Sample-Efficient Safety Assurances using Conformal Prediction

Rachel Luo[1], Shengjia Zhao[1], Jonathan Kuck[2], Boris Ivanovic[1], Silvio Savarese[1], Edward Schmerling[1], and Marco Pavone[1] *

[1] Stanford University, Stanford, CA 94305, USA
{rsluo, sjzhao, borisi, ssilvio, schmrlng, pavone}@stanford.edu
[2] Dexterity, Inc., Redwood City, CA 94063, USA
jonathan@dexterity.ai

Abstract. When deploying machine learning models in high-stakes robotics applications, the ability to detect unsafe situations is crucial. Early warning systems can provide alerts when an unsafe situation is imminent (in the absence of corrective action). To reliably improve safety, these warning systems should have a *provable* false negative rate; i.e. of the situations that are unsafe, fewer than ϵ will occur without an alert. In this work, we present a framework that combines a statistical inference technique known as conformal prediction with a simulator of robot/environment dynamics, in order to tune warning systems to provably achieve an ϵ false negative rate using as few as $1/\epsilon$ data points. We apply our framework to a driver warning system and a robotic grasping application, and empirically demonstrate guaranteed false negative rate while also observing low false detection (positive) rate.

Keywords: Safety assurance · Conformal prediction · Statistical inference

1 Introduction

Monitoring a system for faults, or detecting if unsafe situations will occur is a key problem for high-stakes robotics applications, and indeed the field of fault detection has long been the state of practice for building reliable systems [30, 29, 31, 28, 16, 21, 7, 8, 23, 13, 14]. With the advent of learning-enabled components in robotic systems, robots are performing increasingly complex safety-critical tasks, so reliability has become increasingly important. At the same time, it is less clear how to ensure reliability for these learned systems.

In this work, we present a sample efficient and principled method for detecting unsafe situations based on the statistical inference technique of conformal prediction [32]. Our method provides *provable* false negative rates for warning systems (i.e. among the situations in which an alert should be issued, fewer than ϵ occur without an alert), while achieving low false positive rates (few unnecessary alerts are issued). As a running example, we use our method to design

* The NASA University Leadership Initiative (grant #80NSSC20M0163) provided funds to assist the authors with their research, but this article solely reflects the opinions and conclusions of its authors and not any NASA entity.

an alert system to warn a human operator of impending danger in a driving application (illustrated in Figure 1).

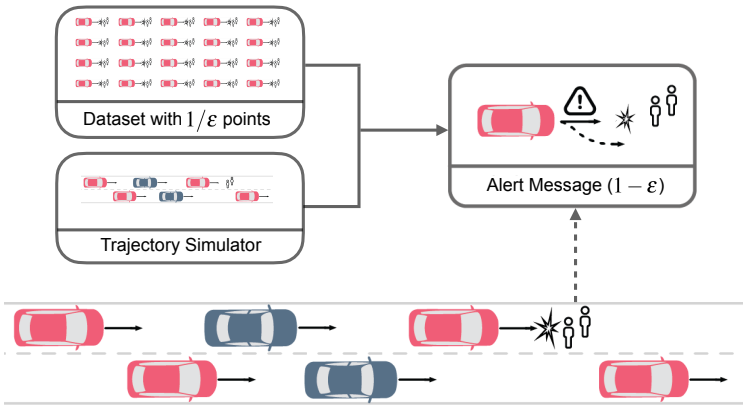


Fig. 1: We design a warning system that achieves a *provable* false negative rate sample efficiently. Among the situations that are dangerous (i.e. lead to an unsafe future situation in the absence of corrective action), fewer than ϵ occur without an alert.

Related Work Traditional fault detection techniques include hardware redundancy, signal processing, and plausibility tests [8, 30, 29, 31, 28]. However, hardware redundancy requires extra components, signal processing works well only for processes in steady state, and plausibility tests do not catch faults that lead to a physically plausible system. Additionally, these methods typically lack performance guarantees. Model-based fault detection techniques [8, 23, 13, 14] involve using a model of the system to determine whether a fault has occurred; they assume that users have a very accurate model of the system dynamics, which is difficult to obtain in practice.

Another common approach for detecting unsafe states employs supervised learning to train a classifier model for labeling states as unsafe, and then the classifier hyperparameters are adjusted until empirically the false negative rate is low. In practice this is typically accomplished by plotting a receiver operating characteristic (ROC) curve and tuning the classification threshold to achieve low false negative rate. However, this approach requires training a new classification model, and provides no performance guarantees.

To guarantee the false negative rate of a learned warning system, the standard statistical learning framework could be used under standard i.i.d. assumptions [17]. A practitioner could collect additional data and use a validation dataset to provably certify the false negative rate. However, the key problem is *data efficiency*, because collecting data for unsafe situations can be very expensive [11, 17, 4].

Main Question Can we tune a warning system and guarantee low false negative rate with only a handful of data points? For example, with only 30 data samples of dangerous situations, can we tune a warning system to have

provable 5% false negative rate? This problem is easy if we allow trivial systems that always issue a warning, but such systems are not practically useful. If we restrict our attention to non-trivial systems, this problem is seemingly impossible because even if a *fixed* warning system successfully identifies all 30 dangerous situations, due to statistical fluctuations, we cannot prove that its false negative rate is less than 5% (with high confidence). If proving that a fixed predictor achieves safety is difficult, tuning a predictor to provably achieve safety seems only more challenging.

Our Contribution We answer our main question affirmatively. We adapt a statistical inference framework known as conformal prediction to a robotics setting in order to tune systems to achieve provable safety guarantees (e.g. 5% false negative rate) with extremely limited data (e.g. 30 samples). We only require a single assumption: the training samples are exchangeable with each test sample, i.e. for each test sample, if we permute the concatenated sequence of the training samples and the test sample, there is no reason to believe that any permutation is more or less likely to occur. This is a weaker assumption than the i.i.d. assumption typically used in the statistical learning framework.

The assumption is reasonable in practice, even in situations in which the statistical inference assumptions fail (e.g. situations with temporal correlations between different test samples). In a driving scenario, for instance, the training dataset could include scene snippets sampled from different scenes, and these will be i.i.d. At test time, there may be one scene with several snippets. These snippets are obviously not independent; however, they are individually exchangeable with the training dataset.

While this answer seems too good to be true, the key insight here is that we provide a type of guarantee that is different from standard statistical learning guarantees. Consider a sequence of test samples $Z_1, \cdots, Z_N$, and event indicators $F_1, \cdots, F_N$ for whether our warning system fails on each test sample.

- **Statistical learning guarantee.** In the statistical learning framework, we assume that the test samples $Z_1, \cdots, Z_N$ are i.i.d., so the failure events $F_1, \cdots, F_N$ are also i.i.d. — we guarantee the failure probability for a sequence of i.i.d. failure events.
- **Conformal prediction guarantee.** In the conformal prediction framework, the test samples are not necessarily i.i.d., so the failure events $F_1, \cdots, F_N$ can be correlated — we guarantee the *marginal* failure probability for each failure event. In other words, we know that each test sample has a low probability of failure (i.e. $F_n = 1$ with low probability), but the failures could be correlated. For example, conditioning on $F_n = 1$ might increase or decrease the probability that $F_{n+1} = 1$ (while in the i.i.d. case, F_n and F_{n+1} are independent events).

The usefulness of the conformal guarantee depends on the intended application. Consider the driver alert system example: for individual drivers, collisions are rare and most drivers will not encounter more than one. Hence, there is little reason to worry about whether the warning failures are correlated between collisions. In other words, the conformal guarantee can convey confidence to individual users who rarely encounter multiple failures.

On the other hand, the conformal guarantee may convey less confidence to a company with a large fleet of vehicles. For example, if $F_n = 1$ increases the probability that $F_{n+1} = 1$, then it is possible to have multiple simultaneous failures. However, this is not a limitation of our method, but rather an unavoidable consequence of the weaker (not i.i.d.) assumptions: if the test data is correlated (which we have no control over), then failure events of a warning system are inherently correlated. The weaker assumption is usually necessary because most robotics applications are deployed in time series or sequential decision making setups, so data from nearby time steps are correlated and not i.i.d. Since standard statistical learning guarantees are not applicable due to violation of the i.i.d. assumption, having some (conformal) guarantee is better than none.

Furthermore, we will show empirically in Section 4 that failures are not highly correlated on two real-world driving datasets. Therefore, despite the lack of formal guarantees, there is strong empirical evidence suggesting that simultaneous failures do not occur in practice.

Thus, our contribution is four-fold: 1) We introduce a new notion of safety guarantee that is satisfactory for many use cases and has extremely good sample efficiency. 2) We show how to leverage the statistical inference tool of conformal prediction for robotics applications. 3) We instantiate a framework for applying conformal prediction to robotic safety. 4) We validate our framework experimentally on both a driver alert safety system and a robotic grasping system, showing that the conformal guarantees hold in practice, without issuing too many false positive alerts (e.g. less than 1% for many setups).

Organization The rest of this paper is organized as follows. In Section 2, we review conformal prediction. In Section 3, we describe our problem setup, introduce our framework, and demonstrate that specific choices for elements of our framework lead to instantiations such as tuning an ROC curve threshold to limit false negatives (though we enrich this classic method with new guarantees). We then explain the differences between the conformal prediction guarantees and the statistical learning guarantees, and discuss when our guarantees should be applied. Finally, in Sections 4 and 5, we evaluate our framework on a driver alert safety system and on a robotic grasping system.

2 Overview of Conformal Prediction

This section provides an overview of conformal prediction, the general framework that we adapt for robotics safety. It may be skipped without breaking the flow of the paper.

Consider a prediction problem where the input feature is denoted by X and the label is denoted by Z. Conformal prediction [26] is a class of methods that can produce prediction sets (i.e. a set of labels), such that the true label belongs to the predicted set with high probability. In its standard form, conformal prediction requires two components: a sequence of validation data $(X_1, Z_1), \cdots, (X_T, Z_T)$ and a non-conformity score ψ, which is any function from the input feature X and the label Z to a real number. Intuitively, the non-conformity score should measure the "unusualness" of the label Z when the input feature is X. An example non-conformity score is $\psi(X, Z) = |h(X) - Z|$ where h is some fixed

prediction function — intuitively, Z is "unusual" if the prediction function has large error.

The conformal prediction algorithm computes the non-conformity score for all samples in the validation set. Given a new test example with input feature $\hat{X}$, the conformal prediction algorithm then "tries" all possible labels z, and measures the non-conformity score $\psi(\hat{X}, z)$. A label is rejected if the computed non-conformity score is greater than $1-\epsilon$ of the non-conformity scores in the validation set. Any label that is not rejected is included in the prediction set. Intuitively, the true label is unlikely to have a non-conformity score higher than $1-\epsilon$ of validation samples; hence the true label is unlikely to get rejected.

If the training data and the new test data point $(\hat{X}, \hat{Z})$ are exchangeable, i.e. the probability of observing any permutation of $(X_1, Z_1), \cdots, (X_T, Z_T), (\hat{X}, \hat{Z})$ is equally likely, then conformal prediction has very strong validity guarantees: the true label will be within the prediction set with $1-\epsilon \pm 1/(T+1)$ probability. We note that this guarantee holds regardless of the nonconformity function ψ.

There are many extensions of conformal prediction, and the most relevant extension to our safety application is Mondrian conformal prediction [33, 32], which partitions the input data into several categories such that each data point belongs to exactly one category, and guarantees validity separately for each category. Our work is based on Mondrian conformal prediction; because we wish to limit the false negative rate in warning systems, we need class-conditional validity for samples in the "unsafe" class.

Works that apply conformal prediction to robotics settings include [5, 3, 22, 12]. [5] uses conformal prediction to predict a set of possible future motion trajectories from out of a set of 17 basis trajectories, [3] uses some ideas from conformal prediction for detecting out of distribution samples in cyber-physical systems, and [22, 12] use conformal prediction for medical diagnosis. However, these works consider very different targeted problems, while we consider the problem of warning systems and provide a general framework for using conformal prediction on a variety of robotics applications.

3 Conformal Prediction Framework for Robotics Applications

3.1 Problem Setup

We consider a model-based planning application where we have some existing simulator or model, and given the current observations (denoted by random variable X), the simulator or model predicts the future states of the system (denoted by Y) in the absence of a warning. For instance, many applications have off-the-shelf simulators: an autonomous driving software might simulate the future trajectories of all traffic participants (up to some time horizon), or an aircraft control software might forward simulate the dynamics of the aircraft. We will use the random variable Z to denote the true unknown future states of the system in the absence of a warning, e.g. the true future trajectories of traffic participants, or the true future dynamics of an aircraft.

In our setup, depending on the model or simulator available, Y could have the same type as Z (e.g. both Y and Z are random variables that represent the

future trajectories of traffic participants), or Y could have a different type from Z (e.g. Y might represent some but not all aspects about the future, such as the direction of movement for traffic participants, or the distance from collision).

Assessing Safety We assume that if we know the *true* future state of the system Z, we can assess whether it is safe or not. Specifically, there exists some **safety score** denoted by $f(Z)$; we specify some threshold (denoted by f_0), and wish to be alerted if the safety score drops below this threshold (i.e. if $f(Z) < f_0$). Most applications have natural safety scores. For instance, an autonomous driving safety score f could be the distance to or time from collision; an aircraft control safety score could be the (negative) absolute difference between the orientation of the aircraft and its ideal orientation.

In addition, we assume that the user provides a **surrogate safety score** $g : Y \mapsto \mathbb{R}$ that maps from the simulator prediction to a "safety score," where a higher score indicates "safe" and a lower score indicates "unsafe". Ideally the surrogate safety score $g(Y)$ should be highly correlated with the true safety score $f(Z)$, but technically g can be any function. None of our technical results depend on any assumptions about g; however, the choice of g affects the empirical performance in terms of false positive rate (i.e. how often our warning system issues unnecessary alerts). When Y and Z have the same type, we can simply choose $g := f$; when Y and Z have different types we need to choose g on a case-by-case basis.

Warning Function We wish to design a warning function (denoted as a function $w(Y)$) that given the simulation or model output Y, decides to issue a warning ($w(Y) = 1$) or not ($w(Y) = 0$). Note that Y depends on previous states, so the warning function implicitly depends on previous observations through Y. Formally we define "safety" as the following requirement:

Definition 1. *For some $0 < \epsilon < 1$, we say that the warning system w is ϵ-safe (with respect to Y, Z, f, and f_0) if*

$$\Pr[w(Y) = 1 \mid f(Z) < f_0] \geq 1 - \epsilon.$$

In words, whenever the true future safety score $f(Z)$ is below f_0, the warning system should issue a warning ($w(Y) = 1$) with at least $1-\epsilon$ probability. Another way to think of this is that the false negative rate is at most ϵ. The main difficulty here is that the warning function w can depend on only the simulated future Y rather than the true future Z (which is not yet observed when the warning is issued), and the simulation might not come with any performance guarantees.

A trivial warning system that always issues a warning (i.e. $w_{\text{trivial}}(Y) \equiv 1$) is always ϵ-safe for any $\epsilon > 0$. However, such a warning system is not useful. A useful warning system should issue as few warnings as possible when safe. Therefore, we should also consider its false positive rate

$$\text{FPR}(w) = \Pr[w(Y) = 1 \mid f(Z) \geq f_0].$$

The false positive rate is of lower priority for safety because issuing an unnecessary warning might only be an inconvenience, while failing to issue a warning when the situation is unsafe can lead to catastrophic outcomes. In summary, our goal is to design a warning function $w(\cdot)$ such that:

Goal: Provably achieve ϵ-safety for small ϵ (e.g. 0.02), while achieving low false positive rate (FPR).

Examples A few examples that illustrate this problem setup are as follows: (1) In a driver alert system, users may want an assurance that among the instances in which the driver is in a dangerous situation, the system will issue a warning the vast majority of the time. The safety score in this case could be the time to collision (TTC), or the nearest distance from another car. (2) In a multi-arm robot collaboration system, users may want an assurance that among the instances in which the robot arms may collide, the system will issue a warning the majority of the time. The safety score could be the nearest distance to another robot arm. (3) In a warehouse robotic box-stacking system, users may want an assurance that among the instances in which the boxes will topple, the system will issue a warning the majority of the time. The safety score could be the probability of a stable stack.

Example 3 can be thought of as ROC curve threshold tuning. If the model used is a binary classifier that predicts whether there is a stable stack, we can use the predicted probability as g. Note that in this special case, our method also tunes the threshold, but adds guarantees on the false negative rate and practical guidelines for sample complexity.

3.2 Analysis of the Trade-off Between the FNR and FPR

In this section, we analyze the fundamental trade-off between the false negative rate (FNR) and the false positive rate (FPR) of a warning system. For example, a trivial system that always issues a warning will have a 0% FNR but 100% FPR. Conversely, a system that never issues a warning will have a 0% FPR but 100% FNR. This suggests a trade-off between the achievable FNR and FPR.

Infinite validation data regime Even with infinite validation data, we may not be able to achieve both perfect FNR and perfect FPR because of inherent limitations of the safety score. For instance, at one extreme, if the safe and unsafe examples have identical safety score distributions, then there is no way to distinguish them according to Definition 1. At the other extreme, if the safe and unsafe examples have disjoint safety score distributions, then we can distinguish them perfectly (i.e. achieve 0% FPR and 0% FNR). A typical real world scenario will likely fall somewhere in between the two extremes, as illustrated in Figure 2. The key quantity is the amount of overlap between the safety score distribution for safe vs. unsafe examples, which will dictate the optimal achievable trade-off between the FPR and the FNR.

Finite validation data regime The lack of sufficient validation data is another source of error that degrades the best achievable FNR/FPR trade-off. Intuitively, because we need to *provably guarantee* the FNR for ϵ-safety, in the absence of sufficient validation data, we must be conservative and issue more warnings than necessary. For example, with zero validation data we have no choice but to issue a warning for nearly every example, leading to a high FPR. In fact, in Appendix 6.2 we show that if we have fewer than T data samples, we cannot guarantee better than $O(1/T)$ FNR without incurring an FPR of close to 1.

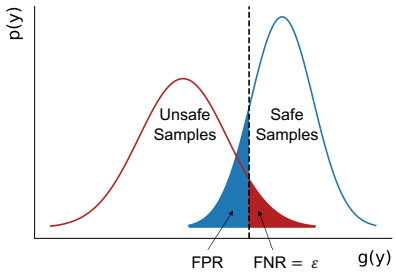


Fig. 2: Even in the limit of infinite validation data, the best false positive rate achievable (for a given ϵ-safety level) is determined by the distribution of the safe samples and the unsafe samples under the surrogate safety score function g.

Our conformal algorithm can guarantee an $O(1/T)$ FNR while the FPR is not much higher than in the infinite data regime, demonstrating the (asymptotic) optimality of the conformal algorithm presented below.

3.3 Algorithm to Achieve Guaranteed Safety Assurances

In this section, we will describe an algorithm that achieves ϵ^*-safety with a low (nontrivial) false positive rate. The setup is as described in Section 3.1, with current observations X, simulator Y, and true unknown future states Z.

If the simulation Y is perfect and has the same type as the ground truth future state Z, i.e. $Z = Y$ almost surely, then we can simply set $g = f$ and choose $w(Y) = \mathbb{I}(g(Y) < f_0)$, and this w will automatically satisfy our definition of safety. However, in most applications, it is difficult to provide any guarantees on the accuracy of the simulation. For example, in autonomous driving situations, traffic participants can behave in unexpected and hard to predict ways.

When we are uncertain about the simulation accuracy, we will require an additional training dataset. With a dataset of (simulated future state, true future state) pairs (Y_1, Z_1), $(Y_2, Z_2), \cdots, (Y_T, Z_T)$, where T is the number of samples, we can guarantee ϵ-safety. Let $(\hat{Y}, \hat{Z})$ denote a new test sample. We require only a single assumption on the dataset:

Assumption 1 *The sequence* (Y_1, Z_1), $\cdots$, (Y_T, Z_T), $(\hat{Y}, \hat{Z})$ *is exchangeable, i.e. the probability of observing any permutation of the sequence is equally likely.*

Exchangeability is a strong assumption. However, it is weaker than typical i.i.d. assumptions that underlie most machine learning methods with performance guarantees: if a sequence of data is i.i.d., then it is also exchangeable. In addition, if the distribution shifts, it is not prohibitively costly to collect a new training dataset from the shifted distribution. This is because we require only a very small dataset (e.g. in most of our experiments, the training dataset contains only about 50 examples of unsafe situations)

Based only on Assumption 1, we design an algorithm to guarantee safety on test data. The algorithm can be thought of as an instantiation of the conformal prediction framework.

Algorithm 1 Approximate ϵ-safety

1: **Input** A set of training data $(Y_1, Z_1), \cdots, (Y_T, Z_T)$, surrogate safety score g, true safety score f, and threshold f_0; a new simulation $\hat{Y}$.
2: Compute $\mathcal{A} = \{g(Y_t) \mid f(Z_t) < f_0, t = 1, \cdots, T\}$
3: Sample U uniformly in $\{0, 1, \cdots, |\{a \in \mathcal{A} \mid a = g(\hat{Y})\}|\}$
4: Compute $q = \frac{|\{a \in \mathcal{A} | a < g(\hat{Y})\}| + U + 1}{|\mathcal{A}| + 1}$
5: If $q \leq 1 - \epsilon$ then **output** 1, otherwise **output** 0

Intuitively, the procedure is as follows. We first compute a predicted safety score (based on the simulator outputs) for each unsafe sample in the training dataset (Line 2). We then sample a number from a uniform distribution (Line 3). We next compute the quantile value for the new test simulation, i.e. the proportion of validation samples with a lower safety score than $\hat{Y}$ (Line 4), with a small randomization factor from the previous step. If this quantile value is smaller than $1 - \epsilon$ (i.e. fewer than $1 - \epsilon$ of the unsafe samples from the training set have a lower safety score), we say that this may be an unsafe situation; otherwise, we say that it is safe (Line 5). The following proposition (proved in Appendix 6.1) shows that Algorithm 1 can guarantee safety.

Proposition 1 *Under Assumption 1, Algorithm 1 is* $\epsilon + 1/(1 + |\mathcal{A}|)$*-safe (with respect to* $\hat{Y}, \hat{Z}$*).*

To use Proposition 1 to provide safety guarantees, we choose ϵ based on the number of samples available $|\mathcal{A}|$. Specifically, if the desired safety level is ϵ^*, then we can choose any $\epsilon < \epsilon^*$ in Algorithm 1 such that

$$\epsilon + 1/(1 + |\mathcal{A}|) \leq \epsilon^* \tag{1}$$

In other words, if our choice of ϵ satisfies Eq. (1), then Algorithm 1 will be ϵ^*-safe. Intuitively, choosing a large ϵ decreases the false positive rate (FPR). This is because according to Algorithm 1 Line 5, choosing a larger ϵ decreases the number of times that a warning is output. Therefore, based on the number of samples in the training dataset $|\mathcal{A}|$, we choose the largest ϵ that satisfies Eq. (1) (i.e. we choose $\epsilon = \epsilon^* - 1/(1 + |\mathcal{A}|)$). We will call $1/(1 + |\mathcal{A}|)$ the discretization error.

Proposition 1 also reveals the sample complexity of the conformal prediction algorithm. If the number of unsafe examples is too small ($|\mathcal{A}| \leq 1/\epsilon^* - 1$), then we must choose $\epsilon < 0$ to ensure ϵ^*-safety according to Proposition 1. Algorithm 1 with $\epsilon < 0$ will trivially always output 1 (i.e. always issue a warning). On the other hand, if the number of unsafe examples exceeds the threshold ($|\mathcal{A}| > 1/\epsilon^* - 1$), then there will be an $\epsilon > 0$ that ensures ϵ^*-safety according to Proposition 1. Consequently, Algorithm 1 will not be trivial. In practice, we find that to get good results and a low false positive rate, it is sufficient to have sample count $|\mathcal{A}|$ that exceeds the threshold by a small margin, such as $|\mathcal{A}| = 1.5/\epsilon^* - 1$. For example, to achieve a 5% false positive rate, it is sufficient to have only about 30 unsafe examples.

Note that there are two major components of our algorithm that can be tuned to achieve a tighter FPR: the surrogate safety score, and the trained simulator or prediction model. A surrogate safety score that is better correlated with the true safety score will lead to a better FPR, as will a more accurate simulator model. The guarantees and analysis of Algorithm 1 will hold regardless of the surrogate safety score and simulator model used; but if these components are chosen poorly, not enough data is available, or the required ϵ^* is too stringent, then this procedure could become trivial (e.g. always issuing a warning). In practice however, we find that we are able to obtain good results for an autonomous driving application with an off-the-shelf prediction model for reasonably low ϵ^*-values with not too much data (see Section 4).

3.4 Comparing Conformal Prediction with PAC Learning

We further compare the statistical learning and conformal prediction guarantees. We first clarify the notation and formally define the different assumptions. Consider a sequence of training data $(Y_1, Z_1), \cdots, (Y_T, Z_T)$ and a sequence of test data $(\hat{Y}_1, \hat{Z}_1), \cdots, (\hat{Y}_N, \hat{Z}_N)$. Let $c_1, \cdots, c_M$ denote the unsafe subsequence of test data, i.e. $(\hat{Y}_{c_1}, \hat{Z}_{c_1}), \cdots, (\hat{Y}_{c_M}, \hat{Z}_{c_M})$ is the subsequence of $(\hat{Y}_1, \hat{Z}_1), \cdots, (\hat{Y}_N, \hat{Z}_N)$ such that, for all m, $f(\hat{Z}_{c_m}) < f_0$.

Two possible assumptions that we could make on the training and test data sequence are shown in Assumptions 2 and 3. In particular, marginal exchangeability (Assumption 2) is the same as Assumption 1 from the previous section. The only difference here is that we explicitly state that we only require exchangeability with *each* test data point.

Assumption 2 (marginal exchangeability) *For each n, the sequence (Y_1, Z_1), $\cdots$, (Y_T, Z_T), $(\hat{Y}_n, \hat{Z}_n)$ is exchangeable.*

Assumption 3 (i.i.d.) *The training / test data sequence $(Y_1, Z_1), \cdots, (Y_T, Z_T)$, $(\hat{Y}_1, \hat{Z}_1), \cdots, (\hat{Y}_N, \hat{Z}_N)$ is drawn from an i.i.d. distribution.*

Given a warning function, we use the random variables $\hat{F}_{c_1}, \cdots, \hat{F}_{c_M}$ to denote failure of the warning function, i.e. $\hat{F}_{c_m} = \mathbb{I}(w(\hat{Y}_{c_m}) = 0)$. Note that $\hat{F}_{c_m}$ depends on w, but we drop this dependence from our notation.

A learning algorithm is a function that takes as input the training data $(Y_1, Z_1), \cdots, (Y_T, Z_T)$ and outputs a warning function $w : X \to \{0, 1\}$. There are two main paradigms for designing learning algorithms with guarantees.

PAC Learning Under Assumption 3, a learning algorithm is (ϵ, δ)-safe if with $1 - \delta$ probability (with respect to randomness of the training data) the learned warning function w satisfies for some $\epsilon' < \epsilon$

$$\hat{F}_{c_1}, \cdots, \hat{F}_{c_M} \sim \text{Bernoulli}(\epsilon') \tag{2}$$

Conformal Learning For completeness we restate the conformal learning guarantee. A learning algorithm is ϵ-safe if the learned function w satisfies for some $\epsilon' < \epsilon$

$$\hat{F}_{c_m} \sim \text{Bernoulli}(\epsilon'), \text{ for all } m = 1, \cdots, M \tag{3}$$

Comparing Assumptions Conformal learning requires weaker assumptions. Assumption 2 is much weaker than Assumption 3, and hence is applicable to a much larger class of problems. For example, consider an autonomous driving application where the training data are snippets from randomly sampled driving scenes (no two training data points come from the same driving scene), and the test data $(\hat{Y}_1, \hat{Z}_1), \cdots, (\hat{Y}_N, \hat{Z}_N)$ is a sequence of driving snippets from a random driving scene. The test data points are not independent because they are from the same scene, and hence Assumption 3 is violated. However, Assumption 2 holds because the training data and any *single* test sample are snippets from randomly sampled driving scenes.

Comparing Sample Complexity Conformal learning requires $\Theta(1/\epsilon)$ training examples of unsafe situations (Proposition 1), while standard analysis in PAC learning requires $\Theta(1/\epsilon^2)$ examples. For example, consider the following (ϵ, δ)-safe algorithm: based on the simulation Y and the surrogate safety function g, we consider the family of warning functions $w_\theta(Y) = \mathbb{I}(g(Y) < \theta)$. Our goal is to estimate the false negative rate of w_θ (denoted by $\epsilon^*(\theta)$ for each θ and select the smallest θ such that $\epsilon^*(\theta) \leq \epsilon$.

To estimate ϵ^*, we compute the (empirical) false negative rate (denoted by $\hat{\epsilon}(\theta)$ on the training data, i.e.

$$\hat{\epsilon}(\theta) = 1/M \sum_m \mathbb{I}(g(Y_{c_m}) \geq \theta) \tag{4}$$

and use a standard concentration inequality (such as Hoeffding) to bound the difference between $\epsilon^*(\theta)$ and $\hat{\epsilon}(\theta)$. Specifically, with probability $1 - \delta$

$$\epsilon^*(\theta) \in \hat{\epsilon}(\theta) \pm \sqrt{\frac{\log(1/\delta)}{2M}} \tag{5}$$

Note that Eq. (5) is already the tightest bound possible up to constants [11]. To verify that w_θ has $\leq \epsilon$ false negative rate, we have to check that $\hat{\epsilon}(\theta) + \sqrt{\frac{\log(1/\delta)}{2M}} \leq \epsilon$, which requires

$$\sqrt{\frac{\log(1/\delta)}{2M}} \leq \epsilon \iff M \geq \frac{\log(1/\delta)}{2\epsilon^2}$$

i.e. we should have at least $\Theta(1/\epsilon^2)$ samples.

In words, even a fixed w_θ requires $\Theta(1/\epsilon^2)$ samples to verify its false negative rate according to Eq. (5). Thus, finding w_θ to provably achieve low false negative rate should require at least as many, if not more, training examples.

Comparing Usefulness of Guarantees PAC learning and conformal learning both have advantages. PAC learning has the advantage that its i.i.d. error

rate guarantee in Eq. (2) is stronger than the marginal error rate guarantee in Eq. (3). For example, if the downstream user is very sensitive to high variance (i.e. it is unacceptable for all test examples to fail simultaneously even if the probability is vanishingly small) then the i.i.d. error rate guarantee in Eq. (2) might be necessary. Nevertheless, the risk can be reduced by alternative methods such as financial tools (insurance). On the other hand, the conformal learning guarantee in Eq. (3) has the advantage that it always holds, while the PAC learning guarantee in Eq. (2) only holds with $1 - \delta$ probability.

To summarize, conformal learning requires much weaker assumptions and fewer samples, and its guarantees always hold (rather than with $1 - \delta$ probability). PAC learning offers stronger guarantees when its assumptions and sample complexity requirements are met.

4 Experiments: Driver Alert System

We empirically validate the guarantees of our framework on a driver alert safety system using real driving data. The system should warn the driver if the driver may get into an unsafe situation, without issuing too many false alarms. We show that the false negative rate (the percentage of unsafe situations that the system fails to identify) is indeed bounded according to Proposition 1, while the FPR remains low.

4.1 Experimental Setup

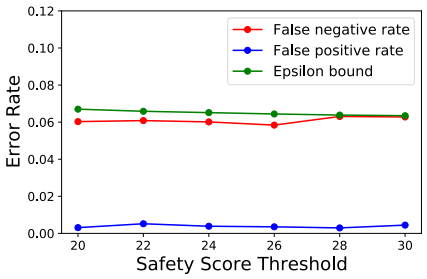
Methods We evaluate our framework on the setup described in Section 3.1. We use Trajectron++ [25] as our future dynamics model (i.e. in the notation of Section 3.1, Y is the output of Trajectron++ and $g = f$). We choose the safety score f as a weighted distance metric, where agents in the direction of the ego-vehicle velocity vector are considered "closer" than agents in the orthogonal direction. Refer to the Appendix for a more detailed explanation.

Datasets We use the nuScenes [2] and the Kaggle Lyft Motion Prediction [1] autonomous driving datasets. Each dataset contains multiple scenes, and each scene contains multiple trajectories. The trajectories in a scene are correlated with each other, but the different scenes are sufficiently distinct from each other to be considered exchangeable. To generate a dataset of exchangeable trajectories, we sample a single trajectory uniformly at random from each scene.

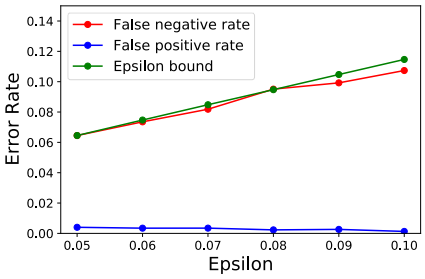
Data Splitting To compute average performance, we use random train and test splits. For both datasets, we first pool together all available data points, randomly shuffle them, and separate them back into training and test splits (with the same size as the original splits). We ran 100 trials for each experiment, and averaged over the results.

4.2 Results and Discussion

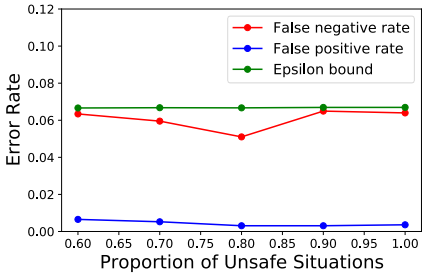
In Fig. 3a, 3b, and 3c we vary several parameters (safety threshold f_0, safety guarantee ϵ, and proportion of unsafe situations) for nuScenes. We show qualitatively similar results for the Lyft dataset in Figure 3d. Our main observations:



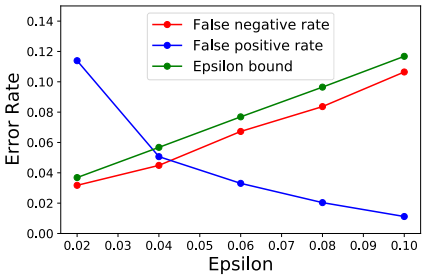
(a) False negative and false positive rates on the nuScenes dataset with varying f_0 and $\epsilon = 0.05$. The theoretical upper bound on epsilon is shown in green. The false negative rate (red) is below the upper bound and the false positive rate (blue) is very low.



(b) False negative and false positive rates on the nuScenes dataset with varying ϵ and $f_0 = 25$. The theoretical upper bound on epsilon is shown in green. The false positive rate (blue) improves with higher ϵ.



(c) False negative and false positive rates on the nuScenes dataset with varying proportions of unsafe samples in the training set. The theoretical upper bound on epsilon is shown in green. Here, $\epsilon = 0.05$ and $f_0 = 25$.



(d) False negative and false positive rates on the Kaggle Lyft dataset with a varying ϵ value. Here, $f_0 = 3.5$, and there are approx. 50 unsafe examples in the training dataset.

Fig. 3: False negative rate, false positive rate, and theoretical upper bound on epsilon for the nuScenes and Lyft datasets while varying several parameters.

1. The false negative rate (i.e. safety) is always within the theoretical bound in Proposition 1. We achieve these false negative rates with very little data. nuScenes has 50-70 unsafe examples in the training dataset, and Lyft has about 50. Yet, even with these few examples, we can ensure a false negative rate to within 1 or 2% of the desired ϵ.

2. The false positive rate (FPR) is generally very good — well below 1% on the nuScenes dataset. We use an off-the-shelf trajectory predictor trained on a small academic dataset; a more accurate trajectory predictor trained on industry-sized datasets might be expected to provide a more discriminative safety score (as in Figure 2), and thus a further improved FPR. Note that as shown in Figure 4 in the Appendix, there is a tradeoff between ϵ and the FPR when there are few (e.g. $< 1/T$) samples, which is consistent with what our theory from Section 3.2 would predict.

3. One previously unmentioned benefit of our approach is that our method is robust to label frequency shift — the frequency of unsafe situations can dif-

fer between the training data and test data. Observe that the output of Algorithm 1 depends only on the unsafe examples; consequently, the safety guarantee in Proposition 1 still holds if we increase or decrease the number of safe examples. For example, the training data collection process could intentionally focus on unsafe situations, so that unsafe examples are over-represented in the training data. We empirically simulate this in Figure 3c where we increase the proportion of unsafe examples in the training set (by deleting safe examples). The performance of our algorithm does not change qualitatively.

In the comparison of PAC learning and conformal learning, we argued that the main advantage of PAC learning is that the failures are i.i.d., so the total number of failures should have low variance (due to the Central Limit Theorem). However, we show empirically that users need not be overly concerned about highly correlated failures, as long as the test samples are not inherently highly correlated. We find that the variance on the false negative rate from different train/test splits is very low. With $\epsilon = 0.06$, for instance, it was only 0.0014, and this variance is representative among the various ϵ experiments. We show more evidence of this in the Appendix.

5 Experiments: Robotic Grasping

Finally, we validate the guarantees of our framework on a robotic grasping system that should warn the user when the robot will fail to pick and transport an object. Picking is a core problem in warehouse robotics [6, 9, 15, 34, 35, 18–20], and failures hurt throughput (potentially even stopping the assembly line). Failures can also lead to dropped or damaged goods.

We again evaluate our framework on the setup described in Section 3.1, using the Grasp Quality Convolutional Neural Network (GQ-CNN) from [18–20] as our predictor model on the DexNet 4.0 dataset of synthetic objects grasped with a parallel-jaw gripper [20]. This model classifies whether a candidate robotic grasp will be successful, and we use the probability of a successful pick as the safety score. We consider a candidate grasp "unsafe" if it will not be able to successfully pick the object (i.e. $Z = 0$). Note that this is exactly an ROC curve tuning setup. We averaged over 100 trials of Algorithm 1 with randomized train/test splits.

With $\epsilon = 0.05$, we achieved a false negative rate of 0.05, and a false positive rate of 0.11. With $\epsilon = 0.1$, we achieved a false negative rate of 0.10 and a false positive rate of 0.04. The conformal guarantees of our framework hold.

6 Conclusion

In this work, we introduce a broadly applicable framework that uses conformal prediction to tune warning systems for robotics applications. This framework allows us to achieve provable safety assurances with very little data. We demonstrate empirically that the guarantees on the false negative rate hold for a driver alert system and for a robotic grasping system. In future work, we would like to explore conformal prediction in non-exchangeable scenarios [27], conditional safety [10], and deployment in industry-scale applications. Another important

future direction is to study the impact of the predictor on the data that it is trying to predict [24].

References

1. Lyft motion prediction dataset (2020), https://www.kaggle.com/c/lyft-motion-prediction-autonomous-vehicles/data
2. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. arXiv preprint arXiv:1903.11027 (2019)
3. Cai, F., Koutsoukos, X.: Real-time out-of-distribution detection in learning-enabled cyber-physical systems. 2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS) pp. 174–183 (2020)
4. Calafiore, G., Campi, M.: The scenario approach to robust control design. IEEE Transactions on Automatic Control **51**(5), 742–753 (2006). https://doi.org/10.1109/TAC.2006.875041
5. Chen, Y., Rosolia, U., Fan, C., Ames, A., Murray, R.: Reactive motion planning with probabilistic safety guarantees. Conference on Robot Learning (2020)
6. Correll, N., Bekris, K.E., Berenson, D., Brock, O., Causo, A., Hauser, K., Okada, K., Rodriguez, A., Romano, J.M., Wurman, P.R.: Analysis and observations from the first amazon picking challenge. IEEE Transactions on Automation Science and Engineering **15**(1), 172–188 (2016)
7. Crestani, D., Godary-Dejean, K., Lapierre, L.: Enhancing fault tolerance of autonomous mobile robots. Robotics and Autonomous Systems **68**, 140–155 (2015)
8. Ding, S.X.: Model-Based Fault Diagnosis Techniques: Design Schemes, Algorithms and Tools, pp. 3–11. Springer London, London (2013)
9. Eppner, C., Höfer, S., Jonschkowski, R., Martín-Martín, R., Sieverling, A., Wall, V., Brock, O.: Lessons from the amazon picking challenge: Four aspects of building robotic systems. In: Robotics: science and systems (2016)
10. Feldman, S., Bates, S., Romano, Y.: Improving conditional coverage via orthogonal quantile regression. arXiv preprint arXiv:2106.00394 (2021)
11. Foody, G.M.: Sample size determination for image classification accuracy assessment and comparison. International Journal of Remote Sensing **30**(20), 5273–5291 (2009). https://doi.org/10.1080/01431160903130937
12. Gammerman, A., Nouretdinov, I., Burford, B., Chervonenkis, A., Vovk, V., Luo, Z.: Clinical mass spectrometry proteomic diagnosis by conformal predictors. Statistical Applications in Genetics and Molecular Biology **7** (2008)
13. Harirchi, F., Ozay, N.: Model invalidation for switched affine systems with applications to fault and anomaly detection. Analysis and Design of Hybrid Systems (ADHS) **48**(27), 260–266 (2015). https://doi.org/10.1016/j.ifacol.2015.11.185
14. Harirchi, F., Ozay, N.: Guaranteed model-based fault detection in cyber-physical systems: A model invalidation approach. arXiv preprint arXiv:1609.05921 (2017)
15. Hernandez, C., Bharatheesha, M., Ko, W., Gaiser, H., Tan, J., van Deurzen, K., de Vries, M., Van Mil, B., van Egmond, J., Burger, R., et al.: Team delft's robot winner of the amazon picking challenge 2016. In: Robot World Cup. pp. 613–624. Springer (2016)
16. Khalastchi, E., Kalech, M.: On fault detection and diagnosis in robotic systems. ACM Computing Surveys (CSUR) **51**(1), 1–24 (2018)
17. von Luxburg, U., Schölkopf, B.: Statistical learning theory: Models, concepts, and results. In: Gabbay, D.M., Hartmann, S., Woods, J. (eds.) Inductive Logic, Handbook of the History of Logic, vol. 10, pp. 651–706. North-Holland (2011). https://doi.org/10.1016/B978-0-444-52936-7.50016-1

18. Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., Ojea, J.A., Goldberg, K.: Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. Robotics: Science and Systems (RSS) (2017)
19. Mahler, J., Matl, M., Liu, X., Li, A., Gealy, D., Goldberg, K.: Dex-net 3.0: Computing robust robot suction grasp targets in point clouds using a new analytic model and deep learning. arXiv preprint arXiv:1709.06670 (2017)
20. Mahler, J., Matl, M., Satish, V., Danielczuk, M., DeRose, B., McKinley, S., Goldberg, K.: Learning ambidextrous robot grasping policies. Science Robotics **4**(26) (2019)
21. Muradore, R., Fiorini, P.: A pls-based statistical approach for fault detection and isolation of robotic manipulators. IEEE Transactions on Industrial Electronics **59**(8), 3167–3175 (2011)
22. Nouretdinov, I., Costafreda, S.G., Gammerman, A., Chervonenkis, A., Vovk, V., Vapnik, V., Fu, C.H.: Machine learning classification with confidence: Application of transductive conformal predictors to mri-based diagnostic and prognostic markers in depression. NeuroImage **56**(2), 809–813 (2011). https://doi.org/10.1016/j.neuroimage.2010.05.023
23. Patton, R., Chen, J.: Observer-based fault detection and isolation: Robustness and applications. Control Engineering Practice **5**(5), 671–682 (1997). https://doi.org/10.1016/S0967-0661(97)00049-X
24. Perdomo, J., Zrnic, T., Mendler-Dünner, C., Hardt, M.: Performative prediction. In: International Conference on Machine Learning. pp. 7599–7609. PMLR (2020)
25. Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M.: Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data (2020)
26. Shafer, G., Vovk, V.: A tutorial on conformal prediction. Journal of Machine Learning Research (JMLR) (2008), https://jmlr.csail.mit.edu/papers/volume9/shafer08a/shafer08a.pdf
27. Tibshirani, R.J., Barber, R.F., Candès, E.J., Ramdas, A.: Conformal prediction under covariate shift. arXiv preprint arXiv:1904.06019 (2019)
28. Vemuri, A.T., Polycarpou, M.M., Diakourtis, S.A.: Neural network based fault detection in robotic manipulators. IEEE Transactions on Robotics and Automation **14**(2), 342–348 (1998)
29. Visinsky, M.L., Cavallaro, J.R., Walker, I.D.: Expert system framework for fault detection and fault tolerance in robotics. Computers & electrical engineering **20**(5), 421–435 (1994)
30. Visinsky, M.L., Cavallaro, J.R., Walker, I.D.: Robotic fault detection and fault tolerance: A survey. Reliability Engineering & System Safety **46**(2), 139–158 (1994)
31. Visinsky, M.L., Cavallaro, J.R., Walker, I.D.: A dynamic fault tolerance framework for remote robots. IEEE transactions on robotics and automation **11**(4), 477–490 (1995)
32. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic learning in a random world (2005)
33. Vovk, V., Lindsay, D., Nouretdinov, I.: Mondrian confidence machine (2003)
34. Yu, K.T., Fazeli, N., Chavan-Dafle, N., Taylor, O., Donlon, E., Lankenau, G.D., Rodriguez, A.: A summary of team mit's approach to the amazon picking challenge 2015. arXiv preprint arXiv:1604.03639 (2016)
35. Zeng, A., Yu, K.T., Song, S., Suo, D., Walker, E., Rodriguez, A., Xiao, J.: Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In: 2017 IEEE international conference on robotics and automation (ICRA). pp. 1386–1383. IEEE (2017)