Adversarial Robustness Guarantees for Classification with Gaussian Processes

Arno Blaas[*]
Department of Engineering Science
University of Oxford

Andrea Patane[*]
Department of Computer Science
University of Oxford

Luca Laurenti[*]
Department of Computer Science
University of Oxford

Luca Cardelli
Department of Computer Science
University of Oxford

Marta Kwiatkowska
Department of Computer Science
University of Oxford

Stephen Roberts
Department of Engineering Science
University of Oxford

Abstract

We investigate adversarial robustness of Gaussian Process Classification (GPC) models. Given a compact subset of the input space $T \subseteq \mathbb{R}^d$ enclosing a test point x^* and a GPC trained on a dataset $\mathcal{D}$, we aim to compute the minimum and the maximum classification probability for the GPC over all the points in T. In order to do so, we show how functions lower- and upper-bounding the GPC output in T can be derived, and implement those in a branch and bound optimisation algorithm. For any error threshold $\epsilon > 0$ selected *a priori*, we show that our algorithm is guaranteed to reach values ϵ-close to the actual values in finitely many iterations. We apply our method to investigate the robustness of GPC models on a 2D synthetic dataset, the SPAM dataset and a subset of the MNIST dataset, providing comparisons of different GPC training techniques, and show how our method can be used for interpretability analysis. Our empirical analysis suggests that GPC robustness increases with more accurate posterior estimation.

1 INTRODUCTION

Adversarial examples (i.e. input points intentionally crafted to trick a model into misclassification) have raised serious concerns about the security and robustness of models learned from data (Biggio & Roli, 2018). Since test accuracy fails to account for the behaviour of a model in adversarial settings, the development of techniques capable of quantifying the adversarial robustness of machine learning models is an essential pre-condition for their application in safety-critical scenarios (Ribeiro et al., 2016). In particular, Gaussian Processes (GPs), thanks to their favourable analytical properties, allow for the computation of the uncertainty over model predictions in Bayesian settings, which can then be propagated through the decision pipeline to facilitate decision-making (Rasmussen, 2004). However, while techniques for the computation of robustness guarantees have been developed for a variety of non-Bayesian machine learning models (Katz et al., 2017; Huang et al., 2017; Biggio & Roli, 2018), to the best of our knowledge studies of *adversarial classification robustness* of GPs have been limited to statistical (i.e. input distribution dependent) (Abdelaziz, 2017) and heuristic analyses (Grosse et al., 2018; Bradshaw et al., 2017), and methods for the computation of adversarial robustness guarantees are missing.

In this work, given a trained GP Classification (GPC) model and a compact subset of the input space $T \subseteq \mathbb{R}^d$, we pose the problem of computing the maximum and minimum of the GPC class probabilities over all $x \in T$. We show that such values naturally allow us to compute robustness properties employed for analysis of deep learning models (Ruan et al., 2018), e.g. can be used to provide guarantees of non-existence of adversarial examples and for the computation of classification ranges for sets of input points. Unfortunately, exact direct computation of the maximum and minimum class probabilities over compact sets is not possible, as these would require providing an *exact solution* of a global non-linear optimisation problem, for which no general method exists (Neumaier, 2004). We show how upper and lower bounds for the maximum and minimum classification probabilities of GPCs can be computed on any given compact set T, and then iteratively refine these bounds in a branch and bound algorithmic scheme until convergence to the minimum and maximum is obtained.

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

[*]Equal Contributions.

Specifically, through discretisation of the GPC latent space, we derive an upper and lower bound on the GPC class confidence output by analytically optimising a set of Gaussian integrals, whose parameters depend upon extrema of the GPC posterior mean and variance in T. We show how the latter can be bounded by solving a set of convex quadratic and linear programming problems, for which solvers are readily available (Boyd & Vandenberghe, 2004). Finally, for any given error tolerance $\epsilon > 0$, we prove that there exists a discretisation of the latent space that ensures convergence of the branch and bound to values ϵ-close to the actual maximum and minimum class probabilities in finitely many steps. The method we propose is anytime (the bounds provided are at every step an over-estimation of the actual classification ranges over T, and can hence be used to provide guarantees) and ϵ-exact (the actual values are retrieved in finitely many steps up to an error ϵ selected a-priori).

We apply our approach to analyse the robustness profile of GPCs on a two-dimensional dataset, the SPAM dataset, and a feature-based analysis of a binary and a 3-class subset of the MNIST dataset. In particular, we compare the guarantees computed by our method with the robustness estimation approximated by adversarial attack methods for GPCs (Grosse et al., 2018), discussing in which settings the latter fails. Then, we analyse the effect of approximate Bayesian inference techniques and hyper-parameter optimisation procedures on the GPC adversarial robustness. Interestingly, across the three datasets analysed here, we observe that approximation based on Expectation Propagation (Minka, 2001) gives more robust classification models than Laplace approximation (Rasmussen, 2004), and that GPC robustness increases with the number of training epochs. Finally, we show how robustness can be used to perform interpretability analysis of GPC predictions and compare our methodology with LIME (Ribeiro et al., 2016).

In summary, the paper presents the following contributions:

- We develop a method for computing lower and upper bounds for GPC probabilities over compact sets.
- We incorporate the bounding procedure in a branch and bound algorithm, which we show to converge for any specified error $\epsilon > 0$ in finitely many steps.
- We empirically evaluate the robustness of a variety of GPC models on three datasets, and demonstrate how our method can be used for interpretability analysis.

Related Work Different notions of robustness have been studied for GPs. For instance, Kim & Ghahramani (2008) consider robustness against outliers, while Hernández-Lobato et al. (2011) study robustness against labelling errors. In this paper we consider robustness against local adversarial perturbations, whose quantification for Bayesian models is a problem addressed in several papers. Heuristic approaches based on studying adversarial examples are developed by Grosse et al. (2018); Feinman et al. (2017). Formal guarantees are derived by Cardelli et al. (2019b); Bogunovic et al. (2018); Smith et al. (2019) for GPs and by Cardelli et al. (2019a) for Bayesian neural networks. In particular, Cardelli et al. (2019b) derive an upper bound on the probability that there exists a point in the neighbourhood of a given test point of a GP such that the prediction of the GP on the latter differs from the initial test input point by at least a specified threshold, whereas Bogunovic et al. (2018) consider a GP optimisation algorithm in which the returned solution is guaranteed to be robust to adversarial perturbations with a certain probability. The problem and the techniques developed in this paper are substantially different from both of these. First, we consider a classification problem, for which the bounds in the referenced papers cannot be applied due to its non-Gaussian nature. Then, the approach in this paper gives stronger (i.e., non-probabilistic) guarantees, is guaranteed to converge to any given error $\epsilon > 0$ in finite time, and is anytime (i.e., at any time it gives sound upper and lower bounds of the classification probabilities). This also differs from Cardelli et al. (2019a), where the authors consider statistical guarantees that require the solution of many non-linear optimisation problems (one for each sample from the posterior distribution). Our approach also differs from that in Smith et al. (2019), where the authors give guarantees for GPC in a binary classification setting under the L_0-norm and only consider the mean of the distribution in the latent space without taking into account the uncertainty intrinsic in the GPC framework. In contrast, our approach also considers multiclass classification, takes into account the full posterior distribution and allows for exact (up to $\epsilon > 0$) computation under any L_p-norm.

2 BAYESIAN CLASSIFICATION WITH GAUSSIAN PROCESSES

In this section we provide background for classification with GP priors. We consider the classification problem associated to a dataset $\mathcal{D} = \{(x, y) \,|\, x \in \mathbb{R}^d, y \in \{1, ..., C\}\}$. In GPC settings, given a test point $x^* \in \mathbb{R}^d$, the probability assigned by the GPC to x^* belonging to class c is given by:

$$\pi^c(x^*|\mathcal{D}) = \int \sigma^c(\bar{f}) p(f(x^*) = \bar{f}|\mathcal{D}) d\bar{f}, \qquad (1)$$

where $f(x^*) = [f^1(x^*), \dots, f^C(x^*)]$ is the latent function vector, $\sigma^c : \mathbb{R}^C \to [0, 1]$ is the likelihood function for class c, $p(f(x^*) = \bar{f}|\mathcal{D})$ is the predictive posterior distribution of the GP, and the integral is computed over the C-dimensional latent space (Rasmussen, 2004). The vector of class probabilities, $\Pi(x^*) = [\pi^1(x^*|\mathcal{D}), ..., \pi^C(x^*|\mathcal{D})]$, can be computed by iterating Eqn (1) for each class $c = 1, \dots, C$. Of particular interest in applications is the binary classification case (i.e., when C=2), which leads to a significant simplification of the inference equations and techniques, while still encompassing important practical applications (Nickisch & Rasmussen, 2008) (notice that the binary case can be used for multi-class classification as well by means of, e.g., one-vs-all classifiers (Rasmussen, 2004; Hsu & Lin, 2002)). More specifically, in this case it suffices to compute $\pi(x^*|\mathcal{D}) = \int \sigma(\bar{f})p(f(x^*) = \bar{f}|\mathcal{D})d\bar{f} := \pi^1(x^*|\mathcal{D})$, with f being a univariate latent function and setting $\pi^2(x^*|\mathcal{D}) := 1 - \pi(x^*|\mathcal{D})$. In other words, when $C = 2$ the latent space of the GPC model is one-dimensional.

Unfortunately, even under the GP prior assumption, the posterior distribution in classification settings, $p(f(x^*) = \bar{f}|\mathcal{D})$, is non-Gaussian and intractable (Rasmussen, 2004). Several approximation methods have been developed to perform GPC inference, either by sampling (e.g. Markov Chain Monte Carlo), or by developing suitable analytic approximations of the posterior distribution. In this work we focus on Gaussian analytic approximations, that is, we employ GPC methods that perform approximate inference of $p(f(x^*) = \bar{f}|\mathcal{D})$ by estimating a Gaussian distribution $q(f(x^*) = \bar{f}|\mathcal{D}) = \mathcal{N}(\bar{f}\,|\,\mu(x^*), \Sigma(x^*))$. The latter is then used at inference time in Eqn (1) in place of the exact posterior $p(f(x^*) = \bar{f}|\mathcal{D})$[1]. In particular, in Section 6 we will give experimental results for when q is derived using either *Laplace approximations* (Williams & Barber, 1998) or *Expectation Propagation* (EP) (Minka, 2001). However, we remark that the methods presented in this paper do not depend on the particular Gaussian approximation method used and can be trivially extended to the case where q is a mixture of Gaussian distributions.

3 ADVERSARIAL ROBUSTNESS

Given a GPC model trained on a dataset $\mathcal{D}$ and a test point x^*, we are interested in quantifying the adversarial robustness of the GPC in a neighborhood of x^*. To do so, for a compact set T and a class $c \in \{1, \dots, C\}$, we pose the problem of computing the minimum and the maximum that the GPC assigns to the probability of class c in T, that is:

$$\pi^c_{\min}(T) := \min_{x \in T} \pi^c(x|\mathcal{D}) \quad \pi^c_{\max}(T) := \max_{x \in T} \pi^c(x|\mathcal{D}) \quad (2)$$

The computation of the classification extrema in T allows us to determine the reachable interval of class probabilities over T. In the case in which T is defined as a neighborhood around a test point x^*, Eqn (2) provides a quantification of the local GPC robustness at x^*, that is, against local adversarial perturbations. Unfortunately, exact computation of Eqn (2) involves the solution of two non-linear optimisation problems, for which no general solution method exists. Nevertheless, in Section 4 we derive a branch and bound scheme for the anytime computation of the classification ranges of Eqn (2) that is guaranteed to converge in finitely many iterations up to any arbitrary error tolerance $\epsilon > 0$.

[1]With an abuse of notation, in the rest of the paper we will consider $\pi^c(x^*|\mathcal{D}) = \int \sigma^c(\bar{f})q(f(x^*) = \bar{f}|\mathcal{D})d\bar{f}$.

In what remains of this section we discuss two notions of adversarial robustness employed for the analysis of deep learning models (Ruan et al., 2018) that arise as particular instances of Eqn (2), which will be investigated in the experimental results discussed in Section 6.

Definition 1. *(Adversarial Local Robustness) Let $T \subseteq \mathbb{R}^d$ and $x^* \in T$. Then, for $\delta > 0$ we say that the classification of x^* is $\delta-$robust in T iff $\forall x \in T$, $|\Pi(x^*) - \Pi(x)| \leq \delta$, where $|\cdot|$ is a given norm.*

If T is a $\gamma-$ball around a test point x^*, then robustness defined in Definition 1 allows one to quantify how much, in the worst case, the prediction in x^* can be affected by input perturbations of radius no greater than γ. Adversarial examples are defined in terms of invariance of the classification in T w.r.t. the label of a test point x^*. For the case of the Bayesian optimal classifier, this is defined as follows.

Definition 2. *(Adversarial Local Safety) Let $T \subseteq \mathbb{R}^d$ and $x^* \in T$. Then, we say that the classification of x^* is safe in T iff $\forall x \in T$, $\arg\max_{c \in \{1,\dots,C\}} \pi^c(x|\mathcal{D}) = \arg\max_{c \in \{1,\dots,C\}} \pi^c(x^*|\mathcal{D})$.*

Adversarial local safety establishes whether adversarial examples exist in T, yielding formal guarantees against adversarial attacks for GPCs. If we again consider T to be a $\gamma-$ball around x^*, the satisfaction of Definition 2 guarantees that it is not possible to cause a misclassification by perturbing x^* by a magnitude of up to γ[2].

4 BOUNDS FOR BINARY CLASSIFICATION

In this section we show how the classification ranges of a two-class GPC model in any given compact set $T \subseteq \mathbb{R}^d$ can be computed up to any arbitrary precision $\epsilon > 0$. As explained in Section 2, the latent space of the GPC model is one-dimensional in this case, and we thus omit the class superscript c in this section. The extension to the multiclass scenario is then described in Section 5. Proofs for the results stated are given in the Supplementary Material.

Outline of Approach An outline of our approach is depicted in Figure 1 for the computation of $\pi_{\min}(T)$ over a one-dimensional set T plotted along the x-axis (the method for the computation of $\pi_{\max}(T)$ is analogous). For any given region T we aim to compute lower and upper bounds on both $\pi_{\min}(T)$ and $\pi_{\max}(T)$, that is, we compute real values $\pi^L_{\min}(T)$, $\pi^U_{\min}(T)$, $\pi^L_{\max}(T)$ and $\pi^U_{\max}(T)$ such that:

$$\pi^L_{\min}(T) \leq \pi_{\min}(T) \leq \pi^U_{\min}(T) \quad (3)$$
$$\pi^L_{\max}(T) \leq \pi_{\max}(T) \leq \pi^U_{\max}(T). \quad (4)$$

[2]In the multiclass case to check if x^* is safe in T, for $\bar{c} = \arg\max_{c \in \{1,\dots,C\}} \pi^c(x^*|\mathcal{D})$, we need to check that $\min_{x \in T} \left(\pi^{\bar{c}}(x|\mathcal{D}) - \max_{c \neq \bar{c}} \pi^c(x|\mathcal{D})\right) > 0$, which can be computed with a trivial extension of the results presented in this paper.

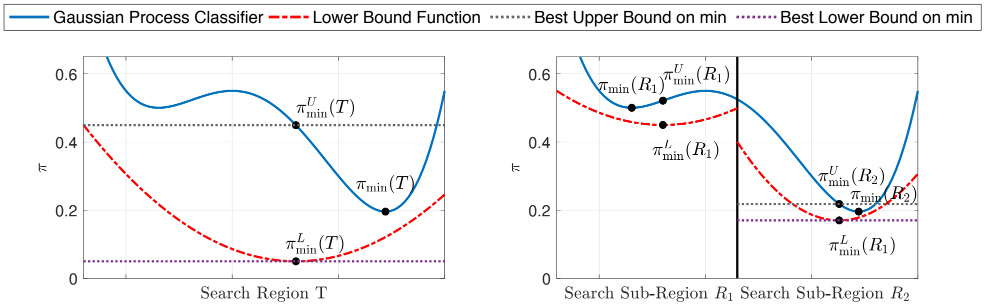

Figure 1: **Left:** Computation of upper and lower bounds on $\pi_{\min}(T)$, i.e. the minimum of the classification range on the search region T. **Right:** The search region is repeatedly partitioned into sub-regions according to Algorithm 1 (only first partitioning visualised), reducing the gap between best lower and upper bounds until convergence (up to ϵ) is reached.

In order to do so, we compute a lower and an upper bound function (the lower bound function is depicted with a dashed red curve in Figure 1) to the GPC output (solid blue curve) in the region T. We then find the minimum of the lower bound function, $\pi^L_{\min}(T)$ (shown in the plot), and the maximum of the upper bound function, $\pi^U_{\max}(T)$ (not shown). Then, valid values for $\pi^U_{\min}(T)$ and $\pi^L_{\max}(T)$ can be computed by evaluating the GPC on any point in T (a specific $\pi^U_{\min}(T)$ is depicted in Figure 1). Finally, we iteratively refine the lower and upper bounds computed in T with a branch and bound algorithm. Namely, the region T is recursively subdivided into sub-regions, for which we compute new (tighter) bounds, until these converge up to a desired tolerance $\epsilon > 0$.

Computation of Bounds In this paragraph we show how to compute $\pi^U_{\max}(T)$, an upper bound on the maximum, and $\pi^L_{\min}(T)$, a lower bound on the minimum of the GPC outputs. We work on the assumption that the likelihood function $\sigma(f)$ is a monotonic, non-decreasing, and continuous function of the latent variable (notice that this is satisfied by commonly used likelihood functions, e.g., logistic and probit (Kim & Ghahramani, 2006)). In the following proposition we show how the GPC output can be upper- and lower-bounded in T by a summation of Gaussian integrals.

Proposition 1. *Let $\mathcal{S} = \{S_i \mid i \in \{1, \ldots N\}\}$ be a partition of $\mathbb{R}$ (the latent space) in a finite set of intervals. Call $a_i = \inf_{\bar{f} \in S_i} \bar{f}$ and $b_i = \sup_{\bar{f} \in S_i} \bar{f}$. Then, it holds that:*

$$\pi_{\min}(T) \geq \sum_{i=1}^{N} \sigma(a_i) \min_{x \in T} \int_{a_i}^{b_i} \mathcal{N}(\bar{f}|\mu(x), \Sigma(x)) d\bar{f} \quad (5)$$

$$\pi_{\max}(T) \leq \sum_{i=1}^{N} \sigma(b_i) \max_{x \in T} \int_{a_i}^{b_i} \mathcal{N}(\bar{f}|\mu(x), \Sigma(x)) d\bar{f}, \quad (6)$$

where $\mu(x)$ and $\Sigma(x)$ are mean and variance of the predictive posterior $q(f(x) = \bar{f}|\mathcal{D})$.

Proposition 1 guarantees that the GPC output in T can be bounded by solving N optimisation problems. Each of these problems seeks to find the mean and variance that maximise or minimise the integral of a Gaussian over T. This has been studied by Cauchi et al. (2019) for variance-independent points and is generalised in the following proposition. We introduce the following notation for lower and upper bounds on mean and variance in T:

$$\mu^L_T \leq \min_{x \in T} \mu(x) \qquad \mu^U_T \geq \max_{x \in T} \mu(x) \quad (7)$$

$$\Sigma^L_T \leq \min_{x \in T} \Sigma(x) \qquad \Sigma^U_T \geq \max_{x \in T} \Sigma(x), \quad (8)$$

Then by inspection of the derivatives of the integrals in Eqns (5) and (6) the following proposition follows.

Proposition 2. *Let $\mu^m = \frac{a+b}{2}$ and $\Sigma^m(\mu) = \frac{(\mu-a)^2 - (\mu-b)^2}{2 \log \frac{\mu-a}{\mu-b}}$. Then it holds that:*

$$\max_{x \in T} \int_a^b \mathcal{N}(\bar{f}|\mu(x), \Sigma(x)) d\bar{f} \leq \int_a^b \mathcal{N}(\bar{f}|\overline{\mu}, \overline{\Sigma}) d\bar{f} = \frac{1}{2}\left(\operatorname{erf}\left(\frac{\overline{\mu} - a}{\sqrt{2\overline{\Sigma}}}\right) - \operatorname{erf}\left(\frac{\overline{\mu} - b}{\sqrt{2\overline{\Sigma}}}\right)\right) \quad (9)$$

$$\min_{x \in T} \int_a^b \mathcal{N}(\bar{f}|\mu(x), \Sigma(x)) d\bar{f} \geq \int_a^b \mathcal{N}(\bar{f}|\underline{\mu}, \underline{\Sigma}) d\bar{f} = \frac{1}{2}\left(\operatorname{erf}\left(\frac{\underline{\mu} - a}{\sqrt{2\underline{\Sigma}}}\right) - \operatorname{erf}\left(\frac{\underline{\mu} - b}{\sqrt{2\underline{\Sigma}}}\right)\right) \quad (10)$$

where: $\overline{\mu} = \arg\min_{\mu \in [\mu^L_T, \mu^U_T]} |\mu^m - \mu|$ and $\overline{\Sigma}$ is equal to Σ^L_T if $\overline{\mu} \in [a, b]$, otherwise $\overline{\Sigma} = \arg\min_{\Sigma \in [\Sigma^L_T, \Sigma^U_T]} |\Sigma^m(\overline{\mu}) - \Sigma|$. Analogously, for the minimum we have: $\underline{\mu} = \arg\max_{\mu \in [\mu^L_T, \mu^U_T]} |\mu^m - \mu|$ and $\underline{\Sigma} = \arg\min_{\Sigma \in \{\Sigma^L_T, \Sigma^U_T\}} [\operatorname{erf}(b|\underline{\mu}, \Sigma) - \operatorname{erf}(a|\underline{\mu}, \Sigma)]$.

That is, given lower and upper bounds for the a-posteriori mean and variance in T, Proposition 2 allows us to analytically bound the N optimisations of Gaussian integrals posed by Equations (5) and (6). Through this, we can compute values for $\pi^L_{\min}(T)$ and $\pi^U_{\max}(T)$, which satisfy the LHS of Eqn (3) and the RHS of Eqn (4). Furthermore, note that by definition of $\pi_{\min}(T)$ and $\pi_{\max}(T)$, we have that, for every $\bar{x} \in T$, setting $\pi^U_{\min}(T) = \pi^L_{\max}(T) = \pi(\bar{x})$ provides values which satisfy the RHS of Eqn (3) and the

LHS of Eqn (4) (in the Supplementary Material we discuss how to pick values for $\bar{x}$ to speed up convergence). Details on the computation of bounds for the a-posteriori mean and variance are discussed in the Supplementary Material. Interestingly, when the (scaled) probit function is chosen for the likelihood, $\sigma(f)$, then the inference integral over $q(f(x^*) = \bar{f}|\mathcal{D})$ can be expressed in closed form (Rasmussen, 2004), which leads to a simplification of Proposition 1. Details are given in the Supplementary Material.

Branch and Bound Algorithm In this paragraph we implement the bounding procedure into a branch and bound algorithm and prove convergence up to any a-priori specified $\epsilon > 0$. We summarise our method for computation of $\pi_{\min}(T)$ in Algorithm 1, which we now briefly describe (analogous arguments hold for $\pi_{\max}(T)$). After initialising $\pi^L_{\min}(T)$ and $\pi^U_{\min}(T)$ to trivial values and initialising the exploration regions stack $\mathbf{R}$ to the singleton $\{T\}$, the main optimisation loop is entered until convergence (lines 2–9). Among the regions in the stack, we select the region R with the most promising lower bound (line 3), and refine its lower bounds using Propositions 1 and 2 (lines 4–5) as well as its upper bounds through evaluation of points in R (line 6). If further exploration of R is necessary for convergence (line 7), then the region R is partitioned into two smaller regions R_1 and R_2, which are added to the regions stack and inherit R's bound values (line 8). Finally, the freshly computed bounds local to $R \subseteq T$ are used to update the global bounds for T (line 9). Namely, $\pi^L_{\min}(T)$ is updated to the smallest value among the $\pi^L_{\min}(R)$ values for $R \in \mathbf{R}$, while $\pi^U_{\min}(T)$ is set to the lowest observed value yet explicitly computed in line 6.

Algorithm 1 Branch and bound for $\pi_{\min}(T)$

Input: Input space subset T; error tolerance $\epsilon > 0$; latent mean/variance functions $\mu(\cdot)$ and $\Sigma(\cdot)$ of $q(f(x) = \bar{f}|\mathcal{D})$
Output: Lower and upper bounds on $\pi_{\min}(T)$ with $\pi^U_{\min}(T) - \pi^L_{\min}(T) \leq \epsilon$

1: *Initialisation:* Stack of regions $\mathbf{R} \leftarrow \{T\}$; $\pi^L_{\min}(T) \leftarrow -\infty$; $\pi^U_{\min}(T) \leftarrow +\infty$
2: **while** $\pi^U_{\min}(T) - \pi^L_{\min}(T) > \epsilon$ **do**
3: Select region $R \in \mathbf{R}$ with lowest bound $\pi^L_{\min}(R)$ and delete it from stack
4: Find bounds $[\mu^L_R, \mu^U_R]$ and $[\Sigma^L_R, \Sigma^U_R]$ for latent mean and variance functions over R
5: Compute $\pi^L_{\min}(R)$ from $[\mu^L_R, \mu^U_R]$ and $[\Sigma^L_R \Sigma^U_R]$ using Propositions 1 and 2
6: Find $\pi^U_{\min}(R)$ by evaluating GPC in a point in R
7: **if** $\pi^U_{\min}(R) - \pi^L_{\min}(R) > \epsilon$ **then**
8: Split R into two sub-regions R_1, R_2, add them to stack and use $\pi^L_{\min}(R), \pi^U_{\min}(R)$ as initial bounds for both sub-regions
9: Update $\pi^L_{\min}(T)$ and $\pi^U_{\min}(T)$ with current best bounds found
10: **return** $[\pi^L_{\min}(T), \pi^U_{\min}(T)]$

For our approach to work, it is crucial that Algorithm 1 converges, i.e. that the loop of lines $2-9$ terminates. Given an a-priori specified threshold ϵ, Theorem 1 ensures that there exists a latent space discretisation such that the bounding error (i.e. the difference between the upper and lower bound) vanishes. Thanks to the properties of branch and bound algorithms (Balakrishnan et al., 1991), this guarantees that our method converges in finitely many iterations.

Theorem 1. *Assume $\mu : \mathbb{R}^d \to \mathbb{R}$ and $\Sigma : \mathbb{R}^d \to \mathbb{R}$ are Lipschitz continuous in $T \subseteq \mathbb{R}^m$. Then, for $\epsilon > 0$, there exists a partition of the latent space $\mathcal{S}$ and $r > 0$ such that, for every $R \subseteq T$ of side length of less than r, it holds that $|\pi^U_{\min}(R) - \pi^L_{\min}(R)| \leq \epsilon$ and $|\pi^U_{\max}(R) - \pi^L_{\max}(R)| \leq \epsilon$.*

Computational Complexity Proposition 2 implies that the bounds in Proposition 1 can be obtained in $\mathcal{O}(N)$, with N being the number of intervals the real line is being partitioned into (this scales like $\frac{1}{\epsilon}$, as discussed in the proof of Theorem 1). Computation of μ^L_T and μ^U_T is performed in $\mathcal{O}(|\mathcal{D}|)$, while obtaining Σ^U_T involves the solution of a convex quadratic problem in $d + |\mathcal{D}|$ variables, where d is the dimension of the input space. Solving for Σ^L_T requires the solution of $2|\mathcal{D}| + 1$ linear programming problems in $d+|\mathcal{D}|$ dimensions. Refining through branch and bound has a worst-case cost exponential in the number of non-trivial dimensions of T. The CPU time required for convergence of our method is analysed in the Supplementary Material.

5 MULTICLASS CLASSIFICATION

In this section we show how the results for binary classification can be generalised to the multi-class case. Given a class index $c \in \{1, \ldots, C\}$, we are interested in computing upper and lower bounds on $\pi^c(x|\mathcal{D})$ for every $x \in T$. In order to do so, we extend Proposition 1 to the multi-class case in Proposition 3, and show that the resulting multi-dimensional integrals can be reduced to the two-class case by marginalisation (Proposition 4).

Proposition 3. *Let $\mathcal{S} = \{S_i \mid i \in \{1, ...N\}\}$ be a finite partition of $\mathbb{R}^C$ (the latent space). Then, for $c \in \{1, ..., C\}$:*

$$\pi^c_{\min}(T) \geq \sum_{i=1}^{N} \min_{x \in S_i} \sigma^c(x) \min_{x \in T} \int_{S_i} \mathcal{N}(\bar{f}|\mu(x), \Sigma(x)) d\bar{f}$$

$$\pi^c_{\max}(T) \leq \sum_{i=1}^{N} \max_{x \in S_i} \sigma^c(x) \max_{x \in T} \int_{S_i} \mathcal{N}(\bar{f}|\mu(x), \Sigma(x)) d\bar{f}$$

Proposition 3 guarantees that, for all $x \in T$, $\pi^c(x|\mathcal{D})$ can be upper- and lower-bounded by solving $2N$ optimisation problems. In Proposition 4, we show that upper and lower bounds for the integral of a multi-dimensional Gaussian distribution, such as those appearing in Proposition 3, can be obtained by optimising uni-dimensional integrals over both the input and latent space. In what follows, we call $\mu_{i:j}(x)$

the subvector of $\mu(x)$ containing only the components from i to j, and similarly we define $\Sigma_{i:k,j:l}(x)$, the submatrix of $\Sigma(x)$ containing rows from i to k and columns from j to l.

Proposition 4. *Let $S = \prod_{i=1}^{C}[k_i^1, k_i^2]$ be an axis-parallel hyper-rectangle. For $i \in \{1, \dots, C-1\}$ and $f \in \mathbb{R}^{C-1-i}$, define $\mathcal{I} := i+1 : C$ and:*

$$\mu_i^f(x) = \mu_i(x) - \Sigma_{i,\mathcal{I}}(x)\Sigma_{\mathcal{I},\mathcal{I}}^{-1}(f - \mu_{\mathcal{I}}(x))$$
$$\Sigma_i^f(x) = \Sigma_{i,i}(x) - \Sigma_{i,\mathcal{I}}(x)\Sigma_{\mathcal{I},\mathcal{I}}^{-1}\Sigma_{i,\mathcal{I}}^T(x).$$

Let $S^{i+1} = \prod_{j=i+1}^{C}[k_j^1, k_j^2]$, then we have that:

$$\max_{x\in T}\int_S \mathcal{N}(z|\mu(x),\Sigma(x)) \leq \max_{x\in T}\int_{k_C^1}^{k_C^2}\mathcal{N}(z|\mu_C(x),$$
$$\Sigma_{C,C}(x))dz \prod_{i=1}^{C-1}\max_{\substack{x\in T\\ f\in S^{i+1}}}\int_{k_i^1}^{k_i^2}\mathcal{N}(z|\mu_i^f(x),\Sigma_i^f(x))dz$$

$$\min_{x\in T}\int_S \mathcal{N}(z|\mu(x),\Sigma(x)) \geq \min_{x\in T}\int_{k_C^1}^{k_C^2}\mathcal{N}(z|\mu_C(x),$$
$$\Sigma_{C,C}(x))dz \prod_{i=1}^{C-1}\min_{\substack{x\in T\\ f\in S^{i+1}}}\int_{k_i^1}^{k_i^2}\mathcal{N}(z|\mu_i^f(x),\Sigma_i^f(x))dz.$$

Proposition 4 reduces the computation of the bounds for the multi-class case to a product of extrema of univariate Gaussian distributions for which Proposition 2 can be iteratively applied. Analogously to what we discussed for the binary case, the resulting bound can be refined through a branch and bound algorithm to ensure convergence up to any desired tolerance $\epsilon > 0$. Notice that the computational complexity for the multi-class case is exponential in C.

6 EXPERIMENTAL RESULTS

We employ our methods to experimentally analyse the robustness profile of GPC models in adversarial settings[3]. We give results for three datasets: (i) Synthetic2D, generated by shifting a two-dimensional standard-normal either along the first dimension (class 1) or the second one (class 2); (ii) the SPAM dataset (Dua & Graff, 2017); (iii) a subset of the MNIST dataset (LeCun, 1998) with classes 3 and 8 (MNIST38) and a subset with classes 3, 5 and 8 (MNIST358). For scalability, results for MNIST38 are given for feature-level analysis (as done in Ruan et al. (2018) for deep networks). Namely, we analyse either salient patches detected by SIFT (Lowe, 2004) or we select the relevant pixels corresponding to the shortest GP length-scales.

6.1 Adversarial Local Safety

We depict the local adversarial safety results for four points selected from the Synthethic2D, SPAM, and MNIST38

[3]Code: *https://github.com/andreapatane/check-GPclass*

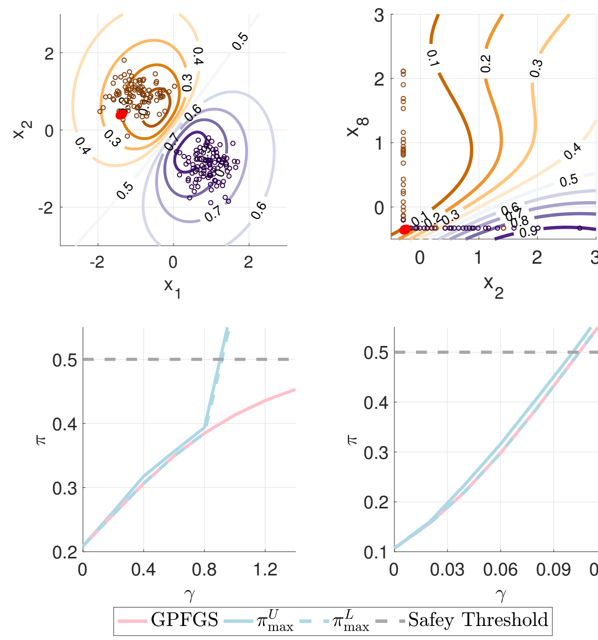

Figure 2: **First row**: Contour plot and test points for Synthetic2D (left); projected contour plot and test points for 2 dimensions of SPAM (right, dimensions 2 and 8 as selected by L_1-penalised logistic regression); red dots mark selected test points. **Second row**: Safety analysis for the two selected test point. Shown are the upper and lower bounds for tolerance $\epsilon = 0.02$ on $\pi_{\max}(T)$ (solid and dashed blue curves) and the GPFGS adversarial attack (pink curve).

datasets in Figures 2 and 3. To this end, we set $T \subseteq \mathbb{R}^d$ to be a L_∞ $\gamma-$ball around the chosen test point and iteratively increase γ (x-axis in the second row plots), checking whether there are adversarial examples in T. Namely, if the point is originally assigned to class 1 (respectively class 2) we check whether the minimum classification probability in T is below the decision boundary threshold, that is, if $\pi_{\min}(T) < 0.5$ (resp. $\pi_{\max}(T) > 0.5$). We compare the values provided by our method (blue solid and dashed line for class 2, green solid and dashed line for class 1) with GPFGS (Grosse et al., 2018), a gradient based heuristic attack for GPC (pink line). Naturally, as γ increases, the neighborhood region T becomes larger, hence the confidence for the initial class can decrease. Interestingly, while our method succeeds in finding adversarial examples in all cases shown (i.e. both the lower and upper bound on the computed quantity cross the decision boundary), the heuristic attack fails to find adversarial examples in the Synthetic2D and in the MNIST38 case. This happens as GPFGS builds on linear approximations of the GPC function, hence failing to find solutions to Eqn (2) when there are non-linearities. In particular, near the point selected for the Synthetic2D dataset (red dot in the contour plot) the gradient of the GPC points away from the decision boundary. Hence, no matter the value of γ, GPFGS will not go above

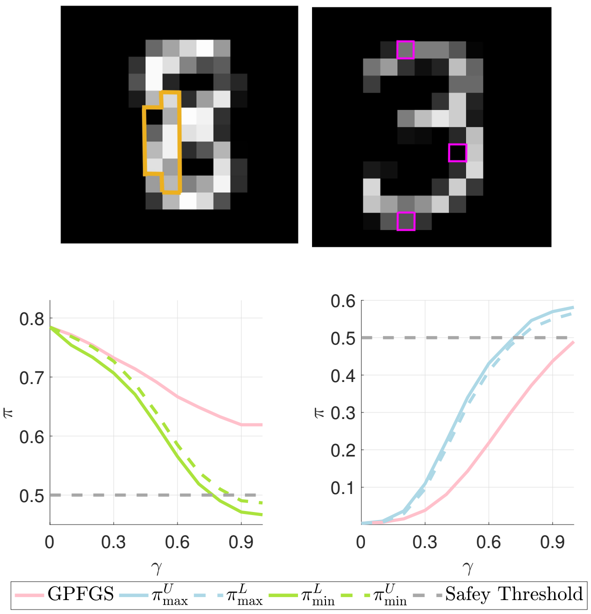

Figure 3: **First row**: Sample of 8 from MNIST38 along with 10 pixels selected by SIFT (left) and sample of 3 from MNIST38 along with the 3 pixels that have the shortest lengthscales after GPC training (right). **Second row**: Safety analysis for the two images. Shown are the upper and lower bounds for $\epsilon = 0.02$ on either $\pi_{\max}(T)$ or $\pi_{\min}(T)$ (solid and dashed blue resp. green curves) and the GPFGS adversarial attack (pink curve).

0.5 in this case (pink line of the bottom-left plot). On the other hand, for the SPAM dataset, the GPC model is locally linear around the selected test point (red dot in top right contour plot). Interestingly, the MNIST38 examples (Figure 3) provide results analogous to those of Synthetic2D. While our method finds adversarial examples on both occasions, GPFGS fails to do so (even with $\gamma = 1.0$ which is the maximum region possible for normalised pixel values).

6.2 Adversarial Local Robustness

We evaluate the empirical distribution of δ-robustness (see Definition 1) on 50 randomly selected test points for each of the three datasets considered. That is, given T, we compute $\delta = \pi_{\max}(T) - \pi_{\min}(T)$. Notice that a smaller value of δ implies a more robust model. In particular, we analyse how the GPC model robustness is affected by the training procedure used. We compare the robustness obtained when using either the Laplace or the EP posterior approximations technique. Further, we investigate the influence of the number of marginal likelihood evaluations (epochs) performed during hyper-parameter optimisation on robustness.

Results are depicted in Figure 4, for 10, 40 and 100 hyper-parameter optimisation epochs. Note that the analyses for the MNIST38 samples are restricted only to the most influential SIFT feature, and thus δ values for MNIST38 are smaller in magnitude than for the other two datasets (for which all the input variables are simultaneously changed). Interestingly, this empirical analysis demonstrates that GPCs trained with EP are consistently more robust than those trained using Laplace. In fact, for both Synthetic2D and MNIST38, EP yields a model about 5 times more robust than Laplace. For SPAM, the difference in robustness is the least pronounced. While Laplace approximation works by local approximations, EP calibrates mean and variance estimation by a global approach, which generally results in a more accurate approximation (Rasmussen, 2004). We compare Laplace and EP posterior approximations with that made by Hamiltonian Monte Carlo (HMC) - that is, as in Minka (2001) we use HMC as gold standard. The empirical distances found on the posterior approximation w.r.t. HMC are on average as follows (smaller values are better): (i) Synthetic2D - Laplace: 1.04, EP: 0.14; (ii) SPAM - Laplace: 0.35, EP: 0.32; (iii) MNIST38 - Laplace: 0.52, EP: 0.32. This shows a correlation between the robustness and the posterior approximation quality in the datasets considered. These results quantify and confirm for GPCs that a more refined estimation of the posterior is beneficial for model adversarial robustness (Cardelli et al., 2019a). Interestingly, the values of δ decrease as the number of training epochs increases, thus robustness improves with training epochs. This is in contrast to what is observed in the deep learning literature (Tsipras et al., 2018). More training in the Bayesian settings may imply better calibration of the latent mean and variance function to the observed data.

6.3 Interpretability Analysis

Finally, we show how adversarial robustness can be used for interpretability analysis for GPC models. We provide comparison with pixel-wise LIME (Ribeiro et al., 2016), a model-agnostic interpretability technique that relies on local linear approximations. Given a test point x^* consider the one-sided intervals $T^i_\gamma(x^*) = [x^*, x^* + \gamma e_i]$ (with e_i being the vector of 0s except for 1 at dimension i). We compute how much the maximum and minimum values can change over the one-sided intervals in both directions:

$$\begin{aligned}\boldsymbol{\Delta}^i_\gamma(x^*) = &\left(\pi_{\max}(T^i_\gamma(x^*)) - \pi_{\max}(T^i_{-\gamma}(x^*))\right)\\ &+ \left(\pi_{\min}(T^i_\gamma(x^*)) - \pi_{\min}(T^i_{-\gamma}(x^*))\right).\end{aligned}$$

Intuitively, this provides a non-linear generalisation of numerical gradient estimation (more details in Supplementary Material) which is close to the metric used in Ribeiro et al. (2016) as γ shrinks to 0. While $\boldsymbol{\Delta}^i_\gamma(x^*)$ is local to a given x^*, following LIME, global interpretability information is obtained by averaging local results over M test points, i.e. by computing $\boldsymbol{\Delta}^i_\gamma = \frac{1}{M}\sum_{j=1}^{M} \boldsymbol{\Delta}^i_\gamma(x^j)$.

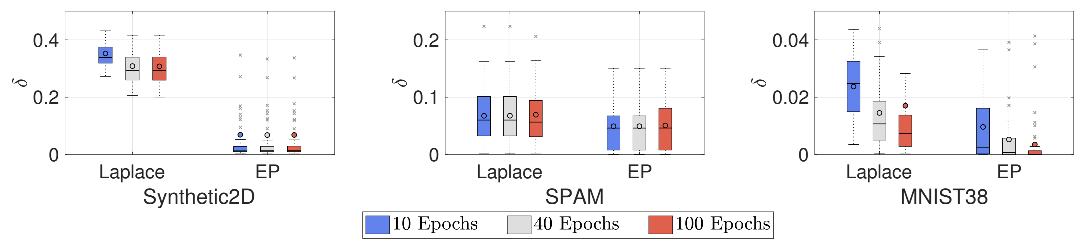

Figure 4: Boxplots for the distribution of robustness on the three datasets, comparing Laplace and EP approximation.

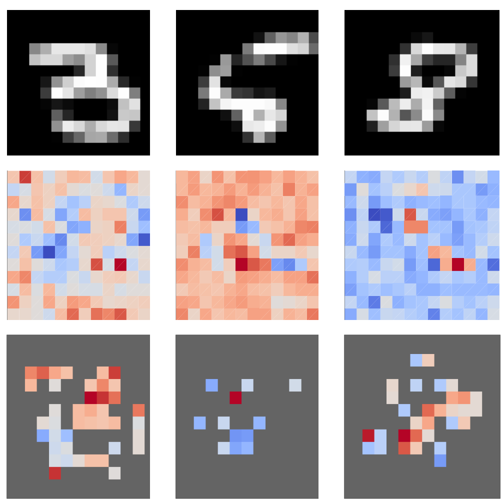

Figure 5: **First row**: Samples selected from MNIST358. **Second row**: Interpretability metric estimation using our method. **Third row**: Results obtained using LIME.

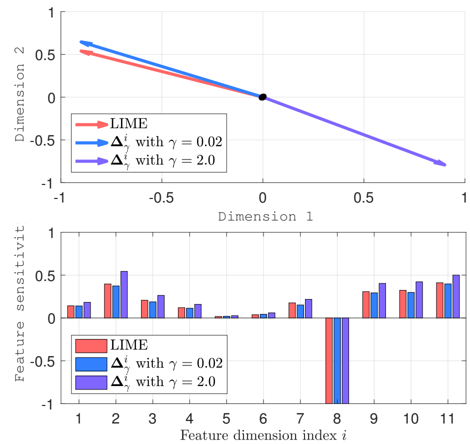

Figure 6: Global feature sensitivity analysed by LIME and our metric $\boldsymbol{\Delta}_\gamma^i$. All values normed to unit scale for better comparison. **Top:** Results for Synthetic2D dataset mapped out on plane. **Bottom:** Results for SPAM dataset.

Local Interpretability for MNIST358 Figure 5 shows the results for three samples selected from MNIST358 (top row), with the heat maps depicting the results of our method (second row) and those for LIME (third row, greyed out pixels are marked as irrelevant by LIME). The colour gradient varies from red (positive impact, pixel value increase causing increased class probability of shown digit) to blue (negative impact, pixel value increase decreasing the class probability). For digit 3, our method obtains for example a contiguous blue patch on the left. Increasing the values of these pixels would modify the 3 into an 8. Indeed, when whitening the pixels of the blue patch, the class 3 probability assigned by the model decreases from 0.58 to 0.40. Similarly, for digit 5, our methods identify a blue patch that would change the 5 into an 8 and again the GPC model indeed lowers its class 5 probability when the patch is whitened. Similarly, for digit 8, our method identifies a blue patch of 3 pixels towards the top left, which would turn it into something resembling digit 3 if whitened.

Global Interpretability for the Binary Datasets We perform global interpretability analysis on the GPC models trained on the Synthetic2D and SPAM datasets, using 50 random test points. The results are shown in Figure 6. For Synthetic2D (top row), LIME suggests that a higher probability of belonging to class 1 (depicted as the direction of the arrow in the plot) corresponds to lower values along dimension 1 and higher values along dimension 2. As can be seen in the corresponding contour plot in Figure 2 (top left), the exact opposite is true however. LIME, being built on linearity approximations, fails to take into account the global behaviour of the GPC. When using a small value of γ our approach obtains similar results to LIME. However, with $\gamma = 2.0$ the global relationship between input and output values is correctly captured. For SPAM, on the other hand (Figure 6, bottom), due to linearity of the dataset and GPC, a local analysis correctly reflects the global picture.

7 CONCLUSION

We presented a method for computing, for any compact set of input points, the class probability range of a GPC model across all points in that set, up to any precision $\epsilon > 0$. This allows us to analyse robustness and safety against adversarial attacks, which we have demonstrated on multiple datasets and approximate Bayesian inference techniques.

8 ACKNOWLEDGMENT

This project was partly funded by the EU's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie (grant agreement No 722022), the ERC under the European Unions Horizon 2020 research and innovation programme (grant agreement No. 834115) and the EPSRC Programme Grant on Mobile Autonomy (EP/M019918/1). Further funding was received from the Konrad-Adenauer-Stiftung and the Oxford Man-Institute of Quantitative Finance.

References

H. Abdelaziz. *A Data-Driven Approach for Modeling, Analysis and Control of Stochastic Hybrid Systems using Gaussian Processes*. PhD thesis, 2017.

V. Balakrishnan, S. Boyd, and S. Balemi. Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems. *International Journal of Robust and Nonlinear Control*, 1(4): 295–317, 1991.

B. Biggio and F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

I. Bogunovic, J. Scarlett, S. Jegelka, and V. Cevher. Adversarially robust optimization with Gaussian processes. In *Advances in Neural Information Processing Systems*, pp. 5760–5770, 2018.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

J. Bradshaw, A. G. d. G. Matthews, and Z. Ghahramani. Adversarial examples, uncertainty, and transfer testing robustness in Gaussian process hybrid deep networks. *arXiv preprint arXiv:1707.02476*, 2017.

L. Cardelli, M. Kwiatkowska, L. Laurenti, N. Paoletti, A. Patane, and M. Wicker. Statistical guarantees for the robustness of Bayesian neural networks. *arXiv preprint arXiv:1903.01980*, 2019a.

L. Cardelli, M. Kwiatkowska, L. Laurenti, and A. Patane. Robustness guarantees for bayesian inference with gaussian processes. 33:7759–7768, 2019b.

N. Cauchi, L. Laurenti, M. Lahijanian, A. Abate, M. Kwiatkowska, and L. Cardelli. Efficiency through uncertainty: Scalable formal synthesis for stochastic hybrid systems. In *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '19, pp. 240–251, New York, NY, USA, 2019. ACM. ISBN 978-1-4503-6282-5. doi: 10.1145/3302504.3311805. URL http://doi.acm.org/10.1145/3302504.3311805.

D. Dua and C. Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

K. Grosse, D. Pfaff, M. T. Smith, and M. Backes. The limitations of model uncertainty in adversarial settings. *arXiv preprint arXiv:1812.02606*, 2018.

D. Hernández-Lobato, J. M. Hernández-Lobato, and P. Dupont. Robust multi-class Gaussian process classification. In *Advances in neural information processing systems*, pp. 280–288, 2011.

C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425, 2002.

X. Huang, M. Kwiatkowska, S. Wang, and M. Wu. Safety verification of deep neural networks. In *International Conference on Computer Aided Verification*, pp. 3–29. Springer, 2017.

G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pp. 97–117. Springer, 2017.

H.-C. Kim and Z. Ghahramani. Bayesian Gaussian process classification with the EM-EP algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28 (12):1948–1959, 2006.

H.-C. Kim and Z. Ghahramani. Outlier robust Gaussian process classification. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pp. 896–905. Springer, 2008.

Y. LeCun. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

T. P. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pp. 362–369. Morgan Kaufmann Publishers Inc., 2001.

A. Neumaier. Complete search in continuous global optimization and constraint satisfaction. *Acta numerica*, 13: 271–369, 2004.

H. Nickisch and C. E. Rasmussen. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9(Oct):2035–2078, 2008.

C. E. Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pp. 63–71. The MIT Press, 2004.

M. T. Ribeiro, S. Singh, and C. Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144. ACM, 2016.

J. B. Rosen and P. M. Pardalos. Global minimization of large-scale constrained concave quadratic problems by separable programming. *Mathematical Programming*, 34(2):163–174, 1986.

W. Ruan, X. Huang, and M. Kwiatkowska. Reachability analysis of deep neural networks with provable guarantees. *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 2651–2659, 2018.

M. T. Smith, K. Grosse, M. Backes, and M. A. Alvarez. Adversarial vulnerability bounds for gaussian process classification. *arXiv preprint arXiv:1909.08864*, 2019.

D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018.

C. K. Williams and D. Barber. Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.