TU WIEN Informatics

Probabilistic Verification of Black-Box Systems

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Logic and Computation

eingereicht von

Peter Blohm, BSc.
Matrikelnummer 11905150

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing.(BA) Dr.rer.nat. Thomas Gärtner, MSc
Mitwirkung: Univ.Ass. Sagar Malhotra, MSc PhD
Univ.Ass. Patrick Indri, MSc

Wien, 25. April 2025

Peter Blohm

Thomas Gärtner

TU WIEN Informatics

Probabilistic Verification of Black-Box Systems

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Logic and Computation

by

Peter Blohm, BSc.
Registration Number 11905150

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing.(BA) Dr.rer.nat. Thomas Gärtner, MSc
Assistance: Univ.Ass. Sagar Malhotra, MSc PhD
Univ.Ass. Patrick Indri, MSc

Vienna, April 25, 2025

Peter Blohm | Thomas Gärtner

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Peter Blohm, BSc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT- Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 25. April 2025

Peter Blohm

Danksagung

Ich will zunächst meine tiefe Dankbarkeit gegenüber meinen Betreuern, Prof. Thomas Gärtner, Sagar Malhotra und Patrick Indri für die unzählbaren Stunden ihrer Arbeit ausdrücken. Ihr Rat und ihre Unterstützung meiner Ideen haben jegliche Erwartungen übertroffen.

Ich danke Sagar Malhotra im Besonderen, welcher die dieser Arbeit zugrundeliegende Idee hatte. Er war es, der mein Interesse an diesem Thema geweckt hat.

Im Verlauf dieser Arbeit haben Prof. Ezio Bartocci und Dr. Dejan Nickovic mich auf Signal-temporale Logik aufmerksam gemacht. Ich danke ihnen für ihre hilfreichen Anregungen, die mir sehr dabei geholfen haben, meine Ideen für Kapitel 4 zu formen.

Meine Freunde und meine Kollegen in RuML waren unglaubliche Motivatoren für mich. Ich möchte mich speziell bei meinen Freunden Dave, Bini und Ivana bedanken, die mich durch mein Studium begleitet und mich reichlich beim Schreibprozess unterstützt haben.

Nicht zuletzt möchte ich mich bei meiner Familie bedanken. Ich wäre nicht da wo ich jetzt bin ohne meine Mutter Renate, meinen Vater Peter und meine Tante Franziska. Die immense Unterstützung meiner Mutter in den letzten Jahren hat es mir ermöglicht mich auf mein Studium zu konzentrieren und diese Arbeit zu beenden.

Acknowledgements

I want to express my deep gratitude towards my supervisors, Prof. Thomas Gärtner, Sagar Malhotra and Patrick Indri for the countless hours of their work. Their guidance and encouragement for my ideas far exceeded all expectations.

I thank Sagar Malhotra in particular, who proposed the idea underlying this thesis. He was the person who sparked my interest in this topic.

During the course of this thesis, Prof. Ezio Bartocci and Dr. Dejan Nickovic introduced me to signal temporal logic. I thank them for their input, which significantly helped me to shape my ideas in Chapter 4.

My friends and my colleagues at RuML were some of the greatest sources of motivation for me. My special thanks go to my friends Dave, Bini and Ivana, who accompanied me through my studies and provided ample support during the writing process.

Most importantly, I thank my family. I would not be where I am without my mother Renate, my father Peter and my aunt Franziska. It was the immense support of my mother over the last years that allowed me to focus on my studies and finish this thesis.

Kurzfassung

Wir entwickeln und untersuchen eine probabilistische Prozedur zur Charakterisierung der Sicherheit von Black Box Systemen, wie neuronalen Netzwerken (NNs) oder cyber-physikalischen Systemen (CPSs). Mit einer einzigen Stichprobe von ausreichender Größe kann unsere Prozedur Spezifikationen in einer vorausgewählten Klasse identifizieren, welche das System mit hoher Wahrscheinlichkeit korrekt beschreiben.

Für viele Probleme sind Lösungen basierend auf maschinellem Lernen der Stand der Technik, doch die Komplexität dieser Systeme macht formale Beweise ihrer Sicherheit oft unmöglich. Ohne Sicherheitsgarantien kann unvorhersehbares Verhalten und Manipulationsanfälligkeit in kritischen Anwendungen nicht ausgeschlossen werden. Dies ist vor allem für CPSs wichtig, welche physisch mit ihrer Umgebung interagieren. Deswegen sind Sicherheitsgarantien für den Einsatz dieser Systeme oft eine Voraussetzung. Für Black-Box Szenarien, in denen formale Methoden nicht einsetzbar sind, untersuchen wir wie probabilistische Garantien *alleine durch Zufallstests* gegeben werden können.

Wir verwenden Methoden aus der Lerntheorie um—für eine gewählte Klasse von möglichen Spezifikationen—zu entscheiden, welche Spezifikationen das System mit großer Wahrscheinlichkeit korrekt beschreiben. Unsere Prozedur benötigt nur eine Stichprobe von ausreichender Größe um eine Aussage über die gesamte Spezifikationsklasse zu treffen. Bemerkenswerterweise ist die Größe dieser Stichprobe unabhängig von Charakteristiken des untersuchten Systems und seiner Daten und ist nur abhängig von der Komplexität der *Spezifikationsklasse* selbst, speziell ihrer Vapnik-Chervonenkis (VC) dimension.

Wir untersuchen die Anwendung unserer Methode in zwei praxisrelevanten Szenarien und erweitern unsere Theorie um Lösungen speziell für diese Probleme zu entwickeln. Zuerst untersuchen wir die Robustheit von NNs gegen gezielte Datenmanipulation und geben *wahrscheinlich annähernd-globale* Robustheitsgarantien. Diese Garantien dienen dann als scharfe untere Schranken für die Robustheit jeder Vorhersage eines NNs in Abhängigkeit der Vorhersagesicherheit. Für CPS Verifizierung selbst untersuchen wir Signal-temporale Logik (STL) als Spezifikationssprache. Erfüllt das CPS eine Bedingung für Wohlverhaltenheit, erhalten wir für jede in STL ausdrückbare Formel eine Oberschranke ihrer VC Dimension. Dadurch können wir die notwendige Stichprobengröße bestimmen, mit der jede aus den Daten gelernte Spezifikation mit hoher Wahrscheinlichkeit generalisiert.

Unsere Experimente zeigen, dass unsere Theorie in die Praxis übertragbar ist und die Stichprobengrößen auch für komplexe Spezifikationsklassen noch handhabbar sind.

Abstract

We devise and investigate a probabilistic procedure for verifying the safety of black-box systems like neural networks (NNs) and cyber-physical systems (CPSs). Given a large enough sample of observations, our procedure allows us to identify specifications in a chosen class that correctly characterise the system with high probability. In particular, we give guarantees for all specifications in the class without the need to resample.

Machine-learning-based solutions are state of the art in many problem settings, yet their internal complexity often renders it infeasible to verify them for safety formally. Without safety guarantees, these systems might behave unpredictably and are vulnerable to manipulation in critical applications. This is especially true for CPSs, which interact with their environment physically. Because of this, guaranteed safety is often a prerequisite for the deployment of CPSs. For black-box settings where formal methods are infeasible, we investigate how to give probabilistic guarantees *from random observations alone*.

We use tools from learning theory to decide—for a chosen class of candidate specifications—which specifications the investigated system will adhere to with high probability. Our procedure requires us to obtain only one sufficiently large sample of observations to make statements about the whole class of specifications. Remarkably, the required size of the sample is independent of any characteristics of the investigated system and observations and only depends on the complexity of the *specification class* itself, specifically its Vapnik-Chervonenkis (VC) dimension.

We apply our verification procedure to two practically relevant settings and extend our theory to devise solutions tailored to these specific problems. We first investigate the robustness of NNs against adversarial perturbations and give *probably, approximately global* robustness guarantees. These guarantees then serve as sharp lower bounds for the robustness of each prediction of an NN, given its prediction confidence. To tackle CPS verification, we investigate signal temporal logic (STL) as a specification language. Assuming the CPS is well-behaved, we can provide VC dimension bounds for any parametrised formula expressible in STL. This allows us to quantify how many samples are required to mine system specifications that are guaranteed to generalise.

Our experimental results show that our theory easily translates into practice and that our requirements on the sample size are manageable even for complex specification classes.

Contents

Kurzfassung xi

Abstract xiii

Contents xv

1 Introduction 1
- 1.1 Research Questions . . . 2
- 1.2 Structure of the Thesis . . . 3

2 Epsilon Nets for Verification 5
- 2.1 Preliminaries: Probabilities and Learning Theory . . . 5
- 2.2 ε-Nets . . . 11
- 2.3 Taxonomy of Sampling-Based Inference Methods . . . 13
- 2.4 A Sampling-Based Probabilistic Verification Procedure . . . 17
- 2.5 The Choice of Range Spaces and Coverage Guarantees . . . 19
- 2.6 Summary . . . 20

3 Sampling-Based Verification of Neural Network Robustness 21
- 3.1 Preliminaries: Robustness in Neural Networks . . . 21
- 3.2 Probably Approximately Global Robustness . . . 25
- 3.3 Bounds for Conjunctive Probabilities . . . 26
- 3.4 Bounds for Neural Network Confidence . . . 28
- 3.5 Sample-Based PAG Robustness Guarantees . . . 29
- 3.6 Robustness Lower-Bounds . . . 30
- 3.7 Experiments . . . 32
- 3.8 Extension to Multiple Properties . . . 37
- 3.9 Summary, Limitations, and Future Work for NN Robustness . . . 42

4 Description-Based Verification of Cyber-Physical Systems 43
- 4.1 Preliminaries: Verification in Cyber-Physical Systems . . . 44
- 4.2 PSTL Formulas as Range Spaces . . . 49
- 4.3 VC-Dimension Bounds for PSTL Formulas in General Settings . . . 51
- 4.4 Signals with Limited Variability . . . 54

4.5 VC-Dimension Bounds for PSTL Formulas in Limited Variability . . . 56
4.6 Sample-Based PSTL Validity Guarantees . . . 59
4.7 Experiments on Anomaly Detection in CPS . . . 60
4.8 Summary, Limitations, and Future Work for CPS Verification . . . 66

5 Discussion **71**
5.1 MAIN-RQ . . . 71
5.2 NN-RQ1 . . . 72
5.3 NN-RQ2 . . . 72
5.4 CPS-RQ1 . . . 72

Overview of Generative AI Tools Used **75**

Symbols **83**

Bibliography **87**

Appendix **97**
A Proofs . . . 97
B Detailed Experimental Results in NN Robustness . . . 99
C Detailed Experimental Results in CPS Verification . . . 99

CHAPTER 1

Introduction

Before the deployment of cyber-physical systems (CPSs) in real-world applications, their safety should be guaranteed. However, this is not always possible. CPSs are systems that continuously interact with their environment, and consequently often need to be certified together with the environment. Furthermore, they increasingly utilise machine learning or other complex internal mechanisms. This shared complexity of the CPS and its environment prohibits the use of formal verification techniques, which rely on exact models and are computationally expensive. In cases where formal verification is not possible, testing is usually the only alternative to show the safety of a system. Testing, however, raises two additional problems. First, *how* to test a system, and second, *for how long*.

For randomised testing, there exists a variety of statistical hypothesis tests to answer these questions. However, the choice, application and interpretation of these tools is not always easy. In CPS settings specifically, naive modelling easily leads to an astronomical amount of required tests [Kalra and Paddock, 2016]. In the absence of statistical methods to quantify the number of required tests, tools that infer statements from complex CPS data do not quantify the uncertainty of their results [Jones et al., 2014, Jha et al., 2017]. This issue is even more pronounced when multiple properties should be tested or learned at the same time, where naive statistical tools would impose compounding requirements on sample size.

In this thesis, we aim to improve this situation and devise statistical tools to characterise the required number of tests to guarantee safety up to a chosen level of uncertainty. We tackle this problem with the use of existing methods from learning theory to devise a sampling-based verification procedure for a predefined *class of hypotheses*. After obtaining a sample of sufficient size, we can certify that all hypotheses that are consistent with all sampled data points will hold for future observations with high probability. Our tools are devised in an abstract setting to allow us to distinguish our method from existing approaches and build necessary intuition. Before approaching CPS directly, we target a

more manageable problem: verification of neural nets. Neural networks suffer the same problems as CPS with respect to formal methods, but have seen much more focused research efforts for the verification of specific properties. Among these properties, we will focus on *robustness*, the resilience of a network against small, adversarial changes to its inputs.

Without special care, NNs used for classification tasks can often be tricked into producing arbitrary classifications when a given input is changed carefully. These adversarial changes are normally imperceptible to the human observer and pose a serious vulnerability [Szegedy et al., 2014, Goodfellow et al., 2015]. Formal verification of robustness against these attacks is often not feasible for NNs due to computational costs. We apply our probabilistic verification procedure to this problem to give robustness guarantees that are conditioned on the prediction confidence. The special nature of NN robustness requires us to extend our theory to provide probabilistic statements that are faithful relaxations of the formal property. We experimentally demonstrate how well our guarantees characterise the behaviour of the network on unseen data and test it with a variety of tools for the quantification of NN robustness.

After the special case of NN robustness, we approach the more general setting of CPS verification by focusing specifically on the specification language signal temporal logic (STL). As our method relies only on the complexity of the *specification class* we want to use to characterise a system, we investigate methods to bound this complexity. The focus here is to augment our certification method to allow us to certify CPS with respect to any class of specifications expressible in STL. This then allows us to quantify the certainty of any preexisting method that mines STL specifications from a sample.

1.1 Research Questions

Our research questions are phrased to focus on the different settings we investigate throughout this thesis. We aim to find **high probability** answers to the following questions.

MAIN-RQ How many random tests are required to characterise a given black-box system with respect to *a given class of properties*?

NN-RQ1 How many samples do we need to decide for all levels of confidence and robustness whether the neural network can be *both confident* and *non-robust*?

NN-RQ2 How can we obtain sharp lower bounds for NN robustness, conditioned on the prediction confidence, that generalise to unseen data?

CPS-RQ1 For *any given STL specification template*, how many simulation traces of real-time CPSs do we need to certify that all specifications we can mine are valid?

1.2 Structure of the Thesis

This thesis is structured in three self-contained chapters and a concluding chapter for discussions. In Chapter 2, we give the necessary background in learning theory, introduce our theory and set it into context with other statistical tools. The concepts introduced in this chapter are central for our results in later chapters and serve as blueprints for our later theorems.

Chapter 3 investigates NN robustness verification. We first formally define local robustness in NNs and give an overview of existing work in statistical and formal verification of NNs. We then introduce our own probabilistic definition of global robustness and give a verification procedure that expands on our theory in Chapter 2. Our experimental evaluation shows that our procedure translates well into practice, scales well with NN size and can be adapted easily to different notions of robustness.

Finally, in Chapter 4 we apply our theory to statistical model checking of CPSs. In this more general setting, we investigate the specification language STL. Our aim is to obtain VC dimension bounds for parametrised specifications in this language. Once we have these VC dimension bounds, we apply our theory to guarantee that mined specifications will hold for future observations. We demonstrate the practicality of our results with experiments in anomaly detection.

Finally, in Chapter 5 we conclude this thesis with a short summary of our results in the context of our initial research question.

CHAPTER 2

Epsilon Nets for Verification

In this chapter, we present our theoretical findings for probabilistic verification in an abstract setting. We introduce the necessary background knowledge in probability theory and learning theory to both understand our main results and contrast them with other statistical tools that might be used for similar purposes. The setting we assume serves as an abstraction of our two applications in NN verification and CPS verification, and allows us to keep the notation light, for now.

We investigate a given black-box system we call f, which is only accessible to us by sampling random observations. These observations follow a fixed but unknown probability distribution. We are then given a class of candidate specifications $\mathcal{R}$ and are tasked with identifying which of the specifications in $\mathcal{R}$ describe f. That is, we want to identify specifications that will be true for any *future* random observation with high probability.

The procedure we present to achieve this utilises ε-nets, a concept from computational geometry that gives a notion of *coverage* in probabilistic settings. We require only a single random sample of sufficient size to achieve coverage of all specifications in $\mathcal{R}$ likely to be violated. All the remaining specifications can be certified to be true in future observations with high probability.

2.1 Preliminaries: Probabilities and Learning Theory

The notation and definitions for concepts we present in this section follow the presentation in Mitzenmacher and Upfal [2017] where possible. We are given a system f from which we can sample observations. Each observation is a point $\mathbf{x} \in \mathcal{X}$, in some metric space $\mathcal{X}$. The observations are sampled independently identically distributed (i.i.d.) according to the *unknown, but fixed* distribution $\mathcal{D}$ over $\mathcal{X}$. Probabilities will be defined with respect to $\mathcal{D}$ and be denoted Pr. We use X as a random variable for an observation following $\mathcal{D}$, i.e., $X \sim \mathcal{D}$. Sets of $n \in \mathbb{N}$ data points sampled from $\mathcal{D}$ will be denoted $N \sim \mathcal{D}^n$.

In general, we reserve lower-case bold font letters for specific data points and reserve upper-case letters for random variables and sets.

We introduce further notation along with the corresponding concepts later in this section.

With this notation introduced, we start to investigate random events in $\mathcal{D}$. The random events of interest for us might be the safety specifications of the system f, as illustrated in the following example.

Example 2.1.1 (Two-State System)**.** In the simplest case, our observations of a given system f just consist of the information if it is in a safe state (0) or an unsafe state (1) with $\mathcal{X} = \{0, 1\}$. We do not know the distribution $\mathcal{D}$ but want to know how likely the system is in an unsafe state, i.e. $\Pr(X = 1)$. We can estimate this probability by taking some sample $N \sim \mathcal{D}^n$ of size $n \in \mathbb{N}$, as

$$\Pr(X = 1) \approx \frac{|\{\mathbf{x} \in N : \mathbf{x} = 1\}|}{n}. \tag{2.1}$$

In words, the fraction of successes in a sample of Bernoulli trials estimates the true success probability.

As n increases in Example 2.1.1, the estimate will approach the true probability. In general, however, we are not interested in merely estimating but rather *upper-bounding* the probability of being in an unsafe state, as a guarantee. Bounds for the deviation from sample estimates are commonly achieved with *concentration-inequalities*, especially Chernoff and Hoeffding bounds [Mitzenmacher and Upfal, 2017, chapter 4]. Out of these inequalities, the Chernoff bound for the sum of Bernoulli trials (or Poisson trials, where the success probability might differ between trials) is especially prominent in various statistical verification methods we discuss in later chapters.

Theorem 2.1.2 (Chernoff Bound, Mitzenmacher and Upfal 2017)**.** *Let $N = \{X_1, \ldots, X_n\}$ be a set of i.i.d. Bernoulli trials such that $Pr(X_i = 1) = p$. Let S be the random variable of the sum of trials, $S = \sum_{i=1}^{n} X_i$ with $\mathbb{E}[S] = np$. Then for a deviation from the expectation $0 < \varepsilon < 1$, the following two bounds hold:*

$$\Pr\left(\frac{np - S}{np} \geq \varepsilon\right) \leq \exp\left(\frac{-np\varepsilon^2}{2}\right) \tag{2.2}$$

$$\Pr\left(\frac{|np - S|}{np} \geq \varepsilon\right) \leq 2\exp\left(\frac{-np\varepsilon^2}{3}\right) \tag{2.3}$$

In words, the probability that S deviates from its expected value by more than an ε-fraction decreases exponentially with the size of the sample.

This is a powerful and well-known result, which allows us to give the desired bound for our estimate in Example 2.1.1.

Example 2.1.3 (Sample Complexity for Two-State System)**.** Inspired by Mitzenmacher and Upfal [2017, Section 4.2.3], we continue Example 2.1.1 and want to guarantee the quality of our estimate by providing a $1-\delta$ confidence interval for the true probability $p = \Pr(X = 1)$ of width 2ε based on a random sample $N \sim D^n$ and the estimate $\hat{p} = S/n$. That is, we want to find the appropriate (smallest) sample size $|N| = n$, such that for any given choice of $0 < \varepsilon, \delta < 1$

$$Pr(|p - \hat{p}| > \varepsilon) < \delta. \tag{2.4}$$

In words, the probability that our estimate is more than ε-bad is bounded by δ. We proceed with Theorem 2.1.2.

$$\begin{aligned}
\Pr(|p-\hat{p}| > \varepsilon) &= \Pr\left(\frac{|np - S|}{n} > \varepsilon\right) && (2.5)\\
&= \Pr\left(\frac{|np - S|}{np} > \frac{\varepsilon}{p}\right) && (2.6)\\
&\leq \Pr\left(\frac{|np - S|}{np} \geq \frac{\varepsilon}{p}\right) && (2.7)\\
&\leq 2\exp\left(\frac{-np(\frac{\varepsilon}{p})^2}{3}\right) && (2.8)\\
&\leq 2\exp\left(\frac{-n\varepsilon^2}{3p}\right) < \delta && (2.9)
\end{aligned}$$

We do not know the exact value of p, but we know the expression is maximal for large p. We use $p \leq 1$ and find a lower bound for n with some standard calculations.

$$\begin{aligned}
\frac{-n\varepsilon^2}{3} &< \ln\frac{\delta}{2} && (2.10)\\
-n &< \frac{3}{\varepsilon^2}\ln\frac{\delta}{2} && (2.11)\\
n &> \frac{3}{\varepsilon^2}\ln\frac{2}{\delta} && (2.12)
\end{aligned}$$

For any choice of parameters ε, δ, we now know how large our sample needs to be to give a confidence interval with the required width 2ε and probability mass δ.

For the purpose of verification, or more precisely estimation with deviation bounds, no matter what property or specification is investigated, the required sample size is in $\mathcal{O}\left(\frac{1}{\varepsilon^2}\ln\frac{1}{\delta}\right)$. If we obtain a sample of size Equation (2.12), we know the true probability p differs from our estimate at most ε, in a $1-\delta$ fraction of the samples we produce. Even though this is very useful for many use cases, there is one essential drawback to this use of Chernoff bounds. This method not only assumes no information about the system but also about the *specification* of interest. If we are interested in multiple specifications, we need to sample again and again to apply Chernoff bounds as described in Example 2.1.3.

This approach is overly naive for many settings, and other tools are better fitted to investigate groups of more complex specifications in these situations.

We illustrate this with a final abstract example, the issue of providing real-valued quantile bounds in a distribution-free setting.

Example 2.1.4. We observe pressure levels from a system and want to certify that the pressure stays below some critical threshold $t \in \mathbb{R}$. That is, $\mathcal{X} = \mathbb{R}$ and our safety property is $X \leq t$. Assume we model this property as boolean predicate with $t_1 = 5$, and after an i.i.d. sample N of size $n > \frac{3}{\varepsilon^2} \ln\left(\frac{2}{\delta}\right)$, we can give an ε, δ confidence interval for the probability of the pressure exceeding t_1, that is $p_{t_1} = \Pr(X > t_1)$.

Now assume we are also interested in $p_{t_2} = \Pr(X > t_2)$ for $t_2 = 6$. Naively, the Chernoff bound gives us no information about $\Pr(X > t_2)$. If we wish to re-use N for the certification of this second property, this is another, potentially unrelated hypothesis to be tested. The probability of *both* $\hat{p}_{t_1}$ *and* $\hat{p}_{t_2}$ being ε-good estimates of the true probabilities is then not $1 - \delta$, but $1 - 2\delta$, by the union bound. Similarly, the more hypotheses we want to test this way, the weaker the guarantees inherently become, so Chernoff bounds alone might not be best suited for this task.

There are more powerful statistical tools that can improve this situation. For Example 2.1.4, the Dvoretzky-Kiefer-Wolfowitz-Massert (DKW) inequality [Dvoretzky et al., 1985, Massart, 1990, Naaman, 2021] helps us to obtain a sample complexity to bound the *worst-case* deviation from the true distribution function in a sample.

Definition 2.1.5 (Multivariate DKW inequality, Naaman 2021)**.** For a $k \in \mathbb{N}$ variate *continuous* cumulative distribution function (cdf) F and an empirical cdf F_n estimated from n i.i.d. samples, it holds that

$$\Pr\left(\sup_{\mathbf{a} \in \mathbb{R}^k} |F_n(\mathbf{a}) - F(\mathbf{a})| > \varepsilon\right) \leq k(n+1)e^{-2n\varepsilon^2}. \tag{2.13}$$

Adapted to our question in the univariate case in Example 2.1.4, it is defined [Massart, 1990] as

$$\Pr\left(\sup_{t \in \mathbb{R}} |p_t - \hat{p}_t| > \varepsilon\right) < 2\exp(-2n\varepsilon^2), \tag{2.14}$$

resulting in a sample complexity of $\mathcal{O}\left(\frac{1}{\varepsilon^2} \ln \frac{1}{\delta}\right)$, similar to Chernoff bounds with worse constants. While the DKW inequality can overcome the issue of multiple hypothesis testing Chernoff bounds have, we discuss in this thesis that the dependence on $1/\varepsilon^2$ can be drastically improved upon, if we move from estimation to testing for low likelihood. We will show that this strategy allows us to make more flexible statements with some additional background from learning theory.

2.1.1 Concepts from Learning Theory

In addition to the introduced statistical concepts for sample-based estimation of parameters, we also heavily rely on computational learning theory. One of the central concepts we use, the Vapnik-Chervonenkis (VC) dimension [Vapnik and Chervonenkis, 2015], allows us to quantify how expressive a class of specifications of interest is. For this, we first formally define what we mean by a class of specifications.

Definition 2.1.6 (Range space)**.** Let $\mathcal{X}$ be a (possibly infinite) set and $\mathcal{R}$ a set of subsets of $\mathcal{X}$ called *ranges*, that is $\forall R \in \mathcal{R} : R \subset \mathcal{X}$. The tuple $(\mathcal{X}, \mathcal{R})$ is then called a *range space* (or *hypothesis space* or *set system*).

This definition of a given range $R \in \mathcal{R}$ is analogous to the set-semantic definition of (unary) predicates in first-order logic, where we define the specification or property R as the set of elements in $\mathcal{X}$ that satisfy R. The VC dimension then defines the combined expressivity of a given range space $(\mathcal{X}, \mathcal{R})$.

Definition 2.1.7 (Vapnik-Chervonenkis Dimension)**.** Let $(\mathcal{X}, \mathcal{R})$ be a range space. The VC dimension $\mathrm{VC}(\mathcal{X}, \mathcal{R})$ is then defined as the size of the largest set $S \subset \mathcal{X}$ that can be *shattered* by $(\mathcal{X}, \mathcal{R})$, which means that

$$\forall S' \subseteq S, \exists R \in \mathcal{R} : S \cap R = S'. \tag{2.15}$$

If $(\mathcal{X}, \mathcal{R})$ can shatter sets of arbitrary size, $\mathrm{VC}(\mathcal{X}, \mathcal{R})$ is unbounded.

VC dimensions are a central concept in learning theory and have been studied extensively. They intuitively capture how expressive or flexible a range space is, for example, the range space that is expressed by a given machine learning algorithm. In the later sections of this thesis, we study fragments of certain logics as range spaces and investigate their VC dimension as a basis for our results. We provide a demonstration of a typical VC dimension proof for a range space, which will serve as a template for VC dimension proofs later.

Example 2.1.8 (VC dimension of axis-aligned halfspaces)**.** Consider the range space $(\mathbb{R}^k, \mathcal{R}_{\leq})$ for $k \in \mathbb{N}_+$, with $\mathcal{R}_{\leq} = \{R_{\leq \mathbf{t}} : \mathbf{t} \in \mathbb{R}^k\}$ where we define each range $R_{\mathbf{t}}$ as

$$R_{\leq \mathbf{t}} = \left\{ \mathbf{x} \in \mathbb{R}^k : \bigwedge_{i=1}^{k} \mathbf{x}_i \leq \mathbf{t}_i \right\}. \tag{2.16}$$

In words, $R_{\leq \mathbf{t}}$ is defined by a set of k thresholds, and contains all points $\mathbf{x} \in \mathbb{R}^k$ which exceed none of these threshold values. This generalises the setting in Example 2.1.4.

Lemma 2.1.9. *For* $(\mathbb{R}^k, \mathcal{R}_{\leq})$ *it holds that* $\mathrm{VC}(\mathbb{R}^k, \mathcal{R}_{\leq}) = k$.

Proof. We first show $\mathrm{VC}(\mathbb{R}^k, \mathcal{R}_{\leq}) \geq k$ by example. Consider the set $S = \{\mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(k)}\} \subset \mathbb{R}^k$ of canonical basis vectors, i.e., for $i \in [1,k] : \mathbf{e}_i^{(i)} = 1$ and for $\forall j \neq i : \mathbf{e}_j^{(i)} = 0$. For any subset $S' \subset S$, we define $\mathbf{t} = \sum_{\mathbf{e} \in S'} \mathbf{e}$. Then $\forall \mathbf{e} \in S' : \mathbf{e} \in R_{\mathbf{t}}$. Further, $\forall \mathbf{e}^{(i)} \notin S' : \mathbf{t}_i = 0$, so $\forall \mathbf{e}^{(i)} \notin S' : \mathbf{e}^{(i)} \notin R_{\mathbf{t}}$. Consequently, $(\mathbb{R}^k, \mathcal{R}_{\leq})$ shatters S, and we have shown the lower bound.

Now we show $\mathrm{VC}(\mathbb{R}^k, \mathcal{R}_{\leq}) \leq k$ by contradiction. Assume $\exists S$ with $|S| = k+1$, such that $(\mathbb{R}^k, \mathcal{R}_{\leq})$ shatters S. As S is shattered, for each $S' \subseteq S$ there exists a $R_{\mathbf{t}}$ such that $R_{\mathbf{t}} \cap S = S'$. We now consider

$$S' = \left\{ \mathbf{x} \in \arg\max_{\mathbf{x}' \in S} \mathbf{x}'_i : i \in [1,k] \right\}, \tag{2.17}$$

a subset of S of points that are maximal in one dimension. We know $|S'| \leq k$ and for any range $R_{\mathbf{t}}$ such that $S \subset R_{\mathbf{t}}$, $\mathbf{t}_i \geq \max_{\mathbf{x} \in S} \mathbf{x}_i$. This, however, implies $S \setminus S' \subset R_{\mathbf{t}}$, a contradiction to $S \cap R_{\mathbf{t}} = S'$. Any set S with $|S| = k+1$ cannot be shattered. □

For a given range space and under a fixed probability distribution $\mathcal{D}$, we can now introduce a notion of coverage central to our approach. This construct from computational geometry is called *ε-net.*

Definition 2.1.10 (ε-net Haussler and Welzl 1986, Mitzenmacher and Upfal 2017)**.** Let $(\mathcal{X}, \mathcal{R})$ be a range space, $\mathcal{D}$ be a distribution over $\mathcal{X}$ and X be a random observation sampled from $\mathcal{D}$. A finite set $N \subset \mathcal{X}$ is called *ε-net*, if and only if

$$\forall R \in \mathcal{R} : \Big(\Pr(X \in R) \geq \varepsilon \implies N \cap R \neq \emptyset \Big). \tag{2.18}$$

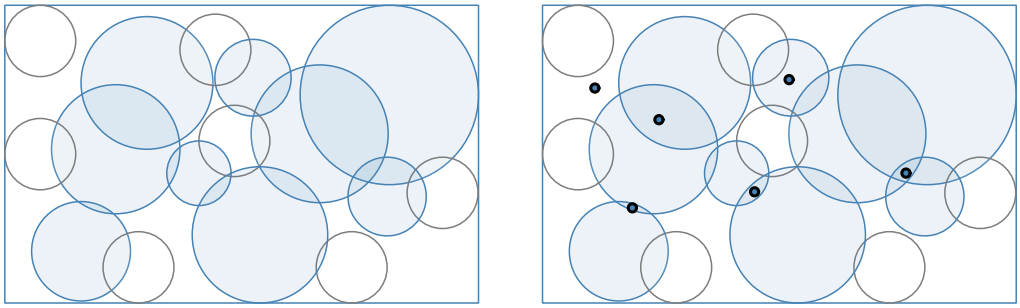
In words, N is an ε-net if and only if it intersects all the ε-likely ranges in $\mathcal{R}$.

Section 2.1.1 illustrates an ε-net for a range space of circles in $\mathbb{R}^2$ with some distribution. The ε-net just ensures that all probably enough ranges will be *intersected at least once.* In contrast to, e.g. Chernoff bounds, we cannot know if the size of the intersection between the ε-net and any particular range is representative of the true probability. A finite set that satisfies this additional requirement of representative intersection sizes is called *ε-sample.*

Definition 2.1.11 (ε-sample Vapnik and Chervonenkis 2015, Mitzenmacher and Upfal 2017)**.** Let $(\mathcal{X}, \mathcal{R})$ be a range space, $\mathcal{D}$ be a distribution over $\mathcal{X}$ and X be a random observation sampled from $\mathcal{D}$. A finite set $N \subset \mathcal{X}$ is called *ε-sample*, if and only if

$$\forall R \in \mathcal{R} : |\Pr(X \in R) - \frac{|N \cap R|}{|N|}| \leq \varepsilon \tag{2.19}$$

In words, N is an ε-sample if and only if it estimates the true probabilities of all ranges in $\mathcal{R}$ with an error of at most ε.



(a) A couple of circles (b) An ε-net

Figure 2.1: Example for an ε-net. Figure 2.1a illustrates a range space. Circles with probabilities larger than or equal to ε are tinted blue. The set of points in Figure 2.1b intersects all blue circles.

As a final result, we present a very helpful result for VC dimension bounds that we will use in later chapters. Goldberg and Jerrum [1993] investigated *parametrised* range spaces, where each range is definable with real-valued parameters and membership tests are fixed boolean formulas over polynomial inequalities.

Theorem 2.1.12 (VC dimension of Parametrised Range Spaces, Goldberg and Jerrum 1993)**.** *Let $(\mathcal{X}_n, \mathcal{R}_k)$ be a range space, where elements $\mathbf{x} \in \mathcal{X}_n$ are representable by n real values, and each range $R \in \mathcal{R}_k$ is representable by k real values. Suppose all membership tests $\mathbf{x} \in R$ can be expressed as a fixed boolean formula $\Phi_{k,n}$. The formula $\Phi_{k,n}$ is built over $\eta = \eta(k,n)$* distinct *atomic predicates, each predicate being a polynomial inequality over $k+n$ variables, of degree at most $\ell = \ell(k,n)$. Then, for the VC dimension of $(\mathcal{X}_n, \mathcal{R}_k)$ it holds that*

$$VC(\mathcal{X}_n, \mathcal{R}_k) \leq 2k \log_2(8e\ell\eta) \tag{2.20}$$

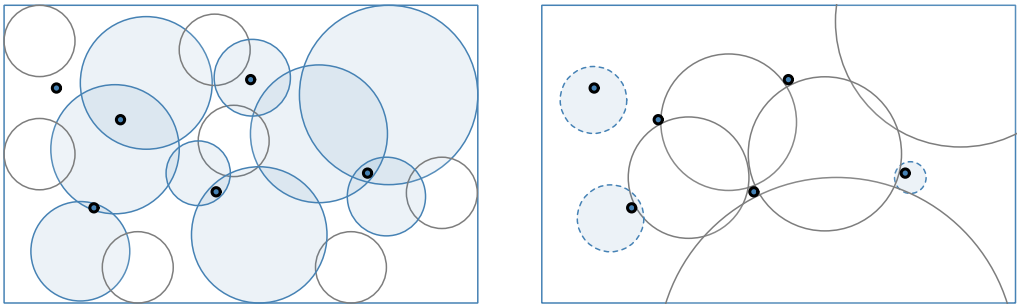
That is, the VC dimension scales linearly with the number k of parameters for each range, logarithmically with the degree of polynomials ℓ and the number η of distinct *inequalities, but is independent of the dimensionality of $\mathcal{X}$.*

2.2 ε-Nets

In this section, we use the tools we introduced in Section 2.1 to present and motivate our approach to probabilistic verification. An ε-net, by definition, intersects any probable enough range in a range space. If our aim is to identify *many* high-probability specifications in $\mathcal{R}$, we can do so with a single ε-net.

Observation 2.2.1. *Given an ε-net N for a range space $(\mathcal{X}, \mathcal{R})$, we can identify low-probability ranges. Definition 2.1.10 with contraposition gives*

$$\forall R \in \mathcal{R} : \Big(N \cap R = \emptyset \implies \Pr(X \in R) < \varepsilon\Big) \tag{2.21}$$



(a) An ε-net for circles in $\mathbb{R}^2$ (b) All empty circles have low probability mass

Figure 2.2: Example for probabilistic inference with an ε-net. Figure 2.2a illustrates a range space with an ε-net. Circles with probabilities larger than or equal to ε are tinted blue. Figure 2.2b shows a different set of circles. With the ε-net, we can infer that all empty circles have a probability mass smaller ε. The dashed blue circles intersect the ε-net, so no statement about them can be made.

We can use this observation to identify high probability ranges *under the complement*. Let R^c denote the complement of the range R, i.e., $R^c = \mathcal{Q} \setminus R$ and analogously $\mathcal{R}^c = \{R^c : R \in \mathcal{R}\}$. as $\Pr(X \in R) = 1 - \Pr(X \in R^c)$, we can use ε-nets to identify specifications that hold with probabilities of at least $1 - \varepsilon$. More specifically, these high-probability specifications are nearly never violated in the investigated system. Remarkably, similar to the DKW-inequality, ε-nets allow us to check all ranges in $\mathcal{R}^c$ without weakening our statement. In order to avoid confusing language, we utilise this argument of reasoning under the complement implicitly going forward. We provide an example of the utility of our observation in a simple range space.

Example 2.2.2 (ε-net over circles)**.** We again consider the range space $(\mathbb{R}^2, \mathcal{R})$ with each $R \in \mathcal{R}$ being a set of points contained in a circle. We now assume we have an ε-net N, illustrated in Example 2.2.2. Any circle we can construct without intersecting N, can be certified to contain a probability mass smaller ε, as shown in Figure 2.2b. If a given circle intersects any point in N, we cannot make a statement about its probability mass.

While we often cannot construct ε-nets deterministically, they can be obtained via i.i.d. samples for range spaces with bounded VC dimension, as stated by the following theorem.

Theorem 2.2.3 (ε-nets from i.i.d. samples, Mitzenmacher and Upfal 2017)**.** *Let $(\mathcal{X}, \mathcal{R})$ be a range space with VC dimension d and let $\mathcal{D}$ be a probability distribution over $\mathcal{X}$. For parameters $0 < \varepsilon, \delta < \frac{1}{2}$, an i.i.d. sample from $\mathcal{D}$ of size s is an ε-net for $(\mathcal{X}, \mathcal{R})$ with probability at least $1 - \delta$ for some*

$$s = \mathcal{O}\left(\frac{d}{\varepsilon} \ln \frac{d}{\varepsilon} + \frac{1}{\varepsilon} \ln \frac{1}{\delta}\right) \tag{2.22}$$

This is a well-known result in learning theory and computational geometry, where the main interest is describing asymptotic behaviour. We, however, want to sample small

ε-nets for the purpose of verification and are interested in obtaining the tightest possible sample complexity. The following proposition reconstructs the proof of Mitzenmacher and Upfal [2017, Theorem 14.8] for Theorem 2.2.3 but more carefully keeps track of constants, in order to obtain a finite expression for the sample complexity of ε-nets.

Proposition 2.2.4 (ε-nets from i.i.d. samples)**.** *Let $(\mathcal{Q}, \mathcal{R})$ be a range space with VC dimension d and let $\mathcal{D}$ be a probability distribution over $\mathcal{Q}$. For parameters $0 < \varepsilon, \delta < \frac{1}{2}$, an i.i.d. sample from $\mathcal{D}$ of size s is an ε-net for $(\mathcal{Q}, \mathcal{R})$ with probability at least $1 - \delta$ if s satisfies*

$$s \geq \frac{2}{\ln(2)\varepsilon}\left(\ln\frac{1}{\delta} + d\ln(2s) - \ln\left(1 - \exp\left(\frac{-s\varepsilon}{8}\right)\right)\right) \tag{2.23}$$

Proof. See Appendix A. □

Careful inspection shows that Proposition 2.2.4 is *worse* than one of the inequalities in the final lines of the proof in Mitzenmacher and Upfal [2017, Theorem 14.8]. This discrepancy is due to an omitted constant factor of $\ln(2)$ in their derivation of the proof, inconsequential to their result. With Proposition 2.2.4 we are now able to obtain a precise integer that is guaranteed to be a sufficient sample size. We denote this integer with the following expression going forward and use a simple numerical method like binary search to find the smallest suitable integer.

$$s(\varepsilon, \delta, d) = \min\left\{s' \in \mathbb{N} : s' \geq \frac{2}{\ln(2)\varepsilon}\left(\ln\frac{1}{\delta} + d\ln(2s') - \ln\left(1 - \exp\left(\frac{-s'\varepsilon}{8}\right)\right)\right)\right\} \tag{2.24}$$

In the spirit of Example 2.2.2, we can now identify certifiably low probability ranges in R with a single sample. The following section describes our verification procedure and discusses differences between the use of ε-nets and other, more common tools like Chernoff bounds for verification in detail.

In the following, we investigate how exactly inference based on ε-nets differs from other sampling-based methods.

2.3 Taxonomy of Sampling-Based Inference Methods

As we choose ε-nets for our approach to probabilistic verification, it is important to motivate this choice in contrast to other, more established constructs used in related work. In this section, we highlight the differences between different mechanisms to achieve a distribution-agnostic bound with a small taxonomy. For the sake of completeness of this taxonomy, we will introduce a construct we call *binomial tail bound*. This is an elementary result, obtained with basic probability theory and introduced here purely to paint a more complete picture.

Lemma 2.3.1 (Binomial Tail Bound)**.** *Let X be a real-valued random variable, following some unknown distribution $\mathcal{D}$ over $\mathbb{R}$. Then, for an i.i.d. random sample $N \sim \mathcal{D}^s$ and parameters $0 < \varepsilon, \delta < \frac{1}{2}$, if s satisfies*

$$s \geq \frac{\ln \frac{1}{\delta}}{\ln(1-\varepsilon)} \tag{2.25}$$

then for a given N it holds with probability of at least $1-\delta$ that

$$\Pr(X > \max(N)) < \varepsilon \tag{2.26}$$

Proof. Consider the (unknown) $1-\varepsilon$ quantile $X_{1-\varepsilon}$ of X, that is $X_{1-\varepsilon} = \inf_{\mathbf{x}} : \Pr(X \leq \mathbf{x}) \geq 1-\varepsilon$. Per definition of quantiles $\Pr(X > X_{1-\varepsilon}) < \varepsilon$, but also $\Pr(X < X_{1-\varepsilon}) \leq 1-\varepsilon$. Now, if $\max(N) \geq X_{1-\varepsilon}$, it holds that $\Pr(X > \max(N)) \leq \Pr(X > X_{1-\varepsilon}) < \varepsilon$. We can now obtain a bound for

$$\begin{aligned}
\Pr(\max(N) < X_{1-\varepsilon}) &= \Pr(\forall \mathbf{x} \in N : \mathbf{x} < X_{1-\varepsilon}) && (2.27)\\
&= \Pr(X < X_{1-\varepsilon})^s && (2.28)\\
&\leq (1-\varepsilon)^s \leq \delta && (2.29)\\
s &\geq \frac{\ln \frac{1}{\delta}}{\ln(1-\varepsilon)} && (2.30)
\end{aligned}$$

□

Lemma 2.3.1 allows us to use the maximum value of our sample (or any value higher than the maximum) as an upper bound for a high quantile in the distribution. Remarkably, it allows us to make the same statement Chernoff bounds give, but only for the special case where the observed probability of a given event is 0. An important observation here is that $\ln \frac{1}{\delta} / \ln(1-\varepsilon) = \mathcal{O}\left(\frac{1}{\varepsilon} \ln \frac{1}{\delta}\right)$. We can now suddenly see how different mechanisms of estimating probabilities relate in terms of their sampling complexity, as illustrated in Figure 2.3. This figure is reductive by design but illustrates an important insight into the costs in terms of sample size we have to pay for different types of information. It is comparatively cheap to check if a probability $p < \varepsilon$, but to estimate any value of p with an error of ε requires a number of samples quadratic in $\frac{1}{\varepsilon}$. Orthogonally to this, if we want to learn the parameter p of a Bernoulli trial, our sample is much smaller than for more complex range spaces. We need to compensate for range spaces with larger VC dimensions d with a factor of $d \ln \frac{1}{\varepsilon}$. The trick used in many settings where Chernoff bounds are applied is that testing for any boolean property can be modelled as a series of Bernoulli trials. We do not need to think about the property in question; we just need to check the fraction of trials in which the property is true to get a good estimate of the true probability. This, however, means we have to pay the cost factor of $\frac{1}{\varepsilon}$, and we cannot easily test multiple complex properties at once. Even when we—instead of Chernoff

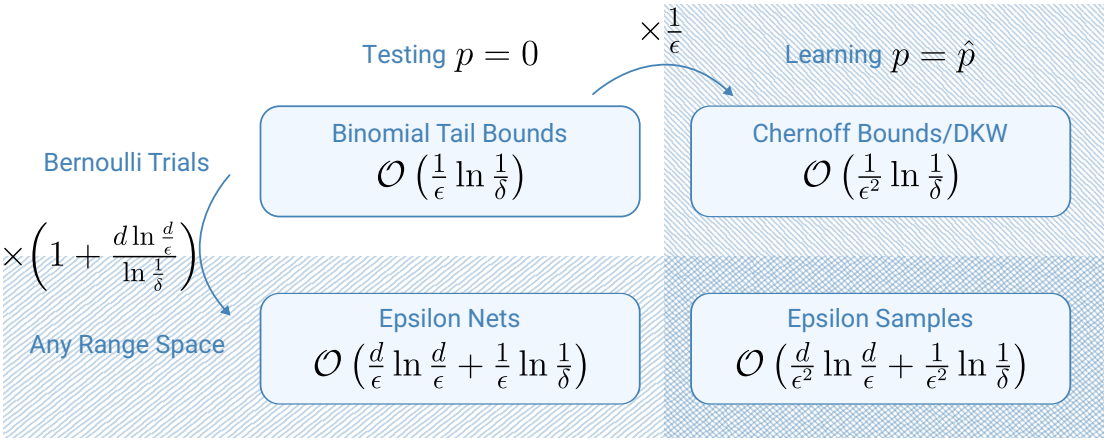


Figure 2.3: Different sampling-based inference methods with their associated sample complexities. The relative costs in terms of sample sizes are illustrated row- and column-wise. The relationships between bounds mainly serve for intuition: marginally tighter bounds can be obtained for ε-nets and ε-samples [Mitzenmacher and Upfal, 2017, Exercise 14.11].

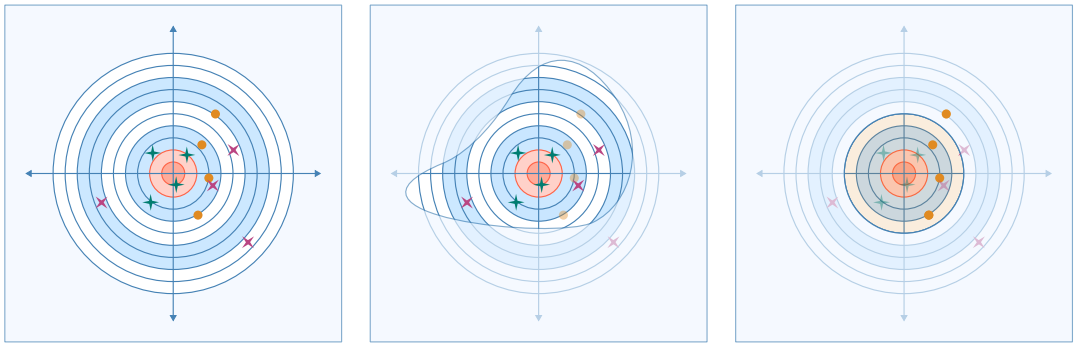
bounds—use DKW bounds, we can only multi-test within a small class of specific range spaces[1].

As an alternative to the use of Chernoff bounds, we investigate the other trade-off depicted in our taxonomy. We restrict ourselves to testing if a probability is close to 0, but explicitly model more complex range spaces. This allows us to simultaneously test all properties in the chosen range space. While this does, in fact, cost more samples, depending on the VC dimension of the range space, for many use cases $d \ll \frac{1}{\varepsilon}$, as we will see in later sections. Consequently, the effective sample complexity of ε-nets is much smaller than for Chernoff bounds. We illustrate the qualitative difference in information obtained by different methods in an example.

Example 2.3.2 (Shooting Range Spaces)**.** Assume we observe archers at a shooting range, noting down the hit pattern in dependence on the wind speed observed in Figure 2.4. We group the shots by wind speed: low wind (✦), medium wind(●) and strong wind (✕). We observe the i.i.d. sample N depicted in Figure 2.4a. Depending on the inference method we use, we can now answer different questions.

With Chernoff bounds, there is no restriction on how we choose our hypothesis, we always have the same sample complexity. One possible hypothesis is depicted in Figure 2.4b, which checks if there was a hit in a specific area of the target during either low or high wind. From the sample, we estimate this probability as $\frac{7}{12}$ and know the true probability of this random event is ε-close to our estimate for a $1 - \delta$ fraction of samples N. We

[1]We recall DKW bounds are designed for the estimation of cumulative distribution functions, which corresponds to learning thresholds or halfspaces.



(a) A shooting range sample (b) A Chernoff range (c) One of many ε-net ranges

Figure 2.4: An i.i.d. sample at a shooting range with different range spaces. Figure 2.4a shows an i.i.d. sample of 12 shots. Figure 2.4b shows a range of ✦ and ✕ shots in a specific area. Figure 2.4c shows one of the ranges in $\mathcal{R} = \{\text{points} \geq x \vee \text{wind} = w : x \in [1, 10], w \in \{✦, ●, ✕\}\}$.

cannot, however, estimate the probability of other random events without additional information.

With an ε-net, the size of the sample depends on the range space. We can, for example, pick the set of ranges $\mathcal{R} = \{(\text{points} \geq x) \vee (\text{wind} = w) : x \in [1, 10], w \in \{✦, ●, ✕\}\}$. Figure 2.4c illustrates one of these ranges $R(7, \text{medium})$, for a threshold 7 points and medium wind. We cannot reliably estimate the probability of the illustrated range, but we can identify the range $R(9, \text{medium})$ to be empty, and assuming N is an ε-net, we now know that, with medium wind, it is less than ε-likely to shot for 9 points or above. From the same sample, we also know that with high wind, most shots will score less than 8 points. All this information is obtainable by checking ranges in $\mathcal{R}$ to be empty or not, without a need to resample. Furthermore, the range space $(\mathbb{R}^3, \mathcal{R})$ has a VC dimension of $d = 3$ and for reasonable levels of uncertainty $d \ll 1/\varepsilon$. Consequently, for a similar uncertainty ε, δ we require much smaller samples for ε-nets than for Chernoff bounds.

The difference between ε-nets and Chernoff bounds or bounds by the DKW inequality becomes apparent from our example: for constant d and δ, both of these two concentration inequalities give sample complexities in $\mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$, but the ε-net only grows with $\mathcal{O}\left(\frac{1}{\varepsilon} \ln \frac{1}{\varepsilon}\right)$. This difference in asymptotic behaviour allows us to obtain samples for very small ε, at the cost of only being able to make a statement about empty ranges. While the sample complexity of an ε-net increases with d, this is not a drawback, but rather a strength of the method. In our context, the concentration inequalities cannot explicitly utilise the structure of our range space. The more complex ε-samples can be used in cases where we explicitly want to combine the capabilities of both tools, but come at a high cost in terms of sample complexity. Going forward, we choose ε-nets as our tools for probabilistic statements, and can now present our verification procedure.

2.4 A Sampling-Based Probabilistic Verification Procedure

In the previous sections, we gave an intuitive explanation for our use of ε-nets and motivated our choice. In this section, we formalise our main approach to verification in an abstract setting. Before this, however, we introduce an additional concept called *quality space*, which is key to the application of our result in the later chapters. This quality space, while not having any utility in isolation, helps us to build powerful intuition for the complexity of range spaces.

We recall that the aim of this thesis is a procedure that is applicable to complex systems like NNs and CPS. If we take an NN for image classification as an example, even in a black-box setting, we have to deal with its complex and high-dimensional data. Departing slightly from our previous notation, an NN is a function $f : \mathcal{X} \to \mathcal{Y}$. Specifications like robustness, which we will discuss in detail in the next chapter, are input-output relations $\mathcal{X} \times \mathcal{Y} \to \{0, 1\}$, and might also take some accessible latent information into account.

A naive approach to defining range spaces seems futile here. The combined space $\mathcal{X} \times \mathcal{Y}$ might be very complex, and a relation R on elements of this space will be difficult to formalize. In most cases, however, we are not directly interested in specific elements in $\mathcal{X} \times \mathcal{Y}$, but rather only a few *qualities* that we can describe in a formal language. We formalise this observation with the introduction of a space $\mathcal{Q}$, the *quality space*. We assume access to a *quality transformation* q with $q : \mathcal{X} \times \mathcal{Y} \to \mathcal{Q}$ that observes the qualities we are interested in. With a slight abuse of notation, we denote q as a unary function from the input space $\mathcal{X}$ going forward, assuming it can access f internally. For example, if we would like to investigate if some function f always produces an output with a larger norm than its input, we could define $q(\mathbf{x}) = (\|\mathbf{x}\|, \|f(\mathbf{x})\|)$ with $\mathcal{Q} = \mathbb{R}^2$. The concept of a quality transformation allows us to make explicit a key observation in our setting: Even if we want to reason over complex spaces, if our properties $\mathcal{R}$ can be defined in a projection of our space into $\mathcal{Q}$, only the structure of $(\mathcal{Q}, \mathcal{R})$ matters.

This intuition is formalised in the following lemma, which describes the VC dimension of range spaces under preprocessing.

Lemma 2.4.1 (Preprocessing)**.** *Let $(\mathcal{X}, \mathcal{R}_\mathcal{X})$ be a range space, $q : \mathcal{X} \to \mathcal{Q}$ be a function from $\mathcal{X}$ into an arbitrary space $\mathcal{Q}$. Let $q(R)$ be the image of $R \in \mathcal{R}_\mathcal{X}$. Then let*

$$\mathcal{R} = \{q(R) : R \in \mathcal{R}_\mathcal{X}\}. \tag{2.31}$$

If q preserves *the ranges, that is,*

$$\forall R \in \mathcal{R}_\mathcal{X} : \forall \mathbf{x} \in \mathcal{X} : \mathbf{x} \notin R \implies q(\mathbf{x}) \notin q(R), \tag{2.32}$$

then it holds that

$$\mathrm{VC}(\mathcal{X}, \mathcal{R}_\mathcal{X}) \leq \mathrm{VC}(\mathcal{Q}, \mathcal{R}) \tag{2.33}$$

In words, if we can project $(\mathcal{X}, \mathcal{R}_\mathcal{X})$ onto some range space $(\mathcal{Q}, \mathcal{R})$ with a function q, then the VC dimension is at most that of the projective range space.

Proof. We show the claim by contradiction. Assume $\mathrm{VC}(\mathcal{Q}, \mathcal{R}) = d$ and there exists a set $S \subset \mathcal{X}$ that is shattered by $(\mathcal{X}, \mathcal{R}_\mathcal{X})$, with $|S| > d$. As S is shattered, we know

$$\forall T \subseteq S : \exists R_\mathcal{X} \in \mathcal{R}_\mathcal{X} : S \cap R_\mathcal{X} = T \tag{2.34}$$

We then know that $|q(S)| = |S| > d$, as for each $\mathbf{x} \in S$ there is a range in $\mathcal{R}_\mathcal{X}$ which contains only this point and, per Equation (2.32), there is also a range in $\mathcal{R}$ which contains only $q(\mathbf{x})$. Consequently, all points in S must be unique under projection with q. This, however, implies

$$\forall T \subseteq S : \exists R \in \mathcal{R} : q(S) \cap R = q(T) \tag{2.35}$$

which means $q(S)$ is shattered by $(\mathcal{Q}, \mathcal{R})$, a contradiction. □

Together with the concept of the quality space, Lemma 2.4.1 helps us to formalize the quasi-independence of our method—and with that our sample complexities—from the properties of the spaces $\mathcal{X}$ and $\mathcal{Y}$. As long as $(\mathcal{Q}, \mathcal{R})$ has a simple structure with a small VC dimension, we can obtain good guarantees with a small number of samples. The following theorem now presents our verification procedure to bound the probability of intersecting a given range in $\mathcal{X}$, without the need to directly characterise it.

Theorem 2.4.2. *Let $\mathcal{D}$ be a probability distribution over some space $\mathcal{X}$, $(\mathcal{Q}, \mathcal{R})$ be a range space with $\mathrm{VC}(\mathcal{Q}, \mathcal{R}) = d$ and $q : \mathcal{X} \to \mathcal{Q}$ be a quality transformation.*

For parameters $0 < \varepsilon, \delta < \frac{1}{2}$, consider a random sample $N \sim \mathcal{D}^s$ with $s = s(\varepsilon, \delta, d)$, as defined in Equation (2.24). Then, with a probability of at least $1 - \delta$, it holds that

$$\forall R \in \mathcal{R} : (q(N) \cap R = \emptyset \implies \Pr(q(X) \in R) < \varepsilon) \tag{2.36}$$

Proof. By Proposition 2.2.4, $q(N)$ is an ε-net for $(\mathcal{Q}, \mathcal{R})$ with probability of at least $1 - \delta$.

If $q(N)$ is an ε-net, it holds that

$$\forall R \in \mathcal{R} : \big(\Pr(q(X) \in R) \geq \varepsilon \implies q(N) \cap R \neq \emptyset\big). \tag{2.37}$$

By contraposition, this statement is equivalent to

$$\forall R \in \mathcal{R} : \big(\Pr(q(X) \in R) < \varepsilon \impliedby q(N) \cap R = \emptyset\big). \tag{2.38}$$

□

With this theorem, we can now confidently—given just one sample of sufficient size—bound the probability of any specification in $\mathcal{R}$ that was *never* intersected with ε. Importantly, we do not need to consider the potentially very complex space $\mathcal{X}$ and construct a range space there, but we can directly work in $\mathcal{Q}$. In order to apply our method, we have just two more requirements. First, we need to obtain a VC-dimension bound for a given range space $(\mathcal{Q}, \mathcal{R})$, which will be the subject of later chapters in this thesis for specific cases. Second, in order to make use of the bound in Theorem 2.4.2, we need to choose a range space with ranges that carry meaning in isolation. This point is more of a qualitative remark to our method, similar to what we discussed in Example 2.3.2. We discuss meaningful range spaces in the following section.

2.5 The Choice of Range Spaces and Coverage Guarantees

Not all range spaces are made equal. In this section, we briefly and informally discuss the fact that properties in formal specification languages are especially advantageous for our method. This helps to motivate our formalisms in the following chapters. Theorem 2.4.2 allows us to bound the probability of any particular range in $\mathcal{R}$ based on a single sample N, but individual ranges must still be selected and tested one by one. If individual (empty) ranges do not bear any meaning, we cannot easily profit from the guarantee provided by an ε-net. In fact, the guarantee allows us to make statements precisely about individual *empty* ranges as noted in Observation 2.2.1. If we want to make a global statement, we can just reason over unions of empty ranges, which makes interpretation difficult for arbitrary geometric shapes. Specification languages like logic are well-suited for this, as their semantics under negation and union are well-defined. Especially formulas over linear inequalities just form intersections and unions of halfspaces, and ease analysis of VC-dimension bounds, as well as interpretation.

We contrast the choice of linear inequalities to the approach by Indri et al. [2024], where the chosen range space consisted of metric balls in the input space of a NN. This work was an early iteration of our method and investigated ε-nets for verification of NN robustness in a slightly different context than this thesis. While metric balls serve as a natural definition of coverage or a similarity relation, the use of ε-nets for establishing coverage leads to problems. Our ε-net N allows us to only make definite statements about ranges that do *not* intersect N. If any particular empty range—in the case of [Indri et al., 2024] a particular metric ball—does not bear meaning to us, we cannot easily avoid our guarantee being vacuous for any particular point in the input space. Furthermore, if we want to make statements about the total probability mass of our distribution $\mathcal{D}$ that is covered in a given region, we need to reason through negation. The following corollary briefly formalises the *only coverage guarantee* we can give with an ε-net:

Corollary 2.5.1 (ε-net Coverage)**.** *Let $(\mathcal{Q}, \mathcal{R})$ be a range space, and $N \subset \mathcal{Q}$ be an ε-net. Let $\mathcal{R}' \subset \mathcal{R}$ be a finite set of ranges with $|\mathcal{R}'| = k$ such that*

$$\forall R' \in \mathcal{R}' : R' \cap N = \emptyset \tag{2.39}$$

Then for the set $\mathcal{C} \subset \mathcal{Q}$ defined as

$$\mathcal{C} = \bigcup_{R' \in \mathcal{R}'} R' \tag{2.40}$$

it holds that

$$\Pr(X \in \mathcal{C}) < k\varepsilon \tag{2.41}$$

In words, if a set $\mathcal{C}$ is the union of k empty ranges, we can bound its probability with $k\varepsilon$.

Proof. The statement follows directly from Definition 2.1.10 with union bounds. □

2.6 Summary

In this chapter, we presented our sampling-based verification procedure in Theorem 2.4.2 and contrasted it to other probabilistic approaches. Our main results here include Proposition 2.2.4 that gives us specific sample complexity bounds for ε-nets, given desired ε,δ and VC dimension d. With the concept of quality spaces and Lemma 2.4.1, we have a tool to obtain good VC bounds for our specific problem settings in the next two chapters. Finally, we have two results that allow us to give guarantees for individual specifications (Theorem 2.4.2) and sets of them for notions of global coverage (Corollary 2.5.1). In the next chapter, we will apply these tools to a specific problem setting.

CHAPTER 3

Sampling-Based Verification of Neural Network Robustness

In the previous chapter, we motivated our theoretical approach to sampling-based verification with ε-nets for any range space with a bounded VC-dimension. We also mentioned that the quality and wealth of information that can be obtained with ε-nets heavily depend on the chosen properties. In this chapter, we move towards one specific application of our verification procedure: checking NN robustness. We first give an overview of (robustness) verification for NNs, from a formal and adversarial perspective. Based on the existing literature, we introduce a notion of probabilistic robustness and the concept of a robustness oracle. These oracles encapsulate local robustness checks for us, and this definition enables us to choose which type of robustness is fitting for a given scenario. Our focus on one specific property in this chapter allows us to expand our theoretical results from Chapter 2, and give sharp lower bounds that generalise to new data with high probability. Finally, we present experimental results that illustrate the practicality and the flexibility of our method in NN verification.

3.1 Preliminaries: Robustness in Neural Networks

In this section, we will introduce the formal notion of NN robustness we consider, as well as all the necessary notation. We largely adhere to our conventions in the previous chapter, with some adaptations better suited to express the notion of a NN as a function. With our notation in place, we give an overview of related work investigating robustness, especially the *certification* of robustness, rather than methods to increase it. This section will first discuss *local* robustness formally and discuss relevant related work. We then discuss formal definitions and limited related work for *global* robustness afterwards.

3.1.1 Local Robustness

We are interested in certifying NNs for *classification* for their robustness against *small input perturbations*. We formalise a given classifier as a function $f : \mathcal{X} \to \mathbb{R}^n$, for an n-class classification task. Here, in contrast to Chapter 2, $\mathcal{X}$ is specifically the input-space of the network. We then say that f predicts class $i \in [1, n]$ if and only if the i-th component of f is maximal. That is, for $\mathbf{x} \in \mathcal{X}$ we define

$$\mathbf{class}_f(\mathbf{x}) = \arg\max_{i \in [1,n]} f_i(\mathbf{x}). \tag{3.1}$$

We then formally define the local robustness of f as follows.

Definition 3.1.1 ((Local) Robustness of a Classifier)**.** Let $\mathcal{X}$ be a metric space with a norm $\|\cdot\|$ and the function $f : \mathcal{X} \to \mathbb{R}^n$ be a classifier. For a point $\mathbf{x} \in \mathcal{X}$ and a parameter $\rho \in \mathbb{R}_+$, we say f is (locally) ρ-robust if and only if

$$\forall \mathbf{x}' \in \mathcal{X} : \Big(\|\mathbf{x} - \mathbf{x}'\| \leq \rho \implies \mathbf{class}_f(\mathbf{x}) = \mathbf{class}_f(\mathbf{x}') \Big). \tag{3.2}$$

In words, f is (locally) robust around $\mathbf{x}$, if in a metric ρ-ball centred around $\mathbf{x}$, the classification of f is constant.

In principle, the choice of metric balls is arbitrary. We can more generally define some bounded neighbourhood around a point $\mathbf{x}$. However, the specific choice of the neighbourhood does not bear any noteworthy consequence on the verification process. We will, therefore, consider metric balls as the natural choice of neighbourhoods and acknowledge that results trivially transfer to other (well-behaved) neighbourhood definitions.

Local robustness can be either assessed with heuristic adversarial attacks on a classifier or formally proven with formal methods. Both approaches have been investigated independently and offer complementary advantages and disadvantages. Formal methods can prove local robustness, but require full knowledge of the model and are intractable for large NNs with many non-linearities, as verifying robustness is hard [Katz et al., 2017]. Adversarial methods can scale well with model complexity and can work with limited information or complete black-box settings. However, they cannot *prove* robustness, only find specific *counterexamples*. Furthermore, Carlini and Wagner [2017b] demonstrated that adversarial attacks do not detect all counterexamples to robustness in practice.

The following sections give an overview of established methods to assess or prove local robustness for NNs. Afterwards, we move on to the more complex issue of global robustness.

3.1.2 Adversarial Methods

NN Robustness has been widely investigated since Szegedy et al. [2014] showed in a seminal work that NNs are susceptible to *adversarial examples*. These inputs are

constructed to be very similar to some specific input, yet having a drastically different prediction. Goodfellow et al. [2015] were among the first to show that such examples can be found easily, with limited access to the classifier. Their Fast Gradient Sign Method (FGSM) accesses the input gradients of a classifier to move towards a close point with a different class. Project Gradient Descent (PGD) [Madry et al., 2018] is an iterative adaptation of FGSM. PGD can be intuitively understood to optimize the *input* of a network towards a different class prediction with gradient descent, with the constraint of staying close to the original input. In addition to this, the Carlini and Wagner [2017a] (C&W) attack uses line-search to find adversarial examples that are particularly close to the original data points. All these methods purely aim to find adversarial examples *for one given point*, their statements are constrained to the corresponding region in the input space $\mathcal{X}$. We discuss how to certify the global robustness of a network with these approaches later in this chapter.

3.1.3 Formal Methods

All the discussed methods to assess NN robustness are heuristic so far. They use techniques to find adversarial examples quickly, but might not find all of them. In contrast to this, methods for formal verification can prove the absence of adversarial examples. Formal methods have been applied to and heavily optimised for NN robustness verification, with various specialised tools currently in use [Meng et al., 2022, Brix et al., 2024]. These tools internally often use satisfiability modulo theory (SMT) solvers or mixed integer programming (MIP) to find a counterexample of local robustness. More formally, they show robustness by *dis*proving (showing unsatisfiability) for a given input

$$\exists \mathbf{x}' \in \mathcal{X} : \Big(\|\mathbf{x} - \mathbf{x}'\| \leq \rho \wedge \mathbf{class}_f(\mathbf{x}) \neq \mathbf{class}_f(\mathbf{x}') \Big). \tag{3.3}$$

Two of the most noteworthy examples for us are $\alpha\beta$-CROWN [Xu et al., 2020, 2021, Wang et al., 2021, Zhang et al., 2022, Shi et al., 2024] and Marabou [Katz et al., 2019, Wu et al., 2024]. In recent competitions [Brix et al., 2024], $\alpha\beta$-CROWN was shown to be the fastest and most versatile formal verification tool for NNs currently recognised. It is composed of a variety of components using different integer-bound propagation techniques to speed up verification and offers efficient implementations that utilise modern GPU architectures well. Given adequate hardware, $\alpha\beta$-CROWN has been reported to tackle even networks with millions of parameters within a few minutes [Brix et al., 2024]. Adequate hardware for $\alpha\beta$-CROWN and its implementation in the LiRPA software library [Xu et al., 2020], however, is outside the reach of consumers for larger networks. Marabou, while not offering the same performance as $\alpha\beta$-CROWN, is designed to be a self-contained, user-friendly NN verification tool. It offers a convenient interface for the Python programming language and can be used out of the box. In this thesis, we use both Marabou and LiRPA for our verification procedure.

3.1.4 Global Robustness

So far, we have introduced methods that certify robustness around a given point in the input space. However, in order to allow NNs to be safely deployed in safety-critical environments, it is not sufficient to show robustness for one point or even a predefined dataset. The NN should be perturbation-robust *everywhere*. This issue has been addressed with different approaches in the literature. Many different training regimes incorporate adversarial attacks into their training regime, producing more robust NNs [Zhang et al., 2019]. Post-processing methods like randomised smoothing [Cohen et al., 2019] can effectively make any given NN certifiably robust, by adding noise to the datapoint before prediction. The NN then produces multiple predictions and returns the average of them. While this result gives robustness guarantees, this does not directly imply absolute resistance to adversarial attacks, due to the stochastic nature of the resulting ensemble model [Maho et al., 2022].

Methods to *improve* robustness aside, a common approach to *assess* robustness is via benchmarking. The model is attacked on a set of points and the fraction of successful attacks is calculated, similar to the evaluation of, e.g., accuracy [Kim et al., 2023]. Commonly, not a lot of thought is given to how the result on the given dataset is representative to the behaviour of the NN as a whole. One noteworthy exception here is the work of Baluta et al. [2021], where a Chernoff bound on NN robustness is produced via sampling, similar to Example 2.1.3. The authors improve on the sample complexity obtained from Theorem 2.1.2, by using an adaptive process with the goal to show that the fraction of robust points is above a specified threshold.

In the realm of formal methods, global robustness has seen limited attention due to the computational limitations of exhaustive methods. Leino et al. [2021], Athavale et al. [2024] use different formalisations of global robustness with the aim of certification. Both of these methods acknowledge that demanding local robustness (Definition 3.1.1) for all inputs is not reasonable, as this would require NNs to give constant predictions across the whole input space. Rather, they give the network the option to *abstain* from predictions and consider only predictions where the network chooses to predict a class. While Leino et al. [2021] introduce an additional class to the network, Athavale et al. [2024] use the *prediction confidence* as a basis for which predictions to consider. The (softmax) confidence is a natural choice for classification tasks. When the network predicts class $c \in [1, n]$, i.e., $\mathbf{class}_f(\mathbf{x}) = c$, the confidence $\mathbf{conf}_f(\mathbf{x})$ is defined as

$$\mathbf{conf}_f(\mathbf{x}) = \frac{\exp(f_c(\mathbf{x}))}{\sum_{i=1}^{n} \exp(f_i(\mathbf{x}))}. \tag{3.4}$$

In principle, different indicators of prediction confidence could be considered. However, the softmax confidence is widely used for classification tasks and is in fact proportional to the distance to the class boundary in the output space, that is,

$$\mathbf{conf}_f(\mathbf{x}) \propto \arg\min_{\mathbf{x}' \in \mathcal{X}} \|f(\mathbf{x}) - f(\mathbf{x}')\| \quad \text{s.t.} \quad \mathbf{class}_f(\mathbf{x}) \neq \mathbf{class}_f(\mathbf{x}'). \tag{3.5}$$

In other words, a larger change in the output of f is needed to change the predicted class and, in expectation, this means a larger change in the input space $\mathcal{X}$ is needed. Because of this, we expect the network to exhibit increasing robustness for more confident predictions. There are different sensible relaxations for global robustness we briefly discuss here. Similarly to Athavale et al. [2024], Kabaha and Drachsler-Cohen [2024] use confidence-based robustness, but choose margin-based confidence instead of softmax. The authors use MIP formulations to then prove margin-confidence-based global robustness with a variety of different neighbourhoods, but are limited to a few thousand neurons in their experiments. Wang et al. [2022] investigate a property that is akin to Lipschitz continuity, where they identify the largest perturbation in the output space of a given NN, given a bounded L_∞ change in the input. The method of the authors focuses on formal verification as well, and their experiments consider networks up to a size of a few thousand neurons. These different definitions of robustness are all reasonable, depending on the exact research questions at play. We use a softmax-confidence-based definition of global robustness, similar to Athavale et al. [2024], because it is the most natural for a classification setting. However, there are no technical limitations that motivate this choice, and our statements can, in principle, be adapted to other definitions of robustness as well.

Definition 3.1.2 (Global Robustness of a Classifier)**.** Let $\mathcal{X}$ be a metric space and the function $f : \mathcal{X} \to \mathbb{R}^n$ be a classifier. For a point $\mathbf{x} \in \mathcal{X}$, a robustness parameter $\rho \in \mathbb{R}_+$ and a real-valued confidence threshold $0 < \kappa < 1$, we say f is globally ρ, κ-robust if and only if

$$\forall \mathbf{x}, \mathbf{x}' \in \mathcal{X} : \Big((\mathbf{conf}_f(\mathbf{x}) \geq \kappa) \wedge (\|\mathbf{x} - \mathbf{x}'\| \leq \rho) \implies \mathbf{class}_f(\mathbf{x}) = \mathbf{class}_f(\mathbf{x}') \Big) \tag{3.6}$$

In words, for all $\mathbf{x} \in \mathcal{X}$ where f is κ-confident, the classification of f is constant in a metric ρ-ball centred around $\mathbf{x}$.

3.2 Probably Approximately Global Robustness

In the previous section, we have defined concepts already used in existing literature, and have motivated our choice of confidence-based global robustness, following the approach of Athavale et al. [2024], Kabaha and Drachsler-Cohen [2024]. In order to stay flexible with respect to the exact mechanism used to quantify robustness locally, we now introduce the notion of a robustness oracle.

Definition 3.2.1 (Robustness Oracle)**.** Let $\mathcal{X}$ be a metric space and $f : \mathcal{X} \to \mathbb{R}^n$ be a classifier. We call a function $\mathbf{rob}_f : \mathcal{X} \to \mathbb{R}$ *robustness oracle*, if it finds an adversarial example $\mathbf{x}'$ using a specified attack model and returns $\|\mathbf{x} - \mathbf{x}'\|$.

We can then redefine local robustness as in Definition 3.1.1 with respect to any given oracle.

Definition 3.2.2 (Local Robustness according to oracle)**.** Let $\mathcal{X}$ be a metric space, the function $f : \mathcal{X} \to \mathbb{R}^n$ be a classifier and $\mathbf{rob}_f$ be a robustness oracle. For a point $\mathbf{x} \in \mathcal{X}$ and a parameter $\rho \in \mathbb{R}_+$, we say f is (locally) ρ-robust according to $\mathbf{rob}_f$, if and only if

$$\mathbf{rob}_f(\mathbf{x}) \geq \rho \tag{3.7}$$

We can now define a probabilistic relaxation of Definition 3.1.2 with respect to some robustness oracle.

Definition 3.2.3 (Approximately Global Robustness)**.** Let $\mathcal{X}$ be a metric space, the function $f : \mathcal{X} \to \mathbb{R}^n$ be a classifier and $\mathbf{rob}_f$ be a robustness oracle. For parameters $\rho \in \mathbb{R}_+$ and $0 < \kappa, \varepsilon < 1$, we say f is *approximately globally robust* under a probability distribution $\mathcal{D}$ according to $\mathbf{rob}_f$ if and only if

$$\Pr(\mathbf{rob}_f(X) < \rho \mid \mathbf{conf}_f(X) \geq \kappa) < \varepsilon. \tag{3.8}$$

The objective of our verification procedure now is to choose a particular κ and infer for which ρ we can obtain a guarantee for this bound. As an extension of the approach described in Chapter 2, we aim to bound a conditional probability here. While this requires more theory, we argue for this choice in the following section.

We choose to bound the conditional probability in particular, as it more naturally models the idea of restricting a statement to only confident enough predictions by Athavale et al. [2024], compared to a bound on the conjunctive probability. We can more easily see this motivation when explicitly rewriting Equation (3.8) as follows using the product rule of probability theory:

$$\Pr(\mathbf{rob}_f(X) < \rho \mid \mathbf{conf}_f(X) \geq \kappa) = \frac{\Pr(\mathbf{rob}_f(X) < \rho \wedge \mathbf{conf}_f(X) \geq \kappa)}{\Pr(\mathbf{conf}_f(X) \geq \kappa)} \tag{3.9}$$

If we opt to just bound the numerator in Equation (3.9), we end up with a vacuous bound for κ where $\Pr(\mathbf{conf}_f(X) \geq \kappa)$ is small.

In the following, we present a method to obtain a bound on this conditional probability. We can obtain an upper bound for the numerator with an ε-net, and use Chernoff (lower) bounds for the denominator. By combining these two bounds, we can show that a given NN is *probably* approximately globally robust.

3.3 Bounds for Conjunctive Probabilities

In this section, we show how to provide a high probability bound for

$$\Pr(\mathbf{rob}_f(X) < \rho \wedge \mathbf{conf}_f(X) \geq \kappa) < \varepsilon. \tag{3.10}$$

The result is a straightforward consequence of Lemma 2.4.1 and Proposition 2.2.4. We begin by defining our quality space $\mathcal{Q}$ for this problem. As our property only depends on

the two real-valued properties $\mathbf{rob}_f$ and $\mathbf{conf}_f$, we can define $\mathcal{Q} = \mathbb{R}^2$ and define the quality transformation q as

$$q(\mathbf{x}) \mapsto (\mathbf{rob}_f(\mathbf{x}), \mathbf{conf}_f(\mathbf{x})). \tag{3.11}$$

In this quality space, we define a range parametrised by a tuple ρ, κ as

$$R(\rho, \kappa) \coloneqq \{(\rho', \kappa') \in \mathbb{R}^2 : \rho' < \rho \wedge \kappa' \geq \kappa\}. \tag{3.12}$$

With this, we can characterise global ρ-κ-robustness trivially: a point $\mathbf{x} \in \mathcal{X}$ is a *counterexample* to ρ-κ-robustness iff $q(\mathbf{x}) \in R(\rho, \kappa)$. We then let $\mathcal{R}_{NN}$ be the set of all these ranges, i.e.,

$$\mathcal{R}_{NN} \coloneqq \{R(\rho, \kappa) : (\rho, \kappa) \in \mathbb{R}^2\}. \tag{3.13}$$

Our range space $(\mathcal{Q}, \mathcal{R}_{NN})$ then coincides with the range space of intersections of axis-aligned half spaces, and has a VC dimension of 2, as shown in Example 2.1.8. With this definition in place, we are ready to obtain the following result with the aid of Lemma 2.4.1.

Lemma 3.3.1. *Let $f : \mathcal{X} \to \mathbb{R}^n$ be a classifier, $\mathcal{D}$ be a probability distribution over $\mathcal{X}$, and q be the quality transformation $q(\mathbf{x}) \mapsto (\mathbf{rob}_f(\mathbf{x}), \mathbf{conf}_f(\mathbf{x}))$.*

For parameters $0 < \varepsilon, \delta < \frac{1}{2}$, consider an i.i.d. sample $N \sim \mathcal{D}$ with $|N| = s(\varepsilon, \delta, 2)$, as defined in Equation (2.24). Then, with a probability of at least $1 - \delta$, it holds that

$$\forall \rho, \kappa : \Big(q(N) \cap R(\rho, \kappa) = \emptyset \implies \Pr(\mathbf{rob}_f(X) < \rho \wedge \mathbf{conf}_f(X) \geq \kappa) < \varepsilon\Big) \tag{3.14}$$

In words, for all tuples ρ,κ, it holds that if we do not find counterexamples in N, f is probably approximately globally ρ-κ robust.

Proof. First, we know that $\mathrm{VC}(\mathbb{R}^2, \mathcal{R}_{NN}) = 2$, as shown in Example 2.1.8. Consequently, $q(N)$ is an ε-net for $(\mathbb{R}^2, \mathcal{R}_{NN})$ under $\mathcal{D}$ with a probability of at least $1 - \delta$, as per Proposition 2.2.4. The claim then follows from Theorem 2.4.2: When $q(N)$ is an ε-net, for all ranges $R(\rho, \kappa)$ that were *not* intersected by $q(N)$, we know that

$$\Pr(q(X) \in R(\rho, \kappa)) = \Pr(\mathbf{rob}_f(X) < \rho \wedge \mathbf{conf}_f(X) \geq \kappa) < \varepsilon. \tag{3.15}$$

□

While we do not directly invoke our preprocessing argument from Lemma 2.4.1 here, we still consider this result partially a consequence of the lemma. Our statement reasons about the quality space $\mathcal{Q}$ as a proxy of $\mathcal{X}$, our preprocessing argument gives us a general explanation for why this is possible. With a method to obtain a bound for the conjunctive probability, we can now continue with a bound for the denominator of Equation (3.9).

3.4 Bounds for Neural Network Confidence

In this section, we show how to obtain a lower bound for $\Pr(\mathbf{conf}_f(X) \geq \kappa) \geq p_{\min}$ from an i.i.d. sample, where $p_{\min}$ is a parameter. We rely on Chernoff bounds to obtain a bound for the *index* of a valid $1 - p_{\min}$ quantile bound for confidence values in an i.i.d. sample N. We temporarily introduce, in the interest of brevity, the real-valued random variable $C = \mathbf{conf}_f(X)$ and use the following lemma to obtain a bound for C based on an i.i.d. sample.

Lemma 3.4.1. *Let C be a real-valued random variable, following some distribution $\mathcal{D}_C$ and N be an i.i.d. sample of C. Denote with $N_{(i)} \in \mathbb{R}$ the i^{th} element in the sample in ascending order, i.e., the smallest index such that $|\{\mathbf{x} \in N : \mathbf{x} \leq N_{(i)}\}| \geq i$.*

For the chosen parameters $0 < \delta < \frac{1}{2}$ and $\frac{1}{2} \leq p < 1$, with a probability of at least $1 - \delta$, we have that

$$\Pr(C \leq N_{(i)}) \leq p \tag{3.16}$$

holds for all $i \in \mathbb{N}$, such that

$$i \leq |N|p - \sqrt{2|N|p \ln \frac{1}{\delta}} \tag{3.17}$$

Proof. This statement is similar to Example 2.1.3, where we obtained a confidence interval for the success probability of a binomially distributed random variable S. For our statement here, we can consider the p-quantile $C_p \in \mathbb{R}$ of the random variable C, and we say that $|N| = s$. For this, we define the p-quantile as $C_p = \inf\{c : \Pr(C \leq c) \geq p\}$. Per definition of C_p, we know that $\Pr(C \leq C_p) \geq p$. Then we let $S = |\{c \in N : c \leq C_p\}|$ be the number of elements in the sample N that are larger than C_p, with $S \sim \text{Binom}(s, p)$. We proceed with the use of Theorem 2.1.2, to find a lower bound for S.

$$\Pr\left(\frac{sp - S}{sp} \geq \varepsilon\right) \leq \exp\left(\frac{-sp\varepsilon^2}{2}\right) \tag{3.18}$$

$$\Pr\left(sp - S \geq sp\varepsilon\right) \leq \exp\left(\frac{-sp\varepsilon^2}{2}\right) \tag{3.19}$$

$$\Pr\left(sp(1-\varepsilon) \geq S\right) \leq \exp\left(\frac{-sp\varepsilon^2}{2}\right) \tag{3.20}$$

$$\Pr\left(S \leq sp(1-\varepsilon)\right) \leq \exp\left(\frac{-sp\varepsilon^2}{2}\right) \leq \delta \tag{3.21}$$

For our choice of δ, p, we now want to find some index i to bound $\Pr(S \leq i) \leq \delta$. We

proceed by choosing $\varepsilon = \frac{sp-i}{sp}$.

$$\Pr\left(S \leq sp(1-\varepsilon)\right) \leq \exp\left(\frac{-sp\varepsilon^2}{2}\right) \leq \delta \tag{3.22}$$

$$\Pr\left(S \leq i\right) \leq \exp\left(\frac{-sp(\frac{sp-i}{sp})^2}{2}\right) \leq \delta \tag{3.23}$$

$$\Pr\left(S \leq i\right) \leq \exp\left(-\frac{(sp-i)^2}{2sp}\right) \leq \delta \tag{3.24}$$

$$-\frac{(sp-i)^2}{2sp} \leq \ln\delta \tag{3.25}$$

$$\frac{(sp-i)^2}{2sp} \geq \ln\frac{1}{\delta} \tag{3.26}$$

$$sp - i \geq \sqrt{2sp\ln\frac{1}{\delta}} \tag{3.27}$$

$$i \leq sp - \sqrt{2sp\ln\frac{1}{\delta}} \tag{3.28}$$

We now know $\Pr\left(S \leq i\right) \leq \delta$ if i satisfies Equation (3.28), and consequently $\Pr\left(S > i\right) \geq 1-\delta$. Finally, as S is the number of elements in our sample N *smaller or equal to* C_p, if and only if $S > i$, then $C_p > N_{(i)}$ which means $\Pr(C \leq N_{(i)}) < p$, and can be relaxed to $\Pr(C \leq N_{(i)}) \leq p$. We have shown $\Pr(C \leq N_{(i)}) \leq p$ holds with a probability of at least $1-\delta$. □

This result allows us to use Chernoff bounds to check which confidence values are safe for us to give our guarantees on. We can then just abstain from giving a guarantee for confidence values which are too rare in our sample to make a statement. For compactness in future reference, we will use

$$i(s,p,\delta) \coloneqq \max_{i\in\mathbb{N}}\left\{i : i \leq sp - \sqrt{2sp\ln\frac{1}{\delta}}\right\} = \left\lfloor sp - \sqrt{2sp\ln\frac{1}{\delta}}\right\rfloor \tag{3.29}$$

In the next section, we will finally combine our two lemmas to bound Equation (3.9).

3.5 Sample-Based PAG Robustness Guarantees

Theorem 3.5.1 (PAG Robustness)**.** *Let $\mathcal{D}$ be a probability distribution, $f : \mathcal{X} \to \mathbb{R}^n$ be a classifier and q be the quality transformation $q(\mathbf{x}) \mapsto (\mathbf{rob}_f(\mathbf{x}), \mathbf{conf}_f(\mathbf{x}))$.*

For parameters $0 < \varepsilon, \delta, p_{\min} < \frac{1}{2}$, consider an i.i.d. sample $N \sim \mathcal{D}^s$ with $s \geq s(\varepsilon, \delta/2, 2)$ as per Equation (2.24) and an integer $i = i(s, 1-p_{\min}, \delta/2)$ as per Equation (3.29). Let $N_{(i)}$ be the i^{th} element in the sample in order of ascending confidence. Then, with

a probability of at least $1-\delta$*, the following implication holds for all* ρ *and for all* $\kappa \leq \mathbf{conf}_f(N_{(i)})$*:*

$$(q(N) \cap R(\rho,\kappa) = \emptyset) \implies \Pr(\mathbf{rob}_f(X) < \rho \mid \mathbf{conf}_f(X) \geq \kappa) < \frac{\varepsilon}{p_{\min}}. \tag{3.30}$$

Proof. We will first show when the statement holds for all ρ and $\kappa \leq \mathbf{conf}_f(N_{(i)})$, where $q(N) \cap R(\rho,\kappa) = \emptyset$.

We will introduce two random events at this point for brevity. Let $E_r = \{\mathbf{rob}_f(X) < \rho\}$ and $E_c = \{\mathbf{conf}_f(X) \geq \kappa\}$. From Lemma 3.3.1, as $s \geq s(\varepsilon, \delta/2, 2)$, we have that with a probability of at least $1-\delta/2$, our sample $q(N)$ is an ε-net, in which case $\Pr(E_r \wedge E_c) < \varepsilon$.

From Lemma 3.4.1, as $i = i(s, 1-p_{\min}, \delta/2)$, with a probability of at least $1-\delta/2$, it holds $\Pr(\mathbf{conf}_f(X) \leq \mathbf{conf}_f(N_{(i)})) \leq 1-p_{\min}$, implying $\Pr(\mathbf{conf}_f(X) \geq \mathbf{conf}_f(N_{(i)})) \geq p_{\min}$ and consequently $\Pr(E_c) \geq p_{\min}$.

Now, by the definition of conditional probability

$$\Pr(E_r|E_c) = \frac{\Pr(E_r \wedge E_c)}{\Pr(E_c)} \tag{3.31}$$

So if $q(N)$ is an ε-net *and* $\Pr(\mathbf{conf}_f(X) \geq N_{(i)}) \geq p_{\min}$ we have

$$\forall \rho, \kappa \leq N_{(i)} : q(N) \cap R(\rho,\kappa) = \emptyset \implies \Pr(E_r|E_c) < \frac{\varepsilon}{p_{\min}}, \tag{3.32}$$

which is equivalent to Equation (3.30). Finally, using the union bound and De Morgan's law,

$$\Pr\left(q(N) \text{ is an } \varepsilon\text{-net} \wedge \Pr(\mathbf{conf}_f(X) \geq N_{(i)}) \geq p_{\min}\right) \geq 1 - \left(\frac{\delta}{2} + \frac{\delta}{2}\right) \tag{3.33}$$

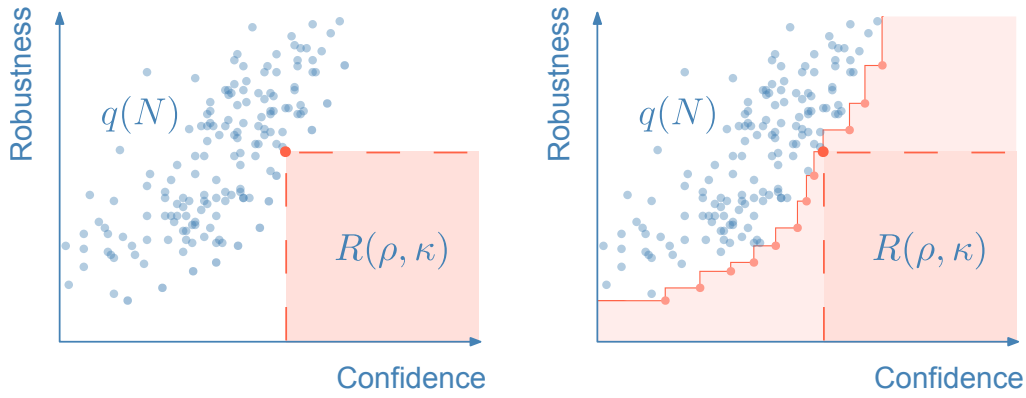
$$\geq 1 - \delta. \tag{3.34}$$

□

This result allows us to sample once and then check whether f is probably approximately globally ρ,κ-robust for any pair of parameters we desire. While, in principle, there is an unbounded number of such checks we could do, we can obtain the full information that our ε-net can provide with just a few checks. We will now show how we can obtain robustness lower bounds from N.

3.6 Robustness Lower-Bounds

In the previous section, we described how to obtain global ρ,κ-robustness guarantees, given an i.i.d. sample N of sufficient size does not contain counterexamples. In this



(a) An ε-net in $\mathcal{Q}$ (b) A cover of maximal empty ranges in $\mathcal{Q}$

Figure 3.1: Construction of lower bounds from an ε-net $q(N)$, depicted in Figure 3.1a. Every empty range $R(\rho, \kappa)$ has a probability smaller than ε. Figure 3.1b shows a cover constructed from the union of all empty ranges. The combined probability mass can be bound by the number of highlighted data points defining the lower envelope.

section, we will exploit the structure of our range space in order to obtain robustness lower bounds conditioned on prediction confidence.

The intuition behind our approach is quite simple: Assume we want to know how robust we can expect f to be for a given datapoint $\mathbf{x}$ from our sample N. We can first measure $\kappa = \mathbf{conf}_f(\mathbf{x})$, and now we search for the largest robustness ρ we can certify with N. We can build a mapping $M(\kappa) \mapsto \rho$ that allows us to access this information efficiently for all choices of κ, defined as

$$M(\kappa) \coloneqq \begin{cases} \min\limits_{\mathbf{x} \in N} & \mathbf{rob}_f(\mathbf{x}) \\ \text{s.t.} & \mathbf{conf}_f(\mathbf{x}) \geq \kappa \end{cases} \quad \text{if } \kappa \leq \kappa_{\max} \qquad \text{UNDEFINED} \quad \text{else} \tag{3.35}$$

where $\kappa_{\max}$ corresponds to the confidence of $N_{(i)}$, with $i = i(|N|, p_{\min}, \delta)$. M gives us a lower envelope of our sample, as illustrated in Section 3.6. Consequently, we know for $\rho = M(\kappa)$ this is the maximal robustness we can guarantee. We do not return any bound for $\kappa > \kappa_{\max}$. We can construct M in time $\mathcal{O}(|N| \log |N|)$ with algorithm Algorithm 3.1. The algorithm follows the following idea: we first sort $q(N)$ by increasing robustness. Thus, for any given pair ρ,κ we encounter, we know we will not see a point with lower robustness than ρ at a later point. Because of this, we can fix $M(\kappa')$ for all values $\kappa' < \kappa$. We can then safely ignore all future confidence values below kappa and have a complete mapping constructed in just a single pass over our sample. The spatial requirement to store the mapping is just the set of tuples that define this lower envelope, highlighted in Figure 3.1b.

With our mapping M we now have a powerful tool for probabilistic inference: we can look at any input $\mathbf{x}$ for our network f, can perform a single forward pass through the

Algorithm 3.1: Obtain ρ-κ-mapping

Input: ε-net N, confidence upper-bound $\kappa_{\max}$
Output: κ-ρ-mapping $M(\kappa)$

```
Let M = ∅
Let κ' = −∞
/* iterate through the sample in order of increasing ρ   */
for (ρ, κ) ∈ q(N) do
    if κ' < κ ≤ κ_max then
        M = M ∪ {κ ↦ ρ} // add new step from κ to ρ to the
            mapping
        κ' = κ
return M
```

network to obtain $\mathbf{conf}_f(\mathbf{x})$ and use M to get a high confidence robustness bound for the point $\mathbf{x}$, without invoking the costly robustness oracle $\mathbf{rob}_f$. However, the guarantee we have obtained with M is conditional, we have not fully addressed how often our bound M is correct, i.e., we have not yet obtained a bound for

$$\Pr\left(\mathbf{rob}_f(X) < M(\mathbf{conf}_f(X)\right). \tag{3.36}$$

Here, our result for ε-net coverage comes into play:

Corollary 3.6.1. *Consider a classifier $f : \mathcal{X} \to \mathbb{R}^n$ and a ρ-κ-mapping M, constructed from an ε-net N as in Equation (3.35). Let $|M|$ be the size of the codomain of M. Then*

$$\Pr\left(\mathbf{rob}_f(X) < |M|\,\mathbf{conf}_f(X)\right) < |M|\varepsilon. \tag{3.37}$$

Proof. The result follows from Corollary 2.5.1. Each tuple in M defines a range; thus, M corresponds to a set of $|M|$ empty ranges. □

3.7 Experiments

Up until now, we focused purely on theoretical statements. We are yet to demonstrate that our procedure is useful in practical settings with real classifiers. This requires addressing the following questions about the practicality of our approach.

Q1 We assume $\mathbf{rob}_f$ *quantifies* robustness, yet common methods focus on true/false statements. How can we model robustness oracles that quantify robustness?

Q2 If we use a heuristic oracle $\mathbf{rob}_f$, it will potentially miss close adversarial examples. What is the precise interpretation of the guarantees if they do not consider these overlooked examples?

Q3 We assume we can sample from $\mathcal{D}$ without limit. How can we sample i.i.d. with just a finite dataset?

Q4 The guarantees are only meaningful if the worst-case robustness of f increases with $\mathbf{conf}_f$. Do NNs show this behaviour in practice?

Q5 We construct the mapping M from our sample N and plan to use it instead of $\mathbf{rob}_f$ at inference time. A size of $\mathcal{O}(|N|)$ can be prohibitively large and make the coverage guarantee in Corollary 3.6.1 vacuous. What is the size of M in practice?

We will address all these questions in this section with a demonstration that our guarantees transfer from theory to practice. At the relevant parts, we will give a brief response to these questions.

3.7.1 Experimental Setup

We train classifiers on two architectures with different training methods for two different image classification datasets for our experiments. The code for the experiments, including hyperparameter settings and training details, can be found at this GitHub repository[1].

Used Software and Hardware We use PyTorch [Ansel et al., 2024] to conduct our experiments and perform training with the MAIR library [Kim et al., 2023]. All the experiments were run on a single desktop machine equipped with an Intel i9-11900KF @ 3.50GHz CPU and an NVIDIA GeForce RTX 3080 GPU.

Architectures The smaller network is a feed-forward NN with ReLU activation functions trained on the MNIST handwritten digit recognition task [Deng, 2012]. The **MNIST** network is fully connected with (768,50,10) neurons for a total of 38900 parameters, trained for 20 epochs. The larger network architecture is a (pretrained) ResNet20 [He et al., 2016], which uses multiple convolutional layers, skip connections and batch normalisation with ReLU activation functions. The total number of parameters is about 0.27M. The pretrained networks were obtained from Github[2]. and trained on **CIFAR10** [Krizhevsky, 2009] for 200 further epochs.

Datasets and Splits Both MNIST and CIFAR10 are 10-class image recognition datasets. The NN takes an image as input and has to classify the depicted digit for MNIST, or the object for CIFAR10. Both of these datasets have predefined splits into a training and testing part. For all experiments, we train the networks on 84% of the training split, and randomly hold out 16% for our verification procedure. For each experimental setup, three random seeds are used, resulting in three different splits in

[1] Code Repository, archived at https://anonymous.4open.science/r/pag-robustness-C500.

[2] Chenyao Yao, *pytorch-cifar-models*, GitHub repository, commit 'd1c8e99' (Mar 3, 2023), archived at https://web.archive.org/web/20250417/https://github.com/chenyaofo/pytorch-cifar-models (accessed April 17, 2025).

train and verification data. The test split is used neither in training nor verification, but will be used to evaluate the guarantees from the verification procedure.

Training- and Verification Procedures For both the **MNIST** and **CIFAR10** architectures, we train three instances of each network, with three random seeds and a mix of training procedures. First, we use standard stochastic gradient descent and compare this with adversarial training, with different weights β on the desired robustness. The adversarial training is performed with the TRADES method [Zhang et al., 2019]. The networks are first trained, and then the verification is performed on the dedicated split of the data. For the verification procedure, we estimate $\mathcal{D}$ with Gaussian noise with a mean of 0 and a standard deviation of 8/256 added to the verification split and sample as many (unlabelled) noisy data points as dictated by $s(\varepsilon, \delta, 2)$. We discuss **Q3** in relation to this choice later. The values of ε and δ are chosen depending on the robustness oracles used.

3.7.2 Robustness Oracles

We use three different robustness oracles for our verification procedures, one based on projected gradient descent (PGD) and two based on exhaustive search with Marabou and LiRPA. We briefly describe how we adapt these oracles in order to obtain a metric output as an answer to **Q1**. Naively, most methods to measure robustness are constructive, so the found adversarial example can be used directly to quantify robustness. This, however, comes with some caveats depending on the method used.

PGD [Madry et al., 2018] uses input gradients in order to optimise the input towards the class boundary. The procedure is iterative, performing a number of gradient steps of fixed size until the predicted class of the NN changes. For normal settings, it is only relevant if an adversarial example was found in a set number of gradient steps. For our setting, we choose a parametrisation with a very small step size and a very high number of steps, performing up to 500 iterations instead of the usual 10-20. While unusual and potentially suboptimal for some settings, this choice of parameters was shown to find very close adversarial examples if they existed and did not often lead to failed attacks. As an alternative to our setup, a line-scan method like in the C&W attack [Carlini and Wagner, 2017a] could be used to obtain even closer adversarial examples with additional computational effort.

In our experiments with PGD as oracle, we choose $\varepsilon = 10^{-4}/\ln(2)$ and $\delta = p_{\min} = 0.01$.

Marabou 2.0 offers a Python interface for NN verification. We transform our MNIST models directly into a set of constraints for Marabou and then, for a given data point $\mathbf{x} \in \mathcal{X}$ check the following query for satisfiability:

$$\exists \mathbf{x}' \in \mathcal{X} : \|\mathbf{x} - \mathbf{x}'\|_\infty < \rho \wedge \mathbf{class}_f(\mathbf{x}) \neq \mathbf{class}_f(\mathbf{x}') \tag{3.38}$$

While the distance constraints are simple linear inequalities, the difference in classes is encoded as piece-wise max linear constraint for $\mathbf{class}_f(\mathbf{x}) = i$ as

$$i \neq \mathbf{class}_f(\mathbf{x}') \iff f_i(\mathbf{x}) < \max_{j \in [1,10]} f_j(\mathbf{x}'). \tag{3.39}$$

For a given query to Marabou, we have to fix ρ, and in general, the found counterexamples tend to be far away from $\mathbf{x}$. As we want to find counterexamples that are as close to $\mathbf{x}$ as possible, we can perform multiple calls as a binary search for the smallest value of ρ where the query for a counterexample is satisfiable. In our experiments with Marabou as oracle, we choose $\varepsilon = 2.5/\ln(2) \cdot 10^{-3}$ and $\delta = 0.01$ and $p_{\min} = 0.05$, with a four-step binary search for ρ. This results in a sample requirement of $s(\varepsilon, \delta/2, 2) = 21294$.

Auto LiRPA is an implementation of $\alpha\beta$-CROWN, and offers a python interface for NN verification. Similar to Marabou, we perform some preprocessing on our MNIST models and use the bound propagation capabilities of LiRPA to obtain a bound. The LiRPA library lets us define a neighbourhood in $\mathcal{X}$ and gives bounds for the logits of f in this neighbourhood. To check if the class of f stays constant around a given point $\mathbf{x}$, with $\mathbf{class}_f(\mathbf{x}) = c$, we query LiRPA for an upper bound $\mathrm{UB}(\mathbf{x})$

$$\mathrm{UB}(\mathbf{x}) = \max_{\mathbf{x}' \in \mathcal{X}} \{f(\mathbf{x}') - f_c(\mathbf{x}') : \|\mathbf{x} - \mathbf{x}'\|_\infty < \rho\}. \tag{3.40}$$

If this upper bound $\mathrm{UB}(\mathbf{x}) = 0$, then f is robust around $\mathbf{x}$. It is worth noting that the bounds reported by LiRPA are not necessarily tight. This means that LiRPA *might underreport* robustness radii. Similarly to Marabou, we then use binary search to find the smallest value for ρ. In our experiments with LiRPA as oracle, we choose $\varepsilon = 2.5/\ln(2) \cdot 10^{-3}$ and $\delta = 0.01$ and $p_{\min} = 0.05$, with a six-step binary search for ρ. This results in a sample requirement of $s(\varepsilon, \delta/2, 2) = 21294$.

Q2: Heuristic Versus Exhaustive Oracles For large NNs, exhaustive methods like Marabou or LiRPA are not feasible. In order to obtain guarantees with low uncertainties, heuristic models have to be used in these cases. But this is not the only reason why a non-exhaustive search method might be used. Exhaustive search is a natural choice for worst-case robustness, but in some settings, a specific *attack model* is of interest. In these settings, attacks are intentionally limited to partial information about the network by assumption. In this scenario, if an attack has, e.g., access to the function output and input gradient only, and the network is resistant to the attack, the network is robust *to this type of attack.*

So, while heuristic methods cannot find all adversarial examples, this does not render the obtained guarantees invalid, but rather guarantees PAG robustness against that particular kind of attack.

3.7.3 Evaluation

For each of our experimental runs, it is not the NNs that are investigated, but our verification procedure. As a consequence, we particularly want to estimate how the

experiments relate to Theorem 3.5.1 and Corollary 3.6.1 based on the test dataset D_{test}. We construct two probability estimators for this purpose.

First, we estimate $\Pr(\mathbf{rob}_f(X) < M(\mathbf{conf}_f(X)))$ with

$$n_c = |\{\mathbf{x}' \in D_{\text{test}} : \mathbf{rob}_f(\mathbf{x}) < M(\mathbf{conf}_f(\mathbf{x}))\}|. \tag{3.41}$$

$n_c/|D_{\text{test}}|$ is an unbiased estimator, and we expect $n_c/|D_{\text{test}}| < |M|\varepsilon$, due to Corollary 3.6.1.

For a given κ we estimate $\Pr(\mathbf{rob}_f(X)) < M(\kappa) \mid \mathbf{conf}_f(X) \geq \kappa)$ with

$$p_\kappa = \frac{|\{\mathbf{x}' \in D_{\text{test}} : \mathbf{rob}_f(\mathbf{x}') < M(\kappa) \wedge \mathbf{conf}_f(\mathbf{x}') \geq \kappa\}|}{|\{\mathbf{x}' \in D_{\text{test}} : \mathbf{conf}_f(\mathbf{x}') \geq \kappa\}|}. \tag{3.42}$$

For a given κ, p_κ is an unbiased estimator, and we expect *all* $p_\kappa < \varepsilon/p_{\min}$, if $\kappa \leq \kappa_{\max}$, due to Theorem 3.5.1.

When evaluating our experiments, we make the intentional choice not to quantify the likelihood of these estimators exceeding our expected thresholds. The reason for this is that we, for our specific setup, have up to around 10000 values for κ and with that 10000 estimators p_κ. Trying to quantify the likelihood that *at least one* of them overestimates the true probability in a way that seemingly breaks our guarantees *is possible*, but we argue this is not meaningful. Any form of hypothesis test would introduce additional probabilistic uncertainties ad absurdum and would require a very high number of performed experiments.

We instead opt for a qualitative investigation: we first define $\hat{p} = \max_{\kappa \leq \kappa_{\max}} p_\kappa$ and check whether $\hat{p} < \varepsilon/p_{\min}$. Using the maximum over all values κ is a very strict estimator, and even in cases where our guarantee holds true, our estimate might exceed the threshold for some κ. If now both $\hat{p}$ and n_c are well below our thresholds anyway, we can assume our guarantees transfer to practice very well, and especially our estimation of the test distribution works sufficiently well. If the estimators do not fit with our parameters ε and δ, we need to investigate possible reasons more closely.

Besides the probabilistic statements, we want to investigate how meaningful the mappings M are: Ideally, they should characterise the worst-case robustness of the given classifier closely on unseen data. Further, we hope that robust and non-robust networks do show a qualitative difference in their behaviour that is captured in the mappings. Both these aspects aim to address **Q4**.

Finally, we are interested in the size of the mapping itself to address **Q5**.

3.7.4 Results

We report an overview of our experimental results in Table 3.1. The full results can be found in Appendix B. Four sets of representative plots are shown in figures Figures 3.2 and 3.3. Table 3.1 shows that our estimators $\hat{p}$ and n_c indicate successful generalization

in most (62/64) of our experiments. A close inspection of the two outliers shows that $\hat{p}$ only ever slightly exceeds our desired threshold of $\varepsilon/p_{\min}$. One of the experiments where this happened is illustrated in Figure 3.2a. We can observe that the apparent violation of our theoretically guaranteed bounds only affects exactly five data points.

Besides our estimators for successful generalisation, visual inspection of our results shows that our bounds do describe the robustness of the network in dependence of the prediction confidence on unseen data very well. Figure 3.2b in particular shows the utility in comparing the performance of two MNIST networks. While the adversarially trained network is much more robust for its high-confidence predictions, the standard training procedure shows higher robustness at lower prediction confidence. This information can be used for conditional ensambling of multiple networks at inference time. The fact that our procedure is able to capture nuanced behaviour like this, not only answers **Q4**, but also shows additional utility of this approach compared to using a simple metric.

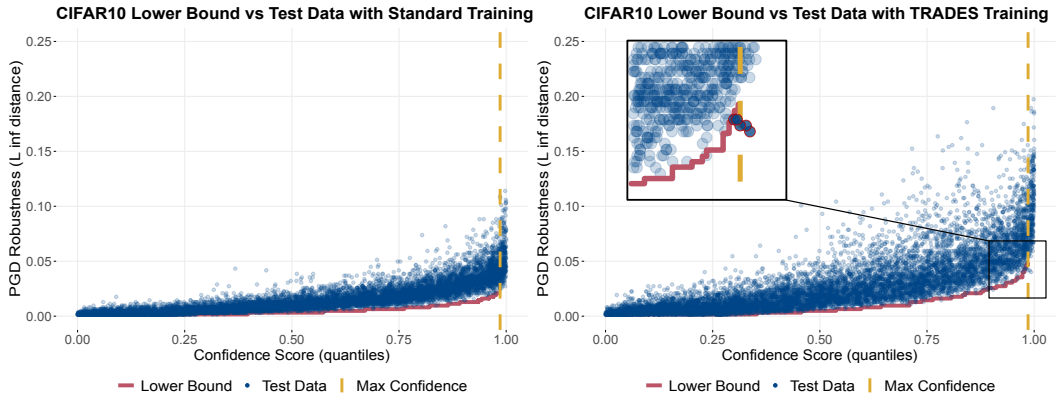
In order to also address **Q5**, the size of the mappings for each network $|M|$ is both shown in Table 3.1 and visually illustrated in Figures 3.2 and 3.3. We can see that in all cases, even though our random samples N are large, $|M|$ is always defined by fewer than 100 data points. This allows us to give tight unconditional bounds (Corollary 3.6.1), and state that the lower bound reported by M will be correct for any data point with a probability of at least $\approx 98\%$ in the *worst case over all experiments*. If this bound is not tight enough for some desired application, it can be artificially discretised by rounding down. This discretisation necessarily will decrease the size of the mapping at the cost of looser robustness bounds. With this, the uncertainty of the lower bound given by M can be reduced in practice.

As a final point of discussion, we address **Q3** and argue for our sampling procedure. We sample from a dataset with additive Gaussian noise to approximate the true distribution of MNIST and CIFAR10, respectively. We do *not* claim this procedure produces samples that are truly i.i.d. from that data distribution. We do remark, however, that despite the simplicity of this approach, our guarantees fit the unseen test data very well. Obtaining i.i.d. samples in practical settings is an issue we cannot give a definitive solution to in this thesis, but our experiments illustrate that even simple approximations seem to suffice as an approximation for non-trivial data.

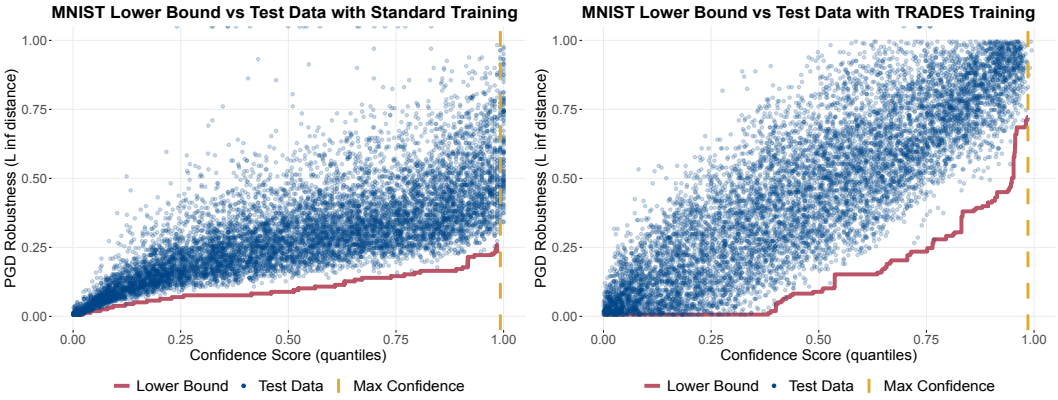
3.8 Extension to Multiple Properties

Before we end this chapter, we briefly discuss a possible extension of our method. We can Theorem 3.5.1 to allow for a more general setting of learning Horn-style clauses over metric properties. A Horn clause is a disjunction of literals, where at most one literal appears unnegated, which can be expressed as a simple rule. In our setting, for a set of $k \in \mathbb{N}$ metric properties $\mathbf{prop}_f$, and a threshold vector $\mathbf{a} \in \mathbb{R}^k$, these formulas then are of the form

$$\mathbf{rob}_f(\mathbf{x}) \geq \rho \leftarrow \mathbf{prop}_f(\mathbf{x}) \geq \mathbf{a}, \tag{3.43}$$

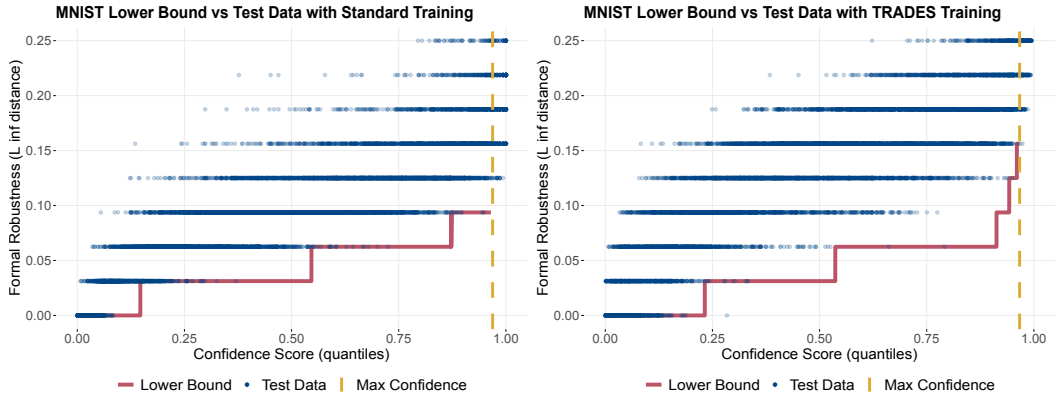


(a) Scatter plot of the CIFAR10 test data set on two ResNet20 networks with PGD oracle. Confidence parameters are $\varepsilon = 10^{-4}/\ln(2)$, $\delta = p_{\min} = 0.01$, with $|N| = s(\varepsilon, \delta/2, 2) = 670313$. On the right-hand side, the 5 counterexamples which cause $\hat{p} > 0.01$ are highlighted. Note that, despite this apparent violation, M tightly fits the test data.

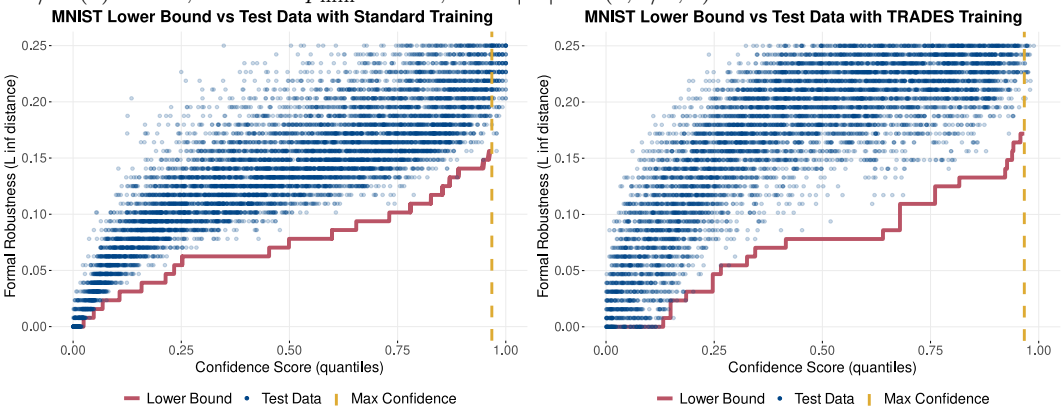


(b) Scatter plot of the MNIST test data set on two feed-forward networks with PGD oracle. For TRADES, $\beta = 2$ was used. Confidence parameters are $\varepsilon = 10^{-4}/\ln(2)$, $\delta = p_{\min} = 0.01$, with $|N| = s(\varepsilon, \delta/2, 2) = 670313$.

Figure 3.2: Scatter plots of the two test datasets D_{test}, with $|D_{\text{test}}| = 10000$, in the quality space $\mathcal{Q}$ with PGD robustness oracles. The networks on the left are trained with standard methods, the right networks are trained robustly with TRADES. The red lines depict the lower bound obtained from the validation sample N. The dashed yellow lines depict $\kappa_{\max}$, the threshold above which M is undefined.



(a) Scatter plot of the MNIST test data set on two feed-forward networks with a four-step Marabou oracle. For TRADES, $\beta = 2$ was used. Confidence parameters are $\varepsilon = 2.5/\ln(2) \cdot 10^{-3}$, $\delta = 0.01$ $p_{\min} = 0.05$, with $|N| = s(\varepsilon, \delta/2, 2) = 21294$.



(b) Scatter plot of the MNIST test data set on two feed-forward networks with a six-step LiRPA oracle. For TRADES, $\beta = 2$ was used. Confidence parameters are $\varepsilon = 2.5/\ln(2) \cdot 10^{-3}$, $\delta = 0.01$ $p_{\min} = 0.05$, with $|N| = s(\varepsilon, \delta/2, 2) = 21294$.

Figure 3.3: Scatter plots of the MNIST test datasets D_{test}, with $|D_{\text{test}}| = 10000$, in the quality space $\mathcal{Q}$ using formal robustness oracles. The networks on the left are trained with standard methods, the right networks are trained robustly with TRADES. The red lines depict the lower bound obtained from the validation sample N. The dashed yellow lines depict $\kappa_{\max}$, the threshold above which M is undefined.

Table 3.1: Summary results for all experiments. We report *worst* results aggregated over 3 random seeds and over the different hyperparameter values used for the TRADES adversarial training. For each experiment, we report the values of $\hat{p}$ and n_c, where bold numbers denote that the estimators are consistent with our guarantees *for all* the $\kappa \leq \kappa_{\max}$, *for all* the runs considered. Moreover, we report the number of individual "good runs" that are consistent with our guarantees when considering the worst-case $\hat{p}$. More extensive results are available in Appendix B.

Dataset	Oracle	Training	$\hat{p} \cdot 10^3$	$\varepsilon/p_{\min} \cdot 10^3$	n_c	$\lvert M \rvert$	good runs
CIFAR	PGD	Standard	**2**	14.4	**2**	12 – 15	3/3
		TRADES	14.8	14.4	**8**	19 – 38	14/15
MNIST	PGD	Standard	**5.3**	14.4	**3**	32 – 42	3/3
		TRADES	19.2	14.4	**5**	7 – 73	34/35
MNIST	Marabou	Standard	**0**	72.1	**0**	3	1/1
		TRADES	**0.1**	72.1	**1**	5	1/1
MNIST	LiRPA	Standard	**0.1**	72.1	**1**	16 – 20	3/3
		TRADES	**0.8**	72.1	**1**	10 – 20	3/3

where $\geq$ is to be interpreted as component-wise comparison. We now show a corollary of Theorem 3.5.1, that extends it to Horn-style clauses over metric properties.

Corollary 3.8.1. *Let $\mathcal{D}$ be a probability distribution, $f : \mathcal{X} \to \mathbb{R}^n$ be a classifier and q be the quality transformation $q(\mathbf{x}) \mapsto (\mathbf{rob}_f(\mathbf{x}), \mathbf{prop}_f(\mathbf{x}))$.*

For parameters $0 < \varepsilon, \delta < \frac{1}{2}$, consider an i.i.d. sample $N \sim \mathcal{D}^s$ with $s \geq s(\varepsilon, \delta/2, k+1)$ as per Equation (2.24).

For any given $\mathbf{a} \in \mathbb{R}^k$, let

$$\hat{p}_{\mathbf{a}} = \frac{|\,\mathbf{prop}_f(\mathbf{x}) \geq \mathbf{a}|}{|N|}. \tag{3.44}$$

and

$$\xi = \sqrt{\frac{1}{2n} \ln \frac{k(n+1)}{2\delta}} \tag{3.45}$$

Then, with a probability of at least $1 - \delta$, the following implication holds for all ρ and for all $\mathbf{a} \in \mathbb{R}^k$, for which $\hat{p}_{\mathbf{a}} > \xi$:

$$(q(N) \cap R(\rho, \mathbf{a}) = \emptyset) \implies \Pr(\mathbf{rob}_f(X) < \rho \mid \mathbf{prop}_f(X) \geq \mathbf{a}) < \frac{\varepsilon}{\hat{p}_{\mathbf{a}} - \xi} \tag{3.46}$$

Proof. We first show that $\Pr\left(\forall \mathbf{a} \in \mathbb{R}^k : p_{\mathbf{a}} \geq \hat{p}_{\mathbf{a}} - \xi\right) \geq 1 - \frac{\delta}{2}$, by derivation analogous

to Lemma 3.4.1. We use the multivariate DKW inequality from Definition 2.1.5.

$$\Pr\left(\sup_{\mathbf{a}\in\mathbb{R}^k} |p_{\mathbf{a}} - \hat{p}_{\mathbf{a}}| > \xi\right) \leq \frac{\delta}{2} \tag{3.47}$$

$$k(n+1)\exp(-2n\xi^2) \leq \frac{\delta}{2} \tag{3.48}$$

$$\exp(-2n\xi^2) \leq \frac{\delta}{2k(n+1)} \tag{3.49}$$

$$-2n\xi^2 \leq \ln\frac{\delta}{2k(n+1)} \tag{3.50}$$

$$2n\xi^2 \geq \ln\frac{k(n+1)}{2\delta} \tag{3.51}$$

$$\xi \geq \sqrt{\frac{1}{2n}\ln\frac{k(n+1)}{2\delta}} \tag{3.52}$$

If Equation (3.52) holds, then we have $\Pr\left(\sup_{\mathbf{a}\in\mathbb{R}^k} |p_{\mathbf{a}} - \hat{p}_{\mathbf{a}}| > \xi\right) \leq \frac{\delta}{2}$, and consequently also $\Pr\left(\forall \mathbf{a} \in \mathbb{R}^k : p_{\mathbf{a}} \geq \hat{p}_{\mathbf{a}} - \xi\right) \geq 1 - \frac{\delta}{2}$.

We now continue in analogy to Theorem 3.5.1. We define the random events $E_r\{\mathbf{rob}_f(X) < \rho\}$ and $E_{\mathbf{a}} = \{\mathbf{prop}_f(X) \geq \mathbf{a}\}$. From Lemma 3.3.1, as $|N| \geq s(\varepsilon, \delta/2, k+1)$, we have that with a probability of at least $1 - \delta/2$, our sample $q(N)$ is an ε-net, in which case $\Pr(E_r \wedge E_{\mathbf{a}}) < \varepsilon$.

From Equation (3.52), we have that with probability of at least $1 - \frac{\delta}{2}$, $\Pr(E_{\mathbf{a}}) \geq \hat{p}_{\mathbf{a}} - \xi$. Again, we have

$$\Pr(E_r \mid E_{\mathbf{a}}) = \frac{\Pr(E_r \wedge E_{\mathbf{a}})}{\Pr(E_{\mathbf{a}})} \tag{3.53}$$

So if $q(N)$ is an ε-net *and* $\forall \mathbf{a} : p_{\mathbf{a}} \geq \hat{p}_{\mathbf{a}} - \xi$ we have

$$\forall \rho, \forall \mathbf{a} : (\hat{p}_{\mathbf{a}} > \xi) \wedge (q(N) \cap R(\rho, \mathbf{a}) = \emptyset) \Longrightarrow \Pr(E_r \mid E_{\mathbf{a}}) < \frac{\varepsilon}{\hat{p}_{\mathbf{a}} - \xi}, \tag{3.54}$$

Which is equivalent to Equation (3.46). Finally, using the union bound and De Morgan's law,

$$\Pr(q(N) \text{ is an } \varepsilon\text{- net} \wedge \forall \mathbf{a} : p_{\mathbf{a}} \geq \hat{p}_{\mathbf{a}} - \xi) \geq 1 - \left(\frac{\delta}{2} + \frac{\delta}{2}\right) \geq 1 - \delta. \tag{3.55}$$

□

This result is a straightforward extension of the rest of our theory. It is interesting to explore properties that might be useful for this form of robustness guarantees besides $\mathbf{conf}_f$, but this question is, unfortunately, out of scope for this thesis. Similarly, using the DKW inequality for the tail bound enables to capture explicitly how weak the bound becomes for any given probability. This, however, leads to a weaker worst-case bound, as the DKW-inequality is weaker than Chernoff bounds for small values of the probability p.

3.9 Summary, Limitations, and Future Work for NN Robustness

We conclude this section with a summary of our results and briefly mention possible future directions of this procedure for NNs. We have applied our sampling-based verification procedure to effectively "convert" the subjective reported prediction confidence of an unknown classifier f to a more objective estimate of metric robustness. This approach can be used regardless of how robustness is defined, as long as access to an oracle $\mathbf{rob}_f$ can be assumed for the construction of our guarantees. After this initial construction phase, we can guarantee that our mapping M provides a lower bound for $\mathbf{rob}_f$ with high probability. Our sample complexities are independent of f and $\mathbf{rob}_f$, only on ε and δ. Our experiments showed that, depending on the utilised oracle, we can certify large NNs in a negligible time compared to their training time. Furthermore, our experimental results illustrated that prediction confidence is a natural choice for predicting confidence, as real NNs showed a strong increase in local robustness for more confident points. This resulted in informative and meaningful guarantees that showed to hold under non-ideal conditions, i.e., real datasets that might not follow our estimated sampling distribution.

We showed a possible direction for future work with Corollary 3.8.1. Besides this and other trivial adaptations of our method, there is one interesting aspect of robustness oracles we did not investigate: their locality in the input space. While our abstraction led to favourable sample complexities, we gave up on the information on how close two data points in our sample are in the input space. Explicitly utilising this information might reduce the number of local robustness checks that need to be performed even further.

Another aspect we did not discuss in this nor the previous chapter is how to interpret the parameter δ. For the event that our procedure fails and our sample does not constitute an ε-net or a valid tail bound, we do not make a statement. In future work, it would be helpful to explicitly mention and formalise the relationship between ε and δ for fixed sample size, and present a statement that combines these two parameters to aid interpretability. An additional important aspect to consider is the precise interpretation of the condition $\mathbf{conf}_f(\mathbf{x}) \geq \kappa$ we use for a given data point $\mathbf{x}$ in contrast to $\mathbf{conf}_f(\mathbf{x}) = \kappa$. The bound we obtain from Theorem 3.5.1 bounds $\Pr(\mathbf{rob}_f(\mathbf{x}) < \rho \mid \mathbf{conf}_f(\mathbf{x}) \geq \kappa)$, which means for *all* predictions with confidence equal to *or more than* κ, we bound the probability of observing a non-robust data point. This is not only different from $\Pr(\mathbf{rob}_f(\mathbf{x}) < \rho \mid \mathbf{conf}_f(\mathbf{x}) = \kappa)$, which gives a bound precisely for κ confident predictions, but also weaker. Assuming robustness increases with confidence, it holds that

$$\Pr(\mathbf{rob}_f(\mathbf{x}) < \rho \mid \mathbf{conf}_f(\mathbf{x}) \geq \kappa) \leq \Pr(\mathbf{rob}_f(\mathbf{x}) < \rho \mid \mathbf{conf}_f(\mathbf{x}) = \kappa). \tag{3.56}$$

This means that our bound should be interpreted precisely as choosing a fixed confidence threshold κ and then giving a probability about how robust the model is when restricted to predictions above this threshold. If we instead want to condition our guarantee on $\mathbf{conf}_f(\mathbf{x}) = \kappa$, we would need other statistical tools, like density estimation. Which bound is preferable depends on the desired interpretation.

CHAPTER 4

Description-Based Verification of Cyber-Physical Systems

Our previous application of NN robustness focuses on one particular and well-motivated type of specification. At the end of Chapter 3, we briefly touch on potential generalisation to Horn Clauses but do not investigate the setting further. In this section, we will apply our theoretical results in Chapter 2 to a different setting: the verification of Cyber-Physical Systems (CPSs). In this chapter, we investigate how we can certify not one specific, but any class of specifications expressible in a particular language: signal temporal logic (STL). STL is a widely used specification language for CPSs. STL extends linear temporal logic (LTL), which is commonly used in model checking applications to deal with real-valued and real-time behaviour common in CPS settings.

In this chapter, we apply our theory to obtaining STL specifications guaranteed to generalise to new observations. We first formally introduce all the additional required notation and definitions we need to translate our theory to a CPS setting, focusing on *parametrised* STL (PSTL) formulas. We then briefly overview related disciplines in CPS verification, especially statistical model checking, specification mining, and anomaly detection. After this overview of related work, the remainder of this chapter focuses on obtaining VC-dimension bounds for PSTL formulas. These VC-dimension bounds are the one requirement we need to apply our verification procedure in this setting. For this, we differentiate between two settings: a general setting where we make no assumption about system behaviour, and the setting of bounded variability systems. We will show that, while the statements we can make in general settings are limited, bounded variability allows us to give a VC-dimension bound for *any* PSTL formula.

4.1 Preliminaries: Verification in Cyber-Physical Systems

This section introduces central concepts in CPS verification, along with the necessary notation. We first formally introduce signals along with the syntax and semantics of STL as a logic that reasons over signals. We then define parametrised STL, which explicitly distinguishes between constants and parameters. Finally, we briefly discuss the concept of well-behaved signals before giving an overview of related work in probabilistic methods in CPS verification.

4.1.1 Signals and Systems

CPSs interact with their environment in real-time, not taking individual points as input but transforming a *signal*. In this section, we formalise what a *system* in the context of CPS verification is. We try to adhere to the notation of Bartocci et al. [2022], with some adaptations to stay consistent with the syntax we introduced for NNs. We first define the *time domain* $\mathbb{T}$ to be a finite interval of the form $\mathbb{T} = [0, t_{\max}] \subset \mathbb{R}_{\geq 0}$. A real-valued *signal* $\mathbf{w}$ then is a curve in some space $\mathcal{X}$, i.e., $\mathbf{w} : \mathbb{T} \to \mathcal{X}$. A *system* f interacts with the environment and produces signals as output. We assume these signals follow a fixed, but unknown distribution $\mathcal{D}$ over $\mathbb{T} \to \mathcal{X}$, and call a randomly sampled signal W with $W \sim \mathcal{D}$. We use (potentially multiple) fixed functions $g : \mathcal{X} \to \mathbb{R}$ to interpret a signal at a given *instant* $t \in \mathbb{T}$ with $g(\mathbf{w}[t]) \in \mathbb{R}$. In analogy to the NN setting, the systems we consider here do not act as functions over points but as functions over curves. The output space we consider at a given instant is implicitly defined as some real space, defined by the real-valued functions g that interpret the value $\mathbf{w}$ at that time.

4.1.2 Signal Temporal Logic

Signal Temporal Logic (STL) [Maler and Nickovic, 2004] is a modal logic, more specifically an extension of Linear Temporal Logic (LTL) to reason over real-valued signals in continuous time. The syntax of an STL formula φ is defined as

$$\varphi ::= \top \mid g(\mathbf{w}) > c \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \mathbf{U}_I \varphi_2 \mid \varphi_1 \mathbf{S}_I \varphi_2. \tag{4.1}$$

Here $\mathbf{U}_I$ and $\mathbf{S}_I$ are the temporal operators *until* and *since*, the subscript I is a time interval in $\mathbb{Q}_{\geq 0} \cup \{\infty\}$ and $g(\mathbf{w}) > c$ is a predicate constructed over the fixed function g and a magnitude value $c \in \mathbb{Q}$. With abuse of notation, we will use $g(\mathbf{w}) > c$ for both a specific signal $\mathbf{w}$ or to denote a free term variable in the formula. In a similar manner, we will often not make the difference between the function symbol g and a specific function in an interpretation explicit.

The semantics of a formula φ in STL is then defined with the relation $(\mathbf{w}, t) \models \varphi$ as

follows [Maler and Nickovic, 2004, Bartocci et al., 2022] for $\mathbf{w} \in \mathbb{T} \to \mathcal{X}$ and $t \in \mathbb{T}$.

$$(\mathbf{w}, t) \models \top \tag{4.2}$$
$$(\mathbf{w}, t) \models g(\mathbf{w}) > 0 \quad \text{iff} \quad g(\mathbf{w}[t]) > 0 \tag{4.3}$$
$$(\mathbf{w}, t) \models \neg\varphi \quad \text{iff} \quad (\mathbf{w}, t) \not\models \varphi \tag{4.4}$$
$$(\mathbf{w}, t) \models \varphi_1 \vee \varphi_2 \quad \text{iff} \quad (\mathbf{w}, t) \models \varphi_1 \text{ or } (\mathbf{w}, t) \models \varphi_2 \tag{4.5}$$
$$(\mathbf{w}, t) \models \varphi_1 \mathbf{U}_I \varphi_2 \quad \text{iff} \quad \exists t' \in t \oplus I : (\mathbf{w}', t') \models \varphi_2 \text{ and } \forall t'' \in (t, t') : (\mathbf{w}, t'') \models \varphi_1 \tag{4.6}$$
$$(\mathbf{w}, t) \models \varphi_1 \mathbf{S}_I \varphi_2 \quad \text{iff} \quad \exists t' \in t \ominus I : (\mathbf{w}', t') \models \varphi_2 \text{ and } \forall t'' \in (t', t) : (\mathbf{w}, t'') \models \varphi_1 \tag{4.7}$$

Here $\oplus$ is the Minkowski sum, defined as $t \oplus I = \{t + a : a \in I\}$, and $\ominus$ analogously denotes the Minkowski difference $t \ominus I = \{t - a : a \in I\}$. Using $\mathbf{U}$ and $\mathbf{S}$ we can then define other standard connectives, like $\bot, \wedge, \to$, the other ordering relations $=, \leq, \geq, >$ in their usual interpretation over the reals and the modal operators *eventually/finally*: $\mathbf{F}_I \varphi = \top \mathbf{U}_I \varphi$ and *always/globally* $\mathbf{G}_I \varphi = \neg \mathbf{F}_I \neg\varphi$. For $t \notin \mathbb{T}$ and *all* STL formulas φ: $(\mathbf{w}, t) \not\models \varphi$. In the following, we will not explicitly mention $\mathbf{S}$ to simplify notation, but all our statements can be rephrased to explicitly include $\mathbf{S}$.

Aside from the interpretation over continuous time, STL differs from LTL in another critical aspect. The (only) predicates in STL are constructed as inequalities over real-valued functions. This imposes an inherent partial order over STL formulas with the same structure, but different timing intervals I and magnitude values c: a formula can be made stricter or less strict by changing these values alone.

In many settings, we want to keep a formula partially fixed but allow for some of the timing or magnitude values to change. We are then interested in finding parameterisations for which the formula is valid for a given system. We will follow Bartocci et al. [2022] and formally introduce *parametric STL* (PSTL) to allow for easy differentiation between constant and parametric magnitudes and intervals, with some adaptations to notation. We define a set of magnitude parameters $A = \{\alpha_1, \ldots, \alpha_m\}$ and a set of timing parameters $T = \{\tau_1, \ldots, \tau_k\}$ with their respective domains $\mathbb{Q}^m$ and $\mathbb{Q}^k$. We then define a PSTL *template* φ as an STL formula, where magnitude and timing constants have been replaced by parameters in A and T. We can then transform a PSTL template back to an STL formula $\varphi_{\mathbf{v}}$ with a set of parameter values $\mathbf{v} \in \mathbb{Q}^m \times \mathbb{Q}^k$. Parameters are simply specific term variables, and $\mathbf{v}$ induces a variable assignment. In order to illustrate how PSTL is used, we will give an example.

Example 4.1.1 (Time to Stopping)**.** Consider PSTL template of the form $\varphi = \mathbf{G}_{[\tau,\infty)}\psi$ with timing parameter τ and some STL formula ψ. In the setting of describing autonomous vehicles, with a constant ε, this could be a formula like

$$\mathbf{G}_{[\tau,\infty)}(\texttt{velocity}(\mathbf{w}) < \varepsilon) \tag{4.8}$$

This formula states that at time τ the vehicle is (close to) stationary. For a system f, we are then interested in checking for which values of $\mathbf{v} \in \mathbb{Q}$ the STL formula $\varphi_{\mathbf{v}}$ is true for all possible signals $\mathbf{w}$ produced by f. Further, we can observe that the possible valuations of φ form a total order: For $\mathbf{v}_1 < \mathbf{v}_2$: $(\mathbf{w}, t) \models \varphi_{\mathbf{v}_1} \implies (\mathbf{w}, t) \models \varphi_{\mathbf{v}_2}$.

4.1.3 Well-Behaved Signals

A signal $\mathbf{w} : \mathbb{T} \to \mathcal{X}$ is a curve over continuous time, and the truth of a formula φ with respect to $\mathbf{w}$ is interpreted at each instant. This raises issues concerning the variability of these signals: in principle, signals can exhibit pathological behaviour that is unrealistic and severely restricts theoretical analysis. An example given by Maler and Nickovic [2004] illustrates this issue.

Example 4.1.2 (Dirichlet Signals)**.** We consider the STL formula $\varphi = g(\mathbf{w}) > 0$ and the signal $\mathbf{w}$ which is defined for $\mathbb{T} = [0, 1]$ as the Dirichlet [1829] function

$$\mathbf{w}[t] = \begin{cases} 1 & \text{if } t \in \mathbb{Q} \\ 0 & \text{else} \end{cases} \tag{4.9}$$

In the finite interval $[0, 1]$ (and any subinterval $[a, b]$ with $0 \leq a < b \leq 1$), the truth value of φ changes an infinite number of times when evaluated on $\mathbf{w}$.

Clearly, the *possibility* of infinite state changes is problematic when a formula should be evaluated on a signal. This evaluation process, called *monitoring* of signals, is nontrivial and has to keep track of the instants where the truth value of subformulas *could* change. Maler and Nickovic [2004] present a monitoring algorithm for STL specifications, which assumes *well-behavedness* of the monitored signals. We will see in later sections that pathological signals also lead to arbitrarily high VC dimensions in most cases. To prevent this, we introduce their additional well-behavedness assumption for signals: *non-Zeno* behaviour, or bounded variability.

The term Zeno behaviour—named after the Greek Philosopher Zeno of Elea—describes a system that exhibits an unbounded number of discrete state transitions in a finite time [Teel et al., 2009]. The term was coined as a reference to Zeno's paradox, put into writing by Aristotle [Sachs and Aristotle., 1995] and embellished by Simplicius [Simplicius, 1989]. A more modern version will be briefly summarised here [Teel et al., 2009]:

A tortoise challenges Achilles to a footrace. Since Achilles is much faster than the tortoise, the tortoise requests two conditions. First, the tortoise gets a head start. Second, Achilles has to keep track of where the tortoise is when he starts running. By the time Achilles has reached the starting position of the tortoise, it has moved forward some distance. Achilles then needs to note where the tortoise currently is again, and so on, every time he reaches the spot the tortoise previously was. Clearly, Achilles can overtake the tortoise in a finite amount of time. However, to do so, he needs to perform an infinite number of tasks: every time he completes a segment of his race, the tortoise has moved on already.

Zeno behaviour is a phenomenon widely recognised and, without careful modelling, can even appear in simple systems, like a model of a bouncing ball [Teel et al., 2009]. We do not plan to investigate this behaviour in this thesis beyond its acknowledgement. To restrict analysis to non-zeno signals, we follow an approach similar to Maler and Nickovic

[2004], Aichernig et al. [2013], Waga et al. [2019] and impose a specific *finite variability* of signals later.

When we require bounded variability, evaluating the semantics of STL simplifies: Maler and Nickovic [2004], Aichernig et al. [2013] argue we can now reduce checking whether a STL formula φ holds on a given signal $\mathbf{w}$ to first detecting events—a well-known problem in numerical integration [Mosterman, 1999]—and subsequently evaluation of φ with $\mathbf{w}$ only for one instant in a time interval. In particular, Waga et al. [2019] extend this idea to PSTL, where intervals can be defined symbolically based on timing parameters. This intuitively means that we can *unroll* our STL formula into a first-order logic formula with inequalities, addition and no other predicates (FO[<,+]) of bounded length. This is similar to the concept of *standard translation* [Kamp, 1968, Blackburn et al., 2001], which is a procedure to translate LTL formulas to FO[<].

4.1.4 Statistical Model Checking and Sequential Testing

Analogous to robustness verification for NNs, both formal and probabilistic methods are used in CPS verification. We are only investigating settings where f is a potentially stochastic black-box system in this thesis. Without a formal model of the system f, formal methods are generally not applicable, and so-called statistical model checking (SMC) has to be performed. The basic approaches in the field are summarised by Agha and Palmskog [2018], Legay et al. [2019] and are briefly presented here. SMC employs simulation-based statistical methods to make statements about the likelihood of specific events in future observations. Depending on the required type of information, SMC differentiates between *quantitative approaches*, where parameter estimation is performed and *qualitative approaches*, which employ hypothesis testing. In the qualitative approach, the task is to confidently decide whether the probability of a specified property φ exceeds a given threshold p_0, i.e., whether $\Pr((W,0) \models \varphi) > p_0$, for a randomly sampled trace W. For each approach, the applicable methods depend on the specific sampling strategy.

Fixed-size sampling is comparable to the standard scenario in statistical hypothesis testing, and admits the application of the standard arsenal of statistical tools [Adcock, 1997]. In *sequential testing*, the idea is that the individual elements of a sample are presented as a data stream. Then the idea is to either accept or reject the tested hypothesis as early as possible. A common test for this setting is the Bayesian sequential probability ratio test (SPRT) [Agha and Palmskog, 2018, Wald, 1992]. SPRT can test the hypotheses $\Pr((W,0) \models \varphi) = p_0$ vs. $\Pr((W,0) \models \varphi) = p_1$ if after a sample N of size $|N| = s$ the specification φ holds true in $s_\varphi = |\{\mathbf{w} \in N : (\mathbf{w},0) \models \varphi\}|$. SPRT keeps track of the likelihood ratio, for $\mathbf{w}_i \in N$:

$$\prod_{i=1}^{s} \frac{\Pr((\mathbf{w}_i,0) \models \varphi \mid p = p_1)}{\Pr((\mathbf{w}_i,0) \models \varphi \mid p = p_0)} = \frac{p_1^{s_\varphi}(1-p_1)^{s-s_\varphi}}{p_0^{s_\varphi}(1-p_0)^{s-s_\varphi}} \tag{4.10}$$

and rejects or accepts the null hypothesis if the ratio exceeds preset thresholds, which need to be chosen depending on the desired confidence in the result [Tartakovsky et al., 2015].

The assumption underlying all the mentioned statistical hypothesis tests is that the investigated system f is stationary, and these tests generally require a fully specified alternative hypothesis. If, instead, the goal is to detect changes in the behaviour of f over time, so-called *changepoint detection* techniques can be employed [Tartakovsky et al., 2015, Montgomery, 2020]. There is a variety of approaches for this problem with nuanced assumptions and properties that are not in the scope of this thesis. As a commonly used method, we briefly introduce the *cumulative sum* (CUSUM) algorithm [Page, 1954, Tartakovsky et al., 2015]. This statistic accumulates deviations from a process mean over time. If the deviation from the mean exceeds a certain threshold at any given point, the process can be considered to be *out of control* or, in our context, the distribution of the system has changed. The CUSUM statistic C_i at the ith simulation in a sequence with a log-likelihood-ratio of LLR_i for $X_i = \mathbb{1}_{(\mathbf{w}_i,0)\models\varphi}$ is defined as [Tartakovsky et al., 2015, eq. (8.72)]

$$C_i = \max\{0, C_{i-1} + \mathrm{LLR}_i\} = \max\left\{0, C_{i-1} + \ln\left(\left(\frac{p_1}{p_0}\right)^{X_i}\left(\frac{1-p_1}{1-p_0}\right)^{1-X_i}\right)\right\}. \quad (4.11)$$

If, after some point in time i, the CUSUM statistic exceeds a chosen threshold h, an anomaly is reported. Choosing h balances false alarm rate with changepoint detection speed, and is often done using a Monte Carlo simulation to tune for the desired average run length of simulations under the null hypothesis until an anomaly is (wrongly) detected. The CUSUM statistic is widely used in statistical quality control settings [Montgomery, 2020] and sometimes in anomaly detection settings [Olufowobi et al., 2019], even though state-of-the-art anomaly detection mechanisms in industrial CPS settings commonly use more involved machine learning based approaches [Acquaah and Roy, 2024]. In our experiments, we will still use this statistically well-founded method to illustrate an application of our results.

In the context of SMC, the setting we investigate, qualitative black-box verification, has seen limited interest. Younes and Simmons [2002] uses an approach based on SPRT we described above. Similar to our approach, Sen et al. [2004], Younes [2005] give statistical guarantees specifically for passively observed (i.i.d.) systems with hypothesis testing. They specifically take the structure of the properties into account to increase the efficiency of their approach compared to standard SPRT. A Bayesian approach for the same problem setting is explored by Zuliani et al. [2013]. The common theme of these works is that they use pre-existing statistical tools and apply them to CPS with a focus on one individual hypothesis. Our additional tools from learning theory will allow us to make statements about classes of hypotheses at once.

4.1.5 Specification Mining

Specification mining in CPS [Bartocci et al., 2022] is a closely related problem to model checking. In qualitative SMC, the question is to *check* whether a given specification φ holds with a given probability. In specification mining, we are interested in knowing *for which parametrisations* $\mathbf{v}$ a given PSTL specification $\varphi_{\mathbf{v}}$ is valid (or holds with high

probability). The aim is for these mind specifications $\varphi_{\mathbf{v}}$ to characterise the system well. As there is typically an unbounded number of admissible parametrisations $\mathbf{v}$, most methods try to find strict solutions. Among the existing work in specification mining, Jones et al. [2014], Jha et al. [2017] specifically assume the setting of passive simulation and learning from positive examples. In their settings, they are given a set of simulation traces and try to find STL specifications that tightly describe the system behaviour. While existing work in specification mining is concerned with the question of how to obtain specifications, in this thesis, we investigate how we can guarantee the quality of STL specifications obtained with such procedures. With this, we aim to bridge the gap between specification mining and SMC to obtain guarantees not only for individually selected hypotheses, but for *all hypotheses* that can be obtained with any particular specification mining procedure.

4.2 PSTL Formulas as Range Spaces

In order to apply our theoretical machinery to PSTL formulas, we first need to formally define range spaces in this context. STL formulas are interpreted by tuples of signals $\mathbf{w} : \mathbb{T} \to \mathcal{X}$ and instants $t \in \mathbb{T}$. For the context of VC-dimension analysis, we do not need to pay special attention to instants t of evaluation, as we will not restrict the classes of possible signals in a time-dependent manner. If there exists a signal $\mathbf{w}$ such that φ is true at some t, we can, in general, construct, via time shifting, signals $\mathbf{w}'$ such that φ is true at any other given $t' \in \mathbb{T}'$. We will therefore define range spaces for PSTL formulas (which are, in essence, just families of STL formulas), as follows:

Definition 4.2.1 (PSTL Range Spaces)**.** Let φ be a PSTL formula with magnitude parameters A, timing parameters T. Let $\mathcal{W}$ be a family of signals $\mathbf{w} \in \mathcal{W}$, such that for some space $\mathcal{X}$ and some time domain $\mathbb{T}$ it holds $\mathbf{w} : \mathbb{T} \to \mathcal{X}$.

We define the PSTL range space $(\mathcal{W}, \varphi)$ to be the range space over the ground set $\mathcal{W}$ where each range $R_{\mathbf{v}}$ is defined by a valuation $\mathbf{v}$ of φ as

$$R_{\mathbf{v}} = \{\mathbf{w} \in \mathcal{W} : \mathbf{w}, 0 \models \varphi_{\mathbf{v}}\} \tag{4.12}$$

With this definition of range spaces in place, we need to be careful in differentiating constants from parameters in our formulas. We therefore introduce a bit of additional notation. In the following, we denote with φ a PSTL formula, and with ψ a *parameter-free*, or *constant* formula in STL. For temporal operators like $\mathbf{U}$, we explicitly write their bounds as $[\tau_1, \tau_2]$ for timing parameters, $[t_1, t_2]$ for timing constants. As an example, the operator $\mathbf{U}_{[t_1,\tau_2]}$ is bound from the constant t_1 to some variable instant τ_2.

With all this notation established, we also need to differentiate between families of possibly Zeno or strictly non-Zeno signals. Allowing Zeno behaviour in signals easily leads to unbounded VC-dimensions, as is illustrated by the following example.

Example 4.2.2 (Families of Pathological Signals)**.** We define the set of signals $\mathcal{W} = \{\mathbf{w}_i : i \in \mathbb{N}_+\}$ in the time domain $\mathbb{T} = [0, 2]$, denoting with r_i the ith prime number, with

$$\mathbf{w}_i[t] = \begin{cases} 1 & \text{if } \exists k \in \mathbb{N}_+ : t = \frac{1}{r_i k} \\ 0 & \text{else} \end{cases} \tag{4.13}$$

Each $\mathbf{w}_i$, has an unbounded amount of instants t where $\mathbf{w}_i[t] = 1$ for $t \to 0$. Now we consider the PSTL formula $\varphi = \mathbf{G}_{[\tau,\tau]} g(\mathbf{w}) = 0$, where τ is a timing parameter. We can shatter a set of signals of unbounded size in $\mathcal{W}$ as follows:

Assume we select a set $\mathbb{I}$ of indices and want to find a parameter valuation $\mathbf{v} \in \mathbb{Q}$ such that

$$(\mathbf{w}_i, 0) \models \varphi_{\mathbf{v}} \iff i \in \mathbb{I}. \tag{4.14}$$

We can simply choose $\mathbf{v} = 1/\prod_{i\in\mathbb{I}} r_i$. For $i \in \mathbb{I}$ we have that $(\mathbf{w}_i, 0) \models \varphi_{\mathbf{v}}$ as

$$t = \frac{1}{k_i \prod_{j\in\mathbb{I}\setminus\{i\}} r_j}. \tag{4.15}$$

For $i \notin \mathbb{I}$ we have that $(\mathbf{w}_i, 0) \not\models \varphi_{\mathbf{v}}$, as can be seen by contradiction: If $(\mathbf{w}_i, 0) \models \varphi_{\mathbf{v}}$, then

$$\exists k \in \mathbb{N}_+ : \mathbf{v} = \frac{1}{\prod_{j\in\mathbb{I}} r_j} = \frac{1}{r_i k} \tag{4.16}$$

$$\exists k \in \mathbb{N}_+ : \prod_{j\in\mathbb{I}} r_j = r_i k \tag{4.17}$$

$$r_i \text{ divides } \prod_{j\in\mathbb{I}} r_j. \tag{4.18}$$

Equation (4.18) is a contradiction to Euclid's lemma: if r_i divides the product, it must divide at least one factor. However, all factors are primes other than r_i.

We have shown that we can find $\mathbf{v}$ such $\varphi_{\mathbf{v}}$ evaluates to true for an arbitrary subset (of unbounded size) of $\mathcal{W}$. The VC dimension of $\mathrm{VC}(\mathcal{W}, \varphi)$ is unbounded.

This example shows that even seemingly simple PSTL formulas can have an unbounded VC dimension if signals with unbounded variability are allowed. We will conduct a more thorough analysis of this setting in the following section. We will show that a wide class of formulas has unbounded VC dimensions and will highlight special cases in which we are able to derive finite bounds. Afterwards, we will see that disallowing Zeno behaviour solves the problem of unbounded VC dimensions, and we give a general result for non-Zeno settings. After our theoretical results, we will briefly see how our theory can connect to practical settings in specification mining and anomaly detection.

4.3 VC-Dimension Bounds for PSTL Formulas in General Settings

In Chapter 3, we argued about the VC dimension of formulas over real-valued thresholds by equating them to intersections of axis-aligned half spaces. The simple structure of these formulas allowed for near-trivial bounds. We now perform a more involved analysis of *exact* (in contrast to asymptotic) VC-dimension bounds of parametrised formulas. In this first step of our analysis, we consider any type of signal without an assumption of bounded variability. We will start with structurally simple formulas, without any temporal operators.

Proposition 4.3.1. *Let $\mathcal{W}$ be the set of all signals over the time domain $\mathbb{T}$ and $\varphi_\wedge$ be a PSTL formula with magnitude parameters $A = \{\alpha_1, \ldots, \alpha_m\}$ of the form*

$$\varphi_\wedge \equiv \bigwedge_{i=1}^{m} g_i(\mathbf{w}) > \alpha_i. \tag{4.19}$$

Then it holds that $\mathrm{VC}(\mathcal{W}, \varphi_\wedge) \leq m$.

Proof. Each fixed function g_i maps signals to a real value. When $\varphi_\wedge$ is evaluated at a fixed instant t, this range space coincides with the range space over axis-aligned half spaces. □

We can extend this result to arbitrary structures of temporal-operator-free formulas, by using the weaker bound of Theorem 2.1.12 by Goldberg and Jerrum [1993].

Corollary 4.3.2. *Let $\mathcal{W}$ be the family of all signals over the time domain $\mathbb{T}$ and φ_{FO} be a temporal-operator-free PSTL formula with magnitude parameters $A = \{\alpha_1, \ldots, \alpha_m\}$, constructed over a number n of distinct inequalities of the form $g(\mathbf{w}) > \alpha_i$. Then it holds that* $\mathrm{VC}(\mathcal{W}, \varphi_{FO}) \leq 2m \log_2(8en)$.

Proof. The result follows from Theorem 2.1.12. □

As the next step, we can simplify the range spaces we analyse by omitting constant subformulas and some temporal operators. In the following, we present a series of small results for such simplifications.

Proposition 4.3.3. *Let $\mathcal{W}$ be the family of all signals over the time domain $\mathbb{T}$ and φ be a PSTL formula with magnitude parameters $A = \{\alpha_1, \ldots, \alpha_m\}$ of the form*

$$\varphi \equiv \mathbf{G}_I(\psi \to \varphi_\wedge), \tag{4.20}$$

such that the subformula ψ and the interval I are parameter free and the subformula $\varphi_\wedge$ is a conjunction of m PSTL literals. Then it holds that $\mathrm{VC}(\mathcal{W}, \varphi) \leq \mathrm{VC}(\mathcal{W}, \varphi_\wedge) \leq m$.

Proof. We can apply the preprocessing Lemma (Lemma 2.4.1) to prove the claim. We define a quality transformation q, that maps signals $\mathbf{w}$ to signals $q(\mathbf{w})$ such that

$$(\mathbf{w}, 0) \models \varphi \iff (q(\mathbf{w}), 0) \models \varphi_\wedge. \tag{4.21}$$

The idea is the following: for each atom of the form $g_i(\mathbf{w}) > \alpha_i$ in $\varphi_\wedge$, we define $q(\mathbf{w})$ such that

$$g_i(q(\mathbf{w})[0]) = \min_{t \in I:(\mathbf{w},t)\models\psi} g_i(\mathbf{w}[t]). \tag{4.22}$$

This preserves classification as defined in Equation (2.32) and proves our claim. □

Corollary 4.3.4. *Let $\mathcal{W}$ be the family of all signals over the time domain $\mathbb{T}$ and φ be a PSTL formula with magnitude parameters $A = \{\alpha_1, \ldots, \alpha_m\}$ of the form*

$$\varphi \equiv \mathbf{F}_I(\psi \wedge \neg\varphi_\wedge) \tag{4.23}$$

Where the subformula ψ is parameter-free and the subformula $\varphi_\wedge$ is a conjunction of m PSTL literals. Then it holds that $\mathrm{VC}(\mathcal{W}, \varphi) \leq \mathrm{VC}(\mathcal{W}, \varphi_\wedge) \leq m$.

Proposition 4.3.5. *Let $\mathcal{W}$ be the family of all signals over the time domain $\mathbb{T}$ and φ be a PSTL formula with magnitude parameters $A = \{\alpha_1, \ldots, \alpha_m\}$, for $\circ \in \{\wedge, \vee\}$ of the form*

$$\varphi \equiv \psi \circ \varphi_1 \tag{4.24}$$

Where the subformula ψ is parameter-free and the subformula φ_1 has VC dimension d. Then it holds that $\mathrm{VC}(\mathcal{W}, \varphi) \leq \mathrm{VC}(\mathcal{W}, \varphi_1) \leq d$.

Proof. Let wlog $\circ = \wedge$. For any signal $\mathbf{w}$ in a shattered set $(\mathbf{w}, 0) \models \psi$, as otherwise the signal could not be classified positively under any parametrisation. Consequently, $(\mathcal{W}, \varphi_1)$ can shatter all sets that are shattered by $(\mathcal{W}, \varphi)$, which proves the claim. □

Proposition 4.3.6. *Let $\mathcal{W}$ be the family of all signals over the time domain $\mathbb{T}$ and φ be a PSTL formula with magnitude parameters $A = \{\alpha_1, \ldots, \alpha_m\}$, of the form*

$$\varphi \equiv (\neg\varphi_\wedge)\mathbf{U}_I\psi \tag{4.25}$$

Where I is a fixed timing interval, the subformula ψ is parameter-free, and the subformula $\varphi_\wedge$ is a conjunction of m literals. Then it holds that $\mathrm{VC}(\mathcal{W}, \varphi) \leq \mathrm{VC}(\mathcal{W}, \psi) \leq m$.

Proof. Maler and Nickovic [2004] show that φ can be rewritten as

$$\varphi \equiv \bigvee_{i=1}^{m} (g_i(\mathbf{w}) > \alpha_i \mathbf{U}_I \psi) \tag{4.26}$$

Then, by a similar argument as in Proposition 4.3.3, we can preprocess any signal $\mathbf{w}$ for which ψ is true at some instant in I, with $t'_{\mathbf{w}} = \arg\min_{t \in I}(\mathbf{w}, t) \models \psi$

$$g_i(q(\mathbf{w})[0]) = \min_{t \in I: t \leq t'_{\mathbf{w}}} g_i(\mathbf{w}[t]) \tag{4.27}$$

This preserves classification. we then have a purely disjunctive clause of literals, and by negation and Proposition 4.3.1 we can show $\mathrm{VC}(\mathcal{W}, \varphi) \leq m$. □

Proposition 4.3.7. *Let $\mathcal{W}$ be the family of all signals over the time domain $\mathbb{T}$ and φ be a PSTL formula with magnitude parameters $A = \{\alpha_1, \ldots, \alpha_m\}$, of the form*

$$\varphi \equiv \psi \mathbf{U}_I(\neg\varphi_\wedge) \tag{4.28}$$

Where I is a fixed timing interval, the subformula ψ is parameter-free, and the subformula $\varphi_\wedge$ is a conjunction of m literals. Then it holds that $\mathrm{VC}(\mathcal{W}, \varphi) \leq \mathrm{VC}(\mathcal{W}, \varphi_\wedge) \leq m$.

Proof. Again, we first consider individual atoms, as $\varphi = \bigvee_{i=1}^{m} \psi \mathbf{U}_I(g_i(\mathbf{w}) > \alpha_i)$. With $t' = \min_{t\in I}(\mathbf{w}, t) \not\models \psi$, we can then preprocess $\mathbf{w}$ such that

$$g_i(q(\mathbf{w})[0]) = \min_{t\in I: t\leq t'_{\mathbf{w}}} g_i(\mathbf{w}[t]) \tag{4.29}$$

and have that

$$(\mathbf{w}, 0) \models \psi \mathbf{U}_I(g_i(\mathbf{w}) > \alpha_i) \iff (q(\mathbf{w}), 0) \models g_i(q(\mathbf{w})) > \alpha_i \tag{4.30}$$

If $(\mathbf{w}, 0) \not\models \psi$, we can let $q(\mathbf{w})$ assume some signal that makes some inequality in $\varphi_\wedge$ false under every parametrisation. By similar reasoning as before, we have that $\mathrm{VC}(\mathcal{W}, \varphi) \leq \mathrm{VC}(\mathcal{W}, \varphi_\wedge) \leq m$. □

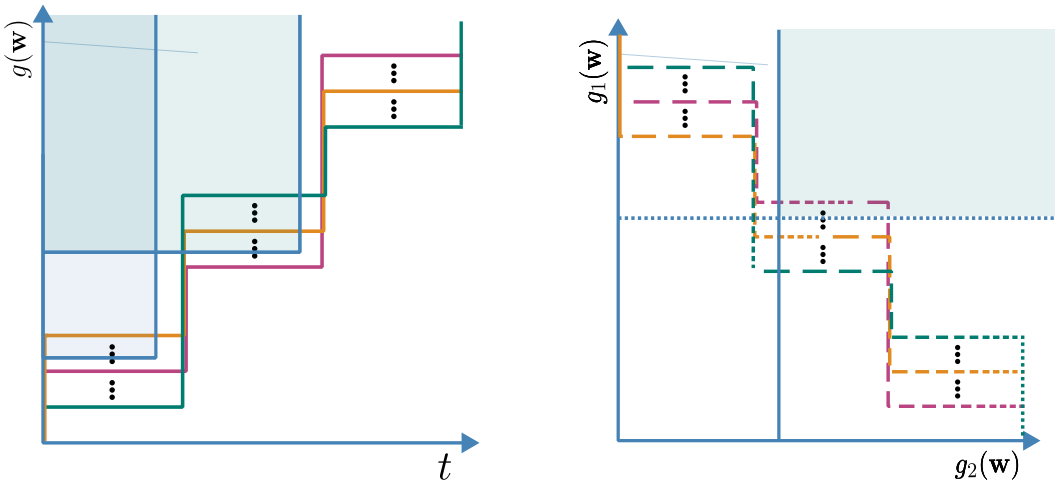
If we want to include temporal operators in our formulas with parameters, the VC dimension is unbounded in most cases. In the following, we show that even seemingly simple formulas outside the classes defined above are of unbounded VC dimension.

Proposition 4.3.8. *The following formulas in PSTL have an unbounded VC dimension.*

1. $\mathbf{F}_{[0,\tau]} g(\mathbf{w}) > \alpha$
2. $\mathbf{F}_{[\tau_1,t_2]} g(\mathbf{w}) > \alpha$
3. $\mathbf{F}(g_1(\mathbf{w}) > \alpha_1 \wedge g_2(\mathbf{w}) > \alpha_2)$
4. $g_1(\mathbf{w}) > \alpha_1 \mathbf{U}\ g_2(\mathbf{w}) > \alpha_2$

Proof. We prove each claim by constructing sets of signals of unbounded size that can be shattered.

1. $\varphi = \mathbf{F}_{[0,\tau]} g(\mathbf{w}) > \alpha$. We assume w.l.o.g. that $\mathbb{T} = [0, n!]$ and denote with π_j the jth permutation of n numbers by some arbitrary numbering. We construct the set of signals for $n \in \mathbb{N}$, $\mathcal{W}_n = \{\mathbf{w}_i : i \in \mathbb{N}, i \leq n\}$, such that $g(\mathbf{w}_i[t]) = \lfloor t \rfloor + \frac{1}{\pi_{\lfloor t \rfloor}(i)}$. We note that $\forall \mathbf{w}, \mathbf{w}' \in \mathcal{W}_n : g(\mathbf{w}[t+1]) > g(\mathbf{w}'[t])$, as $0 < \frac{1}{\pi_{\lfloor t \rfloor}(i)} \leq 1$. or any subset $\mathcal{W}' \subseteq \mathcal{W}_n$, there exists some instant t^*, where $\forall \mathbf{w}' \in \mathcal{W}', \forall \mathbf{w} \in \mathcal{W}_n \setminus \mathcal{W}' : g(\mathbf{w}'[t^*]) > g(\mathbf{w}[t^*])$. f we set $\tau = t^*$ and $\alpha = \max_{\mathbf{w}\in\mathcal{W}_n\setminus\mathcal{W}'} g(\mathbf{w}[t])$, our formula is only true for signals in $\mathcal{W}'$. As n is not bounded, the VC dimension $\mathrm{VC}(\varphi, \mathcal{W})$ is unbounded as well. Figure 4.1a visualizes this construction.



(a) Shattering of $\mathbf{F}_{[0,\tau]}g(\mathbf{w}) > \alpha$ (b) Shattering of $g_1(\mathbf{w}) > \alpha_1 \mathbf{U} g_2(\mathbf{w}) > \alpha_2$

Figure 4.1: Visual Examples of shattered sets of signals of arbitrary size for two PSTL formulas. Figure 4.1a depicts two parametrisations of $\mathbf{F}_{[0,\tau]}g(\mathbf{w}) < \alpha$. A signal satisfies the formula if it touches the corresponding box at any point. Figure 4.1b depicts one parametrisations of $g_1(\mathbf{w}) < \alpha_1 \mathbf{U} g_2(\mathbf{w}) < \alpha_2$. The dash length of the signals indicates time, with later times resulting in shorter dashes. A signal satisfies the formula if it stays above the dotted blue line (α_1) until it is to the right of the solid blue line (α_2) In the depicted set of signals, this corresponds to touching the green box.

2. $\varphi = \mathbf{F}_{[\tau_1,t_2]}g(\mathbf{w}) > \alpha$ the argument is symmetric (as well as the signals) with the function values of $g(\mathbf{w}_i[t]) = -\lfloor t \rfloor - \frac{1}{\pi_{\lfloor t \rfloor}(i)}$.

3. $\varphi = \mathbf{F}(g_1(\mathbf{w}) > \alpha_1 \wedge g_2(\mathbf{w}) > \alpha_2)$. The argument is analogous to the previous examples. We define a set of signals such that $\forall \mathbf{w}, \forall t \in \mathbb{T}$ it holds $g_2(\mathbf{w}[t]) = t$ and proceed as before.

4. $\varphi = g_1(\mathbf{w}) > \alpha_1 \mathbf{U} g_2(\mathbf{w}) > \alpha_2$. The construction of the set of signals is analogous to the previous formulas. We define $g_1(\mathbf{w}[t]) = -\lfloor t \rfloor - \frac{1}{\pi_{\lfloor t \rfloor}(i)}$ and $g_2(\mathbf{w}[t]) = t$. Similarly to our other examples, we can find a parametrisation for φ that is true only for any given subset $\mathcal{W}'$, by finding an instant t^* where $\forall \mathbf{w}' \in \mathcal{W}', \forall \mathbf{w} \in \mathcal{W}_n \setminus \mathcal{W}' : g_1(\mathbf{w}'[t^*]) > g_1(\mathbf{w}[t^*])$ and choosing $\alpha_1 = \min_{\mathbf{w} \in \mathcal{W}_n \setminus \mathcal{W}'} g_1(\mathbf{w}[t^*])$ and $\alpha_2 = t^*$. This is visualised in Figure 4.1b.

□

4.4 Signals with Limited Variability

In order to avoid the issues caused by Zeno behaviour, we need a precise definition for bounded variability. As briefly mentioned in Section 4.1.3, we restrict variability similar

to, e.g., Maler and Nickovic [2004], Waga et al. [2019], who assume interval covers of bounded length. However, our setting requires a very precise definition to allow us to establish specific bounds with Theorem 2.1.12. We recall the requirements for the theorem. First, to apply the bounds to a range space, each range needs to be parametrised with a fixed number of real-valued parameters. This requirement is naturally satisfied in our setting. Second, we need to be able to express membership tests as a boolean formula over polynomial inequalities. The problem in our STL setting is the fact that we potentially evaluate formulas on our signal for an infinite number of instants. If we were to try to translate our STL formula to FO[<,+], this would result in a potentially unbounded number of inequalities. This is where interval covers for formulas come into play. Let us assume, for a signal $\mathbf{w}$ and an STL formula ψ, the existence of an interval cover $\mathcal{I}_{\mathbf{w}}$ with $\bigcup_{I\in\mathcal{I}_{\mathbf{w}}} = \mathbb{T}$, such that

$$\forall I \in \mathcal{I}_{\mathbf{w}} : \forall t, t' \in I : \Big((\mathbf{w}, t) \models \psi \iff (\mathbf{w}, t') \models \psi\Big). \tag{4.31}$$

With an interval cover $\mathcal{I}_{\mathbf{w}}$ we just need to check one instant t per interval $I \in \mathcal{I}_{\mathbf{w}}$, in order to check the truth value of ψ over the whole time domain. This is, however, not enough for us yet. We would require constant truth values not only for one specific valuation of a given PSTL formula, but rather *for all possible valuations of subformulas*. Only then is it ensured that a bounded number of checks is enough to evaluate our formula over the whole time domain and over all possible valuations of a PSTL formula. We will address the issues caused by magnitude and timing parameters separately. For magnitude parameters, we impose the restriction of piecewise-constant signals, where only a bounded number of jumps in $g(\mathbf{w})$ is allowed for any particular function g and signal $\mathbf{w}$. While this restricts our statements, we require this behaviour in order to be able to apply Theorem 2.1.12. Allowing even piecewise-linear signals increases the difficulty of translating PSTL to FO[<,+] significantly for us. In the scope of this thesis, we will, therefore, only consider piecewise-constant signals and define bounded variability as follows, similar to Maler and Nickovic [2004], Younes [2005]. To deal with timing parameters, we can just include them in our definition of intervals, symbolically. Depending on their position in the formula, timing parameters *shift* the instants at which we need to monitor our system. We need a definition of intervals that incorporates these symbolic shifts, which is addressed by Waga et al. [2019].

Definition 4.4.1 (Bounded Variability Signals)**.** Let $\mathbf{w}$ be a signal over the time domain $\mathbb{T}$ and φ be a PSTL formula. We say $\mathbf{w}$ has a *variability* $\zeta = \zeta(\mathbf{w}, \varphi)$ with respect to φ iff over all fixed functions g that occur in φ, the function value $g(\mathbf{w})$ changes at most ζ times in $\mathbb{T}$. More formally, we require the existence of a *symbolic interval cover of size* ζ, a set of intervals $\mathcal{I}_{\mathbf{w}}$, with $|\mathcal{I}_{\mathbf{w}}| \leq \zeta$ and $\bigcup_{\mathcal{I}} = \mathbb{T}$, such that for all valuations of timing parameters and all subformulas φ' that appear in φ

$$\forall I \in \mathcal{I}_{\mathbf{w}} : \forall t, t' \in I : \Big((\mathbf{w}, t) \models \varphi' \iff (\mathbf{w}, t') \models \varphi'\Big). \tag{4.32}$$

In words, $\mathbf{w}$ is piecewise-constant, and can change its *observed* values at most $\zeta - 1$ times, regardless of the value of timing parameters. We call the interval cover symbolic,

as the concrete values might be shifted by timing parameters. Going forward, for an interval cover $\mathcal{I}_{\mathbf{w}}$, we say with abuse of notation that $t_i \in \mathcal{I}_{\mathbf{w}}$ is the first instant of the i-th interval in $\mathcal{I}_{\mathbf{w}}$.

We give a simple example for such an interval cover for the toy PSTL formula $\varphi \equiv g(\mathbf{w}) > 0 \wedge \mathbf{G}_{[\tau_1,\tau_1]}\mathbf{G}_{[\tau_2,\tau_2]}g(\mathbf{w}) > 0$ in $\mathbb{T} = [0,k]$. We assume that for some signal $\mathbf{w}$ we monitor, the value of $g(\mathbf{w})$ can change only at integer times $\{0,\dots,k\}$. The interval cover $\mathcal{I}_{\mathbf{w}}$ can then constructed as $\mathcal{I}_{\mathbf{w}} = \{0,\dots,k\} \cup \{0+\tau_1+\tau_2,\dots,k+\tau_1+\tau_2\}$. This definition makes sure that $\mathbf{w}$ is correctly monitored, regardless of the values of τ_1, τ_2. If we restrict the values of the parameters to integer values, the size of $\mathcal{I}_{\mathbf{w}}$ does not increase. We will make this silent assumption for our experiments for the sake of simplicity, although the use of the full result by Waga et al. [2019] would increase the expressiveness of our method. This aspect is left to future work.

4.5 VC-Dimension Bounds for PSTL Formulas in Limited Variability

In previous sections, we saw that we often have an unbounded VC dimension in PSTL settings. In this section, we show how our assumption of bounded variability can prevent this. We again use the result of Goldberg and Jerrum [1993] and *unroll* PSTL formulas over signals with bounded variability into FO[<,+] formulas, similar to the approach of Aichernig et al. [2013] and the general standard translation method in LTL [Blackburn et al., 2001]. Regardless of the variability, we will assume that constants are eliminated if possible.

To first illustrate how bounded variability prevents unbounded VC dimension, we first inspect Example 4.2.2 again with this added assumption.

Example 4.5.1 (Pathological Signals in non-Zeno Settings)**.** We define the family of signals $\mathcal{W}_\zeta$ with a variability for the formula $G_{[\tau,\tau]}g(\mathbf{w}) > 0$ of at most ζ in the time domain $\mathbb{T} = [0,2]$. That means there exists, for each signal $\mathbf{w} \in \mathcal{W}_\zeta$, an interval cover $\mathcal{I}_{\mathbf{w}}$, where the truth value of $G_{[\tau,\tau]}g(\mathbf{w}) > 0$ does not change. We can translate this formula for a signal $\mathbf{w}$ to FO[<,+] as follows

$$(\mathbf{w},0) \models \mathbf{G}_{[\tau,\tau]}g(\mathbf{w}) > 0 \iff \bigwedge_{t_i \in \mathcal{I}_{\mathbf{w}}} ((\tau \le t_i) \wedge (t_i \le \tau) \rightarrow g(\mathbf{w}[t_i]) > 0). \tag{4.33}$$

This formula can be expressed as a conjunction over at most ζ clauses, with exactly three inequalities each, where t_i are terms that include τ. We can now apply Goldberg and Jerrum [1993]: Each parametrisation is characterised by exactly one real value, the value of q. We have a total of 3ζ inequalities of degree 1. With this, we can now bound the VC dimension $\mathrm{VC}(\mathcal{W}_\zeta, \varphi) \le 2\log_2(24e\zeta)$. We do not have an unbounded VC dimension, and in particular, the VC dimension scales only *logarithmically* with variability ζ.

The contrast between Example 4.2.2 and Example 4.5.1 is substantial. While the unbounded VC dimension in Zeno settings might seem surprising, a logarithmic dependence on variability now might seem counterintuitively low. This scaling does, however, fit the examples we gave in Zeno settings: to increase the size of a shattered set by one, we effectively had to consider twice the number of instants in the time domain. Similarly to this example, we can now obtain VC bounds for all PSTL formulas in a ζ-variability setting, by translating the formula from PSTL into a FOL formula of a length bounded by ζ and then applying the theorem of Goldberg and Jerrum [1993]. This translation takes heavy inspiration from the standard translation of Kamp [1968] from LTL to FOL, and the more accessible version provided by Blackburn et al. [2001].

Definition 4.5.2 (ζ-PSTL Standard Translation)**.** For some integer ζ, let t, t', t'' be *fresh* first-order variables. We define the *standard translation* ST_t, translating PSTL formulas for ζ-variable signals $\mathbf{w}$ to FO[<,+] formulas inductively as follows

$$\mathrm{ST}_t(\top) = \top \tag{4.34}$$
$$\mathrm{ST}_t(g(\mathbf{w}) > c) = g(\mathbf{w}[t]) > c \tag{4.35}$$
$$\mathrm{ST}_t(\neg\varphi) = \neg\,\mathrm{ST}_t(\varphi) \tag{4.36}$$
$$\mathrm{ST}_t(\varphi_1 \vee \varphi_2) = \mathrm{ST}_t(\varphi_1) \vee \mathrm{ST}_t(\varphi_2) \tag{4.37}$$
$$\mathrm{ST}_t(\varphi_1 \mathbf{U}_{[a,b]} \varphi_2) = \exists t' \in t \oplus [a,b] : \Big(\mathrm{ST}_{t'}(\varphi_2) \wedge \forall t'' \in [t+a, t'] : \mathrm{ST}_{t''}(\varphi_1)\Big) \tag{4.38}$$

The standard translation presented here is adapted from Blackburn et al. [2001] with minor adjustments. Our temporal operators have bounded scopes, necessitating additional checks. Each signal $\mathbf{w}$ is then represented by a set of variables $g(\mathbf{w}[t])$ and the corresponding time terms t. To avoid an unbounded number of variables in this formulation, we now use bounded variability: given a symbolic interval cover $\mathcal{I}_\mathbf{w}$ with $|\mathcal{I}_\mathbf{w}| \leq \zeta$, we can reformulate our standard translation as a set of quantifier-free clauses. In each interval, represented by its first instant t_i, we know that the truth values of all subformulas are constant regardless of the parameter values, so we can just check one instant per interval. This results in the following unrolled form of our standard translation.

$$\mathrm{ST}_t(\top) = \top \tag{4.39}$$
$$\mathrm{ST}_t(g(\mathbf{w}) > c) = g(\mathbf{w}[t]) > c \tag{4.40}$$
$$\mathrm{ST}_t(\neg\varphi) = \neg\,\mathrm{ST}_t(\varphi) \tag{4.41}$$
$$\mathrm{ST}_t(\varphi_1 \vee \varphi_2) = \mathrm{ST}_t(\varphi_1) \vee \mathrm{ST}_t(\varphi_2) \tag{4.42}$$
$$\begin{aligned}\mathrm{ST}_t(\varphi_1 \mathbf{U}_{[a,b]} \varphi_2) = \bigvee_{t_i \in \mathcal{I}_\mathbf{w}} \Bigg(&(t + a \leq t_i) \wedge (t_i \leq t + b) \wedge \mathrm{ST}_{t_i}(\varphi_2) \wedge \\ &\Big(\bigwedge_{t_j \in \mathcal{I}_\mathbf{w}} (t + a \leq t_j) \wedge (t_j \leq t_i) \rightarrow \mathrm{ST}_{t_j}(\varphi_1) \Big) \Bigg)\end{aligned} \tag{4.43}$$

In this translation, t, all values t_i and $g(\mathbf{w}[t])$ are first-order term variables. The bounds of temporal operators a, b might be variables or constants, depending on whether they are timing parameters of φ.

Proposition 4.5.3. *Let φ be a PSTL formula, with a set A of magnitude parameters with $|A| = m$, a set T of timing parameters with $|T| = k$, and a total of $\ell \geq 1$ occurrences of temporal operators and n unique inequalities. For a PSTL range space $(\mathcal{W}_\zeta, \varphi)$, where $\mathcal{W}_\zeta$ is the set of all ζ-variable signals over time domain $\mathbb{T}$ with respect to φ, we have*

$$\mathrm{VC}(\mathcal{W}_\zeta, \varphi) \leq 2(m+k)\log_2(8e((n+3)\zeta + (4\ell-3)\zeta^2)) \tag{4.44}$$

Proof. To show our claim, we count the number of *unique* inequalities and free variables in the standard translation $\mathrm{ST}_0(\varphi)$, and then apply Theorem 2.1.12. The number of parameters is $m+k$ by definition. Then, each inequality of the form $g(\mathbf{w}) > c$ in φ introduces one literal in the standard translation *for each instant in* $\mathcal{I}_\mathbf{w}$, on which the inequality is evaluated, up to ζ. For guarding inequalities on *either* t_i or t_j, we introduce ζ literals in the standard translation, as t_i and t_j take all values in $\mathcal{I}_\mathbf{w}$. There are 3 such inequalities per occurrence of an until operator that include t, and 1 such inequality where both t_i *and* t_j appear. The inequalities containing t introduce 3ζ literals for each instant in $\mathcal{I}_\mathbf{w}$, on which the inequality is evaluated, up to $3\zeta^2$. The inequalities containing both t_i *and* t_j introduce ζ^2 literals in the translation. In total, the worst-case number of literals in the translated formula is as follows

- $n\zeta$ literals, for inequalities of the form $g(\mathbf{w}[t]) > c$, evaluated at ζ instants,
- 3ζ literals of the form $(t + a \leq t_i), (t + a \leq t_j), (t_i \leq t + b)$, evaluated only at time 0 for the first temporal operator,
- ζ^2 literals of the form $(t_j \leq t_j)$ for the first temporal operator,
- $3\zeta^2$ literals of the form $(t + a \leq t_i), (t + a \leq t_j), (t_i \leq t + b)$, evaluated at up to ζ instants for the remaining $\ell - 1$ temporal operators,
- ζ^2 literals of the form $(t_j \leq t_j)$ for the remaining $\ell - 1$ temporal operators.

Together, this amounts to $n\zeta + 3\zeta + \zeta^2 + 4(\ell-1)\zeta^2$, or $(n+3)\zeta + (4\ell-3)\zeta^2$.

□

This result is coarser than a direct translation and counting of STL inequalities, but offers the advantage of being general purpose.

4.6 Sample-Based PSTL Validity Guarantees

After our fight through notation and definitions, Proposition 4.5.3 is the result we were hoping to obtain. It allows us to apply the theory from Chapter 2.

Theorem 4.6.1 (Learning Probably Approximately Valid PSTL Parametrisations)**.** *Let φ be a PSTL formula, with a set A of magnitude parameters with $|A| = m$, a set T of timing parameters with $|T| = k$, and a total of n unique inequalitiesand $\ell \geq 1$ occurrences of temporal operators. For parameters $\varepsilon, \delta < \frac{1}{2}$ and a probability distribution $\mathcal{D}_\zeta$ of ζ-variable signals, for a random sample N of signals of size $s(\varepsilon, \delta, 2(m+k)\log_2(8e((n+3)\zeta + (4\ell-3)\zeta^2)))$ according to Equation (2.24), with a probability of at least $1-\delta$ and denoting with W a random signal sampled from $\mathcal{D}_\zeta$, it holds that*

$$\forall \mathbf{v} \in \mathbb{R}^{m+k} : \Big((\forall \mathbf{w} \in N : (\mathbf{w}, 0) \models \varphi_{\mathbf{v}}) \implies \Pr((W, 0) \not\models \varphi_{\mathbf{v}}) < \varepsilon \Big). \tag{4.45}$$

In words, if φ is always true in N with a parametrisation $\mathbf{v}$, the probability of it being false on a random signal W in $\mathcal{D}_\zeta$ is smaller than ε.

Proof. We know that $\mathrm{VC}(\mathcal{W}_\zeta, \varphi) \leq 2(m+k)\log_2(8e((n+3)\zeta + (4\ell-3)\zeta^2))$, by Proposition 4.5.3. Then, N is an ε-net with probability of at least $1-\delta$ by Theorem 2.4.2. The claim follows from the definition of ε-nets. □

With Theorem 4.6.1, we now have a result to apply our verification procedure to any parametrised formula in STL, as long as we can assume bounded variability. In Chapter 3, we continued with the aim of obtaining sharp conditional robustness lower bounds. In this much more general setting, however, we have no motivation to obtain such conditional bounds and will stop our theoretical analysis at this point. The VC-dimension bounds we obtained are, unfortunately, not a single result we can apply indiscriminately. For structurally simple formulas, we can apply our various specialised results, even without assumptions. For more complex formulas, we require limited variability of the system we investigate and need to fall back to the general, but looser Proposition 4.5.3.

Before we continue with an experimental evaluation of our method, we want to set our result in Theorem 4.6.1 into context with some of the existing literature in comparable settings: learning specifications passively from positive examples. While maybe slightly different in their objective, Jha et al. [2017] learn in settings where Theorem 4.6.1 can be applied, even post-hoc.

In their paper Jha et al. [2017] learn parameters for five PSTL templates in an autonomous driving setting, e.g.:

1. $\varphi_1 = \mathbf{G}_{[0,2.2\cdot 10^{11}]}(((\texttt{angle}(\mathbf{w}) \geq 0.2) \vee (\texttt{angle}(\mathbf{w}) \leq -0.2)) \rightarrow (\texttt{speed}(\mathbf{w}) \leq \alpha))$
2. $\varphi_2 = \mathbf{G}_{[0,2.2\cdot 10^{11}]}((\texttt{angle}(\mathbf{w}) \geq 0.06) \rightarrow (\texttt{torque}(\mathbf{w}) \geq \alpha))$

3. $\varphi_3 = \mathbf{G}_{[0,2.2\cdot 10^{11}]}((\texttt{torque}(\mathbf{w}) \leq 0.0) \rightarrow \mathbf{F}_{[0,1.2\cdot 10^{8}]}(\texttt{angle}(\mathbf{w}) \leq \alpha))$

In their more recent work, Nicoletti et al. [2024] consider partially defined STL templates of the form

4. $\varphi_4 = \mathbf{G}((\bigwedge_{i=1}^{d} \mathbf{F}_{[\tau_{i1},\tau_{i2}]}(\alpha_{i1} \leq g_i(\mathbf{w}) \leq \alpha_{i2})) \rightarrow \mathbf{F}\psi)$, for an integer d and temporal-operator-free formula ψ.

We can now immediately and without bounded variability bound the VC dimension of φ_1, φ_2 as 1, via Proposition 4.3.3. For φ_3, we claim that the VC dimension of the PSTL template is unbounded without finite variability. We do not provide a formal proof of this, but argue that this formula can emulate Example 4.2.2. Assuming some bound variability ζ, Proposition 4.5.3 gives the bound $\mathrm{VC}(\mathcal{W}_\zeta, \varphi_3) \leq 2\log_2(8e(5\zeta + 5\zeta^2))$. For the excessively high value $\zeta = 2.2 \cdot 10^{11}$, which is not the number of samples in the trace, but the length of the trace in nanoseconds, this gives $\mathrm{VC}(\mathcal{W}_\zeta, \varphi_3) \leq 164$. The actual number of samples per simulation trace in their experiments was much lower, 13205 samples per trace, giving a VC-dimension bound of 68.

Finally, for φ_4, we need to apply Proposition 4.5.3 as well. We assume ψ contains n_ψ unique inequalities and then get the bound $\mathrm{VC}(\mathcal{W}_\zeta, \varphi_4) \leq 8d\log_2(8e((2d + n_\psi + 3)\zeta + 4(d-1)\zeta^2))$. As their codebase shows, Nicoletti et al. [2024] mine specifications up to $d = 3$, with small n_ψ, and discuss traces of length up to $\zeta = 10^6$. Thus, the highest complexity they consider is $\mathrm{VC}(\mathcal{W}_\zeta, \varphi_4) \leq 1161$, although their actual setup is much more limited in the choice of parameter values.

The complexity of these formulas is comparable to, or in the case of φ_4, with 12 free parameters, even above the complexity of STL templates in comparable work [Bartocci et al., 2022]. Based on the VC-dimension bounds we obtained for these formulas from existing literature, we can give examples for our required sample sizes per Theorem 4.6.1. We fix the parameter $\delta = 0.01$, as it has little impact on the sample complexity and present the results in Table 4.1. We note that the required number of samples for the complex specifications φ_3 and φ_4 is in the hundreds of thousands, if not millions. As these amounts of simulation traces might be computationally costly to obtain, we try to set this number into context and conduct experiments for specification mining and anomaly detection ourselves.

4.7 Experiments on Anomaly Detection in CPS

In the previous section, we have connected our theoretical results to practical settings in a passive manner. For a selection of specifications used in related literature, we gave the required number of i.i.d. simulation traces required to obtain the guarantees offered by Theorem 4.6.1 for any mined specification. However, this comparison is limited as typically not a lot of attention is given to the *amount* of simulation traces. In this section, we actively conduct an experiment for anomaly detection to demonstrate how our results

Specification	Variability ζ	VC dimension	ε		
			0.1	0.01	0.001
φ_1, φ_2	–	1	320	3 919	46 294
φ_3	13205	68	21 022	259 533	3 808 748
φ_3	$2.2 \cdot 10^{11}$	164	55 070	668 853	7 854 135
φ_4	10^6	1161	460 163	5 428 325	62 467 182

Table 4.1: Required sample complexities $s(\varepsilon, \delta, \mathrm{VC}(\mathcal{W}, \varphi))$ as per Equation (2.24) for different PSTL specifications from existing literature given their associated variability and a fixed $\delta = 0.01$.

connect to practice. Similarly to our experiments in Chapter 3, we do not claim to establish empirical evidence for our guarantees. For this, a large-scale experiment would be required to provide evidence beyond all the sources of uncertainty in our setting. We know our sample complexity bounds to be true from theoretical analysis alone, and rather aim to illustrate how our theoretical findings can be used in practice. We want to address the following questions.

Q1 We assumed bounded variability for our theory, and in particular, we needed an interval cover $\mathcal{I}_{\mathbf{w}}$ for our VC-dimension bound. How can we obtain such an interval cover in practical settings?

Q2 Simulation in CPS is very costly compared to inference in NNs. In addition, our VC-dimension bounds are much larger. How long does it take to obtain the required number of samples for complex specifications?

Q3 We assume i.i.d. samples of time series data. Outside of simulations, is i.i.d. sampling a reasonable assumption?

Q4 Do the obtained specifications characterise the CPS well and generalise? Are the sample complexities we require too conservative for mining specifications that generalise?

4.7.1 Experimental Setup

We conduct experiments on a classical toy example in reinforcement learning: the cart-pole balancing problem, studied by—among others—Sutton and Barto [2018], Watkins [1989]. We will set up and train a simple agent with a deep Q-network (DQN) architecture on the cart-pole problem. Once trained, we mine specifications from the system and obtain probabilistic guarantees for their validity from Theorem 4.6.1. Afterwards, we simulate a slow drift in the system by letting the agent perform random actions with increasing probability across simulations. Our hope is that we can detect anomalous system behaviour before a critical failure occurs. We do not claim this setup is comparable

to practically relevant CPS, but it is sufficient to illustrate a practical use case of our method. The code for the experiments, including all hyperparameter settings and training details, can be found at this GitHub repository[1].

Used Software and Hardware We use PyTorch [Ansel et al., 2024] and the Gymnasium library [Brockman et al., 2016] to conduct our experiments in reinforcement learning. All experiments were run on a single desktop machine equipped with an Intel i9-11900KF@3.50GHz CPU and an NVIDIA GeForce RTX 3080 GPU.

Environment and Architectures Our agent is trained in the `CartPole-v1` environment, where it learns to balance a pole vertically against gravity, by moving a cart on a small rail either to the left or to the right. Each simulation begins with the cart and the pole at a random position and speed, each value chosen uniformly in a small range around 0. The simulation lasts for 500 instants and is concluded successfully if the pole has remained upright and the cart is close to a central position for the entire duration. Illustrations of this environment are depicted in Section 4.7.1.

The agent used in the experiments uses a DQN architecture consisting of two three-layer networks and was trained until mastery of the task. For exact training details, we refer to the GitHub repository. For the sake of simulating anomalous behaviour, the agent is given a runtime parameter p_{adv}, which controls the probability it performs a random action at each instant during the simulation.

Verification Procedure We sample two specification templates for the agent.

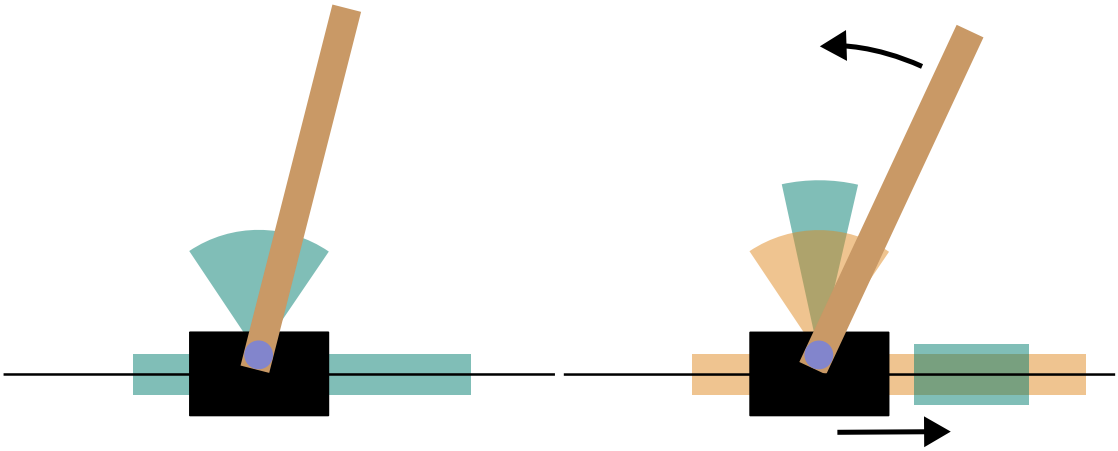
$$\varphi_1 = \mathbf{G}(\texttt{abs\_pole\_angle}(\mathbf{w}) < \alpha_1 \wedge \texttt{abs\_cart\_position}(\mathbf{w}) < \alpha_2) \tag{4.46}$$

which describes the maximal amount of perturbation of the systems per simulation run, with a VC dimension of 2. Figure 4.2a depicts this specification as bounding boxes around the agent. In contrast to this simple property, we will also investigate

$$\begin{aligned}\varphi_2 = &\mathbf{G}_{[10,500]}\Big((\texttt{abs\_pole\_angle}(\mathbf{w}) > \alpha_1 \vee \texttt{abs\_cart\_position}(\mathbf{w}) > \alpha_2) \\ &\rightarrow \mathbf{F}_{[0,\tau]}(\texttt{abs\_pole\_angle}(\mathbf{w}) < \alpha_3 \wedge \texttt{abs\_cart\_position}(\mathbf{w}) > \alpha_4)\Big)\end{aligned} \tag{4.47}$$

The specification template φ_2 imposes a constraint on how quickly the system returns to a stable state after perturbation. Figure 4.2b depicts this specification as two sets of bounding boxes around the agent. If the agent leaves the orange boxes, it has to return to the stable state inside the green boxes in a short amount of time. We parametrize all inequalities in φ_2, resulting in a higher VC dimension for a variability of $\zeta = 500$ of $\text{VC}(\mathcal{W}_\zeta, \varphi_2) \leq \lfloor 2 \cdot 5 \log_2(8e(7\zeta + 5\zeta^2)) \rfloor = 247$. For both of our specifications, we like to obtain an ε-net with the parameters $\varepsilon = \delta = 0.01$. The resulting sample complexities are $s_{\varphi_1} = s(0.01, 0.01, 2) = 6824$ and $s_{\varphi_2} = s(0.01, 0.01, 247) = 1038007$.

[1] "Probably_approximately_valid_STL_mining", Pietreus, GitHub repository, archived at https://github.com/Pietreus/Probably_approximately_valid_STL_mining.



(a) Simple specification φ_1 (b) Complex specification φ_2

Figure 4.2: Illustration of the `CartPole-v1` environment and the specification templates we use in our experiments. Figure 4.2a defines bounding boxes the agent must stay in for the entirety of the simulation. Figure 4.2b defines two sets of bounding boxes. If the agent leaves an acceptable state, i.e., the orange set of boxes, it must return to a stable state inside the green boxes quickly.

Once we have obtained our simulation traces, we mine specifications with simple methods. For φ_1 there is one unique strictest parametrization, for φ_2 we manually choose some parameters based on the behaviour of the system, and then choose the strictest admissible parametrisations for the remaining parameters.

Once our parametrisations have been obtained, we test our chosen specifications in an anomaly detection setting using a CUSUM statistic for changepoint detection, as defined in Equation (4.11). This serves the purpose of demonstrating the practical applicability of our theory and allows us to qualitatively assess how well the mined specifications describe the system. If they do not generalise, we expect to see false alarms during nominal behaviour; if they are not specific enough, they will not detect anomalies before the system fails. We freely choose the parameters for this statistic as null probability $\varepsilon = 0.01$ and alternative probability of $2\varepsilon = 0.02$. The resulting CUSUM statistic is

$$C_i = \max\left\{0, C_{i-1} + X_i \log \frac{2\varepsilon}{\varepsilon} + (1 - X_i) \log \frac{1 - 2\varepsilon}{1 - \varepsilon}\right\} \tag{4.48}$$

$$= \max\left\{0, C_{i-1} + X_i \log 2 + (1 - X_i) \log \frac{0.98}{0.99}\right\} \tag{4.49}$$

We detect an anomaly at simulation i if $C_i > h$, where we estimate h via Monte Carlo simulation as $h = 1.75$.This simple method is in accordance with standard literature [Tartakovsky et al., 2015], as in practice the specific choice of h depends on desired properties which are out of scope of this analysis.

Equipped with this anomaly detection test, we check our specifications in different

settings, where we introduce gradual changes to the agent with its parameter p_{adv}. We will produce 10 sequences of length 20 000 of traces, where the system behaves normally for the first 10 000 traces and then the agent will, for the remaining 10 000 traces, behave randomly with slowly *drifting* probability. That is

$$p_{\text{adv}}^{(i)} = \max\{0, (i - 10^4)/(10^5)\} \tag{4.50}$$

Further, we will produce 10 sequences of length 20 000 of traces where the probability of adversarial movement increases in *steps* of 0.5% every 1000 traces.

$$p_{\text{adv}}^{(i)} = \max\{0, (\lfloor i/10^3 \rfloor)/200\} \tag{4.51}$$

On each of these sequences, we will then calculate our CUSUM statistic for each of the obtained specifications to see how quickly we detect anomalies and whether false positives are detected.

Q1 All our signals are sampled from a simulator at a regular sampling frequency, and we assume piecewise constant signals. This naturally provides us with an interval cover. Furthermore, as our simplistic specification mining process only considers integer values for the timing parameter, the size of this interval cover is identical to the number of samples in the signal. While the assumption of piecewise constant signals is restrictive, signals could be super-sampled artificially to more closely describe more complex behaviour. We then still have piecewise constant signals, but the sampling frequency is artificially increased to reduce potential modelling errors. Other ways to bypass these restrictions could rely on symbolic guards for magnitudes, similar to our symbolic definition of interval covers [Waga et al., 2019].

Q2 With a simple setup without multiprocessing, we can simulate around 50 full simulation traces per second of this agent in this environment. With the simple specification template φ_1, this means we are able to obtain reasonable probabilistic guarantees about our agent in approximately two minutes of simulation. For the much more complex specification φ_2, we would expect around five hours of runtime to sample our ε-net. These runtimes are certainly reasonable, if not surprisingly low. For simple enough specifications like φ_1, the temporal aspects do not influence the VC dimension. In these cases, the sample complexity can even be considered to be of negligible cost compared to training a system.

4.7.2 Evaluation

For our experimental runs, it is not the agent that is investigated, nor how we chose our parameterisations. We want to instead demonstrate the practical feasibility of our verification procedure and the usefulness of its guarantees. We choose the setting of anomaly detection for two reasons. First, a rigorous detection method will require us to know how likely a specification is to hold in a system. Second, the response time and

the false positive rate of any anomaly detection method are coupled to the quality of the specifications used. If the specification is too strict, it does not generalise well, and false positives will be detected. If it is too loose, the specification is non-informative and the system might fail before actual anomalies are detected. Consequently, we want to see that our CUSUM statistics never detect anomalies for simulation traces i where $p_{\text{adv}}^{(i)} = 0$, but *does* detect even small strictly positive $p_{\text{adv}}^{(i)}$ relatively quickly. Further, we will use our specification mining procedure with a small subset of our samples to see if this smaller amount of samples is still sufficient to produce specifications that generalise sufficiently well. We will report for each parametrisation in each setting the smallest, average and maximum value of $p_{\text{adv}}^{(i)}$ at which an anomaly is detected, as well as if system failure was successfully predicted. In addition, we will briefly investigate if our sample complexities are overly conservative. We will assess this by rerunning the experiments with parameterisations obtained from the first 10% of our sampled traces.

4.7.3 Results

We mine one specification for φ_1 ("Optimal") and four for φ_2 ("Balanced", "Angle", "Position", "Combined"), described in detail in Appendix C. The results of our experiments are reported in Table 4.2 and visualised in Figures 4.3 and 4.4. In all 20 experimental runs, not once a process deviation is falsely detected at $p_{\text{adv}} = 0$. Furthermore, with one exception, each specification detected anomalies before the first system failure in at least 18/20 trials.

The single specification obtained for φ_1 proves to be particularly effective and, in the worst trial, detects a deviation from nominal behaviour at $p_{\text{adv}} \approx 1.53\%$. For the different specifications of φ_2, even though in each α_3 and α_4 are chosen as strictly as possible, they show drastic differences in their sensitivity. Both the "Angle" specification as well as the "Position" specification detect deviations with reasonable effectiveness. The "Balanced" specification, however, does not register anomalous behaviour in the system in 17/20 trials. As their conjunction, the "Combined" specification benefits from the sensitivity of all three specifications, and still does not falsely report anomalies at any point, in accordance with Corollary 2.5.1.

In summary, our experimental results demonstrate that our theoretical findings translate into practice with reasonable effectiveness. For the simple PSTL template φ_1, the simulation runtime is negligible for our simulator, and even the more complex formula φ_2 requires only a few hours of simulation runtime in order to require enough traces for our guarantee. Without fail, our specifications show to be conservative: none of our specifications ever reports nominal system behaviour as anomalous. Table 4.3 and the respective figures Figures 4.3 and 4.5 show that the conservative nature of our sample complexity bounds holds for complex specifications. When repeating the experiments the same way with specifications mined from a number of samples of about 10% of the sample complexity Equation (2.24), our anomaly detection setup partially breaks down. Figure 4.3c in particular shows that in nearly all the traces, anomalies are falsely detected during nominal behaviour. With this, we justify our prior assumption for **Q4**: for settings

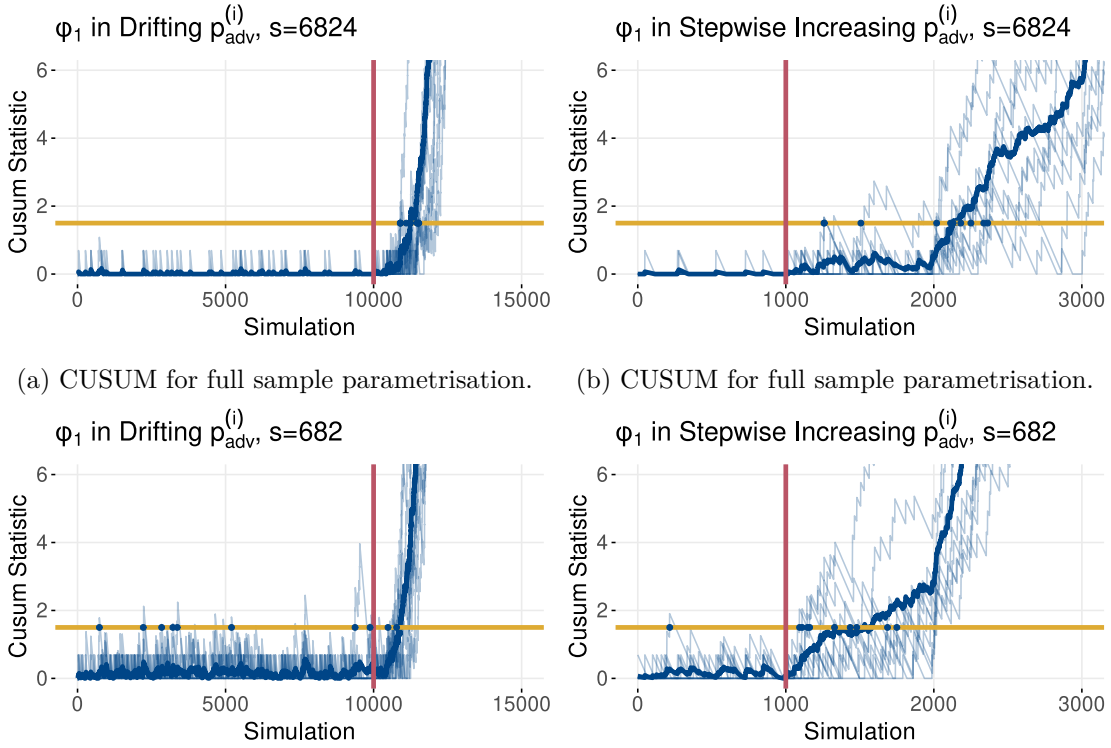
like anomaly detection, we need to know violation probabilities of the specifications we use, otherwise we cannot detect deviations from them reliably with statistical methods. This also shows an advantage of our method in contrast to Chernoff bounds: Estimating this violation-probability directly requires an amount of samples quadratic in ε.

In contrast to φ_1, in Figure 4.5 we can observe that even a fraction of 10% of the required samples is sufficient for our parametrisations to generalise well. Likely reasons for this are that both our sample complexity bound from Equation (2.24) and mainly our VC-dimension bound from Proposition 4.5.3 are not tight. For more complex formulas, these bounds, while still practically feasible, are conservative. Furthermore, to address **Q3**, we acknowledge that the use of full-length, individual traces is statistically inefficient in most cases. For a specification like φ_1, we reduce a full simulation of 500 instants in time to only one data point. Realistically, we could likely consider multiple instants in the same simulation trace as independently sampled from the stationary distribution of the system, and thus obtain the required sample size from a much smaller number of simulation traces.

4.8 Summary, Limitations, and Future Work for CPS Verification

We conclude this chapter with a summary of our results and briefly mention possible future directions for this procedure for CPS verification. While our NN setting was confined to one specific property, in this chapter, we utilised STL as a specification language to allow our results to generalise. We showed that, without *any* assumption on the traces emitted by the system, most specification templates have unbounded VC dimension. This prevents us from providing a sample complexity bound in the general case. With our assumption of bounded variability, as defined in Definition 4.4.1, we can extend our results to *any* PSTL formula, by providing a general VC-dimension bound, parametrised by the allowed signal variability and the number of free parameters. As our experiments illustrated, the sample complexities required by our certification procedure are low enough for practical feasibility for actual, albeit small CPS simulations, like the CartPole problem, with a DQN agent. Our method is useful for specification mining settings, as it certifies *any parametrisation consistent with the obtained sample*. To the best of our knowledge, no existing method can provide comparable probabilistic bounds for mined specifications. We further believe anomaly detection is a useful application of our method, as our generalisation guarantees provide the required prior knowledge for changepoint detection methods.

For more complex specifications, we noticed the sample complexity of our procedure is overly conservative. Besides the slackness of our bounds, this is likely caused by reducing each sampled trace to an individual data point. Future work could investigate settings where relax i.i.d. sampling assumptions and instead sample from, e.g., hidden Markov models, or other models that are better suited to utilise the time-dependent nature of CPS. In addition to this, our bounded variability assumption, together with



(a) CUSUM for full sample parametrisation.

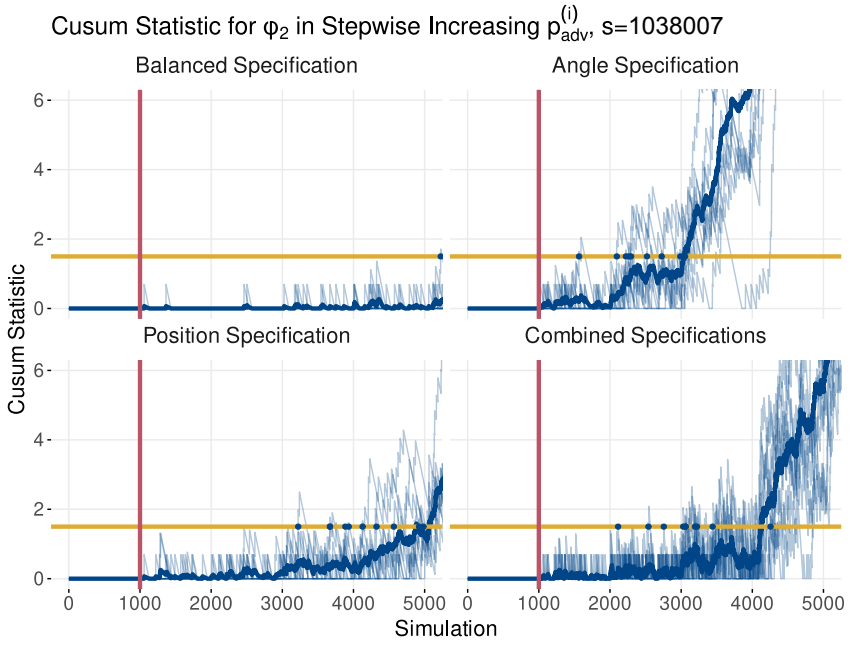
(b) CUSUM for full sample parametrisation.

(c) CUSUM for 10% sample parametrisation.

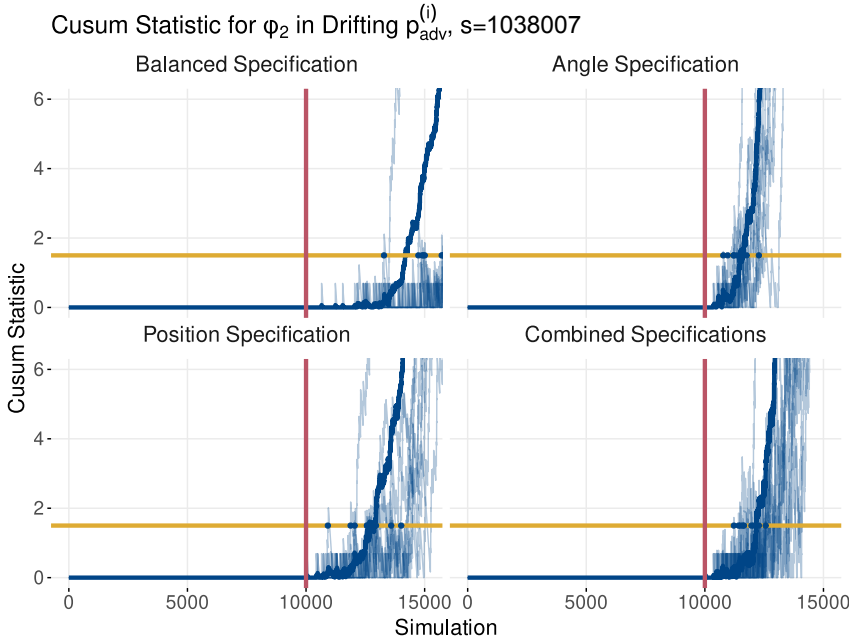
(d) CUSUM for 10% sample parametrisation.

Figure 4.3: CUSUM statistics for different parametrisations of φ_1. The yellow line is the anomaly detection threshold h, the vertical red line marks the beginning of anomalous behaviour. The CUSUM should first cross the yellow line soon after anomalous behaviour begins.

the VC dimension bound we give, can certainly be improved with more involved analysis of the particular formulas. The symbolic monitoring method by Waga et al. [2019] together with a detailed standard translation process from STL to FO[<,+] can certainly drastically reduce the obtained bound. Combining both of these aspects, much lower sample complexities might be possible to obtain guarantees for complex PSTL formulas.

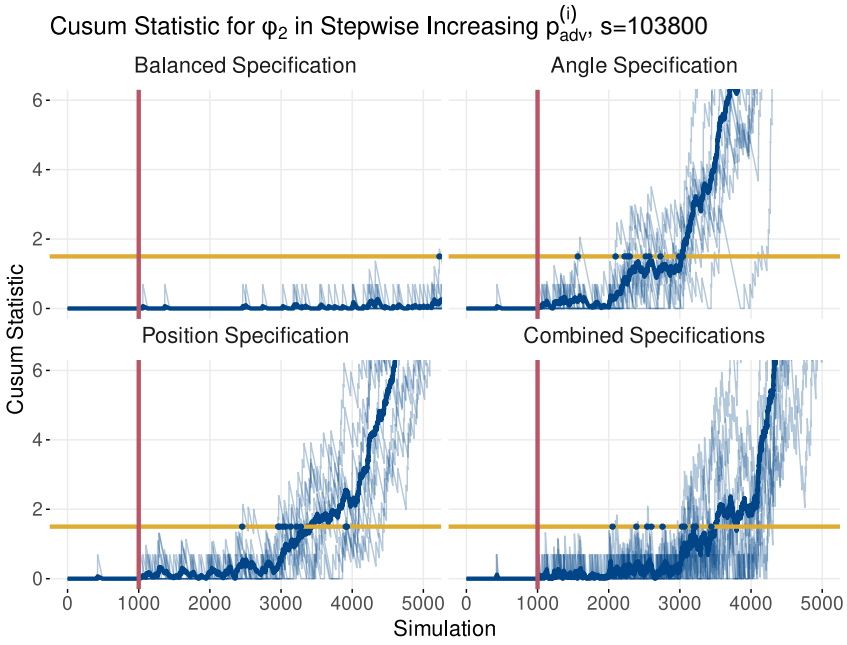


(a) CUSUM for full sample parametrisation for stepwise increasing p_{adv}.

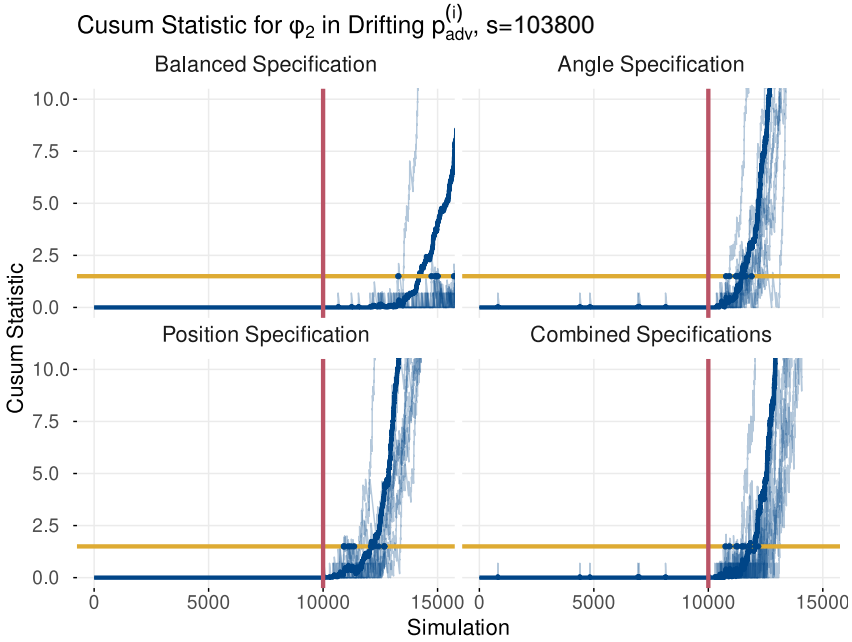


(b) CUSUM for full sample parametrisation for drifting p_{adv}.

Figure 4.4: CUSUM statistics for different parametrisations of φ_2 from the full sample. The yellow line is the anomaly detection threshold h, the vertical red line marks the beginning of anomalous behaviour. The CUSUM should first cross the yellow line soon after anomalous behaviour begins.



(a) CUSUM for 10% sample parametrisation for stepwise increasing p_{adv}.



(b) CUSUM for 10% sample parametrisation for drifting p_{adv}.

Figure 4.5: CUSUM statistics for different parametrisations of φ_2 mined from only 10% of the sample size $s(\varepsilon, \delta, d)$. The yellow line is the anomaly detection threshold h, the vertical red line marks the beginning of anomalous behaviour. The CUSUM should first cross the yellow line soon after anomalous behaviour begins.

Table 4.2: Summary of the CPS experimental results. For each parametrisation, in the two experimental setups, the simulation index of the first alarm, i.e., the smallest index i where $C_i > h$, as well as the corresponding value p_{adv} are reported. The last column reports how often the anomaly was detected before the first failure of the system.

Setup	Specification		First Alarm			$p_{\text{adv}} \cdot 10^3$			Failure Prevented
			min	mean	max	min	mean	max	
Drift	φ_1	Optimal	10899	11228.3	11527	8.99	12.29	15.27	10/10
	φ_2	Balanced	13286	15222.9	16137	32.86	52.23	61.14	1/10
		Angle	10778	11420	12276	7.78	14.20	22.76	10/10
		Position	10919	12599.4	14011	9.19	26.00	40.11	10/10
		Combined	11223	11841.7	12563	12.23	18.42	25.63	9/10
Steps	φ_1	Optimal	1258	1942	2364	5	8.5	10	9/10
	φ_2	Balanced	5226	6864.4	7784	25	32	35	2/10
		Angle	1565	2477.6	3055	5	10.5	15	9/10
		Position	3223	4130.2	4983	15	17.5	20	8/10
		Combined	2114	3067.7	4256	10	14	20	9/10

Table 4.3: Summary of the CPS experimental results for the parametrisations mined from only 10% of the sample size $s(\varepsilon, \delta, d)$. For each parametrisation, in the two experimental setups, the simulation index of the first alarm, i.e., the smallest index i where $C_i > h$, as well as the corresponding value p_{adv} are reported. The last column reports how often the anomaly was detected before the first failure of the system. Bold font indicates false alarm.

Setup	Specification		First Alarm			$p_{\text{adv}} \cdot 10^3$			Failure Prevented
			min	mean	max	min	mean	max	
Drift	φ_1	Optimal	**738**	**5810.5**	10778	**0**	1.2	7.78	10/10
	φ_2	Balanced	13286	15222.9	16137	32.86	52.22	61.37	1/10
		Angle	10778	11358.2	11886	7.78	13.58	18.86	10/10
		Position	10908	11757	12676	9.08	17.57	26.76	10/10
		Combined	10750	11476.4	12176	7.50	14.76	21.76	9/10
Steps	φ_1	Optimal	**216**	1240	1750	**0**	4.5	5	9/10
	φ_2	Balanced	5226	6864.4	7784	25	32	35	2/10
		Angle	1565	2429.2	3044	5	1	15	9/10
		Position	2454	3195.5	3925	10	13.5	15	9/10
		Combined	2054	2828.9	3442	10	12.5	15	9/10

CHAPTER 5

Discussion

We conclude this thesis with a short summary of our results in the context of our initial research questions. For detailed discussions about the individual chapters, we refer to their respective discussion sections.

5.1 MAIN-RQ

> How many random tests are required to characterise a given black-box system with respect to *a given class of properties*?

We investigate this question in Chapter 2. Our main result is the probabilistic certification procedure in Theorem 2.4.2. After one random sample of observations of size $s(\varepsilon, \delta, d)$, Theorem 2.4.2 allows us to certify all properties in our class of properties that are consistent with the sample. This means that we know with high probability $(1 - \delta)$ that *all* properties that were valid in our random sample will be true in any future observation with probability at least $1 - \varepsilon$, for a class of properties of Vapnik-Chervonenkis (VC) dimension d. We introduce the construct of quality transformation in Lemma 2.4.1, as a tool to obtain low VC dimension bounds for interesting properties in the following chapters.

In contrast to the common approach of estimating the probability of an individual given specification, where Chernoff bounds or similar concentration inequalities are utilised, we base our guarantees on ε-nets. This results in a required sample size s that scales better with the desired error probability ε than methods using concentration inequalities. The exact number of samples $s(\varepsilon, \delta, d)$ required to obtain our guarantees is given by Equation (2.24).

5.2 NN-RQ1

> How many samples do we need to decide for all levels of confidence and robustness whether the neural network can be *both confident* and *non-robust*?

We investigate this question in Chapter 3, and answer it with Lemma 3.3.1. The required number of samples follows from our general result in Equation (2.24). The surprising aspect of our result is that certifying confidence-based NN robustness has a constant VC dimension of 2. This means that, independent of the architecture of the NN, the dimensionality of its data or even the precise definition of robustness, we require the same number of data points to give robustness guarantees.

This high degree of abstraction from the specific notion of NN robustness allows us to perform local robustness tests with either formal or heuristic methods, and extend their individual local statements to a probabilistic global guarantee about the network.

5.3 NN-RQ2

> How can we obtain sharp lower bounds for NN robustness, conditioned on the prediction confidence, that generalize to unseen data?

As the bounds we obtain from Lemma 3.3.1 are conjunctive, they will vacuously certify NNs to be robust for high confidence values. We motivate the use of a conditional definition of confidence-based robustness in Equation (3.9) to avoid this and present a method to give these conditional guarantees in Theorem 3.5.1.

This conditional bound, together with Corollary 2.5.1, enables us to construct a mapping from our sample that, for each confidence value, returns the highest robustness radius we can certify. At test time, we can use this mapping to obtain robustness lower bounds for future predictions that hold with high probability for each new observation.

Our experiments show that the lower bounds we obtain from this mapping characterise the NN well and do generalise from seen to unseen data in practice, even when i.i.d. assumptions might not be fully met. Figures 3.2 and 3.3 illustrate this empirical sharpness.

5.4 CPS-RQ1

> For *any given STL specification template*, how many simulation traces of real-time CPSs do we need to certify that all specifications we can mine are valid?

In order to answer this question, we dedicate Chapter 4 to obtaining VC dimension bounds for parametrised signal temporal logic (PSTL) specifications, which would then

allow us to use Theorem 2.4.2. Our results are mixed. Without any assumption on the system we investigate, we can give bounds for some structurally simple classes of STL templates. For many cases, however, we can show that the continuous time semantics of STL lead to unbounded VC dimension. In this case, we cannot apply our verification procedure.

If we assume that our system has a bounded variability as in Definition 4.4.1, however, we can overcome this issue. This definition is comparable to existing definitions of "well-behaved" signals, and allows us to give a VC-dimension bound for *any parametrised STL formula* in Proposition 4.5.3. With this bound, we can extend our procedure to give guarantees for any CPS specification expressible in the language, via Theorem 4.6.1.

We illustrate the practicability of our method with experiments in specification mining and anomaly detection. Our experimental results show that the sample complexity we require is feasible in practical settings, with only a few thousand simulation traces required to obtain high-confidence statements for simple specifications. Because we can quantify the uncertainty of specifications we mined, we can readily combine them with existing tools for anomaly detection. Changes in the system resulted in more frequent violations of our guarantees than expected, which can be detected as anomalous behaviour.

Overview of Generative AI Tools Used

The Grammarly browser plugin has been used to detect individual spelling mistakes and suggest changes for language style.

High-level feedback from Chat-GPT variants 4o, o3 and o4-mini was considered for the improvement of text passages. The models were asked to provide feedback on existing text only, and their output was not directly used to change the manuscript.

List of Figures

2.1 Example for an ε-net. Figure 2.1a illustrates a range space. Circles with probabilities larger than or equal to ε are tinted blue. The set of points in Figure 2.1b intersects all blue circles. 11

2.2 Example for probabilistic inference with an ε-net. Figure 2.2a illustrates a range space with an ε-net. Circles with probabilities larger than or equal to ε are tinted blue. Figure 2.2b shows a different set of circles. With the ε-net, we can infer that all empty circles have a probability mass smaller ε. The dashed blue circles intersect the ε-net, so no statement about them can be made. 12

2.3 Different sampling-based inference methods with their associated sample complexities. The relative costs in terms of sample sizes are illustrated row- and column-wise. The relationships between bounds mainly serve for intuition: marginally tighter bounds can be obtained for ε-nets and ε-samples [Mitzenmacher and Upfal, 2017, Exercise 14.11]. 15

2.4 An i.i.d. sample at a shooting range with different range spaces. Figure 2.4a shows an i.i.d. sample of 12 shots. Figure 2.4b shows a range of ✦ and ✕ shots in a specific area. Figure 2.4c shows one of the ranges in $\mathcal{R} = \{\text{points} \geq x \vee \text{wind} = w : x \in [1, 10], w \in \{✦, ●, ✕\}\}$. 16

3.1 Construction of lower bounds from an ε-net $q(N)$, depicted in Figure 3.1a. Every empty range $R(\rho, \kappa)$ has a probability smaller than ε. Figure 3.1b shows a cover constructed from the union of all empty ranges. The combined probability mass can be bound by the number of highlighted data points defining the lower envelope. 31

3.2 Scatter plots of the two test datasets D_{test}, with $|D_{\text{test}}| = 10000$, in the quality space $\mathcal{Q}$ with PGD robustness oracles. The networks on the left are trained with standard methods, the right networks are trained robustly with TRADES. The red lines depict the lower bound obtained from the validation sample N. The dashed yellow lines depict $\kappa_{\max}$, the threshold above which M is undefined. 38

3.3 Scatter plots of the MNIST test datasets D_{test}, with $|D_{\text{test}}| = 10000$, in the quality space $\mathcal{Q}$ using formal robustness oracles. The networks on the left are trained with standard methods, the right networks are trained robustly with TRADES. The red lines depict the lower bound obtained from the validation sample N. The dashed yellow lines depict $\kappa_{\max}$, the threshold above which M is undefined. 39

4.1 Visual Examples of shattered sets of signals of arbitrary size for two PSTL formulas. Figure 4.1a depicts two parametrisations of $\mathbf{F}_{[0,\tau]}g(\mathbf{w}) < \alpha$. A signal satisfies the formula if it touches the corresponding box at any point. Figure 4.1b depicts one parametrisations of $g_1(\mathbf{w}) < \alpha_1 \mathbf{U} g_2(\mathbf{w}) < \alpha_2$. The dash length of the signals indicates time, with later times resulting in shorter dashes. A signal satisfies the formula if it stays above the dotted blue line (α_1) until it is to the right of the solid blue line (α_2) In the depicted set of signals, this corresponds to touching the green box. 54

4.2 Illustration of the `CartPole-v1` environment and the specification templates we use in our experiments. Figure 4.2a defines bounding boxes the agent must stay in for the entirety of the simulation. Figure 4.2b defines two sets of bounding boxes. If the agent leaves an acceptable state, i.e., the orange set of boxes, it must return to a stable state inside the green boxes quickly. . . 63

4.3 CUSUM statistics for different parametrisations of φ_1. The yellow line is the anomaly detection threshold h, the vertical red line marks the beginning of anomalous behaviour. The CUSUM should first cross the yellow line soon after anomalous behaviour begins. 67

4.4 CUSUM statistics for different parametrisations of φ_2 from the full sample. The yellow line is the anomaly detection threshold h, the vertical red line marks the beginning of anomalous behaviour. The CUSUM should first cross the yellow line soon after anomalous behaviour begins. 68

4.5 CUSUM statistics for different parametrisations of φ_2 mined from only 10% of the sample size $s(\varepsilon, \delta, d)$. The yellow line is the anomaly detection threshold h, the vertical red line marks the beginning of anomalous behaviour. The CUSUM should first cross the yellow line soon after anomalous behaviour begins. 69

List of Tables

3.1 Summary results for all experiments. We report *worst* results aggregated over 3 random seeds and over the different hyperparameter values used for the TRADES adversarial training. For each experiment, we report the values of $\hat{p}$ and n_c, where bold numbers denote that the estimators are consistent with our guarantees *for all* the $\kappa \leq \kappa_{\max}$, *for all* the runs considered. Moreover, we report the number of individual "good runs" that are consistent with our guarantees when considering the worst-case $\hat{p}$. More extensive results are available in Appendix B. 40

4.1 Required sample complexities $s(\varepsilon, \delta, \mathrm{VC}(\mathcal{W}, \varphi))$ as per Equation (2.24) for different PSTL specifications from existing literature given their associated variability and a fixed $\delta = 0.01$. 61

4.2 Summary of the CPS experimental results. For each parametrisation, in the two experimental setups, the simulation index of the first alarm, i.e., the smallest index i where $C_i > h$, as well as the corresponding value p_{adv} are reported. The last column reports how often the anomaly was detected before the first failure of the system. 70

4.3 Summary of the CPS experimental results for the parametrisations mined from only 10% of the sample size $s(\varepsilon, \delta, d)$. For each parametrisation, in the two experimental setups, the simulation index of the first alarm, i.e., the smallest index i where $C_i > h$, as well as the corresponding value p_{adv} are reported. The last column reports how often the anomaly was detected before the first failure of the system. Bold font indicates false alarm. 70

1 Results for the MNIST dataset using PGD as a local robustness oracle. . 100
2 Results for the MNIST dataset using Marabou as a local robustness oracle. 101
3 Results for the MNIST dataset using LiRPA as a local robustness oracle. 101
4 Results for the CIFAR-10 dataset using PGD as a local robustness oracle. 101
5 Mined parameter values for the experiments in Section 4.7.3 using the full samples for the respective specification. Bold parameter values were chosen manually. 102
6 Mined parameter values for the experiments in Section 4.7.3 using 10% of the samples for the respective specification. Bold parameter values were chosen manually. 102

List of Algorithms

3.1 Obtain ρ-κ-mapping 32

Symbols

$\mathcal{X}$ ground set, input space. 5

$\mathbf{x}$ data point $\mathbf{x} \in \mathcal{X}$. 5

f black-box system. 5

$\mathcal{D}$ probability distribution over $\mathcal{X}$ generating observations from f. 5

Pr probability under $\mathcal{D}$. 5

X random variable for a data point sampled from $\mathcal{D}$. 5

$\sim$ "distributed according to", e.g., $X \sim \mathcal{D}$. 5

N finite set, sample of points from $\mathcal{D}$, e.g., $N \sim \mathcal{D}^n$. 5

ε tolerated error in estimates (of probabilities). 5

δ tolerated failure probability of a sampling-based procedure. 7

$\mathcal{O}$ Landau symbol (big-O): an asymptotic upper bound. 7

$\mathcal{R}$ set of ranges (subsets) of some space $\mathcal{X}$, e.g., $R \in \mathcal{R} : R \subset \mathcal{X}$. 9

$(\mathcal{X}, \mathcal{R})$ range space of ranges $\mathcal{R}$ of the ground space $\mathcal{X}$. 9

VC Vapnik-Chervonenkis dimension of a range space. 9

s (smallest permissible) size of a sample. 13

$\mathcal{Q}$ quality space, some projective space to define specifications. 17

q quality transformation, a function $q : \mathcal{X} \to \mathcal{Q}$. 17

$\|\cdot\|$ some norm in metric space $\mathcal{X}$. 22

$\mathbf{class}_f$ predicted class of NN. 22

$\mathbf{conf}_f$ softmax confidence of NN. 24

$\propto$ "proportional to". 24

κ NN softmax confidence value or threshold. 25

$\mathbf{rob}_f$ robustness oracle for NN. 25

ρ NN robustness (radius) value or threshold. 25

$R(\rho, \kappa)$ range for NN verification containing points that are κ-confident but less than ρ robust. 26

$p_{\min}$ probability parameter, for the fraction of NN predictions to abstain. 29

$\kappa_{\max}$ largest confidence in sample for which a guarantee can be given. 30

M mapping from confidence value to robustness lower bound. 30

D_{test} finite test dataset in NN experiments. 35

n_c number of non-robust data points in a dataset. 35

p_κ fraction of non-robust data points in a dataset with confidence at least κ. 35

$\hat{p}$ maximum fraction of non-robust data points in a dataset over values of κ. 35

$\mathbb{T}$ time domain, for some $t_{\max} \in \mathbb{R}_+ : \mathbb{T} = [0, t_{\max}] \subset \mathbb{R}_{\geq 0}$. 44

$\mathbf{w}$ a signal, $\mathbf{w} : \mathbb{T} \to \mathcal{X}$. 44

$\mathbf{w}[t]$ signal $\mathbf{w}$ at instant $t \in \mathbb{T}$, $\mathbf{w}[t] \in \mathcal{X}$. 44

φ (P)STL formula. 44

I time interval I in $\mathbb{Q}_{\geq 0} \cup \{\infty\}$. 44

$(\mathbf{w}, t) \models \varphi$ φ is true for signal $\mathbf{w}$, at instant t. 44

$\mathbf{U}_I$ STL operator until, bound by interval I. As shorthand $\mathbf{U}_{[0,\infty)} = \mathbf{U}$. 44

$\mathbf{S}_I$ STL operator since, bound by interval I. As shorthand $\mathbf{S}_{[0,\infty)} = \mathbf{S}$. 44

$\mathbf{F}_I$ STL operator finally, bound by interval I. As shorthand $\mathbf{F}_{[0,\infty)} = \mathbf{F}$. 45

$\mathbf{G}_I$ STL operator globally, bound by time interval I. As shorthand $\mathbf{G}_{[0,\infty)} = \mathbf{G}$. 45

$\oplus$ Minkowski sum: $t \oplus I = \{t + a : a \in I\}$. 45

$\ominus$ Minkowski difference: $t \ominus I = \{t - a : a \in I\}$. 45

A PSTL magnitude parameters $A = \{\alpha_1, \ldots, \alpha_m\}$. 45

T PSTL timing parameters $T = \{\tau_1, \ldots, \tau_k\}$. 45

$\varphi_{\mathbf{v}}$ PSTL formula φ with valuation $\mathbf{v} \in \mathbb{Q}^{m+k}$. 45

$\mathbb{1}_{(\mathbf{w}_i,0)\models\varphi}$ indicator variable for the random event $(\mathbf{w}_i, 0) \models \varphi$, for some $\mathbf{w}_i \in N$. 48

h threshold for CUSUM anomaly detection. 48, 63

$\mathbb{I}$ set of indices. 50

r_i the ith prime number. 49

$\mathcal{I}_{\mathbf{w}}$ interval cover of signal $\mathbf{w}$ over time domain $\mathbb{T}$. 55

$\zeta = \zeta(\mathbf{w}, \varphi)$ variability of signal $\mathbf{w}$ with respect to PSTL formula φ. 55

ST_t standard translation procedure from STL to FO[<,+]. 57

Bibliography

Yaa Takyiwaa Acquaah and Kaushik Roy. Normal-only anomaly detection in environmental sensors in CPS: A comprehensive review. *IEEE Access*, 12:191086–191107, 2024. doi: 10.1109/ACCESS.2024.3513714. URL https://doi.org/10.1109/ACCESS.2024.3513714.

C. J. Adcock. Sample size determination: a review. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 46(2):261–283, July 1997. ISSN 1467-9884. doi: 10.1111/1467-9884.00082. URL http://dx.doi.org/10.1111/1467-9884.00082.

Gul Agha and Karl Palmskog. A survey of statistical model checking. *ACM Trans. Model. Comput. Simul.*, 28(1):6:1–6:39, 2018. doi: 10.1145/3158668. URL https://doi.org/10.1145/3158668.

Bernhard K. Aichernig, Florian Lorber, and Dejan Nickovic. Time for mutants - model-based mutation testing with timed automata. In Margus Veanes and Luca Viganò, editors, *Tests and Proofs - 7th International Conference, TAP@STAF 2013, Budapest, Hungary, June 16-20, 2013. Proceedings*, volume 7942 of *Lecture Notes in Computer Science*, pages 20–38. Springer, 2013. doi: 10.1007/978-3-642-38916-0_2. URL https://doi.org/10.1007/978-3-642-38916-0_2.

Jason Ansel, Edward Z. Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, and Evgeni Burovski et al. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In Rajiv Gupta, Nael B. Abu-Ghazaleh, Madan Musuvathi, and Dan Tsafrir, editors, *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2, ASPLOS 2024, La Jolla, CA, USA, 27 April 2024- 1 May 2024*, pages 929–947. ACM, 2024. doi: 10.1145/3620665.3640366. URL https://doi.org/10.1145/3620665.3640366.

Anagha Athavale, Ezio Bartocci, Maria Christakis, Matteo Maffei, Dejan Nickovic, and Georg Weissenbacher. Verifying global two-safety properties in neural networks with confidence. In Arie Gurfinkel and Vijay Ganesh, editors, *Computer Aided Verification - 36th International Conference, CAV 2024, Montreal, QC, Canada, July 24-27, 2024, Proceedings, Part II*, volume 14682 of *Lecture Notes in Computer Science*, pages 329–351. Springer, 2024. doi: 10.1007/978-3-031-65630-9_17. URL https://doi.org/10.1007/978-3-031-65630-9_17.

Teodora Baluta, Zheng Leong Chua, Kuldeep S. Meel, and Prateek Saxena. Scalable quantitative verification for deep neural networks. In *43rd IEEE/ACM International Conference on Software Engineering, ICSE 2021, Madrid, Spain, 22-30 May 2021*, pages 312–323. IEEE, 2021. doi: 10.1109/ICSE43902.2021.00039. URL https://doi.org/10.1109/ICSE43902.2021.00039.

Ezio Bartocci, Cristinel Mateis, Eleonora Nesterini, and Dejan Nickovic. Survey on mining signal temporal logic specifications. *Inf. Comput.*, 289(Part):104957, 2022. doi: 10.1016/J.IC.2022.104957. URL https://doi.org/10.1016/j.ic.2022.104957.

Patrick Blackburn, Maarten De Rijke, and Yde Venema. *Modal logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 2001. ISBN 9781107050884. doi: 10.1017/CBO9781107050884. URL https://doi.org/10.1017/CBO9781107050884.

Christopher Brix, Stanley Bak, Taylor T. Johnson, and Haoze Wu. The fifth international verification of neural networks competition (VNN-COMP 2024): Summary and results. *CoRR*, abs/2412.19985, 2024. doi: 10.48550/ARXIV.2412.19985. URL https://doi.org/10.48550/arXiv.2412.19985.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *CoRR*, abs/1606.01540, 2016. URL http://arxiv.org/abs/1606.01540.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57. IEEE Computer Society, 2017a. doi: 10.1109/SP.2017.49. URL https://doi.org/10.1109/SP.2017.49.

Nicholas Carlini and David A. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In Bhavani Thuraisingham, Battista Biggio, David Mandell Freeman, Brad Miller, and Arunesh Sinha, editors, *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, pages 3–14. ACM, 2017b. doi: 10.1145/3128572.3140444. URL https://doi.org/10.1145/3128572.3140444.

Jeremy Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 1310–1320. PMLR, 2019. URL http://proceedings.mlr.press/v97/cohen19c.html.

Li Deng. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process. Mag.*, 29(6):141–142, 2012. doi: 10.1109/MSP.2012.2211477. URL https://doi.org/10.1109/MSP.2012.2211477.

Peter Gustav Lejeune Dirichlet. Sur la convergence des séries trigonométriques qui servent à représenter une fonction arbitraire entre des limites données. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 1829(4):157–169, January 1829. ISSN 1435-5345. doi: 10.1515/crll.1829.4.157. URL http://dx.doi.org/10.1515/crll.1829.4.157.

A. Dvoretzky, J. Kiefer, and J. Wolfowitz. *Asymptotic Minimax Character of the Sample Distribution Function and of the Classical Multinomial Estimator*, page 182–209. Springer US, 1985. ISBN 9781461385059. doi: 10.1007/978-1-4613-8505-9_17. URL http://dx.doi.org/10.1007/978-1-4613-8505-9_17.

Paul Goldberg and Mark Jerrum. Bounding the vapnik-chervonenkis dimension of concept classes parameterized by real numbers. In *Proceedings of the sixth annual conference on Computational learning theory - COLT '93*, COLT '93, page 361–369. ACM Press, 1993. doi: 10.1145/168304.168377. URL http://dx.doi.org/10.1145/168304.168377.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6572.

D Haussler and E Welzl. Epsilon-nets and simplex range queries. In *Proceedings of the second annual symposium on Computational geometry - SCG '86*, SCG '86, page 61–71. ACM Press, 1986. doi: 10.1145/10515.10522. URL http://dx.doi.org/10.1145/10515.10522.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Patrick Indri, Peter Blohm, Anagha Athavale, Ezio Bartocci, Georg Weissenbacher, Matteo Maffei, Dejan Nickovic, Thomas Gärtner, and Sagar Malhotra. Distillation based robustness verification with pac guarantees. In *Proceedings of the ICML 2024 Workshop on Next Generation of AI Safety*, Vienna, Austria, July 2024. PMLR. URL https://openreview.net/forum?id=vflefS3lmB.

Susmit Jha, Ashish Tiwari, Sanjit A. Seshia, Tuhin Sahai, and Natarajan Shankar. Telex: Passive STL learning using only positive examples. In Shuvendu K. Lahiri and Giles Reger, editors, *Runtime Verification - 17th International Conference, RV 2017, Seattle, WA, USA, September 13-16, 2017, Proceedings*, volume 10548 of *Lecture Notes in Computer Science*, pages 208–224. Springer, 2017. doi: 10.1007/978-3-319-67531-2_13. URL https://doi.org/10.1007/978-3-319-67531-2_13.

Austin Jones, Zhaodan Kong, and Calin Belta. Anomaly detection in cyber-physical systems: A formal methods approach. In *53rd IEEE Conference on Decision and*

Control, CDC 2014, Los Angeles, CA, USA, December 15-17, 2014, pages 848–853. IEEE, 2014. doi: 10.1109/CDC.2014.7039487. URL https://doi.org/10.1109/CDC.2014.7039487.

Anan Kabaha and Dana Drachsler-Cohen. Verification of neural networks' global robustness. *Proc. ACM Program. Lang.*, 8(OOPSLA1):1010–1039, 2024. doi: 10.1145/3649847. URL https://doi.org/10.1145/3649847.

Nidhi Kalra and Susan M. Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, 2016. ISSN 0965-8564. doi: https://doi.org/10.1016/j.tra.2016.09.010. URL https://www.sciencedirect.com/science/article/pii/S0965856416302129.

Johan Anthony Willem Kamp. *Tense Logic and the Theory of Linear Order*. Ph.d. dissertation, University of California, Los Angeles, Los Angeles, CA, 1968. URL https://search.proquest.com/openview/408039eb4ed228dc4cba3fe7e1774163/1?pq-origsite=gscholar&cbl=18750&diss=y.

Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In Rupak Majumdar and Viktor Kuncak, editors, *Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I*, volume 10426 of *Lecture Notes in Computer Science*, pages 97–117. Springer, 2017. doi: 10.1007/978-3-319-63387-9_5. URL https://doi.org/10.1007/978-3-319-63387-9_5.

Guy Katz, Derek A. Huang, Duligur Ibeling, Kyle Julian, Christopher Lazarus, Rachel Lim, Parth Shah, Shantanu Thakoor, Haoze Wu, Aleksandar Zeljic, David L. Dill, Mykel J. Kochenderfer, and Clark W. Barrett. The marabou framework for verification and analysis of deep neural networks. In Isil Dillig and Serdar Tasiran, editors, *Computer Aided Verification - 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part I*, volume 11561 of *Lecture Notes in Computer Science*, pages 443–452. Springer, 2019. doi: 10.1007/978-3-030-25540-4_26. URL https://doi.org/10.1007/978-3-030-25540-4_26.

Hoki Kim, Jinseong Park, Yujin Choi, and Jaewook Lee. Fantastic robustness measures: The secrets of robust generalization. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/98a5c0470e57d518ade4e56c6ee0b363-Abstract-Conference.html.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario, Canada, April 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf. Technical Report.

Axel Legay, Anna Lukina, Louis-Marie Traonouez, Junxing Yang, Scott A. Smolka, and Radu Grosu. Statistical model checking. In Bernhard Steffen and Gerhard J. Woeginger, editors, *Computing and Software Science - State of the Art and Perspectives*, volume 10000 of *Lecture Notes in Computer Science*, pages 478–504. Springer, 2019. doi: 10.1007/978-3-319-91908-9_23. URL https://doi.org/10.1007/978-3-319-91908-9_23.

Klas Leino, Zifan Wang, and Matt Fredrikson. Globally-robust neural networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 6212–6222. PMLR, 2021. URL http://proceedings.mlr.press/v139/leino21a.html.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Thibault Maho, Teddy Furon, and Erwan Le Merrer. Randomized smoothing under attack: How good is it in practice? In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*, pages 3014–3018. IEEE, 2022. doi: 10.1109/ICASSP43922.2022.9746293. URL https://doi.org/10.1109/ICASSP43922.2022.9746293.

Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. In Yassine Lakhnech and Sergio Yovine, editors, *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, Joint International Conferences on Formal Modelling and Analysis of Timed Systems, FORMATS 2004 and Formal Techniques in Real-Time and Fault-Tolerant Systems, FTRTFT 2004, Grenoble, France, September 22-24, 2004, Proceedings*, volume 3253 of *Lecture Notes in Computer Science*, pages 152–166. Springer, 2004. doi: 10.1007/978-3-540-30206-3_12. URL https://doi.org/10.1007/978-3-540-30206-3_12.

P. Massart. The tight constant in the dvoretzky-kiefer-wolfowitz inequality. *The Annals of Probability*, 18(3), July 1990. ISSN 0091-1798. doi: 10.1214/aop/1176990746. URL http://dx.doi.org/10.1214/aop/1176990746.

Mark Huasong Meng, Guangdong Bai, Sin Gee Teo, Zhe Hou, Yan Xiao, Yun Lin, and Jin Song Dong. Adversarial robustness of deep neural networks: A survey from a formal verification perspective. *IEEE Transactions on Dependable and Secure Computing*, pages 1–1, 2022. doi: 10.1109/TDSC.2022.3179131. URL https://doi.org/10.1109/TDSC.2022.3179131.

Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press,

USA, 2nd edition, 2017. ISBN 110715488X. URL https://dl.acm.org/doi/abs/10.5555/3134214.

Douglas C Montgomery. *Introduction to statistical quality control*. John Wiley & Sons, Hoboken, NJ, 8th edition, June 2020. ISBN 978-1119723097. URL https://www.wiley.com/en-us/Introduction+to+Statistical+Quality+Control%2C+8th+Edition-p-9781119399308.

Pieter J. Mosterman. An overview of hybrid simulation phenomena and their support by simulation packages. In Frits W. Vaandrager and Jan H. van Schuppen, editors, *Hybrid Systems: Computation and Control, Second International Workshop, HSCC'99, Berg en Dal, The Netherlands, March 29-31, 1999, Proceedings*, volume 1569 of *Lecture Notes in Computer Science*, pages 165–177. Springer, 1999. doi: 10.1007/3-540-48983-5_17. URL https://doi.org/10.1007/3-540-48983-5_17.

Michael Naaman. On the tight constant in the multivariate dvoretzky–kiefer–wolfowitz inequality. *Statistics & Probability Letters*, 173:109088, June 2021. ISSN 0167-7152. doi: 10.1016/j.spl.2021.109088. URL http://dx.doi.org/10.1016/j.spl.2021.109088.

Daniele Nicoletti, Samuele Germiniani, and Graziano Pravadelli. Mining signal temporal logic specifications for hybrid systems. In *Forum on Specification & Design Languages, FDL 2024, Stockholm, Sweden, September 4-6, 2024*, pages 1–8. IEEE, 2024. doi: 10.1109/FDL63219.2024.10673843. URL https://doi.org/10.1109/FDL63219.2024.10673843.

Habeeb Olufowobi, Uchenna Ezeobi, Eric Muhati, Gaylon Robinson, Clinton Young, Joseph Zambreno, and Gedare Bloom. Anomaly detection approach using adaptive cumulative sum algorithm for controller area network. In Ziming Zhao, Qi Alfred Chen, and Gail-Joon Ahn, editors, *Proceedings of the ACM Workshop on Automotive Cybersecurity, AutoSec@CODASPY 2019, Richardson, TX, USA, March 27, 2019*, pages 25–30. ACM, 2019. doi: 10.1145/3309171.3309178. URL https://doi.org/10.1145/3309171.3309178.

E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100, June 1954. ISSN 0006-3444. doi: 10.2307/2333009. URL http://dx.doi.org/10.2307/2333009.

Joe. Sachs and Aristotle. *Aristotle's physics : a guided study / Joe Sachs. [electronic resource]*. Masterworks of discovery. Rutgers University Press, New Brunswick, N.J, 1995. ISBN 0-8135-6620-7.

N Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, July 1972. ISSN 0097-3165. doi: 10.1016/0097-3165(72)90019-2. URL http://dx.doi.org/10.1016/0097-3165(72)90019-2.

Koushik Sen, Mahesh Viswanathan, and Gul Agha. Statistical model checking of black-box probabilistic systems. In Rajeev Alur and Doron A. Peled, editors, *Computer Aided Verification, 16th International Conference, CAV 2004, Boston, MA, USA,*

July 13-17, 2004, Proceedings, volume 3114 of *Lecture Notes in Computer Science*, pages 202–215. Springer, 2004. doi: 10.1007/978-3-540-27813-9_16. URL https://doi.org/10.1007/978-3-540-27813-9_16.

Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, April 1972. ISSN 0030-8730. doi: 10.2140/pjm.1972.41.247. URL http://dx.doi.org/10.2140/pjm.1972.41.247.

Zhouxing Shi, Qirui Jin, Zico Kolter, Suman Jana, Cho-Jui Hsieh, and Huan Zhang. Neural network verification with branch-and-bound for general nonlinearities. *CoRR*, abs/2405.21063, 2024. doi: 10.48550/ARXIV.2405.21063. URL https://doi.org/10.48550/arXiv.2405.21063.

Simplicius. *On Aristotle's "Physics 6"*. Ancient Commentators on Aristotle. Cornell University Press, Ithaca, NY, January 1989. ISBN 9780801422386.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction, 2nd Edition*. MIT Press, 2018. URL http://www.incompleteideas.net/book/the-book-2nd.html.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6199.

Alexander Tartakovsky, I. V. Nikiforov, and M. Basseville. *Sequential analysis: hypothesis testing and changepoint detection*. Monographs on Statistics and Applied Probability. CRC Press, Boca Raton, Florida, online-ausg edition, 2015. ISBN 9781439838204.

Andrew R. Teel, Ricardo G. Sanfelice, and Rafal Goebel. Hybrid control systems. In Robert A. Meyers, editor, *Encyclopedia of Complexity and Systems Science*, pages 4671–4696. Springer, 2009. doi: 10.1007/978-0-387-30440-3_276. URL https://doi.org/10.1007/978-0-387-30440-3_276.

V. N. Vapnik and A. Ya. Chervonenkis. *On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities*, page 11–30. Springer International Publishing, 2015. ISBN 9783319218526. doi: 10.1007/978-3-319-21852-6_3. URL http://dx.doi.org/10.1007/978-3-319-21852-6_3.

Masaki Waga, Étienne André, and Ichiro Hasuo. Symbolic monitoring against specifications parametric in time and data. In Isil Dillig and Serdar Tasiran, editors, *Computer Aided Verification - 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part I*, volume 11561 of *Lecture Notes in Computer Science*, pages 520–539. Springer, 2019. doi: 10.1007/978-3-030-25540-4_30. URL https://doi.org/10.1007/978-3-030-25540-4_30.

A. Wald. *Sequential Tests of Statistical Hypotheses*, pages 256–298. Springer New York, New York, NY, 1992. ISBN 978-1-4612-0919-5. doi: 10.1007/978-1-4612-0919-5_18. URL https://doi.org/10.1007/978-1-4612-0919-5_18.

Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J. Zico Kolter. Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 29909–29921, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/fac7fead96dafceaf80c1daffeae82a4-Abstract.html.

Zhilu Wang, Chao Huang, and Qi Zhu. Efficient global robustness certification of neural networks via interleaving twin-network encoding. In Cristiana Bolchini, Ingrid Verbauwhede, and Elena-Ioana Vatajelu, editors, *2022 Design, Automation & Test in Europe Conference & Exhibition, DATE 2022, Antwerp, Belgium, March 14-23, 2022*, pages 1087–1092. IEEE, 2022. doi: 10.23919/DATE54114.2022.9774719. URL https://doi.org/10.23919/DATE54114.2022.9774719.

Christopher John Cornish Hellaby Watkins. *Learning from Delayed Rewards*. PhD thesis, University of Cambridge, Cambridge, UK, May 1989. URL https://www.cs.rhul.ac.uk/~chrisw/new_thesis.pdf. Ph.D. thesis.

Haoze Wu, Omri Isac, Aleksandar Zeljic, Teruhiro Tagomori, Matthew L. Daggitt, Wen Kokke, Idan Refaeli, Guy Amir, Kyle Julian, Shahaf Bassan, Pei Huang, Ori Lahav, Min Wu, Min Zhang, Ekaterina Komendantskaya, Guy Katz, and Clark W. Barrett. Marabou 2.0: A versatile formal analyzer of neural networks. In Arie Gurfinkel and Vijay Ganesh, editors, *Computer Aided Verification - 36th International Conference, CAV 2024, Montreal, QC, Canada, July 24-27, 2024, Proceedings, Part II*, volume 14682 of *Lecture Notes in Computer Science*, pages 249–264. Springer, 2024. doi: 10.1007/978-3-031-65630-9_13. URL https://doi.org/10.1007/978-3-031-65630-9_13.

Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/0cbc5671ae26f67871cb914d81ef8fc1-Abstract.html.

Kaidi Xu, Huan Zhang, Shiqi Wang, Yihan Wang, Suman Jana, Xue Lin, and Cho-Jui Hsieh. Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers. In *9th International Conference on Learning*

Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021. URL https://openreview.net/forum?id=nVZtXBI6LNn.

Håkan L. S. Younes. Probabilistic verification for "black-box" systems. In Kousha Etessami and Sriram K. Rajamani, editors, *Computer Aided Verification, 17th International Conference, CAV 2005, Edinburgh, Scotland, UK, July 6-10, 2005, Proceedings*, volume 3576 of *Lecture Notes in Computer Science*, pages 253–265. Springer, 2005. doi: 10.1007/11513988_25. URL https://doi.org/10.1007/11513988_25.

Håkan L. S. Younes and Reid G. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In Ed Brinksma and Kim Guldstrand Larsen, editors, *Computer Aided Verification, 14th International Conference, CAV 2002,Copenhagen, Denmark, July 27-31, 2002, Proceedings*, volume 2404 of *Lecture Notes in Computer Science*, pages 223–235. Springer, 2002. doi: 10.1007/3-540-45657-0_17. URL https://doi.org/10.1007/3-540-45657-0_17.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR, 2019. URL http://proceedings.mlr.press/v97/zhang19p.html.

Huan Zhang, Shiqi Wang, Kaidi Xu, Linyi Li, Bo Li, Suman Jana, Cho-Jui Hsieh, and J. Zico Kolter. General cutting planes for bound-propagation-based neural network verification. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/0b06c8673ebb453e5e468f7743d8f54e-Abstract-Conference.html.

Paolo Zuliani, André Platzer, and Edmund M. Clarke. Bayesian statistical model checking with application to stateflow/simulink verification. *Formal Methods Syst. Des.*, 43 (2):338–367, 2013. doi: 10.1007/S10703-013-0195-3. URL https://doi.org/10.1007/s10703-013-0195-3.

Appendix

A Proofs

Proposition 2.2.4 (ε-nets from i.i.d. samples)**.** *Let $(\mathcal{Q}, \mathcal{R})$ be a range space with VC dimension d and let $\mathcal{D}$ be a probability distribution over $\mathcal{Q}$. For parameters $0 < \varepsilon, \delta < \frac{1}{2}$, an i.i.d. sample from $\mathcal{D}$ of size s is an ε-net for $(\mathcal{Q}, \mathcal{R})$ with probability at least $1 - \delta$ if s satisfies*

$$s \geq \frac{2}{\ln(2)\varepsilon}\left(\ln\frac{1}{\delta} + d\ln(2s) - \ln\left(1 - \exp\left(\frac{-s\varepsilon}{8}\right)\right)\right) \tag{2.23}$$

Proof. We follow the argument of "double sampling" from Mitzenmacher and Upfal [2017, Theorem 14.8]. We first define E_1 as the random event that a sample N of size $|N| = s$ is *not* an ε-net:

$$E_1 = \{\exists R \in \mathcal{R} : (\Pr(X \in R) \geq \varepsilon) \wedge (R \cap N = \emptyset)\} \tag{1}$$

We aim to show $\Pr(E_1) \leq \delta$ for large enough s. We proceed by choosing a second sample T with $|T| = s$ and define E_2 as the event that some range R does *not* intersect N, but has a large intersection with T.

$$E_2 = \left\{\exists R \in \mathcal{R} : (\Pr(X \in R) \geq \varepsilon) \wedge (R \cap N = \emptyset) \wedge \left(|R \cap T| \geq \frac{\varepsilon s}{2}\right)\right\} \tag{2}$$

The idea is that, because $\mathbb{E}(|T \cap R|) = \varepsilon s$, the probability of $|R \cap T| \geq \frac{\varepsilon s}{2}$ should be large, and hence E_1 and E_2 should have similar probability in total.

Mitzenmacher and Upfal [2017] formalize this intuition with the following expression, where they consider some fixed range R', such that $R' \cap N = \emptyset$ and $\Pr(X \in R) \geq \varepsilon$. In particular, as $E_2 \subset E_1$ and consequently $E_2 = E_2 \cap E_1$, we know

$$\frac{\Pr(E_2)}{\Pr(E_1)} = \frac{\Pr(E_1 \cap E_2)}{\Pr(E_1)} = \Pr(E_2 \mid E_1) \geq \Pr\left(|T \cap R'| \geq \frac{\varepsilon s}{2}\right) \tag{3}$$

For some fixed range R' the random variable $S = |T \cap R'|$ follows a binomial distribution, and we can proceed similarly to Example 2.1.3. We recall Theorem 2.1.2 for a relative

error $\mu = \varepsilon s - \frac{\varepsilon s}{2}$

$$\Pr\left(\frac{sp - S}{sp} \geq \mu\right) \leq \exp\left(\frac{-sp\mu^2}{2}\right) \tag{4}$$

$$\Pr\left(\frac{sp - S}{sp} \geq \frac{1}{2}\right) \leq \exp\left(\frac{-sp}{8}\right) \leq \exp\left(\frac{-s\varepsilon}{8}\right) \tag{5}$$

Here, the last inequality uses the fact that $p \geq \varepsilon$. While Mitzenmacher and Upfal [2017] relax this expression to the constant $\frac{1}{2}$ we continue without relaxation in the interest of obtaining tighter bounds. Thus,

$$\frac{\Pr(E_2)}{\Pr(E_1)} = \Pr(E_2 \mid E_1) \geq \Pr\left(|T \cap R'| \geq \frac{\varepsilon s}{2}\right) \geq 1 - \exp\left(\frac{-s\varepsilon}{8}\right) \tag{6}$$

and finally we have $\Pr(E_1) \leq \frac{\Pr(E_2)}{1-\exp\left(\frac{-s\varepsilon}{8}\right)}$. As a next step, we bound the probability $\Pr(E_2)$ with a larger event E_2'. For this, we first consider some specific fixed range R again, with

$$E_R = \{(R \cap N = \emptyset) \wedge (|R \cap T| \geq k)\} \tag{7}$$

We want to show that $\Pr(E_R)$ is small. The intuition here is that both N and T are random samples, but all the $k = \frac{\varepsilon s}{2}$ sampled points that intersect some specific range R belong to T and none to N.

Of the $\binom{2s}{s}$ possible partitions of $N \cup T$, in exactly $\binom{2s-k}{s}$ of them, no element of R is in N. Consequently

$$\Pr(E_R) \leq \Pr(N \cap R = \emptyset \mid |R \cap (N \cup T)| \geq k) \tag{8}$$

$$\leq \frac{\binom{2s-k}{s}}{\binom{2s}{s}} \tag{9}$$

$$= \frac{(2s-k)!s!}{(2s)!(s-k)!} \tag{10}$$

$$= \frac{s(s-1)\cdots(s-k+1)}{(2s)(2s-1)\cdots(2s-k+1)} \tag{11}$$

$$\leq 2^{-k} = 2^{-\varepsilon s/2} \tag{12}$$

The inequality in the last line *does relax* the result, but only very marginally so, as $k \ll s$. We then finally consider the event E_2' via the union bound over all the ranges $R \in \mathcal{R}$, that is

$$E_2' = \{\exists R \in \mathcal{R} : (R \cap N = \emptyset) \wedge (|R \cap T| \geq \frac{s\varepsilon}{2})\} \tag{13}$$

We then use the Sauer-Shelah Lemma [Sauer, 1972, Shelah, 1972] to argue that we can consider at most $(2s)^d$ ranges when projecting $\mathcal{R}$ onto $N \cup T$. By the union bound we have

$$\Pr(E_2') \leq (2s)^d 2^{-s\varepsilon/2} \tag{14}$$

Finally, we arrive at

$$\Pr(E_1) \leq \frac{\Pr(E_2)}{1-\exp\left(\frac{-s\varepsilon}{8}\right)} \leq \frac{(2s)^d 2^{-s\varepsilon/2}}{1-\exp\left(\frac{-s\varepsilon}{8}\right)} \leq \delta. \tag{15}$$

We are now left with the strenuous task of simplifying this expression. Routine calculation gives

$$(2s)^d 2^{-s\varepsilon/2} \leq \delta\left(1-\exp\left(\frac{-s\varepsilon}{8}\right)\right) \tag{16}$$

$$d\ln(2s) + \left(\frac{-s\varepsilon}{2}\right)\ln(2) \leq \ln(\delta) + \ln\left(1-\exp\left(\frac{-s\varepsilon}{8}\right)\right) \tag{17}$$

$$d\ln(2s) + \left(\frac{-s\varepsilon}{2}\right)\ln(2) \leq \ln(\delta) + \ln\left(1-\exp\left(\frac{-s\varepsilon}{8}\right)\right) \tag{18}$$

$$d\ln(2s) + \left(\frac{-s\varepsilon}{2}\right)\ln(2) \leq \ln(\delta) + \ln\left(1-\exp\left(\frac{-s\varepsilon}{8}\right)\right) \tag{19}$$

$$s\frac{\varepsilon}{2}\ln(2) \geq \ln\left(\frac{1}{\delta}\right) + d\ln(2s) - \ln\left(1-\exp\left(\frac{-s\varepsilon}{8}\right)\right) \tag{20}$$

$$s\ln(2) \geq \frac{2}{\varepsilon}\left(\ln\left(\frac{1}{\delta}\right) + d\ln(2s) - \ln\left(1-\exp\left(\frac{-s\varepsilon}{8}\right)\right)\right) \tag{21}$$

$$s \geq \frac{2}{\ln(2)\varepsilon}\left(\ln\left(\frac{1}{\delta}\right) + d\ln(2s) - \ln\left(1-\exp\left(\frac{-s\varepsilon}{8}\right)\right)\right) \tag{22}$$

□

B Detailed Experimental Results in NN Robustness

For each of our experimental runs, we report the training procedure used (Procedure), the seed, the TRADES robustness-accuracy parameter β, the mapping size $|M|$, the number of predictions for which κ is smaller than $\kappa_{\max}$ denoted by $|\mathbf{x}_{\kappa \leq \kappa_{\max}}|$, the estimators n_c and $\hat{p}_\kappa$ from Section 3.7.4, the runtime, and the accuracy. We report individual metrics for our experiments in Tables 1 to 4.

C Detailed Experimental Results in CPS Verification

In our experiments in Section 4.7.3, we mine specifications from simulation traces for the two PSTL formulas

$$\varphi_1 = \mathbf{G}(\texttt{abs\_pole\_angle}(\mathbf{w}) < \alpha_1 \wedge \texttt{abs\_cart\_position}(\mathbf{w}) < \alpha_2) \tag{23}$$

and

$$\varphi_2 = \mathbf{G}_{[10,500]}\Big((\texttt{abs\_pole\_angle}(\mathbf{w}) > \alpha_1 \vee \texttt{abs\_cart\_position}(\mathbf{w}) > \alpha_2) \tag{24}$$

$$\rightarrow \mathbf{F}_{[0,\tau]}(\texttt{abs\_pole\_angle}(\mathbf{w}) < \alpha_3 \wedge \texttt{abs\_cart\_position}(\mathbf{w}) > \alpha_4)\Big). \tag{25}$$

Table 1: Results for the MNIST dataset using PGD as a local robustness oracle.

Procedure	seed	β	$\lvert M \rvert$	$\lvert \mathbf{x}_{\kappa \leq \kappa_{\max}} \rvert$	n_c	$\hat{p}_\kappa$	runtime (s)	accuracy
TRADES	10	0.001	31	9743	0	0	85	0.96
		0.01	46	9687	2	0.0017	92	0.98
		0.02	61	9668	1	0.0014	92	0.98
		0.05	50	9794	0	0	79	0.98
		0.1	66	9772	2	0.0002	87	0.98
		0.2	70	9764	5	0.0010	98	0.98
		0.5	58	9874	3	0.0022	67	0.97
		1	49	9904	5	0.0192	67	0.96
		2	57	9879	3	0.0004	64	0.93
		5	47	9894	4	0.0009	64	0.90
		10	40	9893	5	0.0004	101	0.82
		20	56	9924	1	0.0020	99	0.58
TRADES	20	0.001	39	9816	1	0.0020	79	0.95
		0.01	49	9686	1	0.0010	90	0.98
		0.02	48	9651	0	0	91	0.97
		0.05	53	9764	0	0	85	0.98
		0.1	61	9776	3	0.0002	59	0.98
		0.2	73	9811	5	0.0011	58	0.97
		0.5	69	9849	0	0	92	0.97
		1	50	9894	2	0.0003	100	0.95
		2	55	9873	6	0.0118	70	0.92
		5	58	9892	0	0	94	0.88
		10	26	9875	4	0.0016	56	0.78
TRADES	30	0.001	42	9739	0	0	90	0.96
		0.01	44	9840	0	0	55	0.98
		0.02	53	9746	2	0.0030	75	0.98
		0.05	61	9805	0	0	57	0.98
		0.1	67	9810	2	0.0008	58	0.98
		0.2	72	9762	3	0.0031	79	0.97
		0.5	69	9851	3	0.0034	94	0.97
		1	64	9884	2	0.0025	63	0.95
		2	38	9877	0	0	75	0.93
		5	36	9877	0	0	64	0.90
		10	36	9884	5	0.0082	105	0.79
		20	7	9879	0	0	95	0.65
Standard	10	0.1	34	9836	3	0.0053	71	0.97
Standard	20	0.1	32	9794	0	0	90	0.96
Standard	30	0.1	42	9858	0	0	77	0.94

Table 2: Results for the MNIST dataset using Marabou as a local robustness oracle.

Procedure	seed	β	$\lvert M\rvert$	$\lvert\mathbf{x}_{\kappa\leq\kappa_{\max}}\rvert$	n_c	$\hat{p}_\kappa$	runtime (s)	accuracy
TRADES	20	2	5	9659	1	0.0001	≈ 216000	0.92
Standard	20	2	3	9585	0	0	≈ 216000	0.96

Table 3: Results for the MNIST dataset using LiRPA as a local robustness oracle.

Procedure	seed	β	$\lvert M\rvert$	$\lvert\mathbf{x}_{\kappa\leq\kappa_{\max}}\rvert$	n_c	$\hat{p}_\kappa$	runtime (s)	accuracy
TRADES	10	2	20	9726	1	0.0009	484	0.93
TRADES	20	2	17	9659	1	0.0002	481	0.92
TRADES	30	2	10	9628	0	0	480	0.93
Standard	10	2	16	9624	0	0	483	0.96
Standard	20	2	20	9575	0	0	484	0.96
Standard	30	2	18	9627	1	0.0001	483	0.94

Table 4: Results for the CIFAR-10 dataset using PGD as a local robustness oracle.

Procedure	seed	β	$\lvert M\rvert$	$\lvert\mathbf{x}_{\kappa\leq\kappa_{\max}}\rvert$	n_c	$\hat{p}_\kappa$	runtime (s)	accuracy
		0.1	22	9691	2	0.0004	561	0.76
		0.2	24	9735	7	0.0148	639	0.67
TRADES	10	0.5	33	9754	0	0	833	0.69
		1	31	9738	5	0.0022	892	0.81
		2	29	9770	0	0	1087	0.79
		0.1	19	9728	2	0.0010	596	0.82
		0.2	20	9727	2	0.0003	608	0.64
TRADES	20	0.5	31	9717	0	0	947	0.72
		1	38	9800	0	0	1025	0.70
		2	42	9746	2	0.0005	1165	0.82
		0.1	29	9771	8	0.0025	608	0.67
		0.2	21	9717	1	0.0004	666	0.63
TRADES	30	0.5	22	9672	3	0.0009	591	0.83
		1	29	9738	2	0.0025	945	0.81
		2	36	9785	2	0.0052	1184	0.79
Standard	10	0.1	12	9708	0	0	345	0.70
Standard	20	0.1	12	9564	0	0	331	0.75
Standard	30	0.1	15	9694	2	0.0020	423	0.48

Our simple specification mining procedure works as follows. We partially choose parameter values by hand, and find the strictest parametrisation of the remaining parameters admissible by our samples. This procedure is performed twice: one time on the full sample and one time on 10% of our sample to investigate if smaller sample sizes would still lead to good specifications.

For φ_1, there is a unique strictest parametrisation, which we call "Optimal". For φ_2, we partially choose three of the five parameters by hand, and optimise the remaining two parameter values with different weights for getting a tight parametrisation for

- both the angle and the position with equal weights: "Balanced"
- just the angle independent of the position: "Angle"
- just the position independent of the angle "Position"

In addition, we take the *conjunction* of these three parametrisations as per Corollary 2.5.1 for the specification "Combined". We expect that the combined specification holds true with a probability of at least $1 - 3\varepsilon$. The mined parameter values for the full sample are depicted in Table 5 and in Table 6 for the subset of our sample.

Formula	Specification	mined parameter values				
		α_1	α_2	α_3	α_4	τ
φ_1	Optimal	0.09213	0.27972	-	-	-
φ_2	Balanced	**0.05**	**0.27**	0.06739	0.29564	**20**
	Angle	**0.08**	**10**	0.04549	0.12342	**5**
	Position	**1.00**	**0.25**	0.01830	0.29970	**10**

Table 5: Mined parameter values for the experiments in Section 4.7.3 using the full samples for the respective specification. Bold parameter values were chosen manually.

Formula	Specification	mined parameter values				
		α_1	α_2	α_3	α_4	τ
φ_1	Optimal	0.08847	0.27588	-	-	-
φ_2	Balanced	**0.05**	**0.27**	0.06688	0.29564	**20**
	Angle	**0.08**	**10**	0.04407	0.12160	**5**
	Position	**1.00**	**0.25**	0.01235	0.29970	**10**

Table 6: Mined parameter values for the experiments in Section 4.7.3 using 10% of the samples for the respective specification. Bold parameter values were chosen manually.