On the Robustness of Bayesian Neural Networks to Adversarial Attacks

Bortolussi, Luca; Carbone, Ginevra; Laurenti, Luca; Patane, Andrea; Sanguinetti, Guido; Wicker, Matthew

DOI
10.1109/TNNLS.2024.3386642

Publication date
2025

Document Version
Final published version

Published in
IEEE Transactions on Neural Networks and Learning Systems

Citation (APA)
Bortolussi, L., Carbone, G., Laurenti, L., Patane, A., Sanguinetti, G., & Wicker, M. (2025). On the Robustness of Bayesian Neural Networks to Adversarial Attacks. *IEEE Transactions on Neural Networks and Learning Systems*, *36*(4), 6679-6692. https://doi.org/10.1109/TNNLS.2024.3386642

On the Robustness of Bayesian Neural Networks to Adversarial Attacks

Luca Bortolussi, Ginevra Carbone, Luca Laurenti, Andrea Patane, Guido Sanguinetti, and Matthew Wicker

***Abstract*— Vulnerability to adversarial attacks is one of the principal hurdles to the adoption of deep learning in safety-critical applications. Despite significant efforts, both practical and theoretical, training deep learning models robust to adversarial attacks is still an open problem. In this article, we analyse the geometry of adversarial attacks in the over-parameterized limit for Bayesian neural networks (BNNs). We show that, in the limit, vulnerability to gradient-based attacks arises as a result of degeneracy in the data distribution, i.e., when the data lie on a lower dimensional submanifold of the ambient space. As a direct consequence, we demonstrate that in this limit, BNN posteriors are robust to gradient-based adversarial attacks. Crucially, by relying on the convergence of infinitely-wide BNNs to Gaussian processes (GPs), we prove that, under certain relatively mild assumptions, the expected gradient of the loss with respect to the BNN posterior distribution is vanishing, even when each NN sampled from the BNN posterior does not have vanishing gradients. The experimental results on the MNIST, Fashion MNIST, and a synthetic dataset with BNNs trained with Hamiltonian Monte Carlo and variational inference support this line of arguments, empirically showing that BNNs can display both high accuracy on clean data and robustness to both gradient-based and gradient-free adversarial attacks.**

***Index Terms*— Adversarial attacks, adversarial robustness, Bayesian inference, Bayesian neural networks (BNNs).**

I. Introduction

ADVERSARIAL attacks are small, potentially imperceptible, perturbations of test inputs that can lead to catastrophic misclassifications in high-dimensional classifiers such as deep neural networks (NNs). Since the seminal work of Szegedy et al. [1], adversarial attacks have been intensively studied and even highly accurate state-of-the-art deep learning models, trained on very large datasets, have been shown to be susceptible to such attacks [2], [3]. In the absence of effective defenses, the widespread existence of adversarial examples has raised serious concerns about the security and robustness of models learned from data [4], [5]. As a consequence, the development of machine learning models that are robust to adversarial perturbations is an essential precondition for their application in safety-critical scenarios, such as autonomous driving, where model failures can lead to fatal or costly accidents.

Many adversarial attack strategies are based on identifying directions of high variability in the loss function by evaluating the gradient with respect to to the NN input [2], [6]. Since such variability can be intuitively linked to uncertainty in the prediction, Bayesian neural networks (BNNs) [7], [8], [9], [10] have been recently suggested as a more robust deep learning paradigm, a claim that has also found empirical support [11], [12], [13], [14], [15]. However, neither the source of this robustness, nor its general applicability are well-understood mathematically.

In this article, we show a remarkable property of BNNs: in a suitably defined limit, we prove that the gradients of the expected loss function of an infinitely-wide BNN with respect to the input vanish. Our analysis shows that adversarial attacks for highly accurate NNs arise from the low-dimensional support of the data generating distribution. By averaging over nuisance dimensions, under certain assumptions on the geometry of the space where the data come from, assumed to be a manifold, BNNs achieve zero expected gradient of the loss and are, thus, provably immune to gradient-based adversarial attacks. Specifically, we first show that, for any NN achieving zero loss, adversarial attacks arise in directions orthogonal to the data manifold. Then, we rely on the submanifold extension lemma [16] to show that in the limit of infinitely-wide layers, for any NN and any weights set, there exists another weights set (of the same NN architecture) achieving the same loss and with opposite loss gradients orthogonal to the data manifold on a given point. Finally, by relying on the convergence of BNNs to Gaussian processes (GPs) [7] and under the assumption that the data manifold is a subspace, we show that for infinitely-wide BNNs, the expectation of the gradient with respect to the posterior distribution in a direction orthogonal to the data manifold vanishes. Crucially, our results guarantees that, in the limit, BNNs' posteriors are provably robust to gradient-based adversarial attacks even when NNs sampled from the posterior are vulnerable to such attacks.

We experimentally support our theoretical findings on various BNN architectures trained with Hamiltonian Monte Carlo (HMC) and with variational inference (VI), specifically Bayes by Backprop [17], on MNIST, Fashion MNIST, and the half moons datasets, empirically showing that the magnitude of the

Manuscript received 5 July 2023; revised 28 November 2023 and 6 February 2024; accepted 30 March 2024. Date of publication 22 April 2024; date of current version 7 April 2025. *(Corresponding author: Luca Laurenti.)*

Luca Bortolussi is with the Department of Mathematics, Informatics and Geosciences, University of Trieste, Trieste, Italy.

Ginevra Carbone is with the Department of Mathematics and Geosciences, University of Trieste, 34128 Trieste, Italy.

Luca Laurenti is with Delft Center for Systems and Control, TU Delft University, 2628 CN Delft, The Netherlands (e-mail: l.laurenti@tudelft.nl).

Andrea Patane is with the School of Computer Science and Statistics, Trinity College, Dublin D02 PN40, Ireland.

Guido Sanguinetti is with SISSA, International School for Advanced Studies, 34136 Trieste, Italy, and also with the School of Informatics, University of Edinburgh, EH8 9AB Edinburgh, U.K.

Matthew Wicker is with the Department of Computer Science, University of Oxford, OX1 3QG Oxford, U.K.

Digital Object Identifier 10.1109/TNNLS.2024.3386642

gradients decreases as more samples are taken from the BNN posterior. We then explore the robustness of BNNs to adversarial attacks experimentally in these settings. In particular, we conduct a large-scale experiment on thousands of different NNs, empirically finding that, in the cases here analyzed, for BNNs higher accuracy tend to correlate with higher robustness to gradient-based adversarial attacks, contrary to what observed for deterministic NNs (DNNs) trained via standard stochastic gradient descent (SGD). Finally, we also investigate the robustness of BNNs to gradient-free adversarial attacks, empirically showing that BNNs are substantially more robust than their deterministic counterpart even in this setting.

In summary, this article makes the following contributions.

1) A proof that, in the infinitely-wide layers and large data limit setting, the gradient of the loss function with respect to the input only preserves the component, which is orthogonal to the data manifold (Section III) and that for any weights set of an NN, there exists another weight set with the same loss and opposite orthogonal gradients (Section III-A).
2) A proof that for GPs, and consequently for infinitely-wide-limit BNNs, the expected posterior gradient of the loss vanishes when projected in a direction orthogonal to the data manifold, thus providing robustness to BNNs (Section IV).
3) Experiments showing empirically that BNNs are more robust to both gradient-based and gradient-free attacks than their deterministic counterpart and can resist the well known accuracy–robustness tradeoff (Section VI).[1]

A preliminary version of this work appeared in [18]. This work extends [18] in several aspects. Carbone et al. [18] proved that given an infinitely-wide NN with zero loss and a nonzero orthogonal gradient to the data manifold, there exists another zero-loss NN with opposite orthogonal gradient and use this result to conjecture that under certain conditions, BNNs may achieve zero gradients in expectation. In this article, in Section IV, this conjecture is shown to be true and proved explicitly by relying on the convergence of BNNs and GPs. Furthermore, we substantially extend the discussion and the theoretical analysis, and improve the empirical results with gradient-free adversarial attacks and a comparison between the robustness of GPs and BNNs.

This article is structured as follows. In Section II, we introduce background on infinitely-wide NNs and BNNs. In Section III, we will first show that for highly accurate NNs, the gradient of the loss is nonzero only in directions orthogonal to the data manifold. Then, in Section III-A, we will prove that for any NN and weight set, there exists another weight set of the same NN with the same loss and opposite orthogonal gradients to the data manifold. By averaging over these weight sets and relying on the convergence of BNNs to GPs, in Section IV, we prove that for a BNN that achieves zero loss on the data manifold, the expected gradient of the loss is zero, thus making them robust to adversarial attacks. Section V discusses consequences and limitations of our results. Empirical results in Section VI will support our theoretical findings.

A. Related Work

Adversarial attacks for DNNs have been the subject of extensive analyses [19], [20], [21], [22], [23], [24], [25], which have led to the development of multiple defense and attack methods over the recent years [4], [19], [26]. Adversarial examples have been found to be so widespread in the state-of-the-art DNNs that they have even been hypothesized to be an intrinsic property of certain models [24] or datasets [27]. Interestingly, early experimental results with BNNs suggested a diametrically different behavior than that of their deterministic counterpart. In fact, empirical observations on the increased adversarial robustness of BNNs have been made in various works both against gradient-based adversarial attacks [28], [29], [30] and gradient-free adversarial attacks [15] as well as on reinforcement learning settings [31] and more recently also on relatively large convolutional NN architectures [28]. However, while these works present empirical evidences on the robustness of BNNs, they do not give any theoretical justification on the mechanisms that lead to BNN robustness. First attempts to understand the robustness properties of BNNs have been considered in [13] and [32]. In particular, Bekasov and Murray [13] defined Bayesian adversarial spheres and empirically showed that, for BNNs trained with HMC, adversarial examples tend to have high uncertainty. Instead, Gal and Smith [32] derived sufficient conditions for idealized BNNs to avoid adversarial examples. However, it is unclear how such conditions could be checked in practice, as it would require one to check that the BNN architecture is invariant under all the symmetries of the data.

Because of the capabilities of BNNs to model epistemic uncertainty, which can be intuitively linked to their robustness properties, various approaches have been proposed to detect adversarial examples for BNNs. Feinman et al. [11] and Rawat et al. [33] propose to use the uncertainty on the predictions of a BNN as a way to flag adversarial attacks. However, such methods have been shown to be easily fooled by appropriately crafted adversarial attacks [34], [35]. Consequently, formal verification methods [36], [37] to detect adversarial examples for BNNs have been introduced. These methods have been followed by techniques to perform adversarial training for BNNs [12], [14], [38], [39], where additional robustness constraints or penalties are considered directly at training time. Interestingly, empirical results obtained with such techniques, highlighted how, in the Bayesian settings, high accuracy and high robustness often are positively correlated with each other. The theoretical framework we develop in this article further confirms and grounds these findings.

II. Background

Let $f^{\text{true}} : \mathcal{M} \to \mathbb{R}$ be a function defined on a data manifold $\mathcal{M} \subseteq X \subseteq \mathbb{R}^d$ with X being the ambient (or embedding) space.[2] We consider the problem of approximating f^{true} via the learning of an $M + 1$ layers NN $f(\cdot, \mathbf{w})$, with $\mathbf{w} \in \mathbb{R}^{n_{\mathbf{w}}}$ being the aggregate vector of weights and biases. Formally,

[1] The code for the experiments can be found at https://github.com/matthewwicker/OnTheRobustnessOfBNNs.

[2] For simplicity of presentation, we assume a scalar output. The results of this article naturally extend to the multioutput case by treating each output component similar to the single output case.

for $\mathbf{x} = (x_1, \dots, x_d) \in X$, $f(\mathbf{x}, \mathbf{w})$ is defined iteratively over the number of layers as follows:

$$f_i^{(1)}(\mathbf{x}) = \sum_{j=1}^{d} w_{ij}^{(1)} x_j + b_i^{(1)} \tag{1}$$

$$f_i^{(m)}(\mathbf{x}) = \sum_{j=1}^{n_{m-1}} w_{ij}^{(m)} \phi\left(f_j^{(m-1)}(\mathbf{x})\right) + b_i^{(m)} \tag{2}$$

$$f(\mathbf{x}, \mathbf{w}) = f^{(M+1)}(\mathbf{x}) \tag{3}$$

for $m = 2, \dots, M+1$, where n_m is the number of neurons in the mth layer and ϕ is the activation function—which we assume to be continuous and with bounded derivatives. In order to learn the weights of $f(\mathbf{x}, \mathbf{w})$, one considers a dataset D_N composed of N points, $D_N = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathcal{M},\ y_i \in \mathbb{R},\ i = 1, \dots, N\}$. In a frequentist fashion, one can then use the dataset to quantify the distance between f^{true} and $f(\cdot, \mathbf{w})$ by evaluating it on a loss function $L(\mathbf{x}, \mathbf{w})$ of the form $L(\mathbf{x}, \mathbf{w}) = \ell(f(\mathbf{x}, \mathbf{w}), f^{\text{true}}(\mathbf{x}))$, with $\ell(\cdot, \cdot)$ chosen accordingly to the semantic of the problem at hand (e.g., square loss or cross-entropy).[3] Intuitively, minimization of the loss function over the weight vector $\mathbf{w}$ leads to increasing fit of $f(\mathbf{x}, \mathbf{w})$ to $f^{\text{true}}(\mathbf{x})$, with zero loss indicating that the fit is exact on D_N.

In this article, we aim at analyzing the adversarial robustness of $f(\mathbf{x}, \mathbf{w})$. In order to do so, we will rely on crucial results from Bayesian learning and on the properties of infinitely-wide NNs. The remainder of this section is dedicated to the review of such notions.

A. Infinitely-Wide NNs

In our analysis, we will rely on the notion of infinitely-wide NNs, i.e., NNs with an infinite number of neurons.

Definition 1 (Infinitely-Wide NN): Consider a family of NNs $\{f(\mathbf{x}, \mathbf{w}_{n_{\mathbf{w}}})\}_{n_{\mathbf{w}}>0}$ of (1)–(3), with a fixed number of neurons for $m = 1, \dots, M-1$ and a variable number of neurons n_M in the last hidden layer. We say that

$$f^{\infty}(\mathbf{x}) := \lim_{n_M \to \infty} f(\mathbf{x}, \mathbf{w}_{n_{\mathbf{w}}}) \quad \forall \mathbf{x} \in X \tag{4}$$

is an infinitely-wide NN if the limit above exists and if the resulting function defines a mapping from X to $\mathbb{R}$. Furthermore, we call $\mathcal{F}$ the set of such limit functions.

The interest behind the set of infinitely-wide NNs lies in the fact that they are universal approximators [40], [41].[4] More precisely, under the assumption that the true function f^{true} is continuous, we have that

$$\forall \epsilon > 0, \quad \exists f^* \in \mathcal{F}, \quad \text{s.t. } \forall x \in \mathcal{M},\ f^{\text{true}}(x) - f^*(x)| < \epsilon. \tag{5}$$

That is, $\mathcal{F}$ is dense in the space of continuous functions. Furthermore, it is possible to show that any smooth function with bounded derivatives can be represented exactly by an infinitely-wide NN with bounded weights norm (i.e., with bounded sum of the squared Euclidean norm of the weights in the network) [43]. We will rely on these crucial properties of infinitely-wide NNs to reason about their behavior against adversarial attacks. We remark that the existence of such an NN f^* approximating the true underlying function does not necessarily mean that it will be automatically found through gradient descent-based training on a finite dataset. However, recent results [44] have shown that, under mild conditions, the loss function, as a functional over the distribution over weights of an infinitely-wide NN, is a convex functional, and thus, the gradient flow of SGD will converge to a unique distribution over weights. That is, given an infinitely-wide NN f^{∞} and a sequence of datasets $\{D_N\}_{N>0}$ of cardinality N extracted from the data manifold $\mathcal{M}$, we have that

$$\lim_{N \to \infty} \ell\left(f_{D_N}^{\infty}(\mathbf{x}), f^{\text{true}}(\mathbf{x})\right) = 0 \quad \forall \mathbf{x} \in \mathcal{M} \tag{6}$$

where $f_{D_N}^{\infty}$ represents the infinitely-wide NN trained on D_N until convergence. In Section III, we will show how infinitely-wide DNNs, i.e., NNs where each weight or bias is a scalar, can be vulnerable to adversarial attacks even when the loss is zero, while infinitely-wide BNNs, under certain assumptions on the geometry of the data manifold, are provably robust to gradient-based adversarial attacks.

B. Bayesian Neural Networks

Bayesian modeling aims to capture the uncertainty of data driven models by defining ensembles of predictors [45]; it does so by turning model parameters into random variables. In the NN scenario, one starts by putting a prior measure over the network weights $p(\mathbf{w})$ [7].[5] The fit of the network with weights $\mathbf{w}$ to the data D is assessed through the likelihood $p(D|\mathbf{w})$ [46].[6] Bayesian inference then combines likelihood and prior via the Bayes theorem to obtain a posterior distribution over the NN parameters

$$p(\mathbf{w}|D) \propto p(D|\mathbf{w}) p(\mathbf{w}). \tag{7}$$

Unfortunately, it is in general infeasible to compute the posterior distribution exactly for nonlinear/nonconjugate models such as deep NNs, so that approximate Bayesian inference methods are employed in practice. Asymptotically exact samples from the posterior distribution can be obtained via procedures such as HMC [47], while approximate samples can be obtained more cheaply via VI [17]. Irrespective of the posterior inference method of choice, Bayesian empirical predictions at a new input $\mathbf{x}$ are obtained from an ensemble of n NNs, each with its individual weights drawn from the posterior distribution $p(\mathbf{w}|D)$

$$\langle f(\mathbf{x}, \mathbf{w}) \rangle_{p(\mathbf{w}|D)} \simeq \frac{1}{n} \sum_{i=1}^{n} f(\mathbf{x}, \mathbf{w}_i) \tag{8}$$

[3] For simplicity of notation, we omit the explicit dependence on the true function from the loss.

[4] Notice that the limit in Definition 1 is taken only with respect to the last hidden layer. Similar results, albeit with additional care needed for the definition of the limiting sequence, can be obtained by taking the limit with respect to all the hidden layers [42].

[5] In the remainder of this article, we employ the common notation of indicating density functions with p and their corresponding probability measures with P.

[6] Notice that in the Bayesian setting, the likelihood is a transformation of the loss function used in deterministic settings. In the rest of this article, we use both terminologies, and the loss is not to be confused with that used in Bayesian decision theory [46].

where $\mathbf{w}_i \sim p(\mathbf{w}|D)$ and $\langle\cdot\rangle_{p(\mathbf{w}|D)}$ denotes the expectation with respect to the posterior distribution $p(\mathbf{w}|D)$.

Note that the definition of a distribution over the weights $p(\mathbf{w})$ naturally leads to the definition of a probability measure over the set of continuous functions $f : \mathbb{R}^d \to \mathbb{R}$ that can be represented by the NN. In particular, as common in [48] and [49], we consider the probability measure P generated by the finite-dimensional distributions of the BNN, i.e., the joint distributions of $f(\mathbf{x}_1, \mathbf{w}), \dots, f(\mathbf{x}_k, \mathbf{w})$, where k is an arbitrary integer and $\mathbf{x}_1, \dots, \mathbf{x}_k \in X$. We stress that in general not all path properties, such as continuity or differentiability, can be determined using the finite-dimensional distributions. However, as the NNs architectures considered in this article are continuous by assumption, without any lost of generality and as common in [48], we assume that $f(\cdot, \mathbf{w})$ is separable, i.e., countable dense subsets of input points suffice to determine the properties of $f(\cdot, \mathbf{w})$.

In this article, as also common in [17], we will consider Gaussian priors $p(\mathbf{w})$. The following result shows how an independent Gaussian prior over the parameters of an infinitely-wide BNN induces a Gaussian prior over the space of functions.

Proposition 1 [7], [42]: Consider the following NN $f(\mathbf{x}, \mathbf{w})$ with a single hidden layer defined as:

$$f_i^{(1)}(\mathbf{x}) = \sum_{j=1}^{d} w_{ij}^{(1)} x_j + b_i^{(1)} \tag{9}$$

$$f_i^{(2)}(\mathbf{x}) = \sum_{j=1}^{n_1} w_{ij}^{(2)} \phi(f_j^{(1)}(\mathbf{x})) + b_i^{(2)} \tag{10}$$

$$f(\mathbf{x}, \mathbf{w}) = f^{(2)}(\mathbf{x}). \tag{11}$$

Assume that to each weight and bias are associated independent normal priors such that $w_{ij}^{(1)} \sim \mathcal{N}(0, (\sigma_w^2/d))$, $w_{ij}^{(2)} \sim \mathcal{N}(0, (\sigma_w^2/n_1))$, $b_i^{(1)}, b_i^{(2)} \sim \mathcal{N}(0, \sigma_b^2)$. Then, for $n_1 \to \infty$, the prior on $f_i(\mathbf{x}, \mathbf{w})$ converges in distribution to a GP with zero mean and covariance function $K(\mathbf{x}, \mathbf{x}') = \sigma_b^2 + \sigma_w^2 C(\mathbf{x}, \mathbf{x}')$, where $C(\mathbf{x}, \mathbf{x}')$ is a function-dependent on the BNN architecture.

Note that, while Proposition 1 is stated only for BNNs with one hidden layer, analogous results can be derived for the multiple-hidden-layer case and also for more complex architectures such as convolutional NNs [42], [50], [51]. We stress that in the case of multiple hidden layers, care must be taken in how the size of the various layers goes to infinity to guarantee convergence. In what follows, we will simply assume that for an infinitely-wide BNN, the conditions for convergence are always satisfied.

Thanks to Proposition 1, we have that an infinitely-wide BNN is equivalent to a GP. This allows us to use the favorable analytical properties of GPs to study BNN robustness in the limit. In particular, in Section IV, we will rely on the fact that the derivative[7] of a GP is still a GP with a kernel given by the derivative of the kernel of the original GP [48]. This is a key result that we will use to show how the orthogonal gradient of the loss for trained BNNs vanishes along the data manifold.

C. Adversarial Attacks for BNNs

Given an input point $\mathbf{x} \in \mathcal{M}$ and a strength (i.e., maximum perturbation magnitude) $\epsilon > 0$, the worst case adversarial perturbation can be defined as the point $\tilde{\mathbf{x}}$ in the ϵ-neighborhood around $\mathbf{x}$ that maximizes the loss function L

$$\tilde{\mathbf{x}} := \underset{\tilde{\mathbf{x}}:||\tilde{\mathbf{x}}-\mathbf{x}||\le\epsilon}{\arg\max} \langle L(\tilde{\mathbf{x}}, \mathbf{w})\rangle_{p(\mathbf{w}|D)}.$$

If the network prediction on $\tilde{\mathbf{x}}$ differs from the original prediction on $\mathbf{x}$, then we call $\tilde{\mathbf{x}}$ an *adversarial example*. As $f(\mathbf{x}, \mathbf{w})$ is nonconvex, computing $\tilde{\mathbf{x}}$ is a nonconvex optimization problem for which several approximate solution methods have been proposed. In this article, we will primarily focus on what arguably is the most commonly employed class among them, i.e., gradient-based attacks, that is attacks that employ the loss function gradient with respect to $\mathbf{x}$ to maximize the loss [4]. One such attacks is the fast gradient sign method (FGSM) [2] which works by approximating $\tilde{\mathbf{x}}$ by taking an ϵ-step in the direction of the sign of the gradient at $\mathbf{x}$. In the context of BNNs, where attacks are against the posterior distribution, applying FGSM yields

$$\tilde{\mathbf{x}} = \mathbf{x} + \epsilon \,\text{sgn}\, \langle \nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{w})\rangle_{p(\mathbf{w}|D)} \tag{12}$$

$$\simeq \mathbf{x} + \epsilon \,\text{sgn}\left(\sum_{i=1}^{n} \nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{w}_i)\right) \tag{13}$$

where the final expression is a Monte Carlo approximation with n samples $\mathbf{w}_i$ drawn from the posterior $p(\mathbf{w}|D)$. Other gradient-based attacks, as for example, projected gradient descent (PGD) method [6], modify FGSM by taking consecutive gradient iterations or by scaling the attack by the magnitude of the gradient. Crucially, however, they all rely on the gradient vector to guide the attack.

In what follows, we will rely on the fact that the expected loss gradient in (12) can be decomposed into its projection into a direction parallel to the data manifold and into a direction orthogonal to the data manifold. The parallel expected loss gradient naturally vanishes for very accurate NNs as the loss will tend to be zero everywhere in the data manifold. Instead, the orthogonal projection will in general not be zero. However, perhaps surprisingly, we will show that for infinitely-wide BNN also, the orthogonal expected loss gradient vanishes.

Before considering results specific to BNNs in Section IV, in Section III, we will focus on results that hold for both deterministic and BNNs (Lemma 1) and that are specific to DNNs (Proposition 2), showing how these can be vulnerable to adversarial attacks even when they learn the true function perfectly.

III. Gradient-Based Adversarial Attacks for NNs

Equation (12) suggests a possible mechanism through which BNNs might acquire robustness against adversarial attacks: averaging under the posterior might lead to cancellations in

[7] In this article, we will always focus on mean-square derivatives [48] for probabilistic models. Hence, in what follows, we will simply refer to them as derivatives.

the expectation of the gradient of the loss. It turns out that this averaging property is intimately related to the geometry of the data manifold $\mathcal{M}$. As a consequence, in order to study the expectation of the gradient of the loss for BNNs, we first introduce results that link the geometry of $\mathcal{M}$ to adversarial attacks.

We start with a trivial, yet important result, which holds for any NN, both Bayesian and deterministic: for an NN that achieves zero loss on the whole data manifold $\mathcal{M}$, the loss gradient is constant (and zero) along the data manifold for any $\mathbf{x} \in \mathcal{M}$. Therefore, in order to have adversarial examples, the dimension of the data manifold $\mathcal{M}$ must necessarily be smaller than the dimension of the ambient space, that is, $\dim(\mathcal{M}) < \dim(X) = d$, where $\dim(\mathcal{M})$ denotes the dimension of $\mathcal{M}$.

Lemma 1: Assume that $\mathcal{M}$ is a smooth closed manifold and that $\forall \mathbf{x} \in \mathcal{M}\ L(\mathbf{x}, \mathbf{w}) = 0$, that is, $f(\mathbf{x}, \mathbf{w})$ achieves zero loss on $\mathcal{M}$. Then, if f is vulnerable to gradient-based attacks at $\mathbf{x}^* \in \mathcal{M}$, $\dim(\mathcal{M}) < \dim(X)$ in a neighborhood of $\mathbf{x}^*$, i.e., $\mathcal{M}$ is locally homeomorphic to a space of dimension smaller than the ambient space X.

Proof: By assumption $\forall \mathbf{x} \in \mathcal{M}, L(\mathbf{x}, \mathbf{w}) = 0$, which implies that the gradient of the loss is zero along the data manifold. However, if f is vulnerable to gradient-based attacks at $\mathbf{x}^*$, then the gradient of the loss at $\mathbf{x}^*$ must be nonzero. Hence, there exists an open neighborhood $\mathcal{B}$ of $\mathbf{x}^*$ such that $\mathcal{B} \not\subseteq \mathcal{M}$, which implies $\dim(\mathcal{M}) < \dim(X)$ locally around $\mathbf{x}^*$. □

Lemma 1 confirms the widely held conjecture that adversarial attacks may originate from degeneracies of the data manifold [2], [52]. In fact, it has been already empirically noticed [53] that adversarial perturbations often arise in directions normal to the data manifold. The suggestion that lower dimensional data structures might be ubiquitous in NN problems is also corroborated by recent results [54], showing that the characteristic training dynamics of NNs are intimately linked to data lying on a lower dimensional manifold. Notice that the implication is only one way; it is perfectly possible for the data manifold to be low dimensional and still not vulnerable at many points. Consequently, the fact that the data manifold has a smaller dimension than the ambient space is a necessary, but not sufficient, condition for vulnerability to adversarial attacks.

We note that, as discussed in Section II-A, at convergence of the training algorithm and in the limit of infinitely-many data, infinitely-wide NNs are guaranteed to achieve zero loss on the data manifold, satisfying the assumption of Lemma 1. As a result, once an infinitely-wide NN is fully trained, for any $\mathbf{x} \in \mathcal{M}$, the gradient of the loss function is orthogonal to the data manifold as it is zero along the data manifold, i.e., $\nabla_{\mathbf{x}} L(\mathbf{x}, \mathbf{w}) = \nabla_{\mathbf{x}}^{\perp} L(\mathbf{x}, \mathbf{w})$, where $\nabla_{\mathbf{x}}^{\perp}$ denotes the gradient projected into the normal subspace of $\mathcal{M}$ at $\mathbf{x}$. We stress that for a given NN, $\nabla_{\mathbf{x}}^{\perp} L(\mathbf{x}, \mathbf{w})$ is in general nonzero even if the network achieves zero loss on $\mathcal{M}$ (this is formalized in Section III-A), thus explaining the existence of adversarial examples even for very accurate classifiers. Crucially, in Section IV, we show that for BNNs, when averaged with respect to the posterior distribution, the orthogonal gradient vanishes.

A. Symmetry Property of NNs

Before considering the BNN case, in Proposition 2, we show a symmetry property of NNs: given an NN, we can always find an infinitely-wide NN that has the same loss but opposite orthogonal gradient. In order to prove this result, we first introduce Lemma 2, which is a generalization of the submanifold extension lemma and a key result we leverage. It proves that any smooth function defined on a submanifold $\mathcal{M}$ can be extended to the ambient space, in such a way that the choice of the derivatives orthogonal to the submanifold is arbitrary.

Lemma 2 [55]: Assume that $\mathcal{M}$ is a smooth closed manifold. Let $T_{\mathbf{x}}\mathcal{M}$ be the tangent space of $\mathcal{M}$ at a point $\mathbf{x} \in \mathcal{M}$. Let $V = \sum_{i=\dim(\mathcal{M})+1}^{d} v^i \partial_i$ be a conservative vector field along $\mathcal{M}$ which assigns a vector in $T_{\mathbf{x}}\mathcal{M}^{\perp}$ for each $\mathbf{x} \in \mathcal{M}$. For any smooth function $f^{\text{true}} : \mathcal{M} \to \mathbb{R}$, there exists a smooth extension $F : X \to \mathbb{R}$ such that

$$F|_{\mathcal{M}} = f^{\text{true}}$$

where $F|_{\mathcal{M}}$ denotes the restriction of F to the submanifold $\mathcal{M}$, and such that the derivative of the extension F is

$$\nabla_{\mathbf{x}} F(\mathbf{x}) \\ = \nabla_1 f^{\text{true}}(\mathbf{x}), \ldots, \nabla_{\dim(\mathcal{M})} f^{\text{true}}(\mathbf{x}) v^{\dim(\mathcal{M})+1}(\mathbf{x}), \ldots, v^d(\mathbf{x})$$

for all $\mathbf{x} \in \mathcal{M}$.

Notice that in Lemma 2, in $\nabla_{\mathbf{x}} F(\mathbf{x})$, we pick the local coordinates at $\mathbf{x} \in \mathcal{M}$, such that the first set of components parametrizes the data manifold. We stress that, as $\mathcal{M}$ is smooth, this is without any loss of generality [16]. Lemma 2, together with the universal approximation capabilities of NNs [41], is employed in Proposition 2 to show that for any possible value $\mathbf{v}$ of the orthogonal gradient to the data manifold in a point, there exists at least two (possibly not unique) different weight vectors that achieve zero loss and have orthogonal gradients, respectively, equal to $\mathbf{v}$ and $-\mathbf{v}$.

Proposition 2: Consider an infinitely-wide NN f with smooth, bounded, and nonconstant activation functions and an input $\mathbf{x} \in \mathcal{M}$, where $\mathcal{M}$ is a smooth closed manifold. Then, for any smooth function $f^{\text{true}} : \mathcal{M} \to \mathbb{R}$ and vector $\mathbf{v} \in \mathbb{R}^{\dim(X)-\dim(\mathcal{M})}$, there exist $\mathbf{w}_1$ and $\mathbf{w}_2$ such that

$$f(\cdot, \mathbf{w}_1)|_{\mathcal{M}} = f^{\text{true}} = f(\cdot, \mathbf{w}_2)|_{\mathcal{M}} \tag{14}$$

$$\nabla_{\mathbf{x}}^{\perp} f(\mathbf{x}, \mathbf{w}_1) = \mathbf{v} = -\nabla_{\mathbf{x}}^{\perp} f(\mathbf{x}, \mathbf{w}_2). \tag{15}$$

Proof: From Lemma 2, we know that there exist smooth extensions F^+ and F^- of f^{true} to the embedding space such that $\nabla_{\mathbf{x}}^{\perp} F^+(\mathbf{x}) = \mathbf{v} = -\nabla_{\mathbf{x}}^{\perp} F^-(\mathbf{x})$. As a consequence, to conclude the proof, it suffices to apply Theorem 3 in [41] that guarantees that infinitely-wide NNs are *uniformly 1-dense* on compacts in $\mathcal{C}^1(X)$, under the assumptions of smooth, bounded, and nonconstant activation functions. Specifically, for any $F \in \mathcal{C}^1(X)$ and $\epsilon > 0$, for any compact $X' \subseteq X$, there exists a set of weights $\mathbf{w}$ s.t.

$$\max \left\{ \sup_{\mathbf{x}\in X'} ||F(\mathbf{x}) - f(\mathbf{x},\mathbf{w})||_\infty, \right. \\ \left. \sup_{\mathbf{x}\in X'} ||\nabla F(\mathbf{x}) - \nabla f(\mathbf{x},\mathbf{w})||_\infty \right\} \leq \epsilon.$$

As $F^+, F^- \in \mathcal{C}^1(X)$, this concludes the proof. □

Note that by the chain rule, the gradient of the loss is proportional to the gradient of the NN. As a consequence, Proposition 2 guarantees that, for infinitely-wide NNs, for any weights set achieving the minimum loss, there exists another weights set with the same loss and opposite orthogonal gradient of the loss with respect to the input. This has various implications: 1) DNNs trained on a data manifold could exhibit arbitrarily large gradients in directions orthogonal to the data manifold, even when they learn the latent function perfectly and 2) in a Bayesian framework, by averaging over weights sets that have opposite orthogonal gradients, one could achieve a robust model that has vanishing expected orthogonal gradient. In Section IV, in Theorems 1 and 2, we show that, under some relatively mild assumptions, this is indeed the case. However, we should already emphasize that such a result does not hold by simply averaging the set of weights with respect to any distribution. Intuitively, for this result to hold, it is required that each set of weights achieving a given gradient value has the same measure as the set of weights with the same loss and opposite orthogonal gradient value.

IV. Adversarial Robustness via Bayesian Averaging

In order to prove that BNNs have vanishing orthogonal gradients, we start from a general GP [56] trained with a given dataset. We show that, under the assumption that the data manifold is a linear subspace of the ambient space, the projection of the expected gradient of the GP in a direction orthogonal to the data manifold vanishes for all points in the data manifold. This shows that GPs are able to obtain perfect cancellation of the orthogonal gradients. By relying on the convergence of BNNs to GPs, we then extend this result to infinitely-wide BNNs.

We first state our results for a regression setting, the classification setting will then be considered in Section IV-A. Theorem 1 shows that for a wide class of covariance functions, a GP trained on a regression problem has zero expected orthogonal gradients. In Corollary 1, we then extend this result to BNNs.

Theorem 1: Let $z(\mathbf{x})$ be a zero-mean GP with covariance function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Call $z(\mathbf{x}) \mid D_N$ the posterior GP obtained by training z on a regression problem with dataset $D_N = \{(\mathbf{x}_i, \mathbf{y}_i) \mid i = 1, \ldots, N\}$ and additive indepdent and identically distributed (i.i.d.) zero-mean Gaussian observation noise of variance $\sigma^2 \geq 0$. Assume the following.

1) For $\mathbf{x}', \mathbf{x}'' \in \mathcal{M}$, $K(\mathbf{x}', \mathbf{x}'') = g(\sum_{i=1}^d (x_i' - x_i'')^l)$, $l > 1$ or $K(\mathbf{x}', \mathbf{x}'') = g(\sum_{i=1}^d x_i' x_i'')$, where g is any twice differentiable function such that K is a valid kernel.
2) $\mathcal{M}$ is a linear subspace of $X \subseteq \mathbb{R}^d$.

Then, for any $\mathbf{x} \in \mathcal{M}$, it holds that

$$\langle \nabla_{\mathbf{x}}^{\perp} z(\mathbf{x}) \rangle_{p(z(\mathbf{x})|D_N)} = 0. \tag{16}$$

Proof: Conditional distributions of jointly Gaussian random variables are still Gaussian [57]. Consequently, $z \mid D_N$ is a GP with mean at $\mathbf{x}$ given by

$$\langle z(\mathbf{x}) \rangle_{p(z(\mathbf{x})|D_N)} = K(\mathbf{x}, X)\big(K(X,X) + \sigma^2 I\big)^{-1}[\mathbf{y}_1, \ldots, \mathbf{y}_N]^{\mathrm{T}} \tag{17}$$

where

$$K(\mathbf{x}, X) = [K(\mathbf{x}, \mathbf{x}_1), \ldots, K(\mathbf{x}, \mathbf{x}_N)]$$
$$K(X,X) = \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & \cdots & K(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ K(\mathbf{x}_N, \mathbf{x}_1) & \cdots & K(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}$$

and I is the identity matrix of size equal to $K(X,X)$. As the derivative of a GP is still a GP with mean given by the derivative of its mean [57], by the linearity of the expectation and the definition of vector projection, we obtain that

$$\langle \nabla_{\mathbf{x}}^{\perp} z(\mathbf{x}) \rangle_{p(z(\mathbf{x})|D_N)} \tag{18}$$
$$= \left(\sum_{i=1}^d v_i \frac{\partial K(\mathbf{x}, X)}{\partial x_i} \right) \big(K(X,X) + \sigma^2 I\big)^{-1}[\mathbf{y}_1, \ldots, \mathbf{y}_N]^{\mathrm{T}} \cdot \mathbf{v} \tag{19}$$

where $\mathbf{v} = (v_1, \ldots, v_d)$ is a unit vector orthogonal to $\mathcal{M}$ in $\mathbf{x}$, which always exists if $\dim(\mathcal{M}) < \dim(X)$. Hence, if we can show that for any $\mathbf{x}' \in \mathcal{M}$, $\sum_{i=1}^d v_i (\partial K(\mathbf{x}, \mathbf{x}')/\partial x_i) = 0$, then the proof is concluded. In order to do that, we notice that, as $\mathcal{M}$ is a subspace, the orthogonal direction $\mathbf{v}$ is a vector of the orthogonal complement of $\mathcal{M}$. Consequently, it holds that

$$\forall \mathbf{x} \in \mathcal{M}, \quad \sum_{i=1}^d v_i x_i = 0.$$

From this, for the case $K(\mathbf{x}, \mathbf{x}') = g(\sum_{i=1}^d x_i x_i')$, we obtain that for any $\mathbf{x}' \in \mathcal{M}$, it holds that

$$\sum_{i=1}^d v_i \frac{\partial K(\mathbf{x}, \mathbf{x}')}{\partial x_i} = \sum_{i=1}^d v_i g' \left(\sum_{j=1}^d x_j x_j' \right) x_i'$$
$$= g' \left(\sum_{i=1}^d x_i x_i' \right) \left(\sum_{i=1}^d v_i x_i' \right) = 0.$$

The case $K(\mathbf{x}, \mathbf{x}') = g(\sum_{i=1}^d (x_i - x_i')^l)$ follows similarly using the fact that subspaces are closed under linear combination. □

Corollary 1: Let $f(\mathbf{x}, \mathbf{w})$ be a BNN trained on a regression problem with dataset $D_N = \{(\mathbf{x}_i, \mathbf{y}_i) \mid i = 1, \ldots, N\}$ and additive i.i.d. zero-mean Gaussian observation noise of variance $\sigma^2 > 0$. Assume the following.

1) To each weight and bias are associated independent normal priors.
2) $\mathcal{M}$ is a linear subspace of $X \subseteq \mathbb{R}^d$.

Then, for infinitely-wide BNNs, it holds that

$$\langle \nabla_{\mathbf{x}}^{\perp} f(\mathbf{x}, \mathbf{w}) \rangle_{p(f(\mathbf{x},\mathbf{w})|D_N)} = 0. \tag{20}$$

Proof: In the limit of infinitely-wide architecture, thanks to Proposition 1, $f(\mathbf{x}, \mathbf{w})$ converges weakly to a GP. However, in general, the resulting kernel will not directly match the kernels satisfying the assumption of Theorem 1. Rather, in the case of BNNs with fully connected architectures, it will be of the form [42][8]

$$K(\mathbf{x}, \mathbf{x}') = g(\mathbf{x}\mathbf{x}, \mathbf{x}\mathbf{x}', \mathbf{x}'\mathbf{x}')$$

where $\mathbf{x}\mathbf{x}'$ is the dot product between vectors $\mathbf{x}$ and $\mathbf{x}'$. However, by the chain rule for multivariable functions, we get that

$$\frac{\partial K(\mathbf{x}, \mathbf{x}')}{\partial x_i} = 2x_i D_1[g(\mathbf{x}\mathbf{x}, \mathbf{x}\mathbf{x}', \mathbf{x}'\mathbf{x}')] + x_i' D_2[g(\mathbf{x}\mathbf{x}, \mathbf{x}\mathbf{x}', \mathbf{x}'\mathbf{x}')] \quad (21)$$

where $D_i[g]$ indicates the partial derivative of function g with respect to argument i. Consequently, being both $\mathbf{x}$ and $\mathbf{x}'$ in the data manifold $\mathcal{M}$, the derivations in the proof of Theorem 1 apply to each of the argument of (21). To conclude the proof, it is enough to notice that [58, Proposition 1] guarantees that the BNN posterior converges weakly to the posterior induced by the GP limit of the prior. Consequently, being

$$\mathcal{G}_{\mathbf{x}} : f \mapsto \nabla_{\mathbf{x}}^{\perp} f(\mathbf{x}, \mathbf{w})$$

a linear operator [59] and bounded by assumption, we have that

$$\left\langle \nabla_{\mathbf{x}}^{\perp} f(\mathbf{x}, \mathbf{w}) \right\rangle_{p(f(\mathbf{x},\mathbf{w})|D_N)} \rightarrow \left\langle \nabla_{\mathbf{x}}^{\perp} z(\mathbf{x}) \right\rangle_{p(z(\mathbf{x})|D_N)} = 0$$

where $\langle \nabla_{\mathbf{x}}^{\perp} z(\mathbf{x}) \rangle_{p(z(\mathbf{x})|D_N)}$ is as defined in Theorem 1. □

Note that the assumptions in Theorem 1 on K are relatively mild. In fact, not only they include the kernels given by the limiting GP of most common BNN architectures including also convolutional NNs [60], but they are also satisfied by most kernels used in practice for learning with GPs. Consequently, our results also apply to recent approaches that encode informative functional priors for BNNs as GPs [61]. Note also that an implicit assumption of Theorem 1 and Corollary 1 is that the partial derivatives at $\mathbf{x}$ are well-defined. In the case of rectified linear unit (ReLU) activation functions, this may not be always the case. In these cases, the results can be equivalently stated by considering subderivatives.

We should also stress that Theorem 1 and Corollary 1 hold for any size of D_N. Of course, in this general setting, the projection of the expected gradient of the loss to a direction parallel to $\mathcal{M}$ may not be zero, but will only point to directions where the loss increases within the data manifold. Consequently, if a BNN has small loss everywhere in the data manifold, then necessarily $\langle \nabla_{\mathbf{x}}^{\perp} f(\mathbf{x}) \rangle_{p(f(\mathbf{x},\mathbf{w})|D_N)}$ will vanish for any $\mathbf{x} \in \mathcal{M}$.

Remark 1: In Theorem 1, we only consider the expectation of the orthogonal gradient. In fact, we remark that Theorem 1 does not imply that also the variance of the orthogonal projection of the GP vanishes. In particular, in general, the variance of the orthogonal gradient is nonzero, showing how the robustness properties occur only in expectation. This will be empirically investigated in Section VI-A.

Remark 2: The assumption that $\mathcal{M}$ is a linear subspace is needed to guarantee that all training points are orthogonal to the same vector defining the orthogonal direction in $\mathbf{x}$. This guarantees perfect cancellation. However, in the more general case, where $\mathcal{M}$ is not a linear subspace, it is reasonable to expect that Theorem 1 and Corollary 1 can still often approximately hold. In fact, for many kernels, only training points close to the test point $\mathbf{x}$ will influence the prediction at $\mathbf{x}$. Consequently, for the cancellation in Theorem 1 to hold, it would be enough that the orthogonal direction in $\mathbf{x}$ remains approximately orthogonal only in a neighborhood of $\mathbf{x}$, thus extending our result to more general $\mathcal{M}$.

A. Extension to Classification Setting

Theorem 1 is stated for regression, which is a particularly favorable setting for analysis because of the existence of closed-form solutions for the GP posterior. Fortunately, in Theorem 2 and Corollary 2, we show that our results also extend to classification.

Theorem 2: Let $z(\mathbf{x})$ be a zero-mean GP with covariance function $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$. Call $z(\mathbf{x}) \mid D_N$ the posterior GP obtained by training z on a classification problem with dataset $D_N = \{(\mathbf{x}_i, \mathbf{y}_i) \mid i = 1, \ldots, N, y_i \in \{0, 1\}\}$. Then, under the same assumptions of Theorem 1, for any $\mathbf{x} \in \mathcal{M}$, it holds that

$$\langle \nabla_{\mathbf{x}}^{\perp} z(\mathbf{x}) \rangle_{p(z(\mathbf{x})|D_N)} = 0. \quad (22)$$

Proof: Because of the non-Gaussianity of the likelihood, $z(\mathbf{x}) \mid D_N$ is not Gaussian in general. However, it is still possible to show that (see [57, eq. (3.22)])

$$\langle z(\mathbf{x}) \rangle_{p(z(\mathbf{x})|D_N)} = K(\mathbf{x}, X) K(X, X)^{-1} g(D_N) \quad (23)$$

where g is a function given by the expectation of the latent function given the training data. Consequently, as g is independent of $\mathbf{x}$, under boundedness assumption of $K(X, X)^{-1} g(D_N)$, the same approach considered in the proof of Theorem 1 following (17) holds also in this setting. □

Corollary 2: Let $f(\mathbf{x}, \mathbf{w})$ be a BNN trained on a classification problem with dataset $D_N = \{(\mathbf{x}_i, \mathbf{y}_i) \mid i = 1, \ldots, N, y_i \in \{0, 1\}\}$. Then, under the same assumptions of Theorem 1, for any $\mathbf{x} \in \mathcal{M}$, it holds that

$$\langle \nabla_{\mathbf{x}}^{\perp} f(\mathbf{x}, \mathbf{w}) \rangle_{p(f(\mathbf{x},\mathbf{w})|D_N)} \rightarrow 0.$$

Proof: The proof is analogous to that of Corollary 1 by noticing that in the infinitely-width limit (trained) BNNs converge to GPs also in the classification setting [58]. □

V. Consequences and Limitations of Our Results

The results presented in Section IV have the natural consequence of protecting BNNs against gradient-based attacks, due to the vanishing average of the expectation of the gradients in the limit. Its proof also sheds light on a number of observations made in recent years. Before moving on to empirically validating the theorem in the finite-width case, it is worth reflecting on some of its implications and limitations.

[8] Note that the result also holds for convolutional BNNs where for the convolutional kernel instead, each monomial could be weighted differently depending on the number of filters, size and stride of the kernel.

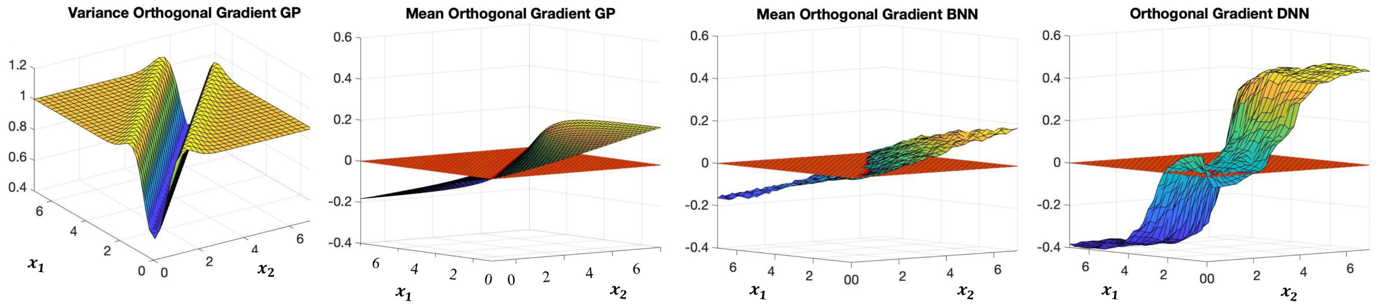


Fig. 1. We consider a regression problem with data manifold given by the line $x_1 = x_2$ and data generated by the function $((2x_1^2)/10) - x_1$. For this problem, we train a BNN with HMC and a DNN with SGD of the same architecture: ReLU activation functions and one hidden layer with 512 neurons. Furthermore, we also train a GP with kernel equal to that of an infinitely-wide BNN with ReLU activation functions and one hidden layer. All learning models achieve accuracy >99%. We plot the mean and variance of the scalar projection of the gradient of the GP in a direction orthogonal to the data manifold for all points in the ambient space and compare it to the mean of the same quantity for the BNN and DNN. Plane $z = 0$ is plotted in red.

1) Theorem 1 holds in a specific thermodynamic limit; however, it is reasonable to expect that the averaging effect of BNN gradients can still often provide considerable protection in conditions when the network architecture leads to high accuracy and strong expressivity.
2) Our results holds when the prediction is obtained by averaging with respect to the true posterior. Unfortunately, for NNs, it is generally unfeasible to obtain the true posterior, and cheaper variational approximations are commonly employed [17]. In practice, depending on the quality of the approximation, these can still provide protection from adversarial attacks.
3) Theorems 1 and 2 are stated for GPs. Consequently, these theorems not only provide theoretical backing to recent empirical observations of the adversarial robustness of GPs [62], [63], [64], but also generalize to all processes that converge to GPs thanks to the central limit theorem.
4) While the Bayesian posterior ensemble may not be the only randomization to provide protection, it is clear that some simpler randomizations such as bootstrap will be ineffective, as noted empirically by Bekasov and Murray [13]. This is because bootstrap resampling introduces variability along the data manifold, rather than in orthogonal directions. In this sense, bootstrap clearly cannot be considered a Bayesian approximation, especially when the data distribution has zero measure support with respect to the ambient space.
5) Our results only guarantee protection against gradient-based adversarial attacks. As a consequence, it is not clear if the robustness properties of BNNs also extend to nongradient based attacks. Empirical results in Section VI-E also suggest that the vanishing gradient properties of BNNs may provide an increased robustness against specific gradient-free attacks.

VI. Empirical Results

In this section, we empirically investigate our theoretical findings on different BNNs. We train a variety of BNNs on the MNIST and Fashion MNIST [65] datasets (using their standard 60k/10k train–test split) and evaluate their posterior distributions using HMC and VI approximate inference methods. Details on the architectural and training hyperparameters used throughout this section are listed in table form in the Appendix. In Section VI-A, we empirically validate the theoretical results of Section IV on a synthetic regression dataset on which we have access to the data manifold. In Section VI-B, we focus on image classification tasks and experimentally verify the validity of the zero-averaging property of gradients implied by the theoretical results discussed in Section IV, and discuss its implications on the behaviors of FGSM and PGD attacks on BNNs in Section VI-C. In Section VI, we analyze the relationship between robustness and accuracy on thousands of different NN architectures, comparing the results obtained by Bayesian and deterministic training. Further, in Section VI-E, we investigate the robustness of BNNs on a gradient-free adversarial attack [66]. Experiments were performed with an NVIDIA 2080Ti GPU on a machine with 20-core Intel Core Xeon 6230 and 256 GB of RAM.

A. Analysis of the Convergence of BNN Gradients

To investigate the limit behavior of the orthogonal and parallel gradients of BNNs, in Fig. 1, we consider a synthetic regression example, where the ambient space is $\mathbb{R}^2$ and the data manifold is the line $x_1 = x_2$. We consider data generated by the function $(2x_1^2/10) - x_1$ and on these data, we train both a one hidden layer BNN with 512 hidden neurons with HMC and the corresponding limiting GP, as well as a DNN trained with SGD on the same architecture. For each point in the ambient space, we compute the variance and mean of the scalar projection of the expected posterior gradient of the prediction of the model into a unit direction orthogonal to the data manifold. As expected (see Remark 1), we observe that the variance is nonvanishing on the ambient space, that is, both inside and outside the data manifold. Furthermore, while for GPs and BNNs, the mean of the orthogonal projection is, respectively, zero and approximately zero in the data manifold; for the DNN, this is not always the case. Additionally, the absolute value of the expected orthogonal projection in any point in the ambient space is substantially greater for the DNN compared to the GP and BNN cases.

TABLE I

FOR THE SETTING DESCRIBED IN FIG. 1, WE TRAINED VARIOUS NNS WITH HMC AND SGD. AVERAGE ORTHOGONAL AND PARALLEL GRADIENTS AND MAXIMUM ORTHOGONAL AND PARALLEL GRADIENTS ARE COMPUTED OVER 70 POINTS SAMPLED FROM THE DATA MANIFOLD. ALL ARCHITECTURES HAVE ONE HIDDEN LAYER

Method	Hidden Neurons	avg ort grad	max ort grad	avg par grad	max par grad
HMC ($\sigma = 0.4$)	16	0.0556	0.1116	1.1961	2.2275
HMC ($\sigma = 0.4$)	64	0.0613	0.0836	1.2028	2.1420
HMC ($\sigma = 0.4$)	512	0.0132	0.0355	1.1991	2.1724
HMC ($\sigma = 0.25$)	16	0.0976	0.1508	1.2537	2.3727
HMC ($\sigma = 0.25$)	64	0.0702	0.1052	1.2402	2.3064
HMC ($\sigma = 0.25$)	512	0.0359	0.0554	1.2487	2.3353
HMC ($\sigma = 0.1$)	16	0.1127	0.1921	1.3214	2.6608
HMC ($\sigma = 0.1$)	64	0.0539	0.1271	1.3171	2.6290
HMC ($\sigma = 0.1$)	512	0.0479	0.0900	1.3251	2.6669
SGD	16	0.4337	0.8056	1.5525	3.1999
SGD	64	0.2612	0.6676	1.5435	3.2238
SGD	512	0.1234	0.2491	1.5262	3.1424

In order to further investigate the qualitative behavior observed in Fig. 1, we consider the same setting of Fig. 1 and, in Table I, we report the average orthogonal and parallel scalar projection of the gradient for various BNN architectures and compare it against the same architectures trained with SGD. In particular, we consider BNNs trained with HMC with different values of σ, the standard deviation of the normal likelihood employed in regression training, and for different number of neurons. All architectures achieve expected loss ≤ 0.02, and average and maximum gradients are evaluated on the same 70 points sampled from the data manifold. As expected from Corollary 1, for all values of σ, both average and maximum orthogonal gradients decrease with a wider network, while the parallel gradient remains approximately constant. For SGD, the orthogonal gradients are in all cases substantially larger compared to those of BNNs. However, interestingly, even in the SGD case, wider networks achieve lower orthogonal gradients. This suggests that the limit established in Corollaries 1 and 2 may also partly benefit NNs trained with SGD although the convergence is much slower. Related to this, we also note that σ has a substantial impact on the robustness of the BNN. This can be understood because, the lower the σ, the smaller the variance of the posterior BNN, and the larger the orthogonal gradient of the limit GP (see Fig. 2). This indicates how the uncertainty can be beneficial in increasing the robustness of the predictions and how training and model parameters can play an essential role for BNN robustness.

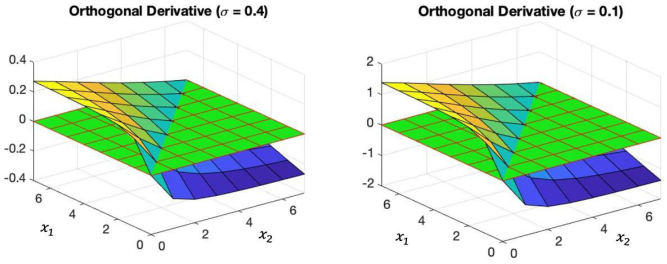


Fig. 2. We plot the scalar projection of the orthogonal gradient of the GP limit of NNs with ReLU activation functions and one hidden layer for $\sigma = 0.1$ and $\sigma = 0.4$ for the settings of Fig. 1, where σ is the standard deviation of the likelihood. It is possible to observe how in both cases the orthogonal derivative is identically 0 on the data manifold. However, outside of the data manifold σ has a large effect on the orthogonal derivative.

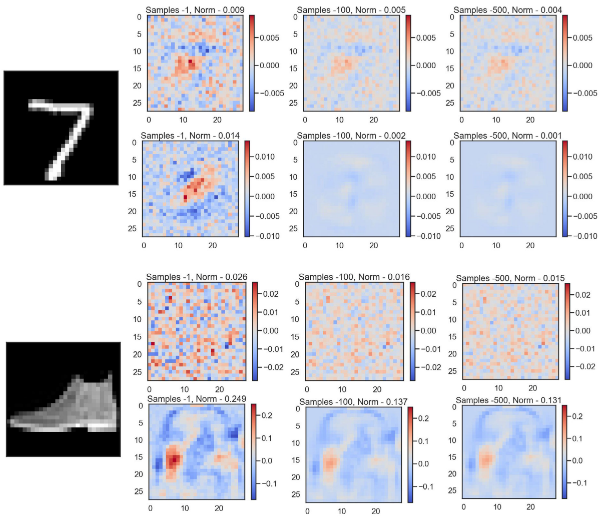


Fig. 3. We plot the input gradient (and report its ℓ_∞-norm under "Norm" in the title of each plot) of the expected loss gradients for two BNNs trained on MNIST (top rows) and Fashion MNIST (bottom rows) for some example images and for different number of samples from the posterior predictive distribution. For training the BNN on MNIST, we employ HMC (top most row of next to each image) and VI (bottom most row next to each image). To the right of the images, we plot a heat map of gradient values. In all cases, we observe that the expected loss gradients decrease when increasing the number of samples.

B. Evaluation of the Gradient of the Loss for BNNs on Image Classification Tasks

We investigate the vanishing behavior of input gradients of the loss—established for the BNN classification case by Corollary 2 for the limit regime—in the finite, practical settings, that is with a finite-width BNN. Specifically, we consider various NN architectures (those that will be reported in Section VI-D) trained with both HMC and VI with Bayes by Backprop [17] and we select the architectures achieving the highest test accuracy: a two-hidden-layer BNN (with 1024 neurons per layer) for HMC and a three-hidden-layer BNN (512 neurons per layer) for VI. These achieve approximately 95% test accuracy on MNIST and 89% on Fashion MNIST when trained with HMC, as well as 95% and 92%, respectively, when trained with VI. Details about the hyperparameters used for training can be found in the Appendix.

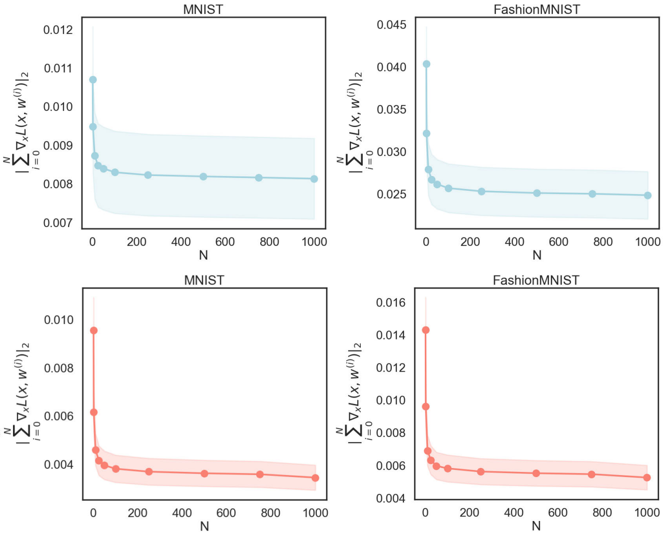


Fig. 4. We plot the ℓ_2-norm of the input gradient as we increase the number of samples. In the top row (in blue), we plot the trend of the input gradient for a BNN trained with VI on MNIST (left) and Fashion MNIST (right). In the bottom row (in red), we plot the trend of the input gradient for an HMC-trained BNN on MNIST (left) and Fashion MNIST (right). In all cases, we observe the expected trend that the norm of the gradient decreases as we increase the number of samples.

Fig. 3 investigates the behavior of the componentwise expectation of the loss gradient as more samples from the posterior distribution are incorporated into the BNN predictive distribution. We observe that as the number of samples taken from the posterior distribution of **w** increases, all the components of the gradient decrease in absolute value. Notice that the gradients of the individual NNs (that is, those with just one sampled weight vector) are markedly larger. This is also confirmed in Fig. 4, where we provide a systematic analysis of the aggregated gradient convergence properties on 250 test images for MNIST and Fashion MNIST. We plot the decrease in the maximum gradient magnitude (with shaded error regions corresponding to the square root of the variance) as we increase the number of samples used to approximate the input gradient. We observe that for both HMC and VI, the magnitude of the gradient components drops as the number of samples increases. Notice that the gradients computed on HMC trained networks drops more quickly and toward a smaller value compared to VI trained networks. This is in accordance to what is discussed in Section V, as VI introduces additional approximations in the Bayesian posterior computation.

C. Gradient-Based Attacks for BNNs

The fact that gradient cancellation occurs in the limit does not directly imply that BNNs' predictions are robust to gradient-based attacks in the finite case. For example, FGSM attacks [2] are crafted such that the direction of the manipulation is given only by the sign of the expectation of the loss gradient and not by its magnitude. Thus, even if the entries of the expectation drop to an infinitesimal magnitude but maintain a meaningful sign, then FGSM could potentially produce effective attacks. In order to test the implications of vanishing gradients on the robustness of the posterior predictive distribution against gradient-based attacks, we compare the behavior of FGSM, PGD[9] [6], and Carlini & Wagner (CW) [67] attacks, arguably the most commonly employed gradient-based attacks.

TABLE II

Adversarial Robustness Defined as the Proportion of Test Samples for Which the Attack Is Not Successful for BNNs Trained With HMC and VI With Respect to FGSM, PGD, and CW

Dataset/Method	Acc.	FGSM	PGD	CW
MNIST/SGD	0.984	0.376	0.334	0.358
MNIST/VI	0.974	0.623	0.508	0.512
MNIST/HMC	0.914	0.876	0.864	0.859
Fashion/SGD	0.891	0.510	0.410	0.432
Fashion/VI	0.862	0.622	0.581	0.588
Fashion/HMC	0.828	0.661	0.639	0.650

In particular, in Table II, we evaluate a single hidden-layer NN architecture with 512 hidden neurons trained on MNIST and Fashion MNIST using HMC, VI (Bayes by Backprop [17]), and SGD. We then attack it using attack strength $\epsilon = 0.1$ for MNIST and $\epsilon = 0.05$ for Fashion MNIST. For each image, we compute the expected gradient using 50 posterior samples. Details on the hyperparameters used in each setting can be found in the Appendix (entries marked with a *). We observe that BNNs are generally more robust against each of the gradient-based attacks. Interestingly, we also notice that for BNNs, FGSM performs comparably to PGD and CW attacks with little increase in attack success rate when using stronger attacks. Furthermore, we should also note that for Fashion MNIST, the attacks are more successful than in the MNIST case. As Fashion MNIST poses a more complicated classification problem than MNIST, the BNNs obtain less accuracy in the former. In agreement with the discussion in Section V, this implies higher loss and that the conditions set up in the main theorems and corollaries are less approximately met.

D. Robustness Accuracy Analysis in Deterministic and BNNs

In Section V, we noticed that, as proxy of the loss and of convergence, high accuracy might be related to high robustness to gradient-based attacks for BNNs. Notice that this would run counter to what has been observed for DNNs trained with SGD [68].

In this section, we look at an array of more than 1000 BNNs with different hyperparameters trained with HMC and VI (Bayes by Backprop [17]) on MNIST and Fashion MNIST (details on training and architecture hyperparameters explored can be found in the Appendix). We experimentally evaluate their accuracy/robustness tradeoff on FGSM attacks as compared to that of the same NN architectures trained via standard (i.e., non-Bayesian) SGD based methods. For the robustness evaluation, we consider the average difference in the softmax prediction between the original test points and the crafted adversarial input, as this provides a quantitative and smooth measure of adversarial robustness that is closely related with misclassification ratios [69]. That is, for a collection of N test

[9] With 15 iterations and one restart.

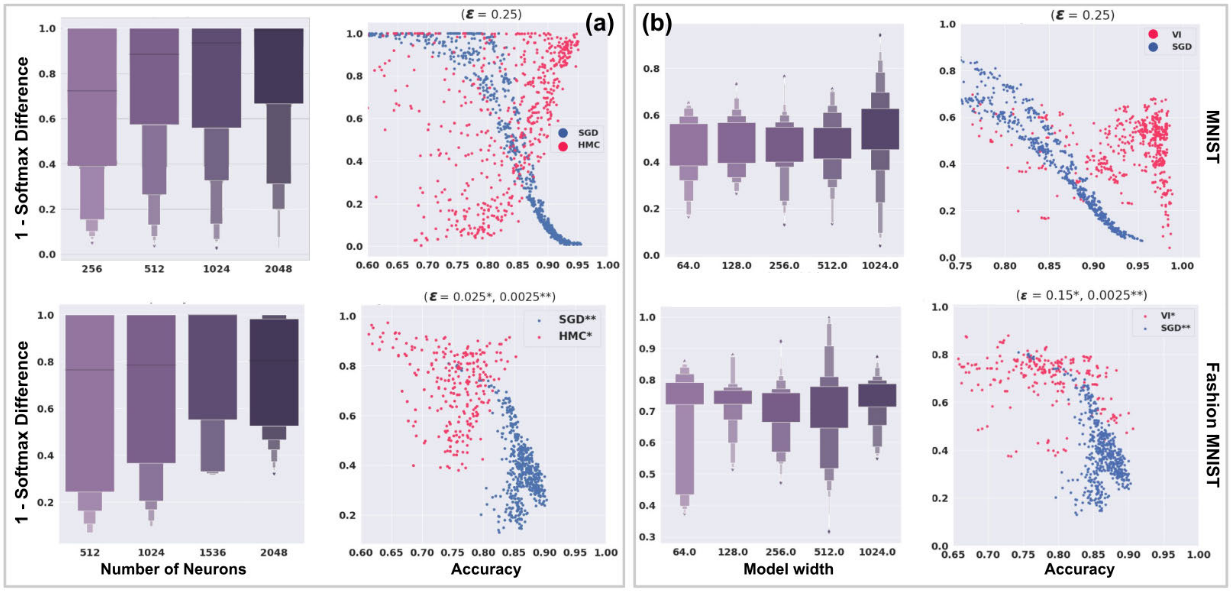


Fig. 5. Robustness–accuracy tradeoff on MNIST (first row) and Fashion MNIST (second row) for BNNs trained with (a) HMC, (b) VI, and SGD (blue dots), where softmax difference is computed according to (24) and denotes the average maximal difference in softmax value for the specific NN for an input ϵ-ball of input points computed via FGSM attack. While a tradeoff between accuracy and robustness occur for DNNs, experiments on HMC show a positive correlation between accuracy and robustness. The boxplots show the correlation between model capacity and robustness. Different attack strengths (ϵ) are used for the three methods accordingly to their average robustness.

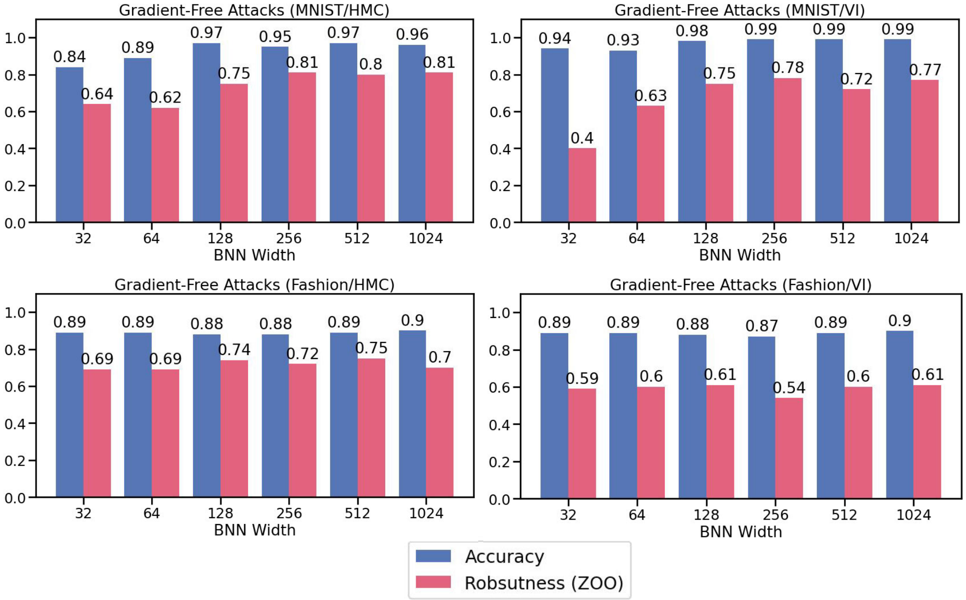


Fig. 6. Gradient-free adversarial attacks on BNNs display similar behavior to gradient-based attacks. We evaluate the more accurate networks from Fig. 5 with gradient-free attacks on MNIST (first row) and Fashion MNIST (second row) for BNNs trained with HMC (left column), VI (center column), and SGD (right column). We use the same attack parameters as in Fig. 5, but use ZOO as an attack method.

point, we compute

$$\frac{1}{N}\sum_{j=1}^{N}\left|\langle f(\mathbf{x}_j,\mathbf{w})\rangle_{p(\mathbf{w}|D)} - \langle f(\tilde{\mathbf{x}}_j,\mathbf{w})\rangle_{p(\mathbf{w}|D)}\right|_\infty. \qquad (24)$$

The results of the analysis are plotted in Fig. 5. Each dot in the scatterplots represents the results obtained for each specific network architecture trained with SGD (blue dots), HMC [pink dots in Fig. 5(a)], and VI [pink dots in Fig. 5(b)]. As already reported in [68], we observe a marked tradeoff between accuracy and robustness (i.e., 1—softmax difference) for high-performing deterministic networks. Interestingly, this trend is reversed for BNNs trained with HMC [Fig. 5(a)] where we find that as networks become more accurate, they additionally become more robust to FGSM attacks as well. We further examine this trend in the boxplots that represent the effect that the network width has on the robustness of the resulting posterior. We find the existence of an increasing trend in robustness as the number of neurons in the network is increased. This is in line with our theoretical findings, i.e., as the BNN approaches the infinite width limit, the conditions for Corollary 2 are approximately met and the network is protected against gradient-based attacks.

On the other hand, the tradeoff behaviors are less obvious for the BNNs trained with VI and on Fashion MNIST. In par-

TABLE III

HYPERPARAMETERS FOR THE BNN USING HMC OF FIGS. 3 AND 4

Training hyperparameters for HMC

Dataset	MNIST	Fashion MNIST
Training inputs	60k	60k
Hidden size	1024	1024
Nonlinear activation	ReLU	ReLU
Architecture	Fully Connected	Fully Connected
Posterior Samples	500	500
Numerical Integrator Stepsize	0.002	0.001
Number of steps for Numerical Integrator	10	10

TABLE IV

HYPERPARAMETERS FOR THE BNN USING VI OF FIGS. 3 AND 4

Training hyperparameters for VI

Dataset	MNIST	Fashion MNIST
Training inputs	60k	60k
Hidden size	512	1024
Nonlinear activation	Leaky ReLU	Leaky ReLU
Architecture	Convolutional	Convolutional
Training epochs	5	10
Learning rate	0.01	0.001

ticular, in Fig. 5(b), we find that, similar to the deterministic case, also for BNNs trained with VI, robustness seems to have a negative correlation with accuracy, albeit less marked than for SGD. Furthermore, similarly than for HMC, we also observe that for VI, the robustness of the BNNs tend to increase with the width of the network. However, the trend is less clear compared to the HMC case. This can be linked to the fact that VI is known to often under-approximate uncertainty [57], which may lead to posterior with excessively small variance, thus behaving similar to a DNN. We should also emphasize that the experimental results in this article have been performed with Bayes by Backprop [17]. However, approximate Bayesian inference techniques for deep learning is an active area of research, including recent developments, e.g., SWAG [70] or Noisy Adam [71], such developments could, in principle, lead to a behavior closer to that exhibited by HMC.

E. Gradient-Free Adversarial Attacks

In this section, we empirically evaluate the most accurate BNN posteriors on MNIST and Fashion MNIST from Fig. 5 against gradient-free adversarial attacks. Specifically, we consider ZOO [72], a gradient-free adversarial attack based on a finite-difference approximation of the gradient of the loss obtained by querying the NN (in the BNN case, the attacker queries the posterior distribution). In particular, we selected ZOO because it has been shown to be effective even when tested on networks that purposefully obfuscate their gradients or have vanishing gradients [66]. In Fig. 6, we observe that similar to gradient-based methods, ZOO is substantially less effective on BNNs compared to DNNs in all cases, with BNNs again achieving both high accuracy and high robustness simultaneously. Furthermore, once again HMC is more robust to the attack than VI, which is in turn substantially more robust than DNNs. This suggests how, similar to what observed in Sections VI-A–VI-D, a more accurate posterior distribution may lead to a more robust model also to gradient-free adversarial attacks.

TABLE V

HYPERPARAMETERS FOR TRAINING BNNS WITH HMC FOR RESULTS REPORTED IN FIG. 5. * INDICATES THE PARAMETERS USED IN TABLE II

HMC MNIST/Fashion MNIST grid search

Posterior samples	{250*, 500, 750}
Numerical Integrator Stepsize	{0.025*, 0.01, 0.005, 0.001, 0.0001}
Numerical Integrator Steps	{10, 15, 20*}
Hidden size	{128, 256, 512*}
Nonlinear activation	{relu*, tanh, sigmoid}
Architecture	{1*,2,3} fully connected layers

TABLE VI

HYPERPARAMETERS FOR TRAINING BNNS WITH VI FOR RESULTS REPORTED IN FIG. 5. * INDICATES THE PARAMETERS USED IN TABLE II

VI MNIST/Fashion MNIST grid search

Learning Rate	{0.001*}
Minibatch Size	{128*, 256, 512, 1024}
Hidden size	{64, 128, 256, 512*, 1024}
Nonlinear activation	{relu*, tanh, sigmoid}
Architecture	{1*,2,3} fully connected layers
Training epochs	{3,5,7,9,12,15*} epochs

TABLE VII

HYPERPARAMETERS FOR TRAINING NNS WITH SGD FOR RESULTS REPORTED IN FIG. 5. * INDICATES THE PARAMETERS USED IN TABLE II

SGD MNIST/Fashion MNIST grid search

Learning Rate	{0.001, 0.005, 0.01, 0.05*}
Hidden size	{64, 128, 256, 512*}
Nonlinear activation	{relu*, tanh, sigmoid}
Architecture	{1*, 2, 3, 4, 5} fully connected layers
Training epochs	{5, 10, 15, 20*, 25} epochs

VII. CONCLUSION

The quest for robust, data-driven models is an essential component toward the construction of AI-based technologies. In this respect, we believe that the fact that Bayesian ensembles of NNs can provide additional robustness against a broad class of adversarial attacks will be of great relevance. While promising, this result comes with some significant limitations. First, while our empirical results present encouraging examples of robustness, our theoretical analysis is only guaranteed to hold for infinite NNs. Second, and perhaps more importantly, performing Bayesian inference in large nonlinear models is extremely challenging. In fact, while in our experiments, cheaper approximations, such as VI, also enjoyed a degree of adversarial robustness, albeit reduced compared to NNs trained with more accurate Bayesian inference methods, there are no guarantees that this will hold in general. To this end, we hope that this result will spark renewed interest in the pursuit of efficient Bayesian inference algorithms, which have the potential to lead to learning models that are intrinsically accurate and robust.

APPENDIX

HYPERPARAMETERS USED FOR TRAINING

Details on hyperparameters used for training of BNNs and DNNs are listed in Tables III–VII.

REFERENCES

[1] C. Szegedy et al., "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.

[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[3] J. Zhang and C. Li, "Adversarial examples: Opportunities and challenges," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 7, pp. 2578–2593, Jul. 2020.

[4] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognit.*, vol. 84, pp. 317–331, Dec. 2018.

[5] Y. Zhuo, Z. Song, and Z. Ge, "Security versus accuracy: Trade-off data modeling to safe fault classification systems," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–12, 2023.

[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.

[7] R. M. Neal, *Bayesian Learning for Neural Networks*, vol. 118. Berlin, Germany: Springer, 2012.

[8] X. Jia et al., "An energy-efficient Bayesian neural network implementation using stochastic computing method," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–11, 2023.

[9] H. Li, P. Barnaghi, S. Enshaeifar, and F. Ganz, "Continual learning using Bayesian neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 9, pp. 4243–4252, Sep. 2021.

[10] J.-T. Chien and Y.-C. Ku, "Bayesian recurrent neural network for language modeling," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 2, pp. 361–374, Feb. 2016.

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

[12] M. Wicker, L. Laurenti, A. Patane, Z. Chen, Z. Zhang, and M. Kwiatkowska, "Bayesian inference with certifiable adversarial robustness," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2021, pp. 2431–2439.

[13] A. Bekasov and I. Murray, "Bayesian adversarial spheres: Bayesian inference and adversarial examples in a noiseless setting," 2018, *arXiv:1811.12335*.

[14] X. Liu, Y. Li, C. Wu, and C.-J. Hsieh, "Adv-BNN: Improved adversarial defense through robust Bayesian neural network," 2018, *arXiv:1810.01279*.

[15] M. Yuan, M. Wicker, and L. Laurenti, "Gradient-free adversarial attacks for Bayesian neural networks," 2020, *arXiv:2012.12640*.

[16] J. M. Lee, "Smooth manifolds," in *Introduction to Smooth Manifolds*. New York, NY, USA: Springer, 2013, pp. 1–31.

[17] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1613–1622.

[18] G. Carbone, M. Wicker, L. Laurenti, A. Patane, L. Bortolussi, and G. Sanguinetti, "Robustness of Bayesian neural networks to gradient-based attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 15602–15613.

[19] C. Liu, M. Salzmann, and S. Süsstrunk, "Training provably robust models by polyhedral envelope regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 6, pp. 3146–3160, Jun. 2021.

[20] R. Reza Wiyatno, A. Xu, O. Dia, and A. de Berker, "Adversarial examples in modern machine learning: A review," 2019, *arXiv:1911.05268*.

[21] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019.

[22] A.-J. Gallego, J. Calvo-Zaragoza, and R. B. Fisher, "Incremental unsupervised domain-adversarial training of neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 11, pp. 4864–4878, Nov. 2021.

[23] P. Panda, "QUANOS: Adversarial noise sensitivity driven hybrid quantization of neural networks," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design*, 2020, pp. 187–192.

[24] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[25] J. Lin, C. Gan, and S. Han, "Defensive quantization: When efficiency meets robustness," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[26] S. Wang, S. Chen, T. Chen, S. Nepal, C. Rudolph, and M. Grobler, "Generating semantic adversarial examples via feature manipulation in latent space," *IEEE Trans. Neural Netw. Learn. Syst.*, pp. 1–15, 2024.

[27] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[28] Y. Pang, S. Cheng, J. Hu, and Y. Liu, "Evaluating the robustness of Bayesian neural networks against different types of attacks," 2021, *arXiv:2106.09223*.

[29] L. Smith and Y. Gal, "Understanding measures of uncertainty for adversarial example detection," 2018, *arXiv:1803.08533*.

[30] A. Uchendu, D. Campoy, C. Menart, and A. Hildenbrandt, "Robustness of Bayesian neural networks to white-box adversarial attacks," in *Proc. IEEE 4th Int. Conf. Artif. Intell. Knowl. Eng. (AIKE)*, Dec. 2021, pp. 72–80.

[31] R. Michelmore, M. Wicker, L. Laurenti, L. Cardelli, Y. Gal, and M. Kwiatkowska, "Uncertainty quantification with statistical guarantees in end-to-end autonomous driving control," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 7344–7350.

[32] Y. Gal and L. Smith, "Sufficient conditions for idealised models to have no adversarial examples: A theoretical and empirical study with Bayesian neural networks," 2018, *arXiv:1806.00667*.

[33] A. Rawat, M. Wistuba, and M.-I. Nicolae, "Adversarial phenomenon in the eyes of Bayesian deep learning," 2017, *arXiv:1711.08244*.

[34] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, Nov. 2017, pp. 3–14.

[35] K. Grosse, D. Pfaff, M. Thomas Smith, and M. Backes, "The limitations of model uncertainty in adversarial settings," 2018, *arXiv:1812.02606*.

[36] M. Wicker, L. Laurenti, A. Patane, and M. Kwiatkowska, "Probabilistic safety for Bayesian neural networks," in *Proc. Conf. Uncertainty Artif. Intell.*, 2020, pp. 1198–1207.

[37] L. Berrada et al., "Make sure you're unsure: A framework for verifying probabilistic specifications," 2021, *arXiv:2102.09479*.

[38] J. Zhang et al., "Robust Bayesian neural networks by spectral expectation bound regularization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3814–3823.

[39] N. Ye and Z. Zhu, "Bayesian adversarial learning," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst* Red Hook, NY, USA: Curran Associates, 2018, pp. 6892–6901.

[40] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control, Signals, Syst.*, vol. 2, no. 4, pp. 303–314, Dec. 1989.

[41] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, no. 2, pp. 251–257, 1991.

[42] A. G. de G. Matthews, M. Rowland, J. Hron, R. E. Turner, and Z. Ghahramani, "Gaussian process behaviour in wide deep neural networks," 2018, *arXiv:1804.11271*.

[43] G. Ongie, R. Willett, D. Soudry, and N. Srebro, "A function space view of bounded norm infinite width ReLU nets: The multivariate case," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–28.

[44] G. M. Rotskoff and E. Vanden-Eijnden, "Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error," 2018, *arXiv:1805.00915*.

[45] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[46] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). Berlin, Germany: Springer-Verlag, 2006.

[47] R. M. Neal et al., "MCMC using Hamiltonian dynamics," in *Handbook of Markov Chain Monte Carlo*, vol. 2, no. 11. Chapman & Hall, 2011, p. 2.

[48] R. J. Adler, *The Geometry of Random Fields*. SIAM, 2010.

[49] P. Billingsley, *Convergence of Probability Measures*. Hoboken, NJ, USA: Wiley, 2013.

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

[51] A. Garriga-Alonso, C. E. Rasmussen, and L. Aitchison, "Deep convolutional networks as shallow Gaussian processes," 2018, *arXiv:1808.05587*.

[52] A. Fawzi, H. Fawzi, and O. Fawzi, "Adversarial vulnerability for any classifier," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1178–1187.

[53] M. Khoury and D. Hadfield-Menell, "On the geometry of adversarial examples," 2018, *arXiv:1811.00525*.

[54] S. Goldt, M. Mézard, F. Krzakala, and L. Zdeborová, "Modelling the influence of data structure on learning in neural networks: The hidden manifold model," 2019, *arXiv:1909.11500*.

[55] C. Anders, P. Pasliev, A.-K. Dombrowski, K.-R. Müller, and P. Kessel, "Fairwashing explanations with off-manifold detergent," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 314–323.

[56] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When Gaussian process meets big data: A review of scalable GPs," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4405–4423, Nov. 2020.

[57] C. E. Rasmussen et al., *Gaussian Processes for Machine Learning*, vol. 1. Springer, 2005.

[58] J. Hron, Y. Bahri, R. Novak, J. Pennington, and J. Sohl-Dickstein, "Exact posterior distributions of wide Bayesian neural networks," 2020, *arXiv:2006.10541*.

[59] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, 2002.

[60] R. Novak et al., "Bayesian deep convolutional networks with many channels are Gaussian processes," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12.

[61] B.-H. Tran, S. Rossi, D. Milios, and M. Filippone, "All you need is a good functional prior for Bayesian deep learning," *J. Mach. Learn. Res.*, vol. 23, no. 74, pp. 1–56, 2022.

[62] L. Cardelli, M. Kwiatkowska, L. Laurenti, and A. Patane, "Robustness guarantees for Bayesian inference with Gaussian processes," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 7759–7768.

[63] K. Grosse, M. T. Smith, and M. Backes, "Killing four birds with one Gaussian process: The relation between different test-time attacks," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 4696–4703.

[64] A. Patane, A. Blaas, L. Laurenti, L. Cardelli, S. Roberts, and M. Kwiatkowska, "Adversarial robustness guarantees for Gaussian processes," *J. Mach. Learn. Res.*, vol. 23, pp. 1–55, Apr. 2022.

[65] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.

[66] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," 2018, *arXiv:1802.00420*.

[67] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, 2017, pp. 39–57.

[68] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao, "Is robustness the cost of accuracy?—A comprehensive study on the robustness of 18 deep image classification models," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 631–648.

[69] L. Cardelli, M. Kwiatkowska, L. Laurenti, N. Paoletti, A. Patane, and M. Wicker, "Statistical guarantees for the robustness of Bayesian neural networks," in *Proc. IJCAI*, 2019, pp. 1–9.

[70] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, and A. G. Wilson, "A simple baseline for Bayesian uncertainty in deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[71] G. Zhang, S. Sun, D. Duvenaud, and R. Grosse, "Noisy natural gradient as variational inference," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5852–5861.

[72] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, Nov. 2017, pp. 15–26.

Luca Bortolussi graduated in mathematics from the University of Trieste, Trieste, Italy, in 2003, and the Ph.D. degree in computer science from the University of Udine, Udine, Italy, in 2007.

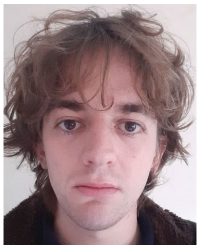
Previously, he was a Guest Professor of modeling and simulation with Saarland University, Saarbrücken, Germany, and a Visiting Researcher with the School of Informatics, University of Edinburgh, Edinburgh, U.K. For some years, he was also an Associate Researcher with CNR-ISTI, Pisa, Italy, within the QUANTICOL FP7 EU Project. He is currently a Full Professor of computer science with the University of Trieste, Trieste, where he leads the AI Laboratory. His research interests are within the large realm of artificial intelligence, and lie at the boundary between symbolic and formal methods in computer science, statistical machine learning, and modeling, simulation, and control. He is further interested in cyber-physical systems, collective adaptive systems, explainable artificial intelligence, and a broad spectrum of applications in medicine, insurance, industry, sustainability, and climate change.

Ginevra Carbone received the bachelor's degree in mathematics from the University of Udine, Udine, Italy, and the master's degree in data science and scientific computing and the Ph.D. degree in computer science from the University of Trieste, Trieste, Italy, in 2019 and 2023, respectively.

She is currently a Senior Machine Learning Scientist with Aindo SpA, Trieste. Her expertise includes adversarial robustness, explainability, Bayesian learning, and generative models.

Luca Laurenti received the master's degree in 2014, and the Ph.D. degree from the Department of Computer Science, University of Oxford, Oxford, U.K.

He was a member of the Trinity College, University of Oxford. He is currently a Tenure Track Assistant Professor with Delft Center for Systems and Control, TU Delft University, Delft, The Netherlands, and the Co-Director of the HERALD Delft AI Laboratory. He has a background in stochastic systems, control theory, formal methods, and artificial intelligence. His research work focuses on developing data-driven systems provably robust to interactions with a dynamic and uncertain world.



Andrea Patane received the master's degree in 2016, and the Ph.D. degree in computer science from the University of Oxford, Oxford, U.K., in 2020, under the Autonomous and Intelligent Machine and Systems Doctoral Program.

His Ph.D. was funded by the European Horizon 2020 Marie Skłodowska-Curie Fellowship. He is currently an Assistant Professor of system and software discipline with the School of Computer Science and Statistics, Trinity College, Dublin, Ireland. His primary research interests include formal analysis and verification of deep learning, in particular under Bayesian learning settings; fairness and biases in machine learning; stability and optimization of deep neural networks in adversarial settings; and optimization techniques for formal analyses.

Guido Sanguinetti received the degree in physics from the University of Genoa, Genoa, Italy, in 1998, and the Ph.D. degree in mathematics from the University of Oxford, Oxford, U.K., in 2002.

He is currently a Professor of applied physics with the International School for Advanced Studies, Trieste, Italy, and a Honorary Professor with the School of Informatics, University of Edinburgh, Edinburgh, U.K. He has authored over 100 research articles.

Dr. Sanguinetti was a recipient of an ERC starting grant as well as several national grants.

Matthew Wicker received the Ph.D. degree in computer science from the University of Oxford, Oxford, U.K., in 2021.

He was a Google DeepMind Scholar with the University of Oxford. He is currently a Lecturer (Assistant Professor) with the Department of Computing, Imperial College London, London, U.K., and also a Researcher with The Alan Turing Institute, London. His research focuses on trustworthy machine learning.