho^{\phi_{\text{out}}}$ of a temporal logic specification ϕ_{out}, which they subsequently conformalize to obtain probabilistic satisfaction guarantees. The works in [235], [245], and [246] propose online verification algorithms, which we will discuss in more detail in the section "Online Verification of LEASs with Conformal Prediction."

Statistical model checking (SMC) has been used as an umbrella term for a set of statistical verification techniques; see, for example, [251], [252], [253], and [254] for an overview. However, SMC techniques in the literature are different from the techniques that we presented here (and also different from the techniques in [235], [243], [244], [245], and [246]) due to 1) the goal being to verify task satisfaction verifying that $z \vDash \phi_{\text{out}}$, as opposed to checking quantitative performance properties, for example, finding bounds $\rho*$ such that $\rho^{\phi_{\text{out}}}(z) \geq \rho*$, and 2) the use of statistical techniques that are different from conformal prediction (details are provided next).

SMC techniques can broadly be divided into two categories: qualitative and quantitative. For a given failure probability $\delta \in (0, 1)$, qualitative techniques aim to check whether or not $\text{Prob}(z \vDash \phi_{\text{out}}) \geq 1 - \delta$ holds. Quantitative techniques, on the other hand, aim to estimate $\text{Prob}(z \vDash \phi_{\text{out}})$ directly. Existing techniques for qualitative SMC use hypothesis testing [248], [254], [255], for example, via single sampling plans [247], [248] or sequential probability ratio tests [249], and more recently Clopper–Pearson bounds [250], [251]. In "Conformal Prediction and Qualitative SMC Techniques," we provide a short summary of existing qualitative SMC techniques and show, in comparison, how a qualitative (and even quantitative) technique based on conformal prediction can be designed. Quantitative SMC techniques, on the other hand, rely on random sampling of trajectories to directly estimate $\text{Prob}(z \vDash \phi_{\text{out}})$. Alongside, quantitative SMC techniques use concentration inequalities, such as the Chernoff–Hoeffding inequality, to obtain calibration conditional coverage guarantees for $Prob(z \vDash \phi_{\text{out}})$; see, for example, [254]. There are various extensions of these techniques that we want to briefly mention for the interested reader, such as the use of importance sampling [256], [257], [258], [259], [260] and importance splitting [261], [262], [263], [264], [265] in the case of rare events when $\text{Prob}(z \vDash \phi_{\text{out}})$ is small.

Finally, we mention statistical techniques that solve other verification problems. We briefly touched upon risk metrics before as an alternative to probabilistic coverage guarantees provided by conformal prediction. The statistical verification of LEASs in terms of their risk, as quantified using risk metrics, was presented in [266], [267], [268], [269], and [270]. Statistical verification techniques for system conformance that check the similarity of two stochastic systems were proposed using the Kolmogorov–Smirnov test and conformal prediction; see [271] and [90], respectively.

ONLINE VERIFICATION OF LEASs WITH CONFORMAL PREDICTION

Model-based verification techniques, such as model checking, exhaustively check all possible system behaviors for correctness. This exhaustive search is a computational bottleneck in practice, usually limiting the applicability of these techniques to low-fidelity models that may not capture the system dynamics sufficiently well. On the other hand, the search space in online verification is much smaller, allowing the use of high-fidelity models. In the section Offline Verification of LEASs with Conformal Prediction," we saw that statistical verification techniques, specifically those using conformal prediction, can alleviate

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

some of these computational challenges present in model-based verification. However, these techniques (similar to the control techniques presented in the section "Control Synthesis for LEASs With Conformal Prediction") provide probabilistic guarantees, meaning that there is still a failure probability (of at most δ). More specifically, a statistical verification technique may certify a system to be safe a priori, for example, with a probability of $1-\delta := 0.999$, but we may encounter one of the $\delta = 0.001$ fraction of realizations that can result in unsafe system behavior when executing the system. This again motivates online verification as a complementary safety mechanism.

Predictive Online Verification

Online verification techniques check if all extensions of a partial trajectory, for example, a trajectory observed online at time t, satisfy a specification. As discussed in "Spectrum of Formal Verification and Control Techniques," online verification can be conservative or inconclusive since no predictions about the system behavior are used. Having efficient and possibly learning-enabled prediction algorithms in hand, and knowing how to efficiently quantify uncertainty, here we are instead interested in predictive online verification of LEASs. Consider again a LEAS as in (37), which we recall for convenience as

$$z_{t+1} = f(z_t, u_t, v_t) \tag{40a}$$

$$y_t = p(z_t, w_t) \tag{40b}$$

$$u_t = \pi(\{y_s\}_{s=0}^{t}) \tag{40c}$$

where we adapt the same notation as before. Furthermore, we assume throughout this section that we have a calibration set of trajectories from (40) available, similarly to Assumption 3 in the section "Offline Verification of LEASs with Conformal Prediction."

Assumption 4

We have independently sampled a calibration dataset of K inputs from the distribution $\mathcal{D}$; that is, we have access to samples $(z_0^{(i)}, \{v_t^{(i)}\}_{t=0}^{T-1}, \{w_t^{(i)}\}_{t=0}^{T-1}) \sim \mathcal{D}$ for $i \in \{1, \ldots, K\}$. We use these to obtain a calibration dataset of K trajectories $z^{(i)} := (z_0^{(i)}, z_1^{(i)}, \ldots, z_T^{(i)})$ via the dynamics in (40).

Additionally, we are given a sound performance function $\rho^{\phi}: \mathbb{R}^{Tn_z} \to \mathbb{R}$ that describes the performance of the LEAS with respect to a specification ϕ that we evaluate for the trajectory z of the LEAS. Such specifications

Conformal Prediction and Qualitative SMC Techniques

Conformal prediction can provide a qualitative SMC technique with the nonconformity score $R^{(i)} := -\mathbb{1}(z^{(i)} \vDash \phi_{\text{out}})$, where $z^{(i)} \sim \mathcal{D}$ are again K independent and identically sampled calibration trajectories. By solving the expression $\text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, \infty) = -1$ for some $\delta \in (0,1)$, we know that $\text{Prob}(\mathbb{1}(z^{(0)} \vDash \phi_{\text{out}}) \geq 1) \geq 1-\delta$ holds, which is equivalent to $\text{Prob}(z^{(0)} \vDash \phi_{\text{out}}) \geq 1-\delta$. If we additionally find the smallest $\delta \in (0,1)$ that satisfies $\text{Quantile}_{1-\delta}(R^{(1)}, \ldots, R^{(K)}, \infty) = -1$, for example, via line search, we obtain a quantitative SMC technique that provides an estimate of $\text{Prob}(z^{(0)} \vDash \phi_{\text{out}})$. In comparison to existing SMC techniques in the literature, which we describe next, it is important to keep in mind that conformal prediction can provide both marginal as well as calibration conditional guarantees.

Let now $p := \text{Prob}(z^{(0)} \vDash \phi_{\text{out}})$ denote the unknown task satisfaction probability. In *hypothesis testing*, we construct two hypotheses $H_0: p \geq p_0$ and $H_1: p \leq p_1$, where $(p_1, p_0) \subseteq (0,1)$ is a user-defined "indifference" region, for example, a region that is small enough for us to ignore. We consider the experiment where we accept H_1 if $\Sigma_{i=1}^{K} \mathbb{1}(z^{(i)} \vDash \phi_{\text{out}}) \leq \xi$, while we accept H_0 if $\Sigma_{i=1}^{K} \mathbb{1}(z^{(i)} \vDash \phi_{\text{out}}) > \xi$, where ξ is a decision threshold. Our task is to find a sampling plan consisting of (K, ξ). An optimal sampling plan (K, ξ) minimizes the number of calibration trajectories K while maintaining a low probability of type-I error (accepting H_1 when H_0 holds), denoted as α, and a low probability of type-II error (accepting H_0 when H_1 holds), denoted as β. For fixed values of p_1, p_0, α, and β, this problem can be formulated as an optimization problem and solved via binary search. We refer the interested reader to [247] and [248] for details and discussions on the subtleties of the methods. It is intuitively clear that K has to become greater as α and β become smaller and as the indifference region (p_1, p_0) shrinks. An optimal plan obtained this way (referred to as the single sampling plan) uses a fixed K, while one can obtain verification results faster with early stopping via sequential probability ratio testing [249].

A qualitative SMC technique can also be obtained via the Clopper–Pearson bound [250], [251]. Consider a probability threshold p_0 and the estimate $\hat{p} := \Sigma_{i=1}^{K} \mathbb{1}(z^{(i)} \vDash \phi_{\text{out}})/K$ of $p := \text{Prob}(z^{(0)} \vDash \phi_{\text{out}})$. If $\hat{p} \geq p_0$, it holds that the probability that $p \geq p_0$ is lower bounded by $1 - \alpha_{\text{clopper}}$,[a] where α_{clopper} is a function of K and p_0, as discussed in detail in [251]. Concretely, α_{clopper} can be found via computing the cumulative distribution function of a beta distribution defined over the parameters K, p_0, and $\Sigma_{i=1}^{K} \mathbb{1}(z^{(i)} \vDash \phi_{\text{out}})$. Clopper–Pearson bounds are tighter than bounds obtained via Hoeffding's inequality. We note that guarantees for the opposite direction can be derived, that is, if $\hat{p} < p_0$, the probability that $p < p_0$ is lower bounded by $1 - \alpha_{\text{clopper}}$.

[a]Note that these are again calibration conditional guarantees where the outer probability is over all calibration trajectories.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

can again conveniently be expressed in a temporal logic formalism, such as STL; recall "LEAS Specifications in STL."

Informally, given a LEAS and a specification ϕ, we want to calculate the probability that the system trajectory $z := (z_0, \ldots, z_T)$ satisfies ϕ based on the partial trajectory $z_{\text{obs}} := (z_0, \ldots, z_t)$ observed at time t. Note that the trajectory z can be written as $z = (z_{\text{obs}}, z_{\text{un}})$, where knowledge of $z_{\text{un}} := (z_{t+1}, \ldots, z_T)$ is not available at time t. Formally, we aim to solve the following problem.

Problem 3

Given the system in (40), a specification ϕ over the trajectory z with sound performance function $\rho^\phi : \mathbb{R}^{Tn_z} \to \mathbb{R}$, and the trajectory $z_{\text{obs}} := (z_0, \ldots, z_t)$ observed at time t, compute a lower performance bound $\rho*$ such that

$$\text{Prob}\,(\rho^\phi(z) \geq \rho*) \geq 1 - \delta.$$

The intuition of the performance bound $\rho*$ is similar to the bounds that we derived for verification algorithms in the sections "LEC Verification with Conformal Prediction" and "Offline Verification of LEASs with Conformal Prediction"; that is, $\rho* > 0$ is a sufficient condition to guarantee that $\text{Prob}\,(z \vDash \phi) \geq 1 - \delta$ holds.

Accurate and Interpretable Predictive Online Verification

We present two predictive online verification algorithms, closely following [245]. The idea, similar to that in the section "Statistical Abstractions of Dynamic Environments," is to use trajectory predictors $\mu : \mathbb{R}^{(t+1)n_z} \to \mathbb{R}^{(T-t)n_z}$ to predict future states $(z_{t+1}, \ldots, z_T)$ from past and present states $(z_0, \ldots, z_t)$. However, unlike in the section "Statistical Abstractions of Dynamic Environments," we now predict the state of the LEAS and not only the state of the environment. We denote predictions of z_τ for future times $\tau > t$ by $\hat{z}_{\tau|t}$; that is, we have

$$(\hat{z}_{t+1|t}, \ldots, \hat{z}_{T|t}) = \mu(z_0, \ldots, z_t).$$

We would now like to use these predictions to compute $\rho^\phi(z)$. However, as future states z_{un} are uncertain and as predictions may not be accurate, we need to construct prediction regions for $\rho^\phi(z)$ by quantifying uncertainty using conformal prediction. We first present an accurate predictive online verification algorithm that constructs such prediction regions directly by relating the nonconformity score to the performance function $\rho^\phi(z)$. As a consequence, this algorithm provides accurate prediction regions, for example, tight lower bounds ρ^*. However, this algorithm lacks interpretability, as it provides no information on why a specification is satisfied or violated. For the case where the specification ϕ expresses an STL specification, we therefore present an interpretable predictive online verification algorithm. This algorithm constructs statistical abstractions for future states z_{un} first, following the same idea as in Theorems 1 and 2 in the section "Statistical Abstractions of Dynamic Environments," and then uses these abstractions to obtain a prediction region for the performance function $\rho^\phi(z)$. We refer the reader to Figure 6 for an illustration that is taken from [245].

Accurate Predictive Online Verification

As in the section "Statistical Abstractions of Dynamic Environments," we compute the predictions

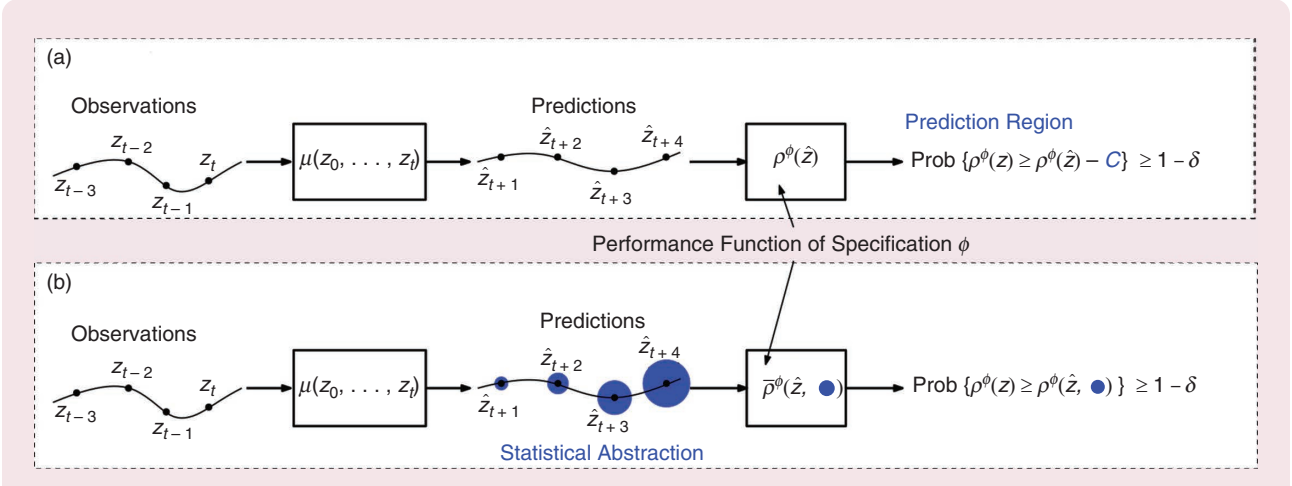


FIGURE 6 (a) Accurate predictive runtime online verification algorithm. (b) Interpretable predictive runtime online verification algorithm. Both algorithms use past observations $(z_0, \ldots, z_t)$ to predict future states $(\hat{z}_{t+1}, \hat{z}_{t+2}, \ldots)$. The accurate algorithm applies conformal prediction to the nonconformity score $\rho^\phi(\hat{z}) - \rho^\phi(z)$ to directly obtain a probabilistic lower bound for $\rho^\phi(z)$. The interpretable algorithm constructs a statistical abstraction, as introduced in the section "Statistical Abstractions of Dynamic Environments," and uses it to compute the worst-case performance function $\bar{\rho}^\phi(\hat{z})$, which is a probabilistic lower bound for $\rho^\phi(z)$. (Figure taken from [245].)

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

$$\left(\hat{z}^{(i)}_{t+1|t},\ldots,\hat{z}^{(i)}_{T|t}\right) := \mu(z^{(i)}_0,\ldots,z^{(i)}_t)$$

for all available calibration data $i \in \{1,\ldots,K\}$. Using these predictions, we define the predicted trajectory

$$\hat{z}^{(i)} := \left(z^{(i)}_0,\ldots,z^{(i)}_t,\hat{z}^{(i)}_{t+1|t},\ldots,\hat{z}^{(i)}_{T|t}\right)$$

which is the concatenation of the partial trajectory $z^{(i)}_0,\ldots,z^{(i)}_t$ and the predictions of future states $\hat{z}^{(i)}_{t+1|t},\ldots,\hat{z}^{(i)}_{T|t}$. We can now use the predicted trajectory to compute the predicted performance $\rho^{\phi}(\hat{z}^{(i)})$. Let us then compute the nonconformity score

$$R^{(i)} := \rho^{\phi}(\hat{z}^{(i)}) - \rho^{\phi}(z^{(i)}) \tag{41}$$

as the difference between the predicted and the true performance. We then immediately get the following result.

Theorem 8

Given an output property ϕ with sound performance function $\rho^{\phi}: \mathbb{R}^{Tn_z} \to \mathbb{R}$ and a test trajectory $z^{(0)}$ obtained via the dynamics in (40) from samples $(z^{(0)}_0,\{v^{(0)}_t\}^{T-1}_{t=0},\{w^{(0)}_t\}^{T-1}_{t=0}) \sim \mathcal{D}$, we then have that

$$\text{Prob}\left(\rho^{\phi}(z^{(0)}) \geq \rho^{\phi}(\hat{z}^{(0)}) - C\right) \geq 1-\delta$$

where $C := \text{Quantile}_{1-\delta}(R^{(1)},\ldots,R^{(K)},\infty)$ with $R^{(i)}$ being defined in (41).

The proof follows immediately by the definition of the nonconformity score in (41). The interpretation of this result is that we obtain a probabilistic lower bound for the performance $\rho^{\phi}(z^{(0)})$ of the test trajectory in terms of the predicted performance $\rho^{\phi}(\hat{z}^{(0)})$, which is calibrated by C. In this way, the accurate predictive online verification algorithm solves Problem 3 with $\rho^{*} := \rho^{\phi}(\hat{z}^{(0)}) - C$. We provide an illustrative simulation example and a comparison with the interpretable predictive online verification algorithm (presented next) at the end of the next section.

Interpretable Predictive Online Verification

As we created statistical abstractions in (14) and (15) that we used for constructing open-loop and closed-loop controllers, we construct statistical abstractions here. However, we are interested in abstractions of the form

$$\text{Prob}\left(\| z^{(0)}_{\tau} - \hat{z}^{(0)}_{\tau|t} \| \leq C_{\tau|t},\ \forall \tau \in \{t+1,\ldots,T\}\right) \geq 1-\delta \tag{42}$$

for example, in prediction regions for τ-step-ahead predictions made at time t for the test trajectory $z^{(0)}$. The challenge again lies in computing nonconservative bounds $C_{\tau|t}$ from the calibration data $z^{(1)},\ldots,z^{(K)}$. We discussed several techniques for this purpose in the section "Statistical Abstractions of Dynamic Environments," specifically in Theorems 1 and 2, that can be easily applied here.

Worst-Case Performance Function

Having obtained statistical abstractions of the form (42) via a valid choice of $C_{\tau|t}$, we now aim to calculate the worst-case performance of the function ρ^{ϕ} over the uncertainty sets

$$\mathcal{B}_{\tau|t} := \{\tilde{z}_{\tau} \in \mathbb{R}^{n_z} \mid \| \tilde{z}_{\tau} - \hat{z}^{(0)}_{\tau|t} \| \leq C_{\tau|t}\}.$$

In an ideal world where we can solve nonconvex optimization problems, this worst-case performance is computed as

$$\min_{\tilde{z}_{t+1}\in\mathcal{B}_{\tau|t},\ldots,\tilde{z}_T\in\mathcal{B}_{T|t}} \rho^{\phi}(\tilde{z})$$

where $\tilde{z} := (z_0,\ldots,z_t,\tilde{z}_{t+1},\ldots,\tilde{z}_T)$ is the concatenation of the partial trajectory $z_0,\ldots,z_t$ and the worst-case values $\tilde{z}_{t+1},,\tilde{z}_T$ from the uncertainty sets $\mathcal{B}_{\tau|t},\ldots,\mathcal{B}_{T|t}$. Unfortunately, this worst-case performance is hard to compute, motivating us to construct a lower bound, $\bar{\rho}^{\phi}$ (referred to as the worst-case performance function), that can be computed recursively over the structure of the STL specification ϕ, for example, for predicates, Boolean operators, and temporal operators.

Predicates

We compute $\bar{\rho}^{\mu}(\hat{z}^{(0)},\tau)$ for the predicates μ as

$$\bar{\rho}^{\mu}(\hat{z}^{(0)},\tau) := \begin{cases} h(z^{(0)}_{\tau}) & \text{if } \tau \leq t \\ \min_{\tilde{z}_{\tau}\in\mathcal{B}_{\tau|t}} h(\tilde{z}_{\tau}) & \text{otherwise.} \end{cases}$$

The intuition here is that we know the value of $z^{(0)}_{\tau}$ and hence also the value of the performance function $\rho^{\mu}(z^{(0)},\tau) = h(z^{(0)}_{\tau})$, at time $\tau \leq t$. For times $\tau > t$, we know that $\text{Prob}(z^{(0)}_{\tau} \in \mathcal{B}_{\tau|t}) \geq 1-\delta$ since the statistical abstraction in (42) is valid, which motivates us to compute $\bar{\rho}^{\mu}(\hat{z}^{(0)},\tau) := \min_{\tilde{z}_{\tau}\in\mathcal{B}_{\tau|t}} h(\tilde{z}_{\tau})$. As a consequence, it is easy to see that $\bar{\rho}^{\mu}(\hat{z}^{(0)},\tau)$ is a lower bound of $\rho^{\mu}(z^{(0)},\tau)$ with a probability no less than $1-\delta$, for example, that

$$\text{Prob}\left(\rho^{\mu}(z^{(0)},\tau) \geq \bar{\rho}^{\mu}(\hat{z}^{(0)},\tau)\right) \geq 1-\delta.$$

Solving $\min_{\tilde{z}_{\tau}\in\mathcal{B}_{\tau|t}} h(\tilde{z}_{\tau})$ can generally be a nonconvex optimization problem. However, for convex predicate functions h, it is easy to see that $\min_{\tilde{z}_{\tau}\in\mathcal{B}_{\tau|t}} h(\tilde{z}_{\tau})$ is a convex optimization problem that can be solved efficiently. For Lipschitz continuous functions h with Lipschitz constant L_h, we obtain an efficient lower bound as

$$\min_{\tilde{z}_{\tau}\in\mathcal{B}_{\tau|t}} h(\tilde{z}_{\tau}) \geq h(\hat{z}_{\tau|t}) - L_h C_{\tau|t}.$$

Boolean and Temporal Operators

We use the probabilistic lower bound $\bar{\rho}^{\mu}(\hat{z}^{(0)},\tau)$ to recursively derive probabilistic lower bounds for Boolean and temporal operators. This requires that the specification ϕ contains no negations as deriving a lower bound for $\rho^{\neg\mu}(z^{(0)},\tau)$ would require knowledge of an upper bound for $\rho^{\mu}(z^{(0)},\tau)$ (see "LEAS Specifications in STL"). This is without loss of generality as every STL specification ϕ can be

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

rewritten into an equivalent STL specification without negations by instead using disjunction ($\vee$), eventually (F_I), and always operators (G_I); see [272] for details.

The worst-case performance function for temporal and Boolean operators is defined in the same way as the performance function is defined. Specifically, following "LEAS Specifications in STL," we have

$$\begin{aligned}
\bar{\rho}^{\top}(\hat{z},t) &:= \infty,\\
\bar{\rho}^{\phi'\wedge\phi''}(\hat{z},t) &:= \min(\bar{\rho}^{\phi'}(\hat{z},t),\bar{\rho}^{\phi''}(\hat{z},t))\\
\bar{\rho}^{\phi'\vee\phi''}(\hat{z},t) &:= \max(\bar{\rho}^{\phi'}(\hat{z},t),\bar{\rho}^{\phi''}(\hat{z},t))\\
\bar{\rho}^{F_I\phi}(\hat{z},t) &:= \max_{t'\in(t\oplus I)\cap\mathbb{N}} \bar{\rho}^{\phi}(\hat{z},t')\\
\bar{\rho}^{G_I\phi}(\hat{z},t) &:= \min_{t'\in(t\oplus I)\cap\mathbb{N}} \bar{\rho}^{\phi}(\hat{z},t')\\
\bar{\rho}^{\phi' U_I\phi''}(\hat{z},t) &:= \max_{t''\in(t\oplus I)\cap\mathbb{N}} \min(\bar{\rho}^{\phi''}(\hat{z},t''), \min_{t'\in[t,t'']\cap\mathbb{N}} \bar{\rho}^{\phi'}(\hat{z},t')).
\end{aligned}$$

We remark that the computation of the worst-case performance function for Boolean and temporal operators is computationally extremely efficient; that is, it does not involve solving an optimization problem as in the case of predicates. By construction, the value of $\bar{\rho}^{\phi}(\hat{z}) := \bar{\rho}^{\phi}(\hat{z},0)$ gives us a probabilistic lower bound of $\rho^{\phi}(z)$.

Theorem 9

Given an STL specification ϕ with sound performance function $\rho^{\phi}:\mathbb{R}^{Tn_z}\to\mathbb{R}$, a valid statistical abstraction of the form (42), and a test trajectory $z^{(0)}$ obtained via the dynamics in (40) from samples $(z_0^{(0)},\{v_t^{(0)}\}_{t=0}^{T-1},\{w_t^{(0)}\}_{t=0}^{T-1})\sim\mathcal{D}$, we then have that

$$\text{Prob}\,(\rho^{\phi}(z^{(0)})\geq\bar{\rho}^{\phi}(\hat{z}^{(0)}))\geq 1-\delta.$$

The interpretable predictive online verification algorithm therefore solves Problem 3 with $\rho^{*}:=\bar{\rho}^{\phi}(\hat{z})$.

Comparing the Presented Predictive Online Verification Algorithms

In comparison with the accurate predictive online verification algorithm, the interpretable predictive online verification algorithm can be more conservative when the statistical abstraction in (42) is conservative. Another reason why the interpretable version may be more conservative is due to the computation of $\bar{\rho}^{\mu}(\hat{z}^{(0)},\tau)$ which involves a worst-case argument over the statistical abstraction for all predicates μ within ϕ and for all times $\tau>t$. Conservatism may also arise when a nonconvex optimization problem needs to be solved for computing $\bar{\rho}^{\mu}(\hat{z}^{(0)},\tau)$. On the other hand, the values of $\bar{\rho}^{\mu}(\hat{z}^{(0)},\tau)$ provide interpretability. We illustrate both predictive online verification algorithms in the following example.

Example 4

Consider the aircraft simulator from [273], in which a highly nonlinear F-16 aircraft with its controller is modeled by a 16D state z_t. For verification, we focus on the altitude z_t^h (in feet) and the speed z_t^s (in feet/second). The aircraft trajectories are generated from the simulator with initial conditions drawn from a normal distribution, for example, $(z_0^h,z_0^s)\sim\mathcal{N}(1{,}000,\,10^2)\times\mathcal{N}(650,5^2)=:\mathcal{D}$. We consider the specification

$$\phi := G_{[0,T]}(z^h\geq\zeta_1\wedge(z^h<\zeta_2\implies z^s\leq\zeta_3))$$

where $T:=150$, $\zeta_1:=100$, $\zeta_2:=300$, and $\zeta_3:=650$. In words, this specification requires that the aircraft 1) always has a safe altitude ($z^h\geq\zeta_1$) and 2) maintains a sufficiently low speed ($z^s\leq\zeta_3$) whenever it is close to the ground ($z^h<\zeta_2$). We next consider online verification at time $t:=80$.

We start by training a long short-term memory network. For illustration, we show three sampled trajectories (in solid lines) and their predictions (in dashed lines) in Figure 7(a). We then set the failure probability to $\delta:=0.05$. To verify statistical validity empirically, we conduct $N:=500$ experiments with $K:=700$ calibration trajectories and $J:=200$ test trajectories each.

Accurate Method. We use the nonconformity score $R^{(i)}$ as per (41). By this choice, we know that $\text{Prob}\,(\rho^{\phi}(z^{(0)})\geq\rho^{\phi}(\hat{z}^{(0)})-C)\geq 1-\delta$ from Theorem 8. We then compute the empirical coverage EC according to (3) as 0.934. We also plot the histogram of the conditional empirical coverage CEC_n according to (5) over all N experiments in Figure 7(b). Note that CEC_n is equivalent to the conditional empirical coverage of $\rho^{\phi}(z^{(0)})\geq\rho^{\phi}(\hat{z}^{(0)})-C$.[22] As expected, the histogram centers around $1-\delta$. For one experiment, we also show in Figure 7(c) the ground truth performance $\rho^{\phi}(z^{(0)})$ with the associated lower performance bound $\rho^{*}:=\rho^{\phi}(\hat{z}^{(0)})-C$ for all test trajectories $z^{(0)}$. It can be seen that the lower bound is tight, specifically in comparison with the interpretable method that we present next.

Interpretable Method. We use a statistical abstraction of the form (42) and obtain the bounds $C_{\tau|t}$ by following the "single nonconformity score approach" discussed in the section "Statistical Abstractions of Dynamic Environments." We also define the worst-case performance function $\bar{\rho}^{\phi}(\tilde{z}^{(0)})$ as instructed before Theorem 9. By this choice, we know that $\text{Prob}\,(\rho^{\phi}(z^{(0)})\geq\bar{\rho}^{\phi}(\hat{z}^{(0)}))\geq 1-\delta$ from Theorem 9. To empirically validate this statement, we compute the conditional empirical coverage of $\rho^{\phi}(z^{(0)})\geq\bar{\rho}^{\phi}(\hat{z}^{(0)})$, which is now not equivalent to CEC_n in (5) (unlike in the case for the accurate method).[23] We plot the corresponding histogram over all N experiments in Figure 7(d). For one experiment, we also show in Figure 7(e) the ground truth performance $\rho^{\phi}(z^{(0)})$ with the associated lower performance bound $\rho^{*}:=\bar{\rho}^{\phi}(\hat{z}^{(0)})$ for all test trajectories $z^{(0)}$. As mentioned

[22] Specifically, note that CEC_n in (5) is equivalent to $\Sigma_{j=1}^{J}\mathbb{1}(\rho^{\phi}(z_{nj}^{(0)})\geq\rho^{\phi}(\hat{z}_{nj}^{(0)})-C_n)/J$, where $z_{nj}^{(0)}$ is the jth test trajectory of the nth experiment, while C_n is as in (5).

[23] Formally, we compute $\Sigma_{j=1}^{J}\mathbb{1}(\rho^{\phi}(z_{nj}^{(0)})\geq\bar{\rho}^{\phi}(\hat{z}_{nj}^{(0)}))/J$.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

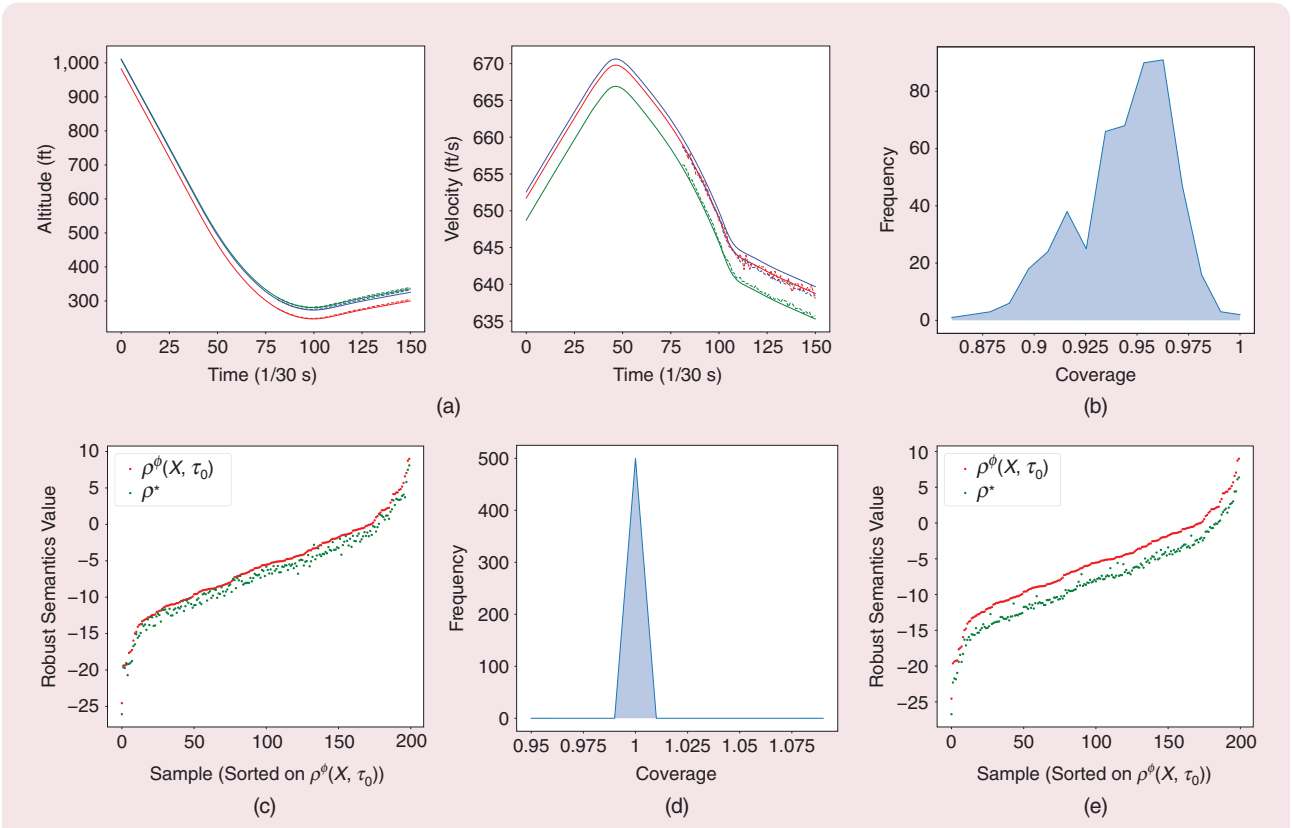


FIGURE 7 Experimental results for Example 4. (a) Example trajectories from $\mathcal{D}$. (b) Histogram of $\rho^{\phi}(z^{(0)}) \geq \rho^*$ over all N experiments for the accurate method. (c) Ground truth performance $\rho^{\phi}(z)$ and lower performance bound ρ^* from the accurate method. (d) Histogram of $\rho^{\phi}(z^{(0)}) \geq \rho^*$ over all N experiments for the interpretable method. (e) Ground truth performance $\rho^{\phi}(z)$ and lower performance bound ρ^* from the interpretable method.

before, the interpretability of the method comes at the cost of conservatism, visible here by coverage much larger than $1-\delta$ and a lower bound that is not as tight as for the accurate method.

Predictive Online Verification Under Distribution Shift

Both predictive online verification algorithms rely on a calibration dataset of trajectories $z^{(1)},\ldots,z^{(K)}$, as per Assumption 4. In practice, these trajectories are typically generated by a simulator or a digital twin model. A simulator implicitly models a distribution over trajectories. However, the system deployed in the real world may encounter a different distribution over trajectories because of the sim2real gap; that is, the actual distribution of trajectories in a deployment setting may be shifted from the distribution assumed by the simulator. Sometimes, one may be able to collect trajectories from the real system. However, the distribution of these trajectories may still be different from the distribution encountered during deployment, for example, when the system encounters new scenarios. To account for such distribution shifts, we follow [246] and construct predictive online verification algorithms by using robust conformal prediction (recall "Conformal Prediction Under Distribution Shift."

Quantifying Distribution Shifts

Let $z^{(1)},\ldots,z^{(K)} \sim \mathcal{Z}$ be trajectories sampled from a calibration distribution $\mathcal{Z}$, that is, as in Assumption 4 where the calibration trajectory $z^{(i)}$ is generated by the dynamics in (40) from an input sample $(z_0^{(i)}, \{v_t^{(i)}\}_{t=0}^{T-1}, \{w_t^{(i)}\}_{t=0}^{T-1}) \sim \mathcal{D}$. By contrast, let $z^{(0)} \sim \mathcal{Z}_0$ now be a test trajectory sampled from a deployment distribution $\mathcal{Z}_0$; that is, the trajectory $z^{(0)}$ is generated by dynamics different from those of (40) and an input sample $(z_0^{(0)}, \{v_t^{(0)}\}_{t=0}^{T-1}, \{w_t^{(0)}\}_{t=0}^{T-1}) \sim \mathcal{D}_0$, where $\mathcal{D}_0$ may also not be the same as $\mathcal{D}$. The starting point in [246] is an assumption on the difference between the trajectories of the calibration and the deployment distribution.

Assumption 5

The deployment and calibration distributions $\mathcal{Z}_0$ and $\mathcal{Z}$ are such that $D_f(\mathcal{Z}_0, \mathcal{Z}) \leq \epsilon$, where $D_f(\cdot)$ is an f-divergence and $\epsilon > 0$ measures the distribution shift.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

Assumption 5 means that the distribution of the test trajectory $z^{(0)} \sim \mathcal{Z}_0$ is unknown but ι-close to the distribution of the calibration trajectories $z^{(1)},\ldots,z^{(K)} \sim \mathcal{Z}$. Closeness here is defined in terms of an f-divergence, for example, the total variation distance or the KL divergence. We remark that similar assumptions on the knowledge of an upper bound of the closeness of systems are common in robust control (see, for example, [274] and [275]). In practice, the distribution shift ι can either be estimated empirically (see, for example, [52]), or it can be a tuning parameter that induces robustness margins.

Robust Predictive Online Verification

We are now interested in solving Problem 3 under Assumption 5. Let $R^{(i)}$ therefore, be a nonconformity score, for example, a general function of $z^{(i)}$. As a consequence, there exist calibration and deployment distributions $\mathcal{R}$ and $\mathcal{R}_0$ such that $R^{(i)} \sim \mathcal{R}$ for $i \in \{1,\ldots,K\}$ and $R^{(0)} \sim \mathcal{R}_0$. Importantly, $D_f(\mathcal{Z}_0,\mathcal{Z}) \leq \epsilon$ implies $D_f(\mathcal{R}_0,\mathcal{R}) \leq \epsilon$ by the data processing inequality. For this reason, we can obtain an accurate and an interpretable predictive online verification algorithm in the same way as before, but now using robust conformal prediction. We next summarize these results, starting with the accurate verification algorithm.

Corollary 5

Given an output property ϕ with sound performance function $\rho^\phi : \mathbb{R}^{Tn_z} \to \mathbb{R}$ and a test trajectory $z^{(0)} \sim \mathcal{Z}_0$ that satisfies Assumption 5, we then have that

$$\text{Prob}\,(\rho^\phi(z^{(0)}) \geq \rho^\phi(\hat{z}^{(0)}) - \tilde{C}) \geq 1-\delta$$

where $\tilde{C} := \text{Quantile}_{1-\tilde{\delta}}(R^{(1)},\ldots,R^{(K)})$ as in (S6) with $R^{(i)}$ being defined in (41) and $\tilde{\delta}$ being defined in (S7).

For the construction of an interpretable predictive online verification algorithm, we first have to construct a valid robust statistical abstraction of the form

$$\text{Prob}\,(\|\, z_\tau^{(0)} - \hat{z}_{\tau|t}^{(0)} \,\| \leq \tilde{C}_{\tau|t}, \forall \tau \in \{t+1,\ldots,T\}) \geq 1-\delta. \tag{43}$$

For example, (43) has to hold for any $z^{(0)} \sim \mathcal{Z}_0$ that satisfies Assumption 5. We can compute $\tilde{C}_{\tau|t}$ in the same way as $C_{\tau|t}$ in (42), but by replacing δ with $\tilde{\delta}$ from (S7).

Corollary 6

Given an STL specification ϕ with a sound performance function $\rho^\phi : \mathbb{R}^{Tn_z} \to \mathbb{R}$, a valid robust statistical abstraction of the form (43), and a test trajectory $z^{(0)} \sim \mathcal{Z}_0$ that satisfies Assumption 5, we then have that

$$\text{Prob}\,(\rho^\phi(z) \geq \bar{\rho}^\phi(\hat{z})) \geq 1-\delta.$$

Example 5

Recall Example 4, in which we obtained calibration trajectories generated from the F-16 aircraft simulator with initial conditions drawn from $\mathcal{N}(1{,}000,\, 10^2) \times \mathcal{N}(650, 5^2) =: \mathcal{D}$. By contrast, here we consider that test trajectories are generated with initial conditions drawn from $\mathcal{N}(998, 10^2) \times \mathcal{N}(651, 5^2) =: \mathcal{D}_0$. The remaining settings (specification ϕ, time t, and predictor μ) are the same as in Example 4. We use the total variation distance to measure the distribution shift, that is, to measure $D_f(\mathcal{D}_0,\mathcal{D})$. We estimate $D_f(\mathcal{D}_0,\mathcal{D})$ from data to be ≈ 0.129 and correspondingly set $\epsilon := 0.129$.[24]

In this example, we set the failure probability to $\delta := 0.2$. To verify the statistical validity empirically, we conduct $N := 500$ experiments with $K := 700$ calibration trajectories (sampled from $\mathcal{D}$) and $J := 200$ test trajectories (sampled from $\mathcal{D}_0$). Our goals here are to 1) illustrate the correctness of the robust predictive online algorithms from Corollaries 5 and 6 and 2) compare to the vanilla predictive online algorithms from Theorems 8 and 9, which will not achieve valid coverage.

Accurate Method. We use the nonconformity score $R^{(i)}$ as per (41). By this choice, we know that $\text{Prob}\,(\rho^\phi(z^{(0)}) \geq \rho^\phi(\hat{z}^{(0)}) - \tilde{C}) \geq 1-\delta$ from Corollary 5, where $\tilde{C}$ follows from (S6) [as opposed to C following from (2) in the vanilla case]. For one experiment, we plot the histogram of the nonconformity scores and the values of $\tilde{C}$ and C in Figure 8(a) for comparison. As expected, we notice that $\tilde{C} > C$. We next plot the histograms of the conditional empirical coverages of $\rho^\phi(z^{(0)}) \geq \rho^\phi(\hat{z}^{(0)}) - \tilde{C}$ and $\rho^\phi(z^{(0)}) \geq \rho^\phi(\hat{z}^{(0)}) - C$ over all N experiments in Figure 8(b). It can be seen that the vanilla method cannot achieve $1-\delta$ coverage because of the distribution shift, while the robust method achieves a coverage greater than $1-\delta$, accounting for the distribution shift. For one experiment, we also show in Figure 8(c) the ground truth performance $\rho^\phi(z^{(0)})$ with the associated lower performance bound $\rho^* := \rho^\phi(\hat{z}^{(0)}) - \tilde{C}$ for all test trajectories $z^{(0)}$. The lower bound appears tight despite the conservatism inherent to robust methods.

Interpretable Method. We use a robust statistical abstraction of the form (43) and obtain the bounds $\tilde{C}_{\tau|t}$ by following the "single nonconformity score approach" discussed in the section Statistical Abstractions of Dynamic Environments" but now using robust conformal prediction. We again define the worst-case performance function $\bar{\rho}^\phi(\tilde{z}^{(0)})$ as instructed before Theorem 9. By this choice, we know that $\text{Prob}\,(\rho^\phi(z^{(0)}) \geq \bar{\rho}^\phi(\hat{z}^{(0)})) \geq 1-\delta$ from Corollary 6. To empirically validate this statement, we compute the conditional empirical coverage of $\rho^\phi(z^{(0)}) \geq \bar{\rho}^\phi(\hat{z}^{(0)})$ for the robust and the vanilla methods. We plot the corresponding histograms over all N experiments in Figure 8(d). From conservatism of the interpretable method, valid $1-\delta$ coverage is achieved for both methods. For one experiment, we also show in Figure 8(e) the ground truth performance $\rho^\phi(z^{(0)})$ with the associated lower performance bound $\rho^* := \bar{\rho}^\phi(\hat{z}^{(0)})$ for all test trajectories $z^{(0)}$. The

[24] We refer the reader to [246] for details on how the total variation $D_f(\mathcal{D}_0,\mathcal{D})$ can be estimated from data samples.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

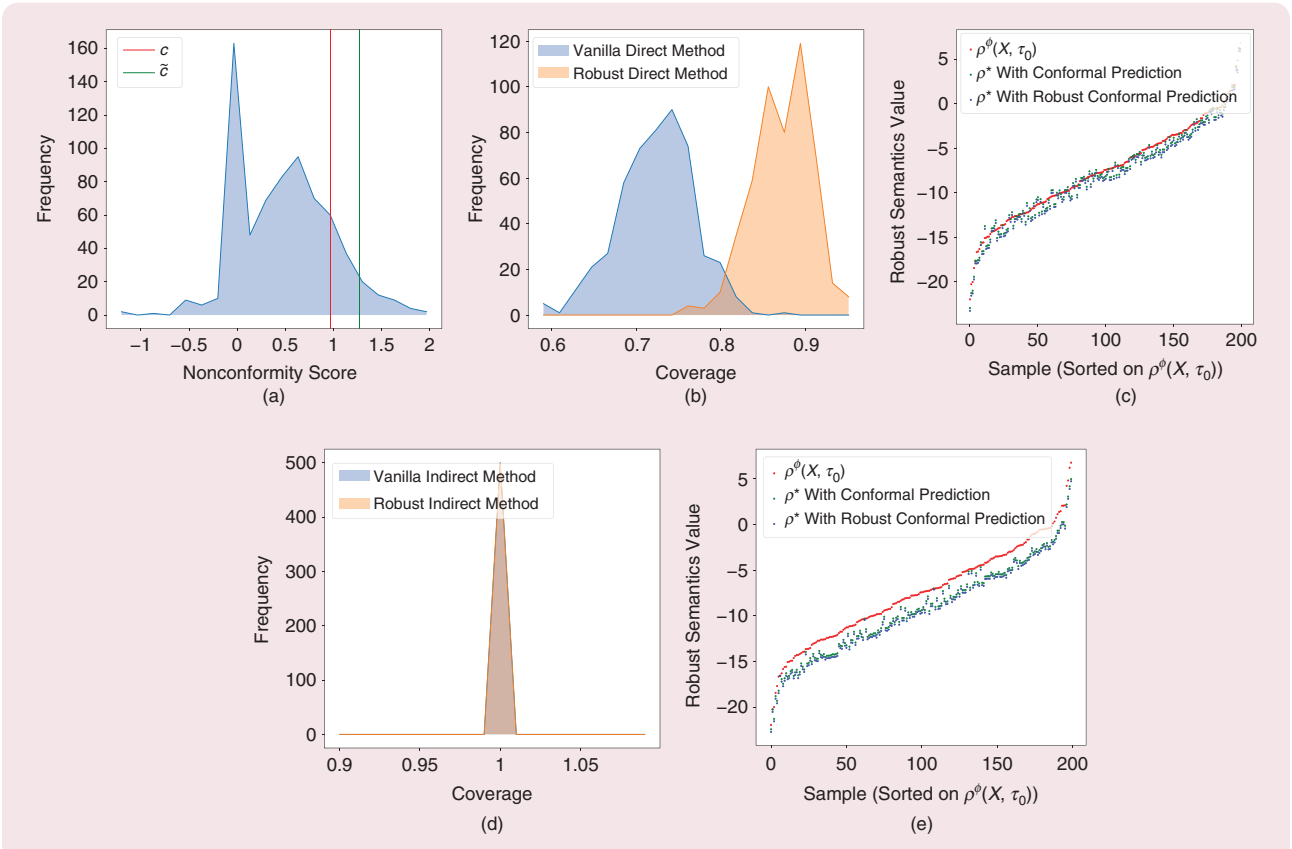


FIGURE 8 Experimental results for Example 5. (a) Histogram of the nonconformity scores $R^{(i)}$ in (41). (b) Histogram of $\rho^{\phi}(z^{(0)}) \geq \rho^{*}$ over all N experiments for the nonrobust and robust accurate methods. (c) Ground truth performance $\rho^{\phi}(z)$ and lower performance bound ρ^{*} from the accurate methods. (d) Histogram of $\rho^{\phi}(z^{(0)}) \geq \rho^{*}$ over all N experiments for the nonrobust and robust interpretable methods. (e) Ground truth performance $\rho^{\phi}(z)$ and lower performance bound ρ^{*} from the interpretable methods.

vanilla method provides tighter bounds than the robust method, as it does not account for all possible distribution shifts of size ϵ.

Related Work: Online Verification Techniques

Online verification algorithms check if a trajectory satisfies a temporal logic specification by using nothing more than the part of the trajectory that has already been observed. Standard algorithms (see, for example, [28], [29], [276], [277], and [278]) provide one out of three verification answers: 1) "satisfied" if all extensions of the partial trajectory satisfy the specification, 2) "dissatisfied" if no extension of the partial trajectory satisfies the specification, and 3) "inconclusive" otherwise. Multivalued temporal logics, such as those in [279] and [280], extend beyond this setting. For temporal logic specifications that admit a sound performance function ρ^{ϕ}, such as STL, there even exist online verification algorithms that provide a real-valued verification answer (see, for example, [30], [280], [281], [282], [283], and [284]). A real-valued verification answer can be interpreted as the robustness of the partial trajectory.

Predictive online verification algorithms use a model of the system (which is usually nondeterministic) to obtain less conservative verification answers. Existing algorithms either provide robust satisfaction guarantees for all possible behaviors of the nondeterministic models (see, for example, [285], [286], and [287]), or probabilistic satisfaction guarantees for stochastic models (see, for example, [31], [32], [288], [289], and [290]). The authors of [291], [292], and [293] follow a conceptually similar idea but integrate model predictions and prediction uncertainties into the semantics of a temporal logic formalism. Predictive online verification algorithms based on Bayesian inference were presented in [294], [295], and [296]. The aforementioned techniques, however, often only apply to specific system models or do not provide strong probabilistic coverage guarantees as can be obtained with conformal prediction.

As demonstrated before, conformal prediction can be used to design predictive online verification algorithms that provide probabilistic verification guarantees for LEASs. Conformal prediction was also used in [235] to design predictive online verification algorithms. The idea

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

We want to emphasize our conviction that statistical techniques can be beneficial in integrating foundational and generative models into system design.

in [235] is to train a quantile regressor that directly predicts the quantile of the distribution of the performance function ρ^{ϕ}. Subsequently, CQR (see "Heteroskedasticity and Conformal Prediction") is used to capture uncertainty and variability in these predictions. By contrast, in this section we trained trajectory predictors that may generally be harder to obtain and require more engineering effort during training. However, the technique from [235] requires one to train a new quantile regressor each time the specification ϕ changes or a new specification is added. Similar to the accurate predictive online verification algorithm from the section "Accurate Predictive Online Verification," the technique from [235] does not provide interpretability in case of failure as opposed to the interpretable predictive online verification algorithms from the section "Interpretable Predictive Online Verification." Along the same lines, we briefly point the reader to the work in [297], which presents online causation monitoring. Finally, we mention the use of conformal prediction for out-of-distribution detection [298], [299], [300], [301], anomaly detection [302], [303], [304], and failure detection [305], [306].

LIMITATIONS AND OPEN PROBLEMS

To conclude this article, we discuss limitations and list open problems that we believe are important to address in our pursuit of designing safe autonomous systems. This list is by no means exhaustive and naturally is biased by our own views and ideas.

Limitations

Scalability

Conformal prediction involves computing the empirical quantile $\text{Quantile}_{1-\delta}(R^{(1)},\dots,R^{(K)},\infty)$, which, as mentioned previously, can be done using standard sorting algorithms with $O(K\log(K))$ time complexity. While this is generally a scalable approach, larger datasets (which are needed to achieve a smaller failure probability δ) lead to larger time complexity, which may be limiting in some situations, for example, when computation is done on embedded hardware. Another bottleneck in practice may be memory and the ability to store large datasets. This can become particularly challenging when images/videos are used.

Conservatism

We recall from the section "Conformal Prediction in a Nutshell" that conformal prediction provides tight upper and lower bounds for $\text{Prob}(R^{(0)} \le C(R^{(1)},\dots,R^{(K)}))$ in the sense that

$$1-\delta \le \text{Prob}(R^{(0)} \le C(R^{(1)},\dots,R^{(K)})) \le 1-\delta+\frac{1}{K+1}$$

when a prespecified nonconformity score $R^{(i)}$ is used. In this survey, we then spend a lot of time engineering the "right" nonconformity score in the hope of obtaining tight prediction regions for the problem at hand. While the flexibility in designing $R^{(i)}$ is generally an advantage, it is worth recognizing that we are not guaranteed to obtain the tightest possible prediction region, which motivates further research, as discussed in the next section.

Practical Limitations

There are many real-world challenges that statistical tools, in general and, by extension, conformal prediction, face. One challenge that we mentioned throughout this survey is related to various forms of distribution shift. Although there exist some robust extensions of conformal prediction (see "Conformal Prediction Under Distribution Shift"), it is worth noting that all these extensions have their own limitations. Another important challenge in practice is selecting proper calibration datasets. While companies may have a lot of data, these datasets may not be well curated. Therefore, an important task is the construction of good datasets.

Open Problems

One major research direction will be the *tight integration of statistical techniques in the system design* of an autonomous system. Some of the open problems here, as already discussed in this article, are to deal with distribution shifts caused by 1) sim2real gaps that arise in practice and 2) couplings between the controllers and statistical abstractions, and hence in dealing with distribution shifts. Another problem of a more academic nature is to investigate the similarities and differences of the various statistical techniques mentioned throughout this survey and to gain a better understanding of when to use which technique. Finally, we want to emphasize our conviction that statistical techniques can be beneficial in integrating foundational and generative models into system design.

Another impactful research direction will be to continue *the design of efficient and accurate statistical and perceptual abstractions*. Although we have already presented techniques to construct the abstractions in great detail in this article, we believe that there are important

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

improvements that need to be made. Some of these improvements are the design of statistical abstractions that 1) integrate contemporary probabilistic predictors (such as diffusion models and decision transformers), 2) accurately capture geometric and temporal patterns of the underlying data-generating distribution, and 3) can account for multimodality in the data. We also believe that techniques are needed for constructing perceptual abstractions that 1) are data efficient and 2) can provide end-to-end control guarantees. Another important milestone will be the joint design of statistical and perceptual abstractions.

Another broad research direction is *the actual design of formal verification and control techniques*. In practice, autonomous systems are usually organized and designed within layers, for example, with layers that separately perform behavior planning, path planning, motion planning, and feedback control. It is unclear in which layers we should be using statistical techniques and if we should use the same statistical and perceptual abstractions across layers or if individual abstractions should be tailored to specific layers. There are many other open technical questions that are important in practice but were not addressed in this survey article. How can we guarantee the optimality and recursive feasibility of control algorithms? How can we design formal verification and control techniques for long-term autonomy, for example, to accommodate asymptotic properties, such as stability and forward invariance?

The next important research direction that we would like to mention is the *design of learning-enabled multiagent systems*. While learning-enabled single-agent systems are fairly well understood, it is unclear how to design learning-enabled multiagent systems due to their size, complex network structure, and the data dependencies between agents. New specification languages may be needed to formulate safety requirements, while at the same time, the need for scalable and distributed verification and control techniques is apparent. While statistical techniques appear promising toward obtaining scalability, it is unclear how statistical techniques can be distributed.

Finally, we believe that it will be critical for us as a community to *showcase practical impact across different application domains and within other communities*. To achieve this goal, it will be important to provide well-documented and reusable toolboxes. We also think that the design of photorealistic and high-fidelity simulators plays a critical role toward accomplishing impressive real-world demonstrations and experiments.

ACKNOWLEDGMENT

Lars Lindemann, Yiqi Zhao, Xinyi Yu, and Jyotirmoy V. Deshmukh were supported in part by the National Science Foundation through Grants SLES-2417075, SHF-2048094, CNS-1932620, CNS-2039087, FMitF-1837131, and CCF-SHF-1932620, Toyota R&D and Siemens Corporate Research through the University of Southern California Center for Autonomy and AI, an Amazon Faculty Research Award, and the Airbus Institute for Engineering Research. George J. Pappas was supported in part by the National Science Foundation through Grant SLES-2331880 and the Air Force Office of Scientific Research through the HYDRA program.

AUTHOR INFORMATION

Lars Lindemann (llindemann@ethz.ch) is an assistant professor of algorithmic systems theory in the Automatic Control Laboratory at ETH Zürich, 8092 Zürich, Switzerland. From 2023 to 2025 he was an assistant professor in the Thomas Lord Department of Computer Science at the University of Southern California. From 2020 to 2022 he was a postdoctoral fellow in the Department of Electrical and Systems Engineering at the University of Pennsylvania. He received the Ph.D. degree in electrical engineering from KTH Royal Institute of Technology in 2020. His research interests include systems and control theory, formal methods, machine learning, and autonomous systems. He received the Outstanding Student Paper Award at the 58th IEEE Conference on Decision and Control (CDC) and the Student Best Paper Award (as an advisor) at the 60th CDC. He was a finalist for the Best Paper Award (as an advisor) at the 2024 International Conference on Cyber-Physical Systems and the Best Paper Award at the 2022 Conference on Hybrid Systems: Computation and Control and the Best Student Paper Award at the 2018 American Control Conference.

Yiqi Zhao received the B.S. degree with honors in computer science from Vanderbilt University, Nashville, TN, USA, in 2023. He also completed a secondary major in mathematics and minors in electrical engineering and data science at Vanderbilt University. He is working toward a Ph.D. degree in computer science in the Thomas Lord Department of Computer Science, at the University of Southern California, Los Angeles, CA 90089 USA. His research interests include formal methods, cyberphysical systems, systems and control theory, and mathematical optimization. He was a finalist for the Best Paper Award at the 2024 International Conference on Cyber-Physical Systems.

Xinyi Yu received the M.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2023, and the B.Eng degree from China University of Petroleum (East China), Qingdao, Shandong, China, in 2020, both in automation. She is pursuing a Ph.D. degree in computer science in the Thomas Lord Department of Computer Science at the University of Southern California, Los Angeles, CA 90089 USA. Her research interests include formal methods and safety control.

George J. Pappas received the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, Berkeley, CA, USA, in 1998. He is the Joseph Moore Professor and chair of the Department of

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA. He also holds a secondary appointment with the Department of Computer and Information Sciences and the Department of Mechanical Engineering and Applied Mechanics. He is a member of the General Robotics, Automation, Sensing, and Perception Lab and the Penn Research Center for Embedded Computing and Integrated Systems Engineering Center. He was previously the deputy dean for research with the School of Engineering and Applied Science. His research interests include control theory with a particular focus on hybrid systems, embedded systems, cyber-physical systems, and hierarchical and distributed control systems, with applications to unmanned aerial vehicles, distributed robotics, green buildings, and biomolecular networks. He was a recipient of various awards, such as the Antonio Ruberti Young Researcher Prize, the IEEE Control Systems Society George S. Axelby Outstanding Paper Award, the O. Hugo Schuck Best Paper Award, the George H. Heilmeier Award, the National Science Foundation Presidential Early Career Award for Scientists and Engineers, and numerous best student papers awards. He is a Fellow of IEEE.

Jyotirmoy V. Deshmukh is an associate professor of computer science in the Thomas Lord Department of Computer Science, University of Southern California (USC), Los Angeles, CA 90089 USA. and the codirector of the USC Center for Autonomy and AI. Previously, he was a principal research engineer at Toyota Motors R&D. He was a postdoctoral research scholar at the University of Pennsylvania and received the Ph.D. degree from the University of Texas at Austin.

REFERENCES

[1] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.

[2] Z. Liu, Z. Wu, and R. Tóth, "SMOKE: Single-stage monocular 3D object detection via keypoint estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 4289–4298, doi: 10.1109/CVPRW50498.2020.00506.

[3] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, "Learning to estimate 3D human pose and shape from a single color image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 459–468, doi: 10.1109/CVPR.2018.00055.

[4] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Piscataway, NJ, USA: IEEE Press, 2017, pp. 1722–1729, doi: 10.1109/ICRA.2017.7989203.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[6] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 6000–6010.

[7] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 961–971, doi: 10.1109/CVPR.2016.110.

[8] L. Liu, D. Dugas, G. Cesari, R. Siegwart, and R. Dubé, "Robot navigation in crowded environments using deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Piscataway, NJ, USA: IEEE Press, 2020, pp. 5671–5677, doi: 10.1109/IROS45743.2020.9341540.

[9] V. Mnih et al., "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.

[10] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017, doi: 10.1109/MSP.2017.2743240.

[11] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *Proc. 18th Eur. Control Conf. (ECC)*, Piscataway, NJ, USA: IEEE Press, 2019, pp. 307–312, doi: 10.23919/ECC.2019.8795639.

[12] A. Robey et al., "Learning control barrier functions from expert demonstrations," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, Piscataway, NJ, USA: IEEE Press, 2020, pp. 3717–3724, doi: 10.1109/CDC42340.2020.9303785.

[13] C. Szegedy et al., "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.

[14] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[15] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part I," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 99–110, Jun. 2006, doi: 10.1109/MRA.2006.1638022.

[16] E. A. Wan and R. Van Der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. IEEE Adaptive Syst. Signal Process., Commun., Control Symp.*, Piscataway, NJ, USA: IEEE Press, 2000, pp. 153–158.

[17] P. M. Djuric et al., "Particle filtering," *IEEE Signal Process. Mag.*, vol. 20, no. 5, pp. 19–38, Sep. 2003, doi: 10.1109/MSP.2003.1236770.

[18] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Hoboken, NJ, USA: Wiley, 2006.

[19] G. Shafer and V. Vovk, "A tutorial on conformal prediction," *J. Mach. Learn. Res.*, vol. 9, no. 3, pp. 371–421, 2008.

[20] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*, vol. 29. New York, NY, USA: Springer-Verlag.

[21] A. N. Angelopoulos and S. Bates, "Conformal prediction: A gentle introduction," *Found. Trends® Mach. Learn.*, vol. 16, no. 4, pp. 494–591, 2023, doi: 10.1561/2200000101.

[22] M. Fontana, G. Zeni, and S. Vantini, "Conformal prediction: A unified review of theory and new challenges," *Bernoulli*, vol. 29, no. 1, pp. 1–23, 2023, doi: 10.3150/21-BEJ1447.

[23] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—A survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989, doi: 10.1016/0005-1098(89)90002-2.

[24] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica*, vol. 34, no. 10, pp. 1205–1217, 1998.

[25] M. Morari and J. H. Lee, "Model predictive control: Past, present and future," *Comput. Chem. Eng.*, vol. 23, no. 4–5, pp. 667–682, 1999, doi: 10.1016/S0098-1354(98)00301-9.

[26] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Trans. Autom. Control*, vol. 51, no. 5, pp. 742–753, May 2006, doi: 10.1109/TAC.2006.875041.

[27] M. C. Campi, S. Garatti, and M. Prandini, "The scenario approach for systems and control design," *Annu. Rev. Control*, vol. 33, no. 2, pp. 149–157, 2009, doi: 10.1016/j.arcontrol.2009.07.001.

[28] A. Bauer, M. Leucker, and C. Schallhart, "Runtime verification for LTL and TLTL," *ACM Trans. Softw. Eng. Method.*, vol. 20, no. 4, pp. 1–64, 2011, doi: 10.1145/2000799.2000800.

[29] M. Leucker and C. Schallhart, "A brief account of runtime verification," *J. Log. Algebr. Program.*, vol. 78, no. 5, pp. 293–303, 2009, doi: 10.1016/j.jlap.2008.08.004.

[30] J. V. Deshmukh, A. Donzé, S. Ghosh, X. Jin, G. Juniwal, and S. A. Seshia, "Robust online monitoring of signal temporal logic," *Formal Methods Syst. Des.*, vol. 51, no. 1, pp. 5–30, 2017, doi: 10.1007/s10703-017-0286-7.

[31] R. Babaee, A. Gurfinkel, and S. Fischmeister, "Prevent: A predictive run-time verification framework using statistical learning," in *Proc. Int. Conf. Softw. Eng. Formal Methods*, Cham, Switzerland: Springer-Verlag, 2018, pp. 205–220.

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

[33] I. Ziemann, A. Tsiamis, B. Lee, Y. Jedra, N. Matni, and G. J. Pappas, "A tutorial on the non-asymptotic theory of system identification," in *Proc. 62nd IEEE Conf. Decis. Control (CDC)*, Piscataway, NJ, USA: IEEE Press, 2023, pp. 8921–8939, doi: 10.1109/CDC49753.2023.10383489.

[34] A. Tsiamis, I. Ziemann, N. Matni, and G. J. Pappas, "Statistical learning theory for control: A finite-sample perspective," *IEEE Control Syst. Mag.*, vol. 43, no. 6, pp. 67–97, Dec. 2023, doi: 10.1109/MCS.2023.3310345.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

[35] S. A. Seshia, D. Sadigh, and S. S. Sastry, "Toward verified artificial intelligence," *Commun. ACM*, vol. 65, no. 7, pp. 46–55, 2022, doi: 10.1145/3503914.
[36] M. Ganai, S. Gao, and S. Herbert, "Hamilton-Jacobi reachability in reinforcement learning: A survey," *IEEE Open J. Control Syst.*, vol. 3, pp. 310–324, 2024, doi: 10.1109/OJCSYS.2024.3449138.
[37] X. Yin, B. Gao, and X. Yu, "Formal synthesis of controllers for safety-critical autonomous systems: Developments and challenges," *Annu. Rev. Control*, vol. 57, 2024, Art. no. 100940, doi: 10.1016/j.arcontrol.2024.100940.
[38] F. Cairoli, L. Bortolussi, and N. Paoletti, "Learning-based approaches to predictive monitoring with conformal statistical guarantees," in *Proc. Int. Conf. Runtime Verification*, Cham, Switzerland: Springer-Verlag, 2023, pp. 461–487.
[39] J. Kapinski, J. V. Deshmukh, X. Jin, H. Ito, and K. Butts, "Simulation-based approaches for verification of embedded control systems: An overview of traditional and advanced modeling, testing, and verification techniques," *IEEE Control Syst. Mag.*, vol. 36, no. 6, pp. 45–64, Dec. 2016, doi: 10.1109/MCS.2016.2602089.
[40] A. Corso, R. Moss, M. Koren, R. Lee, and M. Kochenderfer, "A survey of algorithms for black-box safety validation of cyber-physical systems," *J. Artif. Intell. Res.*, vol. 72, pp. 377–428, 2021, doi: 10.1613/jair.1.12716.
[41] A. Majumdar and M. Pavone, "How should a robot assess risk? Towards an axiomatic theory of risk in robotics," in *Robotics Research*, N. M. Amato, G. Hager, S. Thomas, M. Torres-Torriti, Eds., Cham, Switzerland: Springer-Verlag, 2020, pp. 75–84.
[42] Y. Wang and M. P. Chapman, "Risk-averse autonomous systems: A brief history and recent developments from the perspective of optimal control," *Artif. Intell.*, vol. 311, Oct. 2022, Art. no. 103743, doi: 10.1016/j.artint.2022.103743.
[43] P. Akella et al., "Risk-aware robotics: Tail risk measures in planning, control, and verification," 2024, *arXiv:2403.18972*.
[44] H. A. David and H. N. Nagaraja, *Order Statistics*. Hoboken, NJ, USA: Wiley, 2004.
[45] Y. Zhao, X. Yu, M. Sesia, J. V. Deshmukh, and L. Lindemann, "Conformal predictive programming for chance constrained optimization," 2024, *arXiv:2402.07407*.
[46] R. J. Tibshirani, R. F. Barber, E. Candes, and A. Ramdas, "Conformal prediction under covariate shift," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 2530–2540.
[47] J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman, "Distribution-free predictive inference for regression," *J. Am. Stat. Assoc.*, vol. 113, no. 523, pp. 1094–1111, 2018, doi: 10.1080/01621459.2017.1307116.
[48] R. Durrett, *Probability: Theory and Examples*, vol. 49. Cambridge, U.K.: Cambridge Univ. Press, 2019.
[49] V. Vovk, "Conditional validity of inductive conformal predictors," in *Proc. Asian Conf. Mach. Learn.*, PMLR, 2012, pp. 475–490.
[50] J. C. Duchi, "A few observations on sample-conditional coverage in conformal prediction," 2025, *arXiv:2503.00220*.
[51] A. N. Angelopoulos, R. F. Barber, and S. Bates, "Theoretical foundations of conformal prediction," 2024, *arXiv:2411.11824*.
[52] M. Cauchois, S. Gupta, A. Ali, and J. C. Duchi, "Robust validation: Confident predictions even when distributions shift," *J. Am. Stat. Assoc.*, vol. 119, no. 548, pp. 3033–3044, 2024, doi: 10.1080/01621459.2023.2298037.
[53] Y. Zhou, L. Lindemann, and M. Sesia, "Conformalized adaptive forecasting of heterogeneous trajectories," in *Proc. 41st Int. Conf. Mach. Learn. (ICML)*, 2024, pp. 62,002–62,056.
[54] N. O'Sullivan, L. Romao, and K. Margellos, "Bridging conformal prediction and scenario optimization," 2025, *arXiv:2503.23561*.
[55] R. Vershynin, *High-Dimensional Probability: An Introduction with Applications in Data Science*, vol. 47. Cambridge, U.K.: Cambridge Univ. Press, 2018.
[56] S. Boucheron, G. Lugosi, and O. Bousquet, "Concentration inequalities," in *Summer School on Machine Learning*, O. Bousquet, U. von Luxburg, and G. Rätsch, Eds., Berlin, Heidelberg, Germany: Springer-Verlag, 2003, pp. 208–240.
[57] G. Calafiore and M. C. Campi, "Uncertain convex programs: Randomized solutions and confidence levels," *Math. Program.*, vol. 102, no. 1, pp. 25–46, 2005, doi: 10.1007/s10107-003-0499-y.
[58] S. Garatti and M. C. Campi, "Non-convex scenario optimization," *Math. Program.*, vol. 209, no. 1–2, pp. 557–608, 2025, doi: 10.1007/s10107-024-02074-3.
[59] A. Lin and S. Bansal, "Verification of neural reachable tubes via scenario optimization and conformal prediction," in *Proc. 6th Annu. Learn. Dyn. Control Conf.*, PMLR, 2024, pp. 719–731.
[60] H. Rahimian and S. Mehrotra, "Frameworks and results in distributionally robust optimization," *Open J. Math. Optim.*, vol. 3, pp. 1–85, 2022, doi: 10.5802/ojmo.15.
[61] G. A. Hanasusanto, V. Roitch, D. Kuhn, and W. Wiesemann, "A distributionally robust perspective on uncertainty quantification and chance constrained programming," *Math. Program.*, vol. 151, no. 1, pp. 35–62, 2015, doi: 10.1007/s10107-015-0896-z.
[62] P. Mohajerin Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations," *Math. Program.*, vol. 171, no. 1–2, pp. 115–166, 2018, doi: 10.1007/s10107-017-1172-1.
[63] P. Yash, C. Guyang, and A. Tewari, "Non-parameteric conformal distributionally robust optimization," in *Proc. ICML Workshop Struct. Probabilistic Inference Generative Model.*, 2024.
[64] E. E. Vlahakis, L. Lindemann, P. Sopasakis, and D. V. Dimarogonas, "Conformal prediction for distribution-free optimal control of linear stochastic systems," *IEEE Contr. Syst. Lett.*, vol. 8, pp. 2835–2840, 2024, doi: 10.1109/LCSYS.2024.3514472.
[65] R. Tumu, M. Cleaveland, R. Mangharam, G. Pappas, and L. Lindemann, "Multi-modal conformal prediction regions by optimizing convex shape templates," in *Proc. 6th Annu. Learn. Dyn. Control Conf.*, PMLR, 2024, pp. 1343–1356.
[66] A. Maurer, "A note on the PAC Bayesian theorem," 2004, *arXiv:cs/0411099*.
[67] D. A. McAllester, "Some PAC-Bayesian theorems," in *Proc. 11th Annu. Conf. Comput. Learn. Theory*, 1998, pp. 230–234.
[68] J. Shawe-Taylor and R. C. Williamson, "A PAC analysis of a Bayesian estimator," in *Proc. 10th Annu. Conf. Comput. Learn. Theory*, 1997, pp. 2–9.
[69] P. Alquier, "User-friendly introduction to PAC-Bayes bounds," *Found. Trends® Mach. Learn.*, vol. 17, no. 2, pp. 174–303, 2024, doi: 10.1561/2200000100.
[70] A. Sharma, S. Veer, A. Hancock, H. Yang, M. Pavone, and A. Majumdar, "PAC-Bayes generalization certificates for learned inductive conformal prediction," in *Proc. 37th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2024, pp. 46,807–46,829.
[71] C. Liu, T. Arnon, C. Lazarus, C. Strong, C. Barrett, and M. J. Kochenderfer, "Algorithms for verifying deep neural networks," *Found. Trends® Optim.*, vol. 4, no. 3–4, pp. 244–404, 2021, doi: 10.1561/2400000035.
[72] M. Kwiatkowska and X. Zhang, "When to trust AI: Advances and challenges for certification of neural networks," in *Proc. 18th Conf. Comput. Sci. Intell. Syst. (FedCSIS)*, Piscataway, NJ, USA: IEEE Press, 2023, pp. 25–37, doi: 10.15439/2023F2324.
[73] S. Dutta, S. Jha, S. Sankaranarayanan, and A. Tiwari, "Output range analysis for deep feedforward neural networks," in *Proc. NASA Formal Methods Symp.*, Cham, Switzerland: Springer-Verlag, 2018, pp. 121–138.
[74] M. Fischetti and J. Jo, "Deep neural networks and mixed integer linear optimization," *Constraints*, vol. 23, no. 3, pp. 296–309, 2018, doi: 10.1007/s10601-018-9285-6.
[75] V. Tjeng, K. Y. Xiao, and R. Tedrake, "Evaluating robustness of neural networks with mixed integer programming," in *Proc. Int. Conf. on Learn. Representations*, 2019.
[76] A. Lomuscio and L. Maganti, "An approach to reachability analysis for feed-forward ReLU neural networks," 2017, *arXiv:1706.07351*.
[77] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient SMT solver for verifying deep neural networks," in *Proc. Int. Conf. Comput. Aided Verification*, Cham, Switzerland: Springer-Verlag, 2017, pp. 97–117.
[78] R. Ehlers, "Formal verification of piece-wise linear feed-forward neural networks," in *Proc. 15th Int. Symp. Automated Technol. Verification Anal.*, Cham, Switzerland: Springer-Verlag, 2017, pp. 269–286.
[79] G. Katz et al., "The Marabou framework for verification and analysis of deep neural networks," in *Proc. 31st Int. Conf. Comput. Aided Verification*, Cham, Switzerland: Springer-Verlag, 2019, pp. 443–452.
[80] M. Fazlyab, M. Morari, and G. J. Pappas, "Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming," *IEEE Trans. Autom. Control*, vol. 67, no. 1, pp. 1–15, Jan. 2022, doi: 10.1109/TAC.2020.3046193.
[81] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. Pappas, "Efficient and accurate estimation of lipschitz constants for deep neural networks," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 11,427–11,438.
[82] S. Wang et al., "Beta-CROWN: Efficient bound propagation with per-neuron split constraints for neural network robustness verification," in *Proc. 35th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2021, pp. 29,909–29,921.
[83] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

functions," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 4944–4953.
[84] G. Singh, T. Gehr, M. Püschel, and M. Vechev, "An abstract domain for certifying neural networks," *Proc. ACM Program. Lang.*, vol. 3, no. POPL, pp. 1–30, 2019, doi: 10.1145/3290354.
[85] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. Vechev, "AI2: Safety and robustness certification of neural networks with abstract interpretation," in *Proc. IEEE Symp. Secur. Privacy (SP)*, Piscataway, NJ, USA: IEEE Press, 2018, pp. 3–18, doi: 10.1109/SP.2018.00058.
[86] G. Singh, T. Gehr, M. Mirman, M. Püschel, and M. Vechev, "Fast and effective robustness certification," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 10,825–10,836.
[87] A. Donzé and O. Maler, "Robust satisfaction of temporal logic over real-valued signals," in *Proc. Int. Conf. Formal Model. Anal. Timed Syst.*, Berlin, Heidelberg, Germany: Springer-Verlag, 2010, pp. 92–106.
[88] G. E. Fainekos and G. J. Pappas, "Robustness of temporal logic specifications for continuous-time signals," *Theor. Comput. Sci.*, vol. 410, no. 42, pp. 4262–4291, 2009, doi: 10.1016/j.tcs.2009.06.021.
[89] S. A. Seshia et al., "Formal specification for deep neural networks," in *Proc. 16th Int. Symp. Automated Technol. Verification Anal.*, Cham, Switzerland: Springer-Verlag, 2018, pp. 20–34.
[90] X. Qin, N. Hashemi, L. Lindemann, and J. V. Deshmukh, "Conformance testing for stochastic cyber-physical systems," in *Proc. 23rd Conf. Formal Methods Comput.-Aided Des. (FMCAD)*, 2023, p. 294.
[91] L. Haan and A. Ferreira, *Extreme Value Theory: An Introduction*, vol. 3. Cham, Switzerland: Springer-Verlag, 2006.
[92] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: A survey," *Int. J. Rob. Res.*, vol. 39, no. 8, pp. 895–935, 2020, doi: 10.1177/0278364920917446.
[93] A. Graves, "Generating sequences with recurrent neural networks," 2013, *arXiv:1308.0850*.
[94] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in *Proc. Europ. Conf. Comp. Vis. (ECCV)*, 2020, pp. 683–700.
[95] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, "Wayformer: Motion forecasting via simple & efficient attention networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Piscataway, NJ, USA: IEEE Press, 2023, pp. 2980–2987, doi: 10.1109/ICRA48891.2023.10160609.
[96] M. A. Benjamin, R. A. Rigby, and D. M. Stasinopoulos, "Generalized autoregressive moving average models," *J. Am. Stat. Assoc.*, vol. 98, no. 461, pp. 214–223, 2003, doi: 10.1198/016214503388619238.
[97] K. Stankeviciute, A. M. Alaa, and M. van der Schaar, "Conformal time-series forecasting," in *Proc. 35th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2021, pp. 6216–6228.
[98] L. Lindemann, M. Cleaveland, G. Shim, and G. J. Pappas, "Safe planning in dynamic environments using conformal prediction," *IEEE Robot. Autom. Lett.*, vol. 8, no. 8, pp. 5116–5123, Aug. 2023, doi: 10.1109/LRA.2023.3292071.
[99] M. Cleaveland, I. Lee, G. J. Pappas, and L. Lindemann, "Conformal prediction regions for time series using linear complementarity programming," *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 19, pp. 20,984–20,992, 2024, doi: 10.1609/aaai.v38i19.30089.
[100] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999, doi: 10.1016/S0005-1098(98)00178-2.
[101] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," *in Proc. IEEE Int. Conf. Robot. Automat*, Piscataway, NJ, USA: IEEE Press, 2008, pp. 1928–1935, doi: 10.1109/ROBOT.2008.4543489.
[102] P. Kothari, S. Kreiss, and A. Alahi, "Human trajectory forecasting in crowds: A deep learning perspective," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7386–7400, Jul. 2022, doi: 10.1109/TITS.2021.3069362.
[103] X. Yu, Y. Zhao, X. Yin, and L. Lindemann, "Signal temporal logic control synthesis among uncontrollable dynamic agents with conformal prediction," 2023, *arXiv:2312.04242*.
[104] A. Muthali et al., "Multi-agent reachability calibration with conformal prediction," in *Proc. 62nd IEEE Conf. Decis. Control (CDC)*, Piscataway, NJ, USA: IEEE Press, 2023, pp. 6596–6603, doi: 10.1109/CDC49753.2023.10383723.
[105] J. Lei, J. Robins, and L. Wasserman, "Efficient nonparametric conformal prediction regions," 2011, *arXiv:1111.1418*.
[106] J. Lei, J. Robins, and L. Wasserman, "Distribution-free prediction sets," *J. Am. Stat. Assoc.*, vol. 108, no. 501, pp. 278–287, 2013, doi: 10.1080/01621459.2012.751873.
[107] J. Lei and L. Wasserman, "Distribution-free prediction bands for non-parametric regression," *J. Roy. Statist. Soc. Ser. B: Statist. Methodol.*, vol. 76, no. 1, pp. 71–96, 2014, doi: 10.1111/rssb.12021.
[108] J. Smith, I. Nouretdinov, R. Craddock, C. Offer, and A. Gammerman, "Anomaly detection of trajectories with kernel density estimation by conformal prediction," in *Proc. IFIP Int. Conf. Artif. Intell. Appl. Innov.*, Berlin, Heidelberg, Germany: Springer-Verlag, 2014, pp. 271–280.
[109] D. Stutz, K. D. Dvijotham, A. T. Cemgil, and A. Doucet, "Learning optimal conformal classifiers," in *Proc. Int. Conf. Learn. Representations*, 2022.
[110] G. Thurin, K. Nadjahi, and C. Boyer, "Optimal transport-based conformal prediction," in *Proc. 42nd Int. Conf. Mach. Learn.*, 2025.
[111] S. Braun, L. Aolaritei, M. I. Jordan, and F. Bach, "Minimum volume conformal sets for multivariate regression," 2025, *arXiv:2503.19068*.
[112] C. Gao, L. Shan, V. Srinivas, and A. Vijayaraghavan, "Volume optimality in conformal prediction with structured prediction sets," in *Proc. 42nd Int. Conf. Mach. Learn.*, 2025.
[113] S. Sun and R. Yu, "Copula conformal prediction for multi-step time series forecasting," 2022, *arXiv:2212.03281*.
[114] L. Marques and D. Berenson, "Quantifying aleatoric and epistemic dynamics uncertainty via local conformal calibration," 2024, *arXiv:2409.08249*.
[115] B. Lee and N. Matni, "Single trajectory conformal prediction," in *Proc. IEEE 63rd Conf. Decis. Control (CDC)*, Piscataway, NJ, USA: IEEE Press, 2024, pp. 3019–3024, doi: 10.1109/CDC56724.2024.10886644.
[116] M. Zecchin, S. Park, and O. Simeone, "Forking uncertainties: Reliable prediction and model predictive control with sequence models via conformal risk control," *IEEE J. Sel. Areas Inf. Theory*, vol. 5, pp. 44–61, 2024, doi: 10.1109/JSAIT.2024.3368229.
[117] Z. Wang, R. Gao, M. Yin, M. Zhou, and D. Blei, "Probabilistic conformal prediction using conditional random samples," in *Proc. Int. Conf. Artif. Intell. Statist.*, PMLR, 2023, pp. 8814–8836.
[118] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
[119] C. Stamouli, L. Lindemann, and G. Pappas, "Recursively feasible shrinking-horizon MPC in dynamic environments with conformal prediction guarantees," in *Proc. 6th Annu. Learn. Dyn. Control Conf.*, PMLR, 2024, pp. 1330–1342.
[120] S. Tonkens, S. Sun, R. Yu, and S. Herbert, "Scalable safe long-horizon planning in dynamic environments leveraging conformal prediction and temporal correlations," in *Proc. Long-Term Human Motion Prediction Workshop, Int. Conf. Robot. Autom.*, 2023.
[121] E. Rimon, *Exact Robot Navigation Using Artificial Potential Functions*. New Haven, CT, USA: Yale Univ. Press, 1990.
[122] D. V. Dimarogonas and K. J. Kyriakopoulos, "On the rendezvous problem for multiple nonholonomic agents," *IEEE Trans. Autom. Control*, vol. 52, no. 5, pp. 916–922, May 2007, doi: 10.1109/TAC.2007.895897.
[123] B. Şenbaşlar and G. S. Sukhatme, "DREAM: Decentralized real-time asynchronous probabilistic trajectory planning for collision-free multirobot navigation in cluttered environments," *IEEE Trans. Robot.*, vol. 41, pp. 573–592, 2025, doi: 10.1109/TRO.2024.3509015.
[124] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, no. 5, pp. 4282–4286, 1995, doi: 10.1103/PhysRevE.51.4282.
[125] R. F. Barber, E. J. Candes, A. Ramdas, and R. J. Tibshirani, "Conformal prediction beyond exchangeability," *Ann. Statist.*, vol. 51, no. 2, pp. 816–845, 2023, doi: 10.1214/23-AOS2276.
[126] A. Dixit, L. Lindemann, S. X. Wei, M. Cleaveland, G. J. Pappas, and J. W. Burdick, "Adaptive conformal prediction for motion planning among dynamic agents," in *Proc. 5th Annu. Learn. Dyn. Control Conf.*, PMLR, 2023, pp. 300–314.
[127] S. Sheng, P. Yu, D. Parker, M. Kwiatkowska, and L. Feng, "Safe POMDP online planning among dynamic agents via adaptive conformal prediction," *IEEE Robot. Autom. Lett.*, vol. 9, no. 11, pp. 9946–9953, Nov. 2024, doi: 10.1109/LRA.2024.3468092.
[128] J. Yao, X. Zhang, Y. Xia, Z. Wang, A. K. Roy-Chowdhury, and J. Li, "SoNIC: Safe social navigation with adaptive conformal inference and constrained reinforcement learning," 2024, *arXiv:2407.17460*.
[129] V. Raman, A. Donzé, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Model predictive control with signal temporal logic specifications," in *Proc. Conf. Decis. Control*, Los Angeles, CA, USA, 2014, pp. 81–87.
[130] V. Raman, A. Donzé, D. Sadigh, R. M. Murray, and S. A. Seshia, "Reactive synthesis from signal temporal logic specifications," in *Proc. 18th Int. Conf. Hybrid Syst.: Comput. Control (HSCC)*, 2015, pp. 239–248.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

[131] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proc. IEEE*, vol. 111, no. 3, pp. 257–276, Mar. 2023, doi: 10.1109/JPROC.2023.3238524.
[132] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.
[133] C. Hsieh, Y. Li, D. Sun, K. Joshi, S. Misailovic, and S. Mitra, "Verifying controllers with vision-based perception using safe approximate abstractions," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 41, no. 11, pp. 4205–4216, Nov. 2022, doi: 10.1109/TCAD.2022.3197508.
[134] A. Astorga, C. Hsieh, P. Madhusudan, and S. Mitra, "Perception contracts for safety of ML-enabled systems," *Proc. ACM Program. Lang.*, vol. 7, no. OOPSLA2, pp. 2196–2223, 2023, doi: 10.1145/3622875.
[135] D. Sun, B. C. Yang, and S. Mitra, "Learning-based inverse perception contracts and applications," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2024, pp. 11,612–11,618, doi: 10.1109/ICRA57147.2024.10610329.
[136] S. Dean, A. Taylor, R. Cosner, B. Recht, and A. Ames, "Guaranteeing safety of learned perception modules via measurement-robust control barrier functions," in *Proc. Conf. Robot Learn.*, PMLR, 2021, pp. 654–670.
[137] S. Dean, N. Matni, B. Recht, and V. Ye, "Robust guarantees for perception-based control," in *Proc. Learn. Dyn. Control*, PMLR, 2020, pp. 350–360.
[138] S. Yang, G. J. Pappas, R. Mangharam, and L. Lindemann, "Safe perception-based control under stochastic sensor uncertainty using conformal prediction," in *Proc. 62nd IEEE Conf. Decis. Control (CDC)*, Piscataway, NJ, USA: IEEE Press, 2023, pp. 6072–6078, doi: 10.1109/CDC49753.2023.10384075.
[139] X. Li, A. Girard, and I. Kolmanovsky, "Safe adaptive cruise control under perception uncertainty: A deep ensemble and conformal tube model predictive control approach," 2024, *arXiv:2412.03792*.
[140] T. Waite, Y. Geng, T. Turnquist, I. Ruchkin, and R. Ivanov, "State-dependent conformal perception bounds for neuro-symbolic verification of autonomous systems," in *Proc. Int. Conf. Neurosymbolic Syst.*, 2025.
[141] A. Z. Ren et al., "Robots that ask for help: Uncertainty alignment for large language model planners," in *Proc. Mach. Learn. Res.*, 2023.
[142] A. Z. Ren, J. Clark, A. Dixit, M. Itkina, A. Majumdar, and D. Sadigh, "Explore until confident: Efficient exploration for embodied question answering," in *Proc. 1st Workshop Vis.-Lang. Models Navigation Manipulation ICRA*, 2024.
[143] A. Dixit, Z. Mei, M. Booker, M. Storey-Matsutani, A. Z. Ren, and A. Majumdar, "Perceive with confidence: Statistical safety assurances for navigation with learning-based perception," in *Proc. 8th Annu. Conf. Robot Learn.*, 2024, pp. 2517–2541.
[144] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Syst. Mag.*, vol. 36, no. 6, pp. 30–44, Dec. 2016, doi: 10.1109/MCS.2016.2602087.
[145] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017, doi: 10.1109/TAC.2016.2638961.
[146] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. 18th Eur. Control Conf. (ECC)*, Piscataway, NJ, USA: IEEE Press, pp. 3420–3431, 2019, doi: 10.23919/ECC.2019.8796030.
[147] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.
[148] A. Clark, "Control barrier functions for stochastic systems," *Automatica*, vol. 130, Aug. 2021, Art. no. 109688, doi: 10.1016/j.automatica.2021.109688.
[149] S. Yaghoubi, K. Majd, G. Fainekos, T. Yamaguchi, D. Prokhorov, and B. Hoxha, "Risk-bounded control using stochastic barrier functions," *IEEE Contr. Syst. Lett.*, vol. 5, no. 5, pp. 1831–1836, Nov. 2021, doi: 10.1109/LCSYS.2020.3043287.
[150] C. P. Bechlioulis and G. A. Rovithakis, "Robust adaptive control of feedback linearizable MIMO nonlinear systems with prescribed performance," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2090–2099, Oct. 2008, doi: 10.1109/TAC.2008.929402.
[151] A. Ilchmann, "Tracking with prescribed transient behaviour," *ESAIM: Control, Optimisation Calculus Variations*, vol. 7, pp. 471–493, Sep. 2002, doi: 10.1051/cocv:2002064.
[152] F. Mehdifar, L. Lindemann, C. P. Bechlioulis, and D. V. Dimarogonas, "Control of nonlinear systems under multiple time-varying output constraints: A single funnel approach," in *Proc. 62nd IEEE Conf. Decis. Control (CDC)*, Piscataway, NJ, USA: IEEE Press, 2023, pp. 6743–6748, doi: 10.1109/CDC49753.2023.10383359.
[153] S. Sui, C. L. Philip Chen, and S. Tong, "A novel adaptive NN prescribed performance control for stochastic nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 7, pp. 3196–3205, Jul. 2021, doi: 10.1109/TNNLS.2020.3010333.
[154] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE Contr. Syst. Lett.*, vol. 3, no. 1, pp. 96–101, Jan. 2019, doi: 10.1109/LCSYS.2018.2853182.
[155] L. Lindemann and D. V. Dimarogonas, "Barrier function based collaborative control of multiple robots under signal temporal logic tasks," *IEEE Control Netw. Syst.*, vol. 7, no. 4, pp. 1916–1928, Dec. 2020, doi: 10.1109/TCNS.2020.3014602.
[156] L. Lindemann and D. V. Dimarogonas, "Funnel control for fully actuated systems under a fragment of signal temporal logic specifications," *Nonlinear Anal. Hybrid Syst.*, vol. 39, Feb. 2021, Art. no. 100973, doi: 10.1016/j.nahs.2020.100973.
[157] L. Lindemann and D. V. Dimarogonas, "Feedback control strategies for multi-agent systems under a fragment of signal temporal logic tasks," *Automatica*, vol. 106, pp. 284–293, Aug. 2019, doi: 10.1016/j.automatica.2019.05.013.
[158] S. S. Farahani, R. Majumdar, V. S. Prabhu, and S. Soudjani, "Shrinking horizon model predictive control with signal temporal logic constraints under stochastic disturbances," *IEEE Trans. Autom. Control*, vol. 64, no. 8, pp. 3324–3331, Aug. 2019, doi: 10.1109/TAC.2018.2880651.
[159] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *J. Mach. Learn. Res.*, vol. 17, no. 39, pp. 1–40, 2016.
[160] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine, "How to train your robot with deep reinforcement learning: Lessons we have learned," *Int. J. Rob. Res.*, vol. 40, no. 4-5, pp. 698–721, Apr. 2021, doi: 10.1177/0278364920987859.
[161] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Rob. Auton. Syst.*, vol. 57, no. 5, pp. 469–483, 2009, doi: 10.1016/j.robot.2008.10.024.
[162] F. Torabi, G. Warnell, and P. Stone, "Behavioral cloning from observation," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 4950–4957.
[163] M. Srinivasan, A. Dabholkar, S. Coogan, and P. A. Vela, "Synthesis of control barrier functions using a supervised machine learning approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Piscataway, NJ, USA: IEEE Press, 2020, pp. 7139–7145, doi: 10.1109/IROS45743.2020.9341190.
[164] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Proc. Learn. Dyn. Control*, PMLR, 2020, pp. 708–717.
[165] W. Jin, Z. Wang, Z. Yang, and S. Mou, "Neural certificates for safe control policies," 2020, *arXiv:2006.08465*.
[166] L. Lindemann et al., "Learning hybrid control barrier functions from data," in *Proc. Conf. Robot Learn.*, 2021, PMLR, pp. 1351–1370.
[167] L. Lindemann, A. Robey, L. Jiang, S. Das, S. Tu, and N. Matni, "Learning robust output control barrier functions from safe expert demonstrations," *IEEE Open J. Control Syst.*, vol. 3, pp. 158–172, 2024, doi: 10.1109/OJCSYS.2024.3385348.
[168] L. Ljung, "System identification," in *Signal Analysis and Prediction*, A. Procházka, J. Uhlíř, P. W. J. Rayner, and N. G. Kingsbury, Eds., Boston, MA, USA: Birkhäuser, 1998, pp. 163–173.
[169] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. New York, NY, USA: Courier Corporation, 2012.
[170] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence and Robustness*. New York, NY, USA: Courier Corporation, 2011.
[171] N. Matni, A. Proutiere, A. Rantzer, and S. Tu, "From self-tuning regulators to reinforcement learning and back again," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Piscataway, NJ, USA: IEEE Press, 2019, pp. 3724–3740, doi: 10.1109/CDC40024.2019.9029916.
[172] Y. Chen, U. Rosolia, C. Fan, A. Ames, and R. Murray, "Reactive motion planning with probabilistic safety guarantees," in *Proc. Conf. Robot Learn.*, PMLR, 2021, pp. 1958–1970.
[173] C. Yeh, N. Christianson, A. Wu, A. Wierman, and Y. Yue, "End-to-end conformal calibration for optimization under uncertainty," 2024, *arXiv:2409.20534*.
[174] S. Kiyani, G. Pappas, A. Roth, and H. Hassani, "Decision theoretic foundations for conformal prediction: Optimal uncertainty quantification for risk-averse agents," 2025, *arXiv:2502.02561*.
[175] K. J. Strawn, N. Ayanian, and L. Lindemann, "Conformal predictive safety filter for RL controllers in dynamic environments," *IEEE Robot. Autom. Lett.*, vol. 8, no. 11, pp. 7833–7840, Nov. 2023, doi: 10.1109/LRA.2023.3322644.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

[176] K. Driggs-Campbell and R. Dong, "Interaction-aware conformal prediction for crowd navigation," 2024, *arXiv:2502.06221*.
[177] J. L. Contreras, O. Shorinwa, and M. Schwager, "Safe, out-of-distribution-adaptive MPC with conformalized neural network ensembles," in *Proc. 7th Annu. Learn. Dyn. Control Conf.*, PMLR, 2025, pp. 194–207.
[178] M. D. Zhao, H. Admoni, R. Simmons, A. Ramdas, and A. Bajcsy, "Conformalized interactive imitation learning: Handling expert shift and intermittent feedback," in *Proc. 7th Robot Learn. Workshop: Towards Robots Human-Level Abilities*, 2024.
[179] J. Lekeufack, A. N. Angelopoulos, A. Bajcsy, M. I. Jordan, and J. Malik, "Conformal decision theory: Safe autonomous decisions from imperfect predictions," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Piscataway, NJ, USA: IEEE Press, 2024, pp. 11,668–11,675, doi: 10.1109/ICRA57147.2024.10610041.
[180] D. Foffano, A. Russo, and A. Proutiere, "Conformal off-policy evaluation in markov decision processes," in *Proc. 62nd IEEE Conf. Decis. Control (CDC)*, 2023, Piscataway, NJ, USA: IEEE Press, pp. 3087–3094, doi: 10.1109/CDC49753.2023.10383469.
[181] M. F. Taufiq, J.-F. Ton, R. Cornish, Y. W. Teh, and A. Doucet, "Conformal off-policy prediction in contextual bandits," in *Proc. 36th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2022, pp. 31,512–31,524.
[182] T. Kuipers, R. Tumu, S. Yang, M. Kazemi, R. Mangharam, and N. Paoletti, "Conformal off-policy prediction for multi-agent systems," in *Proc. IEEE 63rd Conf. Decis. Control (CDC)*, Piscataway, NJ, USA: IEEE Press, 2024, pp. 1067–1074, doi: 10.1109/CDC56724.2024.10886791.
[183] J. Zhang, B. Hoxha, G. Fainekos, and D. Panagou, "Conformal prediction in the loop: Risk-aware control barrier functions for stochastic systems with data-driven state estimators," *IEEE Contr. Syst. Lett.*, vol. 9, pp. 282–287, 2025, doi: 10.1109/LCSYS.2025.3571828.
[184] T.-W. Hsu and H. Tsukamoto, "Statistical guarantees in data-driven nonlinear control: Conformal robustness for stability and safety," *IEEE Contr. Syst. Lett.*, vol. 9, pp. 997–1002, 2025, doi: 10.1109/LCSYS.2025.3578062.
[185] M. Tayal, A. Singh, P. Jagtap, and S. Kolathaya, "CP-NCBF: A conformal prediction-based approach to synthesize verified neural control barrier functions," 2025, *arXiv:2503.17395*.
[186] S. Wei, M. Ornik, and H. Tsukamoto, "Conformal contraction for robust nonlinear control with distribution-free uncertainty quantification," 2025, *arXiv:2507.13613*.
[187] M. S. Sumeadh, K. Dsouza, and R. Prakash, "CPED-NCBFs: A conformal prediction for expert demonstration-based neural control barrier functions," 2025, *arXiv:2507.15022*.
[188] E. E. Vlahakis, L. Lindemann, and D. V. Dimarogonas, "Conformal data-driven control of stochastic multi-agent systems under collaborative signal temporal logic specifications," 2025, *arXiv:2504.04615*.
[189] K. Y. Chee, T. C. Silva, M. A. Hsieh, and G. J. Pappas, "Uncertainty quantification and robustification of model-based controllers using conformal prediction," in *Proc. 6th Annu. Learn. Dyn. Control Conf.*, PMLR, 2024, pp. 528–540.
[190] N. Gupta, S. Nath, and S. E. Kahou, "CAMMARL: Conformal action modeling in multi agent reinforcement learning," 2023, *arXiv:2306.11128*.
[191] J. Wang, G. He, and Y. Kantaros, "Probabilistically correct language-based multi-robot planning using conformal prediction," *IEEE Robot. Autom. Lett.*, vol. 10, no. 1, pp. 160–167, Jan. 2025, doi: 10.1109/LRA.2024.3504233.
[192] S. Huriot and H. Sibai, "Safe decentralized multi-agent control using black-box predictors, conformal decision policies, and control barrier functions," 2024, *arXiv:2409.18862*.
[193] J. Wang, J. Tong, K. Tan, Y. Vorobeychik, and Y. Kantaros, "Conformal temporal logic planning using large language models: Knowing when to do what and when to ask for help," 2023, *arXiv:2309.10092*.
[194] J. Sun, Y. Jiang, J. Qiu, P. Nobel, M. J. Kochenderfer, and M. Schwager, "Conformal prediction for uncertainty-aware planning with diffusion dynamics model," in *Proc. 37th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2024, pp. 80,324–80,337.
[195] J. Lidard, H. Pham, A. Bachman, B. Boateng, and A. Majumdar, "Risk-calibrated human-robot interaction via set-valued intent prediction," in *Proc. ICRA Workshop Back Future: Robot Learn. Going Probabilistic*, 2024.
[196] M. Zhao, R. Simmons, H. Admoni, and A. Bajcsy, "Conformalized teleoperation: Confidently mapping human inputs to high-dimensional robot actions," 2024, *arXiv:2406.07767*.
[197] T. Beckers, D. Kulić, and S. Hirche, "Stable Gaussian process based tracking control of Euler–Lagrange systems," *Automatica*, vol. 103, pp. 390–397, May 2019, doi: 10.1016/j.automatica.2019.01.023.
[198] M. Peter Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 408–423, Feb. 2015, doi: 10.1109/TPAMI.2013.218.
[199] A. Jain, T. Nghiem, M. Morari, and R. Mangharam, "Learning and control using Gaussian processes," in *Proc. ACM/IEEE 9th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, 2018, Piscataway, NJ, USA: IEEE Press, pp. 140–149, doi: 10.1109/ICCPS.2018.00022.
[200] M. Liu, G. Chowdhary, B. Castra Da Silva, S.-Y. Liu, and J. P. How, "Gaussian processes for learning and control: A tutorial with examples," *IEEE Control Syst. Mag.*, vol. 38, no. 5, pp. 53–86, Oct. 2018, doi: 10.1109/MCS.2018.2851010.
[201] G. C. Calafiore and L. Fagiano, "Robust model predictive control via scenario optimization," *IEEE Trans. Autom. Control*, vol. 58, no. 1, pp. 219–224, Jan. 2013, doi: 10.1109/TAC.2012.2203054.
[202] G. Schildbach, L. Fagiano, C. Frei, and M. Morari, "The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations," *Automatica*, vol. 50, no. 12, pp. 3009–3018, 2014, doi: 10.1016/j.automatica.2014.10.035.
[203] A. A. D. Nascimento, A. Papachristodoulou, and K. Margellos, "Probabilistically safe controllers based on control barrier functions and scenario model predictive control," in *Proc. IEEE 63rd Conf. Decis. Control (CDC)*, Piscataway, NJ, USA: IEEE Press, 2024, pp. 1814–1819, doi: 10.1109/CDC56724.2024.10886282.
[204] A. Majumdar, A. Farid, and A. Sonar, "PAC-Bayes control: Learning policies that provably generalize to novel environments," *Int. J. Rob. Res.*, vol. 40, nos. 2–3, pp. 574–593, 2021, doi: 10.1177/0278364920959444.
[205] A. Majumdar and M. Goldstein, "PAC-Bayes control: Synthesizing controllers that provably generalize to novel environments," in *Proc. Conf. Robot Learn.*, PMLR, 2018, pp. 293–305.
[206] S. Safaoui, B. J. Gravell, V. Renganathan, and T. H. Summers, "Risk-averse RRT* planning with nonlinear steering and tracking controllers for nonlinear robotic systems under uncertainty," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Piscataway, NJ, USA: IEEE Press, 2021, pp. 3681–3688, doi: 10.1109/IROS51168.2021.9636834.
[207] A. Dixit, M. Ahmadi, and J. W. Burdick, "Risk-sensitive motion planning using entropic value-at-risk," in *Proc. Eur. Control Conf. (ECC)*, Piscataway, NJ, USA: IEEE Press, 2021, pp. 1726–1732, doi: 10.23919/ECC54610.2021.9655104.
[208] M. Ahmadi, X. Xiong, and A. D. Ames, "Risk-averse control via CVaR barrier functions: Application to bipedal robot locomotion," *IEEE Contr. Syst. Lett.*, vol. 6, pp. 878–883, 2022, doi: 10.1109/LCSYS.2021.3086854.
[209] H. Hu, M. Fazlyab, M. Morari, and G. J. Pappas, "Reach-SDP: Reachability analysis of closed-loop systems with neural network controllers via semidefinite programming," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, Piscataway, NJ, USA: IEEE Press, 2020, pp. 5929–5934, doi: 10.1109/CDC42340.2020.9304296.
[210] S. Chen, V. M. Preciado, and M. Fazlyab, "One-shot reachability analysis of neural network dynamical systems," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Piscataway, NJ, USA: IEEE Press, 2023, pp. 10,546–10,552, doi: 10.1109/ICRA48891.2023.10160643.
[211] S. Chen, M. Fazlyab, M. Morari, G. J. Pappas, and V. M. Preciado, "Learning Lyapunov functions for piecewise affine systems with neural network controllers," 2020, *arXiv:2008.06546*.
[212] R. Mazouz, K. Muvvala, A. R. Babu, L. Laurenti, and M. Lahijanian, "Safety guarantees for neural network dynamic systems via stochastic barrier functions," in *Proc. 36th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2022, pp. 9672–9686.
[213] C. Huang, J. Fan, W. Li, X. Chen, and Q. Zhu, "ReachNN: Reachability analysis of neural-network controlled systems," *ACM Trans. Embedded Comput. Syst.*, vol. 18, no. 5s, pp. 1–22, 2019, doi: 10.1145/3358228.
[214] H.-D. Tran et al., "NNV: The neural network verification tool for deep neural networks and learning-enabled cyber-physical systems," in *Proc. Int. Conf. Comput. Aided Verification*, Cham, Switzerland: Springer-Verlag, 2020, pp. 3–17.
[215] S. Dutta, X. Chen, and S. Sankaranarayanan, "Reachability analysis for neural feedback systems using regressive polynomial rule inference," in *Proc. 22nd ACM Int. Conf. Hybrid Syst.: Comput. Control*, 2019, pp. 157–168.
[216] R. Ivanov, J. Weimer, R. Alur, G. J. Pappas, and I. Lee, "Verisig: Verifying safety properties of hybrid systems with neural network controllers," in *Proc. ACM Int. Conf. Hybrid Syst.: Comput. Control*, 2019, pp. 169–178.
[217] R. Ivanov, T. J. Carpenter, J. Weimer, R. Alur, G. J. Pappas, and I. Lee, "Verifying the safety of autonomous systems with neural network

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

controllers," *ACM Trans. Embedded Comput. Syst.*, vol. 20, no. 1, pp. 1–26, 2021, doi: 10.1145/3419742.
[218] R. Ivanov, T. J. Carpenter, J. Weimer, R. Alur, G. J. Pappas, and I. Lee, "Case study: Verifying the safety of an autonomous racing car with a neural network controller," in *Proc. 23rd Int. Conf. Hybrid Syst.: Comput. Control*, 2020, pp. 1–7.
[219] H.-D. Tran, F. Cai, M. L. Diego, P. Musau, T. T. Johnson, and X. Koutsoukos, "Safety verification of cyber-physical systems with reinforcement learning control," *ACM Trans. Embedded Comput. Syst.*, vol. 18, no. 5s, pp. 1–22, 2019, doi: 10.1145/3358230.
[220] X. Sun, H. Khedr, and Y. Shoukry, "Formal verification of neural network controlled autonomous systems," in *Proc. 22nd ACM Int. Conf. Hybrid Syst.: Comput. Control*, 2019, pp. 147–156.
[221] S. Bak, C. Liu, and T. Johnson, "The second international verification of neural networks competition (VNN-COMP 2021): Summary and results," 2021, *arXiv:2109.00498*.
[222] S. Bak and H.-D. Tran, "Neural network compression of ACAS Xu early prototype is unsafe: Closed-loop verification through quantized state backreachability," in *Proc. NASA Formal Methods Symp.*, Cham, Switzerland: Springer-Verlag, 2022, pp. 280–298.
[223] T. Dreossi, A. Donzé, and S. A. Seshia, "Compositional falsification of cyber-physical systems with machine learning components," *J. Automated Reasoning*, vol. 63, no. 4, pp. 1031–1053, 2019, doi: 10.1007/s10817-018-09509-5.
[224] A. Bianco and L. d Alfaro, "Model checking of probabilistic and nondeterministic systems," in *Proc. Int. Conf. Found. Softw. Technol. Theor. Comput. Sci.*, Berlin, Germany: Springer-Verlag, 1995, pp. 499–513.
[225] H. Hansson and B. Jonsson, "A logic for reasoning about time and reliability," *Formal Aspects Comput.*, vol. 6, no. 5, pp. 512–535, 1994, doi: 10.1007/BF01211866.
[226] J.-P. Katoen, M. Khattri, and I. S. Zapreevt, "A Markov reward model checker," in *Proc. 2nd Int. Conf. Quantitative Eval. Syst. (QEST)*, Piscataway, NJ, USA: IEEE Press, 2005, pp. 243–244, doi: 10.1109/QEST.2005.2.
[227] M. Kwiatkowska, G. Norman, and D. Parker, "Stochastic model checking," in *Proc. Int. School Formal Methods Des. Comput., Commun. Softw. Syst.*, Berlin, Germany: Springer-Verlag, 2007, pp. 220–270.
[228] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in *Proc. Int. Conf. Comput. Aided Verification*, Berlin, Germany: Springer-Verlag, 2011, pp. 585–591.
[229] C. Daws, "Symbolic and parametric model checking of discrete-time Markov chains," in *Proc. Int. Colloq. Theor. Aspects Comput.*, Berlin, Germany: Springer-Verlag, 2005, pp. 280–294.
[230] E. M. Hahn, H. Hermanns, B. Wachter, and L. Zhang, "PARAM: A model checker for parametric Markov models," in *Proc. Int. Conf. Comput. Aided Verification*, Berlin, Germany: Springer-Verlag, 2010, pp. 660–664.
[231] E. M. Hahn, H. Hermanns, and L. Zhang, "Probabilistic reachability for parametric Markov models," *Int. J. Softw. Tools Technol. Trans.*, vol. 13, no. 1, pp. 3–19, 2011, doi: 10.1007/s10009-010-0146-x.
[232] E. M. Hahn, T. Han, and L. Zhang, "Synthesis for PCTL in parametric Markov decision processes," in *Proc. NASA Formal Methods Symp.*, Cham, Switzerland: Springer-Verlag, 2011, pp. 146–161.
[233] L. Bortolussi, F. Cairoli, N. Paoletti, S. A. Smolka, and S. D. Stoller, "Neural predictive monitoring," in *Proc. Runtime Verification: 19th Int. Conf.*, 2019, pp. 129–147.
[234] F. Cairoli, L. Bortolussi, and N. Paoletti, "Neural predictive monitoring under partial observability," in *Proc. Int. Conf. Runtime Verification*, Cham, Switzerland: Springer-Verlag, 2021, pp. 121–141.
[235] F. Cairoli, N. Paoletti, and L. Bortolussi, "Conformal quantitative predictive monitoring of STL requirements for stochastic processes," in *Proc. 26th ACM Int. Conf. Hybrid Syst.: Comput. Control*, 2023, pp. 1–11.
[236] N. Hashemi, X. Qin, L. Lindemann, and J. V. Deshmukh, "Data-driven reachability analysis of stochastic dynamical systems with conformal inference," in *Proc. 62nd IEEE Conf. Decis. Control (CDC)*, Piscataway, NJ, USA: IEEE Press, 2023, pp. 3102–3109, doi: 10.1109/CDC49753.2023.10384213.
[237] N. Hashemi, L. Lindemann, and J. V. Deshmukh, "Statistical reachability analysis of stochastic cyber-physical systems under distribution shift," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 43, no. 11, pp. 4250–4261, Nov. 2024, doi: 10.1109/TCAD.2024.3438072.
[238] Y. Kwon et al., "Conformalized reachable sets for obstacle avoidance with spheres," 2024, *arXiv:2410.09924*.
[239] A. Tebjou, G. Frehse, and F. Chamroukhi, "Data-driven reachability using Christoffel functions and conformal prediction," in *Proc. Conformal Probabilistic Prediction Appl.*, PMLR, 2023, pp. 194–213.
[240] A. Devonport, F. Yang, L. El Ghaoui, and M. Arcak, "Data-driven reachability analysis with Christoffel functions," in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, Piscataway, NJ, USA: IEEE Press, 2021, pp. 5067–5072, doi: 10.1109/CDC45484.2021.9682860.
[241] A. Devonport and M. Arcak, "Estimating reachable sets with scenario optimization," in *Proc. Learn. Dyn. Control*, PMLR, 2020, pp. 75–84.
[242] E. Dietrich, A. Devonport, and M. Arcak, "Nonconvex scenario optimization for data-driven reachability," in *Proc. 6th Annu. Learn. Dyn. Control Conf.*, PMLR, 2024, pp. 514–527.
[243] X. Qin, Y. Xia, A. Zutshi, C. Fan, and J. V. Deshmukh, "Statistical verification of cyber-physical systems using surrogate models and conformal inference," in *Proc. ACM/IEEE 13th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Piscataway, NJ, USA: IEEE Press, 2022, pp. 116–126, doi: 10.1109/ICCPS54341.2022.00017.
[244] X. Qin, Y. Xia, A. Zutshi, C. Fan, and J. V. Deshmukh, "Statistical verification using surrogate models and conformal inference and a comparison with risk-aware verification," *ACM Trans. Cyber-Phys. Syst.*, vol. 8, no. 2, pp. 1–25, 2024, doi: 10.1145/3635160.
[245] L. Lindemann, X. Qin, J. V. Deshmukh, and G. J. Pappas, "Conformal prediction for STL runtime verification," in *Proc. ACM/IEEE 14th Int. Conf. Cyber-Phys. Syst. (CPS-IoT Week)*, 2023, pp. 142–153, doi: 10.1145/3576841.358592.
[246] Y. Zhao, B. Hoxha, G. Fainekos, J. V. Deshmukh, and L. Lindemann, "Robust conformal prediction for STL runtime verification under distribution shift," in *Proc. ACM/IEEE 15th Int. Conf. Phys. Syst. (ICCPS)*, Piscataway, NJ, USA: IEEE Press, 2024, pp. 169–179, doi: 10.1109/ICCPS61052.2024.00022.
[247] F. E. Grubbs, "On designing single sampling inspection plans," *Ann. Math. Statist.*, vol. 20, no. 2, pp. 242–256, 1949, doi: 10.1214/aoms/1177730033.
[248] H. L. S. Younes, *Verification and Planning for Stochastic Processes With Asynchronous Events*. Pittsburgh, PA, USA: Carnegie Mellon Univ., 2004.
[249] A. Wald, "Sequential tests of statistical hypotheses," in *Breakthroughs in Statistics: Foundations and Basic Theory*, New York, NY, USA: Springer-Verlag, 1992, pp. 256–298.
[250] Y. Wang, M. Zarei, B. Bonakdarpour, and M. Pajic, "Statistical verification of hyperproperties for cyber-physical systems," *ACM Trans. Embedded Comput. Syst.*, vol. 18, no. 5s, pp. 1–23, 2019, doi: 10.1145/3358232.
[251] M. Zarei, Y. Wang, and M. Pajic, "Statistical verification of learning-based cyber-physical systems," in *Proc. 23rd Int. Conf. Hybrid Syst.: Comput. Control*, 2020, pp. 1–7.
[252] K. G. Larsen and A. Legay, "Statistical model checking: Past, present, and future," in *Proc. Int. Symp. Leveraging Appl. Formal Methods, Verification Validation*, Berlin, Germany: Springer-Verlag, 2016, pp. 3–15.
[253] G. Agha and K. Palmskog, "A survey of statistical model checking," *ACM Trans. Model. Comput. Simul.*, vol. 28, no. 1, pp. 1–39, 2018, doi: 10.1145/3158668.
[254] A. Legay, A. Lukina, L. M. Traonouez, J. Yang, S. A. Smolka, and R. Grosu, "Statistical model checking," in *Computing and Software Science: State of the Art and Perspectives*, B. Steffen and G. Woeginger, Eds., Cham, Switzerland: Springer-Verlag, 2019, pp. 478–504.
[255] K. Sen, M. Viswanathan, and G. Agha, "Statistical model checking of black-box probabilistic systems," in *Proc. Int. Conf. Comput. Aided Verification*, Berlin, Germany: Springer-Verlag, 2004, pp. 202–215.
[256] H. Kahn and A. W. Marshall, "Methods of reducing sample size in Monte Carlo computations," *J. Oper. Res. Soc. Amer.*, vol. 1, no. 5, pp. 263–278, 1953, doi: 10.1287/opre.1.5.263.
[257] D. Zhao et al., "Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 595–607, Mar. 2017, doi: 10.1109/TITS.2016.2582208.
[258] Y. Kim and M. J. Kochenderfer, "Improving aircraft collision risk estimation using the cross-entropy method," *J. Air Transp.*, vol. 24, no. 2, pp. 55–62, 2016, doi: 10.2514/1.D0020.
[259] J. Uesato et al., "Rigorous agent evaluation: An adversarial approach to uncover catastrophic failures," in *Proc. Int. Conf. Learn. Representations*, 2018.
[260] M. O'Kelly, A. Sinha, H. Namkoong, R. Tedrake, and J. C. Duchi, "Scalable end-to-end autonomous vehicle testing via rare-event simulation," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2018, pp. 9849–9860.
[261] H. Kahn and T. E. Harris, "Estimation of particle transmission by random sampling," *Nat. Bur. Standards Appl. Math. Ser.*, vol. 12, pp. 27–30, 1951.
[262] F. Cérou and A. Guyader, "Adaptive multilevel splitting for rare event analysis," *Stochastic Anal. Appl.*, vol. 25, no. 2, pp. 417–443, 2007, doi: 10.1080/07362990601139628.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.

[263] C. Jegourel, A. Legay, and S. Sedwards, "Importance splitting for statistical model checking rare properties," in *Proc. Int. Conf. Comput. Aided Verification*, Berlin, Germany: Springer-Verlag, 2013, pp. 576–591.
[264] C. Jegourel, A. Legay, and S. Sedwards, "An effective heuristic for adaptive importance splitting in statistical model checking," in *Proc. Int. Symp. Leveraging Appl. Formal Methods, Verification Validation*, Berlin, Germany: Springer-Verlag, 2014, pp. 143–159.
[265] J. Norden, M. O'Kelly, and A. Sinha, "Efficient black-box assessment of autonomous vehicle safety," 2019, *arXiv:1912.03618*.
[266] M. P. Chapman, R. Bonalli, K. M. Smith, I. Yang, M. Pavone, and C. J. Tomlin, "Risk-sensitive safety analysis using conditional value-at-risk," *IEEE Trans. Autom. Control*, vol. 67, no. 12, pp. 6521–6536, Dec. 2022, doi: 10.1109/TAC.2021.3131149.
[267] L. Lindemann, A. Rodionova, and G. Pappas, "Temporal robustness of stochastic signals," in *Proc. 25th ACM Int. Conf. Hybrid Syst.: Comput. Control*, 2022, pp. 1–11.
[268] P. Akella, M. Ahmadi, and A. D. Ames, "A scenario approach to risk-aware safety-critical system verification," 2022, *arXiv:2203.02595*.
[269] P. Akella, A. Dixit, M. Ahmadi, J. W. Burdick, and A. D. Ames, "Sample-based bounds for coherent risk measures: Applications to policy synthesis and verification," *Artif. Intell.*, vol. 336, Nov. 2024, Art. no. 104195, doi: 10.1016/j.artint.2024.104195.
[270] L. Lindemann, L. Jiang, N. Matni, and G. J. Pappas, "Risk of stochastic systems for temporal logic specifications," *ACM Trans. Embedded Comput. Syst.*, vol. 22, no. 3, pp. 1–31, 2023, doi: 10.1145/3580490.
[271] Y. Wang, M. Zarei, B. Bonakdarpoor, and M. Pajic, "Probabilistic conformance for cyber-physical systems," in *Proc. ACM/IEEE 12th Int. Conf. Cyber-Phys. Syst.*, 2021, pp. 55–66.
[272] S. Sadraddini and C. Belta, "Robust temporal logic model predictive control," in *Proc. 53rd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Piscataway, NJ, USA: IEEE Press, 2015, pp. 772–779, doi: 10.1109/ALLERTON.2015.7447084.
[273] P. Heidlauf, A. Collins, M. Bolender, and S. Bak, "Verification challenges in F-16 ground collision avoidance and other automated maneuvers," in *Proc. 5th Int. Workshop Appl. Verification Continuous Hybrid Syst.*, 2018, pp. 208–217.
[274] R. Freeman and P. V. Kokotovic, *Robust Nonlinear Control Design: State-Space and Lyapunov Techniques*. Boston, MA, USA: Birkhäuser, 2008.
[275] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Upper Saddle River, NJ, USA: Prentice Hall, 1996.
[276] I. Cassar, A. Francalanza, L. Aceto, and A. Ingólfsdóttir, "A survey of runtime monitoring instrumentation techniques," in *Proc. 2nd Int. Workshop Pre- Post-Deployment Verification Techn., PrePost@ iFM*, 2017, pp. 15–28.
[277] H.-M. Ho, J. Ouaknine, and J. Worrell, "Online monitoring of metric temporal logic," in *Proc. Int. Conf. Runtime Verification*, Cham, Switzerland: Springer-Verlag, 2014, pp. 178–192.
[278] M. Abate, E. Feron, and S. Coogan, "Monitor-based runtime assurance for temporal logic specifications," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Piscataway, NJ, USA: IEEE Press, 2019, pp. 1997–2002, doi: 10.1109/CDC40024.2019.9029716.
[279] F. Lercher and M. Althoff, "Using four-valued signal temporal logic for incremental verification of hybrid systems," in *Proc. Int. Conf. Comput. Aided Verification*, Cham, Switzerland: Springer-Verlag, 2024, pp. 259–281.
[280] C. Mascle, D. Neider, M. Schwenger, P. Tabuada, A. Weinert, and M. Zimmermann, "From LTL to rLTL monitoring: Improved monitorability through robust semantics," in *Proc. 23rd Int. Conf. Hybrid Syst.: Computat. Control*, 2020, pp. 1–12.
[281] A. Dokhanchi, B. Hoxha, and G. Fainekos, "On-line monitoring for temporal logic robustness," in *Proc. Int. Conf. Runtime Verification*, 2014, Cham, Switzerland: Springer-Verlag, pp. 231–246.
[282] E. Bartocci et al., "Specification-based monitoring of cyber-physical systems: A survey on theory, tools and applications," in *Lectures on Runtime Verification (Lecture Notes in Computer Science)*, E. Bartocci and Y. Falcone, Eds., Cham, Switzerland: Springer-Verlag, 2018, pp. 135–175.
[283] S. Jakšić, E. Bartocci, R. Grosu, and D. Ničković, "An algebraic framework for runtime verification," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 11, pp. 2233–2243, 2018, doi: 10.1109/TCAD.2018.2858460.
[284] E. Visconti, E. Bartocci, M. Loreti, and L. Nenzi, "Online monitoring of spatio-temporal properties for imprecise signals," in *Proc. 19th ACM-IEEE Int. Conf. Formal Methods Models Syst. Des.*, 2021, pp. 78–88, doi: 10.1145/3487212.3487344.
[285] X. Yu, W. Dong, S. Li, and X. Yin, "Model predictive monitoring of dynamical systems for signal temporal logic specifications," *Automatica*, vol. 160, Feb. 2024, Art. no. 111445, doi: 10.1016/j.automatica.2023.111445.
[286] B. Ghosh and É. André, "Offline and online monitoring of scattered uncertain logs using uncertain linear dynamical systems," in *Proc. Int. Conf. Formal Techn. Distrib. Objects, Compon., Syst.*, Cham, Switzerland: Springer-Verlag, 2022, pp. 67–87.
[287] H. Abbas and B. Bonakdarpour, "Leveraging system dynamics in runtime verification of cyber-physical systems," in *Proc. Int. Symp. Leveraging Appl. Formal Methods*, Cham, Switzerland: Springer-Verlag, 2022, pp. 264–278.
[288] X. Qin and J. V. Deshmukh, "Clairvoyant monitoring for signal temporal logic," in Proc. Int. Conf. Formal Model. Anal. Timed Syst., Cham, Switzerland: Springer-Verlag, 2020, pp. 178–195.
[289] R. Babaee, A. Gurfinkel, and S. Fischmeister, "Predictive run-time verification of discrete-time reachability properties in black-box systems using trace-level abstraction and statistical learning," in *Proc. Int. Conf. Runtime Verification*, Cham, Switzerland: Springer-Verlag, 2018, pp. 187–204.
[290] A. Prasad Sistla, M. Žefran, and Y. Feng, "Runtime monitoring of stochastic cyber-physical systems with hybrid state," in *Proc. Int. Conf. Runtime Verification*, Berlin, Germany: Springer-Verlag, 2011, pp. 276–293.
[291] M. Leucker, "Sliding between model checking and runtime verification," in *Proc. Int. Conf. Runtime Verification*, Berlin, Germany: Springer-Verlag, 2012, pp. 82–87.
[292] X. Zhang, M. Leucker, and W. Dong, "Runtime verification with predictive semantics," in *Proc. NASA Formal Methods Symp.*, Berlin, Germany: Springer-Verlag, 2012, pp. 418–432.
[293] M. Ma, J. Stankovic, E. Bartocci, and L. Feng, "Predictive monitoring with logic-calibrated uncertainty for cyber-physical systems," *ACM Trans. Embedded Comput. Syst.*, vol. 20, no. 5s, pp. 1–25, 2021, doi: 10.1145/3477032.
[294] H. Yoon and S. Sankaranarayanan, "Predictive runtime monitoring for mobile robots using logic-based Bayesian intent inference," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Piscataway, NJ, USA: IEEE Press, 2021, pp. 8565–8571, doi: 10.1109/ICRA48506.2021.9561193.
[295] C. M. Wilcox and B. C. Williams, "Runtime verification of stochastic, faulty systems," in *Proc. Int. Conf. Runtime Verification*, Berlin, Germany: Springer-Verlag, 2010, pp. 452–459.
[296] M. Jaeger, K. G. Larsen, and A. Tibo, "From statistical model checking to run-time monitoring using a Bayesian network approach," in *Proc. Int. Conf. Runtime Verification*, Cham, Switzerland: Springer-Verlag, 2020, pp. 517–535.
[297] Z. Zhang, J. An, P. Arcaini, and I. Hasuo, "Online causation monitoring of signal temporal logic," in *Proc. Int. Conf. Comput. Aided Verification*, Cham, Switzerland: Springer-Verlag, 2023, pp. 62–84.
[298] F. Cai and X. Koutsoukos, "Real-time out-of-distribution detection in learning-enabled cyber-physical systems," in *Proc. ACM/IEEE 11th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Piscataway, NJ, USA: IEEE Press, 2020, pp. 174–183, doi: 10.1109/ICCPS48487.2020.00024.
[299] F. Cai, A. I. Ozdagli, N. Potteiger, and X. Koutsoukos, "Inductive conformal out-of-distribution detection based on adversarial autoencoders," in *Proc. IEEE Int. Conf. Omni-Layer Intell. Syst. (COINS)*, Piscataway, NJ, USA: IEEE Press, 2021, pp. 1–6, doi: 10.1109/COINS51742.2021.9524167.
[300] R. Kaur et al., "CODiT: Conformal out-of-distribution detection in time-series data for cyber-physical systems," in *Proc. ACM/IEEE 14th Int. Conf. on Cyber-Phys. Syst. (CPS-IoT Week)*, 2023, pp. 120–131.
[301] M. Bashari, M. Sesia, and Y. Romano, "Robust conformal outlier detection under contaminated reference data," in *Proc. 42nd Int. Conf. Mach. Learn.*, 2025.
[302] C. Xu and Y. Xie, "Conformal anomaly detection on spatio-temporal observations with missing data," 2021, *arXiv:2105.11886*.
[303] R. Laxhammar, "Conformal anomaly detection: Detecting abnormal trajectories in surveillance applications," PhD thesis, Univ. of Skövde, Skövde, Sweden, 2014.
[304] R. Laxhammar and G. Falkman, "Inductive conformal anomaly detection for sequential detection of anomalous sub-trajectories," *Ann. Math. Artif. Intell.*, vol. 74, nos. 1–2, pp. 67–94, 2015, doi: 10.1007/s10472-013-9381-7.
[305] R. Luo et al., "Sample-efficient safety assurances using conformal prediction," in *Proc. Int. Workshop Algorithmic Found. Robot.*, Cham, Switzerland: Springer-Verlag, 2022, pp. 149–169.
[306] R. Sinha, E. Schmerling, and M. Pavone, "Closing the loop on runtime monitors with fallback-safe MPC," in *Proc. 62nd IEEE Conf. Decis. Control (CDC)*, Piscataway, NJ, USA: IEEE Press, 2023, pp. 6533–6540, doi: 10.1109/CDC49753.2023.10383965.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 13:02:28 UTC from IEEE Xplore. Restrictions apply.