Journal of Machine Learning Research 23 (2022) 1-55 Submitted 4/21; Revised 12/21; Published 4/22

Adversarial Robustness Guarantees for Gaussian Processes

Andrea Patanè ANDREA.PATANE@CS.OX.AC.UK
Department of Computer Science
University of Oxford
Oxford, OX1 3QG, United Kingdom

Arno Blaas ARNO@ROBOTS.OX.AC.UK
Department of Engineering Science
University of Oxford
Oxford, OX2 6ED, United Kingdom

Luca Laurenti L.LAURENTI@TUDELFT.NL
Department of Computer Science
University of Oxford
Oxford, OX1 3QG, United Kingdom

Luca Cardelli LUCA.CARDELLI@CS.OX.AC.UK
Department of Computer Science
University of Oxford
Oxford, OX1 3QG, United Kingdom

Stephen Roberts SJROB@ROBOTS.OX.AC.UK
Department of Engineering Science
University of Oxford
Oxford, OX2 6ED, United Kingdom

Marta Kwiatkowska MARTA.KWIATKOWSKA@CS.OX.AC.UK
Department of Computer Science
University of Oxford
Oxford, OX1 3QG, United Kingdom

Editor: John Cunningham

Abstract

Gaussian processes (GPs) enable principled computation of model uncertainty, making them attractive for safety-critical applications. Such scenarios demand that GP decisions are not only accurate, but also robust to perturbations. In this paper we present a framework to analyse adversarial robustness of GPs, defined as invariance of the model's decision to bounded perturbations. Given a compact subset of the input space $T \subseteq \mathbb{R}^d$, a point x^* and a GP, we provide provable guarantees of adversarial robustness of the GP by computing lower and upper bounds on its prediction range in T. We develop a branch-and-bound scheme to refine the bounds and show, for any $\epsilon > 0$, that our algorithm is guaranteed to converge to values ϵ-close to the actual values in finitely many iterations. The algorithm is anytime and can handle both regression and classification tasks, with analytical formulation for most kernels used in practice. We evaluate our methods on a collection of synthetic and standard benchmark data sets, including SPAM, MNIST and FashionMNIST. We study the effect of approximate inference techniques on robustness and demonstrate how our method can be used for interpretability. Our empirical results suggest that the adversarial robustness of GPs increases with accurate posterior estimation.

Keywords: Gaussian processes, adversarial robustness, non-linear optimisation, Bayesian learning, branch-and-bound methods

1. Introduction

Adversarial examples are input points intentionally crafted to trick a machine learning model into a misclassification. Imperceptible perturbations that can fool deep learning models in computer vision have been popularised by Szegedy et al. (2013) and, in the context of security, account for the growth in adversarial machine learning techniques, see review in (Biggio and Roli, 2018). Since test accuracy fails to account for the behaviour of a model in adversarial settings, algorithmic techniques for quantifying *adversarial robustness* of machine learning models are needed to aid their deployment in safety-critical scenarios. As a consequence, a number of methods that provide exact or approximate guarantees on the model output have been developed for neural networks, e.g., (Huang et al., 2017; Katz et al., 2017; Zhang et al., 2018).

Gaussian process (GP) models (Rasmussen and Williams, 2006) provide a flexible probabilistic framework for performing inference over functions, which integrates information from prior and data into a predictive posterior distribution that informs the optimal model decision. GPs are particularly attractive in view of their favourable analytical properties and support for Bayesian inference. One advantage of GPs compared to neural network models is that they support the computation of uncertainty over model predictions, which can then be propagated through the decision-making pipelines. Various notions of robustness have been investigated for Gaussian process models, such as robustness against outliers (Kim and Ghahramani, 2008) or against labelling errors (Hernández-Lobato et al., 2011). However, to the best of our knowledge, studies of adversarial 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) or limited to an analysis on the behaviour of the latent mean (Smith et al., 2019).

In this work, we develop a novel algorithmic framework to quantify the adversarial robustness of optimal predictions of Gaussian process models trained on a data set $\mathcal{D}$. To this end, we adapt the notion of adversarial robustness commonly employed for neural networks models to the GP setting, defined as the invariance of the decision in a small neighbourhood of a test point (Huang et al., 2017), and thus study the worst-case effect of bounded perturbations of the input on the GP optimal decision. We represent bounded perturbations by a compact subset of the input space $T \subseteq \mathbb{R}^d$ enclosing a test point $x^* \in \mathbb{R}^d$, and consider the prediction range of the GP over T. Similarly to (Ruan et al., 2018), we observe that, to provide provable guarantees on the model prediction over T, it suffices to compute the minimum and maximum of the reachable prediction range. Unfortunately, exact direct computation of the minimum and maximum 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). Instead, we approximate each extremum of the prediction range by lower and upper bounds. We show how such upper and lower bounds for the minimum and maximum prediction probabilities of the GP can be computed on any given compact set T, and then iteratively refine these bounds in a branch-and-bound algorithmic optimisation scheme until convergence to the minimum and maximum is obtained. 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 thus be used to provide guarantees) and ϵ-exact (the actual values are reached in finitely many steps up to an error ϵ selected a-priori). Our framework can handle robustness for both regression and classification tasks, with analytical formulation for most kernels used in practice, including generalised spectral kernels.

We implement the methods in Matlab and apply our approach to analyse the robustness profile of GP models on a synthetic two-dimensional data set, the SPAM data set, feature-based analysis of both binary and 3-class subsets of the MNIST and Fashion-MNIST (F-MNIST) data sets, and on a Water Quality multi-output regression data set (Džeroski et al., 2000).[1] In particular, we compare the guarantees computed by our method with the robustness estimation approximated by adversarial attack methods for GPs (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 GP model robustness. Across the four data sets analysed here, we observe that approximations based on Expectation Propagation (Minka, 2001) give more robust classification models than approximations based on Laplace approximation. We further find that GP robustness increases with the number of hyper-parameter training epochs, and that sparse GP model robustness generally increases with the number of training points (for a fixed number of inducing points). Finally, we show how our framework can be used to perform global interpretability analysis of GP predictions, highlighting differences over LIME (Ribeiro et al., 2016)

To the best of our knowledge, ours is the first comprehensive framework that provides methods to compute provable guarantees for the adversarial robustness of Gaussian process models. In summary, the paper presents the following contributions:

- We design a flexible framework for the bounding of the posterior mean and variance of GPs in compact subsets of the input space.
- Using the mean and variance bounds, we develop methods to lower- and upper-bound the minimum and maximum of a GP output over compact sets for the adversarial analysis of GPs in both classification and regression settings.
- We incorporate the bounding procedures in a branch-and-bound algorithmic optimisation scheme, which we show converges for any specified error $\epsilon > 0$ in finitely many steps.
- We empirically evaluate the robustness of a variety of GP models on four classification and a multi-output regression data sets, for different training regimes including sparse approximations, and demonstrate how our method can be used for global interpretability analysis of classification models.

A preliminary version of this work appeared in (Blaas et al., 2020). This paper extends previous work in several aspects. In (Blaas et al., 2020) we provide analytical bounds only for

[1] The code can be found at `https://github.com/andreapatane/check-GPclass`.

GP classification using a probit link function and consider GPs with the squared exponential kernels. Here, we also derive analytical bounds in the case of logit link function, show that regression models can be analysed using a subset of the methods developed for classification, and extend our framework to a class of kernel functions that satisfy certain smoothness conditions (see Section 4). Furthermore, we extend the experimental evaluation to show that our framework can be employed to analyse the robustness of sparse GP approximations, and additionally consider the Fashion-MNIST data set and a multi-output regression task.

This paper is structured as follows. In Section 2 we introduce background on GP regression and classification. The definition of adversarial robustness of GP models and the problem statements we consider are given in Section 3. Computation of adversarial robustness of a GP requires lower- and upper-bounding of the variation of the GP mean and variance in a neighbourhood of a test point. These bounds are presented in Section 4 and then employed in Section 5 to compute adversarial robustness for both (binary and multiclass) classification and regression. A branch-and-bound algorithm that incorporates the bounding methods is presented in Section 6, where we also show that it is guaranteed to converge to the true adversarial robustness of a GP model. Finally, empirical results on multiple data sets are discussed in Section 7.

1.1 Related Works

Following on from seminal work that drew attention to deep learning models being susceptible to adversarial attacks in computer vision (Szegedy et al., 2013) and security (Biggio and Roli, 2018), a range of techniques have been proposed for the analysis of adversarial robustness of machine learning models. The developed techniques mainly focus on neural networks and the prevailing approach is to compute worst-case guarantees on the model prediction at a given test point (Huang et al., 2017; Katz et al., 2017). Various approaches have been considered to compute such robustness measures, including constraint solving (Huang et al., 2017; Katz et al., 2017), optimisation (Ruan et al., 2018; Bunel et al., 2020), convex relaxation (Zhang et al., 2018), and abstract interpretation (Gehr et al., 2018). Such methods have also been extended to Bayesian Neural Networks (BNNs) (i.e., neural networks with a prior distribution over their weights and biases) with both sampling-based (Cardelli et al., 2019a; Wicker et al., 2021) and numerical (Wicker et al., 2020; Berrada et al., 2021) solution methods. However, these techniques rely on the parametric nature of neural networks, and therefore cannot be directly applied to GPs.

While various notions of robustness have been studied for Gaussian process models, such as robustness against outliers (Kim and Ghahramani, 2008) or against labelling errors (Hernández-Lobato et al., 2011), studies of adversarial robustness of GPs have been limited to heuristic analyses (Grosse et al., 2018; Bradshaw et al., 2017) and binary classification (Smith et al., 2019). In particular, in Smith et al. (2019), the authors give guarantees for GPs 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 GP framework. In contrast, our approach also considers multi-class classification and regression, takes into account the full posterior distribution, and allows for exact (up to $\epsilon > 0$) computation under any ℓ_p-norm.

Formal probabilistic guarantees for learning with GPs have been developed in the context of GP optimisation (Bogunovic et al., 2018) and GP regression (Cardelli et al., 2019b). Cardelli et al. (2019b) derive an upper bound on the probability that a function sampled from a trained GP is invariant to bounded perturbations at a specific test point, 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. We note that our problem formulation is different, and the methods developed in the above papers cannot be applied to classification with GPs due to its non-Gaussian nature. Further, our approach yields guarantees on the optimal model decision rather than on the latent GP posterior, 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). We also remark that the guarantees we provide in this paper are substantially different from those that can be obtained via randomized smoothing (Cohen et al., 2019). Randomized smoothing can "smooth" any base classifier at a given input point by perturbing the point with Gaussian isotropic noise. In contrast, our methods aim to directly quantify the robustness of the base machine learning model obtained by Gaussian process classification or regression.

2. Bayesian Learning with Gaussian Processes

This section provides background material on Gaussian process modelling for regression and classification. More information can be found in (Rasmussen and Williams, 2006). An $\mathbb{R}^m$-valued Gaussian process over a real-valued vector space $\mathbb{R}^d$ is a particular stochastic process $\boldsymbol{f} : \Omega \times \mathbb{R}^d \to \mathbb{R}^m$, where Ω is a suitable sample space, such that for every finite subset of input points their joint distribution under the GP is Gaussian. Namely, denoting with $\boldsymbol{f}(x) := \boldsymbol{f}(\cdot, x) : \Omega \to \mathbb{R}^m$ the random variable induced by the stochastic process in the input point x, and given a collection of input points $\mathbf{x} = [x^{(1)}, \dots, x^{(N)}]$, with $x^{(i)} \in \mathbb{R}^d$, a GP is such that $\boldsymbol{f}(\mathbf{x}) \sim \mathcal{N}(\mu(\mathbf{x}), \Sigma_{\mathbf{x},\mathbf{x}})$, where $\mu : \mathbb{R}^d \to \mathbb{R}^m$ is the mean function and $\Sigma : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{m^2}$ is the covariance (or *kernel*) function, which fully characterise the behaviour of the GP.

Consider now a data set $\mathcal{D} = \{(x^{(i)}, y^{(i)}) \mid x^{(i)} \in \mathbb{R}^d,\ y^{(i)} \in \mathcal{Y},\ i = 1, \dots, N\}$ for some input space $\mathbb{R}^d$ and output space $\mathcal{Y}$. We denote with $\mathbf{x} = [x^{(1)}, \dots, x^{(N)}]$ the aggregate vector of input points, and similarly $\mathbf{y} = [y^{(1)}, \dots, y^{(N)}]$ is the aggregate vector of output points. We let $\mathcal{Y}$ to be (a subset of) $\mathbb{R}^m$ for regression, and the discrete set $\{1, \dots, m\}$ in case of an m-class classification problem. Gaussian processes provide a probabilistic framework for performing inference over functions, where a prior is combined with data through an appropriate likelihood to obtain a posterior process that is consistent with the prior and data. In a Bayesian framework this is done by introducing a latent space $\mathcal{F} = \mathbb{R}^m$, and defining a GP prior $\boldsymbol{f}$ over the latter by instantiating a specific form for its mean, μ, and kernel, Σ, functions. The prior is updated to take into account the information contained in the data set $\mathcal{D}$ by means of the Bayes formula $p(f(\mathbf{x})|\mathcal{D}) \propto p(\mathbf{y}|f(\mathbf{x}))p(f(\mathbf{x}))$, where $p(\mathbf{y}|f(\mathbf{x}))$ denotes the likelihood function, resulting in the *posterior* distribution, $p(f(\mathbf{x})|\mathcal{D})$, over the latent space $\mathcal{F}$. Given a previously unseen point x^*, the *predictive posterior* distribution over its associated output y^* can be obtained by marginalising the posterior evaluated on x^* over the latent space, i.e., $p(y^*|\mathcal{D}) = \int_{\mathcal{F}} p(y^*|f(x^*))p(f(x^*)|\mathcal{D})df(x^*)$. For practical applications, we typically extract a point value, $\hat{y}(x^*)$, from the posterior predictive distribution $p(y^*|\mathcal{D})$ that satisfies specific criteria. In Bayesian decision theory, one proceeds by assuming a loss function, $L(y^*, \hat{y}^*)$, and minimising it with respect to the posterior distribution on the specific test point, that is,

$$\hat{y}(x^*) = \arg\min_{y \in \mathcal{Y}} \int_{\mathcal{Y}} L(y^*, y) p(y^*|\mathcal{D}) dy^*.$$

Since y is a continuous variable for regression models and a discrete variable for classification, different likelihood and loss functions are used in each case, resulting in different treatment for the posterior distribution and the model decision. Below, we review the specific details separately.

Regression For regression models we typically assume a Gaussian likelihood function with uncorrelated noise σ^2_{noise}, i.e., $p(y|f) = \mathcal{N}(y|f, \sigma^2_{\text{noise}} I_N)$. The posterior distribution over $\mathcal{F}$ is still Gaussian, and is characterised by the following inference equations for its posterior mean and variance:

$$\bar{\mu}(x^*) = \mu(x^*) + \Sigma_{x^*,\mathbf{x}} \mathbf{t} \tag{1}$$

$$\bar{\Sigma}(x^*) := \bar{\Sigma}_{x^*,x^*} = \Sigma_{x^*,x^*} - \Sigma_{x^*,\mathbf{x}} S \Sigma_{\mathbf{x},x^*}, \tag{2}$$

where S and $\mathbf{t}$ are computed using the conditioning formula for Gaussian distributions (Rasmussen and Williams, 2006). Namely, S is a matrix in $\mathbb{R}^{N\times N}$ with $S = (\Sigma_{\mathbf{x},\mathbf{x}} + \sigma^2 I_N)^{-1}$ and $\mathbf{t}$ is a vector in $\mathbb{R}^N$ with $\mathbf{t} = S(\mathbf{y} - \mu(\mathbf{x}))$. Furthermore, the predictive posterior distribution over $\mathcal{Y}$ has the same mean as the posterior and variance equal to that of the posterior plus the underlying noise σ^2_{noise}. Assuming a symmetric loss (e.g. the squared distance loss), which we refer to as the canonical loss for regression, the optimal *model decision* is simply given by the posterior mean, i.e., $\hat{y}(x^*) = \mu(x^*)$.

Classification For classification models, the likelihood is generally defined in terms of a sigmoid function $p(y = i|f) = \sigma_i(f)$, for $i \in \{1, \ldots, m\}$, as the *probit* or *softmax* function. Unfortunately, this does not result in a Gaussian posterior and is intractable. Instead, analytical approximations are applied to estimate a Gaussian distribution of the form $q(f|\mathcal{D}) = \mathcal{N}(f \mid \bar{\mu}(x^*), \bar{\Sigma}(x^*))$, which approximates the true distribution $p(f|\mathcal{D})$. In this paper we consider q derived using the *Laplace* approximation method (Williams and Barber, 1998), the *Expectation Propagation* (EP) method (Minka, 2001), as well as several *sparse approximation* techniques (Snelson and Ghahramani, 2005); more details can be found in Section 7. We observe that, in all these settings, the inference equations for $q(f|\mathcal{D})$ have the same form as those given in Equations (1) and (2), with S and $\mathbf{t}$ defined depending on the method chosen (Rasmussen and Williams, 2006).[2] Once the approximate posterior q has been computed, the predictive posterior distribution for class $i \in \{1, \ldots, m\}$ is

$$\pi_i(x^*) := p(y^* = i|\mathcal{D}) = \int_{\mathcal{F}} \sigma_i(\xi) \mathcal{N}(\xi | \bar{\mu}(x^*), \bar{\Sigma}(x^*)) d\xi. \tag{3}$$

[2] We remark that this form of inference equations is common for Gaussian approximations, as it results from conditioning formulas for multivariate Gaussian distribution. Our method can thus be applied in any situation in which Gaussian approximations are used (i.e., not necessarily resulting from Laplace or EP techniques).

Since Equation (3) includes a non-linear multi-dimensional integral, its solution cannot in general be found in closed form. However, when there are two classes, i.e. $\mathcal{Y} = \{1, 2\}$, it suffices to compute π_1 and then simply set $\pi_2 = 1 - \pi_1$. This allows us to simplify the latent variable space as uni-dimensional, so that $\xi \in \mathbb{R}$. Assuming standard 0-1 loss,[3] which we consider canonical for classification, the optimal *model decision* is the class that maximises the predictive posterior distribution, that is, $\hat{y}(x^*) = \arg\max_{i=1,\dots,m} p(y^* = i|\mathcal{D})$.

3. Problem Formulation

Let $\boldsymbol{f}$ be a Gaussian process trained on a data set $\mathcal{D}$. We wish to analyse its *adversarial robustness*, in the sense of studying the worst-case effect of bounded perturbations on the model's optimal decision $\hat{y}(x)$. For a generic test point x^*, we represent the possible adversarial perturbations by defining a compact neighbourhood T around x^*, and measure the changes in the decisions caused by limiting the perturbations to lie within T.

Definition 1 (Adversarial Robustness w.r.t. Model Decision) *Let $T \subseteq \mathbb{R}^d$ be a compact subset and $x^* \in T$. Consider a GP $\boldsymbol{f}$, a loss function L and the resulting optimal decision $\hat{y}(\cdot)$. Given an ℓ_p norm $||\cdot||$, we say that $\boldsymbol{f}$ is* δ-adversarially robust *in T at a point x^* with respect to the optimal decision induced by L iff*

$$||\hat{y}(x^*) - \hat{y}(x)|| \leq \delta \qquad \forall x \in T. \tag{4}$$

In the remainder of this paper, we will formulate a method for the worst-case analysis of the GP decision function $\hat{y}(x)$, which enables computing provable guarantees on whether a given $\boldsymbol{f}$ is adversarially robust around a test point x^* (that is, whether it satisfies the condition in Equation 4). Since regression and classification differ in how optimal decisions are made, which is reflected in the definition of the function $\hat{y}(\cdot)$, to simplify the presentation we will discuss the two cases separately.

3.1 Classification

For classification problems, adversarial robustness is customarily defined in terms of *invariance* of the decision over the neighbourhood of an input (Huang et al., 2017; Ruan et al., 2018). This can be obtained by selecting $\delta = 0$ in Definition 1, and noting that for classification $\hat{y}(x) = \arg\max_{i\in\{1,\dots,m\}} \pi_i(x)$.

Definition 2 (Adversarial Robustness in Classification) *Let $T \subseteq \mathbb{R}^d$ be a compact subset and $x^* \in T$. Consider a classification GP $\boldsymbol{f}$ and its predictive posterior distribution $\pi_i(x), i \in \{1, \dots, m\}$, defined as in Equation* (3)*. Then we say that $\boldsymbol{f}$ is* adversarially robust *in T at a point x^* iff*

$$\arg\max_{i\in\{1,\dots,m\}} \pi_i(x) = \arg\max_{i\in\{1,\dots,m\}} \pi_i(x^*) \qquad \forall x \in T. \tag{5}$$

Adversarial robustness therefore provides guarantees that the classification decision is not influenced by adversarial perturbations applied to x^*, as long as the perturbations are constrained to remain within T. Recall that the optimal decision for classification accounts for

[3] Our method is sufficiently general to accommodate other loss functions, e.g., weighted loss, which can be computed from the predictive posterior.

model uncertainty by moderating class probabilities with respect to the posterior distribution. In general, the outcome differs from the most likely class, because the decisions are affected by variance.

Similarly to (Ruan et al., 2018), we note that, in order to check the condition in Equation (5), it suffices to compute the minimum and maximum of the *prediction ranges* in T, i.e.:

$$\pi_{\min,i}(T) = \min_{x \in T} \pi_i(x) \qquad \pi_{\max,i}(T) = \max_{x \in T} \pi_i(x), \tag{6}$$

for $i = 1, \dots, m$. It is easy to see that the knowledge of $\pi_{\min,i}(T)$ and $\pi_{\max,i}(T)$ for all $i = 1, \dots, m$ can be used to provide guarantees on the absence of *adversarial attacks* of the model output, where an adversarial attack is a point $x \in T$ that is classified differently from x^*, that is, such that $\arg\max_{i \in \{1,\dots,m\}} \pi_i(x) \neq \arg\max_{i \in \{1,\dots,m\}} \pi_i(x^*)$. More specifically, by letting $\hat{y}^* = \arg\max_{i \in \{1,\dots,m\}} \pi_i(x^*)$ and defining the vector

$$\pi^*_i(T) = \begin{cases} \pi_{\max,i}(T) & \text{if } i \neq \hat{y}^* \\ \pi_{\min,i}(T) & \text{if } i = \hat{y}^*, \end{cases} \tag{7}$$

we can check whether the (stronger) condition $\arg\max_{i \in \{1,\dots,m\}} \pi^*_i(T) = \arg\max_{i \in \{1,\dots,m\}} \pi_i(x^*)$ holds. That is, in order to decide whether a GP classification model $\boldsymbol{f}$ satisfies Definition 2 around a point x^* we need to solve the following problem.

Problem 1 (Computation of Adversarial Prediction Ranges) *Let $T \subseteq \mathbb{R}^d$ be a compact subset. Consider a classification GP $\boldsymbol{f}$ and its predictive posterior distribution $\pi_i(x), i \in \{1, \dots, m\}$, defined as in Equation* (3)*. For $i = 1, \dots, m$, compute the adversarial prediction ranges for $\pi_i(x)$ in T, that is:*

$$\pi_{\min,i}(T) = \min_{x \in T} \pi_i(x) \qquad \pi_{\max,i}(T) = \max_{x \in T} \pi_i(x).$$

Unfortunately, the solution of Problem 1 requires solving $2m$ non-linear optimisation problems, for which no general solution method exists (Neumaier, 2004). We discuss the bounding of Problem 1 in Sections 5.1 and 5.2, and then show how to refine the bounds in Section 6.[4]

3.2 Regression

For regression models, since the output is a continuous variable, we define adversarial robustness in terms of a small, bounded variation of the decision over a compact neighbourhood T of a test point x^*. This follows from Definition 1, since for regression $\hat{y}(x) = \bar{\mu}(x)$. Formally, we have the following.

Definition 3 (Adversarial Robustness in Regression) *Let $T \subseteq \mathbb{R}^d$ be a compact subset, $x^* \in T$ and consider a GP $\boldsymbol{f}$. We say that $\boldsymbol{f}$ is* adversarially δ-robust *in T at a point x^* with respect to ℓ_p norm $|| \cdot ||$ iff*

$$||\bar{\mu}(x^*) - \bar{\mu}(x)|| \leq \delta \qquad \forall x \in T, \tag{8}$$

where $\bar{\mu}(x) = \mathbb{E}[\boldsymbol{f}(x)]$ is the posterior mean of the GP.

[4] While we focus on adversarial robustness w.r.t. the 0-1 loss, the computation of the prediction ranges poses a more general problem and the methods developed here can be used for classifiers associated to different loss functions (e.g., a weighted classification loss) through an appropriate definition of a vector in Equation (7).

This definition is analogous to the computation of the reachable set of outputs (or confidence values) for neural networks (Ruan et al., 2018). Since for a GP the mean corresponds to the maximum of the distribution, it thus follows, under the assumption of convergence, that it can be computed by a deterministic scheme that relies on regularised maximum likelihood estimation. We remark that, in contrast to classification, adversarial robustness for GP regression does not take into consideration model variance, and analyses only the most likely model among those obtained by Bayesian inference. As a consequence, the computation of adversarial robustness for regression reduces to the adversarial robustness of the posterior mean function. More specifically, Definition 3 can be checked once the value of $\sup_{x \in T} ||\bar{\mu}(x^*) - \bar{\mu}(x)||$ is known. That is, in order to decide whether a GP regression model $\boldsymbol{f}$ satisfies Definition 3 around a point x^* we need to solve the following problem.

Problem 2 (Computation of Posterior Mean Ranges) *Let $T \subseteq \mathbb{R}^d$ be a compact subset. Consider a regression GP $\boldsymbol{f}$ and its posterior mean $\mu_i(x), i \in \{1, \ldots, m\}$, defined as in Equation* (1)*. For $i = 1, \ldots, m$, compute the minimum and maximum of the posterior mean $\mu_i(x)$ in T, that is:*

$$\mu_{\min,i}(T) = \min_{x \in T} \mu_i(x) \qquad \mu_{\max,i}(T) = \max_{x \in T} \mu_i(x).$$

As for Problem 1, solving Problem 2 requires the solution of 2m non-linear optimisation problems. Similarly to classification, for regression we will develop a bound for Problem 2 in Section 5.3 and refine it through a branch-and-bound technique in Section 6.

3.3 Outline of the Approach

We now give an outline of a computational scheme to solve Problems 1 and 2 introduced in Sections 3.1 and 3.2, respectively, which will be developed in detail in Section 5. We first discuss classification, and then show how the regression scenario can be obtained as a special case of classification.

Classification For Problem 1, we devise a branch-and-bound optimisation scheme for the lower- and upper-bounding computation of the prediction ranges of a GP classification model over the input region T. In particular, for $i = 1, \ldots, m$, we first compute lower and upper bounds for $\pi_{\min,i}(T)$ and $\pi_{\max,i}(T)$, that is, we compute a set of real values $\pi^L_{\min,i}(T)$, $\pi^U_{\min,i}(T)$, $\pi^L_{\max,i}(T)$ and $\pi^U_{\max,i}(T)$ such that

$$\pi^L_{\min,i}(T) \leq \pi_{\min,i}(T) \leq \pi^U_{\min,i}(T) \tag{9}$$

$$\pi^L_{\max,i}(T) \leq \pi_{\max,i}(T) \leq \pi^U_{\max,i}(T). \tag{10}$$

We refer to $\pi^L_{\min,i}(T)$ and $\pi^U_{\max,i}(T)$ as *over-approximations* of the ranges, as they provide pessimistic estimation of the actual values of $\pi_{\min,i}(T)$ and $\pi_{\max,i}(T)$ for the purpose of adversarial robustness, and hence tighter guarantees. On the other hand, we refer to $\pi^U_{\min,i}(T)$ and $\pi^L_{\max,i}(T)$ as *under-approximations*, because they provide an optimistic estimation of the actual values that we want to compute.

The branch-and-bound scheme proceeds by iterative refinement of lower and upper bounds for the minimum and maximum of prediction ranges and is illustrated in Figure 1

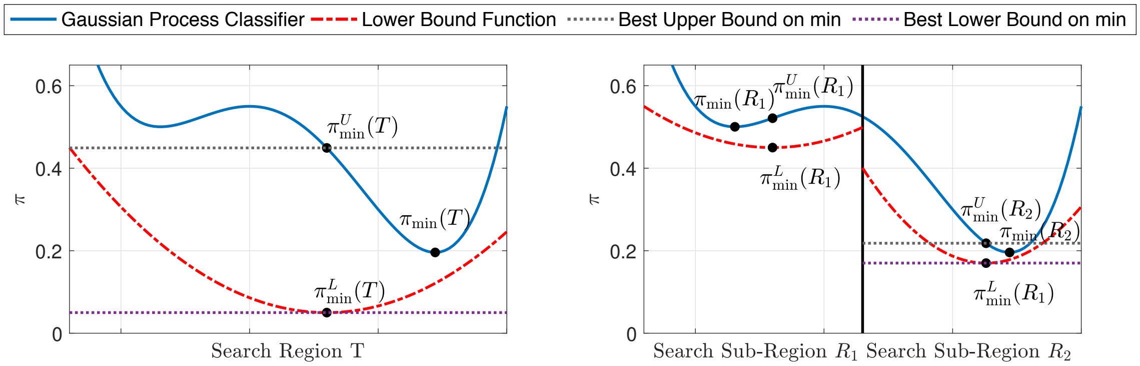

Figure 1: **Left:** Computation of upper and lower bounds on $\pi_{\min}(T)$, i.e., the minimum of the classification range on T. **Right:** The search region is repeatedly partitioned into sub-regions (only first partitioning is visualised), reducing the gap between best lower and upper bounds until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

for the simplified case of a GP with a single output value. First, we compute a lower- and an upper-bound function (the lower-bound function is depicted with a dashed red curve in Figure 1) for the GP output (solid blue curve) in the region T. We obtain this by deriving explicit bounds over the posterior mean and variance and relying on kernel bounding, which we then propagate through the predictive function. 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 GP predictive distribution on any point in T (a specific $\pi^U_{\min}(T)$ is depicted in Figure 1). Next, the region T is iteratively subdivided into sub-regions (R_1 and R_2 in the plot), for which we compute new (tighter) bounds by repeating the procedure previously applied to T. This procedure repeats until the bounds converge up to a desired tolerance $\epsilon > 0$. For each iteration, the bounds computed are valid, and therefore our method is anytime and can be terminated after a fixed number of iterations, at a cost of precision.

The bounds on the predictive distribution depend analytically on the maximum variations of the posterior mean and variance over the region T, which we therefore need to compute beforehand. For this purpose, in Section 4, we develop an optimisation framework for the computation of a set of real values $\mu^L_{T,i}$, $\mu^U_{T,i}$, $\Sigma^L_{T,i,j}$ and $\Sigma^U_{T,i,j}$ that under- and over-approximate the posterior mean and variance in T, i.e.

$$\mu^L_{T,i} \leq \min_{x \in T} \bar{\mu}_i(x) \qquad \mu^U_{T,i} \geq \max_{x \in T} \bar{\mu}_i(x) \tag{11}$$

$$\Sigma^L_{T,i,j} \leq \min_{x \in T} \bar{\Sigma}_{i,j}(x) \quad \Sigma^U_{T,i,j} \geq \max_{x \in T} \bar{\Sigma}_{i,j}(x), \tag{12}$$

for a general GP. We will utilise this framework in Section 5 to compute the desired upper and lower bounds on the ranges of the predictive posterior distribution.

Regression For regression, in Section 5.3 we develop a similar branch-and-bound approach to that for classification, except that (see Problem 2) we only need to consider the mean of the predictive posterior distribution (discussed in the next section).

4. Bounding Posterior Mean and Variance Function

In Section 5 we will develop a method for the computation of adversarial robustness guarantees for GPs. This method utilises upper and lower bounds on the variation of the mean and variance in the compact region T. Therefore, in this section, we formulate a general framework for the computation of lower and upper bounds on the posterior mean (Section 4.1) and variance (Section 4.2) of a GP model. Hence, we propose a method for the computation of $\mu^L_{T,i}$, $\mu^U_{T,i}$, $\Sigma^L_{T,i,j}$ and $\Sigma^U_{T,i,j}$ that satisfy Equations (11) and (12), which will be used in Section 5.

To simplify the presentation, we consider a GP with a single output value, eliding the explicit dependence on i. Since T is compact and therefore bounded, it can be covered by a finite union of hyper-boxes T_l, $l = 1, \dots, n_L$, i.e., $T \subseteq \bigcup_{l=1}^{n_L} T_l$, and furthermore the over-approximation error can be made vanishingly small. The bounds can thus be computed for each of the boxes, T_l, and the minimum and maximum across $l = 1, \dots, n_L$ can be used as bounds for the infimum and supremum over the original set T. Thus, without loss of generality, in the following we assume that T is a box in the input space, i.e., $T = [x^L, x^U]$.

We proceed by restricting the setting to kernel functions that admit an upper-bounding function U, which we propagate through inference equations to obtain bounds on mean and variance. Our construction admits analytical bounds for a large class of kernels.

Definition 4 (Bounded Kernel Decomposition) *Consider a one-dimensional kernel function $\Sigma : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and a compact set T. We say that (φ, ψ, U) is a* bounded decomposition *for Σ in T if $\Sigma_{x',x''} = \psi(\varphi(x', x''))$ and the following conditions are satisfied:*

1. *$\varphi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is linearly separable and continuously differentiable along each coordinate, so that $\varphi(x', x'') = \sum_{j=1}^{d} \varphi_j(x'_j, x''_j)$;*
2. *$\psi : \mathbb{R} \to \mathbb{R}$ is continuously differentiable and with a finite number of flex points;*
3. *U is an upper bounding function such that, for any vector of coefficients $\mathbf{c} = [c_1, \dots, c_N] \in \mathbb{R}^N$ and finite set of associated input points $[x^{(1)}, \dots, x^{(N)}]$, with $N \in \mathbb{N}$, we have that $U(\mathbf{c}) \geq \sup_{x \in T} \sum_{i=1}^{N} c_i \varphi(x, x^{(i)})$.*

Intuitively, a kernel decomposition separates the part of the kernel function that depends on the two inputs (represented by φ) with the part of the kernel that relates their dependence to the variance of the GP (represented by ψ). Assumptions 1 and 2 usually follow immediately from the smoothness of kernel functions.[5] Assumption 3 guarantees that we are able to upper bound the kernel function. The key idea is that, in view of the linearity of the inference equations for GPs, we can then propagate this bound through the inference equations to obtain bounds on the posterior mean and variance of the GP. We remark that, although not all kernel functions Σ admit kernel decomposition (for example if they are not smooth), the majority of kernel functions used in practice do. In Appendix B, we provide

[5] The finite number of flex points can be guaranteed, for example, by inspecting the function derivatives.

explicit computations for the *squared exponential, the Matern, rational quadratic*, and the *periodic* families of kernels, as well as *sums and products* thereof. Further, we describe the computation of bounded kernel decompositions for *generalised* (*stationary* and *non-stationary*) *spectral kernels*, which by means of Bochner's theorem can be shown to define a dense subset of the set of all the possible covariance functions (Samo and Roberts, 2015). In the remainder of the paper we assume that we are dealing with a kernel that admits a bounded decomposition.

Before computing bounds on mean and variance, we state the following result (proved in Appendix A) that ensures that the knowledge of a kernel decomposition allows us to compute a Lower Bounding Function (LBF) and an Upper Bounding Function (UBF) on the kernel values, linearly on φ.

Lemma 1 *Let Σ be a kernel and (φ, ψ, U) be a bounded decomposition. Let $T = [x^L, x^U] \subseteq \mathbb{R}^d$ be a box in the input space, then for every $\bar{x} \in \mathbb{R}^d$ there exists a set of real coefficients $\bar{a}_L$, $\bar{b}_L$, $\bar{a}_U$ and $\bar{b}_U$ such that:*

$$g_L(x) := \bar{a}_L + \bar{b}_L \varphi\left(x, \bar{x}\right) \leq \Sigma_{x,\bar{x}} \leq \bar{a}_U + \bar{b}_U \varphi\left(x, \bar{x}\right) =: g_U(x) \quad \forall x \in T.$$

In other words, g_L and g_U respectively represent an LBF and a UBF for the kernel function, given a fixed input point.

The above proposition allows us to explicitly compute coefficients of an LBF and a UBF on the overall kernel value, for any fixed point $\bar{x}$ in the input space. The main idea is that, since the posterior mean and variance are defined in terms of the summation and multiplication of pieces of the form $\Sigma_{x,x^{(i)}}$, for all $x^{(i)}$ in the training data set $\mathcal{D}$, we can compute LBFs and UBFs corresponding to each point in the training set, and propagate them through the inference equations for any unseen test point in T. By the design of the upper-bounding function U, we can then use the resulting LBFs and UBFs to bound the overall mean and variance functions. This is formalised in the following two subsections.

4.1 Bounding the Posterior Mean

Let $T \subseteq \mathbb{R}^d$ be an axis aligned hyper-rectangle. In this section we show how to compute a lower bound μ_T^L for the posterior mean function in T, i.e. such that $\mu_T^L \leq \inf_{x \in T} \bar{\mu}(x)$, for a kernel Σ with an associated bounded kernel decomposition (φ, ψ, U). Analogous techniques can be used to compute an upper bound μ_T^U by considering the function $-\bar{\mu}(x)$. We will then show that the bounds provided on the mean converge to the actual values as the diameter of the input region T tends to 0.

For simplicity, we assume that the prior mean function is identically null (Rasmussen and Williams, 2006). Then, the posterior mean (Equation 1) can be written down as

$$\bar{\mu}(x) = \Sigma_{x,\mathbf{x}} \mathbf{t} = \sum_{i=1}^{N} \Sigma_{x,x^{(i)}} t_i. \tag{13}$$

A lower bound for the mean function can thus be computed analytically, as shown in the following proposition.

Proposition 2 *Let Σ be a kernel with bounded decomposition (φ, ψ, U). Consider $a_L^{(i)}$, $b_L^{(i)}$, $a_U^{(i)}$ and $b_U^{(i)}$, the set of coefficients for LBFs and UBFs associated to each training point $x^{(i)}$, $i = 1, \ldots, N$, in an axis-aligned hyper-rectangle $T \subseteq \mathbb{R}^d$ (computed as for Lemma 1). Define*

$$(\bar{a}_L^{(i)}, \bar{b}_L^{(i)}) = \begin{cases} (a_L^{(i)}, b_L^{(i)}), & \textit{if } t_i \geq 0 \\ (a_U^{(i)}, b_U^{(i)}), & \textit{otherwise} \end{cases}.$$

Then

$$\mu_T^L := \sum_{i=1}^{N} \bar{a}_L^{(i)} - U([-\bar{b}_L^{(1)}, \ldots, -\bar{b}_L^{(N)}]) \leq \inf_{x \in T} \bar{\mu}(x).$$

Proof By construction of the coefficients $a_L^{(i)}$, $b_L^{(i)}$, $a_U^{(i)}$ and $b_U^{(i)}$ we have that

$$a_L^{(i)} + b_L^{(i)} \varphi(x, x^{(i)}) \leq \Sigma_{x,x^{(i)}} \leq a_U^{(i)} + b_U^{(i)} \varphi(x, x^{(i)}).$$

We can propagate the bounding functions through linear transformations (see Lemma 15 in Appendix A), so that we obtain

$$\Sigma_{x,x^{(i)}} t_i \geq \bar{a}_L^{(i)} + \bar{b}_L^{(i)} \varphi(x, x^{(i)}) \quad \forall x \in T. \tag{14}$$

By summing over the index i and taking the infimum of both sides of the inequality above we obtain

$$\inf_{x \in T} \sum_{i=1}^{N} \Sigma_{x,x^{(i)}} t_i \geq \sum_{i=1}^{N} \bar{a}_L^{(i)} + \inf_{x \in T} \sum_{i=1}^{N} \bar{b}_L^{(i)} \varphi(x, x^{(i)}). \tag{15}$$

We then observe that $\inf_{x \in T} \sum_{i=1}^{N} \bar{b}_L^{(i)} \varphi(x, x^{(i)}) = -\sup_{x \in T} \sum_{i=1}^{N} -\bar{b}_L^{(i)} \varphi(x, x^{(i)})$ and that according to the definition of U (point 3 of Definition 4) we have that $\sup_{x \in T} \sum_{i=1}^{N} -\bar{b}_L^{(i)} \varphi(x, x^{(i)}) \leq U([-\bar{b}_L^{(1)}, \ldots, -\bar{b}_L^{(N)}])$. By putting these two equations together we have that

$$\inf_{x \in T} \sum_{i=1}^{N} \bar{b}_L^{(i)} \varphi(x, x^{(i)}) \geq -U([-\bar{b}_L^{(1)}, \ldots, -\bar{b}_L^{(N)}]).$$

Finally, chaining the inequality above with that in Equation (15), we obtain

$$\inf_{x \in T} \sum_{i=1}^{N} \Sigma_{x,x^{(i)}} t_i \geq \sum_{i=1}^{N} \bar{a}_L^{(i)} - U([-\bar{b}_L^{(1)}, \ldots, -\bar{b}_L^{(N)}]),$$

which proves the proposition statement. ■

4.1.1 Convergence of Mean Bounds

We are able to show, importantly, that the bounds provided for the mean converge uniformly to the actual mean function, when the input region T is small enough. We first state the following lemma that proves that the LBFs and UBFs given by Lemma 1 yield converging bounds. The proof is provided in Appendix A.

Lemma 3 *Let Σ be a kernel with bounded decomposition (φ, ψ, U). Let $T = [x^L, x^U] \subseteq \mathbb{R}^d$, $\bar{x} \in T$, and let, for every axis-aligned hyper-rectangle $R \subseteq T$, $g_L^R(x)$ and $g_U^R(x)$ be the LBF and UBF computed on R for $\Sigma_{\bar{x},x}$ using Lemma 1. Then we have that g_L^R and g_U^R converge uniformly to $\Sigma_{\bar{x},x}$ as $diam(R) \to 0$.*

As the lower bound that we compute on the mean over T is obtained by summing together the individual LBFs g_L^R computed over each training point $x^{(i)}$ on R, it then follows that convergence of all LBFs g_L^R combined with a tight bounding function U implies convergence of the posterior mean lower bound, and similarly for the upper bound. This is formally shown in the proposition below.

Proposition 4 *Let Σ be a kernel with bounded decomposition (φ, ψ, U). Then bounds for the posterior mean μ_R^L and μ_R^U computed through the application of Proposition 2 converge if the bounds provided by U do so.*

Proof We discuss the case of μ_R^L; the arguments are analogous for μ_R^U.

We have that $\bar{\mu}(x) = \sum_{i=1}^N \Sigma_{x,x^{(i)}} t_i$. By Proposition 2, we obtain that:

$$\sum_{i=1}^N t_i \bar{g}_L^{(i)}(x) \leq \sum_{i=1}^N \Sigma_{x,x^{(i)}} t_i, \tag{16}$$

where $\bar{g}_L^{(i)}(x) = g_L^{(i)}(x)$ if $t_i \geq 0$ and $\bar{g}_L^{(i)}(x) = g_U^{(i)}(x)$ otherwise. For Lemma 3 we have that each $g_L^{(i)}$ converges uniformly to $\Sigma_{x,x^{(i)}}$ for each $x^{(i)}$. As t_i is a scalar quantity, we also have that each $t_i \bar{g}_L^{(i)}(x)$ converges uniformly to $\Sigma_{x,x^{(i)}} t_i$. Hence, we obtain that the bounds in Equation (16) converge uniformly as $\text{diam}(R) = r \to 0$, by virtue of being a linear combination of bounds that converge uniformly. The statement of the proposition then follows by the definition of U. ■

Therefore, convergence of the bounds for the posterior mean is reduced to a property of the kernel bounding function U. In Appendix B we show how explicit kernel decomposition can be computed for many kernel functions used in practice, where the derived functions U converge to the actual desired values (as further discussed in Appendix B.8).

4.2 Bounding the Posterior Variance

We now show how to find a lower and an upper bound for the posterior variance from Equation (2). For simplicity, we assume that $\Sigma_{x,x} = \sigma_p^2$ for all $x \in \mathbb{R}^d$,[6] so that we need

[6] This is always the case for stationary kernels. In the general case $\Sigma_{x,x}$ can be replaced by either its maximum or minimum value depending on whether we want to compute the minimum or the maximum of the posterior variance.

only to compute:

$$\min_{x \in T} \bar{\Sigma}(x) = \sigma_p^2 + \min_{x \in T} -\Sigma_{x,\mathbf{x}} S \Sigma_{x,\mathbf{x}}^T \tag{17}$$

$$\max_{x \in T} \bar{\Sigma}(x) = \sigma_p^2 - \min_{x \in T} \Sigma_{x,\mathbf{x}} S \Sigma_{x,\mathbf{x}}^T. \tag{18}$$

We first show how an upper bound for Equation (18) can be computed by means of convex quadratic programming.

4.2.1 Variance Upper Bound

The key observation is that S given in Equation (2) is a positive semi-definite matrix, so that the objective function to optimise in the case of the upper-bounding computation is a quadratic convex function on the variables $\Sigma_{x,\mathbf{x}}$ (but not on the optimisation variable x). In the following proposition, we show how the problem can be relaxed to obtain a quadratic convex program on the variable x and a suitably defined vector of slack variables.

Proposition 5 *Let Σ be a kernel with bounded decomposition (φ, ψ, U) and $T = [x^L, x^U]$ a box of the input space $\mathbb{R}^d$. Consider $a_L^{(i)}$, $b_L^{(i)}$, $a_U^{(i)}$ and $b_U^{(i)}$, a set of coefficients for LBFs and UBFs associated to each training point $x^{(i)}$, $i = 1, \ldots, N$, computed according to Lemma 1. Let $\mathbf{r} = [r^{(1)}, \ldots, r^{(N)}]$, $\varphi^{(i)}$, $\varphi_j^{(i)}$, for $i = 1, \ldots, N$ and $j = 1, \ldots, d$, be slack continuous variables. Let $\bar{\sigma}^2$ be the solution of the following convex quadratic programming problem:*

$$\begin{aligned}
&\min_{x \in T} \mathbf{r} S \mathbf{r}^T \\
&\textit{subject to:} \quad && r^{(i)} + a_L^{(i)} + b_L^{(i)} \varphi^{(i)} \leq 0 && i = 1, \ldots, N \\
& && r^{(i)} - a_U^{(i)} - b_U^{(i)} \varphi^{(i)} \leq 0 && i = 1, \ldots, N \\
& && a_{j,L}^{(i)} + b_{j,L}^{(i)} x_j - \varphi_j^{(i)} \leq 0 && i = 1, \ldots, N \quad j = 1, \ldots, d \\
& && \varphi_j^{(i)} - a_{j,U}^{(i)} - b_{j,U}^{(i)} x_j \leq 0 && i = 1, \ldots, N \quad j = 1, \ldots, d \\
& && \varphi^{(i)} = \sum_{j=1}^{d} \varphi_j^{(i)} && i = 1, \ldots, N \quad j = 1, \ldots, d.
\end{aligned}$$

Then $\Sigma_T^U := \sigma_p^2 - \bar{\sigma}^2$ is an upper bound for the posterior variance $\bar{\Sigma}(x)$ in T.

Proof By setting $\mathbf{r} = \Sigma_{x,\mathbf{x}}$ in the minimum computation in Equation (18), we obtain the objective function of the problem statement, $\mathbf{r} S \mathbf{r}^T$, which is quadratic on the vector variable $\mathbf{r}$. Since S is symmetric and positive semi-definite it follows that the objective function is a quadratic convex function in the slack variable vector $\mathbf{r}$. In order to obtain a convex program we then need to linearise the constraint $\mathbf{r} = \Sigma_{x,\mathbf{x}}$ We show how this is done for a generic index $i = 1, \ldots, N$.

We have that $r^{(i)} = \Sigma_{x,x^{(i)}} = \psi(\varphi(x, x^{(i)}))$. By Lemma 1 we obtain that

$$a_L^{(i)} + b_L^{(i)} \varphi\left(x, x^{(i)}\right) \leq \Sigma_{x,x^{(i)}} \leq a_U^{(i)} + b_U^{(i)} \varphi\left(x, x^{(i)}\right).$$

Hence, the dependence of ψ on the constraints can be linearised by considering the following over-approximation for the definition of $r^{(i)}$:

$$r^{(i)} + a_L^{(i)} + b_L^{(i)} \varphi\left(x, x^{(i)}\right) \leq 0$$
$$r^{(i)} - a_U^{(i)} - b_U^{(i)} \varphi\left(x, x^{(i)}\right) \leq 0.$$

The final step is to linearise the dependency over $\varphi\left(x, x^{(i)}\right)$. We introduce slack variables $\varphi^{(i)} = \varphi(x, x^{(i)})$, and $\varphi_j^{(i)} = \varphi_j(x_j, x_j^{(i)})$. For Assumption 1 of Definition 4 we have that $\varphi(x, x^{(i)}) = \sum_{j=1}^{d} \varphi_j(x_j, x_j^{(i)})$. Let $i \in \{1, \dots, N\}$ and let $j \in \{1, \dots, d\}$, then by applying Lemma 1 with $\psi := \varphi_j(\cdot, x_j^{(i)})$ and $\varphi := x$, we obtain that there exists a set of coefficients $a_{j,L}^{(i)}$, $b_{j,L}^{(i)}$, $a_{j,U}^{(i)}$ and $b_{j,U}^{(i)}$ such that:

$$a_{j,L}^{(i)} + b_{j,L}^{(i)} x_j \leq \varphi_j(x_j, x_j^{(i)}) \leq a_{j,U}^{(i)} + b_{j,U}^{(i)} x_j.$$

Hence, we can over-approximate the set of constraints $\varphi^{(i)} = \varphi(x, x^{(i)})$ and $\varphi_j^{(i)} = \varphi(x_j, x_j^{(i)})$ with the following set of linear constraints:

$$\begin{aligned} a_{j,L}^{(i)} + b_{j,L}^{(i)} x_j - \varphi_j^{(i)} &\leq 0 \\ \varphi_j^{(i)} - a_{j,U}^{(i)} - b_{j,U}^{(i)} x_j &\leq 0 \\ \varphi^{(i)} &= \sum_{j=1}^{d} \varphi_j^{(i)}. \end{aligned}$$

The formula for Σ_T^U then follows by the definition of minimum and by Equation (18). ■

Crucially, the proposition above casts the computation of the quantity Σ_T^U as the solution of a convex quadratic programming problem, for which ready-made solver software exists (Rosen and Pardalos, 1986).

4.2.2 Variance Lower Bound

The situation is, unfortunately, more complicated for the lower-bounding computation of $\min_{x \in T} -\Sigma_{x,\mathbf{x}} S \Sigma_{x,\mathbf{x}}^T$. In fact, though we can write down an optimisation problem akin to that of Proposition 5, since S is positive definite we have that $-S$ is negative definite, which implies that the function we want to optimise is quadratic concave. Thus, a number of local minima may exist, and simple quadratic optimisation is not guaranteed to yield the global solution. However, as we are interested in worst-case scenario analysis, we need to compute the global minimum. Unfortunately, this is an NP-hard problem, whose exact solution would be impractical to compute.

Instead, we apply the methods proposed in (Rosen and Pardalos, 1986) and proceed by computing a safe lower bound to the global minimum, that is, we want to compute a lower bound to the solution of:

$$
\begin{aligned}
&\min_{x \in T} -\mathbf{r} S \mathbf{r}^T && && (19)\\
&\text{subject to:} \quad r^{(i)} + a_L^{(i)} + b_L^{(i)} \varphi^{(i)} \leq 0 && i = 1, \dots, N \\
&\qquad r^{(i)} - a_U^{(i)} - b_U^{(i)} \varphi^{(i)} \leq 0 && i = 1, \dots, N \\
&\qquad a_{j,L}^{(i)} + b_{j,L}^{(i)} x_j - \varphi_j^{(i)} \leq 0 && i = 1, \dots, N \quad j = 1, \dots, d \\
&\qquad \varphi_j^{(i)} - a_{j,U}^{(i)} - b_{j,U}^{(i)} x_j \leq 0 && i = 1, \dots, N \quad j = 1, \dots, d \\
&\qquad \varphi^{(i)} = \sum_{j=1}^{d} \varphi_j^{(i)} && i = 1, \dots, N \quad j = 1, \dots, d.
\end{aligned}
$$

We highlight the details of the procedure applied to our specific setting below. First, we start by re-writing the constraints of the optimisation problem above in matrix form. Next, we introduce the aggregate variable vector $\mathbf{z} = [x_1, \dots, x_d, \varphi^{(1)}, \dots, \varphi^{(N)}, \varphi_1^{(1)}, \dots, \varphi_d^{(N)}]$. Since the constraints are linear, it is possible to define two matrices A_r and A_z such that the optimisation problem above can be equivalently written down as:

$$
\begin{aligned}
\min \; & -\mathbf{r}^T S \mathbf{r} \qquad (20)\\
\text{Subject to:} \quad & A_r \mathbf{r} + A_z \mathbf{z} \leq b \\
& \mathbf{r}^L \leq \mathbf{r} \leq \mathbf{r}^U \\
& \mathbf{z}^L \leq \mathbf{z} \leq \mathbf{z}^U,
\end{aligned}
$$

for suitably defined vectors b, $\mathbf{r}^L$, $\mathbf{r}^U$, $\mathbf{z}^L$, $\mathbf{z}^U$. Now, as S is symmetric and positive definite, there exists a matrix of eigenvectors $U = [\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(N)}]$ and a diagonal matrix Λ of the associated eigenvalues $\lambda^{(i)}$, for $i = 1, \dots, N$, such that $S = U \Lambda U^T$. We then define $\hat{r}^{(i)} = \mathbf{u}^{(i)} \cdot \mathbf{r}$ for $i = 1, \dots, N$, the rotated variables, and $\hat{\mathbf{r}}$ the aggregated vector of rotated variables, and compute their ranges $[\hat{r}^{(i),L}, \hat{r}^{(i),U}]$ by solving the following $2N$ linear programming problems:

$$
\begin{aligned}
\min / \max \quad & \mathbf{u}^{(i)} \cdot \mathbf{r} \\
\text{Subject to:} \quad & A_r \mathbf{r} + A_z \mathbf{z} \leq b \\
& \mathbf{r}^L \leq \mathbf{r} \leq \mathbf{r}^U \\
& \mathbf{z}^L \leq \mathbf{z} \leq \mathbf{z}^U.
\end{aligned}
$$

Implementing the change of variables into the optimisation problem defined in Equation (20), we obtain

$$
\begin{aligned}
\min \; & -\hat{\mathbf{r}}^T \Lambda \hat{\mathbf{r}} \\
\text{Subject to:} \quad & A_{\hat{r}} \hat{\mathbf{r}} + A_z \mathbf{z} \leq b \\
& \hat{\mathbf{r}}^L \leq \hat{\mathbf{r}} \leq \hat{\mathbf{r}}^U \\
& \mathbf{z}^L \leq \mathbf{z} \leq \mathbf{z}^U,
\end{aligned}
$$

where we have set $A_{\hat{r}} = A_r U$. We then notice that $\hat{\mathbf{r}}^T \Lambda \hat{\mathbf{r}} = \sum_{i=1}^{N} \lambda^{(i)} \hat{r}^{(i)2}$. Each summand is a simple one-dimensional quadratic function, for which we can find a linear LBF by relying on Lemma 1. Let $\alpha^{(i)}$ and $\beta^{(i)}$ be coefficients of such LBFs, then we have that $\alpha^{(i)} + \beta^{(i)} \hat{r}^{(i)} \leq -\lambda^{(i)} \hat{r}^{(i),2}$ for all $i = 1, \ldots, N$. Let $\boldsymbol{\beta} = [\beta^{(1)}, \ldots, \beta^{(N)}]$ and $\hat{\alpha} = \sum_{i=1}^{N} \alpha^{(i)}$, then we can lower-bound the optimisation problem defined in Equation (20) with the following linear programming problem:

$$\begin{aligned} &\min \left(\hat{\alpha} + \boldsymbol{\beta}^T \hat{\mathbf{r}} \right) \\ &\text{Subject to:} \quad A_{\hat{r}} \hat{\mathbf{r}} + A_z \mathbf{z} \leq b \\ &\qquad\qquad\quad\;\; \hat{\mathbf{r}}^L \leq \hat{\mathbf{r}} \leq \hat{\mathbf{r}}^U \\ &\qquad\qquad\quad\;\; \mathbf{z}^L \leq \mathbf{z} \leq \mathbf{z}^U. \end{aligned} \tag{21}$$

Hence, we have that a solution of the latter problem yields a lower bound for the solution of the optimisation problem in Equation (19). That is, we have proved the following statement.

Proposition 6 *Let $\underaccent{\tilde}{\sigma}^2$ be the solution of the linear programming problem defined in Equation* (21)*. Then $\Sigma_T^L := \sigma_p^2 + \underaccent{\tilde}{\sigma}^2$ is a lower bound for the posterior variance $\bar{\Sigma}(x)$ in T.*

4.2.3 Convergence of Variance Bounds

The convergence of the bounds computed for the variance to the actual values in hyper-rectangles $R \subseteq T$, with $\mathrm{diam}(R) \to 0$, is an immediate consequence of Lemma 3, and proceeds similarly to what we have shown for the posterior mean. In fact, the objective function for the upper bound (Proposition 5) is exact, and the over-approximation results only from the feasible region of the optimisation problem. This is relaxed by using LBFs and UBFs introduced in Lemma 1, so that their uniform convergence implies that the over-approximated feasible region converges to the actual one in the limit of the diameter $\mathrm{diam}(R)$ tending to 0. Similarly, for the lower-bounding of the variance the only difference arises from the use of Lemma 1, also for the lower-bounding of the optimisation function. However, this also converges to the actual objective function. Thus, the exact solution of both optimisation problems converges uniformly to the actual values, for R small enough. We summarise the discussion as the following proposition. The proof is a straightforward generalisation of the proof of Proposition 4 and is therefore omitted.

Proposition 7 *Let Σ be a kernel with bounded decomposition (φ, ψ, U). Then bounds on the posterior variance, Σ_R^L and Σ_R^U, computed through the application of Propositions 5 and 6 converge if the bounds provided by U do so.*

5. Bounds on Adversarial Robustness

In this section we show how the lower and upper bounds for the posterior mean and variance can be propagated through the predictive distribution of a GP to compute adversarial robustness guarantees, in the sense of ensuring invariance of the GP decision to perturbations constrained to a small neighbourhood around a test point. Thus developed bound will then be included in a branch-and-bound scheme in Section 6 for its iterative refinement. Recall from Section 3.1 and Problem 1 that for classification this reduces to bounding the minimum and maximum of the prediction ranges over the neighbourhood. We first discuss the bound for the two-class classification problem, and then show how the two-class bound can be extended to the multi-class setting. Finally, we discuss how to obtain guarantees for regression (Problem 2) as a particular case of the techniques derived for classification.

5.1 Bounds for Two-class Classification

As discussed in Section 2, for a two-class GP it suffices to consider a one-dimensional output space, which greatly simplifies the computations. Namely, we have that the predictive posterior distribution of Equation (3) evaluated on a generic point x can be simplified to one-dimensional integral, i.e.

$$\pi(x) = \int_{\mathbb{R}} \sigma(\xi) \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi, \tag{22}$$

where $\bar{\mu}$ and $\bar{\Sigma}$ are the posterior mean and variance functions, respectively, and $\sigma(\cdot)$ denotes the likelihood function. We give analytical bounds for the case where the likelihood function is either the probit function or the logistic sigmoid, which entail the majority of applications for GP classification (Rasmussen and Williams, 2006). A general bound based on latent space discretisation is discussed for the multi-class problem in Section 5.2, and can also be used for a generic two-class likelihood function.

Let μ_T^L, μ_T^U, Σ_T^L and Σ_T^U be lower and upper bounds for the posterior mean and variance of the GP, computed according to the methods discussed Section 4. We consider the function that describes the dependence of the predictive posterior distribution directly on the mean and variance by dropping their dependence on x:

$$\Pi(\mu, \Sigma) = \int_{\mathbb{R}} \sigma(\xi) \mathcal{N}(\xi | \mu, \Sigma) d\xi \quad \text{for} \quad \mu \in [\mu_T^L, \mu_T^U], \Sigma \in [\Sigma_T^L, \Sigma_T^U]. \tag{23}$$

Then, by definition of lower and upper bounds we have that:

$$\min_{\substack{\mu \in [\mu_T^L, \mu_T^U] \\ \Sigma \in [\Sigma_T^L, \Sigma_T^U]}} \Pi(\mu, \Sigma) \leq \min_{x \in T} \pi(x) \quad \text{and} \quad \max_{\substack{\mu \in [\mu_T^L, \mu_T^U] \\ \Sigma \in [\Sigma_T^L, \Sigma_T^U]}} \Pi(\mu, \Sigma) \geq \max_{x \in T} \pi(x),$$

that is, over-approximations of the prediction ranges can be found by optimising the function Π over the mean/variance box domain $[\mu_T^L, \mu_T^U] \times [\Sigma_T^L, \Sigma_T^U]$. In the next two subsections we show how this can be done depending on the particular form of the chosen likelihood σ.

5.1.1 Classification with the Probit Likelihood

We first consider the probit likelihood, i.e., $\sigma(\xi) = \Phi(\lambda \xi)$ is the cdf of the univariate standard Gaussian distribution scaled by $\lambda > 0$. In this case, the predictive distribution can be written down in closed form, which greatly simplifies the computation of the bounds:

$$\pi(x) = \Phi\left(\frac{\bar{\mu}(x)}{\sqrt{\lambda^{-2} + \bar{\Sigma}(x)}} \right).$$

We can use this explicit form to derive analytic upper and lower bounds by direct inspection of the predictive distribution derivatives with respect to the induced mean and variance variables. The following proposition provides a solution for Problem 1 in the case of two-class classification with the probit likelihood.

Proposition 8 *Consider a predictive posterior distribution $\pi(x)$ defined as in Equation (22), input box $T \subseteq \mathbb{R}^d$, and μ_T^L, μ_T^U, Σ_T^L and Σ_T^U, lower and upper bounds on the GP posterior variance, computed as detailed in Section 4. Let $\sigma(\xi) = \Phi(\lambda\xi)$, with $\lambda > 0$, then we have that*

$$\pi_{\min}^L(T) := \Phi\left(\frac{\mu_T^L}{\sqrt{\lambda^{-2} + \underline{\Sigma}^*}}\right) \leq \pi_{\min}(T) \tag{24}$$

$$\pi_{\max}(T) \leq \Phi\left(\frac{\mu_T^U}{\sqrt{\lambda^{-2} + \bar{\Sigma}^*}}\right) =: \pi_{\max}^U(T), \tag{25}$$

with

$$\underline{\Sigma}^* = \begin{cases} \Sigma_T^U & \textit{if} \quad \mu_T^L \geq 0 \\ \Sigma_T^L & \textit{otherwise} \end{cases} \qquad \bar{\Sigma}^* = \begin{cases} \Sigma_T^L & \textit{if} \quad \mu_T^U \geq 0 \\ \Sigma_T^U & \textit{otherwise.} \end{cases}$$

Proof We have:

$$\Pi(\mu, \Sigma) = \Phi\left(\frac{\mu}{\sqrt{\lambda^{-2} + \Sigma}}\right).$$

As Φ is monotonically increasing, it suffices to optimise for the argument $\phi(\mu, \Sigma) = \frac{\mu}{\sqrt{\lambda^{-2}+\Sigma}}$. By computing the partial derivatives it is easy to see that $\frac{\partial\phi(\mu,\Sigma)}{\partial\mu} > 0$ for all values of μ and Σ. Therefore, for every value of Σ the minimum is obtained for $\mu = \mu_T^L$. On the other hand, for the derivative wrt to Σ we have that:

$$\frac{\partial\phi(\mu_T^L, \Sigma)}{\partial\Sigma} \begin{cases} < 0 & \text{if} \quad \mu_T^L > 0 \\ = 0 & \text{if} \quad \mu_T^L = 0 \\ > 0 & \text{if} \quad \mu_T^L < 0 \end{cases}$$

as $\Sigma > 0$. Hence, given μ_T^L, we have that ϕ is monotonic in Σ and the proposition follows.■

5.1.2 Classification via Logistic Likelihood

We now consider the case where σ is defined as the logistic sigmoid. We will show that the minimum and maximum are to be found in the same extrema as for the probit likelihood. However, as the predictive distribution cannot be expressed in closed form (Rasmussen and Williams, 2006), we first show that the derivative of the predictive distribution can be computed by passing the sign of the derivative under the integral sign.

First, we note that upper and lower bounds on the variance Σ naturally induce upper and lower bounds on the standard deviation $s = \sqrt{\Sigma}$, which we denote s_T^L and s_T^U. By substituting s in the definition of Π in Equation (23), which yields $\Phi(\mu, s) := \Pi(\mu, s^2) = \Pi(\mu, \Sigma)$, and changing the integration variable to $t = (\xi - \mu)/s$, we have:

$$\Pi(\mu, \Sigma) =: \Phi(\mu, s) = \int_{\mathbb{R}} h(t, \mu, s)dt \quad \text{where} \quad h(t, \mu, s) = \sigma(st + \mu)\mathcal{N}(t|0, 1).$$

We now want to compute $\frac{\partial \Phi}{\partial \mu}$ and $\frac{\partial \Phi}{\partial s}$. It is easy to show that all the conditions to apply differentiation under the integral sign theorem are satisfied. Thus, we have:

$$\frac{\partial \Phi(\mu, s)}{\partial \mu} = \int \sigma'(st+\mu)\mathcal{N}(t|0,1)dt, \quad \frac{\partial \Phi(\mu, s)}{\partial s} = \int t\sigma'(st+\mu)\mathcal{N}(t|0,1)dt.$$

By relying on the derivatives, we can establish the following bounds. Specifically, the following proposition provides a solution for Problem 1 for two-class classification with the logistic likelihood.

Proposition 9 *Consider T, $\pi(x)$, μ_T^L, μ_T^U, $\underline{\Sigma}^*$ and $\bar{\Sigma}^*$ defined as in Proposition 8. Let $\sigma(\xi)$ be the sigmoid, then we have that:*

$$\pi_{\min}^L(T) := \Pi\left(\mu_T^L, \underline{\Sigma}^*\right) \leq \pi_{\min}(T) \tag{26}$$

$$\pi_{\max}(T) \leq \Pi\left(\mu_T^U, \bar{\Sigma}^*\right) =: \pi_{\max}^U(T). \tag{27}$$

Proof We show that the derivatives have the same sign as the probit, and then the proof follows as for probit. More specifically, we have that

$$\frac{\partial \Phi(\mu, s)}{\partial \mu} = \int \sigma'(st+\mu)\mathcal{N}(t|0,1)dt > 0,$$

since the sigmoid is a monotonically increasing function.

For the derivative with respect to s we want to show that

$$\frac{\partial \Phi(\mu, s)}{\partial s} = \int t\sigma'(st+\mu)\mathcal{N}(t|0,1)dt = \begin{cases} <0 & \text{if} \quad \mu > 0 \\ =0 & \text{if} \quad \mu = 0 \\ >0 & \text{if} \quad \mu < 0 \end{cases}.$$

The case for $\mu = 0$ is trivial. For the remaining cases we have:

$$\begin{aligned} &\int t\sigma'(st+\mu)\mathcal{N}(t|0,1)dt = \int_{-\infty}^{0} t\sigma'(st+\mu)\mathcal{N}(t|0,1)dt + \int_{0}^{+\infty} t\sigma'(st+\mu)\mathcal{N}(t|0,1)dt \\ &= \int_{0}^{+\infty} t\sigma'(-st+\mu)\mathcal{N}(t|0,1)dt + \int_{0}^{+\infty} t\sigma'(st+\mu)\mathcal{N}(t|0,1)dt \\ &= \int_{0}^{+\infty} t\left(\sigma'(\mu+st) - \sigma'(\mu-st)\right)\mathcal{N}(t|0,1)dt, \end{aligned}$$

and Lemma 16 (see Appendix A) can be applied to get the sign of the integral, since t and $\mathcal{N}(t|0,1)$ are always positive in $[0, +\infty)$. ■

Though the methods provided in this section suffice for the solution of Problem 1 in the two-class case, in Section 6 we will show how the bounds described above can be utilised to develop a branch-and-bound scheme for their refinement to ensure convergence to $\pi_{\min}(T)$ and $\pi_{\max}(T)$. Before we do this, we show in the next subsection how to compute bounds for multi-class classification.

5.2 Bounds for Multi-class Classification

In this section we generalise the results for two-class classification. Given a class index $i \in \{1, \dots, m\}$, we are interested in computing upper and lower bounds on the ith component of the predictive posterior distribution $\pi_i(x)$ (see Equation 3) for every $x \in T$, with T an axis-aligned hyper-rectangle in the input space. For simplicity, we explicitly tackle only the softmax likelihood, but similar arguments can be applied to the case of the multi-dimensional probit, as well as other likelihood functions that have similar monotonicity properties.

In the following we show that bounds on the multi-class predictive distribution can be computed by discretising the integral over the latent space.

Proposition 10 *Consider a predictive posterior distribution $\pi(x)$ defined as in Equation* (3)*, an input box $T \subseteq \mathbb{R}^m$, and define $\pi_{\min,i}(T)$ and $\pi_{\max,i}(T)$ as in Equation* (6)*. Let $\mathcal{S} = \{S_l = [a_l, b_l] \mid l \in \{1, \dots, M\}\}$ be a finite partition of the latent space $\mathcal{F} = \mathbb{R}^m$, with $[a_l, b_l] = [a_{l,1}, b_{l,1}] \times \dots \times [a_{l,m}, b_{l,m}]$. Then, for $i \in \{1, \dots, m\}$:*

$$\pi_{\min,i}(T) \geq \sum_{l=1}^{M} \sigma_i(\underline{\xi}^l) \min_{x \in T} \int_{S_l} \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi$$
$$\pi_{\max,i}(T) \leq \sum_{l=1}^{M} \sigma_i(\bar{\xi}^l) \max_{x \in T} \int_{S_l} \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi.$$

where

$$\underline{\xi}^l = [b_{l,1}, \dots, b_{l,i-1}, a_{l,i}, b_{l,i+1}, \dots, b_{l,m}]$$
$$\bar{\xi}^l = [a_{l,1}, \dots, a_{l,i-1}, b_{l,i}, a_{l,i+1}, \dots, a_{l,m}].$$

Proof We prove the statement for the minimum; the arguments for the maximum are analogous. By simple properties of integrals and definition of the minimum we have that:

$$\begin{aligned}\pi_{\min,i}(T) &= \min_{x \in T} \int_{\mathbb{R}^m} \sigma(\xi) \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi = \min_{x \in T} \sum_{l=1}^{M} \int_{S_l} \sigma(\xi) \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi \\ &\geq \sum_{l=1}^{M} \min_{x \in T} \int_{S_l} \sigma(\xi) \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi.\end{aligned}$$

Taking the partial derivatives of the softmax function with respect to coordinate $k \in \{1, \dots, m\}$ we have that:

$$\frac{\partial \sigma_i(\xi)}{\partial \xi_k} = \begin{cases} \sigma_i(\xi)(1 - \sigma_i(\xi)) & \text{if } k = i \\ -\sigma_i(\xi)\sigma_k(\xi) & \text{if } k \neq i \end{cases}$$

and hence we obtain that the i-th component of the softmax function is monotonically increasing along the direction i and monotonically decreasing along all the other dimensions $k \neq i$. Thus, its minimum in a generic axis-aligned hyper-rectangle $[a_{l,1}, b_{l,1}] \times \dots \times [a_{l,m}, b_{l,m}]$ will be found in the vertex defined as $\underline{\xi}^l = [b_{l,1}, \dots, b_{l,i-1}, a_{l,i}, b_{l,i+1}, \dots, b_{l,m}]$. Therefore, we have that the chain of inequalities above can be lower-bounded by computing the softmax on $\underline{\xi}^l$ and taking it outside of the integral computation, which yields:

$$\sum_{l=1}^{M} \sigma_i(\underline{\xi}^l) \min_{x \in T} \int_{S_l} \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi.$$

■

Summing up, Proposition 10 guarantees that, for all $x \in T$, $\pi_i(x)$ can be upper- and lower-bounded by solving M optimisation problems over a multi-dimensional Gaussian integral. In Proposition 11 below, we show that upper and lower bounds for the integral of a multi-dimensional Gaussian distribution, such as those appearing in Proposition 10, can be obtained by optimising a marginalised product of uni-dimensional Gaussian integrals over both the input and the latent space.

We first introduce the following notation. We denote with $\bar{\mu}_{i:j}(x)$ the subvector of $\bar{\mu}(x)$ containing only the components from i to j, with $i \leq j$, and similarly we define $\bar{\Sigma}_{i:k,j:l}(x)$ to be the submatrix of $\bar{\Sigma}(x)$ containing rows from i to k and columns from j to l, with $i \leq k$ and $j \leq l$.

Proposition 11 *Let $S = \prod_{i=1}^{m} [a_i, b_i] \subseteq \mathbb{R}^m$ be an axis-aligned hyper-rectangle in the latent space, and consider the posterior mean and variance functions $\bar{\mu}(x)$ and $\bar{\Sigma}(x)$. For $i \in \{1, \ldots, m-1\}$ and $f_{\mathcal{I}} \in \mathbb{R}^{m-i-1}$, define $\mathcal{I} = (i+1) : m$ and*

$$\bar{\mu}_i^f(x) = \bar{\mu}_i(x) - \bar{\Sigma}_{i,\mathcal{I}}(x) \bar{\Sigma}_{\mathcal{I},\mathcal{I}}^{-1}(x)(f_{\mathcal{I}} - \bar{\mu}_{\mathcal{I}}(x)) \tag{28}$$

$$\bar{\Sigma}_i^f(x) = \bar{\Sigma}_{i,i}(x) - \bar{\Sigma}_{i,\mathcal{I}}(x) \bar{\Sigma}_{\mathcal{I},\mathcal{I}}^{-1}(x) \bar{\Sigma}_{i,\mathcal{I}}^T(x). \tag{29}$$

Let $S_{\mathcal{I}} = \prod_{j=i+1}^{m} [a_i, b_i]$, then we have that:

$$\max_{x \in T} \int_S \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi \leq \max_{x \in T} \int_{a_m}^{b_m} \mathcal{N}(\xi | \bar{\mu}_m(x), \bar{\Sigma}_{m,m}(x)) d\xi \prod_{i=1}^{m-1} \max_{\substack{x \in T \\ f \in S_{\mathcal{I}}}} \int_{a_i}^{b_i} \mathcal{N}(\xi | \bar{\mu}_i^f(x), \bar{\Sigma}_i^f(x)) d\xi \tag{30}$$

$$\min_{x \in T} \int_S \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi \geq \min_{x \in T} \int_{a_m}^{b_m} \mathcal{N}(\xi | \bar{\mu}_m(x), \bar{\Sigma}_{m,m}(x)) d\xi \prod_{i=1}^{m-1} \min_{\substack{x \in T \\ f \in S_{\mathcal{I}}}} \int_{a_i}^{b_i} \mathcal{N}(\xi | \bar{\mu}_i^f(x), \bar{\Sigma}_i^f(x)) d\xi. \tag{31}$$

Proof We consider the case of the minimum; the maximum follows similarly.

Consider the latent posterior process $\boldsymbol{f}$, whose mean and variance function we denote with $\bar{\mu}(x)$ and $\bar{\Sigma}(x)$. Then, we have

$$\begin{aligned}
&\min_{x\in T}\int_S \mathcal{N}(\xi|\bar{\mu}(x),\bar{\Sigma}(x))d\xi = \min_{x\in T} P(\boldsymbol{f}(x)\in S) = \min_{x\in T} P(a_i \leq \boldsymbol{f}_i(x) \leq b_i, i=1,\dots,m) = \\
&\quad \min_{x\in T} P(a_m \leq \boldsymbol{f}_m(x) \leq b_m) \prod_{i=1}^{m-1} P(a_i \leq \boldsymbol{f}_i(x) \leq b_i | \boldsymbol{f}_{\mathcal{I}}(x) \in S_{\mathcal{I}}) \geq \\
&\qquad \text{(By Lemma 17 included in the Appendix A)} \\
&\quad \min_{x\in T} P(a_m \leq \boldsymbol{f}_m(x) \leq b_m) \prod_{i=1}^{m-1} \min_{f_{\mathcal{I}}\in S_{\mathcal{I}}} P(a_i \leq \boldsymbol{f}_i(x) \leq b_i | \boldsymbol{f}_{\mathcal{I}}(x) = f_{\mathcal{I}}) \geq \\
&\quad \min_{x\in T} P(a_m \leq \boldsymbol{f}_m(x) \leq b_m) \prod_{i=1}^{m-1} \min_{\substack{x\in T \\ f_{\mathcal{I}}\in S_{\mathcal{I}}}} P(a_i \leq \boldsymbol{f}_i(x) \leq b_i | \boldsymbol{f}_{\mathcal{I}}(x) = f_{\mathcal{I}}).
\end{aligned}$$

Notice that, for each $i \in \{1,\dots,m-1\}$, $P(a_i \leq \boldsymbol{f}_i(x) \leq b_i | \boldsymbol{f}_{\mathcal{I}}(x) = f_{\mathcal{I}})$ is the integral of a uni-dimensional Gaussian random variable conditioned on a jointly Gaussian random variable. The statement of the proposition then follows by the application of the conditioning equations for Gaussian distributions. ■

Proposition 11 reduces the computation of the multi-class bounds to a product of extrema computations over univariate Gaussian distributions. To solve this, we first need to compute lower and upper bounds for the conditional latent mean and the conditional latent variance defined in Equations (28) and (29). Observe that Equations (28) and (29) can be expressed as a rational function in the entries of the mean vector, variance matrix and latent variable vector. We can thus propagate the upper and lower bound of each entry from the mean vector and covariance matrix down through the rational function equations by simple interval bound propagation techniques, which results in an upper and lower bound on $\bar{\mu}_i^f(x)$ and $\bar{\Sigma}_i^f(x)$ for $x \in T$ and $f \in S_{\mathcal{I}}$, which we denote with $\mu_{i,T}^{L,f}$, $\mu_{i,T}^{U,f}$, $\Sigma_{i,T}^{L,f}$ and $\Sigma_{i,T}^{U,f}$. This process can then be iterated backward from $i = m$ to $i = 1$, up until all the required bounds are computed. Unfortunately, because of the need to symbolically compute a matrix inversion, the explicit formulas for the computation of $\mu_{i,T}^{L,f}$, $\mu_{i,T}^{U,f}$, $\Sigma_{i,T}^{L,f}$ and $\Sigma_{i,T}^{U,f}$ in general are rather convoluted and long (though still in the form of a simple ratio between polynomials).

Once those bounds are computed, we rely on the following lemma for the solution of the optimisation problem over the Gaussian integrals.

Lemma 12 *Consider $S_{\mathcal{I}}$, a_i, b_i, $\bar{\mu}_i^f(x)$ and $\bar{\Sigma}_i^f(x)$ defined as in Proposition 11, an input box $T \subseteq \mathbb{R}^d$, and $\mu_{i,T}^{L,f}$, $\mu_{i,T}^{U,f}$, $\Sigma_{i,T}^{L,f}$ and $\Sigma_{i,T}^{U,f}$, lower and upper bounds on $\bar{\mu}_i^f(x)$ and $\bar{\Sigma}_i^f(x)$ in T computed as discussed above. Define $\zeta := [x, f]$ and its input region as $Z = T \times S_{\mathcal{I}}$.*

Let $i = \{1, \dots, m\}$*,* $\mu_i^c = \frac{a_i + b_i}{2}$ *and* $\Sigma_i^c(\mu) = \frac{(\mu - a_i)^2 - (\mu - b_i)^2}{2 \log \frac{\mu - a_i}{\mu - b_i}}$*. Then it holds that:*

$$\begin{aligned}
&\max_{\zeta \in Z} \int_{a_i}^{b_i} \mathcal{N}(\xi | \bar{\mu}_i^f(x), \bar{\Sigma}_i^f(x)) d\xi \leq \int_{a_i}^{b_i} \mathcal{N}(\xi | \bar{\mu}^*, \bar{\Sigma}^*) d\xi \\
&= \frac{1}{2} \left(\mathrm{erf}\left(\frac{\bar{\mu}^* - a_i}{\sqrt{2 \bar{\Sigma}^*}} \right) - \mathrm{erf}\left(\frac{\bar{\mu}^* - b_i}{\sqrt{2 \bar{\Sigma}^*}} \right) \right) \qquad (32) \\
&\min_{\zeta \in Z} \int_{a_i}^{b_i} \mathcal{N}(\xi | \bar{\mu}_i^f(x), \bar{\Sigma}_i^f(x)) d\xi \geq \int_{a_i}^{b_i} \mathcal{N}(\xi | \underline{\mu}^*, \underline{\Sigma}^*) d\xi \\
&= \frac{1}{2} \left(\mathrm{erf}\left(\frac{\underline{\mu}^* - a_i}{\sqrt{2 \underline{\Sigma}^*}} \right) - \mathrm{erf}\left(\frac{\underline{\mu}^* - b_i}{\sqrt{2 \underline{\Sigma}^*}} \right) \right), \qquad (33)
\end{aligned}$$

where we have:

$$\bar{\mu}^* = \operatorname*{arg\,min}_{\mu \in [\mu_{i,T}^{L,f}, \mu_{i,T}^{U,f}]} |\mu_i^c - \mu|, \quad \bar{\Sigma}^* = \begin{cases} \Sigma_{i,T}^{L,f} & \text{if} \quad \bar{\mu}^* \in [a_i, b_i] \\ \operatorname{arg\,min}_{\Sigma \in [\Sigma_{i,T}^{L,f}, \Sigma_{i,T}^{U,f}]} |\Sigma_i^c(\bar{\mu}^*) - \Sigma| & \text{otherwise} \end{cases}$$

$$\underline{\mu}^* = \operatorname*{arg\,max}_{\mu \in [\mu_{i,T}^{L,f}, \mu_{i,T}^{U,f}]} |\mu_i^c - \mu|, \quad \underline{\Sigma}^* = \operatorname*{arg\,min}_{\Sigma \in \{\Sigma_{i,T}^{L,f}, \Sigma_{i,T}^{U,f}\}} [\mathrm{erf}(b_i | \underline{\mu}^*, \Sigma) - \mathrm{erf}(a_i | \underline{\mu}^*, \Sigma)].$$

By iterating the computation of Lemma 12 for each integral in Proposition 11, we obtain the bounds on the predictive distribution. The discretised bound can also be used for two-class classification, in cases where a likelihood function different from the probit and the logistic sigmoid is desired.

5.3 Bounds for Regression

While computing adversarial robustness guarantees for classification models involves the computation of upper and lower bounds on the GP posterior predictive distribution, the analysis is much simpler for regression. As stated in Section 2 and formalised in Problem 2, for the canonical loss function the optimal decision corresponds to the posterior latent mean function $\bar{\mu}(x)$ of the posterior GP distribution, whose computation is given in Section 3. Guarantees over the decision can then be made simply by relying on upper and lower bounds for the mean function, that is, $\mu_{i,T}^L$ and $\mu_{i,T}^U$ for every $i = 1, \dots, m$, which makes over-approximation of Definition 3 much faster and simpler in practice.

Proposition 13 *Consider a box* $T \subseteq \mathbb{R}^d$ *of the input space, a test point* $x^* \in T$*, an* ℓ_p *metric* $|| \cdot ||$ *in the output space* $\mathbb{R}^m$ *and a* $\delta > 0$*. Let* $\bar{\mu}$ *be the predictive posterior mean, and* $\mu_{i,T}^L$ *and* $\mu_{i,T}^U$*, for every* $i = 1, \dots, m$*, its upper and lower bounds computed according to Proposition 2. Define* μ_T^* *as the vector of entries:*

$$\mu_{T,i}^* = \begin{cases} \mu_{i,T}^L & \text{if} \quad |\bar{\mu}_i(x^*) - \mu_{i,T}^L| \geq |\bar{\mu}_i(x^*) - \mu_{i,T}^U| \\ \mu_{i,T}^U & \text{if} \quad |\bar{\mu}_i(x^*) - \mu_{i,T}^L| < |\bar{\mu}_i(x^*) - \mu_{i,T}^U|. \end{cases}$$

Then:

$$\sup_{x \in T} ||\bar{\mu}(x^*) - \bar{\mu}(x)|| \leq ||\bar{\mu}(x^*) - \mu_T^*||.$$

Consequently, if:

$$||\bar{\mu}(x^*) - \mu^*_T|| \leq \delta$$

then the GP is δ-robust in x^* w.r.t. T and norm $||\cdot||$.

Proof By construction, we have that $\bar{\mu}_i(x) \in [\mu^L_{i,T}, \mu^U_{i,T}]$ for every $x \in T$. Hence, by monotonicity of ℓ_p norms along the coordinate directions and by definition of $\bar{\mu}(x)$, it follows that $\sup_{x\in T} ||\bar{\mu}(x^*) - \bar{\mu}(x)|| \leq ||\bar{\mu}(x^*) - \mu^*_T||$. Thus:

$$\delta \geq ||\bar{\mu}(x^*) - \mu^*_T|| \geq \sup_{x\in T} ||\bar{\mu}(x^*) - \bar{\mu}(x)|| \geq ||\bar{\mu}(x^*) - \bar{\mu}(x)||, \quad \text{for} \quad x \in T.$$

In particular, $\delta \geq ||\bar{\mu}(x^*) - \bar{\mu}(x)||$, which is equivalent to Definition 3. ■

6. Branch-and-Bound Algorithm

In this section we formulate a branch-and-bound algorithmic scheme that incorporates the lower- and upper-bounding procedures for Gaussian process models introduced in Section 5 and prove its convergence up to any a-priori specified $\epsilon > 0$. For simplicity of exposition, we restrict the discussion to two-class classification, noting that the multi-class classification and regression problems follow analoguously by substituting appropriate bounding procedures. The main idea behind branch-and-bound optimisation is to alternate between bounding the function we are interested in optimising in our input box T and splitting T into smaller boxes, i.e., *candidate search regions*, on which we compute the bound in the next iteration. This procedure creates a search tree, in which descending depth implies smaller search regions. The intuition is that, as we explore the branch-and-bound search tree depth-first, the search regions become smaller, so that the bounds get closer to the true function, and we thus slowly converge to the actual optimum. By computing lower and upper bounds on the quantity of interest, we are then able to prune our search tree for regions in which optimal values cannot occur.

We now describe the proposed branch-and-bound scheme for the computation of lower and upper bounds for $\pi_{\min}(T)$ derived in Section 5.1, which is summarised in Algorithm 1. After initialising $\pi^L_{\min}(T)$ and $\pi^U_{\min}(T)$ to trivial values and the exploration regions stack $\mathbf{R}$ to the singleton $\{T\}$, the main optimisation loop is entered until convergence (lines 2–10). Among the regions in the stack, we select the region R with the most promising lower bound (line 3). After bounding posterior mean and variance in R (line 4), we refine its lower bound using Proposition 8 for the probit likelihood and Proposition 9 for the logistic sigmoid likelihood (line 5), as well as its upper bound 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). We perform the split by randomly selecting an index $j \in \{1, \dots, d\}$ from the input dimensions, and by splitting R at the mid-point along the jth dimension. Finally, the freshly computed bounds local to $R \subseteq T$ are used to update the global bounds for T (line 10). Namely, $\pi^L_{\min}(T)$ is updated to the smallest value among the

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

Input: Input space subset T; error tolerance $\epsilon > 0$; latent mean/variance functions $\bar{\mu}(\cdot)$ and $\bar{\Sigma}(\cdot)$.

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 $[\mu^L_R, \mu^U_R]$ and $[\Sigma^L_R, \Sigma^U_R]$ applying Propositions 2, 5 and 6 over $R$
 5:     Compute $\pi^L_{\min}(R)$ from $[\mu^L_R, \mu^U_R]$ and $[\Sigma^L_R, \Sigma^U_R]$ using Proposition 8 or 9 resp.
 6:     Find $\pi^U_{\min}(R)$ by evaluating $\pi(x)$ 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
 9:         Use $\pi^L_{\min}(R), \pi^U_{\min}(R)$ as initial bounds for both sub-regions $R_1, R_2$
10:     end if
11:     Update $\pi^L_{\min}(T)$ and $\pi^U_{\min}(T)$ with current best bounds found
12: end while
13: return $[\pi^L_{\min}(T), \pi^U_{\min}(T)]$
```

$\pi^L_{\min}(R)$ values for $R \in \mathbf{R}$, while $\pi^U_{\min}(T)$ is set to the lowest observed value yet explicitly computed in line 6.

We remark that to derive a branch-and-bound scheme for multi-class classification (respectively, regression) it suffices to replace line 5 in Algorithm 1 with the bounding methods of Section 5.2 (respectively, Section 5.3).

Computation of Under-approximations As discussed in Section 3.3, in order to obtain valid values for $\pi^U_{\min}(T)$ and $\pi^L_{\max}(T)$ it suffices to evaluate the GP posterior predictive distribution in any point of T. However, the closer $\pi^U_{\min}(T)$ and $\pi^L_{\max}(T)$ are to $\pi_{\min}(T)$ and $\pi_{\max}(T)$, respectively, the faster a branch-and bound-algorithm will converge. By solving the optimisation problems associated to μ^L_T, μ^U_T, Σ^L_T and Σ^U_T, we obtain four extrema points in T on which the GP assumes the optimal values for the posterior mean and variance bounds. As these points belong to T and provide extreme points for the latent function, they are promising candidates for the evaluation of $\pi^U_{\min}(T)$ and $\pi^L_{\max}(T)$. We thus evaluate the GP predictive posterior distribution on all four extremal points and select the one that yields the best bound.

Convergence By construction it is clear that, if Algorithm 1 terminates, the resulting values over- and under-approximate the true value $\pi_{\min}(T)$ with a known error $\epsilon > 0$. We now show, by relying on the theory of convergence for branch-and-bound algorithms, that the loop of lines $2 - 9$ terminates in a finite number of iterations. In particular, to prove convergence of a branch-and-bound scheme up to an error $\epsilon > 0$ it suffices to show that the two following conditions hold (Balakrishnan et al., 1991):

1. *Consistency Condition*: $\pi^L_{\min}(R) \leq \pi_{\min}(R) \leq \pi^U_{\min}(R) \qquad \forall R \subseteq T$.

2. *Uniform Convergence*: $\forall \epsilon > 0 \;\; \exists\, r > 0 \;$ s.t. $\forall R \subseteq T$ with $\text{diam}(R) \leq r \Rightarrow |\pi^U_{\min}(R) - \pi^L_{\min}(R)| \leq \epsilon$.

Intuitively, the first condition ensures that the computed bounds are consistent across all the subsets of the initial input region T. This is clearly satisfied by construction, see Section 5.1. The second condition enforces that the lower and the upper bounds converge uniformly to each other as we reduce the maximum diameter of the branch-and-bound search region to zero. In the following theorem we show that the bound based on latent space discretisation has the uniform convergence property and converges in finitely many steps. Consequently, as the analytical bounds that we compute for the probit and the logistic function are tighter than for discretisation, they will also converge. For simplicity of exposition, we prove convergence for two-class classification, which also captures regression as a special case; we provide details below for how the result can be generalised to the multi-class case.

Theorem 14 *Let T be a box in the input space $\mathbb{R}^d$. Consider a two-class classification GP with posterior mean and variance given by $\bar{\mu}(x)$ and $\bar{\Sigma}(x)$. Assume that μ_R^L, μ_R^U, Σ_R^L, Σ_R^U are bounding functions for the posterior mean and variance such that:*

$$\mu_R^L \to \min_{x \in R} \bar{\mu}(x), \quad \mu_R^U \to \max_{x \in R} \bar{\mu}(x), \quad \Sigma_R^L \to \min_{x \in R} \bar{\Sigma}(x), \quad \Sigma_R^U \to \max_{x \in R} \bar{\Sigma}(x) \tag{34}$$

whenever diam$(R) \to 0$. Then, for $\epsilon > 0$, there exists a partition of the latent space $\mathcal{S}$ and $\bar{r} > 0$ such that, for every $R \subseteq T$ with diam$(R) < \bar{r}$, it holds that

$$|\pi_{\min}^U(R) - \pi_{\min}^L(R)| \leq \epsilon. \tag{35}$$

Proof Consider an $\epsilon > 0$, and a generic axis-aligned hyper-rectangle $R \subseteq T$ of diameter $\mathrm{diam}(R) := r > 0$ less than a fixed $\bar{r} > 0$. We want to find a value for $\bar{r}$ for which the condition in Equation (35) is surely met. We start by observing that $\pi_{\min}^U(R)$ is defined by computing the predictive posterior distribution on a fixed point of R. Let $\bar{x} \in R$ be such a point, and define $\bar{\mu} := \bar{\mu}(\bar{x})$ and $\bar{\Sigma} := \bar{\Sigma}(\bar{x})$, then we have that

$$\pi_{\min}^U(R) = \int \sigma(\xi) \mathcal{N}(\xi | \bar{\mu}, \bar{\Sigma}) d\xi.$$

Now consider a generic $M > 0$; we define the discretisation of the latent space $\mathcal{S}_M = \{[a_l, b_l] \mid l = 1 \ldots, M\}$ with the following equations:

$$\begin{aligned} a_1 &= -\infty \\ b_l &= \sigma^{-1}\left(\sigma(a_l) + \frac{1}{M}\right) \qquad && l = 1, \ldots, M \\ a_{l+1} &= b_l && l = 1, \ldots, M, \end{aligned}$$

that is, we discretise the y-axis into M equally distanced intervals and map that discretisation back to the x-axis through the link function, σ^{-1}. We then have that the left-hand-side of Equation (35) can be written explicitly as

$$\left| \int \sigma(\xi) \mathcal{N}(\xi | \bar{\mu}, \bar{\Sigma}) d\xi - \sum_{l=1}^{M} \sigma(a_l) \min_{\substack{\mu \in [\mu_R^L, \mu_R^U] \\ \Sigma \in [\Sigma_R^L, \Sigma_R^U]}} \int_{a_l}^{b_l} \mathcal{N}(\xi | \mu, \Sigma) d\xi \right|. \tag{36}$$

Let $\mu^{*,(l)}$ and $\Sigma^{*,(l)}$ be the solutions to the lth minimisation problems defined inside the summation of the equation above, then we have

$$\begin{aligned}
&\left| \int \sigma(\xi) \mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma}) d\xi - \sum_{l=1}^{M} \sigma(a_l) \int_{a_l}^{b_l} \mathcal{N}(\xi|\mu^{*,(l)}, \Sigma^{*,(l)}) d\xi \right| \\
=& \left| \sum_{l=1}^{M} \left(\int_{a_l}^{b_l} \sigma(\xi) \mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma}) d\xi - \sigma(a_l) \int_{a_l}^{b_l} \mathcal{N}(\xi|\mu^{*,(l)}, \Sigma^{*,(l)}) d\xi \right) \right| \\
\leq& \left| \sum_{l=1}^{M} \left(\left(\sigma(a_l) + \frac{1}{M} \right) \int_{a_l}^{b_l} \mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma}) d\xi - \sigma(a_l) \int_{a_l}^{b_l} \mathcal{N}(\xi|\mu^{*,(l)}, \Sigma^{*,(l)}) d\xi \right) \right| \\
\leq& \left| \frac{1}{M} \sum_{l=1}^{M} \int_{a_l}^{b_l} \mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma}) d\xi \right| + \left| \sum_{l=1}^{M} \sigma(a_l) \int_{a_l}^{b_l} \left(\mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma}) - \mathcal{N}(\xi|\mu^{*,(l)}, \Sigma^{*,(l)}) \right) d\xi \right| \\
\leq& \frac{1}{M} \left| \int_{\mathbb{R}} \mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma}) d\xi \right| + \sum_{l=1}^{M} \sigma(a_l) \left| \int_{a_l}^{b_l} \left(\mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma}) - \mathcal{N}(\xi|\mu^{*,(l)}, \Sigma^{*,(l)}) \right) d\xi \right| \\
\leq& \frac{1}{M} + \sum_{l=1}^{M} \left| \int_{a_l}^{b_l} \left(\mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma}) - \mathcal{N}(\xi|\mu^{*,(l)}, \Sigma^{*,(l)}) \right) d\xi \right|. \qquad (37)
\end{aligned}$$

Now, thanks to the conditions in Equation (34), we have that as $r \to 0$ both mean and variance converge to the actual maximum and minimum values in R. By further observing that $\bar{\mu}$ and $\bar{\Sigma}$ are by construction always inside the (vanishing) interval $[\mu_R^L, \mu_R^U] \times [\Sigma_R^L, \Sigma_R^U]$, then for continuity of the Gaussian pdf we have that for each $l = 1, \ldots, M$:

$$\lim_{r \to 0} \left| \int_{a_l}^{b_l} \left(\mathcal{N}(\xi|\bar{\mu}, \bar{\Sigma}) - \mathcal{N}(\xi|\mu^{*,(l)}, \Sigma^{*,(l)}) \right) d\xi \right| = 0$$

which means that the second term in Equation (37) can be made vanishingly small, in particular less than $\frac{\epsilon}{2}$. By selecting $M = \lceil \frac{2}{\epsilon} \rceil$ the theorem statement holds. ■

We have proved in Propositions 4 and 7 that the bounds for the mean and variance of Section 4 guarantee that the condition in Equation (34) holds. For multi-class classification (case $m > 2$), Theorem 14 can be generalised by further noticing that the error introduced by Proposition 11 also vanishes. For any $\epsilon > 0$, to ensure that convergence holds for the multi-class problem one has to select a number of discretisation boxes of the order of $\frac{1}{\epsilon^m}$.

6.1 Time Complexity

The method we have developed for the computation of adversarial robustness properties of GPs relies on the bounding of the posterior GP statistics, integrated within a branch-and-bound scheme for the iterative refinement of the bound.

Cost of Bounding Consider a kernel Σ with bounded kernel decomposition (φ, ψ, U), and let $\mathcal{K}$ denote the time complexity for the evaluation of the bounding function U. This is dependent on the particular function chosen, and in Appendix B we discuss its value for each kernel that we analyse. The time complexity for the computation of the mean bound is $\mathcal{O}(m\mathcal{K})$, where m is the output dimension of the GP. The computation of an upper bound on the posterior variance requires solving a convex quadratic problem, whose computational complexity is cubic in the number of input variables (Nesterov and Nemirovskii, 1994), i.e. $\mathcal{O}((d + 2N + Nd)^3)$, where d is the input dimensionality of the GP and N is the number of training points. Concerning the computation of the lower bound on the variance, we have to solve $2N + 1$ linear programming problems, where N is the size of the training set. This again depends on the number of optimisation variables and can be done in $\mathcal{O}((d + 2N + Nd)^{2.5} \log(d + 2N + Nd))$ (Cohen et al., 2021). We emphasise that, while computing the mean is straightforward, bound computations for the variance are more involved. As a result, adversarial robustness for multi-output regression can be obtained much faster in practice than for multi-class classification. To demonstrate this, in Section 7 we investigate a multi-output regression problem with 14 output dimensions.

Cost of Refinement Once the bounds on the mean and variance have been computed, refining them through branch-and-bound up to a desired threshold $\epsilon > 0$ has a worst-case cost exponential in the number of dimensions of T. Furthermore, for multi-class classification, to guarantee convergence we have to discretise the region into a grid of size $\frac{1}{\epsilon^m}$, where $m > 2$ is the number of classes. This adds to the overall time complexity, which in the multi-class case is exponential also with respect to the number of classes.

7. Experimental Results

We employ our methods to experimentally analyse the robustness of GP models in adversarial settings. We give results for four classification data sets: (i) Synthetic2D, generated by shifting a two-dimensional standard-normal either along the first (class 1) or second dimension (class 2); (ii) the SPAM data set (Dua and Graff, 2017), a binary data set with the split between the negative and positive classes respectively of 41% and 59%; (iii) a two-class subset of the MNIST data set (LeCun, 1998) with classes 3 and 8 (i.e., MNIST38) and a three-class subset with classes 3, 5 and 8 (i.e., MNIST358); (iv) a two-class subset of FashionMNIST (F-MNIST) (Xiao et al., 2017) with classes "t-shirt/top" and "shirt" (which we refer to as F-MNIST-TS) and a three-class subset with classes "t-shirt/top", "shirt" and "pullover" (F-MNIST-TSP). Furthermore, we analyse the robustness of the Water Quality data set (Džeroski et al., 2000) for multi-output regression.

Training We learn the GP models using a squared-exponential kernel and zero mean prior and select the hyper-parameters by means of MLE (Rasmussen and Williams, 2006). For the Synthetic2D data set we learn the GP over 1000 training samples and test it over 200 test samples, obtaining an accuracy of $\approx 98\%$. For the SPAM data set we first standardise the data to zero mean and unit variance. Then, we perform feature-reduction by iteratively training an ℓ_1-penalised logistic regression classifier and discarding the least relevant features, up until test set accuracy starts to diminish. This procedure leaves us with 11 features out of the initial 57. We then train a two-class classification GP over the resulting reduced feature vector. The GP thus computed achieves a test set accuracy of around 93%.

For MNIST and F-MNIST we first sub-sample the images to 14 × 14 pixels,[7] and use similar learning settings as for the SPAM data set, with 1000 training samples randomly picked from the two data sets. We achieve a test set accuracy of around 98% for MNIST38 and 90% for F-MNIST-TS. For the two multi-class problems we use the softmax likelihood function and training setting similar to the two-class classification problems, obtaining a test set accuracy of around 93% for MNIST358 and 85% for F-MNIST-TSP. Finally, we standardise the Water Quality data set and use the full set of 16 input features to predict the 14 regression outputs. To do so, we learn a multi-output regression GP over a 50%/50% train/validation split of the full data set (1060 data points), obtaining a mean absolute error of around 0.15.

We rely on the GPML Matlab toolbox for the training of two-class GPs (Rasmussen and Nickisch, 2010) and on the GPStuff toolbox for the training of multi-class GPs, sparse GP models and the multi-output regression model (Vanhatalo et al., 2013).

Parameter Selection for the Analysis We compute adversarial robustness in neighbourhoods of the form $T = [x^* - \gamma, x^* + \gamma]$ around a given point x^* and for a range of $\gamma > 0$. Unless otherwise stated, we run the branch-and-bound algorithm until convergence up to an error threshold $\epsilon = 0.01$. For MNIST38 and F-MNIST-TS we perform feature-level analysis for scalability reasons, similarly to Ruan et al. (2018). Namely, we restrict our methods to salient patches of each image only, as detected by SIFT (Lowe, 2004). We note that any other image feature extraction method can be used instead.

In the remainder of this section, we discuss results concerning three types of analyses. First (Section 7.1), on four samples selected from the classification data sets, we provide empirical evidence illustrating the advantages of computing guarantees (as those provided by our branch-and-bound method) versus evaluating model robustness using gradient-based attacks for classification models. Next we consider the robustness of GPs learned by using a selection of latent-variable methods (Section 7.2) and sparse approximation techniques (Section 7.3), discussing the adversarial robustness properties of these state-of-the-art approximate inference methods. Finally, we show how the techniques developed here for adversarial robustness can be applied to perform interpretability analysis of classification GP models predictions (Section 7.4).

7.1 Local Adversarial Safety

We study local adversarial safety for four points selected from the classification data sets, i.e. Synthethic2D, SPAM, MNIST38 and F-MNIST-TS data sets and summarise the results in Figure 2. 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 lines for class 2, green solid and dashed lines for class 1) with GPFGS, a gradient-based method for attacking GP mean prediction by Grosse et al. (2017) (pink curve in the plot), and Carlini & Wagner (CW) attack (Carlini

[7] This reduces the number of hyper-parameters that need to be estimated by MLE and increases the numerical stability of the GP, while achieving comparable accuracy.

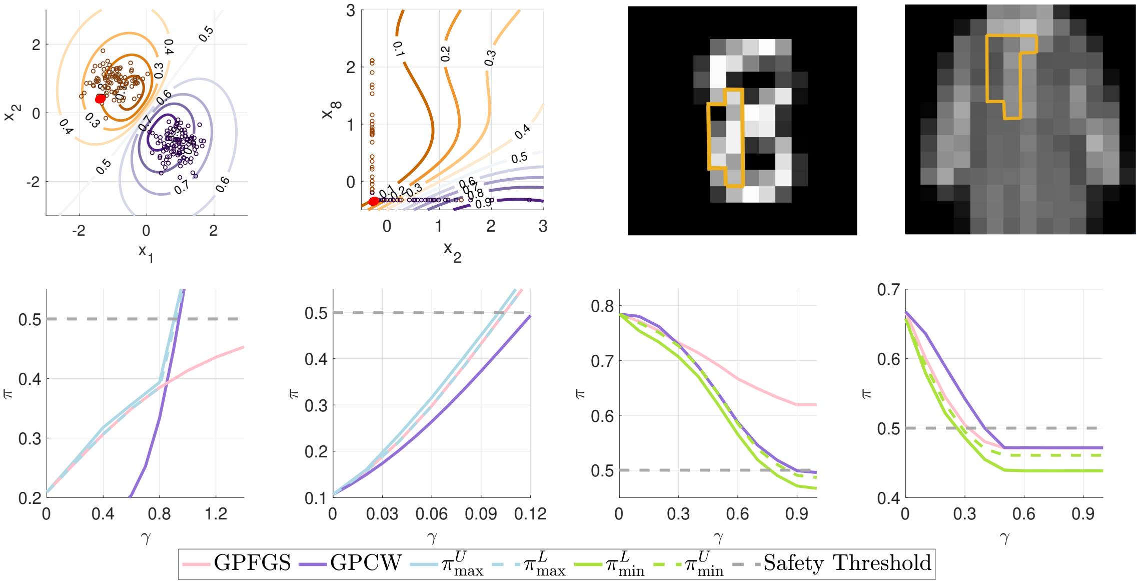

Figure 2: **First row**: Contour plot and test points for Synthetic2D; projected contour plot and test points for 2 dimensions of SPAM (dimensions 2 and 8 as selected by ℓ_1-penalised logistic regression); sample of 8 from MNIST38 along with 10 pixels selected by SIFT; sample of shirt from F-MNIST-TS along with the 10 pixels selected by SIFT. **Second row**: Safety analysis for the four selected test points. Shown are the upper and lower bounds on $max(T)$ (solid and dashed blue curves), $min(T)$ (solid and dashed green curves), the GPFGS adversarial attack (pink curve), and the GPCW attack (violet curve).

and Wagner, 2017) for the ℓ_∞ norm (GPCW) (violet curve in the plot). Naturally, as γ increases, the neighbourhood 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), both GPFGS and GPCW often underestimate the effect of the worst-case perturbations, such as in Figure 2 (bottom left) for GPFGS and in Figure 2 (bottom, second figure from left) for GPCW. In particular, GPFGS, because of its local nature, tends to underestimate the true robustness values for large values of γ. On the other hand, GPCW, while more accurate for large values of γ, in some cases generates less accurate attacks than GPCW for small values of γ, such as Figure 2 (bottom left). We stress that our method provides converging bounds of the true robustness, and as a consequence GPCW and GPFSM attacks are always contained within the bounds given by our method.

7.2 Local Adversarial Robustness

We now evaluate the empirical distribution of the adversarial robustness of the trained GP models. To this end, we introduce a quantitative measure of robustness analogous to that used by Ruan et al. (2018). More specifically, we consider the difference between the max-

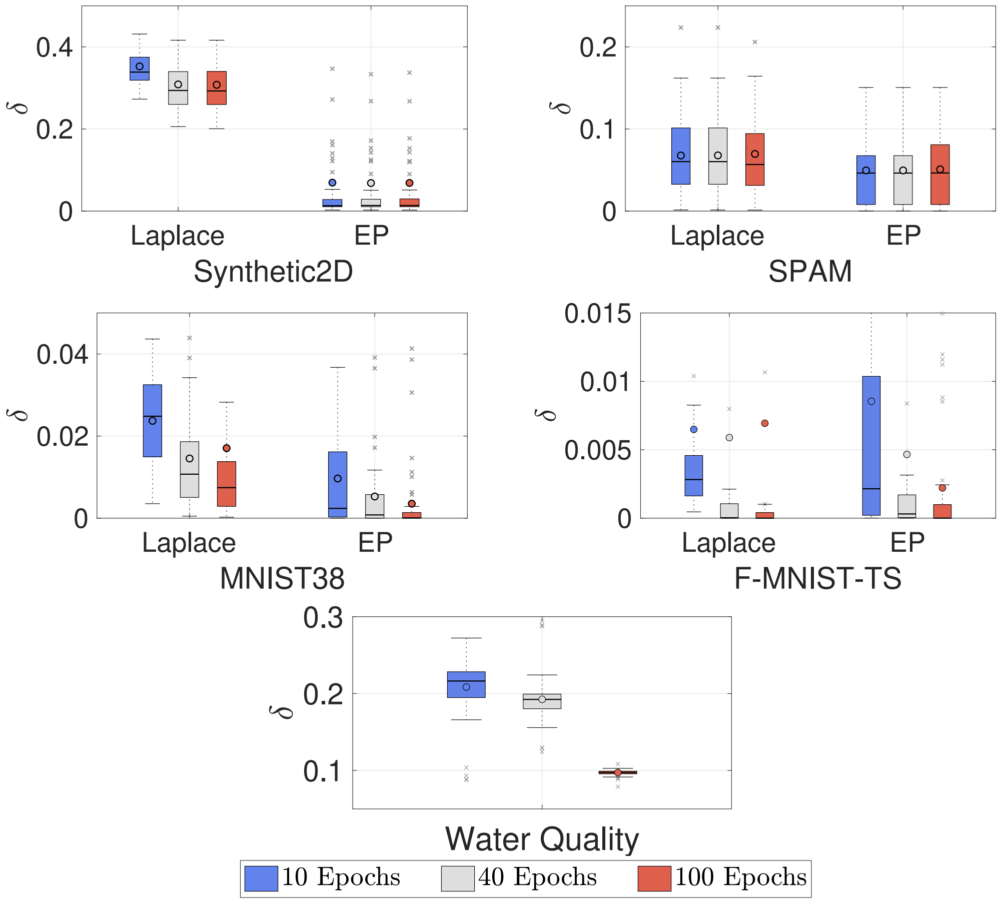

Figure 3: Boxplots for the distribution of robustness on the five data sets studied, comparing Laplace and EP approximation for the classification models, for $\gamma = 0.1$.

imum and minimum prediction probability in the region T, $\delta = \pi^U_{\max}(T) - \pi^L_{\min}(T)$, which utilises the computed quantities. We evaluate the moments of the empirical distribution of values of δ on 50 randomly selected test points for each of the four data sets considered. Note that a smaller value of δ implies a more robust model. Furthermore, in the classification cases, we analyse how the GP model robustness is affected by the training procedure used. To achieve this, we compare the robustness obtained when using either the Laplace or the Expectation Propagation (EP) (Rasmussen and Williams, 2006) posterior approximations technique, and investigate the influence of the number of marginal likelihood evaluations (epochs) performed during MLE hyper-parameter optimisation on robustness.

Results for this analysis are depicted in Figure 3, for 10, 40 and 100 hyper-parameter optimisation epochs. As explained above, the analyses for the MNIST38 and F-MNIST-TS samples are restricted to the most influential SIFT features only, and thus δ values for them are smaller in magnitude than for the other two data sets (for which all the input variables are simultaneously changed). Interestingly, this empirical analysis demonstrates that GPs 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 and Williams, 2006). These results quantify and confirm for GPs that the posterior distribution is robust to adversarial attacks in the limit, as theorethically analysed by Carbone et al. (2020) in the case of over-parameterised Bayesian neural networks, of which GPs are a particular case (De Matthews et al., 2018). We observe that the values of δ decrease as the number of training epochs increases, and thus robustness improves with the increase in the number of training epochs. More training in Bayesian settings may imply better calibration of the latent mean and variance function to the observed data. Interestingly, we note that, also in the regression case, we observe the same trend as in the classification experiments, with robustness of the GP increasing with the number of hyper-parameter training epochs.

7.3 Robustness of Sparse Approximations

In Section 7.2 we have empirically observed that a more refined training procedure may lead to more robust GP models. In standard GP settings it is infeasible to work with large-scale data sets that approximate the exact data manifold, as inference scales with the cube of the number of data (and storage with its square) (Bauer et al., 2016). For large-scale data sets, sparse GPs (Bauer et al., 2016) are customarily used for approximating the GP posterior distribution. While sparse GP approximations are usually evaluated in terms of mean and uncertainty calibration, here we consider adversarial robustness of GP sparse approximation techniques.

As inference equations for sparse approximation can be generally cast in the form of Equations (1)–(2), our methodology can be applied directly, modulo the definition of the matrix S, vector $\mathbf{t}$ and the vector of inducing points $\mathbf{u}$ (that is, the set of eventually synthetic points on which training is performed). We rely on the EP latent method and compare the results for FIC, DTC, and VAR sparse approximation methods (Quiñonero-Candela and Rasmussen, 2005) on the MNIST38 and F-MNIST-TS data sets. We vary the number of training points from 250 to 7500 and the number of inducing points (selected at random from the training points) from 100 to 500. For each of the resulting GPs we analyse the empirical distribution of δ-robustness on 50 randomly sampled test points with respect to their most relevant features (as detected by SIFT) with $\gamma = 0.15$.

Results for this analysis are plotted in Figure 4, where boxplots are grouped according to the number of training points, with each boxplot in each group representing an increase of 100 inducing points (starting from 100). The test set accuracy of each GP, as estimated over 1000 test samples, is plotted in the same figure on a separate y-axis (red dots). In agreement with the literature on sparse GPs (Bauer et al., 2016), we observe that an increasing number of training and/or inducing points generally leads to more accurate models. Among the two data sets analysed here, this aspect is more pronounced on F-MNIST ($\approx 6\%$ increase), which poses a more complex classification task than MNIST ($\approx 2\%$ increase), so that the GP further benefits from more information from data.

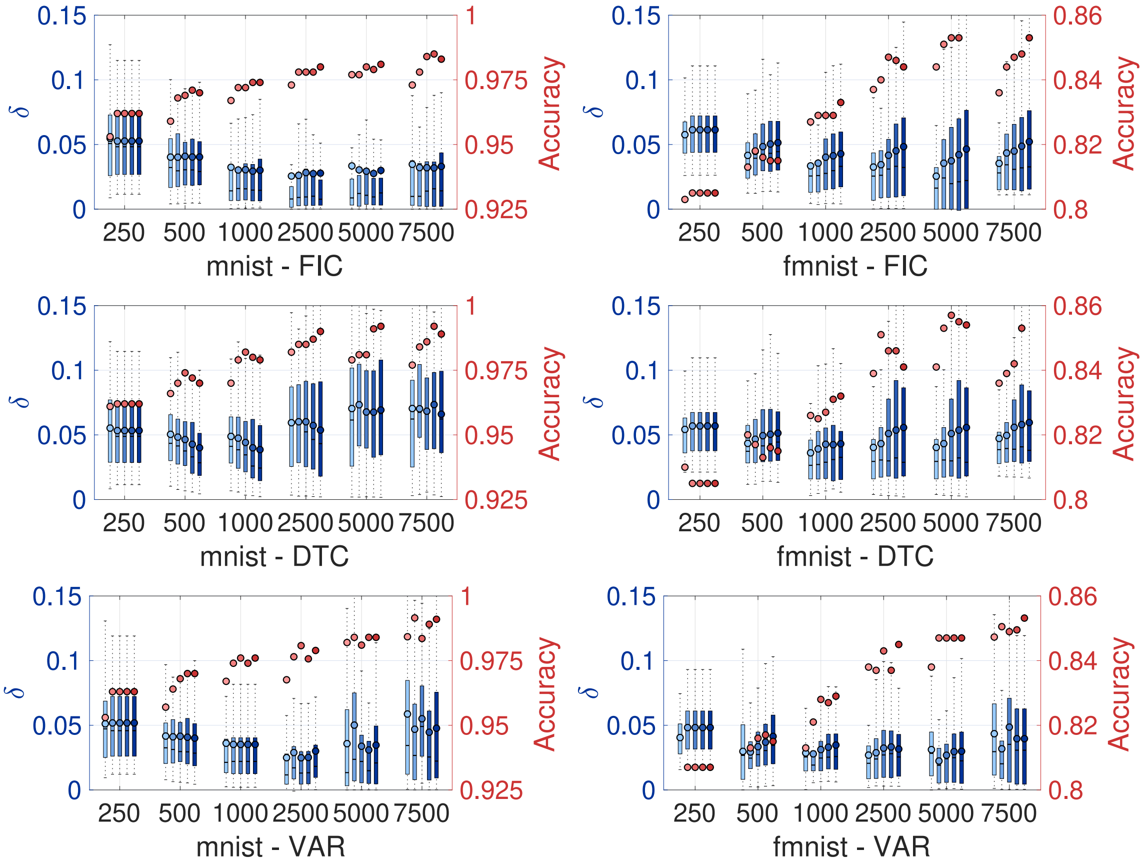

Figure 4: Empirical distribution of δ-robustness for $\gamma = 0.15$. **First Row:** FIC sparse approximation. **Second Row:** DTC sparse approximation. **Third Row:** VAR sparse approximation.

The robustness trends instead depend on the approximation techniques used. For FIC and VAR, we generally obtain that more training input points corresponds to an increase in the model robustness (i.e., lower value of δ). More specifically, sparse GPs successfully take into account the information from a larger pool of training samples in refining its posterior estimation. Unfortunately, for the VAR models the EP computations become numerically unstable after 2500 training samples and we have to increase the data jitter (which results in a widening of the boxplot and reduced robustness). For DTC, instead, we observe that the robustness slightly worsens in the case of MNIST and remains stable for F-MNIST. Finally, we remark that the number of inducing points has little effect on the overall robustness when compared to the number of training points used.

7.4 Interpretability Analysis

Adversarial robustness and model prediction interpretability are closely linked together (Darwiche, 2020). To demonstrate this, we can utilise the bounds we compute on $\pi_{\min}(T)$ and $\pi_{\max}(T)$ to formulate an interpretability metric similar to that defined for linear classifiers in (Ribeiro et al., 2016) and implemented in a black-box tool called LIME. In particular, we consider a test point x^* and the one-sided input box $T^i_\gamma(x^*) = [x^*, x^* + \gamma e_i]$ (where e_i denotes 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:

$$\mathbf{\Delta}^i_\gamma(x^*) = \left(\pi_{\max}(T^i_\gamma(x^*)) - \pi_{\max}(T^i_{-\gamma}(x^*))\right) + \left(\pi_{\min}(T^i_\gamma(x^*)) - \pi_{\min}(T^i_{-\gamma}(x^*))\right).$$

If increasing the value of dimension i makes the model favour assigning lower class probabilities, we would expect this value to be negative and vice versa. Intuitively, this provides a non-linear generalisation of numerical gradient estimation, which resembles exactly the metric used by Ribeiro et al. (2016) as γ tends to 0 or if the model considered is linear. Global estimation measures can be computed by estimating the expected value of $\mathbf{\Delta}^i_\gamma(x^*)$ with x^* sampled from a test set. However, since our method relies on the analytic form of the inference equations of GPs (rather than being model-agnostic, which LIME is), we are able to formally bound these quantities, which allows as to provide guarantees over interpretability results. Next, we first graphically demonstrate why linear approximation can be misleading for global interpretability analysis for the 2D-synthetic and SPAM data sets, and then show how we can rely on formal quantification of interpretability to investigate the adversarial vulnerability of a GP model around specific test points.

Global Interpretability Analysis for Synthethic2D and SPAM We perform global interpretability analysis on GP models trained on the Synthetic2D and SPAM data sets, estimating the expected value of $\mathbf{\Delta}^i_\gamma$ with 50 random test points. The results are shown in Figure 5. 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, as it is built on linearity approximations, fails to take into account the global behaviour of the GP. 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

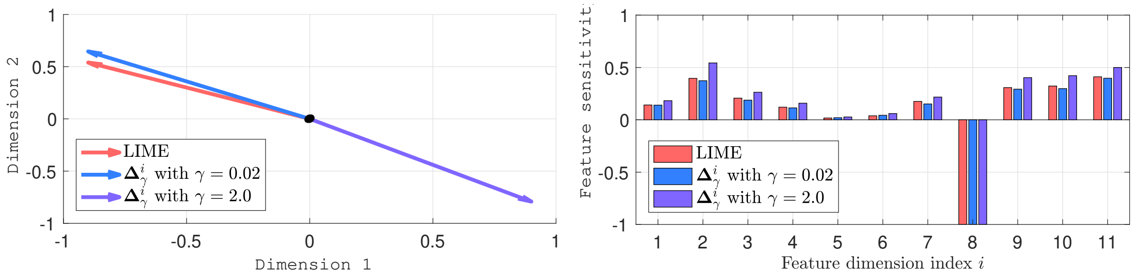

Figure 5: Global interpretability, $\boldsymbol{\Delta}^i_\gamma$, as analysed by LIME and our method. **Left:** Results for the Synthetic2D data set. **Right:** Results for the SPAM data set.

SPAM, on the other hand (Figure 5, bottom), due to the linearity of the data set and the GP, a local analysis correctly reflects the global picture.

Interpretability for MNIST358 and F-MNIST-TSP Predictions As shown in (Darwiche, 2020), interpretability metrics can be used to synthesise adversarial examples, because pixels and features that are deemed important for the prediction are also likely to be highly vulnerable to adversarial perturbations. These results can be used to glimpse further information about interpretability of the predictions by qualitatively examining the obtained adversarial examples. To this end, given a test point x^* and a point x taken from a small neighbourhood around x^*, we define the adversarial gap, $\pi_{\text{gap}}(x)$, as the minimum difference between the confidence in the true class and those of the remaining classes on x, so that $\pi_{\text{gap}}(x) < 0$ implies that x is an adversarial example for x^*. We analyse how π_{gap} changes as we change an increasing number of pixels, β, and compare the results obtained with our method to those of LIME.

We plot the results on six images randomly selected from the MNIST358 and F-MNIST-TSP data sets in Figure 6. The selected clean test images are reported in the first row of the figure, and the interpretability values are reported as a heatmap in the second row directly below the corresponding images. The colour gradient varies from red (positive impact, pixel value increase resulting in increased class probability of shown digit) to blue (negative impact, pixel value increase decreasing the class probability). The values of π_{gap} obtained with our method (blue line) are compared with those from LIME (red line) in the third row, and in the fourth row we plot the minimal adversarial examples found with our method.

We observe that, for each image, and for each value of β, relying on the values estimated by LIME leads to an over-estimation of model robustness, and in some cases (e.g., third and fifth column) more than triple the number of pixel modifications is required to find an adversarial example. We note that the adversarial examples that we obtained for MNIST and F-MNIST are qualitatively different. For the MNIST image, our method modified salient bits of the image. For digit 3, for example, the interpretability analysis retrieves a contiguous blue patch on the left, which is deemed to be the most important part for the prediction. When this is modified in adversarial fashion, the image obtained resembles an 8 in the upper part, and a 3 in the lower part, and is (understandably) classified as an 8 by the GP. Similarly, digit 5 is modified so that the lower part resembles an 8, whereas in the image of the 8 a 3 shaped contour is highlighted in the adversarial example. For

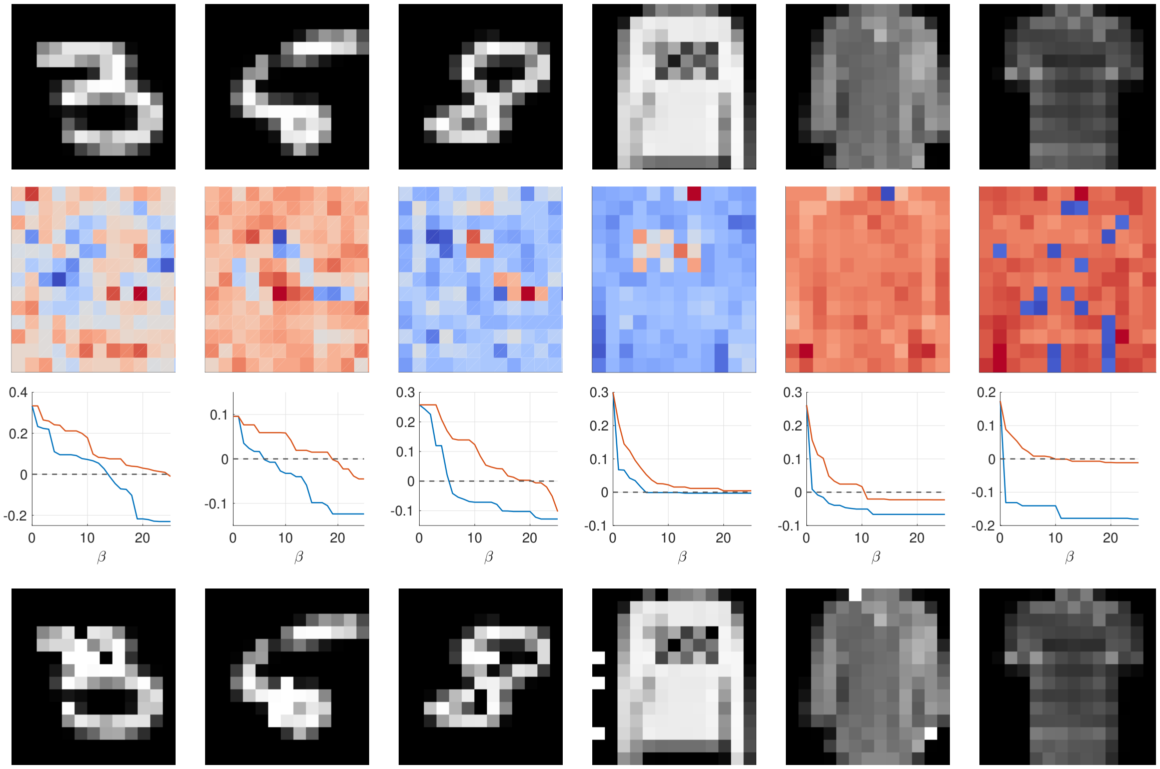

Figure 6: **First row**: 6 test images randomly selected from MNIST358 and F-MNIST-TSP. **Second row**: Interpretability metric estimation using our method. **Third row**: Comparison of adversarial gaps (y-axis) obtained for a given budget β (x-axis) when using our method for interpretability estimation (blue line) and when using LIME (red line). The dashed grey line represents the threshold below which an adversarial is found. **Fourth row**: Minimal adversarial examples found by utilising our interpretability metric.

the F-MNIST image, instead, our method detects pixels that are important for the GP prediction but have little semantic meaning for a human, that is, where modifying pixels in the border of the image suffices to find adversarial examples.

8. Conclusion

We presented a method for computing, for any compact region of the input space surrounding a test point, provable guarantees on the adversarial robustness of a GP model for all points in that region, up to any desired precision $\epsilon > 0$. To achieve this, we have developed a branch-and-bound optimisation scheme that computes upper and lower bounds on the minimum and maximum of the model prediction ranges, and proved that it converges in finitely many steps up to an error tolerance $\epsilon > 0$ selected a-priori.

We have experimentally evaluated our method on four classification data sets and a regression one, providing results for adversarial robustness, bounds over the predictive posterior distribution and local/global interpretability analysis. Empirically, we have observed that, in Bayesian prediction settings and with GPs, the adversarial robustness of the model increases with the accuracy of the posterior distribution approximation, and with better hyper-parameter calibration. This differs from what is generally observed in frequentist approaches to learning, for example, in deep neural networks, where better accuracy was empirically observed to imply worse adversarial robustness (Zhang et al., 2019; Su et al., 2018). We have also observed that increasing the number of training samples might still be beneficial for adversarial robustness even when using sparse approximations for GP training.

One limitation of the approach presented in this paper is its high, exponential time, computational complexity. This unsurprising since the problem we are solving is non-linear optimisation. To reduce the computational time requirement, we have formulated analytical solutions for the main types of kernels used in practical applications. We have also observed that sparse GPs, as well as improving training time, can significantly reduce the time requirement of our methods, as bounding functions need to be computed only with respect to the inducing points. We believe that the methods proposed in this paper are therefore widely applicable in practice.

Acknowledgments

SR is grateful for support from the UK Royal Academy of Engineering and the Oxford-Man Institute. AB thanks the Konrad-Adenauer-Stiftung and the Oxford-Man Institute for their support. MK, LL and AP received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (FUN2MODEL, grant agreement No. 834115). AP and MK acknowledge partial funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 722022 "AffecTech". LC is supported by a Royal Society Research Professorship.

Appendix A. Additional Lemmas and Proofs

In this section we provide statements of additional lemmas referred to in the main text of the paper, as well as proofs of that were omitted for space reasons.

Lemma 15 *Let $g_L(t) = a_L + b_L t$ and $g^U(t) = g_U(t) = a_U + b_U t$ be an LBF and UBF to a function $g(t)$ $\forall t \in \mathcal{T}$, i.e. $g_L(t) \leq g(t) \leq g_U(t)$ $\forall t \in \mathcal{T} \subseteq \mathbb{R}$. Consider two real coefficients $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$. Define*

$$\bar{b}_L = \begin{cases} \alpha b_L \text{ } if \alpha \geq 0 \\ \alpha b_U \text{ } if \alpha < 0 \end{cases} \quad \bar{a}_L = \begin{cases} \alpha a_L + \beta \text{ } if \alpha \geq 0 \\ \alpha a^U + \beta \text{ } if \alpha < 0 \end{cases} \tag{38}$$

$$\bar{b}_U = \begin{cases} \alpha b_U \text{ } if \alpha \geq 0 \\ \alpha b_L \text{ } if \alpha < 0 \end{cases} \quad \bar{a}_U = \begin{cases} \alpha a_U + \beta \text{ } if \alpha \geq 0 \\ \alpha a_L + \beta \text{ } if \alpha < 0. \end{cases} \tag{39}$$

Then:

$$\bar{g}_L(t) := \bar{a}_L + \bar{b}_L t \leq \alpha g(t) + \beta \leq \bar{a}_U + \bar{b}_U t =: \bar{g}_U(t).$$

That is, LBFs can be propagated through linear transformation by redefining the coefficients through Equations (38)–(39).

Proof The proof is an immediate consequence of multiplying the inequalities $g_L(t) \leq g(t) \leq g_U(t)$ with the coefficients α and β and re-writing the new inequality using the constants defined in Equations (38)–(39). ■

Lemma 16 *Consider the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$.Let $z > 0$, then we have:*

$$\sigma'(\mu - z) \begin{cases} > \sigma'(\mu + z) & if \quad \mu > 0 \\ < \sigma'(\mu + z) & if \quad \mu < 0. \end{cases}$$

Proof Let $\mu > 0$; the proof when $\mu < 0$ is similar, because σ' is an even function.

Case 1: $\mu - z \geq 0$. Since σ is strictly concave in $[0, +\infty)$, the derivative is a monotonic crescent in the relevant region. Thus, $\sigma'(\mu - z) > \sigma'(\mu + z)$.

Case 2: $\mu - z < 0$. Since σ' is even we have $\sigma'(\mu - z) = \sigma'(z - \mu)$. Now $z - \mu > 0$, similarly to Case 1 we have $\sigma'(z - \mu) < \sigma'(z + \mu)$, which proves the lemma. ■

Lemma 17 *Let X and Y be random variables with joint density function $f_{X,Y}$. Consider measurable sets A and B. Then, it holds that*

$$P(X \in A | Y \in B) \geq \inf_{y \in B} P(X \in A | Y = y).$$

Proof

$$
\begin{aligned}
P(X \in A|Y \in B) &= \frac{P(X \in A, Y \in B)}{P(Y \in B)} = \frac{\int_B \int_A f_{X,Y}(x,y)dxdy}{\int_B f_Y(y)dy} \\
&= \frac{\int_B \int_A f_{X|Y}(x|y) f_Y(y) dxdy}{\int_B f_Y(y)dy} \geq \frac{\int_B f_Y(y)dy \inf_{y \in B} \int_A f_{X|Y}(x|y)dx}{\int_B f_Y(y)dy} \\
&= \inf_{y \in B} P(X \in A|Y = y).
\end{aligned}
$$

■

A.1 Proof of Lemma 1

Proof We show how to compute $\bar{a}_L$ and $\bar{b}_L$; the same arguments also apply to the computation of $\bar{a}_U$ and $\bar{b}_U$ by simply considering $-\Sigma_{x,\bar{x}}$.

Consider $c_L = -1$ and $c_U = 1$ coefficients associated to the input point $\bar{x}$. Let $\varphi_L = U(c_L)$ and $\varphi_U = U(c_U)$, then by Assumption 3 of bounded kernel decomposition we have that $\varphi(x, \bar{x}) \in [\varphi_L, \varphi_U]$ for all $x \in T$. Consider now the function ψ restricted to the interval $[\varphi_L, \varphi_U]$. Then there are four cases to consider for ψ.

Case 1 If ψ happens to be concave in $[\varphi_L, \varphi_U]$, then, by definition of concave function, a lower bound is given by the line that links the points $(\varphi^L, \psi(\varphi^L))$ and $(\varphi^U, \psi(\varphi^U))$, that is, g_L is simply the LBF with coefficients:

$$
\begin{aligned}
\bar{b}_L &= \frac{\psi(\varphi^L) - \psi(\varphi^U)}{\varphi^L - \varphi^U} \\
\bar{a}_L &= \psi(\varphi^L) - \bar{b}_L \varphi^L.
\end{aligned}
$$

Case 2 If ψ happens to be a convex function, then, by definition of convex function and by the differentiability of ψ, a valid lower bound is given by the tangent line in the middle point $\varphi^C = (\varphi^L + \varphi^U)/2$ of the interval, that is, g_L is the LBF with coefficients:

$$
\begin{aligned}
\bar{b}_L &= \frac{d\psi(\varphi^C)}{d\varphi} \\
\bar{a}_L &= \psi(\varphi^L) - \bar{b}_L \varphi^L.
\end{aligned}
$$

Case 3 Assume now that ψ is concave in $[\varphi^L, \varphi^F]$, and convex in $[\varphi^F, \varphi^U]$ (the arguments are very similar if we assume the first interval is that in which ψ is convex and the second concave). In other words, there is only one flex point $\varphi^F \in (\varphi^L, \varphi^U)$. Let $\bar{a}'_L$, $\bar{b}'_L$ be coefficients for linear lower approximation in $[\varphi^L, \varphi^F]$ and $\bar{a}''_L$, $\bar{b}''_L$ analogous coefficients in $[\varphi^F, \varphi^U]$ (respectively computed as for Case 1 and Case 2 above), and call g' and g'' the corresponding functions. Define g_L to be the LBF function of coefficients $\bar{a}_L$ and $\bar{b}_L$ that goes through the two points $(\varphi^L, \min(g'(\varphi^L), g''(\varphi^L)))$ and $(\varphi^U, g''(\varphi^U))$. We then have that g_L is a valid lower bound function for ψ in $[\varphi^L, \varphi^U]$. In order to prove this we distinguish between two cases:

1. If $\min(g'(\varphi^L), g''(\varphi^L)) = g'(\varphi^L)$, then we have that $g_L(\varphi^L) = g'(\varphi^L) \leq g''(\varphi^L)$, and $g_L(\varphi^U) = g''(\varphi^U)$. Hence, because of linearity, $g_L(\varphi) \leq g''(\varphi)$ in $[\varphi^L, \varphi^U]$, and in particular in $[\varphi^F, \varphi^U]$ as well. This also implies that $g_L(\varphi^F) \leq g''(\varphi^F) \leq g'(\varphi^F)$. On the other hand, $g_L(\varphi^L) = g'(\varphi^L)$, hence $g_L(\varphi) \leq g'(\varphi)$ in $[\varphi^L, \varphi^F]$. Combining these two results and by construction of g' and g'' we have that $g_L(\varphi) \leq \psi(\varphi)$ in $[\varphi^L, \varphi^U]$.

2. If $\min(g'(\varphi^L), g''(\varphi^L)) = g''(\varphi^L)$, then in this case we have $g_L = g''$, and just have to show that $g(\varphi) \leq g'(\varphi)$ in $[\varphi^L, \varphi^F]$. This immediately follows by noticing that $g''(\varphi^F) \leq g'(\varphi^F)$ and $g''(\varphi^L) \leq g'(\varphi^L)$.

Case 4 In the general case, as we have a finite number of flex points, we can divide $[\varphi^L, \varphi^U]$ in subintervals in which ψ is either convex or concave. We can then proceed iteratively from the two left-most intervals by repeatedly applying Case 3. ■

A.2 Proof of Lemma 3

Proof We prove the lemma for the LBF. An analogous argument can be made for the UBF.

Letting $\epsilon > 0$, we want to find an $\bar{r} > 0$ such that $\text{diam}(R) < \bar{r}$ implies $\max_{x \in R} |g_L^R(x) - \Sigma_{\bar{x},x}| < \epsilon$. Consider φ_L^R and φ_U^R, lower and upper bound values for φ in R. By taking $\bar{r}$ small enough we can assume without loss of generality that $\psi(\varphi)$ has at most one flex point in $[\varphi_L^R, \varphi_U^R]$. We then have the following three cases.

CASE 1: if $\psi(\varphi)$ is concave then g_L^R is defined as the line connecting the two extreme points of the interval $[\varphi_L^R, \varphi_U^R]$. Since $\psi(\varphi)$ is concave, we have that it obtains its minimum in one of these two extrema, so that we have

$$\min_{x \in R} g_L^R(x) = \min_{x \in R} \Sigma_{\bar{x},x}.$$

By Assumption 2 of kernel decompositions (see Definition 4), it follows that ψ is Lipschitz continuous on any compact interval, so that we have that

$$\lim_{r \to 0} \left| \min_{x \in R} \Sigma_{\bar{x},x} - \max_{x \in R} \Sigma_{\bar{x},x} \right| = 0,$$

where $r = \text{diam}(R)$. Putting the two results together we have that the difference between $\min_{x \in R} g_L^R(x)$ and $\max_{x \in R} \Sigma_{\bar{x},x}$ vanishes whenever that r tends to zero, which proves the statement.

CASE 2: if $\psi(\varphi)$ is convex then g_L^R is the Taylor expansion of $\psi(\varphi)$ around the mid-point of the interval, truncated at the first-order. By continuity of φ we then obtain that shrinking r shrinks also the width of the interval $[\varphi_L^R, \varphi_U^R]$, which then, relying on the properties of truncation error of Taylor expansions, proves the lemma statement.

CASE 3: in the case in which a flex point exists, g_L^R is defined to be the maximum line that is below the two LBFs respectively defined over the convex and the concave part of the interval. Since by Case 1 and Case 2 these converge, we also have that g_L^R converges. ■

A.3 Proof of Lemma 12

Proof We provide the proof for the minimum; similar arguments also hold for the maximum.

By definition of μ_T^L, μ_T^U, Σ_T^L, Σ_T^U, we have that for every $x \in T$, $\bar{\mu}(x) \in [\mu_T^L, \mu_T^U]$ and $\bar{\Sigma}(x) \in [\Sigma_T^L, \Sigma_T^U]$. Thus:

$$\min_{x \in T} \int_a^b \mathcal{N}(\xi | \bar{\mu}(x), \bar{\Sigma}(x)) d\xi \geq \min_{\substack{\mu \in [\mu_T^L, \mu_T^U] \\ \Sigma \in [\Sigma_T^L, \Sigma_T^U]}} \int_a^b \mathcal{N}(\xi | \mu, \Sigma) d\xi =$$

$$\frac{1}{2} \min_{\substack{\mu \in [\mu_T^L, \mu_T^U] \\ \Sigma \in [\Sigma_T^L, \Sigma_T^U]}} \left(\mathrm{erf}\left(\frac{\mu - a}{\sqrt{2\Sigma}} \right) - \mathrm{erf}\left(\frac{\mu - b}{\sqrt{2\Sigma}} \right) \right) = \frac{1}{2} \min_{\substack{\mu \in [\mu_T^L, \mu_T^U] \\ \Sigma \in [\Sigma_T^L, \Sigma_T^U]}} \Phi(\mu, \Sigma),$$

where we have set $\Phi(\mu, \Sigma) := \mathrm{erf}\left(\frac{\mu - a}{\sqrt{2\Sigma}} \right) - \mathrm{erf}\left(\frac{\mu - b}{\sqrt{2\Sigma}} \right)$. By looking at the partial derivatives we have that:

$$\frac{\partial \Phi(\mu, \Sigma)}{\partial \mu} = \frac{\sqrt{2}}{\sqrt{\pi \Sigma}} \left(e^{-\frac{(\mu - b)^2}{2\Sigma}} - e^{-\frac{(\mu - a)^2}{2\Sigma}} \right) \geq 0 \Leftrightarrow \mu \leq \frac{a + b}{2} = \mu^c$$

and that if $\mu \notin [a, b]$:

$$\frac{\partial \Phi(\mu, \Sigma)}{\partial \Sigma} = \frac{1}{\sqrt{2\pi\Sigma^3}} \left((\mu - b_i) e^{-\frac{(\mu - b_i)^2}{2\Sigma^2}} - (\mu - a_i) e^{-\frac{(\mu - a_i)^2}{2\Sigma^2}} \right) \geq 0$$

$$\Leftrightarrow \Sigma \leq \frac{(\mu - a)^2 - (\mu - b)^2}{2 \log \frac{\mu - a}{\mu - b}} = \Sigma^c(\mu)$$

as otherwise the last inequality has no solutions. As such, μ^c and Σ^c will correspond to global maximum with respect to μ and Σ, respectively. As Φ is symmetric w.r.t. μ^c we have that the minimum value w.r.t. to μ is always obtained for the point furthest away from μ^c, that is, at $\underline{\mu}^* = \arg\max_{\mu \in [\mu_T^L, \mu_T^U]} |\mu^c - \mu|$. The minimum value w.r.t. to Σ will hence be either for Σ_T^L or Σ_T^U, that is $\underline{\Sigma}^* = \arg\min_{\Sigma \in \{\Sigma_T^L, \Sigma_T^U\}} \Phi(\underline{\mu}^*, \Sigma)$. ■

Appendix B. Kernel Function Decomposition

In this section, we compute explicit kernel decompositions (φ, ψ, U) for several kernels of practical relevance in applications. In particular, we give explicit formulas for the squared-exponential, the rational quadratic, the Matérn (for half-integer values) and the periodic kernels, along with how kernel decomposition can be propagated through addition and multiplication with kernels. We remark that the formula for addition and multiplication can be used recursively so to obtaine bounded decomposition ofr variously composed kernels. Furthermore, we show how to compute kernel decompositions for generalised spectral kernels, both in the stationary and non-stationary case.

Throughout this section, we assume $T = [x^L, x^U]$, for some $x^L, x^U \in \mathbb{R}^d$. For building the bounding function U we use the notation $x^{(i)}$, $i = 1, \ldots, N$ for the set of input points, and c_i, $i = 1, \ldots, N$, for their associated multiplying coefficients.

B.1 Squared-Exponential Kernel

For the squared-exponential kernel, we build a bounded kernel decomposition by setting:

$$\begin{aligned} \psi(\varphi) &= \sigma^2 \exp\left(-\varphi\right) \\ \varphi(x', x'') &= \sum_{j=1}^{d} \theta_j (x'_j - x''_j)^2. \end{aligned}$$

It is straightforward to notice that Assumptions 1 and 2 of Definition 4 are met by this decomposition. Concerning the definition of U, consider a set $x^{(1)}, \dots, x^{(N)}$ of N points in the input space and associated real coefficients $c_1, \dots, c_N$. For a hyper-rectangle $T = [x^L, x^U]$ we have that:

$$\begin{aligned} \sup_{x \in T} \sum_{i=1}^{N} c_i \varphi(x, x^{(i)}) &= \sup_{x \in T} \sum_{i=1}^{N} c_i \sum_{j=1}^{d} \theta_j (x_j - x_j^{(i)})^2 = \sup_{x \in T} \sum_{j=1}^{d} \theta_j \sum_{i=1}^{N} c_i (x_j - x_j^{(i)})^2 \\ &= \sup_{x \in T} \sum_{j=1}^{d} \left(\theta_j \sum_{i=1}^{N} c_i x_j^2 - 2\theta_j \sum_{i=1}^{N} c_i x^{(i)} x_j + \theta_j \sum_{i=1}^{N} c_i x^{(i)2} \right) \\ &= \sum_{j=1}^{d} \sup_{x_j \in [x_j^L, x_j^U]} \left(\theta_j \sum_{i=1}^{N} c_i x_j^2 - 2\theta_j \sum_{i=1}^{N} c_i x^{(i)} x_j + \theta_j \sum_{i=1}^{N} c_i x^{(i)2} \right). \end{aligned}$$

The right-hand-side of the last equation simply involves the computation of the maximum of a 1-d parabola over an interval of the real line, which can be done exactly and in constant time by simple inspection of the derivative function and by evaluating the function at the extrema of the interval. Call $\bar{x}_j$ the only critical point of the jth parabola, and denote with $h_j(x_j) = \alpha_j x_j^2 + \beta_j x_j + \gamma_j$ the parabola associated with the jth coordinate value, with $\alpha_j = \theta_j \sum_{i=1}^{N} c_i$, $\beta_j = -2\theta_j \sum_{i=1}^{N} c_i x^{(i)}$ and $\gamma_j = \theta_j \sum_{i=1}^{N} c_i x^{(i)2}$, then we set

$$U(\mathbf{c}) = \sum_{j=1}^{d} U_j(\mathbf{c}), \tag{40}$$

where:

$$U_j(\mathbf{c}) = \begin{cases} \max\left\{ h_j(x_j^L), h_j(x_j^U), h_j(\bar{x}_j) \right\} & \text{if} \quad \bar{x}_j \in [x_j^L, x_j^U] \\ \max\left\{ h_j(x_j^L), h_j(x_j^U) \right\} & \text{otherwise} \end{cases}.$$

Furthermore it follows that the time complexity for the computation of U in the squared-exponential case is $\mathcal{O}(N + d)$.

B.2 Rational Quadratic Kernel

An analogous argument to the above holds for the rational quadratic kernel, where we can set

$$\psi(\varphi) = \sigma^2 \left(1 + \frac{\varphi}{2}\right)^{-\alpha}$$
$$\varphi(x', x'') = \sum_{j=1}^{d} \theta_j (x'_j - x''_j)^2.$$

As the definition of φ is exactly the same as for the squared-exponential kernel, the bounding function U can be defined as in Equation (40).

B.3 Matérn Kernel

For half-integer values, the explicit form of the Matérn Kernel allows us to find an analogous kernel decomposition to the two discussed above:

$$\psi(\varphi) = \sigma^2 k_p \exp\left(-\sqrt{\hat{k}_p \varphi}\right) \sum_{l=0}^{p} k_{l,p} \sqrt[p-l]{\hat{k}_p \varphi}$$
$$\varphi(x', x'') = \sum_{j=1}^{d} \theta_j (x'_j - x''_j)^2.$$

B.4 Periodic Kernel

For the periodic kernel we define

$$\psi(\varphi) = \sigma^2 \exp(-0.5\varphi)$$
$$\varphi(x', x'') = \sum_{j=1}^{d} \theta_j \sin(p_j(x'_j - x''_j))^2.$$

Assumptions 1 and 2 are trivially satisfied because of the smoothness of ψ and φ. For the definition of the bounding function U we have that:

$$\begin{aligned} \sup_{x \in T} \sum_{i=1}^{N} c_i \varphi(x, x^{(i)}) &= \sup_{x \in T} \sum_{i=1}^{N} c_i \sum_{j=1}^{d} \theta_j \sin(p_j(x_j - x_j^{(i)}))^2 \qquad (41) \\ &\leq \sum_{i=1}^{N} \sum_{j=1}^{d} \sup_{x_j \in [x_j^L, x_j^U]} c_i \theta_j \sin\left(p_j(x_j - x_j^{(i)})\right)^2. \end{aligned}$$

The supremum in the final equation can be obtained by simply inspecting the derivative of $c_i \theta_j \sin\left(p_j(x_j - x_j^{(i)})\right)^2$ and its function value at the extrema of each interval $[x_j^L, x_j^U]$. Let $U_{ij}(c_i)$ be the value computed in such a way for each i and j, then we define:

$$U(\mathbf{c}) = \sum_{j=1}^{d} \sum_{i=1}^{N} U_{ij}(c_i). \qquad (42)$$

Furthermore it follows that the time complexity for the computation of U in the squared-exponential case is $\mathcal{O}(Nd)$.

B.5 Kernel Addition

Consider now the case in which the kernel function Σ is defined by linear composition of two kernels Σ' and Σ'' such as:

$$\Sigma_{x',x''} = k'\Sigma'_{x',x''} + k''\Sigma''_{x',x''} \qquad \forall x', x'' \in \mathbb{R}^d, \tag{43}$$

for some given k' and $k'' \geq 0$. Then, we have that kernel decomposition for Σ' and Σ'' can be simply propagated through the sum. To see that, let (φ', ψ', U') and (φ'', ψ'', U'') be the two kernel decomposition. Then, by simply summing up the LBFs and UBFs for Σ' and Σ'', Lemma 1 can be generalised to this case as follows.

Proposition 18 *Let g'_L, g'_U, g''_L and g''_U be lower and upper bounding function for $\Sigma'_{x,\bar{x}}$ and $\Sigma''_{x,\bar{x}}$, for all $x \in T$, as computed in Lemma 1. Then*

$$g_L(x) = k'g'_L(x) + k''g''_L(x)$$
$$g_U(x) = k'g'_U(x) + k''g''_U(x)$$

are respectively lower and upper bounding functions on $\Sigma_{x,\bar{x}}$.

As a consequence of the above proposition, it immediately follows that the infimum of the posterior mean function over the compact set T can be safely lower-bounded for the kernel Σ by setting:

$$\mu_T^L = k'\mu_T'^L + k''\mu_T''^L,$$

where $\mu_T'^L$ and $\mu_T''^L$ are computed by applying Proposition 2 to the kernels Σ' and Σ''. Similarly, Propositions 5 and 6 can be generalised by considering two sets of slack variables, one associated to φ' and one to φ'', and relying directly on the lower- and upper-bounding functions defined in Proposition 18.

B.6 Kernel Multiplication

When two kernels are combined through multiplication, we have that $\Sigma_{x',x''} = \Sigma'_{x',x''}\Sigma''_{x',x''}$. This case can be reduced to the addition by considering the following McCormick's inequalities (McCormick, 1976):

$$\Sigma_{x',x''} = \Sigma'_{x',x''}\Sigma''_{x',x''} \geq \Sigma'_L\Sigma''_{x',x''} + \Sigma'_{x',x''}\Sigma''_L - \Sigma'_L\Sigma''_L \tag{44}$$

$$\Sigma_{x',x''} = \Sigma'_{x',x''}\Sigma''_{x',x''} \leq \Sigma'_U\Sigma''_{x',x''} + \Sigma'_{x',x''}\Sigma''_L - \Sigma'_U\Sigma''_L, \tag{45}$$

where Σ'_L, Σ'_U, Σ''_L and Σ''_U are lower and upper bound values for Σ' and Σ'' in T, respectively. Then we can proceed by using the kernel summation of Equation (44) when computing lower bounding function on the kernel, and Equation (45) when computing the upper bounding function, and by using the techniques discussed in the section just above.

B.7 Generalised Spectral Kernel

We show how to find kernel decompositions compatible with our optimisation framework for generalised spectral kernels (Samo and Roberts, 2015). We note that these are dense in the space of kernel functions, so that they can be used to derive any kernel up to an arbitrary small error tolerance.

Stationary Kernel For the stationary case, we have:

$$\Sigma_{x',x''} = \sum_{k=1}^{K} \sigma^2 h((x'-x'') \odot \theta^{(k)}) \cos(w_k^T(x'-x'')), \tag{46}$$

where $\theta^{(k)} \geq 0$, and h is any given positive definite function; in particular, we choose $h((x'-x'') \odot \theta^{(k)}) = \exp\left(-\sum_{j=1}^{m} \theta_j^{(k)} (x'-x'')^2\right)$. We now show how a bounded kernel decomposition (φ, ψ, U) can be derived for this kernel.

We first observe that the kernel is obtained by summing over K different kernel components. According to the results for kernel addition described in Appendix B.5, it suffices to find a bounded kernel decomposition for each summand of Equation (46), i.e., for

$$k(x',x'') = \sigma^2 \exp\left(-\sum_{j=1}^{m} \theta_j (x'-x'')^2\right) \cos(w^T(x'-x'')).$$

In turn, by setting $k_1(x',x'') = \sigma^2 \exp\left(-\sum_{j=1}^{d} \theta_j (x'-x'')^2\right)$ and $k_2(x',x'') = \cos(w^T(x'-x''))$, we have that $k(x',x'') = k_1(x',x'')k_2(x',x'')$, and thus a kernel decomposition can be found by using the formulas for kernel multiplication derived in Appendix B.6 to $k_1(x',x'')$ and $k_2(x',x'')$.

Observe that $k_1(x',x'')$ has the same shape as the squared-exponential kernel, for which kernel decomposition was derived in Appendix B.1. For $k_2(x',x'')$, we set

$$\varphi(x',x'') = \sum_{j=1}^{d} w_j (x'_j - x''_j) \tag{47}$$

$$\psi(\varphi) = \cos(\varphi). \tag{48}$$

We note that Assumptions 1 and 2 of Definition 4 are satisfied by this decomposition. For the definition of a bounding function U, i.e., Assumption 3, we have

$$\sup_{x \in T} \sum_{i=1}^{N} c_i \varphi(x, x^{(i)}) = \sup_{x \in T} \sum_{i=1}^{N} c_i \sum_{j=1}^{d} w_j (x_j - x_j^{(i)}) = \sup_{x \in T} \sum_{j=1}^{d} \sum_{i=1}^{N} c_i w_j (x_j - x_j^{(i)}) =$$
$$\sup_{x \in T} \sum_{j=1}^{d} \left(\sum_{i=1}^{N} c_i\right) w_j x_j - \sum_{j=1}^{d} \sum_{i=1}^{N} w_j x_j^{(i)} = \sum_{j=1}^{d} \sup_{x \in T} \bar{w}_j x_j - \beta,$$

where $\bar{w}_j = \left(\sum_{i=1}^{N} c_i\right) w_j$ and $\beta = \sum_{j=1}^{d} \sum_{i=1}^{N} w_j x_j^{(i)}$. As the above is a linear form, we have that the supremum of $\bar{w}_j x_j$ occurs in the point $x_j^* = x_j^U$ if $\bar{w}_j \geq 0$ and in $x_j^* = x_j^L$ otherwise. Thus, we have that $U_{k_2}(\mathbf{c}) = \sum_{j=1}^{d} \bar{w}_j x_j^* - \beta$ is a valid upper-bound function for the sub-kernel k_2.

Non-Stationary Kernel In the non-stationary case, we have:

$$\Sigma_{x',x''} = \sum_{k=1}^{K} \sigma_k^2 \bar{k}\left(x' \odot \theta^{(k)}, x'' \odot \theta^{(k)}\right) \Psi_k(x')^T \Psi_k(x''),$$

where $\theta^{(k)} \geq 0$, $\bar{k}$ is a positive semi-definite, continuous and integrable function and

$$\Psi_k(x) = \left[\cos\left(x^T w_1^{(k)}\right) + \cos\left(x^T w_2^{(k)}\right), \sin\left(x^T w_1^{(k)}\right) + \sin\left(x^T w_2^{(k)}\right)\right].$$

In particular we choose

$$\bar{k}\left(x' \odot \theta, x'' \odot \theta\right) = k(x' \odot \theta)k(x'' \odot \theta) = e^{x' \odot \theta} e^{x'' \odot \theta} = e^{\sum_j \theta_j (x'_j + x''_j)}.$$

Proceeding similarly to the case of stationary kernels, we can analyse each summand and factor in isolation. The final decomposition can then be obtained by using the addition and multiplication formulas for kernel decompositions derived in Appendix B.5 and B.6.

For $\bar{k}\left(x', x''\right)$, we select

$$\varphi(x', x'') = \sum_j \theta_j (x'_j + x''_j)$$
$$\psi(\varphi) = e^{\varphi}.$$

Since φ has exactly the same shape as for that in Equation (47), a similar argument can be used for finding the upper-bound function.

It remains only to find a decomposition for $\Psi_k(x')^T \Psi_k(x'')$. In particular, we have

$$\begin{aligned}
\Psi(x')^T \Psi(x'') &= \left[\cos(x'^T w_1) + \cos(x'^T w_2)\right]\left[\cos(x''^T w_1) + \cos(x''^T w_2)\right] \\
&+ \left[\sin(x'^T w_1) + \sin(x'^T w_2)\right]\left[\sin(x''^T w_1) + \sin(x''^T w_2)\right] \\
&= \cos(x'^T w_1)\cos(x''^T w_1) + \cos(x'^T w_1)\cos(x''^T w_2) \\
&+ \cos(x'^T w_2)\cos(x''^T w_1) + \cos(x'^T w_2)\cos(x''^T w_2) \\
&+ \sin(x'^T w_1)\sin(x''^T w_1) + \sin(x'^T w_1)\sin(x''^T w_2) \\
&+ \sin(x'^T w_2)\sin(x''^T w_1) + \sin(x'^T w_2)\sin(x''^T w_2)
\end{aligned}$$

Again, we can focus on the single factor from the equation above, and rely on the addition and multiplication formulas to obtain the overall result. We consider the first factor, $\cos(x'^T w_1)\cos(x''^T w_1)$, and select:

$$\varphi(x', x'') = \cos(x'^T w_1)\cos(x''^T w_1)$$
$$\psi(\varphi) = \varphi.$$

For the computation of the upper-bound function, we have the following:

$$\sup_x \sum_i c_i \cos(x^{(i),T} w_1)\cos(x^T w_1) \leq \sum_i \sup_x c_i \cos(x^{(i),T} w_1)\cos(x^T w_1) = \sum_i \sup_x \gamma_i \cos(x^T w_1),$$

where we define $\gamma_i = c_i \cos(x^{(i),T} w_1)$. It is thus straightforward to find the maximum of the right-hand-side equation by inspecting the derivatives of the cosine function.

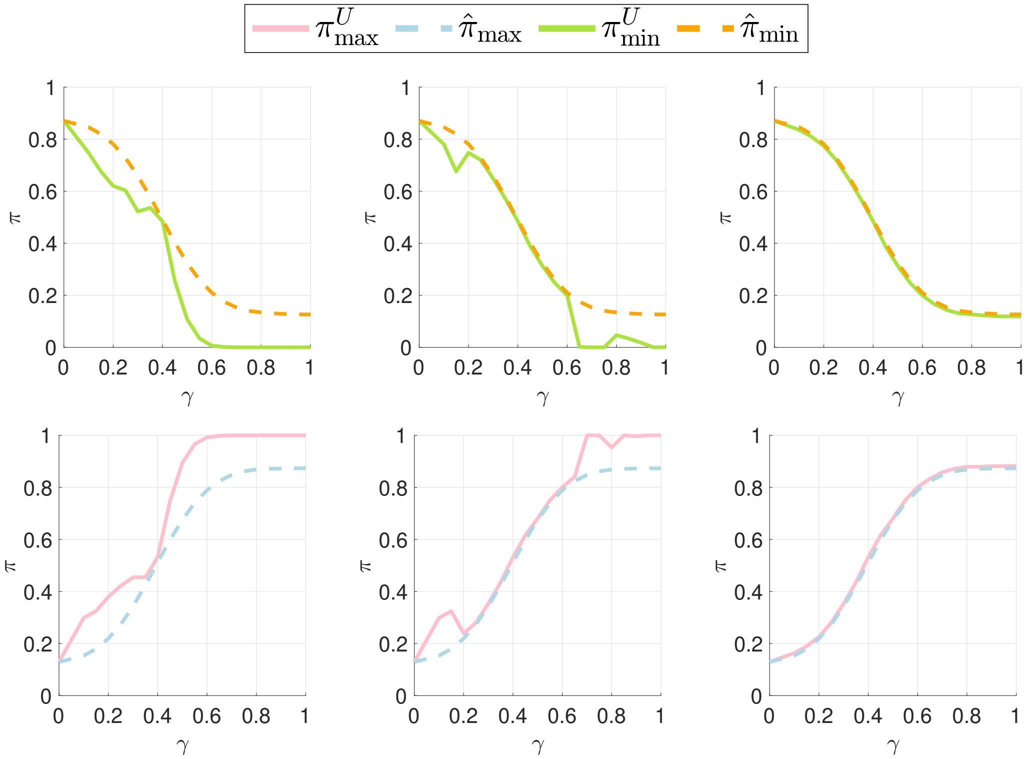

Figure 7: Convergence of upper and lower bounds to the maximum and minimum estimated via grid search for the Synthetic2D data set for varying values of γ. Each column corresponds to the converging computation of the branch-and-bound algorithm for up to a maximum specified number of iterations, namely (from left to right): 10, 100 and 10000. **Top row**: Lower bound (solid line) and estimated minimum (dashed line) for a GP trained on the Synthethic2D data set on a test point from class 2. **Bottom row**: Upper bound (solid line) and estimated maximum (dashed line) for a GP trained on the Synthethic2D data set on a test point from class 1.

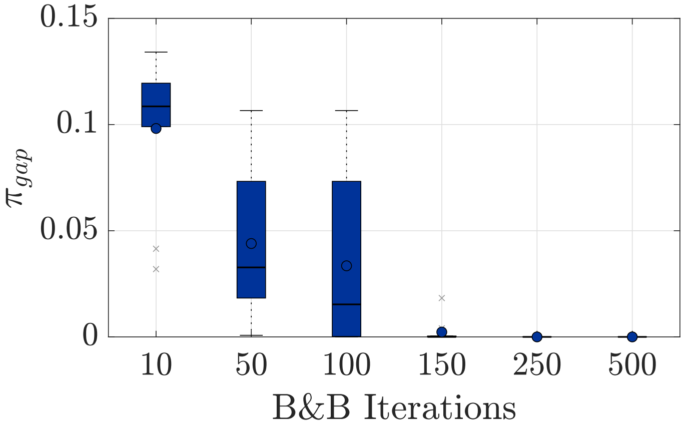

Figure 8: Boxplots of the empirical distribution for the gap (π_{gap}) between the bound and the optimum value estimated for 50 test points selected at random from the Synthetic2D test data set plotted against the number of branch-and-bound iterations.

B.8 Convergence of Kernel Bounding functions

As stated in the proof of Theorem 14, the finite-time convergence of our branch-and-bound methodology relies on the convergence discussed in Propositions 4 and 7, which in turn rests on the convergence of the kernel bounding function U to the actual supremum as the diameter of the input region T shrinks to zero.

The convergence of U itself naturally depends on the explicit form derived for each specific kernel type. For the kernels that we have discussed above the following observations hold, from which convergence can be shown to follow.

For the squared-exponential kernel, the rational quadratic kernel, the Matèrn kernel and for the kernel addition, the bound U we provide is the exact analytical solution of the supremum computations. As a consequence, the bounds trivially converge as the compact region T decreases in size.

Regarding the periodic kernel, in Section B.4 we have shown how to compute a bounding function U that is an over-approximation of the supremum. The over-approximation arises because, in the equations for the supremum computation, we pass the supremum under the sign of summation (Equation 41). However, it is easy to see that, for every finite set of continuous functions $g_i(x) : \mathbb{R}^m \to \mathbb{R}$, $i = 1, \dots, N$, if we set $r = \text{diam}(T)$ then $\sup_{x \in T} \sum_{i=1}^N g_i(x)$ is equal to $\sum_{i=1}^N \sup_{x \in T} g_i(x)$ in the limit of $r \to 0$, while T remains compact. Therefore, the bound U in Equation (42) converges to the actual supremum as T shrinks, and hence the branch-and-bound computation on the periodic kernel converges too.

For the multiplication of kernels, McCormick inequalities are known to converge to the actual multiplication when the region of variability of the multiplication variables decreases (McCormick, 1976), and thus convergence of the bound U in the case of multiplication of kernels ultimately depends on the convergence of the upper and lower bounding $\Sigma'_L, \Sigma'_U, \Sigma''_L$ and Σ''_U that we compute. These, in turn again, depend on the specific forms of the subkernels. If the kernels used are squared-exponential, rational quadratic, Matèrn kernel, periodic kernel or their addition, then convergence follows by the above argument.

Finally, for the generalised spectral kernels, in the stationary case we have that for kernels derived as a summation or multiplication of kernels, for which we compute the exact form of U, convergence follows as above. For the non-stationary case, we have to compute $\sup_x \sum_i c_i \cos(x^{(i),T} w_1) \cos(x^T w_1)$, for which we again give an upper bound by swapping the summation and the supremum signs, as we did in the case of the periodic kernel. Hence, convergence of the function U as T reduces in size follows by a similar argument.

Appendix C. Empirical Convergence Analysis

We empirically investigate the convergence of our branch-and-bound methodology on two points selected from the Synthetic2D test data set. In particular, as an exact computation of $\pi_{\min}(T)$ and $\pi_{\max}(T)$ is not possible, we compare the bounds with an empirical approximation obtained by discretising each γ-ball using 10000 grid points, and computing the minimum and maximum over the grid by brute force search. We refer to the minimum and maximum thus estimated, respectively, as $\hat{\pi}_{\min}$ and $\hat{\pi}_{\max}$.

We report the results of this analysis in Figure 7 for two specular points selected from the test set, namely $[-0.4, 0.4]$ (top row) and $[0.4, -0.4]$ (bottom row). We vary γ from 0 to 1 and report the results for three different values of the maximum number of branch-and-bound iterations (from left to right: 10, 100 and 10000). First, we empirically confirm that the bounds are entirely safe, since the lower bound is always below $\hat{\pi}_{\min}$ and the upper bound is always above $\hat{\pi}_{\max}$. It is interesting to note that there is a non-trivial relationship between the value of γ and the tightness of the bounds. When γ is equal to 0.4 the point that optimises the adversarial classification probability happens to be one of the vertices of the initial search space (i.e., the point $[0, 0]$), to which the branch-and-bound algorithm immediately converges. Convergence is slower when the optimal points lie further away from the vertices of the initial search space. However, after 10000 iterations the two curves overlap almost perfectly. In Figure 8, we show the boxplots of the empirical distribution for the gap (π_{gap}) between the bound and the optimum value estimated for 50 test points selected at random. After 250 iterations the gap between our bound and the brute force empirical estimation is already, on average, almost zero.

References

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

Venkataramanan Balakrishnan, Stephen Boyd, and Silvano 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.

Matthias Bauer, Mark van der Wilk, and Carl Edward Rasmussen. Understanding probabilistic sparse Gaussian process approximations. In *International Conference on Neural Information Processing Systems*, pages 1533–1541, 2016.

Leonard Berrada, Sumanth Dathathri, Robert Stanforth, Rudy Bunel, Jonathan Uesato, Sven Gowal, M Pawan Kumar, et al. Verifying probabilistic specifications with functional Lagrangians. *arXiv preprint arXiv:2102.09479*, 2021.

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

Arno Blaas, Andrea Patane, Luca Laurenti, Luca Cardelli, Marta Kwiatkowska, and Stephen Roberts. Adversarial robustness guarantees for classification with Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 3372–3382. PMLR, 2020.

Ilija Bogunovic, Jonathan Scarlett, Stefanie Jegelka, and Volkan Cevher. Adversarially robust optimization with Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 5760–5770, 2018.

John Bradshaw, Alexander G de G Matthews, and Zoubin Ghahramani. Adversarial examples, uncertainty, and transfer testing robustness in Gaussian process hybrid deep networks. *arXiv preprint arXiv:1707.02476*, 2017.

Rudy Bunel, Jingyue Lu, Ilker Turkaslan, Philip H. S. Torr, Pushmeet Kohli, and M. Pawan Kumar. Branch and bound for piecewise linear neural network verification. *Journal of Machine Learning Research*, 21:42:1–42:39, 2020.

Ginevra Carbone, Matthew Wicker, Luca Laurenti, Andrea Patane, Luca Bortolussi, and Guido Sanguinetti. Robustness of Bayesian neural networks to gradient-based attacks. *In Advances in Neural Information Processing Systems*, 2020.

Luca Cardelli, Marta Kwiatkowska, Luca Laurenti, Nicola Paoletti, Andrea Patane, and Matthew Wicker. Statistical guarantees for the robustness of Bayesian neural networks. In *International Joint Conference on Artificial Intelligence*, pages 5693–5700, 7 2019a.

Luca Cardelli, Marta Kwiatkowska, Luca Laurenti, and Andrea Patane. Robustness guarantees for Bayesian inference with Gaussian processes. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 7759–7768, 2019b.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, pages 39–57, 2017.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019.

Michael B Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. *Journal of the ACM*, 68(1):1–39, 2021.

Adnan Darwiche. Three modern roles for logic in AI. In *ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 229–243, 2020.

AGG De Matthews, J Hron, M Rowland, RE Turner, and Z Ghahramani. Gaussian process behaviour in wide deep neural networks. In *International Conference on Learning Representations*, 2018.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Sašo Džeroski, Damjan Demšar, and Jasna Grbović. Predicting chemical parameters of river water quality from bioindicator data. *Applied Intelligence*, 13(1):7–17, 2000.

Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *IEEE Symposium on Security and Privacy*, pages 3–18, 2018.

Kathrin Grosse, David Pfaff, Michael Thomas Smith, and Michael Backes. How wrong am I? Studying adversarial examples and their impact on uncertainty in Gaussian process machine learning models. *arXiv preprint arXiv:1711.06598*, 2017.

Kathrin Grosse, David Pfaff, Michael T Smith, and Michael Backes. The limitations of model uncertainty in adversarial settings. *arXiv preprint arXiv:1812.02606*, 2018.

Daniel Hernández-Lobato, José M Hernández-Lobato, and Pierre Dupont. Robust multi-class Gaussian process classification. In *Advances in Neural Information Processing Systems*, pages 280–288, 2011.

Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety verification of deep neural networks. In *International Conference on Computer Aided Verification*, pages 3–29. Springer, 2017.

Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

Hyun-Chul Kim and Zoubin Ghahramani. Outlier robust Gaussian process classification. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition and Structural and Syntactic Pattern Recognition*, pages 896–905. Springer, 2008.

Yann LeCun. The MNIST database of handwritten digits. 1998. URL `http://yann.lecun.com/exdb/mnist/`.

David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part I - Convex underestimating problems. *Mathematical Programming*, 10(1):147–175, 1976.

Thomas P Minka. Expectation propagation for approximate Bayesian inference. In *Conference on Uncertainty in Artificial Intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming.* SIAM, 1994.

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

Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6(Dec): 1939–1959, 2005.

Carl Edward Rasmussen and Hannes Nickisch. Gaussian Processes for Machine Learning (GPML) toolbox. *Journal of Machine Learning Research*, 11(Nov):3011–3015, 2010.

Carl Edward Rasmussen and Christopher K Williams. *Gaussian Processes for Machine Learning.* MIT press Cambridge, MA, 2006.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you? Explaining the predictions of any classifier. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

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

Wenjie Ruan, Xiaowei Huang, and Marta Kwiatkowska. Reachability analysis of deep neural networks with provable guarantees. In *International Joint Conference on Artificial Intelligence*, pages 2651–2659, 2018.

Yves-Laurent Kom Samo and Stephen Roberts. Generalized spectral kernels. *arXiv preprint arXiv:1506.02236*, 2015.

Michael Thomas Smith, Kathrin Grosse, Michael Backes, and Mauricio A Alvarez. Adversarial vulnerability bounds for Gaussian process classification. *arXiv preprint arXiv:1909.08864*, 2019.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems*, 18:1257–1264, 2005.

Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy? A comprehensive study on the robustness of 18 deep image classification models. In *European Conference on Computer Vision*, pages 631–648, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Jarno Vanhatalo, Jaakko Riihimäki, Jouni Hartikainen, Pasi Jylänki, Ville Tolvanen, and Aki Vehtari. GPstuff: Bayesian modeling with Gaussian processes. *Journal of Machine Learning Research*, 14(Apr):1175–1179, 2013.

Matthew Wicker, Luca Laurenti, Andrea Patane, and Marta Kwiatkowska. Probabilistic safety for Bayesian neural networks. In *Conference on Uncertainty in Artificial Intelligence*, pages 1198–1207. PMLR, 2020.

Matthew Wicker, Luca Laurenti, Andrea Patane, Zhuotong Chen, Zheng Zhang, and Marta Kwiatkowska. Bayesian inference with certifiable adversarial robustness. In *International Conference on Artificial Intelligence and Statistics*, pages 2431–2439. PMLR, 2021.

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

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Advances in Neural Information Processing Systems*, pages 4939–4948, 2018.