Imperial College London

MEng Individual Project

Department of Computing

Imperial College of Science, Technology and Medicine

# Robustness-Privacy Trade-Off In Bayesian Neural Networks

*Author:*
Mihnea Ghitu

*Supervisor:*
Dr Matthew Wicker

*Second Marker:*
Prof. William Knottenbelt

July 8, 2024

## Abstract

*Substantial developments have recently been made to devise provable methods that ensure the trustworthiness of deep neural networks. Most of these pieces of work study properties that constitute the trustworthy aspect individually, often isolating factors such as safety, transparency and fairness. Additionally, the majority focus on deterministic techniques, with only a few examining probabilistic ones. This work takes a holistic approach to the AI trustworthiness problem, by exploring the connections and trade-offs between robustness, privacy, uncertainty and performance through the lens of Bayesian learning. By studying a novel adaptation of the Hamiltonian Monte Carlo inference technique, we show empirically that it is possible to achieve adversarial robustness, differential privacy, accurate uncertainty estimation and good performance. We examine each of these desirable properties in isolation, quantify their contribution and clearly show how they interact with each other, presenting the effect they have on the overall system and their trade-offs, while underlining their crucial importance in practice.*

**Acknowledgements**

I am beyond grateful to my supervisor, Dr Matthew Wicker, for his invaluable guidance, support, and mentorship throughout this project. His insight and knowledge have been instrumental in shaping this work.

I would also like to thank my parents, Stelian and Mirela, for their unwavering support throughout my life and for instilling in me the values that drive me to strive for excellence.

# Contents

# Chapter 1

# Introduction

## 1.1 Motivation

One of the most crucial requirements for deep learning methods in real-life applications is trustworthiness. While these methods are ubiquitously used in practice, their utility is secondary to their reliability; without trust, a tool that could benefit society instead becomes harmful. Three particularly significant facets that we study in this work are robustness, privacy and uncertainty. The first, robustness, describes the ability of an artificial intelligence system to maintain its functionality and respond appropriately in the face of ill-suited adversaries that arise accidentally or in malicious circumstances. The second, privacy, is concerned with protecting these systems from attackers who want to obtain information regarding confidential data. The latter, uncertainty, ensures informative responses when encountering previously unseen data and the ability to make a clear distinction between known and unknown inputs when generating predictions.

Robustness has been widely studied by the machine learning community and the process of making a system robust consists of generating adversarial examples from already existing data [1] and using this data to train a safe model [2]. This is especially important for users, as they receive formal (mathematical) guarantees regarding the behaviour of a deep learning model depending on the input being fed into it. A well-known adversarial example found in [1] is that of an image showing a panda, which is being fed into a classifier. After adding a small amount of noise to the image, while there is no discernible difference to the human eye, a Convolutional Neural Network (CNN) classifies it as a gibbon with *99.3%* accuracy. Such an effect is only amplified when a model is used in a critical industry, such as healthcare.

Privacy is another essential aspect of trustworthiness and it involves safeguarding sensitive information within the data used to train and operate machine learning models. Privacy preservation is a necessary property for deep learning models deployed in the real world as it prevents data leakage, which can be catastrophic when, for example, it contains private medical records or information protected by legal policies. To defend against such attacks [3], a method called differential privacy

(DP) [4] has been proposed and successfully used in deep learning [5] to offer strong privacy guarantees.

Uncertainty in deep learning has gathered significant attention from the research community [6, 7], focusing on the development of methods to quantify and manage it in model predictions. This is crucial in concrete, real-life scenarios, as it enables users to assess the reliability of the model's decisions and understand the confidence associated with each prediction. The importance of managing uncertainty becomes even more pronounced in critical applications like medical diagnosis, where a misclassified image could lead to incorrect treatment. By incorporating uncertainty estimation in their systems, practitioners can identify and mitigate risks, ensuring that deep learning models make safer and more reliable decisions.

Although very desirable simultaneously, robustness and privacy guarantees have been shown to be at odds in Deterministic (deep) Neural Networks (DNNs) [8] due to the way they modify their responses given a certain input. Additionally, deterministic techniques poorly quantify uncertainty on out-of-distribution tasks, due to the fact that they offer point estimate predictions, which are unable to reason well on different tasks. This result calls for further investigation of different approaches to deep learning that can offer all guarantees at the same time.

This is where a probabilistic approach to deep learning comes into play, more specifically Bayesian Neural Networks (BNNs). In the Bayesian approach to neural network learning, a probabilistic model is introduced to represent our beliefs regarding the likelihood of different parameters (i.e. weights and biases) given the data, and the goal is to find the predictive distribution which not only minimizes the learning error, but also generalizes well on unseen data. One other essential property that these models exhibit is the ability to quantify the uncertainty of predictions, property which is extremely important in industries where a single decision can have a significant impact, such as transport and finance. Multiple methods have been developed to learn from these probabilistic models [6, 9] and to establish equivalences with deterministic neural networks [7].

# Chapter 2

# Background

## 2.1 Failings of Deterministic Neural Networks

Even though deterministic neural networks provide a powerful framework that is widely used in machine learning for a variety of tasks (ranging from classification to synthetic datasets generation), they have inherent faults and flaws that need to be remediated. Their usual workflow of producing fixed predictions given a certain input lacks the capacity to express the level of confidence or variability associated with those predictions (i.e. quantify uncertainty). This limitation becomes particularly pronounced in scenarios where the data is noisy, ambiguous, or when there is inherent uncertainty in the underlying processes.

Furthermore, deterministic neural networks often do not perform well when handling input variations and perturbations. This can be attributed to multiple factors, such as: lack of explicit uncertainty modelling, the distribution of training data in the input space, overfitting on training data or failure to capture inherent variability and noise. Thus, the same model can provide vastly different outputs for slightly modified inputs, making them sensitive to small changes, as it has been shown in [1]. In domains where robustness is critical, such as medical diagnosis or autonomous systems, deterministic models may fail in unforeseen circumstances and make life-threatening decisions.

Another notable drawback is the fact that the current principled methods for handling model complexity (L1 or L2 regularization, dropout etc.) have limited capacity for modelling uncertainty, can run into the risk of underfitting and are sensitive to the chosen dataset, thus performing poorly in environments where the data is limited, sparse or noisy. On the flip side, deterministic models which are trained on large amounts of data are prone to overfitting, which results in poor generalization performance on new data.

Moreover, neural networks often struggle with capturing complex distributions as their inherent behaviour is to try and find one single minima of the loss function. Thus, deterministic models are not well suited for scenarios where the same input can correspond to multiple plausible outputs because they fail to capture all the function space landscape and will converge to the closest minima, subject to hyper-parameters values. This limitation is especially significant in applications such as natural language processing and image generation, where multiple valid interpretations may exist for a given input.

In addressing these failings, researchers have turned towards stochastic approaches, such as Bayesian neural networks. By introducing uncertainty into the model through probabilistic representations, these methods have the ability to model more faithfully uncertain and dynamic real-world phenomena. Bayesian neural networks, for instance, can provide not only accurate predictions on unseen data but also probabilistic (multimodal) distributions over possible outcomes, enabling more informed decision-making and handling uncertainty in a principled manner.

## 2.2 Trustworthiness

As mentioned in section 1.1, the three aspects of trustworthiness, namely *(adversarial) robustness*, *privacy* and *uncertainty*, are crucial properties machine learning systems need to possess in order to be successfully deployed in real-life scenarios. As such, significant efforts have been invested into developing the necessary theoretical basis to achieve these properties.

Firstly, to overcome the lack of uncertainty quantification of deterministic neural networks, Bayesian approaches such as Hamiltonian Monte Carlo (**HMC**) [9], Variational Inference (**VI**) [10] or Bayes by Backprop (**BBP**) [6] have been used to compute the true posterior distribution of a neural network's parameters. Apart from the advantage of better generalization outside the training samples and these new frameworks and training methods provide the necessary tools to detect out-of-distribution data, and by doing so, they massively outperform deterministic techniques on all metrics.

Then, the aforementioned probabilistic approaches to machine learning have been equipped with certifiable guarantees for robustness against numerous types of adversaries [11], by leveraging existing work on robust deterministic techniques [1, 2, 12] and adapting it to the probabilistic setting. This is an important result the trustworthiness research community produced, not only because it provides resilience to accidental or malignant perturbations of input data at inference time, but also because it manages to achieve at once two out of the three desirable properties a real-world intelligent solution requires.

Similarly, the privacy trait has not been neglected either and building upon work on statistical databases [4], a measure named *Differential Privacy* (**DP**) has been introduced in the standard deep learning context [5], as well as in the probabilistic one [13, 14] to quantify the loss of information during the learning process. This ensures the fact that now practitioners can be confident that the datasets used in the training process is protected and that privacy and uncertainty are simultaneously achievable using Bayesian neural networks.

Studying these properties in isolation naturally leads us to formulate an essential question: *Is it possible to achieve robustness, privacy and uncertainty* ***all at once****?*. This work attempts to unify these features by introducing a novel training procedure and demonstrating its effectiveness through carefully designed experiments on a variety of datasets.

# Chapter 3

# Preliminaries

## 3.1 Deterministic Neural Networks

The framework used by neural networks to learn how to solve a task through pattern recognition is widely known and used in practice. In 1989, G. Cybenko showed that finite linear combinations of composition univariate function and a set of non-linear activations can uniformly approximate any continuous function with support in the unit hypercube [15]. This essentially states that neural networks are universal function approximators, hence why they have gained in popularity with the increase of computational power. The modern formulation of such structures assumes the existence of an input set $\mathcal{X}$, a number $K$ of layers, a set of network weights $\mathcal{W}_k$ and biases $b_k$, a set of activation functions $\sigma_k$, a loss function $\mathcal{L}(\cdot,\cdot)$ and because we will be focusing solely on supervised learning (that is, learning from labeled data), a label set $\mathcal{Y}$. Mathematically, given layer $Z_1 = \mathcal{X}$, then for each layer $Z_k$ with $k \in \{2, 3, ...K-1\}$ and output $\hat{\mathcal{Y}}$, we have:

$$Z_k = \sigma(\mathcal{W}_{k-1} Z_{k-1} + b_{k-1}) \text{ and } \hat{\mathcal{Y}} = \sigma(\mathcal{W}_K Z_{K-1} + b_{K-1})$$

By computing the loss $\mathcal{L}(\hat{\mathcal{Y}}, \mathcal{Y})$ and performing gradient descent on the network parameters (that is, the weights and biases) we can optimise the neural network to learn any suitable task. The gradient descent update phase can be written as follows for the parameters of a layer (where $\alpha$ is the learning rate):

$$\mathcal{W}_k \leftarrow \mathcal{W}_k - \alpha \frac{\partial \mathcal{L}(\hat{\mathcal{Y}}, \mathcal{Y})}{\partial \mathcal{W}_k} \text{ and } b_k \leftarrow b_k - \alpha \frac{\partial \mathcal{L}(\hat{\mathcal{Y}}, \mathcal{Y})}{\partial b_k}$$

## 3.2 Bayesian Learning

In the quest to develop better methods to learn from data, scientists have turned to probability theory and integrated it with the neural network framework to provide better predictions and, in contrast to previous techniques, more flexible models and most importantly, uncertainty quantification. As the name suggests, the battery of existing techniques use at their core the essential *Bayes Theorem* to adjust prior knowledge about a probabilistic model given new information inferred from data and provide a new, more accurate model. The theorem states that for a probability measure $\mu$ and two events $A$ and $B$, the conditional probability of an event is based on prior knowledge of conditions related to the event:

$$\mu(A|B) = \frac{\mu(B|A)\mu(A)}{\mu(B)}$$

Therefore, in general, the objective in Bayesian learning is to find or approximate the posterior distribution of a network's parameters ($\theta$), given a dataset $D$, a prior $p(\theta)$ and a likelihood probability distribution $p(D|\theta)$, where the latter is strongly tied to the loss function of a neural network:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}$$

Because the model architecture is usually fixed and model selection is not included in the optimisation process, then $p(D) = 1$ and can thus be omitted in the above equation.

### 3.2.1 Approximate inference

Approximate inference is a computational approach used in Bayesian learning to estimate quantities of interest when exact inference is computationally intractable or too expensive. Multiple approximate inference methods have been proposed to tackle hard problems, and some of the best-working approaches will be presented below.

1. **Variational Bayes**: These type of methods, such as [10] or [6], aim to find an analytic variational approximation to the posterior probability distribution we are interested in finding. This is achieved by minimizing a measure that quantifies the loss of information when encoding when using a different data distribution rather than the true one. This measure is called KL (Kullback-Leibler) divergence and is defined as:

$$\mathbf{KL}(P \mid\mid Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx$$

where $p$ and $q$ are the probability density function of the distributions $P$ and $Q$. In general, in Variational Bayes, we are interested in finding the posterior distribution $p(\boldsymbol{\omega}|D)$ of the weights and biases $\boldsymbol{\omega} = (\mathbf{w}, \mathbf{b})$ of a neural network given a dataset $D$. To this end, we try to approximate it using a distribution belonging to a given parametric family $q(\boldsymbol{\omega}|\theta)$, by minimizing the KL divergence:

$$\begin{aligned}\theta^* &= \arg\min_{\theta} \mathbf{KL}(q(\boldsymbol{\omega}|\theta) \mid\mid p(\boldsymbol{\omega}|D)) \\ &= \arg\min_{\theta} \mathbf{KL}(q(\boldsymbol{\omega}|\theta) \mid\mid p(\boldsymbol{\omega})) - \mathbb{E}_{q(\boldsymbol{\omega}|\theta)}[log(p(D|\boldsymbol{\omega}))]\end{aligned}$$

This objective function is a lower bound on the marginal likelihood $p(D)$ and expressing variational inference in this manner allows us to use well-known optimisation techniques in the parameter space, such as stochastic gradient descent (SGD) on the components of $\theta$.

2. **Markov Chain Monte Carlo (MCMC)**: First introduced by [16] and later generalized by [17], the Metropolis-Hastings algorithm uses Monte-Carlo sampling to randomly walk in the distribution space and generate a sequence of samples from a probability distribution. The sequence of samples represents an ergodic Markov chain with the invariant distribution being the desired one.

   Building on top of that, [9] uses Hamiltonian dynamics to define the desired probability distribution in terms of an energy function and sample from it in a smart way, by using the gradient of the potential energy to move towards the high probability zones and thus avoiding the random walk behaviour in the process. This new algorithm, named Hamiltonian Monte Carlo is now used to find the posterior distribution of a neural network's parameters and is one of the *de facto* Bayesian inference techniques used in practice, due to its modelling flexibility (it can approximate arbitrarily complex probability distributions). While one can argue that not having a closed-form of the density function for the posterior is a disadvantage, practically speaking, HMC is consistently significantly more accurate than techniques that rely on making assumptions regarding the shapes of distributions or the family they belong to, and thus generalizes much better at inference time.

   It is worth thoroughly defining HMC mathematically, as it will be the main method used to find the Bayesian networks' distributions in chapters 4 and 5. This technique is inspired from classical mechanics, where the total energy of a dynamical system, named *Hamiltonian*, is expressed as a function of the coordinates in space $q$ and momentum $p$ of said system. Thus, given the Hamiltonian:

$$\mathcal{H}(p,q) = \mathcal{K}(p) + \mathcal{U}(q)$$

   where $\mathcal{K}(\cdot)$ and $\mathcal{U}(\cdot)$ denote the kinetic and respectively, the potential energy, then the equations of motion are:

$$\frac{dq_i}{dt} = \frac{\partial \mathcal{H}}{dp_i} \qquad\qquad \frac{dp_i}{dt} = -\frac{\partial \mathcal{H}}{dq_i}$$

   This is extended to a probabilistic setting by considering the potential energy to be the true posterior distribution of our parameters of interest, and the kinetic energy to be a zero-mean Gaussian corresponding to the momenta of the system. The latter is one of the significant improvements HMC exhibits over random walk algorithms, as it is able to traverse the parameter space in a smart, informed manner. In Bayesian statistics, where the posterior is the quantity of interest, the two energy quantities are defined mathematically as:

$$\mathcal{U}(q) = -log[\pi(q|D)] = -log[\pi(D|q)\pi(q)]$$

$$\mathcal{K}(p) = p^T M^{-1} p/2$$

In order to simulate the evolution of the system, the motion is discretized using the leapfrog (integration) method:

$$p_i(t + \frac{\epsilon}{2}) = p_i(t) - \frac{\epsilon}{2}\frac{\partial \mathcal{U}}{\partial q_i} q(t)$$

$$q_i(t + \epsilon) = q_i(t) - \epsilon \frac{p_i(t + \frac{\epsilon}{2})}{m_i}$$

$$p_i(t + \epsilon) = p_i(t + \frac{\epsilon}{2}) - \frac{\epsilon}{2}\frac{\partial \mathcal{U}}{\partial q_i} q(t)$$

The last step of this process is called the Metropolis-Hasting acceptance step and involves computing the difference between energy quantities at the beginning and end of the simulation and decide probabilistically whether such a simulation is a realistic one. Given the Hamiltonian energy function and the canonical distribution over states with probability density:

$$P(q,p) = \frac{1}{Z} exp(\frac{-\mathcal{H}(q,p)}{T})$$

the probability of a leapfrog simulation step knowing the initial states $q$ and $p$ and the proposed states $q^*$ and $p^*$:

$$min[1, exp(-\mathcal{H}(q^*, p^*) + \mathcal{H}(q,p)]$$

**Algorithm 1** Hamiltonian Monte Carlo algorithm

```
function HMC(U, grad_U, epsilon, L, current_q)
    q = current_q
    # independent standard normal sample
    p = rnorm(length(q), 0, 1)
    # Make a half step for momentum at the beginning
    current_p = p - epsilon * grad_U(q) / 2
    # Alternate full steps for position and momentum
    for i in 1:L do
        # Make a full step for the position
        q = q + epsilon * p
        # Make a full step for the momentum, except at the end of trajectory
        if i $\neq$ L then
```

```
            p = p - epsilon * grad_U(q)
        end if
    end for
    # Make a half step for momentum at the end
    current_p = p - epsilon * grad_U(q) / 2
    # Negate the momentum at end of trajectory to make the proposal symmetric
    p = -p
    # Evaluate potential and kinetic energy at start and end of trajectory
    current_U = U(current_q)
    current_K = sum(current_p ^ 2) / 2
    proposed_U = U(q)
    proposed_K = U(p ^ 2) / 2
    # Accept or reject the state at the end of trajectory, returning either the position
    # at the end of the trajectory or the initial position
    if 1 < exp(current_U - proposed_U + current_K - proposed_K) then
        return q # Accept proposal
    else
        return current_q # Reject proposal
    end if
end function
```

After a large number of iterations, using this method, we can numerically compute the posterior distribution of our quantity of interest through the posterior samples gathered in the process.

The pseudocode for HMC can be found above in algorithm 1. Our exposition on HMC in this section has been closely derived from [9] (the algorithm being reproduced verbatim), and we will build on this framework in section 4.3 with novel modification.

## 3.3 Adversarial Robustness

Adversarial robustness is a critical aspect of neural network security, aiming to enhance the resilience of models against adversarial attacks. Adversarial attacks involve the intentional manipulation of input data to mislead a neural network's predictions. This is achieved by adding carefully chosen perturbations to the input data, often imperceptible to the human eye, which causes the model to make incorrect predictions that can potentially have severe consequences when used in situations with high societal impact.

### 3.3.1 Attacks

Various attack methods have been developed to exploit vulnerabilities in neural networks, highlighting the need for robust defenses. Common adversarial attack strategies include:

1. **Fast Gradient Sign Method (FGSM)**: This attack is detailed in [1] and involves perturbing input features by a small amount in the direction of the gradient of the loss with respect to the input. FGSM is the quickest to generate and most effective adversarial attack, which makes it a perfect candidate for incorporation into the training procedure in order to gain robustness guarantees. Formally, for a neural network model with parameters $\boldsymbol{\omega}$, one input data point $x$, true label $y$ and loss function $J(\boldsymbol{\omega}, x, y)$, an adversarial example can be generated by perturbing the input x by:

$$\eta = \epsilon \text{sign}(\nabla_x J(\boldsymbol{\omega}, \mathbf{x}, y))$$

2. **Projected Gradient Descent (PGD)**: Described in [12], PGD is an iterative variant of FGSM, where perturbations are applied successively over multiple iterations. This attack aims to find the best adversarial example by refining the perturbations over a fixed amount of iterations by moving in the direction of the gradient of the loss function (i.e. towards where the loss is greates) and then projecting the result onto the possible set, which is a constrained $\epsilon$-ball around a data point. The best possible attack at iteration $t + 1$ is defined as:

$$x^{t+1} = \Pi_S(x^t + \epsilon \text{sign}(\nabla_x J(\mathbf{x}, y))$$

where the constraint set $S$ is :

$$S = [x^t - \epsilon, x^t + \epsilon]$$

3. **Transfer Attacks**: This is a type of black-box attack formalized in [18]. In this framework, adversarial examples generated for one model can often mislead other models, even if they have used the same machine learning technique (*intra-technique transferability*) or using a different one(*cross-technique transferability*). It can be observed that these type of attacks generalize across a wide-range of models and thus represent a serious threat.

In response to the development of adversarial attacks techniques, the machine learning community has focused on finding ways to train robust models, by incorporating a mix of clean and adversarial examples in the training process to improve resilience. Additionally, these newly developed methods also have the advantage of providing formal mathematical guarantees regarding the robustness properties of a certain model. This embodies the concept of certification, which will be detailed in the next part.

### 3.3.2 Certification

Certification of neural networks for adversarial robustness involves establishing formal guarantees regarding the model's behaviour under various types of adversarial attacks. Formal robustness certification often involves defining a perturbation region (or budget) around each data point within which the aim is to have the machine learning model make the same predictions, irrespective of the perturbation. Therefore, when deploying neural networks in safety-critical applications where the consequences of missclassification or missprediction can be severe, robust certification techniques offer a level of confidence that a certain model will behave in a stable, expected manner inside the verified regions.

Some of the most recognized certification methods include:

1. **Interval Bound Propagation (IBP)**: This method described in [2] involves minimizing the upper bound of the worst-case prediction of a neural network under $l_\infty$ norm adversarial perturbations. This means propagating a bounding box (polygon) computed using through the network's layers to create a verified output range during the forward pass and thus provides a certified guarantee that the model's predictions remains within these bounds for any perturbation within the initially defined $\epsilon$-ball. In practice, this means that for every input data point $x$, the upper $\bar{z}_0 = x + \epsilon\mathbf{1}$ and lower $\underline{z}_0 = x - \epsilon\mathbf{1}$ bounds perturbations are computed and then propagated through the network. Solving the optimisation problems that aim to find the worst case bounds after propagation through an affine layer yield the following solution, where $W$ and $b$ are the weights and biases of a layer and $k$ represents the layer's index in a network.

   $$\mu_{k-1} = \frac{\bar{z}_{k-1} + \underline{z}_{k-1}}{2} \qquad r_{k-1} = \frac{\bar{z}_{k-1} - \underline{z}_{k-1}}{2}$$

   $$\mu_k = W\mu_{k-1} + b \qquad r_k = |W| r_{k-1}$$

   $$\underline{z}_k = \mu_k - r_k \qquad \bar{z}_k = \mu_k + r_k$$

   The lower and upper bounds preservation is ensured by the monotonicity property of the nonlinear activation functions (denoted $h_k(\cdot)$). Thus, it is easy to see that:

   $$\underline{z}_k = h_k(\underline{z}_{k-1}) \qquad \bar{z}_k = h_k(\bar{z}_{k-1})$$

   Lastly, in the context of classification under adversarial perturbation, which we will focus on, we can calculate the worst-case logits using the aforementioned bounds in the following way:

   $$\hat{z}_{K,y} = \begin{cases} \underline{z}_{K,y}(\epsilon) & \text{if } y = y_{true} \\ \bar{z}_{k,y}(\epsilon) & \text{otherwise} \end{cases}$$

   This function definition ensures that every label every incorrect label is associated to the upper bound worst case, and the correct label to the lower bound. Optimising against this essentially means quantifying how much we can perturb the input without the upper bound confidence exceeding the lower bound confidence, which would result in an incorrect prediction.

2. **Mixed-Integer Linear Programming (MILP)**: MILP-based approaches such as [19] formulate robustness certification as a Mixed Integer Linear Programming problem, where the objective is to maximize the size of a verified region around each data point. MILP solvers are then used to find the optimal solution.

3. **Randomized Smoothing**: This complete certification method formalized in [20] involves perturbing input data with random Gaussian noise, observing

the model's predictions and then providing a robustness guarantee based on the variance of the noise and the probability of classification of the two most probable classes.

Adversarial robustness certification thus plays a crucial role in enhancing the trustworthiness of neural networks, particularly in safety-critical domains such as autonomous vehicles, healthcare, and finance. As research advances in this field, more generic and faster methods are being developed, which generalize for multiple types of attacks and hence make large-scale, real-world tasks easier to reason about in terms of resilience.

### 3.3.3 Robust Training

Since both attacks and certification methods are successful in finding malignant (adversarial) examples of perturbed data points that can trick carefully-designed intelligent systems' predictions, we are interested in designing procedures that defend against these types of out-of-distribution inputs. It turns out that using the attacks themselves as defense mechanisms allows us to prevent failures at inference time.

In practice, this means dynamically augmenting our datasets during training, computing the loss with respect to both our initial and perturbed inputs and optimising both at the same time. Mathematically, if $\hat{y}$ and $\hat{y}_{adv}$ are the logits associated to the initial and respectively, the perturbed (adversarial) inputs, then, given the criterion $\ell(\cdot, \cdot)$ used in normal training (e.g. cross entropy for classification problems) and the hyperparameter $\alpha$, which controls the trade-off between learning from the standard and adversarial examples, the final loss function is:

$$\mathcal{L}(y_{true}, \hat{y}, \hat{y}_{adv}) = \alpha \cdot \ell(\hat{y}, y_{true}) + (1 - \alpha) \cdot \ell(\hat{y}_{adv}, y_{true})$$

The aforementioned logits associated with the adversarial examples are computed by doing a forward pass in the case of attacks such as FGSM and PGD, whereas in the case of IBP the procedure is slightly more involved, but described in detail in section 3.3.2. After computing the loss defined above, standard back-propagation is performed on the network's parameters to optimise predictions.

Lastly, it is worth mentioning the *certifiable robustness* of a neural network with parameters $\theta$ defined by the function $f^{\theta} : \mathbb{R}^n \rightarrow \mathbb{R}^C$, where $C$ is the number of classes and $\sigma(\cdot)$ is the softmax function, is defined by:

$$\hat{\mathcal{R}}_{\epsilon}(\theta, D) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}[\forall \bar{x} s.t. |x_i^* - \bar{x}| \leq \epsilon, \ \underset{c \in \{1, \ldots, C\}}{\arg\max}(f^{\theta}(\bar{x}) = c_i^*)]$$

For every fixed $\epsilon$, the value obtained from the above equation at inference time for an IBP-trained network provides a *worst-case lower bound* of the robustness, guarantee which is extremely valuable for trustworthy systems deployed in critical scenarios, such as medicine or finance.

## 3.4 Privacy

### 3.4.1 Differential Privacy

Stemming from work on privacy-preserving statistical databases, differential privacy is a technique that offers statistical guarantees for the anonymity of individuals, while allowing extraction of useful insights from data. More specifically, a randomized mechanism $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies $(\epsilon - \delta)$-differential privacy if for any two adjacent inputs $d, d' \in \mathcal{D}$ and for any subset of outputs $S \subseteq R$ it holds that:

$$Pr[\mathcal{M}(d) \subseteq S] = \epsilon Pr[\mathcal{M}(d') \subseteq S] + \delta$$

The $\epsilon$ parameter controls the privacy guarantee (a smaller $\epsilon$ providing stronger privacy), while $\delta$ allows for a small probability of failure of said guarantee. This definition exhibits multiple extremely convenient properties, especially regarding the composition of mechanisms. Practically, an algorithm that satisfies $(\epsilon - \delta)$-DP is guaranteed to be able defend against a large variety of attacks, such as membership [3] or property [21] inference attacks.

### 3.4.2 Mechanisms of Differential Privacy

There are multiple mechanisms used to achieve differential privacy. To understand how they work, we need to first define a property of the function of interest, called *sensitivity*. Given a function $f : \mathcal{D} \rightarrow \mathbb{R}^k$, where $\mathcal{D}$ is the domain of the dataset and $k$ is the dimension of the output, the $l_1$-sensitivity $\Delta_1 f$ of a function $f$ is the maximum difference in the output of $f$ when applied to any two adjacent datasets $d$ and $d'$:

$$\Delta_1 f = \max_{d,d' \subset \mathcal{D}} ||f(d) - f(d')||_1$$

The $l_2$-sensitivity is defined similarly, but instead uses the $l_2$ norm:

$$\Delta_2 f = \max_{d,d' \subset \mathcal{D}} ||f(d) - f(d')||_2$$

Now we are prepared to introduce the most well-known DP mechanisms:

1. **The Laplace Mechanism**:
   The Laplace mechanism adds Laplace noise to the output of $f$. Let b be the scale parameter of the Laplace distribution, then we can write the probability density function of the Laplace distribution as: Given a dataset $\mathcal{D}$, the Laplace mechanism $\mathcal{A}$ is defined as:

   $$\mathcal{A}(\mathcal{D}) = f(\mathcal{D}) + p_{Laplace}(0|b)$$

   To make the mechanism $\mathcal{A}$ satisfy $(\epsilon, 0)$-differential privacy, all we need to do is set the scale parameter $b$ of the Laplace distribution to be large enough such that the noise makes the data anonymous:

   $$b = \frac{\Delta_1 f}{\epsilon}$$

The meaning of $\delta$ is that there is a small probability of accidental information leakage over multiple runs of a mechanism (i.e. the mechanism is $\epsilon$-DP with probability $1-\delta$). This approach is powerful because it allows for setting an arbitrarily small privacy loss that will be maintained every time the algorithm is applied.

2. **The Gaussian Mechanism**:
The Gaussian mechanism works similarly to the Laplace mechanism and its cumulative effect over many computations provides comparable privacy guarantees, by adding a Gaussian noise sampled from $\mathcal{N}(0, \sigma^2)$, where in general the density function of a Gaussian with mean $\mu$ and variance $\sigma$ is:

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Thus, the Gaussian mechanism $\mathcal{M}$ defined as:

$$\mathcal{M}(\mathcal{D}) = f(\mathcal{D}) + \mathcal{N}(0, \sigma^2)$$

is $(\epsilon, \delta)$-differentially private for an arbitrary $\epsilon \in (0,1)$ if $\sigma \geq \frac{2ln(\frac{1.25}{\delta})\Delta_2 f}{\epsilon}$, as proven in [22].

### 3.4.3 DP–SGD

Taking a small leap from the work on differential privacy, it is easy to notice its application in the field of deep learning, because of a common key component of any such system, the *dataset*, which acts as a (albeit small) database. It turns out that from of the probability distributions of the data, as well as from the optimisation technique of backpropagation, an attacker can gain knowledge about the data used in training by making associations between the magnitude of the gradient and a data point. One well-known attack of this type is called a *membership inference attack* [3].

Fortunately, since at the heart of the whole neural network's paradigm resides the sequential composition of functions, the deeep learning framework lends itself to application of the sequential composition theorem [22], which states that the composition of a sequence of $k$ $(\epsilon, \delta)$-DP algorithms $\mathcal{M}_k$ is $(k\epsilon, k\delta)$-DP.

Thus, thinking of a neural network as a sequence of mechanisms, [5] introduces a training technique that provides DP guarantees, as well as a *moments accountant*, which is a method that computes tight bounds for the value of $\epsilon$ and $\delta$. The training technique consists of clipping the gradient of the loss function for each data point in a batch to a maximum bound $b_g$ at every backpropagation step:

$$\hat{g}_x = clip_{b_g}(\nabla_\theta \mathcal{L}(x, \theta)) \ \ \forall x \in \mathcal{B}_i$$

to bound the $l_2$ sensitivity of the function for every element in a batch $\mathcal{B}_i$. The average value of the gradient is then computed, and a proportional amount of noise satisfying the $\sigma$ inequality in section 2 is added to ensure the DP guarantees of the data, before stepping in the opposite direction of the average noisy gradient:

$$\tilde{g}_i = \frac{1}{|\mathcal{B}_i|}(\sum_{x \in \mathcal{B}_i} \hat{g}_x + \mathcal{N}(0, \sigma^2))$$

In practice, this means that (i) the gradients are made more uninformative, so that an attacker is not able to infer easily that a certain data point heavily influences the decision boundary and that (ii) the general learning process is obfuscated through randomization of the optimisation stage (i.e. stepping toward the minima).

Lastly, it is worth mentioning that the privacy landscape has been further widened by extending the concept of differential privacy to probabilistic methods [13, 14], one of which will be a key part of section 4.2. Not only that, but it is possible to write the privacy guarantees of BNNs such as *Stochastic Gradient Langevin Dynamics* (SGLD), *Bayes by Backprop* and *Monte Carlo Dropout* (MC-Dropout) in terms of the DP-SGD guarantees. Once again, while previous work has only been concerned with the effect of introducing individual property guarantees in BNNs (i.e. private BNNs or adversarially robust BNNs), this thesis will study the relationship and overall impact of using mechanisms that ensure multiple aspects of trustworthiness concurrently.

# Chapter 4

# Methodology

In order to perform an original analysis at the intersection of privacy, robustness, and uncertainty, we establish a novel modification of the Hamiltonian Monte Carlo Bayesian inference technique that allows for the optimisation of each desirable property. Training a probabilistic machine learning model using this new method aims to show the connection between robustness, privacy and uncertainty in Bayesian neural networks by simultaneously achieving and and quantifying the guarantees for each of these properties of safe and trustworthy AI. In order to understand our methodology improvement, we will first present a number of methods which have been proved to attain the aforementioned security guarantees in theory, as well as in practice. Then, we will build upon these pieces of work and introduce our novel algorithm, which allows us to jointly learn adversarially and privately, all while retaining accurate uncertainty estimates. The presented methodology and experiments will be conducted using the Hamiltonian Monte Carlo probabilistic inference technique, described in detail in section 3.2.1, due to its well-known performance with respect to uncertainty and accuracy metrics.

## 4.1 Certifiable Bayesian Inference

Naturally, with the advantageous development of the new method of Bayesian learning, new attacks that target the predictive distribution given by a BNN at inference time arise, making it essential to ensure the robustness of such networks. To generate an adversarial data point that can be used to attack a BNN, one only needs to modify the attacks described in 3.3.1 by taking the expectation with respect to network's parameters distribution, which can be computed through Monte Carlo methods. For example, given a posterior probability distribution $p(\boldsymbol{\omega}|D)$ over the set of parameters of a BNN, the adversarial perturbation in FGSM, which is now directed at the predictive distribution, becomes:

$$\tilde{\eta} = \epsilon sign(\mathbb{E}_{p(\mathbf{w}|D)}[\nabla_x J(\boldsymbol{\omega}, x, y)]) \approx \epsilon sign(\sum_{i=1}^{n} \nabla_x J(\boldsymbol{\omega_i}, x_i, y_i))$$

In order to achieve certifiable adversarial robustness in BNNs, a novel method has been introduced in [11], which involves defining a new type of likelihood called robust likelihood, and then using it in conjunction with an Interval Bound Propagation (IBP) technique. The robust likelihood formula is achieved by first defining the minimizer of the softmax for class $y$, given a network with parameters $\boldsymbol{\omega}$ and an $\epsilon$-ball perturbation around a data point $x$:

$$\sigma_y(f_{min}^{\boldsymbol{\omega},\epsilon}(x)) = \min_{x':|x-x'|\leq\epsilon} \sigma_y(f_{min}^{\boldsymbol{\omega}}(x))$$

Then, for the likelihood $p(y|x, \mathbf{w})$ to also be influenced by the robustness of a BNN in the neighbourhood of a data point, $\epsilon$ is modelled as a random variable and the likelihood is expressed as the expectation of the softmax minimizer over the distribution given by $p_\epsilon(\epsilon)$:

$$\begin{aligned} p(y|x,\omega) &= \int_{\mathbb{R}_+} \sigma_y(f_{min}^{\boldsymbol{\omega},\epsilon}(x))p_\epsilon(\epsilon)d\epsilon \\ &= \mathbb{E}_{\epsilon\sim p_\epsilon}[\sigma_y(f_{min}^{\boldsymbol{\omega},\epsilon}(x))] \end{aligned}$$

Considering $0 \leq \lambda \leq 1$ and $\eta > 0$, having the probability density function for $\epsilon$ as:

$$p_\epsilon(\epsilon) = \begin{cases} \lambda & \text{if } \epsilon = 0 \\ 1-\lambda & \text{if } \epsilon = \eta \end{cases}$$

leads to the following form of the robust likelihood:

$$p(y|x,\boldsymbol{\omega}) = \lambda \cdot \sigma_y(f^{\boldsymbol{\omega}}(x)) + (1-\lambda) \cdot \sigma_y(f_{min}^{\eta,\boldsymbol{\omega}}(x))$$

This particular shape of the likelihood is essential, because it allows us to use the same framework defined in section 3.3.3 during the training of the Bayesian neural network to learn how to both predict the correct labels and be secure against adversarial perturbations.

The actual worst case logits (i.e. the softmax minimizer) are obtained using exactly the IBP method introduced in [2] and defined mathematically in section 3.3.2. This method is very powerful, because it provides a lower bound on the certified robustness of a BNN for any given perturbation radius, result that (i) is generic and can be applied to multiple instances of probabilistic inference and (ii) combines the advantageous properties of statistical learning with security guarantees, which is very useful in real-life applications.

## 4.2 DP–HMC

As it has been described in section 3.4.3, differential privacy mathematical guarantees can be formulated for both deterministic and probabilistic deep learning techniques. Naturally, this work has been extended to cover the HMC method, as presented in [14]. The main results of this paper reside in (i) recognising that the leapfrog integration techniques equipped with the DP Gaussian Mechanism are applied sequentially and thus fit the premise of the strong composition theorem and that (ii) the data leakage does not only happen when stepping in the opposite direction of the potential energy gradient but also during the Metropolis-Hasting acceptance step. An intuitive explanation for the latter relies on remembering the fact that the probability of accepting a sample proposal is proportional (and thus dependent) on the energy difference between the last step in the Markov chain and the proposed one. This means that an attacker can infer the relative positions of two consecutive Markov chain samples, inference which signifies information leakage of the data used in the learning process.

To address these shortcomings, firstly, the gradient is clipped and noise is added to the descent step, similarly to the equations of DP-SGD, described in section 3.4.3. Considering our network parameters to be $\theta$ and the current value of the momentum to be $p$, the descent step becomes:

$$p = q - \epsilon \left[ \nabla \ln p(\theta) + \left( \sum_{x \in \mathcal{X}} clip_{b_g}(\nabla \ln p(x|\theta)) \right) + \xi \right]$$

where $\xi$ is sampled from a normal distribution:

$$\xi \sim \mathcal{N}(0, \sigma_g^2) \ \text{ with } \ \sigma_g = 2 \cdot \tau_g \cdot b_g$$

In a similar manner, noise is added to the Metropolis-Hastings step in order to fool possible attackers. Because this operation changes the acceptance probability and can thus heavily influence the convergence properties, ergodicity and irreducibility of the Markov chain simulation, a correction proportional to the norm of the difference between the current $(\theta, p)$ and proposed sample $(\theta^*, p^*)$ is performed. Applying this DP-penalty algorithm results in the acceptance probability:

$$\alpha_{DP}(\theta, p, \theta^*, p^*) = \min \left[1, \mathcal{H}(\theta, p) - \mathcal{H}(\theta^*, p^*) + \xi - \sigma_l^2(\theta, \theta^*)\right]$$

where, again, $\xi$ is sampled from a normal distribution:

$$\xi \sim \mathcal{N}(0, \sigma_l^2(\theta, \theta^*)) \ \text{ with } \ \sigma_l(\theta, \theta^*) = 2 \cdot \tau_l \cdot b_l \cdot ||\theta - \theta^*||$$

Lastly, it is important to quantify the privacy guarantees this method offers. Given the number of leapfrog steps $L$ for each proposal and the total number of steps in the simulation (we can refer to this as *number of epochs* to make a link to the deep learning framework) $k$, for a user defined epsilon, the DP-HMC algorithm is $(\epsilon, \delta(\epsilon))$-DP with:

$$\delta(\epsilon) = \frac{1}{2} \left[ \operatorname{erfc} \left( \frac{\epsilon - \mu}{2\sqrt{\mu}} \right) - e^{\epsilon} \operatorname{erfc} \left( \frac{\epsilon + \mu}{2\sqrt{\mu}} \right) \right] \ \text{ with } \ \mu = \frac{k}{2\tau_l^2} + \frac{k(L+1)}{2\tau_g^2}$$

Using just these small modifications to the original HMC algorithm, we now have an essential tool at our disposal that allows us to achieve privacy guarantees without harming any of the properties of a Markov chain simulation, and we can thus be confident that this method can be successfully applied in the context of Bayesian inference for computing the posterior distribution of the network parameters.

## 4.3 ADV–DP–HMC

Having now all the tools described in sections 4.1 and 4.2, we are now at a point where we can define the ***ADV-DP-HMC*** algorithm, which allows us to learn the posterior distribution of a neural network's parameters (weights and biases) by using HMC to sample from the distribution, all while preserving adversarial robustness and privacy, and having the benefit of performant uncertainty quantification.

Before presenting the algorithm, we need to introduce some new notation, so that the pseudocode is succinct and easy to understand. For any input $\mathcal{X}$ and network $f^{\theta}(\cdot)$, we regard the adversarial worst case logits computed by IBP as described in section 3.3.2 to be $\mathcal{X}_{IBP} = f^{\theta}_{IBP}(\mathcal{X}) = \hat{z}_{\mathcal{K},y}$. We will also regard the likelihood of the same data, given parameters $\theta$ to be $p(\mathcal{X}|\theta) = \mathcal{L}(f^{\theta}(\mathcal{X}), y_{true})$, where $y_{true}$ is the baseline truth and $\mathcal{L}(\cdot)$ is the loss function (for example, in a multi-class classification setting, that would be *Cross Entropy Loss*). The notation $\text{clip}_b(\mathcal{X})$ signifies the bounding (or clipping) of each of the elements in $\mathcal{X}$ at maximum bound $b$, or, in mathematical terms: $\text{clip}_b(\mathcal{X}) = \mathcal{X} * \min\left\{1, \frac{b}{||\mathcal{X}||_2}\right\}$, where $*$ is element-wise multiplication, and the norm is taken separately $\forall x \in \mathcal{X}$. Lastly, when we refer to the gradient of a batch with respect to the network's parameters, we mean the average of the gradients with respect to the parameters taken across each element in that batch: $\nabla^{\mathcal{B}}_{\theta}(\cdot) = \frac{1}{|\mathcal{B}|}\sum_{\forall x \in \mathcal{B}} \nabla^{x}_{\theta}(\cdot)$.

**Algorithm 2** ADV-DP-HMC

```
    **Input** Neural network $f^{\theta}(\cdot)$, initial network parameters $\theta$, leapfrog step size $l_f$, leapfrog steps $L$, number of epochs $n_e$, momentum covariance matrix $M$, data $\mathcal{X}$, learning trade-off parameter $\alpha$, perturbation radius $\epsilon$, gradient clip bounds $b_g$ and $b_g$, sensitivity parameters $\tau_g$ and $\tau_l$
    **Output** Posterior samples $p_s$
    $\sigma_g = 2\tau_g b_g$  and  $\sigma_l(\theta, \theta^*) = 2\tau_l b_l ||\theta - \theta^*||_2$
  **function** ADV-DP-HMC
    $p_s$ = **empty list**
    **for** epoch **in** $n_e$ **do**
      **sample** p $\sim \mathcal{N}(0, M)$, batch $\mathcal{B} \sim \mathcal{X}$ and $\xi_g \sim \mathcal{N}(0, \sigma_g^2)$
      # Make a half step descent
      $p^* = p - \frac{l_f}{2} \cdot \left\{\text{clip}_{b_g}\left[\nabla^{\mathcal{B}}_{\theta}\left(p(\theta) \cdot [\alpha \cdot p(\mathcal{B}|\theta) + (1-\alpha) \cdot p(\mathcal{B}_{IBP}|\theta)]\right)\right] + \xi_g\right\}$
      $\theta^* = \theta$
      # Run Markov chain simulation
      **for** $\ell$ **in** $L$ **do**
        $\theta^* = \theta^* + l_f \cdot p^*$
        **sample** batch $\mathcal{B} \sim \mathcal{X}$ and $\xi_g \sim \mathcal{N}(0, \sigma_g^2)$
        $p^* = p^* - l_f \cdot \left\{\text{clip}_{b_g}\left[\nabla^{\mathcal{B}}_{\theta}\left(p(\theta) \cdot [\alpha \cdot p(\mathcal{B}|\theta) + (1-\alpha) \cdot p(\mathcal{B}_{IBP}|\theta)]\right)\right] + \xi_g\right\}$
      **end for**
      **sample** batch $\mathcal{B} \sim \mathcal{X}$ and $\xi_g \sim \mathcal{N}(0, \sigma_g^2)$
```

```
            # Make the last half step
            $p^* = p^* - \frac{l_f}{2} \cdot \left\{ \text{clip}_{b_g} \left[ \nabla_\theta^{\mathcal{B}} \left( p(\theta) \cdot [\alpha \cdot p(\mathcal{B}|\theta) + (1-\alpha) \cdot p(\mathcal{B}_{IBP}|\theta)] \right) \right] + \xi_g \right\}$
            # Ensure the Markov chain is reversible
            $p* = -p^*$
            # DP Metropolis-Hastings acceptance state
            sample $\xi \sim \mathcal{N}(0, \sigma_l^2(\theta, \theta^*))$
            $\Delta U = \text{clip}_{\sigma_l^2(\theta,\theta^*)} \ln \left[ \frac{p(\mathcal{X})|\theta^*)}{p(\mathcal{X})|\theta} \right] + \ln \frac{p(\theta^*)}{p(\theta)}$
            $\Delta K = p^T M^{-1} p - p^{*^T} M^{-1} p^*$
            $\Delta H = \Delta U + \Delta K + \xi_l - \frac{1}{2}\sigma_l^2(\theta, \theta^*)$
            sample $u \sim \mathcal{U}(0, 1)$
            if $u \leq \exp(\Delta H)$ then
                append $\theta^*$ to $p_s$
                $\theta = \theta^*$
            end if
        end for
        return $p_s$
    end function
```

It can easily be noticed that the novelty of Algorithm 2 lies in the complex gradient descent step, which combines the differential privacy mechanism and adversarial training loss function to simultaneously learn how to be robust and safe. Moreover, the whole theoretical structure of HMC is preserved (i.e. *momentum* and *position* variables, leapfrog integration and Metropolis-Hastings acceptance step), thus ensuring convergence. Lastly, the privatization of the acceptance step is adapted to our use case using the ideas in [14]. This algorithm, combined with the results obtained from the experiments in comparison to previously known results represent the main contributions of this thesis.

# Chapter 5

# Experiments

## 5.1 Hypotheses

In this research, we hypothesize that a Bayesian neural network (BNN) system, trained by simultaneously incorporating adversarial robustness techniques and differential privacy mechanisms, can perform efficiently in tasks pertaining to:

1. **Uncertainty Quantification**
2. **Privacy Preservation**
3. **Robustness Against Adversarial Attacks**

Specifically, we claim that integrating these methods in the process of building trustworthy AI systems will enable them to provide reliable uncertainty estimates for its predictions while safeguarding sensitive data and maintaining resilience against adversarial perturbations. By leveraging the probabilistic nature of BNNs, we expect the system to effectively balance these objectives, resulting in a model that not only converges to the true underlying data distribution but also adheres to privacy and robustness criteria dictated by the environment.

Furthermore, we hypothesize that the proposed HMC-trained systems will experimentally demonstrate convergence to true results despite the complexity of the data manifold introduced by the addition of the two new learning objectives. While traditional deterministic methods often face trade-offs when attempting to optimise for multiple objectives, we believe that our approach can overcome these challenges. In particular, the inherent uncertainty estimation capability of BNNs is expected to provide better generalization and avoid overfitting, while the advanced robustness and privacy mechanisms introduced in the training process adversarial attacks will ensure that the confidentiality of the training data is not compromised and that our model is resistant to perturbations. Through rigorous experimentation and evaluation, we aim to validate that our BNN system can achieve these goals concurrently, thereby advancing the state-of-the-art in trustworthy probabilistic machine learning.

## 5.2 Settings

In this section, we will provide a detailed introduction to the datasets selected for the conducted experiments. Furthermore, we will elaborate on the experimental setup, which includes the metrics utilized and the training methodologies that will be subjected to comparative analysis.

### 5.2.1 MNIST

The **MNIST** dataset [23] (shown in 5.1) is a well-established benchmark in the field of machine learning and computer vision, widely used for the evaluation and comparison of various algorithms. It consists of 70,000 grayscale images of handwritten digits, partitioned into a training set of 60,000 examples and a test set of 10,000 examples. Each image is of size 28x28 pixels, representing digits from 0 to 9, making our prediction problem a multi-class classification task. The examples are evenly distributed among both the train and test sets, thus removing the need for any augmentation techniques due to class imbalance. In order to align ourselves with previous literature, the images values are normalized in the $[0, 1]$ interval, as well as all attacks and subsequent images' modifications.

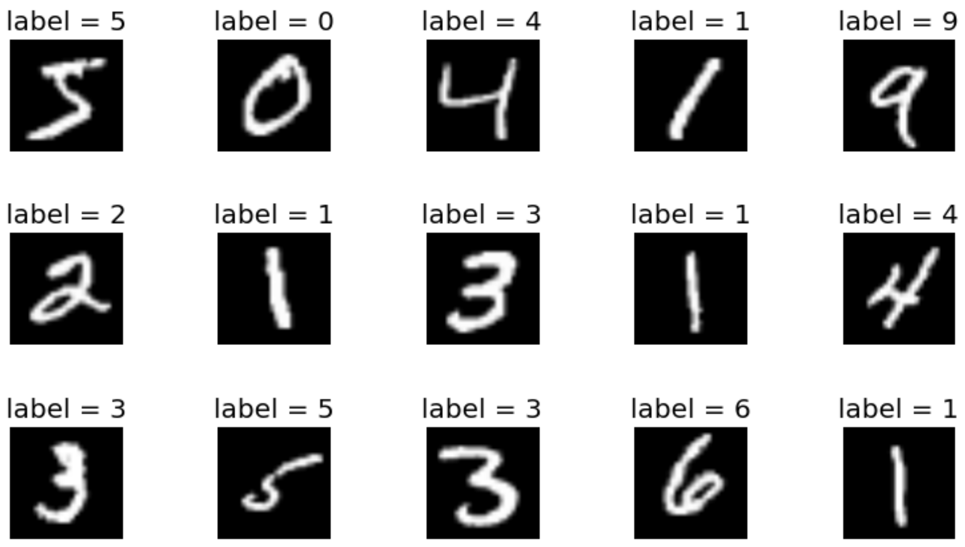

**Figure 5.1:** Example images taken from the MNIST database

### 5.2.2 PneumoniaMNIST

The PneumoniaMNIST dataset [24] is a specialized medical imaging dataset derived from the National Institutes of Health (NIH) Chest X-ray dataset, focusing on pediatric chest X-ray images to identify the presence of pneumonia.

The PneumoniaMNIST dataset (shown in Figure 5.2) consists of 5,856 chest X-ray images, divided into three categories: normal, bacterial pneumonia, and viral pneumonia. The machine learning task is a binary classification one, namely to distinguish between the samples that exhibit pneumonia and those who appear normal. These images are preprocessed and standardized to a resolution of 28x28 pixels, enabling efficient training and evaluation of machine learning models. The dataset is split into 4,708 training images, 524 validation images, and 624 test images. It is worth mentioning the slight class imbalance, the dataset containing almost twice as more images that show pneumonia than those that do not. Again, we normalize the grayscale images in the $[0, 1]$ interval, to align ourselves with previous work.

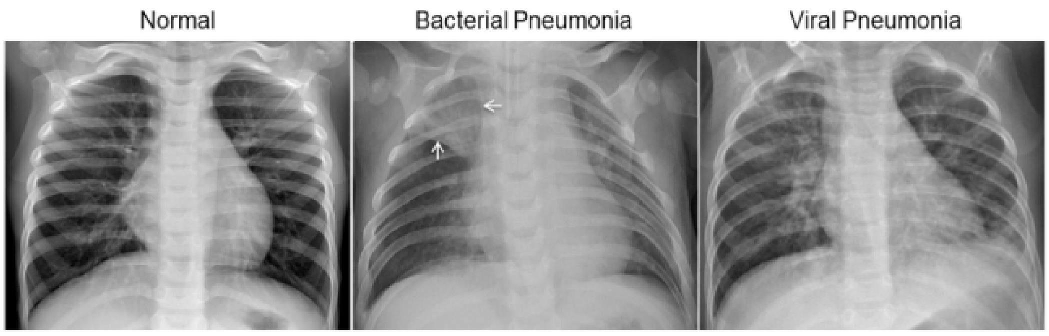

**Figure 5.2:** Example images taken from the PneumoniaMNIST dataset [25]

This dataset is not only important in a quantitative sense, but also in a qualitative one, because of the implications of potential data leakage and malignant adversarial attacks. In particular, a compelling example consists on the existence of a diagnosis and medication administration system trained on a similar dataset that does not exhibit robustness or privacy properties. A potential attacker can take advantage of such a system and feed it adversarial inputs in a manner that leads to the incorrect prediction of necessary medication, which can have catastrophic effects for a patient's health, and even cause death.

### 5.2.3 FashionMNIST

The Fashion MNIST dataset (shown in Figure 5.3) is a widely utilized benchmark dataset designed as a direct drop-in replacement for the original MNIST dataset, offering a more challenging alternative. It consists of 70,000 grayscale images, each 28x28 pixels, depicting various fashion items across 10 categories, including T-shirts, trousers, bags, and shoes. The dataset is divided into a training set of 60,000 images and a test set of 10,000 images, with each category represented equally. Fashion MNIST provides a more complex and realistic testing ground for machine learning algorithms due to its increased variability and subtle distinctions between classes. Because the data provided by the FashionMNIST dataset differs significantly in distribution in comparison to the aforementioned two datasets, this thesis will employ it to evaluate the uncertainty on an Out-Of-Distribution (OOD) detection task.

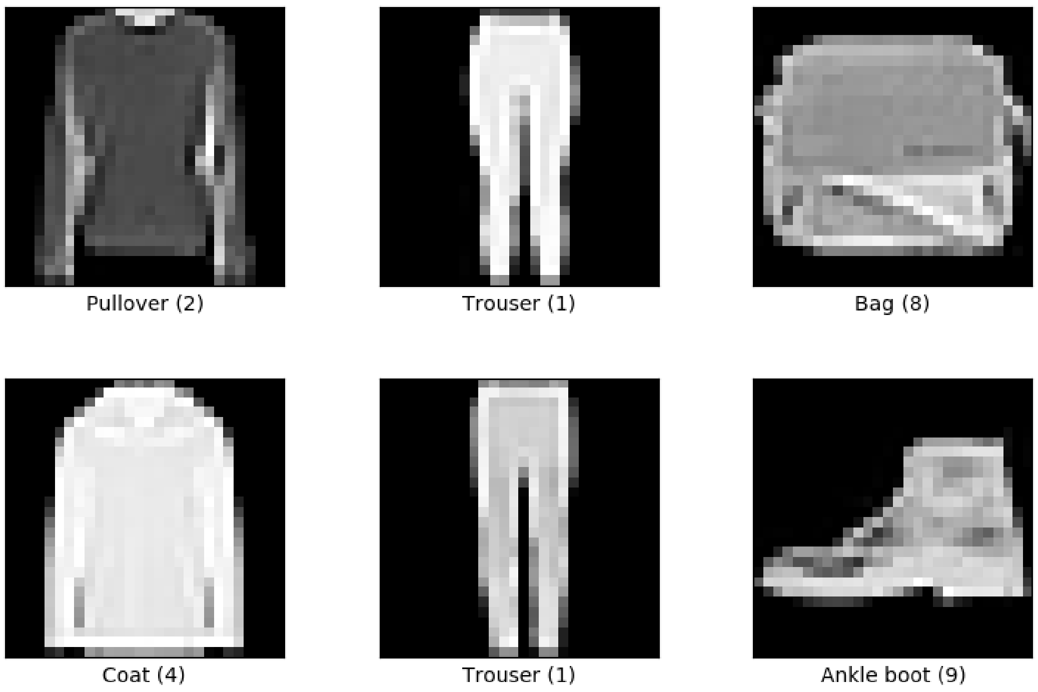

**Figure 5.3:** Example images taken from the FashionMNIST dataset [25]

### 5.2.4 Experimental setup

For each dataset, two groups, each consisting of 5 types of models trained using different training methods will be a presented and analysed.

The first group consists of a (i) deterministic neural network trained using stochastic gradient descent with learning rate decay and four Bayesian neural networks trained with Hamiltonian Monte Carlo (ii) without adversarial attacks, (iii) with adversarial attacks generated by the FGSM method, (iv) with adversarial attacks generated by the PGD method and (v) with IBP. This first group is included to show the reproducibility of our method established by previous literature such as [1, 2, 11, 12]. From now on, we will name these models during our analysis, in order: **SGD**, **HMC**, **FGSM-HMC**, **PGD-HMC** and **IBP-HMC**.

The second group consists of a (i) deterministic neural network trained using stochastic gradient descent with learning rate decay and four Bayesian neural networks trained with Hamiltonian Monte Carlo (ii) without DP, (iii) with DP, (iv) with IBP and (v) with IBP and DP. Out of these models, it is worth noting that (iii) and (v) are both novel methodologies, the first never having its privacy guarantees analysed in the context of Bayesian neural networks (i.e. when used for inferring the network parameters), and the latter filling a significant gap in the literature. This group of models represents the main quantitative result of this work and shows the performance of the newly introduced methodology in section 4.3 in comparison to previously studied training techniques in all four areas of our interest: robustness, privacy, uncertainty and precision. Similarly to the previous group, the models belonging to this group will be from now on referred to as: **SGD**, **HMC**, **DP-HMC**,

**IBP-HMC** and **IBP-DP-HMC**.

For each model in both groups, we will perform our comparative analysis by examining:

1. ***Model performance*** on the test set. This will be measured by standard accuracy.
2. ***Robustness***. This will be quantified at a fixed perturbation budget $\epsilon$, by evaluating the accuracy under FGSM, PGD and IBP-based attacks. The value of the latter will also indicate the *certified robustness* of a given model at the maximum deviation level, $\epsilon$.
3. ***Uncertainty***. This will be computed by using the AUROC (Area Under the Receiver Operating Characteristic) and ECE (Expected Calibration Error) metrics. The resultant values of these performance metrics will be shown for both the associated In-Distribution (ID) test set of the task, as well as for an Out-Of-Distribution (OOD) detection task.

The model architectures of the neural networks used across all models for each dataset are:

1. **MNIST**: A network with a single hidden layer consisting of 512 neurons.
2. **PneumoniaMNIST**: A Convolutional neural network (CNN) consisting of 2 convolutional layers, one with 1 input channel, 16 output channels, kernel size 4x4 and stride 2 and one with 16 input channels, 32 output channels, kernel size 4x4 and stride 1, as well as 2 linear layers, one with 4800 and one with 100 neurons.

The loss functions used are *cross entropy* for the first dataset and *binary cross entropy* for the second one.

## 5.3 Robustness

### 5.3.1 MNIST

Figure 5.4 shows the resistance of our first group of models (trained with $\epsilon = 0.1$ and tested with $\epsilon = 0.075$ perturbation radii) under FGSM and PGD attacks, as well as their certified IBP robustness. Our implementation replicates previous work [2, 11], which demonstrates that (i) BNNs are naturally more robust, (ii) the strength of the attack decreases robustness and (iii) the strength of the attack used in the training process increases robustness. In particular, it is important to notice that out of all the training methods, only IBP-training achieves a non-zero robustness guarantee.

Our novel training methods include (i) the first mention in the literature of a BNN trained using HMC equipped with DP mechanisms and (ii) the new ADV-HMC-DP training methods, described in section 4.3, which incorporates both DP mechanisms and adversarial training using IBP. Figure 5.5 similarly shows their resistance against

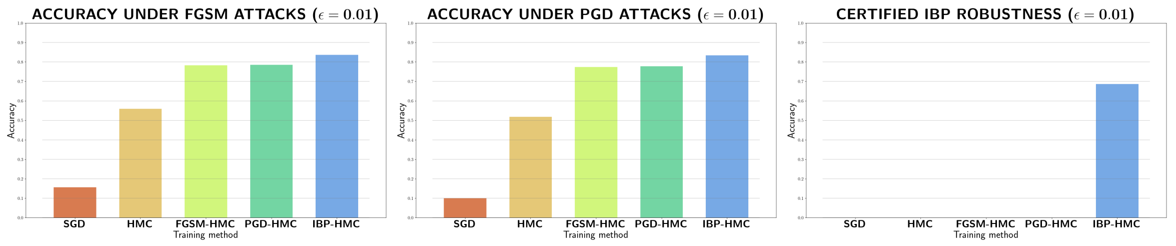

**Figure 5.4:** SGD and ADV-HMC robustness profiles on the MNIST test set

FGSM and PGD attack vectors, as well the the certified guarantee given at inference time by IBP. Our new techniques perform similarly to standard and IBP-trained HMC, HMC-DP achieving 45.08% and 42.11% accuracy when attacked with FGSM and, respectively PGD. Again, only the IBP-trained methods have non-zero robustness guarantees, IBP-DP-HMC exhibiting a slight decrease and thereby attaining 57.47% robustness in comparison to the 68.68% of the standard IBP-HMC. These results validate the part of our hypothesis which states that (differential) privacy and resilience can be attained simultaneously, in our case with only a small loss of our certified robustness guarantees. It is also worth mentioning that all models in Figure 5.5 have been trained with $\epsilon = 0.1$, with the exception of IBP-DP-HMC, which has been trained with $\epsilon = 0.075$. This is once more an artifact of the fact that adding DP mechanisms in the training process makes the data manifold significantly more complex and thus, the task becomes more difficult to optimise.

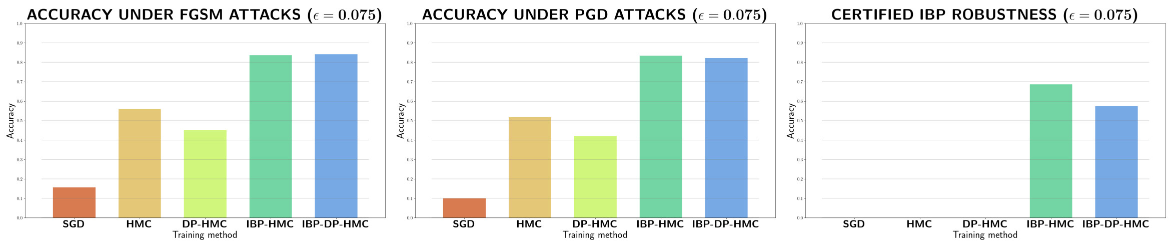

**Figure 5.5:** SGD and ADV-DP-HMC robustness profiles on the MNIST test set

The exact results obtained for each model and attack are shown in Table 5.1.

| Attack Type | Model Name | | | | | | |
|---|---|---|---|---|---|---|---|
| | **SGD** | **HMC** | **FGSM-HMC** | **PGD-HMC** | **IBP-HMC** | **DP-HMC** | **IBP-DP-HMC** |
| **FGSM** | 15.67% | 56.01% | 78.26% | 78.51% | 83.7% | 45.08% | 84.13% |
| **PGD** | 10.03% | 51.86% | 77.36% | 77.75% | 83.42% | 42.11% | 82.17% |
| **IBP** | 0.0% | 0.0% | 0.0% | 0.0% | 68.68% | 0.0% | 57.47% |

**Table 5.1:** Robustness of MNIST trained models under different types of attacks

## 5.3.2 PneumoniaMNIST

Once again, as in the case of the MNIST dataset, we validate our methodology approach by recreating experiments which are consistent with the well-known, already existing literature results, as shown in Figure 5.6.

It is worth mentioning that predicting pneumonia occurrence in patients by analysing X-rays is a much harder task than the one introduced by the MNIST dataset. Because of that and the increased complexity of the networks' architecture, it is difficult to non-trivially certify robustness. For this reason, all the models used for this task have been trained and tested at $\epsilon = 0.01$. In particular, while the small perturbation radius makes defence against FGSM and PGD attacks easier, the IBP robustness is always zero, with the exception of IBP-HMC and IBP-DP-HMC, as expected. Moreover, even if the FGSM and PGD adversaries for this particular robustness budget makes are less potent, fact which determines the higher vanilla HMC FGSM and PGD accuracies, models trained with these attacks can scale to higher perturbations, while SGD and HMC are not able to do that.

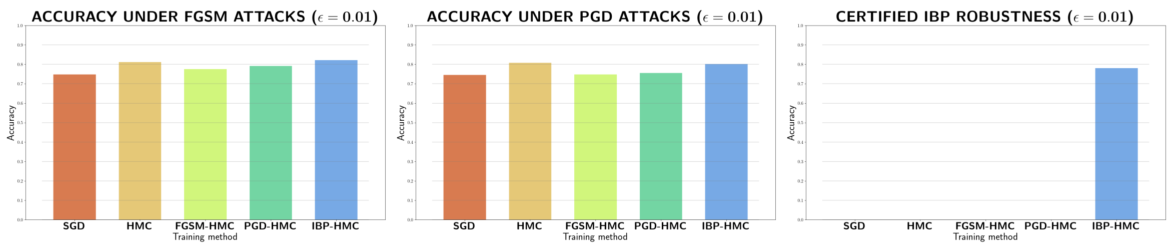

**Figure 5.6:** SGD and ADV-DP-HMC robustness profiles on the PneumoniaMNIST test set

Similarly to the MNIST case, the best model obtained by training using both IBP and DP mechanisms (found in Figure 5.7) confirms our hypothesis, guaranteeing 73.88% robustness, only 4.16% less than its IBP-HMC counterpart, which does not use any privacy enhancing techniques during training.

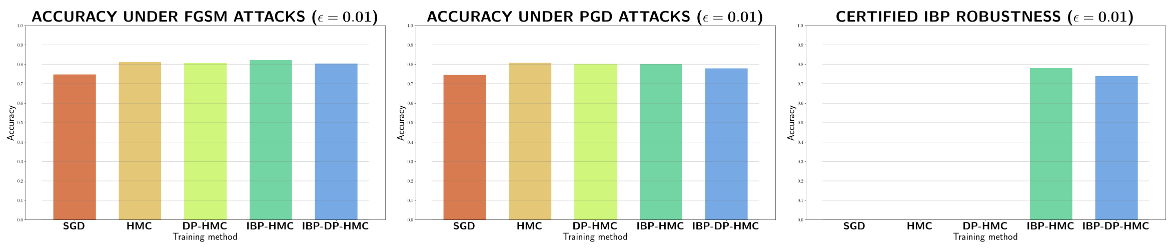

**Figure 5.7:** SGD and ADV-DP-HMC robustness profiles on the PneumoniaMNIST test set

| Attack Type | Model Name | | | | | | |
|---|---|---|---|---|---|---|---|
| | **SGD** | **HMC** | **FGSM-HMC** | **PGD-HMC** | **IBP-HMC** | **DP-HMC** | **IBP-DP-HMC** |
| **FGSM** | 74.84% | 81.09% | 77.56% | 79.17% | 82.21% | 80.61% | 80.45% |
| **PGD** | 74.52% | 80.77% | 74.84% | 75.48% | 76.28% | 80.29% | 77.88% |
| **IBP** | 80.13% | 0.0% | 0.0% | 0.0% | 78.04% | 0.0% | 73.88% |

**Table 5.2:** Robustness of PneumoniaMNIST trained models under different types of attacks

Interestingly, both DP-HMC and IBP-DP-HMC perform comparably, or even better than HMC and IBP-HMC, respectively, when attacked using the FGSM and PGD techniques. There are be two possible reasons for this behaviour: (i) the aggressive optimisation parameters needed to overcome local minima during training or (ii) the random noise added as part of the DP mechanism, which acts as regularization, leading to accurate results.

The exact results obtained for each model and attack are shown in Table 5.2.

## 5.4 Uncertainty

Firstly, in order to understand the results obtained, it is important to define the *AUROC* and *ECE* uncertainty metrics, as well as the tasks' performance they quantify.

**AUROC** (Area Under the Receiver Operating Characteristic Curve) measures the ability of a model to discriminate between classes, with a value of 1 indicating perfect discrimination and 0.5 indicating no better than random guessing.

**ECE** (Expected Calibration Error) quantifies the difference between predicted probabilities and the actual outcomes, providing an overall measure of how well the predicted probabilities are calibrated. A lower ECE value indicates a small calibration error, while a high value suggests our model is not performant enough.

The **In-Distribution (ID)** task involves taking the posterior samples of the trained network and computing the mean of the posterior predictive distribution on the test set of the dataset used in training and feeding the predictions into the AUROC computation.

The **Out-Of-Distribution (OOD)** task involves taking the test set of a dataset used in training and another test set of a significantly different (OOD) task (in our case, classifying a fashion item in the FashionMNIST dataset). The maximum softmax probability given by our predictive distribution is regarded as the result of a binary classification problem (where $0$ represents and OOD sample and $1$ an ID sample), and thresholded, with the values above the threshold being considered ID ($1$) and the values below OOD ($0$). The AUROC and ECE metrics are then computed for these predictions. The thresholds chosen are $0.5$ for multi-class classification (MNIST) and $0.65$ for binary classification (PneumoniaMNIST). These both allow for a large margin of confidence between the highest two softmax probabilities, meaning that the system confidently made a prediction, regardless of whether the prediction is correct or not.

### 5.4.1 MNIST

The evaluation of the predictions obtained in both our In-Distribution and Out-Of-Distribution tasks for the first group of models are included in Figure 5.8, which shows the value of the metrics defined above. Our baseline results are consistent with previous work, such as [11], and show that probabilistic methods (in our case

HMC) exhibit similar performance for the ID case (that is, testing on a distribution similar to the one the system was trained), and perform significantly better in quantifying uncertainty in the OOD case. In particular, it is easily noticeable that across all our probabilistic methods, they perform at least 20% better when computing AUROC, meaning that they consistently are less certain of unseen results than the deterministic network. Similarly, the ECE is almost double for SGD compared to other methods, showing poor generalization on OOD data, which is expected. Interestingly, the probabilistic methods performs slightly worse when measuring ECE for ID data, which can be attributed to their inherent nature, i.e. their ability to characterize a whole distribution instead of just providing an accurate point estimate.

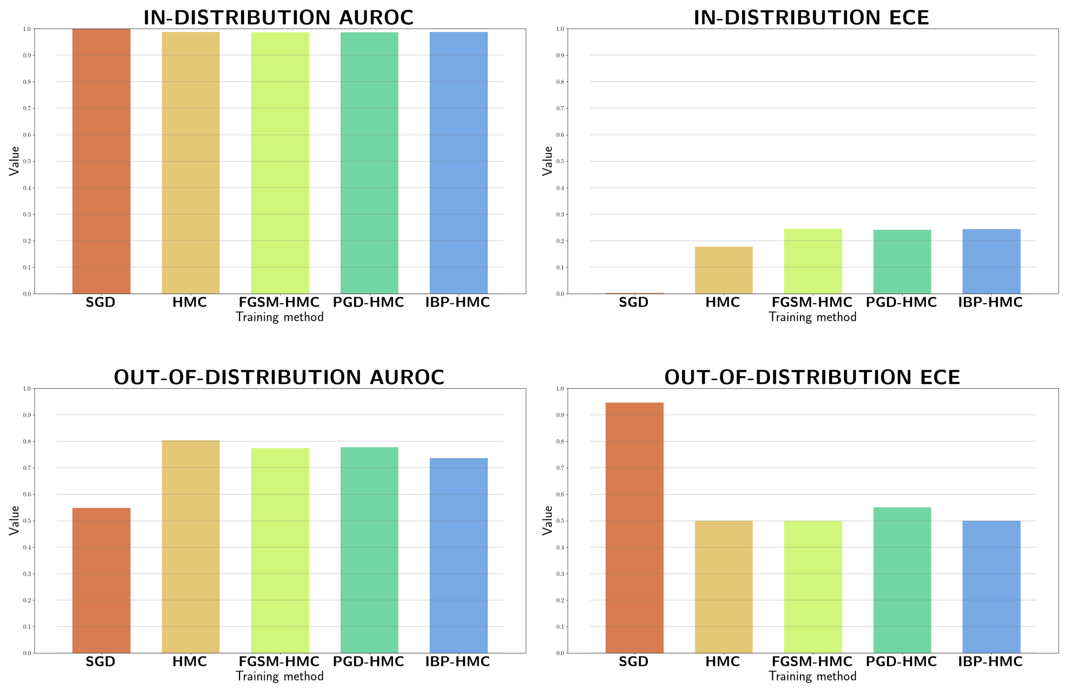

**Figure 5.8:** AUROC and ECE values on ID and OOD tasks using SGD and ADV-HMC

Figure 5.9 holds essential significance in this thesis, because it shows the ability of our newly designed methodology and algorithm to quantify uncertainty with minimal loss in value compared to purely robust techniques. DP-HMC and IBP-DP-HMC achieve 76.42% and 72.07% on the OOD task, values which highly exceed the 54.98% obtained by SGD, validating the component of our hypothesis which states that privacy and effective uncertainty quantification are simultaneously achievable.

Notably, this does not hurt the AUROC value in the ID case, and when looking at ECE values, we observe that it even performs better for the same case. A potential explanation for this is that introducing DP mechanisms acts as regularization, due to the random noise being added at each training step.

The exact uncertainty metric scores obtained by each model are shown in Table 5.3.

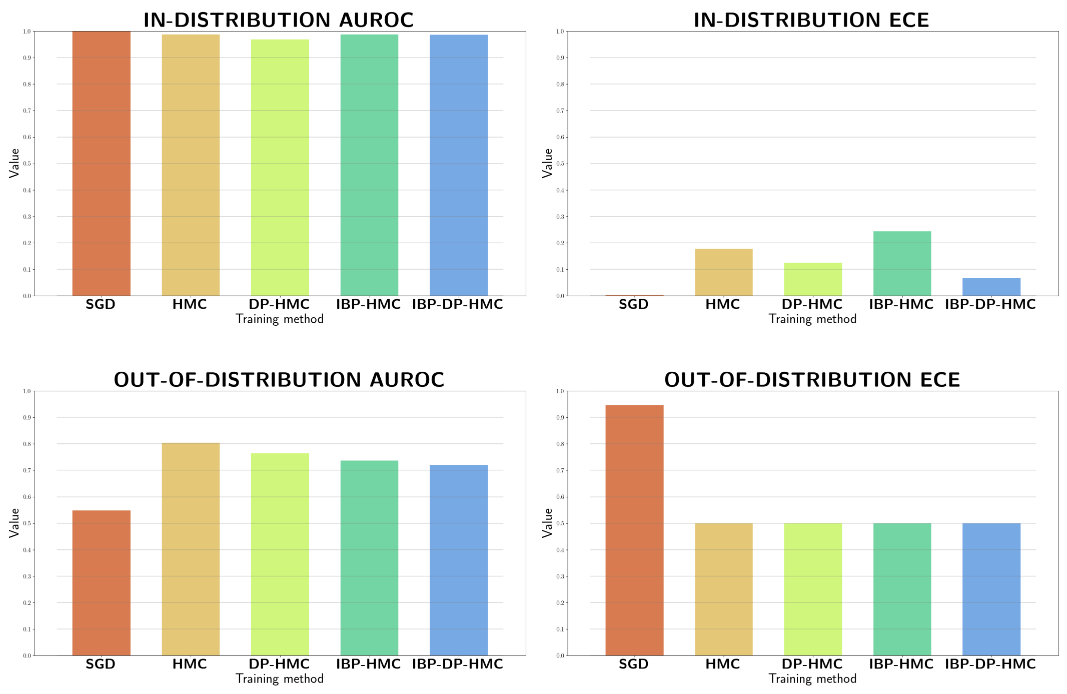

**Figure 5.9:** AUROC and ECE values on ID and OOD tasks trained on MNIST using SGD and ADV-DP-HMC

| Metric Name | Model Name | | | | | | |
|---|---|---|---|---|---|---|---|
| | **SGD** | **HMC** | **FGSM-HMC** | **PGD-HMC** | **IBP-HMC** | **DP-HMC** | **IBP-DP-HMC** |
| **ID AUROC** | 99.97% | 98.72% | 98.62% | 98.64% | 98.78% | 96.86% | 98.67% |
| **ID ECE** | 0.33% | 17.75% | 24.48% | 24.12% | 24.37% | 12.57% | 6.71% |
| **OOD AUROC** | 54.89% | 80.36 % | 77.38% | 77.76% | 73.69% | 76.42% | 72.07% |
| **OOD ECE** | 94.59% | 50.0% | 50.0% | 55.02% | 50.0% | 50.0% | 50.0% |

**Table 5.3:** Uncertainty metrics of MNIST trained models

### 5.4.2 PneumoniaMNIST

The behaviour for the ID task is similar to the one observed when training on the MNIST dataset. Hence, we can easily observe in the top row of Figure 5.10 that the AUROC values remain consistently high across all models, suggesting good discrimination between classes, while the ECE values for the probabilistic models are slightly higher, likely because of their ability to better generalize and defend against adversarial attacks, which in turn means that mathematically, around an input data point, its perturbed variants will behave the same at inference time, which is consistent with the results obtained. The only exception to that is the vanilla HMC, which achieves very low ECE, as it is highly correlated with its better performance in comparison to the other models.

While still behaving better in uncertain scenarios, as shown in the bottom row of Figure 5.10, it is interesting to see that the AUROC values of the probabilistic robust models are only marginally higher than the deterministic model, going only up to

4.71%. There are three explanations for this behaviour. Firstly, the task induced by this dataset is significantly harder than the task induced by the MNIST dataset, thus making optimisation more difficult. Secondly, the dataset used in the OOD task (FashionMNIST) differs in distribution more with respect to PneumoniaMNIST than it does to MNIST. That is why, even with the advantage of having multiple samples from the posterior rather than just a point estimate, it is difficult to quantify uncertainty significantly better when hardly any distribution overlap is present. Lastly, it is important to remember the fact that there is a small trade-off between uncertainty and performance, and also that this work has been focused on optimising against robustness, privacy and uncertainty simultaneously. Thus, it is possible to achieve even better uncertainty for the BNNs, but at the price of a small decrease in accuracy. In the end, it is the practitioners' choice what properties are most important in their systems.

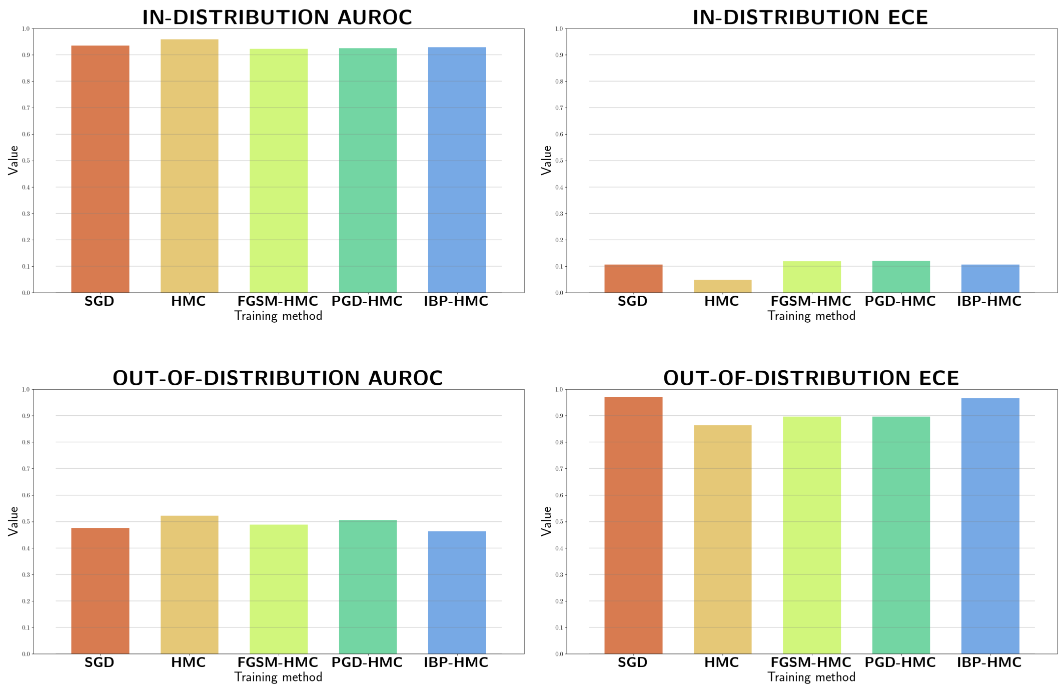

**Figure 5.10:** AUROC and ECE values on ID and OOD tasks trained on PneumoniaMNIST using SGD and ADV-HMC

Once again, the section of our hypothesis that claims simultaneously attaining privacy and accurately estimating uncertainty is proven by the metric values obtained, as shown in Figure 5.11. While the AUROC measured in the context of the ID task on the IBP-DP-HMC model performs almost equivalently to the non-DP probabilistic models, the smaller ID ECE is indicative of a better performing system at inference time. As it was the case previously, we hypothesize that these results are the effect of random noise introduced at training time, which behaves akin to regularization.

For the same reason, DP-HMC and IBP-DP-HMC are able to discriminate significantly better between classes in an OOD setting than SGD and IBP-HMC, achieving AUROC values of 49.51% and 53.83%, 1.97% and respectively, 6.29% more than the deterministic network. Seeing this from a different perspective is a better indicator of performance, since relative to the baseline SGD value, this is an increase of more than 12%. The OOD ECE values corroborate our claim, by outputting values which suggest lower errors than a deterministic approach would provide. Once again, albeit the complexity of the task and the disparity between distributions in the OOD settings makes the uncertainty only slightly better, it is possible to achieve higher values at the cost of performance, subject to the use case in which such a model is utilised in.

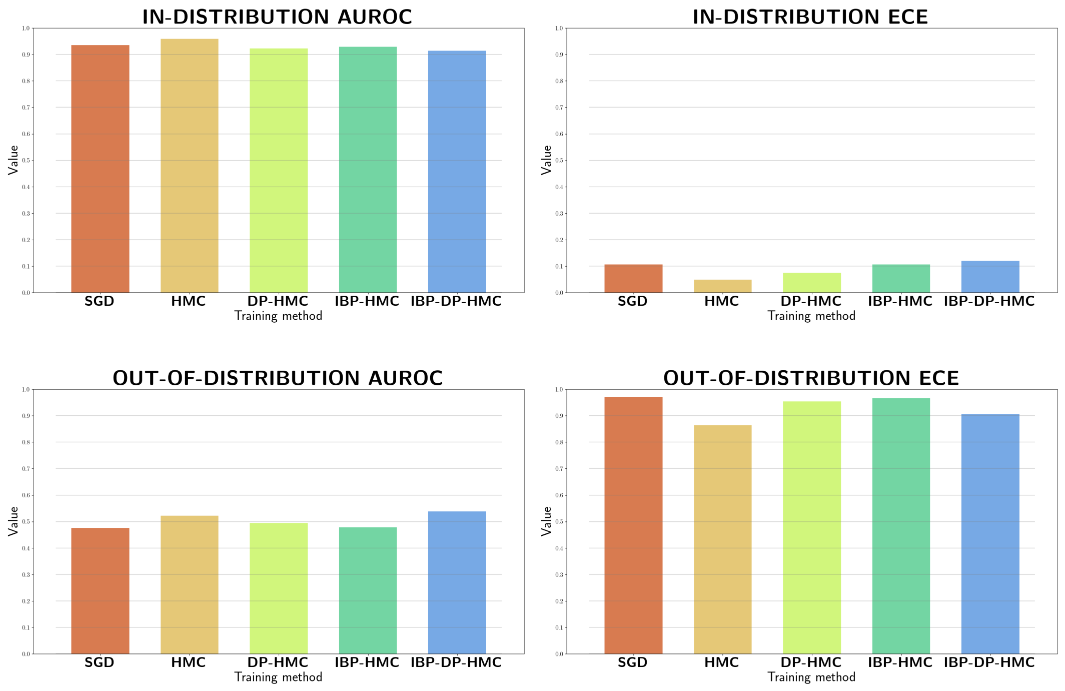

**Figure 5.11:** AUROC and ECE values on ID and OOD tasks trained on PneumoniaMNIST using SGD and ADV-DP-HMC

The exact uncertainty metric scores obtained by each model are shown in Table 5.4.

| Metric Name | Model Name | | | | | | |
|---|---|---|---|---|---|---|---|
| | **SGD** | **HMC** | **FGSM-HMC** | **PGD-HMC** | **IBP-HMC** | **DP-HMC** | **IBP-DP-HMC** |
| **ID AUROC** | 93.48% | 95.85% | 92.28% | 92.51% | 92.86% | 92.27% | 91.33% |
| **ID ECE** | 10.95% | 4.93% | 11.95% | 12.09% | 10.67% | 7.53% | 12.07% |
| **OOD AUROC** | 35.71% | 52.25% | 48.82% | 50.64% | 46.32% | 49.51% | 53.83% |
| **OOD ECE** | 88.12% | 86.33% | 89.59% | 89.59% | 96.67% | 95.32% | 90.64% |

**Table 5.4:** Uncertainty metrics of PneumoniaMNIST trained models

## 5.5 Performance

Performance, measured by standard accuracy, is the last metric we are interested in, in order to prove our hypothesis. Since the aim of any intelligent system deployed in a real life scenario is to correctly predict or classify a given phenomena, it is critical to ensure such a system is efficient when receiving previously unseen data as input.

### 5.5.1 MNIST

Figure 5.12 shows the accuracy of the MNIST test set of our two groups of models. It is easily noticeable that our new algorithm (IBP-DP-HMC) achieves 87.12% accuracy using both adversarial training and DP mechanisms, scoring similarly to IBP-HMC and vanilla HMC, corroborating the element of our hypothesis which claims that using robustness and privacy techniques in the learning process does not hinder attaining high accuracy and effectively learning from the actual input data.

It is important nonetheless to underline the fact that the newly introduced methodologies suffer from a slight decrease in performance compared to their non-DP counterparts. This is an expected behaviour and can be motivated by our multiple previous mentions of the regularization argument, as well as by the fact that introducing multiple objectives in the training process makes it more difficult to learn a multi-component ground truth, where each of the components behave as if they would act individually. Lastly, there is also a correlation between lower accuracy and better uncertainty estimation, as less probable posterior samples capture more uncertainty but decrease precision.

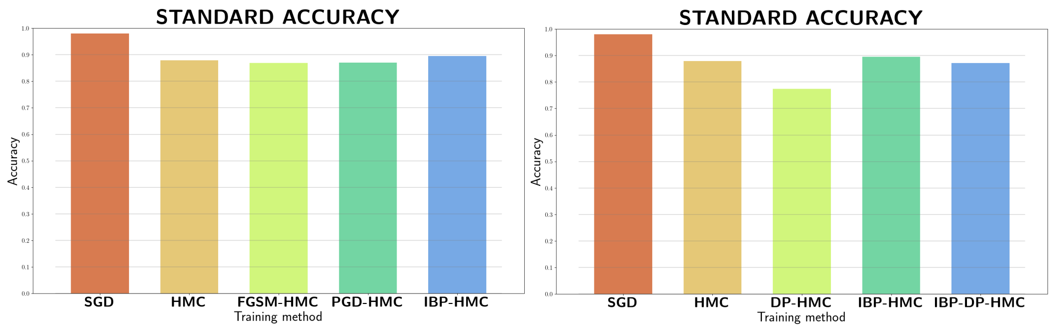

**Figure 5.12:** SGD, ADV-HMC and ADV-DP-HMC accuracy on the MNIST test set

| | Model Name | | | | | | |
|---|---|---|---|---|---|---|---|
| | **SGD** | **HMC** | **FGSM-HMC** | **PGD-HMC** | **IBP-HMC** | **DP-HMC** | **IBP-DP-HMC** |
| **Accuracy** | 98.03% | 87.9% | 86.85% | 86.99% | 89.47% | 77.4% | 87.12% |

**Table 5.5:** Standard accuracy of MNIST trained models

The standard accuracy scores obtained by the two groups of models can be found in Table 5.5.

### 5.5.2 PneumoniaMNIST

The performance on the PneumoniaMNIST dataset (which can be found in Figure 5.13) once again validates our hypothesis, exhibiting accuracy values of 86.86% and 83.81% when trained with DP-HMC and IBP-DP-HMC. The increased complexity of the task, as well as the hypothesis that DP acts akin to regularization makes the DP-models score lower than their non-DP counterparts. However, the score difference (2.4% and 1.93%) is almost insignificant, and taking into consideration the considerable uncertainty and robustness advantages described in Section 5.4.2 and 5.3.2, it is a small price to pay for achieving multiple desirable properties at once.

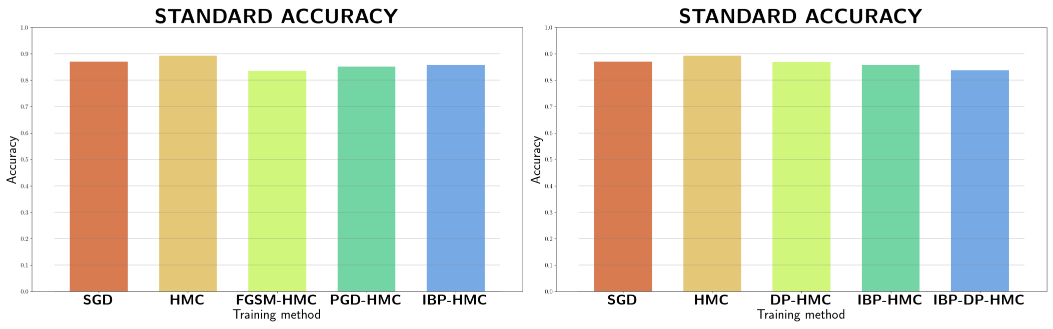

**Figure 5.13:** SGD, ADV-HMC and ADV-DP-HMC accuracy on the PneumoniaMNIST test set

| | Model Name | | | | | | |
|---|---|---|---|---|---|---|---|
| | **SGD** | **HMC** | **FGSM-HMC** | **PGD-HMC** | **IBP-HMC** | **DP-HMC** | **IBP-DP-HMC** |
| **Accuracy** | 87.02% | 89.26% | 83.49% | 85.1% | 85.74% | 86.86% | 83.81% |

**Table 5.6:** Standard accuracy of MNIST trained models

The standard accuracy scores obtained by the two groups of models can be found in Table 5.6.

## 5.6 Privacy

Before we investigate the DP properties of our new training method, it is important to mention the parameters with which the best performing model (whose performance, as well as uncertainty and robustness properties have been presented above) has been trained with and which influence the DP guarantee. These can be found in table 5.7.

It should be mentioned that the *epochs* ($n_e$) parameter does not have the same meaning as the traditional *epoch* used in deterministic learning scenarios, but is rather equivalent to the number of simulation steps. Even more than that, in comparison to the deterministic case, here the dependence of each parameter in an epoch on previous epochs is subject to the Metropolis-Hastings acceptance step, and thus the

| IBP-DP-HMC Dataset | Parameter | | | | | | |
|---|---|---|---|---|---|---|---|
| | Epochs($n_e$) | Leapfrog steps ($l_f$) | $\tau_g$ | $\tau_l$ | $b_g$ | $b_l$ | Chains($n_c$) |
| **MNIST** | 60 | 120 | 0.05 | 0.05 | 0.5 | 0.5 | 3 |
| **PneumoniaMNIST** | 80 | 24 | 0.1 | 0.1 | 0.5 | 0.5 | 3 |

**Table 5.7:** Best performing DP models' parameters

final number of posterior samples gathered is less than or equal to the number of *epochs*. Lastly, each *chain* is an independent HMC simulation.

Reiterating the guarantees given by [14] and described in 4.2, using our parameters, we have, using any $\epsilon$ of our choosing:

$$\delta(\epsilon) = \frac{1}{2}\left[\operatorname{erfc}\left(\frac{\epsilon-\mu}{2\sqrt{\mu}}\right) - e^{\epsilon}\operatorname{erfc}\left(\frac{\epsilon+\mu}{2\sqrt{\mu}}\right)\right] \text{ with } \mu = n_c\left[\frac{n_e}{2\tau_l^2} + \frac{2n_e(l_f+1)}{2\tau_g^2}\right] \quad (5.1)$$

Notably, this privacy bound is the baseline one (i.e. which one would obtain when performing full-batch learning), but the bound obtained in practice is tighter, due to subsampling and gradient averaging. The use case we apply these guarantees on requires multiplying by a factor of 2 the second addend of $\mu$, because two gradients are computed and clipped at each backpropagation step, the one with respect to the standard input and the one with respect to the adversarial input. Using this equation, we achieve vacuous bounds for both models trained using IBP-DP-HMC. That is to say, we can obtain $(\epsilon, \delta(\epsilon))$-DP with $\epsilon = 0$ and $\delta(\epsilon) = 1$, or $\epsilon \gg 100$ and $\delta(\epsilon) < 1$. As mentioned in [14], this is expected for DP-MCMC methods trained on datasets with less than $10^5$ data points.

The reason we obtain these guarantees is the tension introduced by the two opposing forces at play in DP-BNNs: *privacy* and *parameter space exploration*. Because a high $\mu$ requires a high $\epsilon$ in order to achieve non-vacuous bounds, equation 5.1 clearly shows that when $n_c$, $n_e$ and $l_f$ are large and $\tau_l$ and $\tau_g$ are small, the DP guarantee is very loose. That being said, the advantage of BNNs over DNNs in terms of uncertainty comes exactly from the fact that they are able to fully characterize complex distribution either through inferring their parameters or, in our case, by sampling. This inherently implies the fact uncertainty and the number of samples are correlated (especially when the number of samples cover a large portion of the distribution of interest), the accuracy of OOD uncertainty estimation increasing proportionally with the of the value exploration parameters, which are precisely $n_c$, $n_e$ and $l_f$. To complement this, the sensitivity parameters $\tau_g$ and $\tau_l$ are required to stay small, otherwise the backpropagation step is uninformative because too much random noise is added. For this reason, we shall study the effect of modifications of these parameters with respect to uncertainty and privacy, and also show what happens to our robustness guarantees and performance, although they are not the main forces at stake in this trade-off. In particular, while we have shown in the previous section (and it is also well known) that the two latter properties are not at odds with uncertainty, it will be interesting to observe what a lack of efficient exploration in the statistical parameters space limits the utility of learning with respect to these two quantities of interest.

To observe this trade-off, we modify our aforementioned parameters to obtain tight robustness guarantees, and observe the effect these have on the other properties of our IBP-DP-HMC model at inference time. Thus, the new relevant parameters for our models can be found in Table 5.8.

| IBP-DP-HMC Dataset | Parameter | | | | | | |
|---|---|---|---|---|---|---|---|
| | Epochs($n_e$) | Leapfrog steps ($l_f$) | $\tau_g$ | $\tau_l$ | $b_g$ | $b_l$ | Chains($n_c$) |
| **MNIST** | [1..21]; step 2 | 10 | 2 | 2 | 0.1 | 0.1 | 1 |
| **PneumoniaMNIST** | [1..21]; step 2 | 6 | 2 | 2 | 0.1 | 0.1 | 1 |

**Table 5.8:** Privacy study models' parameters

It can easily be observed that the parameters are chosen specifically in order to maximize the $(\epsilon, \delta(\epsilon))$ DP guarantee. Thus, doing an ablation study on one of them (i.e. either increasing $n_e$, $l_f$ or $n_c$, or decreasing $\tau_g$ or $\tau_l$), shows us what our model is able to learn.

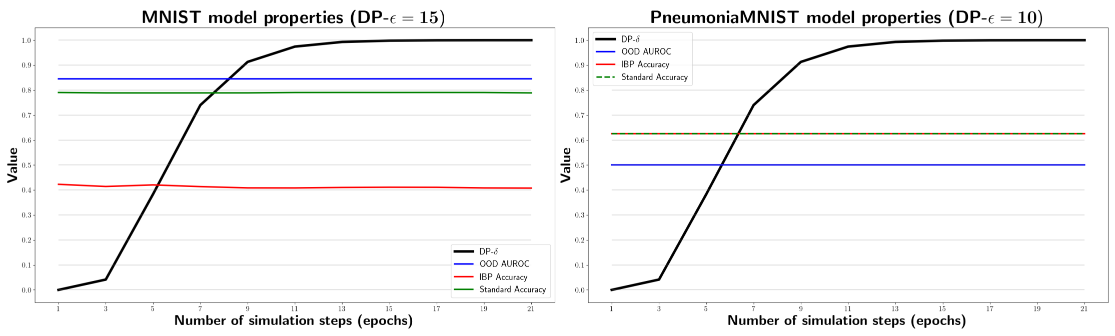

**Figure 5.14:** Effects of tight privacy guarantees on the robustness, accuracy and uncertainty of IBP-DP-HMC

Figure 5.14 clearly shows us the trade-off between DP and the other desirable properties of our HMC-trained models. In order to achieve non-vacuous bounds (that is, $\delta(\epsilon) < 1$) for a good enough $\epsilon$ (in the range of tens), the added noise at each step utilised to privatize the data ($\mathcal{N}(0, \sigma_g^2)$) makes the algorithm only sample from a very narrow region of the posterior distribution, oscillating around the initialization point. Since this equates to poor exploration of the parameter space, the performance, robustness and uncertainty properties remain almost exactly the same and the model behaves as if having a deterministic neural network with a point estimate of the weights and biases located at the region in space given by the initialization. Adding to that, because the DP $\delta(\epsilon)$ saturates quickly, by the time the relevant hyperparameters have values that allow for exploring the space, and thus learning, efficiently, the privacy guarantees are already vacuous.

## 5.7 Ablation Study

We will now study the effect of decreasing the clip bound $b_g$ and increasing the perturbation radius $\epsilon$ of IBP-DP-HMC at train time, parameters which control our privacy budget and our final robustness with respect to adversarial attack shifted at most $\epsilon$ from our initial data point. While decreasing $b_g$ gives us stronger DP guarantees for our best-performing models, it makes our gradients more uninformative. Similarly, while increasing $\epsilon$ makes our models robust to stronger attacks, it increases the complexity of the loss landscape and makes it challenging to optimise. The baseline crucial hyperparameters of IBP-DP-HMC we start from are shown in table 5.9, where $\alpha$ represents the trade-off between the standard and robust learning objectives.

| IBP-DP-HMC | Hyperparameters | | | | | | |
|---|---|---|---|---|---|---|---|
| Dataset | Epochs | Leapfrog steps | $b_g$ | $\epsilon$ | $\alpha$ | $\mathbb{V}_\theta[p(\theta)]$ | Chains |
| **MNIST** | 60 | 120 | 0.05 | 0.1 | 0.993 | 15 | 3 |
| **PneumoniaMNIST** | 80 | 24 | 0.1 | 0.01 | 0.975 | 5 | 3 |

**Table 5.9:** Best performing DP models' initial hyperparameters

### 5.7.1 MNIST

When training IBP-DP-HMC on the MNIST dataset, decreasing the gradient clip bound limits the learning utility, as demonstrated by Figure 5.15. The standard accuracy gradually declines from 87.12% to 79.06%, while the IBP accuracy steeply diminishes from 57.97% and settles at $\approx 40\%$. The latter is caused by the fact that the first chain is initialized from a pre-trained network, which causes it to remain in the neighbourhood of the accurate point estimate when running the HMC simulation.

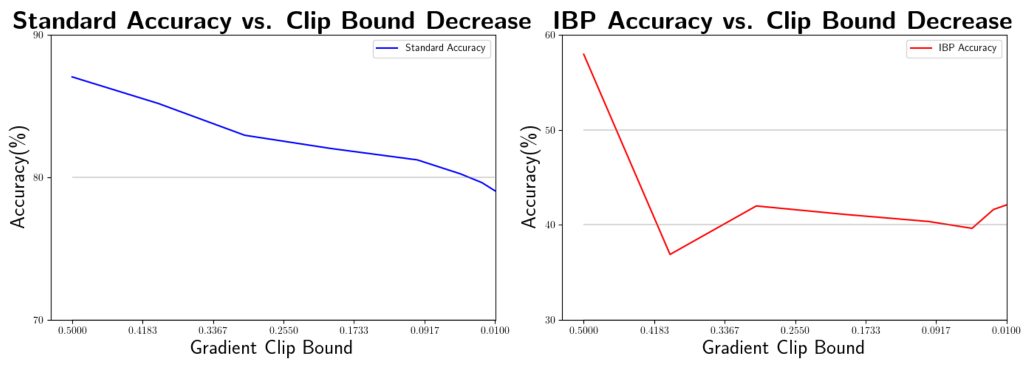

**Figure 5.15:** MNIST standard accuracy and certified robustness against $b_g$ decrease

Figure 5.16 depicts the uncertainty estimation evolution of IBP-DP-HMC when $b_g$ is varied. Naturally, as the ID AUROC is closely linked to the performance of the model, it slowly decreases $\approx 1\%$ starting from 98.67%, in a manner similar to the standard reduction in standard accuracy. The uncertainty on the OOD task is better as $b_g$ gets smaller because the learning process is more stable and general, and thus performs better on OOD data, even if it slightly underfits the training data.

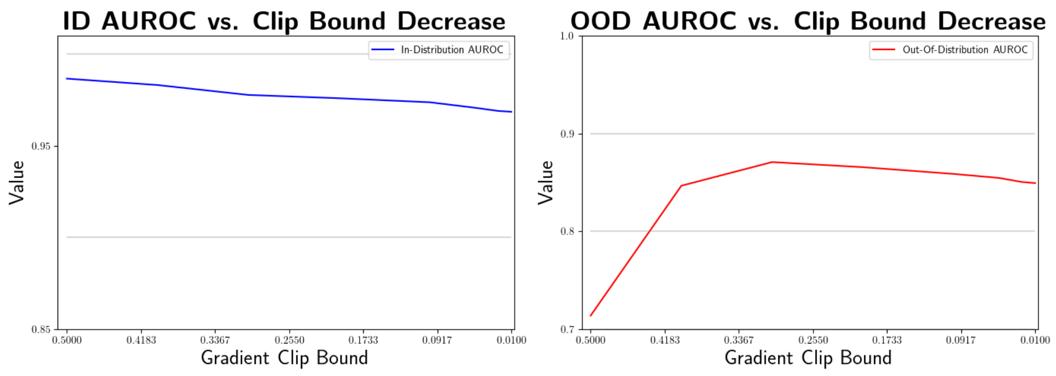

**Figure 5.16:** MNIST ID and OOD AUROC against $b_g$ decrease

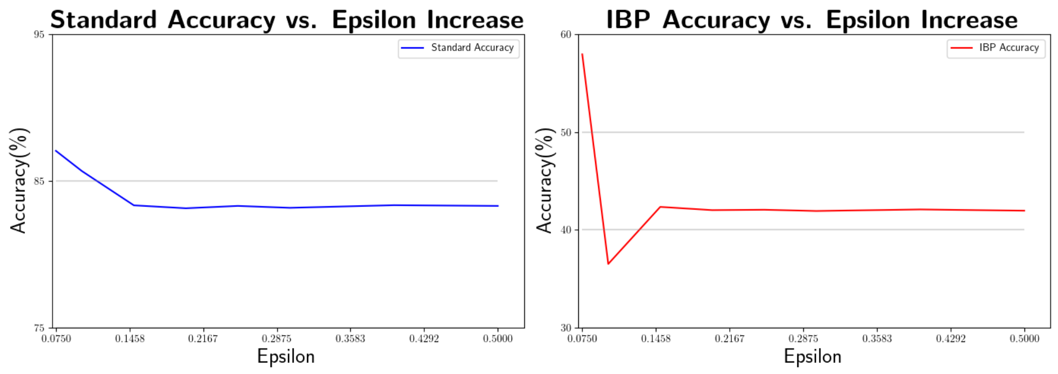

**Figure 5.17:** MNIST standard accuracy and certified robustness against $\epsilon$ increase

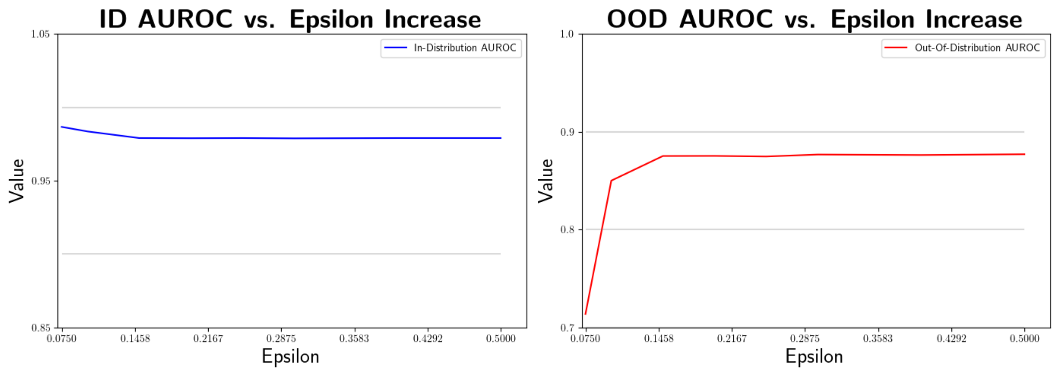

**Figure 5.18:** MNIST ID and OOD AUROC against $\epsilon$ increase

Figures 5.17 and 5.18 picture the effects of increasing the perturbation radius parameter $\epsilon$ on the accuracy, robustness and uncertainty properties of IBP-DP-HMC. Because the loss landscape is highly complex, both the performance and the certified robustness sharply drop, and then settle at the value given by a model with parameter configuration equal to their initialization. The ID AUROC only slightly decreases, confirming its strong link with accuracy, while the model performs better in the OOD task, reaffirming the fact that there is a small trade-off between accurate predictions and uncertainty guarantees when the samples from the posterior distribution of HMC are finite.

### 5.7.2 PneumoniaMNIST

For PneumoniaMNIST, decreasing the clip bound $b_g$ from $0.5$ to $0.01$, we can observe in Figure 5.19 that both the performance and certified robustness of the model take a significant hit starting with $b_g = 0.4$ and flat-lining at 62.5% when $b_g = 0.2$. This confirms the fact that gradient clipping is strongly tied with learning utility and that better privacy guarantees imply either more time to converge at train time or equivalently poorer performance when hyperparameters remain fixed. The reason for flat-lining at 62.5% is twofold: the class imbalance and the fact that when there is almost no utility from the gradients' value, the network defaults to predicting one class. Therefore, this is equivalent to the performance of random guessing.

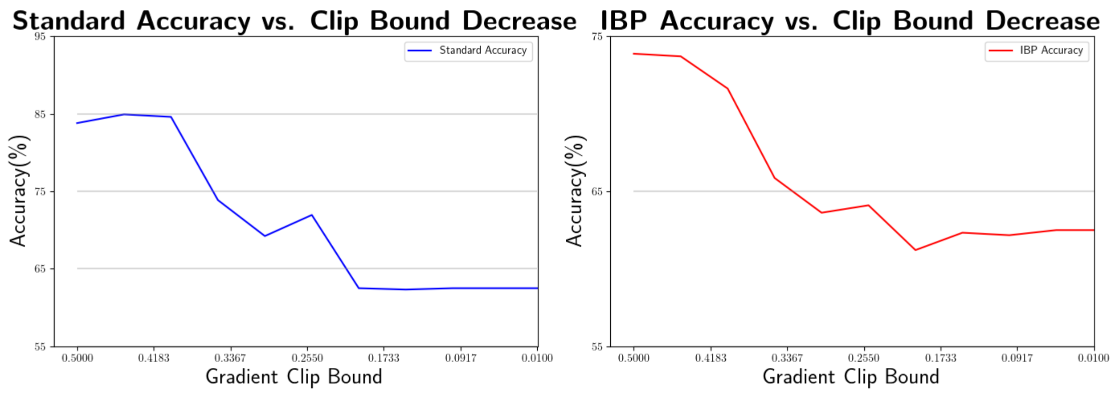

**Figure 5.19:** PneumoniaMNIST standard accuracy and certified robustness against $b_g$ decrease

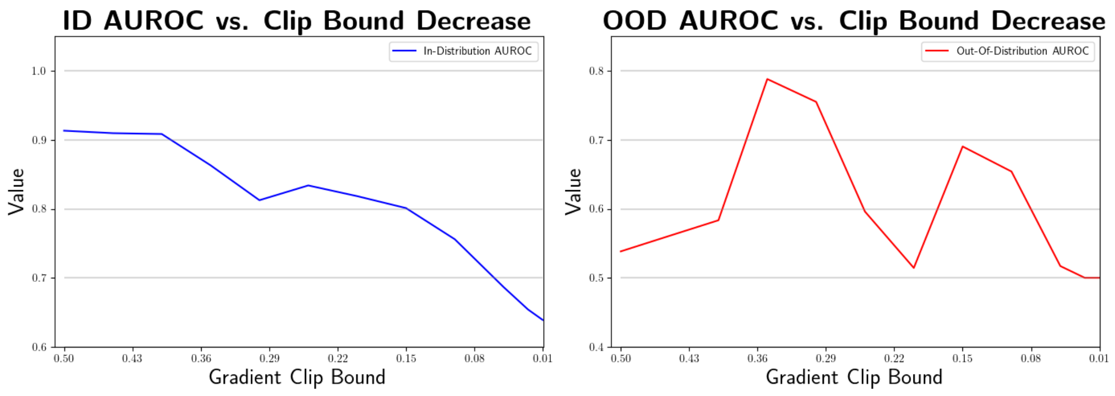

**Figure 5.20:** PneumoniaMNIST ID and OOD AUROC against $b_g$ decrease

In terms of uncertainty, Figure 5.20 shows that the ID AUROC decreases significantly, from 91.33% to 63.89%. This value is strongly correlated with standard accuracy, and we can notice that at $b_g = 0.01$ the ID AUROC is very close to the metric value when using random guessing for predictions, which is 0.5. The OOD AUROC is highly volatile and spikes at 78.82% when $b_g = 0.35$. While higher OOD AUROC might suggest more accurate uncertainty estimation, it is not the case here, because the seemingly better generalization happens only because the limited size and imbalance of the dataset makes the learned predictions match the distribution of the test set for certain specific parameters. Even more than that, the accuracy at $b_g = 0.01$ is very poor, achieving only 50%, which is exactly the random baseline. What is very interesting is that the fluctuation of this values suggests that a lower clip bounds is not necessary correlated with worse uncertainty, supporting the hypothesis (although with weak evidence) that good privacy guarantees and DP can be achieved at the same time.

Increasing the $\epsilon$ parameter has a similar effect on our IBP-DP-HMC model as decreasing $b_g$ does. Figure 5.21 pictures the fast, sharp decrease in both standard accuracy and certifiable robustness, which drop from $\approx 83\%$ and $\approx 73\%$ at $\epsilon = 0.01$ and flat-line at $62.5\%$, where $\epsilon = 0.09$. The reason for the flat-line value of $62.5\%$ is the same as the one mentioned in the DP case.

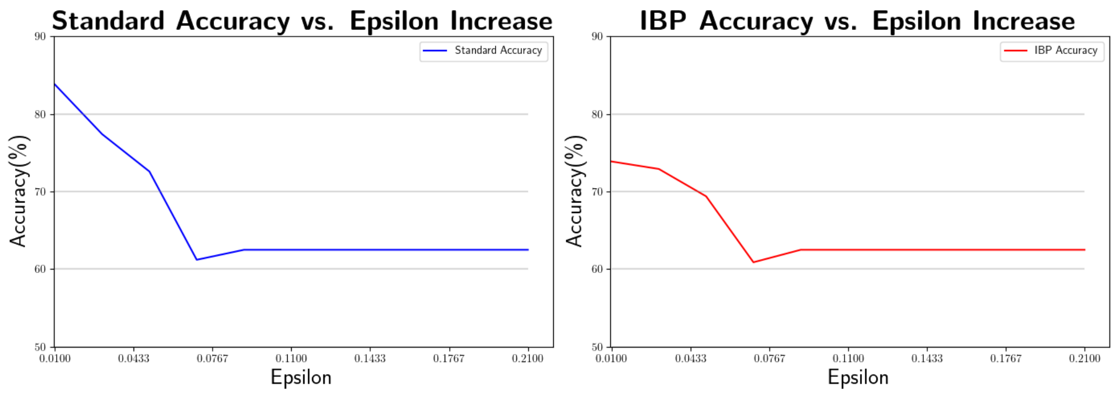

**Figure 5.21:** PneumoniaMNIST standard accuracy and certified robustness against $\epsilon$ increase

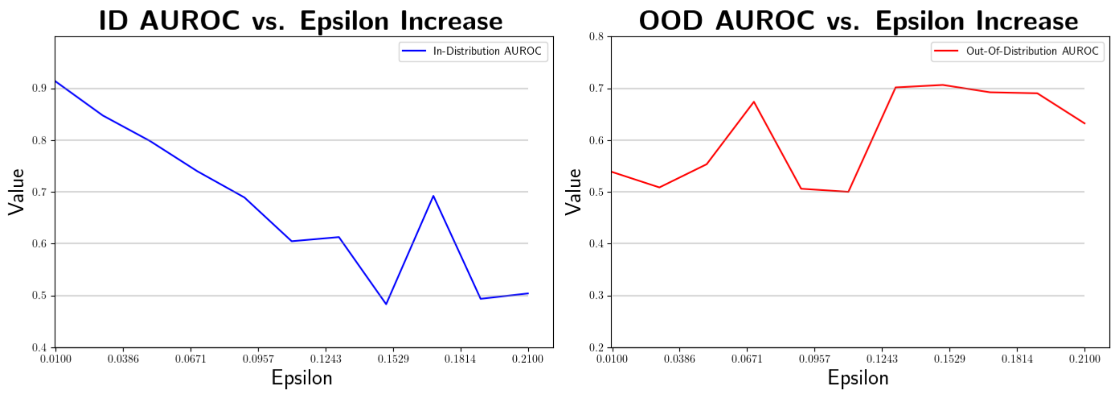

**Figure 5.22:** PneumoniaMNIST ID and OOD AUROC against $\epsilon$ increase

We also plot the variation in the uncertainty metrics' values in Figure 5.22. Once again, the ID AUROC is strongly correlated with the decrease in standard accuracy, dropping more than 40% and reaching the random guessing AUROC value, namely 0.5. Albeit still highly volatile, the OOD AUROC gains $\approx 20\%$ in value, suggesting that the random baseline is well suited for the OOD task.

## 5.8 Conclusion

The aim of this thesis has been to extend work on trustworthy properties of neural networks by shifting from the deterministic perspective and considering a Bayesian inference framework for all the experiments. Moreover, in comparison to previous literature, here we took a holistic approach to trustworthy probabilistic machine learning and investigated the intricate connections between adversarial robustness, privacy, uncertainty and performance.

Firstly, we introduced a novel training algorithm that can be used to train HMC-BNNs (and can easily be extended to other inference techniques) with simultaneous robustness and privacy guarantees, while achieving convergence to the true functions of interest. Furthermore, we successfully trained a wide variety of networks using our method and have experimentally shown their efficiency.

Secondly, we empirically demonstrated that adversarial robustness against a wide variety of attacks, as well as performance are not significantly affected by the addition of privacy-ensuring mechanisms at train time. We have quantified the differences in performance using carefully designed experiments, which measure numerous metrics, the most relevant ones being certified robustness under worst-case attack vectors and standard accuracy.

Thirdly, we found that there is an inversely-proportional correlation between tight privacy guarantees and accurate uncertainty estimation. This connection comes from the inefficient exploration of the statistical parameter space when the number of samples is small and the noise added at train time is high. Moreover, we measured this trade-off with an in-depth privacy study that shows the simultaneous evolution of privacy guarantees versus the aforementioned trustworthiness properties.

Lastly, we performed an ablation study that proves how our safety properties of interest vary with the tightness of privacy and robustness bounds. This study adds crucial insight to our previous experiments and furthers the description of the links between all the characteristics of trustworthy probabilistic machine learning systems.

# Chapter 6

# Discussion

## 6.1 Future Work

### 6.1.1 Other Bayesian Inference Methods

A wide range of state of the art Bayesian inference techniques are used in practice, such as Bayes by Backprop (**BBB**) [6], Stochastic Weight Averaging - Gaussian (**SWAG**) [26], NoisyAdam (**NA**) [27] or Variational Online Gauss-Newton (**VOGN**) [28]. Because of the generalizability of our novel training technique, as well as the fact that differential privacy has been studied and extended to multiple Bayesian learning frameworks [13], it would be interesting to extend the work done in this paper by training probabilistic models using the aforementioned techniques, and investigating how the robustness, privacy and uncertainty guarantees vary and whether better performing results can be achieved in that regard.

### 6.1.2 Membership Inference Attacks

One avenue of research that follows naturally from the work done in this thesis is how effective are black-box, privacy-breaching attacks against the models trained with our method. While multiple attacks of this kind exist, such as property inference [29] or model inversion [30] attacks, one particularly important one is the *membership inference attack* [3], because it has been used to show that in deterministic neural networks, adversarial training is detrimental to privacy [8]. Therefore, it would be useful to see whether actual adversaries are able to infer whether a data point is part of the training set as part of a concrete real-life simulation, rather than having only the mathematical DP guarantees as proof to the privacy of the system. Even more than that, extending the membership inference attacks to the probabilistic case is trivial, as the only thing one needs to do is to change the forward pass function to compute the predictive mean. However, one important consideration is that for a large number of posterior samples (which is the case for HMC), this type of attack would be extremely slow to train. Lastly, it is worth noting that initial work has already been done to implement the attack for the probabilistic case and the last steps needed to complete the pursuit of this research idea are to find the optimal hyperparameters and run experiments on the datasets presented in this thesis.

### 6.1.3 Other Privacy Mechanisms and Tighter Bounds

Firstly, while there exist other sophisticated state-of-the-art propagation techniques that can achieve tighter robustness bounds, such as Linear Bound Propagation (LBP), CROWN or MILP, they come with a significant computational overhead. Concerning privacy, as it was also mentioned in section 5.6, tighter differential privacy bounds can be computed by applying the improvements described in [5], which take into consideration the budget saving achieved by batching and subsampling . Other improvements that can be done regarding privacy is to consider other private training techniques, such as Homomorphic Encryption and Secure-Multi-Pary Computation. Although these are somewhat outside the scope of this thesis, because they are used in other contexts, it would be interesting to see whether other methods exhibit the same connections with robustness, uncertainty and performance as DP does, or if the link is different in nature.

## 6.2 Ethical Considerations

### 6.2.1 Protection of Personal Data

This study is exempt from ethical approval as it relies on secondary data sources, specifically the MNIST and PneumoniaMNIST datasets, which do not contain any personal data and are publicly accessible without the need for permission. These datasets are open source and licensed under the Creative Commons (CC) License, permitting distribution, remixing, adaptation, and building upon the material as long as proper attribution is provided. As this research involves only modifying the inputs by perturbing them and developing new training techniques noise for purposes of adversarial robustness, differential privacy and uncertainty estimation, no further consideration for the protection of personal data is needed.

### 6.2.2 Potential Misuse

Because this piece of work concerns itself with the study of properties of Safe and Trusted AI, all new developments that have been presented can be used to gain insight on probabilistic machine learning systems inner workings and exploit them in a malignant manner. Technically speaking, our concerns lie in the utility of black-box and white-box attacks, respectively. While the contributions of this thesis are helpful in limiting the amount of damage that can be done having access only to the inputs and outputs of a certain model, white-box attacks can be very efficient in finding vulnerabilities, precisely because a potential attacker knows everything about our model in such a scenario.

What is, however, perhaps even more important to consider in this section is the effect of such attacks in real-life scenarios. As with every advancement of any system's security aspect, gaining new insight is a double-edged sword, and can be used both for advancing the knowledge of humankind, as well as for destructive action. In particular, potential misuse can incur significant damage in critical industries such as healthcare or autonomous driving. For the first, if we consider a medical diagnosis system that recommends treatment (which can be very well be the case with using the PneumoniaMNIST dataset we examined), an attacker knowing what techniques were used for robustness, privacy and uncertainty, as well as the trade-off between them can craft specific attacks that fool the system and can potentially recommend treatment detrimental to a patient, causing complications or even death. Similarly, in the case of autonomous driving, if we consider a model that interprets road signs and steers a vehicle accordingly, an attacker can once again craft malignant attack vectors, such that the road sign is misinterpreted and accidents with potential human life loss are caused. Lastly, it is important to note the fact that successful attacks that target privacy break data protection regulations active in numerous states, identifying previously anonymous data, problem which is important and highly sensitive nowadays.

# Bibliography

[1] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. pages 1, 3, 4, 11, 25

[2] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018. pages 1, 4, 12, 18, 25, 26

[3] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017. pages 1, 14, 15, 43

[4] Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006. pages 2, 4

[5] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016. pages 2, 4, 15, 44

[6] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015. pages 2, 4, 7, 43

[7] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016. pages 2

[8] Liwei Song, Reza Shokri, and Prateek Mittal. Privacy risks of securing machine learning models against adversarial examples. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 241–257, 2019. pages 2, 43

[9] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011. pages 2, 4, 8, 10

[10] Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13, 1993. pages 4, 7

[11] Matthew Wicker, Luca Laurenti, Andrea Patane, Zhuotong Chen, Zheng Zhang, and Marta Kwiatkowska. Bayesian inference with certifiable adversarial robustness. In *International Conference on Artificial Intelligence and Statistics*, pages 2431–2439. PMLR, 2021. pages 4, 18, 25, 26, 29

[12] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. pages 4, 11, 25

[13] Qiyiwen Zhang, Zhiqi Bu, Kan Chen, and Qi Long. Differentially private bayesian neural networks on accuracy, privacy and reliability. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 604–619. Springer, 2022. pages 4, 16, 43

[14] Ossi Räisä, Antti Koskela, and Antti Honkela. Differentially private hamiltonian monte carlo. *arXiv preprint arXiv:2106.09376*, 2021. pages 4, 16, 18, 21, 36

[15] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989. pages 6

[16] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953. pages 8

[17] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970. pages 8

[18] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016. pages 11

[19] Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. *arXiv preprint arXiv:1711.07356*, 2017. pages 12

[20] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR, 2019. pages 12

[21] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 619–633, 2018. pages 14

[22] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014. pages 15

[23] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. pages 23

[24] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10 (1):41, 2023. pages 23

[25] Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *cell*, 172(5):1122–1131, 2018. pages 24, 25

[26] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *Advances in neural information processing systems*, 32, 2019. pages 43

[27] Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger Grosse. Noisy natural gradient as variational inference. In *International conference on machine learning*, pages 5852–5861. PMLR, 2018. pages 43

[28] Kazuki Osawa, Siddharth Swaroop, Mohammad Emtiyaz E Khan, Anirudh Jain, Runa Eschenhagen, Richard E Turner, and Rio Yokota. Practical deep learning with bayesian principles. *Advances in neural information processing systems*, 32, 2019. pages 43

[29] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 619–633, 2018. pages 43

[30] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015. pages 43