Robustness Guarantees for Bayesian Inference with Gaussian Processes*

Luca Cardelli,[1] **Marta Kwiatkowska,**[2] **Luca Laurenti,**[2] **Andrea Patane**[2]
[1]Microsoft Research Cambridge,
[2]University of Oxford
luca.a.cardelli@gmail.com, marta.kwiatkowska@cs.ox.ac.uk, luca.laurenti@cs.ox.ac.uk,
andrea.patane@cs.ox.ac.uk

Abstract

Bayesian inference and Gaussian processes are widely used in applications ranging from robotics and control to biological systems. Many of these applications are safety-critical and require a characterization of the uncertainty associated with the learning model and formal guarantees on its predictions. In this paper we define a robustness measure for Bayesian inference against input perturbations, given by the probability that, for a test point and a compact set in the input space containing the test point, the prediction of the learning model will remain $\delta-$close for all the points in the set, for $\delta > 0$. Such measures can be used to provide formal probabilistic guarantees for the absence of adversarial examples. By employing the theory of Gaussian processes, we derive upper bounds on the resulting robustness by utilising the Borell-TIS inequality, and propose algorithms for their computation. We evaluate our techniques on two examples, a GP regression problem and a fully-connected deep neural network, where we rely on weak convergence to GPs to study adversarial examples on the MNIST dataset.

Introduction

The widespread deployment of machine learning models, coupled with the discovery of their fragility against carefully crafted manipulation of training and/or test samples (Biggio and Roli 2017; Grosse et al. 2017a; Szegedy et al. 2013), calls for safe approaches to AI to enable their use in safety-critical applications, as argued, e.g., in (Seshia, Sadigh, and Sastry 2016; Dreossi, Donzé, and Seshia 2017). Bayesian techniques, in particular, provide a principled way of combining a-priori information into the training process, so as to obtain an a-posteriori distribution on test data, which also takes into account the uncertainty in the learning process. Recent advances in Bayesian learning include adversarial attacks (Grosse et al. 2017b) and methods to compute pointwise uncertainty estimates in Bayesian deep learning (Gal and Ghahramani 2016). However, much of the work on formal guarantees for machine learning models has focused on non-Bayesian models, such as deep neural networks (NNs) (Huang et al. 2017; Hein and Andriushchenko 2017) and, to the best of our knowledge, there is no work directed at providing formal guarantees for the absence of adversarial local input perturbations in Bayesian prediction settings.

Gaussian processes (GPs) are a class of stochastic processes that are, due to their many favourable properties, widely employed for Bayesian learning (Rasmussen 2004), with applications spanning robotics, control systems and biological processes (Sadigh and Kapoor 2015; Laurenti et al. 2017; Bortolussi et al. 2018). Further, driven by pioneering work that first recognized the convergence of fully-connected NNs to GPs in the limit of infinitely many neurons (Neal 2012), GPs have been used recently as a model to characterize the behaviour of NNs in terms of convergence analysis (Matthews et al. 2018), approximated Bayesian inference (Lee et al. 2017) and training algorithms (Chouza, Roberts, and Zohren 2018).

In this paper we compute formal local robustness guarantees for Bayesian inference with GP priors. The resulting guarantees are probabilistic, as they take into account the uncertainty intrinsic in the Bayesian learning process and explicitly work with the a-posteriori output distribution of the GP. More specifically, given a GP model trained on a given data set, a test input point and a neighborhood around the latter, we are interested in computing the probability that there exists a point in the neighbourhood such that the prediction of the GP on the latter differs from the initial test input point by at least a given threshold. This implicitly gives guarantees on the absence of adversarial examples, that is, input samples that trick a machine learning model into performing wrong predictions.

Unfortunately, computing such a probability is far from trivial. In fact, given a compact set $T \subseteq \mathbb{R}^m, m > 0$, and $x^* \in T$, the above measure reduces to computing the probability that there exists a function f sampled from the GP such that there exists $x \in T$ for which $||f(x^*) - f(x)|| > \delta$, where $\delta > 0$ and $|| \cdot ||$ is a metric norm. Since the set T is composed of an infinite number of points, computing such a measure for general stochastic processes is extremely challenging. However, for GPs we can obtain tight upper bounds on the above probability by making use of inequalities developed in the theory of GPs, such as the *Borell-TIS inequality* (Adler and Taylor 2009) and the *Dudley's Entropy Integral*

*This work has been partially supported by a Royal Society Professorship, by the EU's Horizon 2020 program under the Marie Skłodowska-Curie grant No 722022 and by the EPSRC Programme Grant on Mobile Autonomy (EP/M019918/1).

(Dudley 1967). To do this, we need to obtain lower and upper bounds on the extrema of the a-posteriori GP mean and variance functions on neighborhoods of a given test point. We obtain these bounds by constructing lower and upper approximations for the GP kernel as a function of the test point, which are then propagated through the GP inference formulas. Then, safe approximations for these values are obtained by posing a series of optimization problems that can be solved either analytically or by standard quadratic convex optimization techniques. We illustrate the above framework with explicit algorithmic techniques for GPs built with squared-exponential and ReLU kernel.

Finally, we apply the methods presented here to characterize the robustness of GPs with ReLU kernel trained on a subset of images included in the MNIST dataset. Relying on the weak convergence between fully-connected NNs with ReLU activation functions and the corresponding GPs with ReLU kernel, we analyze the behaviour of such networks on adversarial images in a Bayesian setting. We use SIFT (Lowe 2004) to focus on important patches of the image, and perform feature-level safety analyses of test points included in the dataset. We apply the proposed methods to evaluate the resilience of features against (generic) adversarial perturbations bounded in norm, and discuss how this is affected by stronger perturbations and different misclassification thresholds. We perform a parametric optimization analysis of maximum prediction variance around specific test points in an effort to characterize active defenses against adversarial examples that rely on variance thresholding. In the examples we studied, we have consistently observed that, while an increased number of training samples may significantly help detect adversarial examples by means of prediction uncertainty, the process may be undermined by more complex architectures.

In summary, the paper makes the following main contributions:

- We provide tight upper bounds on the probability that the prediction of a Gaussian process remains close to a given test point in a neighbourhood, which can be used to quantify local robustness against adversarial examples.
- We develop algorithmic methods for the computation of extrema of GP mean and variance over a compact set.
- Relying on convergence between fully-connected NNs and GPs, we apply the developed methods to provide feature-level analysis of the behaviour of the former on the MNIST dataset[1].

An extended version of this paper with proofs and further details can be found in (Cardelli et al. 2018).

Why probabilistic local robustness guarantees? Our results provide formal local robustness guarantees in the sense that the resulting bounds are sound with respect to a neighbourhood of an input, and numerical methods have not been used. This enables certification for Bayesian methods that is necessary in safety-critical applications, and is in contrast with many existing pointwise approaches to detect adversarial examples in Bayesian models, which are generally based on heuristics, such as to reject test points with high uncertainty (Li and Gal 2017; Feinman et al. 2017). We illustrate the intuition with the following simple example.

Example 1. *Let $(\mathbf{z}(x), x \in \mathbb{N})$ be a zero-mean stochastic process with values in $\mathbb{R}$. Consider the following widely used definition of safety for a set $T = [x_1, ..., x_{10}]$*

$$P_{safe}(\mathbf{z}, T, \delta) = \textit{Prob}(\forall x \in T,\, \mathbf{z}(x) < \delta), \qquad (1)$$

where $\delta \in \mathbb{R}$ is a given threshold. Assume that, for all $x_i, x_j \in T$, $\mathbf{z}(x_i)$ and $\mathbf{z}(x_j)$ are independently and equally distributed random variables such that for each $x \in T$ we have Prob$(\mathbf{z}(x) < \delta) = 0.85$. Then, if we compute the above property we obtain

$$P_{safe}(\mathbf{z}, T, \delta) = 0.85^{10} \approx 0.197.$$

Thus, even though at each point $\mathbf{z}(x)$ has relatively high probability of being safe, $P_{safe}(\mathbf{z}, T, \delta)$ is still small. This is because safety, as defined in Eqn 1, depends on a set of points, and this must be accounted for to give robustness guarantees for a given stochastic model. Note that, to simplify, we used a discrtete set T, but the same reasoning remains valid even if $T \subseteq \mathbb{R}^m, m > 0$, as in this paper.

Related Work

Existing formal approaches for machine learning models mostly focus on computing non-probabilistic local safety guarantees (Raghunathan, Steinhardt, and Liang 2018; Huang et al. 2017; Ruan, Huang, and Kwiatkowska 2018) and generally neglect the uncertainty of the learning process, which is intrinsic in a Bayesian model. Recently, empirical methods to detect adversarial examples for Bayesian NNs that utilise pointwise uncertainty have been introduced (Li and Gal 2017; Feinman et al. 2017). However, these approaches can be fooled by attacks that generate adversarial examples with small uncertainty as shown in (Carlini and Wagner 2017). Unfortunately, obtaining formal guarantees for Bayesian NNs is challenging since their posterior distribution, which can be obtained in closed form for GPs, is generally analytically intractable (Gal and Ghahramani 2016). In (Grosse et al. 2017b) attacks for Bayesian inference with Gaussian processes based on local perturbations of the mean have been presented.

Notions of safety for Gaussian processes have been recently studied in the context of system design for stochastic models (see, e.g. (Wachi et al. 2018; Bartocci et al. 2015; Sadigh and Kapoor 2015; Sui et al. 2015)). In (Sadigh and Kapoor 2015), the authors synthesize safe controllers against *Probabilistic Signal Temporal Logic (PrSTL)* specifications, which suffer from the issue illustrated in Example 1. Another related approach is that in (Sui et al. 2015), where the authors build on (Srinivas et al. 2012) and introduce SAFEOPT, a Bayesian optimization algorithm that additionally guarantees that, for the optimized parameters, with high probability the resulting objective function (sampled from a GP) is greater than a threshold. However, they do not give

[1] Code available at: https://github.com/andreapatane/checkGP.

guarantees against perturbation of the synthesized parameters. For instance, their method cannot guarantee that the resulting behaviour will still be safe and close to the optimal value if parameters corrupted by noise are applied. Our approach allows one to quantify such a probability. We should also stress that, while it is often the case that the guarantees provided by existing algorithms are statistical (i.e., given in terms of confidence intervals), the bounds presented in this paper are probabilistic.

Problem Formulation

We consider a Gaussian process $\big(\mathbf{z}(x), x \in \mathbb{R}^m, m > 0\big)$ with values in $\mathbb{R}^n, n > 0$ and with a Gaussian probability measure P such that, for any $x_1, x_2, ..., x_k \in \mathbb{R}^m$, $P(\mathbf{z}(x_1), \mathbf{z}(x_2), ..., \mathbf{z}(x_k))$ is a multivariate normal distribution[2]. We consider Bayesian inference for $\mathbf{z}$. That is, as illustrated in detail in the next section, given a dataset $\mathcal{D} = \{\mathbf{z}(x_i) = y_i, i \in \{1, ..., N\}\}$ of $|\mathcal{D}| := N$ samples, we consider the process

$$\mathbf{z}(x)|\mathcal{D},\, x \in \mathbb{R}^m,$$

which represents the conditional distribution of $\mathbf{z}$ given the set of observations in $\mathcal{D}$. The first problem we examine is Problem 1, where we want to compute the probability that local perturbations of a given test point result in predictions that remain close to the original.

Problem 1. *(Probabilistic Safety). Consider the training dataset $\mathcal{D}$. Let $T \subseteq \mathbb{R}^m$ and fix $x^* \in T$. For $\delta > 0$ call*

$$\phi_1^i(x^*, T, \delta \,|\, \mathcal{D}) = P(\exists x' \in T \; s.t. \; \big(\mathbf{z}^{(i)}(x^*) - \mathbf{z}^{(i)}(x')\big) > \delta \,|\, \mathcal{D}),$$

where $\mathbf{z}^{(i)}$ is the i-th component of $\mathbf{z}$. Then we say that component i in $\mathbf{z}$ is safe with probability $1 - \epsilon > 0$ for x^* with respect to set T and perturbation $\delta > 0$ iff

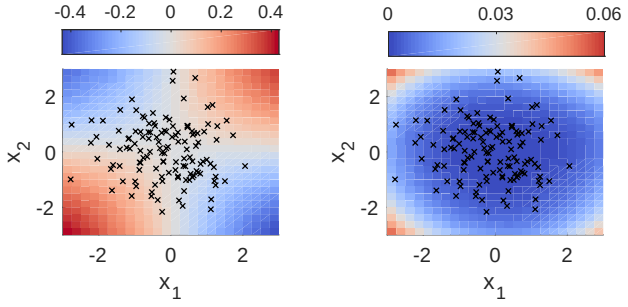
$$\phi_1^i(x^*, T, \delta|\mathcal{D}) \leq \epsilon. \tag{2}$$

Intuitively, we consider a test point x^* and a compact set T containing x^*, and compute the probability that the predictions of $\mathbf{z}$ remain $\delta-$close for each $x' \in T$. We consider the components of the GP individually and with sign, enabling one-sided analysis. Note that T is composed of an uncountable number of points, making the probability computation challenging. Moreover, Problem 1 will still represent a sound notion of safety even in the case that a distribution on the input space can be assumed. Problem 1 can be generalized to local invariance of $\mathbf{z}$ with respect to a given metric (Problem 2 below). In the next section, for the corresponding solution, we will work with the L_1 norm, but all the results can be easily extended to any L_p norm, including L_∞.

Problem 2. *(Probabilistic Invariance) Consider the training dataset $\mathcal{D}$. Let $T \subseteq \mathbb{R}^m$ and assume $x^* \in T$. For metric $||\cdot||_d : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ and $\delta > 0$ call*

$$\phi_2(x^*, T, \delta|\mathcal{D}) = P(\exists x' \in T \; s.t. \; ||\mathbf{z}(x') - \mathbf{z}(x^*))||_d > \delta|\mathcal{D})$$

[2]In this paper we assume $\mathbf{z}$ is a separable stochastic process. This is a standard and common assumption (Adler and Taylor 2009). The separability of $\mathbf{z}$ guarantees that Problem 1 and 2 are measurable.



(a) Mean. (b) Variance.

Figure 1: Results of GP training.

Then we say that $\mathbf{z}$ is $\delta-$invariant with respect to metric $||\cdot||_d$ for x^* in T and perturbation $\delta > 0$ with probability $1 - \epsilon > 0$ iff

$$\phi_2(x^*, T, \delta|\mathcal{D}) \leq \epsilon. \tag{3}$$

Probabilistic invariance, as defined in Problem 2, bounds the probability that each function sampled from $\mathbf{z}$ remains within a distance of at most δ to the initial point. Note that both Problem 1 and 2 quantify the probability of how the output of a learning process changes its value in a set around a given test input point, which implicitly gives probabilistic guarantees for local robustness against adversarial examples. In the next section, in Theorem 1 and 2, we give analytic upper bounds for Problem 1 and 2. In fact, analytic distributions of the supremum of a GP, which would allow one to solve the above problems, are known only for a very limited class of GPs (and always for GPs evolving over time) (Adler and Taylor 2009), making exact computation impossible. However, first, we illustrate the intuition behind the problems studied here on a GP regression problem.

Example 2. *We consider a regression problem taken from (Bach 2009), where we generate 128 samples from a random two-dimensional covariance matrix, and define labels as a (noisy) quadratic polynomial of the two input variables. We train a GP with squared-exponential kernel on this dataset, using a maximum likelihood estimation of the kernel hyper-parameters (Rasmussen 2004). The mean and variance of the GP obtained after training are plotted in Figure 1, along with the samples used for training. Consider the origin point $x^o = (0, 0)$, let $\gamma = (0.1, 0.1)$ and define $T^o_\gamma = [x^o - \gamma, x^o + \gamma]$. As x^o is a saddle point for the mean function, variations of the mean around it are relatively small. Analogously, the variance function exhibits a flat behaviour around x^o, meaning greater confidence of the GP in performing predictions around x^o. As such we expect realizations of the GP to be consistently stable in a neighbourhood of x^o, which in turn translates to low values for $\phi_1(x^o, T^o_\gamma, \delta)$ and $\phi_2(x^o, T^o_\gamma, \delta)$, where in ϕ_1 and ϕ_2, to simplify the notation, we omit the dataset used for training. On the other hand, around $x^* = (3, 3)$ the a-posteriori mean changes quickly and the variance is high, reflecting higher uncertainty. Hence, letting $T^*_\gamma = [x^* - \gamma, x^* + \gamma]$, we expect*

the values of $\phi_1(x^*, T^*_\gamma, \delta)$ and $\phi_2(x^*, T^*_\gamma, \delta)$ to be greater than those computed for x^o.

In the next section we show how $\phi_1(x, T_\gamma, \delta)$ and $\phi_2(x, T_\gamma, \delta)$ can be computed to quantify the uncertainty and variability of the predictions around x^o and x^*.

Theoretical Results

Since $\mathbf{z}$ is a Gaussian process, its distribution is completely defined by its mean $\mu : \mathbb{R}^m \to \mathbb{R}^n$ and covariance (or kernel) function $\Sigma : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^{n\times n}$. Consider a set of training data $\mathcal{D} = \{\mathbf{z}(x_i) = y_i, i \in \{1, ..., N\}\}$, and call $\mathbf{y} = [y_1, ..., y_N]$. Training $\mathbf{z}$ in a Bayesian framework is equivalent to computing the distribution of $\mathbf{z}$ given the dataset $\mathcal{D}$, that is, the distribution of the process

$$\bar{\mathbf{z}} = \mathbf{z} \,|\, \mathcal{D}.$$

Given a test point $x^* \in \mathbb{R}^m$ and $x_1, ..., x_N$ training inputs in $\mathcal{D}$, consider the joint distribution $[\mathbf{z}(x^*), \mathbf{z}(x_1), ..., \mathbf{z}(x_N)]$, which is still Gaussian with mean and covariance matrix given by

$$\mu = [\mu(x^*), \mu(x_1), ..., \mu(x_N)] \quad \Sigma = \begin{bmatrix} \Sigma_{x^*,x^*} & \Sigma_{x^*,\mathcal{D}} \\ \Sigma^T_{x^*,\mathcal{D}} & \Sigma_{\mathcal{D},\mathcal{D}} \end{bmatrix},$$

where $\Sigma_{D,D}$ is the covariance matrix relative to vector $[\mathbf{z}(x_1), ..., \mathbf{z}(x_{|\mathcal{D}|})]$. Then, it follows that $\bar{\mathbf{z}}$ is still Gaussian with mean and covariance matrix defined as follows:

$$\bar{\mu}(x^*) = \mu(x^*) + \Sigma_{x^*,\mathcal{D}}\Sigma^{-1}_{\mathcal{D},\mathcal{D}}(\mathbf{y} - \mu_{\mathcal{D}}) \tag{4}$$

$$\bar{\Sigma}_{x^*,x^*} = \Sigma_{x^*,x^*} - \Sigma_{x^*,\mathcal{D}}\Sigma^{-1}_{\mathcal{D},\mathcal{D}}\Sigma^T_{x^*,\mathcal{D}}, \tag{5}$$

where $\mu_{\mathcal{D}} = [\mu(x_1), ..., \mu(x_N)]$. Hence, for GPs the distribution of $\bar{\mathbf{z}}(x^*)$ can be computed exactly.

Given two test points x^*_1, x^*_2 and $x^* = [x^*_1, x^*_2]$, the above calculations can still be applied to compute the joint distribution

$$\bar{\mathbf{z}}(x^*) = \big([\mathbf{z}(x^*_1), \mathbf{z}(x^*_2)] \,|\, \mathcal{D}\big).$$

In particular, $\bar{\mathbf{z}}(x^*)$ is still Gaussian and with mean $\bar{\mu}$ and covariance matrix $\bar{\Sigma}$ given by Eqns (4) and (5) but with $\mu(x^*) = [\mu(x^*_1), \mu(x^*_2)]$ and $\Sigma_{x^*,x^*} = \begin{bmatrix} \Sigma_{x^*_1,x^*_1} & \Sigma_{x^*_1,x^*_2} \\ \Sigma^T_{x^*_1,x^*_2} & \Sigma_{x^*_2,x^*_2} \end{bmatrix}$. From $\bar{\mathbf{z}}(x^*)$ we can obtain the distribution of the following random variable

$$\mathbf{z}^o(x^*_1, x^*_2) = \big(\mathbf{z}(x^*_1) - \mathbf{z}(x^*_2)\big) \,|\, \mathcal{D},$$

which represents the difference of $\mathbf{z}$ at two distinct test points after training. It is straightforward to show that, given $B \in \mathbb{R}^{n\times 2n}$ such that $B = [I; -I]$, where I is the identity matrix of dimension n, $\mathbf{z}^o(x^*_1, x^*_2)$ is Gaussian with mean and variance

$$\mu^o(x^*_1, x^*_2) = B\bar{\mu}(x^*) \qquad \Sigma^o_{x^*_1,x^*_2} = B\bar{\Sigma}_{x^*,x^*}B^T.$$

$\mathbf{z}^o(x^*_1, x^*_2)$ is the distribution of how $\mathbf{z}$, after training, changes with respect to two different test points. However, to solve Problem 1 and 2, we need to take into account all the test points in $T \subseteq \mathbb{R}^m$ and compute the probability that in at least one of them $\mathbf{z}^o$ exits from a given set of the output space. This is done in Theorem 1 by making use of the Borell-TIS inequality and of the Dudley's entropy integral (Adler and Taylor 2009; Dudley 1967). The above inequalities allow one to study Gaussian processes by appropriately defining a metric on the variance of the GPs. In order to define such a metric we call $\hat{\mathbf{z}}^o$ the GP with the same covariance matrix as $\mathbf{z}^o$ but with zero mean and $\hat{\mathbf{z}}^{o,(i)}$ its i-th component. For $i \in \{1, ..., n\}$, a test point $x^* \in \mathbb{R}^m$, and $x_1, x_2 \in \mathbb{R}^m$ we define the (pseudo-)metric $d^{(i)}_{x^*}(x_1, x_2)$ by

$$\begin{aligned} d^{(i)}_{x^*}(x_1, x_2) =& \sqrt{\mathbb{E}[(\hat{\mathbf{z}}^{o,(i)}(x^*, x_1) - \hat{\mathbf{z}}^{o,(i)}(x^*, x_2))^2]} \\ =& \sqrt{\mathbb{E}[(\hat{\mathbf{z}}^{(i)}(x_2) - \hat{\mathbf{z}}^{(i)}(x_1))^2]}, \end{aligned} \tag{6}$$

where $\hat{\mathbf{z}}^{(i)}$ is the i-th component of the zero-mean version of $\bar{\mathbf{z}}$. Note that $d^{(i)}_{x^*}(x_1, x_2)$ does not depend on x^*. Additionally, we assume there exists a constant $K^{(i)}_{x^*} > 0$ such that for a compact $T \subseteq \mathbb{R}^m$ and $x^*, x_1, x_2 \in T$[3]

$$d^{(i)}_{x^*}(x_1, x_2) \leq K^{(i)}_{x^*}||x_1 - x_2||_2.$$

Now, we are finally ready to state the following theorem.

Theorem 1. *Assume $T \subseteq \mathbb{R}^m, m > 0$, is a hyper-cube with layers of length $D > 0$. For $x^* \in T, \delta > 0$, and $i \in \{1, \dots, n\}$ let*

$$\begin{aligned} &\eta_i = \delta - \big(sup_{x\in T}\mu^{o,(i)}(x^*, x) + \\ &12\int_0^{\frac{1}{2}sup_{x_1,x_2\in T} d^{(i)}_{x^*}(x_1,x_2)} \sqrt{ln\left(\left(\frac{\sqrt{m}K^{(i)}_{x^*}D}{z} + 1\right)^m\right)}dz\big). \end{aligned}$$

Assume $\eta_i > 0$. Then, it holds that

$$\phi^i_1(x^*, T, \delta|\mathcal{D}) \leq \hat{\phi}^i_1(x^*, T, \delta|\mathcal{D}) := e^{-\frac{\eta_i^2}{2\xi^{(i)}}},$$

where $\xi^{(i)} = \sup_{x\in T}\Sigma^{o,(i,i)}_{x^*,x}$ is the supremum of the component (i, i) of the covariance matrix $\Sigma^o_{x^*,x}$.

Proof. [Sketch.]

$$\begin{aligned} &\phi^i_1(x^*, T, \delta|\mathcal{D}) \\ &\quad \text{(By definition of } \phi_1\text{)} \\ =&P(\exists x \in T\, s.t.\, \big(\mathbf{z}^{(i)}(x) - \mathbf{z}^{(i)}(x^*) > \delta \,|\, \mathcal{D}\big) \\ &\quad \text{(By definition of supremum)} \\ =&P\big(\sup_{x\in T}\mathbf{z}^{o,(i)}(x^*, x) > \delta\big) \\ &\quad \text{(By linearity of GPs)} \\ =&P\big(\sup_{x\in T}\hat{\mathbf{z}}^{o,(i)}(x^*, x) + \mathbb{E}[\mathbf{z}^{o,(i)}(x^*, x)] > \delta\big) \\ &\quad \text{(By definition of supremum)} \\ \leq&P\big(\sup_{x\in T}\hat{\mathbf{z}}^{o,(i)}(x^*, x) > \delta - sup_{x_1\in T}\mathbb{E}[\mathbf{z}^{o,(i)}(x^*, x_1)]\big). \end{aligned}$$

where $\hat{\mathbf{z}}^{o,(i)}(x^*, x)$ is the zero mean Gaussian process with same variance of $\mathbf{z}^{o,(i)}(x^*, x)$. The last inequality can be bounded from above using the Borell-TIS inequality (Adler

[3]Note that here we work with the L_2 norm, but any other L_p metric would work.

and Taylor 2009). To use such an inequality we need to derive an upper bound of $\mathbb{E}[sup_{t\in T}\mathbf{z}^{o,(i)}(x^*,x)]$. This can be done by employing the Dudley's Entropy integral (Dudley 1967).

The extended version of the proof can be found in the Appendix. □

In Theorem 1 we derive $\hat{\phi}_1^i(x^*,T,\delta|\mathcal{D})$ as an upper bound for $\phi_1^i(x^*,T,\delta|\mathcal{D})$. Considering that $\mathbf{z}$ is a Gaussian process, it is interesting to note that the resulting bounds still follow an exponential distribution. From Theorem 1 we have the following result.

Theorem 2. *Assume $T \subseteq \mathbb{R}^m, m > 0$ is a hyper-cube with layers of length $D > 0$. For $x^* \in T, \delta > 0$ let*

$$\bar{\eta}_i = \frac{\delta - sup_{x\in T}|\mu^o(x^*,x)|_1}{n} - 12\int_0^{\frac{1}{2}sup_{x_1,x_2\in T} d_{x^*}^{(i)}(x_1,x_2)} \sqrt{ln\left(\left(\frac{\sqrt{m}K_{x^*}^{(i)}D}{z}+1\right)^m\right)}dz.$$

For each $i \in \{1,...,n\}$ assume $\bar{\eta}_i > 0$. Then, it holds that

$$\phi_2(x^*,T,\delta|\mathcal{D}) \leq \hat{\phi}_2(x^*,T,\delta|\mathcal{D}) := 2\sum_{i=1}^{n} e^{-\frac{\bar{\eta}_i^2}{2\xi^{(i)}}},$$

where $\xi^{(i)} = \sup_{x\in T}\Sigma_{x^*,x}^{o,(i,i)}$.

Note that in Theorem 1 and 2 we assume that T is a hyper-cube. However, proofs of both theorems can be easily extended to more general compact sets, at a cost of more complex analytic expressions or less tight bounds.

Both theorems require the computation of a set of constants, which depends on the particular kernel. In particular, $\xi^{(i)}$ and $\sup_{x\in T}\mu^{o,(i)}(x^*,x)$ are upper bound of variance and mean a-posteriori while, for a test point x^*, $K_{x^*}^{(i)}$ and $\sup_{x_1,x_2\in T} d_{x^*}^{(i)}(x_1,x_2)$ represent local Lipschitz constant and upper bound for $d_{x^*}^{(i)}$ in T. In the next section, we show how these constants can be computed.

Example 3. *We illustrate the upper bounds for ϕ_1 and ϕ_2, as given by Theorem 1 and 2, on the GP introduced in Example 2. Figure 2 shows the values obtained for $\hat{\phi}_1$ and $\hat{\phi}_2$ on x^o and x^* for δ between 0 and 0.2. We observe that values computed for x^* are consistently greater than those computed for x^o, which captures and probabilistically quantifies the increased uncertainty of the GP around x^*, as well as the increased ratio of mean variation around it (see Figure 1). Notice also that values for $\hat{\phi}_1$ are always smaller than the corresponding $\hat{\phi}_2$ values. This is a direct consequence of the fact that probabilistic invariance is a stronger requirement than probabilistic safety, as defined in Problem 1, as the latter is not affected by variations that tend to increase the value of the GP output (translating to increased confidence in classification settings). In Figure 2 we also compare the upper bounds obtained with estimation for ϕ_1 and ϕ_2 based on sampling of the GP in a discrete grid around the test points. Specifically, we sample 10000 functions from the GP and evaluate them in 2025 points uniformly spaced around the test point. We remark that this provides us with just an empirical approximation of the actual values of ϕ_1 and ϕ_2, referred to as $\bar{\phi}_1$ and $\bar{\phi}_2$ respectively. Note also that $\bar{\phi}_1$ and $\bar{\phi}_2$ have an exponential decay. The results suggest that the approximation is tighter around x^o than around x^*. In fact, higher variance will generally imply a looser bound, also due to the over-approximations introduced in the computation of the constants required in the theorems.*

Constant Computation

We introduce a general framework for the computation of the constants involved in the bounds presented in the previous section with an approach based on a generalisation of that of (Jones, Schonlau, and Welch 1998) for squared-exponential kernels in the setting of Kriging regression. Namely, we assume the existence of a suitable decomposition of the kernel function as $\Sigma_{x,x_i} = \psi_\Sigma(\varphi_\Sigma(x,x_i))$ for all x and $x_i \in \mathbb{R}^m$, such that:

1. $\varphi_\Sigma : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is a continuous function;
2. $\psi_\Sigma : \mathbb{R} \to \mathbb{R}$ is differentiable, with $\frac{d\psi_\Sigma}{d\varphi_\Sigma}$ continuous;
3. $\sup_{x\in T}\sum_{i=1}^{N} c_i\varphi_\Sigma(x,x_i)$ can be computed for each $c_i \in \mathbb{R}$ and $x_i \in \mathbb{R}^m$, $i = 1,\dots,N$.

While assumptions 1 and 2 usually follow from smoothness of the kernel used, assumption 3 depends on the particular φ_Σ defined. Intuitively, φ_Σ should represent the smallest building block of the kernel which captures the dependence on the two input points. For example for the squared exponential kernel this has the form of a separable quadratic polynomial so that assumption 3 is verified. Similarly, for the ReLU kernel φ_Σ can be defined as the dot product between the two input points. A list of valid decomposition functions φ_Σ and ψ_Σ for commonly used kernels that satisfy Assumptions 1,2, and 3 is given in the Appendix.

Assumptions 1 and 2 guarantee the existence for every $x_i \in \mathbb{R}^m$ of a set of constants a_L^i, b_L^i, a_U^i and b_U^i such that:

$$a_L^i + b_L^i\varphi_\Sigma(x,x_i) \leq \Sigma_{x,x_i} \leq a_U^i + b_U^i\varphi_\Sigma(x,x_i) \quad \forall x \in T. \tag{7}$$

In fact, it follows from those that ψ_Σ has a finite number of flex points. Hence, we can iteratively find lower and upper approximation in convex and concave parts, and merge them together as detailed in (Cardelli et al. 2018). The key point is that, due to linearity, this upper and lower bound on the kernel can be propagated through the inference equations for Gaussian processes, so as to obtain lower and upper linear bounds on the a-posteriori mean and variance with respect to φ_Σ. Thanks to assumption 3, these bounds can be solved for optimal points exactly, thus providing formal lower and upper values on optimization over a-posteriori mean and variance of the Gaussian process in T. The described approach can be used to compute $\xi^{(i)}$, $K_{x^*}^{(i)}$ and $\sup_{x\in T}\mu^o(x^*,x)$. In the following subsection we give details for the computation of $\sup_{x\in T}\mu^o$. We refer to the (Cardelli et al. 2018) for the details on the computations of the other constants.

Mean Computation As x^* is fixed, we have that $\sup_{x\in T}\mu^o = \bar{\mu}(x^*) - \inf_{x\in T}\bar{\mu}(x)$, hence we need just to compute $\inf_{x\in T}\bar{\mu}(x)$. Using Eqn (7) we can compute

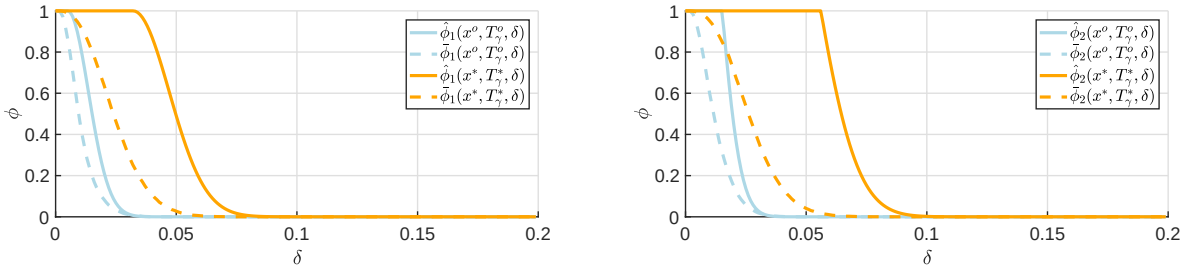


Figure 2: Upper bounds (solid lines) and sampling approximations (dashed lines) for ϕ_1 (left) and ϕ_2 (right) on x^o and x^*.

a lower and upper bound to this inferior, which can be refined using standard branch and bound techniques. Let $t = \Sigma^{-1}_{\mathcal{D},\mathcal{D}}(\mathbf{y} - \mu_{\mathcal{D}})$, then by the inference formula for Gaussian processes and Eqn (7) we have that:

$$\bar{\mu}(x) = \sum_{i=1}^{N} t_i \Sigma_{x,x_i} \geq \sum_{i=1}^{N} t_i \left(a^i + b^i \varphi_\Sigma(x, x_i)\right) \quad \forall x \in T$$

where we choose: $(a^i, b^i) = \begin{cases} (a^i_L, b^i_L), & \text{if } t_i \geq 0. \\ (a^i_U, b^i_U), & \text{otherwise.} \end{cases}$. Let $\bar{x}$ be an inferior point for $x \in T$ to the right-hand side Equation (that can be computed thanks to Assumption 3, with $c_1 := t_i b^i$), then, by the definition of inferior we have that:

$$\bar{\mu}(\bar{x}) \geq \inf_{x \in T} \bar{\mu}(x) \geq \sum_{i=1}^{N} t_i a_i + \sum_{i=1}^{N} t_i b_i \varphi_\Sigma(\bar{x}, x_i).$$

The latter provide bounds on $\inf_{x \in T} \bar{\mu}(x)$ that can be used within a branch and bound algorithm for further refinement.

Computational Complexity

Performing inference with GPs has a cost that is $\mathcal{O}(N^3)$, where N is the size of the dataset. Once inference has been performed the cost of computing upper and lower bounds for $\sup_{x \in T} \mu^{o,(i)}(x^*, x)$ is $\mathcal{O}(NC)$, where C is a constant that depends on the particular kernel. For instance, for the squared-exponential kernel $C = 1$, while for the ReLU kernel (Eqn (8)) $C = L$, where L is the number of layers of the corresponding neural network. The computation of the bounds for $\xi^{(i)}$ requires solving a convex quadratic problem in $m + N + 1$ variables, while $K^{(i)}_{x^*}$ and $\sup_{x_1,x_2 \in T} d^{(i)}_{x^*}(x_1, x_2)$ can be bounded in constant time. Refining the bounds with a branch and bound approach has a worst-case cost that is exponential in m, the dimension of the input space. Hence, sparse approximations, which mitigate the cost of performing inference with GP (Seeger, Williams, and Lawrence 2003), are appealing.

Experimental Evaluation: Robustness Analysis of Deep Neural Networks

In this section we apply the methods presented above to GP defined with deep kernels, in an effort to provide a probabilistic analysis of adversarial examples. This analysis is exact for GPs, but only approximate for fully-connected NNs, by virtue of *weak convergence* of the induced distributions between deep kernel GPs and deep fully-connected NNs.

Experimental Setting

We focus on GPs with ReLU kernel, which directly correspond to fully-connected NNs with ReLU activation functions. Given the number of layers L, the regularization parameters σ_w (prior variance on the weights) and σ_b (prior variance on the bias), the ReLU covariance $\Sigma^L(x_1, x_2)$ between two input points is iteratively defined by the set of equations (Lee et al. 2017):

$$\begin{aligned} \Sigma^l(x_1, x_2) = \sigma_b^2 + \frac{\sigma_w^2}{2\pi} \sqrt{\Sigma^{l-1}(x_1, x_1)\Sigma^{l-1}(x_2, x_2)} \\ \left(\sin \beta^{l-1}_{x_1,x_2} + (\pi - \beta^{l-1}_{x_1,x_2}) \cos \beta^{l-1}_{x_1,x_2}\right) \\ \beta^l_{x_1,x_2} = \cos^{-1}\left(\frac{\Sigma^l(x_1, x_2)}{\sqrt{\Sigma^l(x_1, x_1)\Sigma^l(x_2, x_2)}}\right) \end{aligned} \quad (8)$$

for $l = 1, \ldots, L$, where $\Sigma^0(x_1, x_2) = \sigma_b^2 + \frac{\sigma_w^2}{m} x_1 \cdot x_2$.

Training We follow the experimental setting of (Lee et al. 2017), that is, we train a selection of ReLU GPs on a subset of the MNIST dataset using least-square classification (i.e. posing a regression problem to solve the classification task) and rely on optimal hyper-parameter values estimated in the latter work. Note that the methods we presented are not constrained to specific kernels or classification models, and can be generalized by suitable modifications to the constant computation part. Classification accuracy obtained on the full MNIST test set varied between 77% (by training only on 100 samples) to 95% (training on 2000 samples). Unless otherwise stated, we perform analysis on the best model obtained using 1000 training samples, that is, a two-hidden-layer architecture with $\sigma_w^2 = 3.19$ and $\sigma_b^2 = 0.00$.

Analysis For scalability purposes we adopt the idea from (Wicker, Huang, and Kwiatkowska 2018; Ruan, Huang, and Kwiatkowska 2018) of performing a feature-level analysis. Namely, we pre-process each image using SIFT (Lowe 2004). From its output, we keep salient points and their relative magnitude, which we use to extract relevant patches from each image, in the following referred to as *features*. We apply the analysis to thus extracted features. Unless otherwise stated, feature numbering follows the descending order of magnitude.

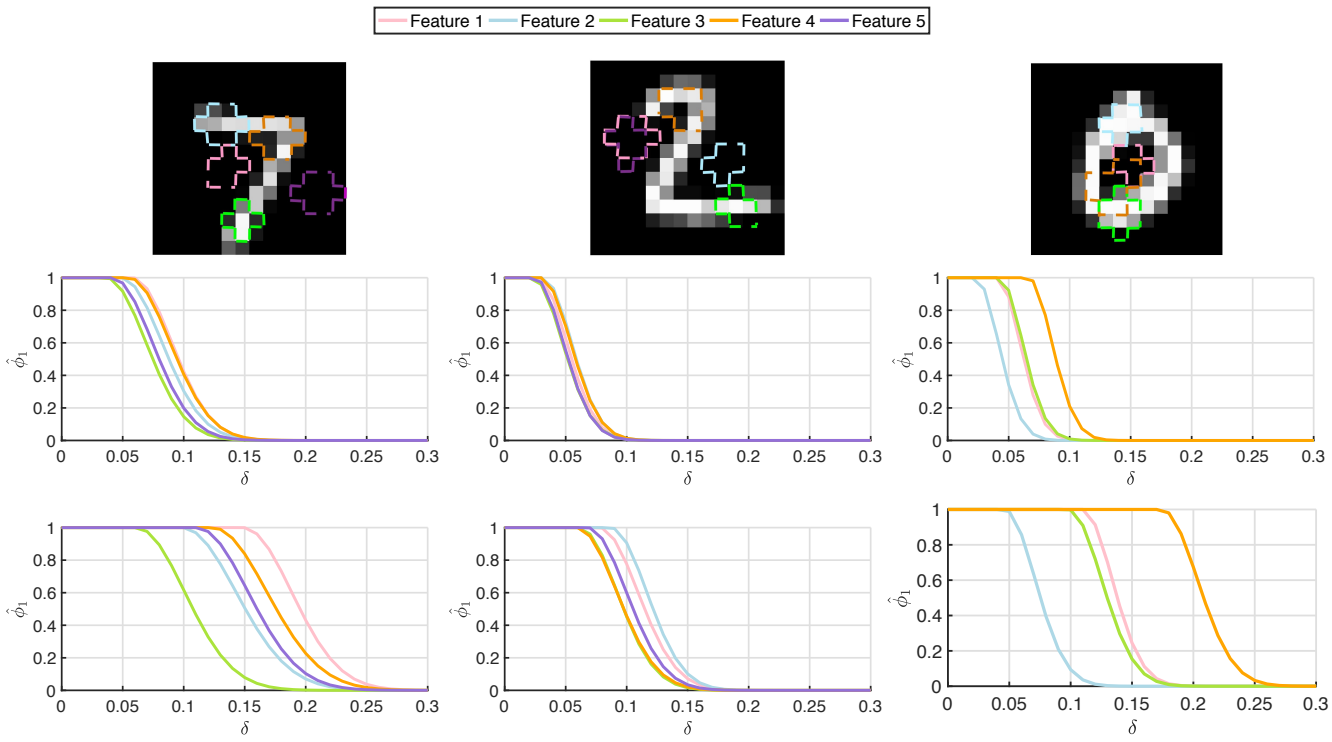


Figure 3: First row: three images randomly selected from the MNIST test set, along with detected SIFT features. Second row: respective $\hat{\phi}_1$ values for $\gamma = 0.05$. Third row: respective $\hat{\phi}_1$ values for $\gamma = 0.15$.

Feature-based Analysis

In the first row of Figure 3 we consider three images from the MNIST test data, and for each we highlight the first five features extracted by SIFT (or less if SIFT detected less than five features). For each image x_i, feature f_j and $\gamma > 0$ we consider the set of images $T_{x_i}^{\mathrm{f}_j,\gamma}$ given by the images differing from x_i in only the pixels included in f_j and by no more than γ for each pixel.

We plot the values obtained for $\hat{\phi}_1$ as a function of δ for $\gamma = 0.05$ and $\gamma = 0.15$, respectively, on the second and third row of Figure 3. Recall that $\hat{\phi}_1$ represents an upper-bound on the probability of finding $x \in T_{x_i}^{\mathrm{f}_j,\gamma}$ such that the classification confidence for the correct class in x drops by more than δ compared to that of x_i. Since a greater γ value implies a larger neighborhood $T_{x_i}^{\mathrm{f}_j,\gamma}$, intuitively $\hat{\phi}_1$ will monotonically increase along with the value of γ. Interestingly, the rate of increase is significantly different for different features. In fact, while most of the 14 features analyzed in Figure 3 have similar $\hat{\phi}_1$ values for $\gamma = 0.05$, the values computed for some of the features using $\gamma = 0.15$ are almost double (e.g. feature 4 for the third image), and remains fairly similar for others (e.g. feature 3 for the first image). Also the relative ordering in robustness for different features is not consistent for different values of γ (e.g. features 2 and 5 from the first image). This highlights the need of performing parametric analysis of adversarial attacks, which take into account different strengths and misclassification thresholds, as suggested in (Biggio and Roli 2017). Finally, notice that, though only 14 features are explored here, the experiment shows no clear relationship between feature magnitude as estimated by SIFT and feature robustness, which calls for caution in adversarial attacks and defences that rely on feature importance. Note also that an empirical analysis of the robustness based on sampling, as performed in Figure 2, becomes infeasible for this example as, in order to have good accuracy, a fine grid over a high-dimensional input space would be required.

Variance Analysis

Most active defences are based upon rejecting input samples characterized by high uncertainty values. After uncertainty is estimated, defences of this type usually proceed by setting a meta-learning problem whose goal is to distinguish between low and high variance input points, so as to flag potential adversarial examples (Grosse et al. 2017b; Feinman et al. 2017). However, mixed results are obtained with this approach (Carlini and Wagner 2017).

In this subsection we aim at analyzing how the variance around test samples changes with different training settings for the three test points previously discussed. We use the method developed for variance optimisation to compute:

$$\bar{\sigma}^2(x^*) = \frac{1}{\bar{\Sigma}_{x^*,x^*}} \sup_{x \in T_{x^*}^{\mathrm{f}_1,\gamma}} \bar{\Sigma}_{x,x},$$

that is, we look for the highest variance point in the $T_{x^*}^{\mathrm{f}_1,\gamma}$

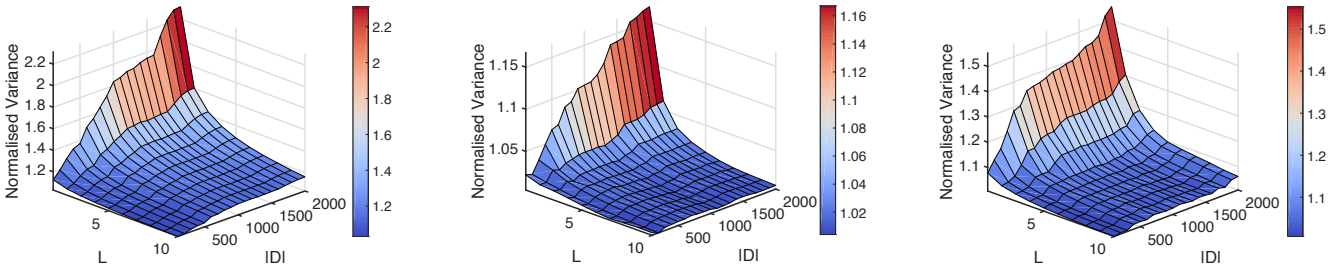


Figure 4: Normalized variance $\bar{\sigma}^2$ as a function of L (number of layers of the corresponding NN) and $|D|$ (number of training point).

neighbourhood of x^*, and normalise its value with respect to the variance at x^*. We use $\gamma = 0.15$ and perform the analysis only on feature 1 of each image.

Figure 4 plots values of $\bar{\sigma}^2(x^*)$ as a function of the number of layers (from 1 to 10) and samples (from 100 to 2000) included in the training set. Firstly, notice how maximum values of $\bar{\sigma}^2(x^*)$ are perfectly aligned with the results of Figure 3. That is, less robust features are associated with higher values of $\bar{\sigma}^2(x^*)$ (e.g. feature 1 for image 1). This highlights the relationship between the existence of adversarial examples in the neighbourhood of a point and model uncertainty. We observe the normalised variance value to consistently monotonically increase with respect to the number of training samples used. This suggests that, as more and more training samples are input into the training model, the latter become more confident in predicting "natural" test samples compared to "artificial" ones. Unfortunately, as the number of layers increases, the value of $\bar{\sigma}^2(x^*)$ decreases rapidly to a plateau. This seems to point to the fact that defence methods based on a-posteriori variance thresholding become less effective with more complex neural network architectures, which could be a justification for the mixed results obtained so far using active defences.

Conclusion

In this paper we presented a formal approach for safety analysis of Bayesian inference with Gaussian process priors with respect to adversarial examples and invariance properties. As the properties considered in this paper cannot be computed exactly for general GPs, we compute their safe over-approximations. Our bounds are based on the Borell-TIS inequality and the Dudley entropy integral, which are known to give tight bounds for the study of suprema of Gaussian processes (Adler and Taylor 2009). On examples of regression tasks for GPs and deep neural networks, we showed how our results allow one to quantify the uncertainty associated to a given prediction, also taking into account of local perturbations of the input space. Hence, we believe our results represent a step towards the application of Bayesian models in safety-critical applications.

Appendix: Kernel Function Decompositions

We provide decomposition of commonly used kernel functions that satisfy Assumptions 1,2 and 3 stated in the main text. The kernels analyzed are the following ones:

Rational Quadratic Kernel defined as:

$$\Sigma_{x_1,x_2} = \sigma^2 \left(1 + \frac{1}{2}\sum_{j=1}^{m} \theta_j \left(x_1^{(j)} - x_2^{(j)}\right)^2\right)^{-\alpha}$$

with hyper-parameters σ, α and θ_j, for $j = 1, \ldots, m$.

Linear Kernel defined as:

$$\Sigma_{x_1,x_2} = \sigma^2 \sum_{j=1}^{m} (x_1^{(j)} - \theta_j)(x_2^{(j)} - \theta_j)$$

with hyper-parameters σ and θ_j, for $j = 1, \ldots, m$.

Periodic Kernel defined as:

$$\Sigma_{x_1,x_2} = \sigma^2 \exp\left(-\frac{1}{2}\sum_{j=1}^{m} \theta_j \sin\left(p_j(x_1^{(j)} - x_2^{(j)})\right)^2\right)$$

with hyper-parameters σ, θ_j and p_j for $j = 1, \ldots, m$.

Matérn Kernel for half-integers values, defined as:

$$\Sigma_{x_1,x_2} = \sigma^2 k_p \exp\left(-\sqrt{\hat{k}_p \sum_{j=1}^{m} \theta_j (x_1^{(j)} - x_2^{(j)})}\right) \cdot \cdot \sum_{l=0}^{p} k_{l,p} \sqrt{\hat{k}_p \sum_{j=1}^{m} \theta_j (x_1^{(j)} - x_2^{(j)})}^{p-l}$$

with hyper-parameters σ, θ_j, for $j = 1, \ldots, m$, and (integer valued) p; while k_p, $\hat{k}_p$ and $k_{l,p}$ are constants.

Decompositions for the kernels listed above are provided in Table 1. Those satisfy Assumptions 1,2 and 3, ensuring that the framework can be applied for these kernels.

Specifically, for the periodic kernel Assumption 3 is not strictly satisfied as it is equivalent to the computation of:

$$\sup_{x \in T} \sum_{i=1}^{N} c_i \sum_{j=1}^{m} \theta_j \sin(p_j(x^{(j)} - x_i^{(j)}))^2.$$

Each summand separately can be trivially optimized; summing together the individual optima provides a sound over-approximation of the sup. As such, the decomposition will provide formal lower and upper bounds that can be used for branch and bound, though in general those will be looser requiring an increased number of iterations in practice.

Kernel	$\psi_\Sigma(\varphi_\Sigma)$	$\varphi_\Sigma(x_1, x_2)$
Squared Exponential	$\sigma^2 \exp\left(-\varphi_\Sigma(x_1, x_2)\right)$	$\sum_{j=1}^{m} \theta_j (x_1^{(j)} - x_2^{(j)})^2$
ReLU	$\sigma_b^2 + \frac{\sigma_w^2}{2\pi}\left(\sigma_b^2 + \frac{\sigma_w^2}{m}\right)\left(\sin\left(\cos^{-1}\varphi_\Sigma(x_1, x_2)\right) + \right.$ $\left. + \varphi_\Sigma(x_1, x_2)\left(\pi - \cos^{-1}\varphi_\Sigma(x_1, x_2)\right)\right)$	$k_1 + k_2(x_1 \cdot x_2)$
Rational Quadratic	$\sigma^2(1 + \frac{\varphi_\Sigma}{2})^{-\alpha}$	$\sum_{j=1}^{m} \theta_j (x_1^{(j)} - x_2^{(j)})^2$
Linear	$\sigma^2 \varphi_\Sigma$	$\sum_{j=1}^{m} (x_1^{(j)} - \theta_j)(x_2^{(j)} - \theta_j)$
Periodic	$\sigma^2 \exp(-0.5\varphi_\Sigma)$	$\sum_{j=1}^{m} \theta_j \sin(p_j(x_1^{(j)} - x_2^{(j)}))^2$
Matérn	$\sigma^2 k_p \exp\left(-\sqrt{\varphi_\Sigma}\right) \sum_{l=0}^{p} k_{l,p} \sqrt{\varphi_\Sigma^{p-l}}$	$\hat{k}_p \sum_{j=1}^{m} \theta_j (x_1^{(j)} - x_2^{(j)})$

Table 1: Kernels decomposition that satisfy the three assumptions stated in the main text (Section Constant Computation). Decomposition for Matérn kernel is given only for half-integer values. Decomposition for the ReLU kernel is given in the case of one-hidden layer, generalisation to an arbitrary number of layers can be obtained by recursive application of the formulas.

Appendix: Proof

Proof of Theorem 1

$$
\begin{aligned}
&P(\exists x \in T \; s.t. \; \big(\mathbf{z}^{(i)}(x^*) - \mathbf{z}^{(i)}(x) > \delta \,|\, \mathcal{D}\big) \\
&\qquad \text{(By definition of supremum)} \\
=&P\big(\sup_{x \in T} \mathbf{z}^{o,(i)}(x^*, x) > \delta\big) \\
&\qquad \text{(By linearity of GPs)} \\
=&P\big(\sup_{x \in T} \hat{\mathbf{z}}^{o,(i)}(x^*, x) + \mathbb{E}[\mathbf{z}^{o,(i)}(x^*, x)] > \delta\big) \\
&\qquad \text{(By definition of supremum)} \\
\leq&P\big(\sup_{x \in T} \hat{\mathbf{z}}^{o,(i)}(x^*, x) > \delta - sup_{x_1 \in T}\mathbb{E}[\mathbf{z}^{o,(i)}(x^*, x_1)]\big).
\end{aligned}
$$

where $\hat{\mathbf{z}}^{o,(i)}(x^*, x)$ is the zero mean Gaussian process with same variance of $\mathbf{z}^{o,(i)}(x^*, x)$. The last inequality can be bound from above using the following inequality, called Borell-TIS inequality (Adler and Taylor 2009).

Theorem 3. *(Borell-TIS inequality) Let $\hat{\mathbf{z}}$ a zero-mean unidimensional Gaussian process with covariance matrix Σ. Assume $E[sup_{x \in T}\hat{\mathbf{z}}(x)] < \infty$. Then, for any $u > \mathbb{E}[sup_{x \in T}\hat{\mathbf{z}}(x)]$ it holds that*

$$
P(sup_{x \in T}\hat{\mathbf{z}}(x) > u) \leq e^{\frac{(u - \mathbb{E}[sup_{t \in T}\hat{\mathbf{z}}(x)])^2}{2\sigma_T^2}}, \qquad (9)
$$

where $\sigma_T^2 = sup_{x \in T}\Sigma(x)$.

In order to use the Borell-TIS inequality we need to bound from above $\mathbb{E}[sup_{t \in T}\hat{\mathbf{z}}(x)]$, the expectation of the supremum of $\hat{\mathbf{z}}$. For Gaussian processes we can use the Dudley's entropy integral (Adler and Taylor 2009), which guarantees that

$$
\mathbb{E}[\sup_{x \in T} \hat{\mathbf{z}}(x)] \leq 12 \int_0^{sup_{x_1, x_2 \in T} d(x_1, x_2)} \sqrt{ln(N(d, x, T))} dx,
$$

where $N(d, x, T)$ is the smallest number of balls of radius x according to metric d that completely cover T (see (Adler and Taylor 2009) for further details). For a hyper-cube T of dimension D, in order to compute $N(d, x, T)$, we first need to compute $N(L_2, r, T)$, the number of covering balls of diameter r of T under L_2 norm. As the largest hyper-cube contained inside a $m-$sphere of diameter r has a side of length $\frac{r}{\sqrt{m}}$, we obtain

$$
N(L_2, r, T) \leq \big(1 + \frac{D\sqrt{m}}{r}\big)^m.
$$

Now we know that for $x^* \in T$

$$
sup_{x_1, x_2 \in T} d_{x^*}^{(i)}(x_2, x_1) \leq K_{\hat{x}}^{(i)} ||x_2 - x_1||_2,
$$

Thus, this implies that all the points inside a ball of radius $r = \frac{x}{K_{\hat{x}}^{(i)}}$ will have a distance in the d metric smaller or equal than x. Thus, the number of covering balls of radius x for T, according to pseudo-metric d is upper-bounded by

$$
N(d, x, T) \leq \big(\frac{\sqrt{m} D K_{\hat{x}}^{(i)}}{x} + 1\big)^m.
$$

References

Adler, R. J., and Taylor, J. E. 2009. *Random fields and geometry*. Springer Science & Business Media.

Bach, F. R. 2009. Exploring large feature spaces with hierarchical multiple kernel learning. In *Advances in neural information processing systems*, 105–112.

Bartocci, E.; Bortolussi, L.; Nenzi, L.; and Sanguinetti, G. 2015. System design of stochastic models using robustness of temporal properties. *Theoretical Computer Science* 587:3–25.

Biggio, B., and Roli, F. 2017. Wild patterns: Ten years after the rise of adversarial machine learning. *arXiv preprint arXiv:1712.03141*.

Bortolussi, L.; Cardelli, L.; Kwiatkowska, M.; and Laurenti, L. 2018. Central limit model checking. *arXiv preprint arXiv:1804.08744*.

Cardelli, L.; Kwiatkowska, M.; Laurenti, L.; and Patane, A. 2018. Robustness guarantees for bayesian inference with gaussian processes. *arXiv preprint arXiv:1809.06452*.

Carlini, N., and Wagner, D. 2017. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 3–14. ACM.

Chouza, M.; Roberts, S.; and Zohren, S. 2018. Gradient descent in Gaussian random fields as a toy model for high-dimensional optimisation in deep learning. *arXiv preprint arXiv:1803.09119*.

Dreossi, T.; Donzé, A.; and Seshia, S. A. 2017. Compositional falsification of cyber-physical systems with machine learning components. In *NASA Formal Methods Symposium*, 357–372. Springer.

Dudley, R. M. 1967. The sizes of compact subsets of Hilbert space and continuity of Gaussian processes. *Journal of Functional Analysis* 1(3):290–330.

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

Gal, Y., and Ghahramani, Z. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, 1050–1059.

Grosse, K.; Manoharan, P.; Papernot, N.; Backes, M.; and McDaniel, P. 2017a. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*.

Grosse, K.; Pfaff, D.; Smith, M. T.; and Backes, M. 2017b. How wrong am I? - Studying adversarial examples and their impact on uncertainty in Gaussian process machine learning models. *arXiv preprint arXiv:1711.06598*.

Hein, M., and Andriushchenko, M. 2017. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems*, 2263–2273.

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

Jones, D. R.; Schonlau, M.; and Welch, W. J. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global optimization* 13(4):455–492.

Laurenti, L.; Abate, A.; Bortolussi, L.; Cardelli, L.; Ceska, M.; and Kwiatkowska, M. 2017. Reachability computation for switching diffusions: Finite abstractions with certifiable and tuneable precision. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control*, 55–64. ACM.

Lee, J.; Bahri, Y.; Novak, R.; Schoenholz, S. S.; Pennington, J.; and Sohl-Dickstein, J. 2017. Deep neural networks as Gaussian processes. *arXiv preprint arXiv:1711.00165*.

Li, Y., and Gal, Y. 2017. Dropout inference in Bayesian neural networks with alpha-divergences. *arXiv preprint arXiv:1703.02914*.

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

Matthews, A. G. d. G.; Rowland, M.; Hron, J.; Turner, R. E.; and Ghahramani, Z. 2018. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*.

Neal, R. M. 2012. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.

Raghunathan, A.; Steinhardt, J.; and Liang, P. 2018. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*.

Rasmussen, C. E. 2004. Gaussian processes in machine learning. In *Advanced lectures on machine learning*. Springer. 63–71.

Ruan, W.; Huang, X.; and Kwiatkowska, M. 2018. Reachability analysis of deep neural networks with provable guarantees. *arXiv preprint arXiv:1805.02242*.

Sadigh, D., and Kapoor, A. 2015. Safe control under uncertainty. *arXiv preprint arXiv:1510.07313*.

Seeger, M.; Williams, C.; and Lawrence, N. 2003. Fast forward selection to speed up sparse Gaussian process regression. In *Artificial Intelligence and Statistics 9*.

Seshia, S. A.; Sadigh, D.; and Sastry, S. S. 2016. Towards verified artificial intelligence. *arXiv preprint arXiv:1606.08514*.

Srinivas, N.; Krause, A.; Kakade, S. M.; and Seeger, M. W. 2012. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory* 58(5):3250–3265.

Sui, Y.; Gotovos, A.; Burdick, J.; and Krause, A. 2015. Safe exploration for optimization with Gaussian processes. In *International Conference on Machine Learning*, 997–1005.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Wachi, A.; Sui, Y.; Yue, Y.; and Ono, M. 2018. Safe exploration and optimization of constrained MDPs using Gaussian processes. In *AAAI Conference on Artificial Intelligence*.

Wicker, M.; Huang, X.; and Kwiatkowska, M. 2018. Feature-guided black-box safety testing of deep neural networks. In *International Conferfence on Tools and Algorithms for the Construction and Analysis of Systems*, 408–426. Springer.