Proceedings of Machine Learning Research vol vvv:1–13, 2025

Risk-Averse Certification of Bayesian Neural Networks

Xiyue Zhang[1]* XIYUE.ZHANG@CS.OX.AC.UK
Zifan Wang[2] ZIFANW@KTH.SE
Yulong Gao[3] YULONG.GAO@IMPERIAL.AC.UK
Licio Romao[4] LICRO@DTU.DK
Alessandro Abate[1] ALESSANDRO.ABATE@CS.OX.AC.UK
Marta Kwiatkowska[1] MARTA.KWIATKOWSKA@CS.OX.AC.UK
[1] *Department of Computer Science, University of Oxford, UK*
[2] *Division of Decision and Control Systems, KTH Royal Institute of Technology, Sweden*
[3] *Department of Electrical and Electronic Engineering, Imperial College London, UK*
[4] *Department of Wind and Energy Systems, Technical University of Denmark, Denmark*

Abstract

In light of the inherently complex and dynamic nature of real-world environments, incorporating risk measures is crucial for the robustness evaluation of deep learning models. In this work, we propose a **R**isk-**A**verse **C**ertification framework for Bayesian neural networks called RAC-BNN. Our method leverages sampling and optimisation to compute a sound approximation of the output set of a BNN, represented using a set of template polytopes. To enhance robustness evaluation, we integrate a coherent distortion risk measure—Conditional Value at Risk (CVaR)—into the certification framework, providing probabilistic guarantees based on empirical distributions obtained through sampling. We validate RAC-BNN on a range of regression and classification benchmarks and compare its performance with a state-of-the-art method. The results show that RAC-BNN effectively quantifies robustness under worst-performing risky scenarios, and achieves tighter certified bounds and higher efficiency in complex tasks.

Keywords: Uncertainty, Bayesian neural networks, risk measure, probabilistic certification

1. Introduction

There has been growing interest in formal verification of neural networks (Huang et al., 2017; Katz et al., 2017; Zhang et al., 2018; Singh et al., 2019; Tjeng et al., 2019; Xu et al., 2020), in particular when deploying deep neural models to safety- and security-critical systems, autonomous vehicles (Bojarski et al., 2016; Codevilla et al., 2018), healthcare systems (Babak et al., 2015), and cyber security (Dahl et al., 2013; Shin et al., 2015). Different from deterministic neural networks which learn a fixed set of weights and biases from a set of training data, Bayesian neural networks (BNNs) provide a principled approach to modelling uncertainty (Neal, 2012) and learn a posterior distribution over these network parameters. During inference, BNNs quantify uncertainty and assign high uncertainty values to out-of-distribution inputs instead of being overconfident in wrong predictions (Kahn et al., 2017). At the same time, the stochastic nature of BNNs complicates certification, as both the model parameters and, as a result, the predictive outputs are probability distributions rather than point estimates. Even when applying relaxation-based certification techniques to BNNs, the computational complexity can increase drastically.

* Current address: School of Computer Science, University of Bristol, UK (xiyue.zhang@bristol.ac.uk)

© 2025 X. Zhang, Z. Wang, Y. Gao, L. Romao, A. Abate & M. Kwiatkowska.

In order to reliably deploy BNN solutions and reason about their safety in the presence of uncertainty, techniques have been developed to handle stochastic constraints within BNNs and compute certified bounds on their reachable outputs. These approaches typically fall into two categories: sampling-based techniques, which provides probabilistic guarantees (Cardelli et al., 2019; Wicker et al., 2020; Michelmore et al., 2020), and approximation-based techniques, which evaluate the robustness of BNNs by computing the expectation of the output distributions over an input set (Adams et al., 2023; Wicker et al., 2024b). While approximation techniques significantly increase the scalability of evaluating larger-size BNNs, they often introduce relaxation losses to the certified output range. Such relaxation can lead to conservative output bounds, limiting the precision of robustness evaluations. Furthermore, existing works focus on bounding the expectation of the entire output distribution. However, in real-world decision-making scenarios, considering the average performance over the full distribution may not suffice. Instead, it is important to account for challenging scenarios by adopting a *risk-averse* perspective; that is, to evaluate robustness under adverse conditions (e.g., the most adversarially unstable 25% cases).

In this work, we highlight the importance of a risk-averse perspective for BNN certification. Specifically, we propose a principled approach to BNN certification that incorporates coherent distortion risk measures – Conditional Value at Risk (CVaR) (Rockafellar et al., 2000) – which enables flexible and targeted evaluation of BNN performance. The key idea of our method is to sample the input points and the parameters of the BNN weights, obtaining the empirical output distribution, to compute a sound approximation of the output set (using template polytopes) and certified CVaR bounds with probabilistic guarantees. We implement our method as a prototype tool, RAC-BNN, and demonstrate that it achieves tighter certification bounds with better efficiency than state-of-the-art techniques on a range of regression and classification benchmarks. To the best of our knowledge, RAC-BNN is the only method capable of computing certified bounds under different risk levels (denoted by α), enabling the flexibility between analysing average robustness over the entire output distribution ($\alpha = 1$) and evaluating robustness against worst-performing outcomes ($\alpha < 1$).

2. Related Work

We now discuss closely related works in the certification of BNNs and risk-averse learning.

Robustness Certification of BNNs The last decade has witnessed a growing interest in formal certification of neural networks, including complete verification methods based on constraint solving (Huang et al., 2017; Katz et al., 2017; Tjeng et al., 2019) and incomplete verifiers based on convex relaxation (Zhang et al., 2018; Singh et al., 2019; Xu et al., 2020). However, these methods all assume deterministic neural networks with fixed weights and thus cannot be directly applied to certify BNNs. To this end, a series of certification techniques have been proposed for the certification of BNNs (Cardelli et al., 2019; Wicker et al., 2020; Adams et al., 2023; Wicker et al., 2024b).

Cardelli et al. (2019) proposed a statistical approach to estimate the probability of the existence of adversarial examples with *a priori* guarantees by viewing the robustness of a BNN as a Bernoulli random variable. Wicker et al. (2020) focused on the probabilistic robustness of BNNs, that is, the probability of the sampled weights from the posterior for which the resulting deterministic neural network satisfying a safety property. The method computes a certified lower bound for the probabilistic safety based on relaxation techniques of interval and linear bound propagation. These bounds are later generalised in Wicker et al. (2021), where they are applied to bound

sequential decisions on BNN-based models, and specifically in Wicker et al. (2024a) for reach-avoid (bounded-until) specifications. Wicker et al. (2024b) further investigated decision robustness, which focuses on the decision step aligned with Bayesian decision theory, as also used in this work, and proposed a unified approach to compute certified lower and upper bounds for both probabilistic robustness and decision robustness. Adams et al. (2023) leveraged dynamic programming to bound the output range of BNNs over an input region. There are also certification methods that are able to reason about the robustness of the closed-loop systems where BNNs are applied for decision-making. Michelmore et al. (2020) introduced a statistical framework to evaluate the safety of end-to-end BNN controllers in autonomous driving.

Risk-Averse Learning Risk-averse learning has emerged as a critical area in machine learning, particularly for applications where decisions have significant consequences under uncertainty. Traditional machine learning models often focus on minimising expected loss, which may not adequately capture the potential for rare but severe adverse outcomes. To solve this, researchers have investigated risk-averse approaches that consider not just the expected performance but also the tail risks Vitt et al. (2019); Lakdawalla and Phelps (2021); O'Donoghue and Somerville (2018); Tamar et al. (2015). For example, Vitt et al. (2019) introduced a risk-averse classification framework leveraging coherent risk measures to address class-specific misclassification risks, demonstrating its effectiveness through applications to support vector machines. In healthcare engineering, Lakdawalla and Phelps (2021) introduced the risk-adjusted cost-effectiveness framework, which integrates risk aversion and diminishing returns into health technology assessments. By accounting for tail risks and variability in treatment outcomes, this approach addresses limitations of traditional cost-effectiveness analysis, particularly for severe illnesses and uncertain interventions.

3. Preliminaries and Problem Formulation

Next we introduce the necessary background and notations to be employed throughout the paper.

Notation We denote the input space by $\mathcal{X} \subseteq \mathbb{R}^m$, the output space by $\mathcal{Y} \subseteq \mathbb{R}^n$, and the parameter space by $\mathcal{W} \subseteq \mathbb{R}^p$. We denote by $\mathcal{P}(\mathcal{X})$ the set of probability distribution over $\mathcal{X}$, that is, $\mathcal{P}(\mathcal{X}) = \{\mu : \int_{\mathcal{X}} \mu(d\xi) = 1, \mu \geq 0\}$, and similarly for $\mathcal{Y}$ and $\mathcal{W}$. For a finite collection of points $\{x_1, \ldots, x_N\}$ in $\mathcal{X}$, we denote the corresponding empirical distribution as $\hat{\mu}_N(x) = \frac{1}{N}\sum_{i=1}^{N} \delta_{x_i}(x)$, where $\delta_{x_i}(x)$ denotes the Dirac measure centered at x_i. Similarly, for points in the output space $\mathcal{Y}$ and parameter space $\mathcal{W}$, we denote the empirical distribution by $\hat{\nu}_N$ and $\hat{\lambda}_N$, respectively. Given two probability distributions μ and μ' defined on the input space, i.e., $\mu, \mu' \in \mathcal{P}(\mathcal{X})$, we denote by $W_1(\mu, \mu')$ the *type-1 Wasserstein distance* between these measures, defined as

$$W_1(\mu, \mu') = \inf_{\pi \in \Pi(\mu, \mu')} \int_{\mathcal{X} \times \mathcal{X}} \|\xi_1 - \xi_2\| d\pi(\xi_1, \xi_2), \tag{1}$$

where $\Pi(\mu, \mu')$ is the set of couplings (or joint distributions) with marginals given by μ and μ'.

Conditional Value at Risk (CVaR) is a coherent risk measure. For a random variable X with the cumulative distribution function (CDF) denoted by F and a specified risk level $\alpha \in (0, 1]$, the CVaR value is given by $\text{CVaR}_\alpha[X] = \mathbb{E}_F[X | X > \text{VaR}_\alpha[X]]$, where $\text{VaR}_\alpha[X] = \inf\{y : F_X(y) \geq 1 - \alpha\}$ represents the $1 - \alpha$ quantile of the distribution, also known as the Value at Risk (VaR). Intuitively, CVaR captures the average of the worst-case outcomes within the upper $\alpha\%$ of the distribution.

Bayesian Neural Networks In this section, we define Bayesian neural networks (BNNs) and review related concepts using the notation introduced in the previous section.

Definition 1 (Bayesian Neural Network) *Given a distribution $\lambda \in \mathcal{P}(\mathcal{W})$ over the parameter space $\mathcal{W}$, a Bayesian Neural Network (BNN) is defined as a continuous stochastic function $f : \mathcal{X} \times \mathcal{W} \mapsto \mathcal{Y}$, where the weight w is sampled from the distribution λ, i.e., $w \sim \lambda$.*

In the training of BNNs, we start with a prior distribution $p(w)$ over the parameters w and then compute the posterior distribution $p(w|\mathcal{D})$ conditioned on dataset $\mathcal{D} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} : i = 1, \ldots, N\}$. Note that the measure $\lambda \in \mathcal{P}(\mathcal{W})$ in Definition 1 refers to the posterior distribution $p(w|\mathcal{D})$. With dataset $\mathcal{D}$ observed, the prior distribution of a BNN is updated according to the likelihood, $p(\mathcal{D}|w) = \prod_{i=1}^{n_\mathcal{D}} p(y_i|x_i, w)$, which models how likely the outputs are observed under the stochasticity of model parameters and the inputs. The posterior distribution, given the dataset, is then computed by virtue of the Bayes formula, i.e., $p(w|\mathcal{D}) \propto p(\mathcal{D}|w)p(w)$. In practice, the posterior distribution $p(w|\mathcal{D})$ can be obtained by different inference techniques, e.g., Hamiltonian Monte Carlo (HMC) (Neal, 2012), Variational Inference (VI) (Blundell et al., 2015), and Monte Carlo Dropout (MCD) (Gal and Ghahramani, 2016).

The posterior $p(w|\mathcal{D})$ then induces the distribution over outputs called the posterior predictive distribution for an input point x^*, which is defined as $p(y^*|x^*, \mathcal{D}) = \int p(y^*|x^*, w)p(w|\mathcal{D})dw$. The final decision is obtained using Bayesian decision theory for regression and classification, which selects the value $\hat{y}$ that minimises the corresponding loss function $\mathcal{L}$ averaged over the predictive distribution: $\hat{y} = \arg\min_y \int_{\mathbb{R}^n} \mathcal{L}(y, y^*)p(y^*|x^*, \mathcal{D})dy^*$.

Motivating Example In this section, we present a motivating example to demonstrate why investigating the expectation of the entire distribution alone is insufficient to evaluate the performance of BNNs. In high-stakes applications, outputs that deviate significantly from the safe region and lead to catastrophic consequences are unacceptable, even if their probability of occurrence is low. In such cases, relying solely on the expectation of the outputs fails to account for the risks, as illustrated in the following example.

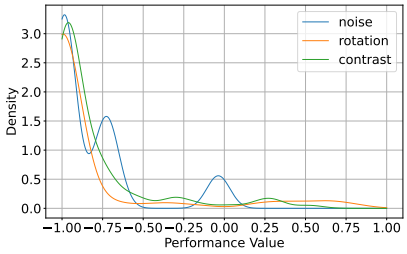

Figure 1: Tail distributions exist in Bayesian neural networks when recognising images with different types of perturbations.

We consider the MNIST classification task and visualise the empirical output distribution of a BNN under three types of perturbations: Gaussian noise applied to all image pixels, rotation that alters image orientation, and changes in brightness contrast. To evaluate the robustness performance, we define the function $h(y) = \max_{t \in [10] \setminus c} y_t - y_c$, where y_t denotes the random variable of the BNN output for the second-largest probability class, and y_c denotes the BNN output for the ground-truth class c. The value of y ranges from -1 to 1, with -1 indicating the BNN is robust and correctly classifies the input, while 1 indicates the BNN makes a confident incorrect decision. A value $h(y) > 0$ indicates that the BNN is not robust to the perturbations.

As shown in Figure 1, we observe tail distribution bumps under Gaussian noise within the interval around $[-0.1, 0.1]$, contrast perturbations within the interval around $[0.1, 0.3]$ and under rotation perturbations lying around $[0.25, 0.75]$. These tails correspond to cases where the BNN

confidently makes incorrect predictions. In risk-sensitive or safety-critical applications, such tail risks can lead to severe consequences. Standard expected values fail to capture the reliability of a BNN in these high-risk scenarios, highlighting the need for risk-averse certification techniques.

3.1. Problem Statement

Consider a BNN f, with the input $x \sim \mu$ and the parameters $w \sim \lambda$, and the output given by $y = f(x, w)$. Let ν denote the distribution of y. We assume that the output set $\mathcal{Y}$ is compact. This assumption holds for many tasks of BNNs. One can further render this assumption true by mapping the output to a prescribed compact set. For a given risk-averse level α and an evaluation function $h : \mathcal{Y} \to \mathbb{R}$, we define the risk-averse evaluation as $\phi^{\texttt{Perf}} = \text{CVaR}_{\alpha, y\sim\nu}[h(y)]$. In this work, we evaluate a BNN through the following problems.

1. Output support set computation: approximate $\mathcal{Y} = \{f(x, w) \mid x \in \mathcal{X}, w \in \mathcal{W}\}$;
2. Risk-averse evaluation: compute certified bounds on the CVaR value of the BNN output.

In the motivating example, evaluating the performance of the BNN involves addressing two interrelated tasks: approximating the output set of y_t and y_c, and computing the CVaR value of the performance function h. Approximating the output set enables a visual assessment of the BNN's outputs, allowing us to determine whether they fall within a safe region and meet desirable criteria. However, the set approximation alone cannot capture the probabilistic information about the likelihood of outputs lying in the safe region. The second task, i.e., computing the CVaR value, addresses this drawback, as it quantifies the tail risks by focusing on the most extreme and potentially hazardous outcomes. Together, these tasks provide a comprehensive framework for assessing the reliability and robustness of the BNN in high-stakes applications, where both the nature of the outputs and their risk profiles are crucial considerations.

4. Methodology

4.1. Output Set Approximation

In this section, we present our sampling-based solution to approximate the output support set. According to the posterior distribution λ and the input distribution μ, we first collect a group of i.i.d. sampled inputs $x_i \in \mathcal{X}$ and a group of i.i.d. sampled parameters $w_j \in \mathcal{W}$. Based on these collected samples, we can then compute the corresponding output samples $y_{ij} = f(x_i, w_j)$. We rewrite the output samples as y_k for notation simplicity and use N to denote the total number of samples.

For the output set approximation, we aim to compute a convex approximation of $\mathcal{Y}$. Leveraging the output samples, we build the approximation by taking the intersection of all half-spaces that contain $\{y_k\}_{k=1}^N$. This convex hull for the output samples and the tractable over-approximations can be conveniently represented as a template polytope, which provides high-confidence guarantees for the approximation gap by applying the scenario optimisation theory to our problem.

Consider a convex template polytope $\mathbb{V} = \{z \in \mathbb{R}^n \mid Vz \leq \mathbf{1}\}$ where $V \in \mathbb{R}^{L\times n}$ and L is the number of half spaces or inequalities. Given the set $\mathbb{V}$, and $\boldsymbol{\theta} \in \mathbb{R}^L$, we introduce a parameterised set in the form of $\mathcal{H}(\boldsymbol{\theta}) := \{z \in \mathbb{R}^n \mid Vz \leq \boldsymbol{\theta}\}$. We now approximate the output set $\mathcal{Y}$ by computing the optimal parameterised set $\mathcal{H}(\boldsymbol{\theta}^\star_N)$ with respect to the output samples, where $\boldsymbol{\theta}^\star_N$ is

the optimal solution to following optimisation problem

$$\begin{cases} \min\limits_{\boldsymbol{\theta}\in\mathbb{R}^L} & \mathbf{1}^T\boldsymbol{\theta} \\ \text{s.t} & Vy_k \leq \boldsymbol{\theta}, k = 1, \cdots, N. \end{cases} \tag{2}$$

The optimisation result is presented in the following proposition. The proof can be found in (Zhang et al., 2024).

Proposition 2 *The optimal solution $\boldsymbol{\theta}_N^\star$ to the optimisation problem in Equation (2) is*

$$[\boldsymbol{\theta}_N^\star]_i = \max_{k=1,\cdots,N} [V]_i y_k, \tag{3}$$

where $[V]_i$ denotes the i-th row of V. Let $\hat{\mathcal{Y}}_N = \mathcal{H}(\boldsymbol{\theta}_N^\star)$. Given $\epsilon_1 \in (0,1)$, $\beta_1 \in (0,1)$, and the Euler's constant e, if $N \geq \frac{1}{\epsilon_1}\frac{e}{e-1}\Big(\ln\frac{1}{\beta_1} + n + L\Big)$, then, with probability no less than $1-\beta_1$, $\mathbb{P}[y \in \mathcal{Y} : y \notin \hat{\mathcal{Y}}_N] = \int_{\mathcal{Y}\setminus\hat{\mathcal{Y}}_N} \nu(d\xi) \leq \epsilon_1$.

Proposition 2 provides a statistical bound on the discrepancy between the estimated output set $\hat{\mathcal{Y}}_N$ and the true output set $\mathcal{Y}$. The error bound ϵ_1 has an inverse relationship with N, indicating that achieving tighter error bounds necessitates a significantly larger sample size. The dimensionality of the sample space, n, and of the structured polytope, L, contributes linearly to the sample size, reflecting the increased effort needed to address higher-dimensional or structurally complex problems. Besides, the term $\ln\frac{1}{\beta_1}$ introduces a logarithmic dependence on the confidence level. These results indicate that tighter precision and higher confidence come at the cost of increased computational and data collection demands.

4.2. Risk-Averse Evaluation

Given the output samples $\{y_k\}_{k=1}^N$, we define $\hat{\nu}_N(y) = \frac{1}{N}\sum_{i=1}^N \delta_{y_i}(y)$ as the empirical distribution of ν. Our goal is to estimate the value of $\text{CVaR}_{\alpha,y\sim\nu}[h(y)]$, which represents the CVaR value of the performance function h at level α under the distribution ν. Specifically, we aim to construct a certified bound for this value with probabilistic confidence guarantees, which is presented in the following proposition. The proof can be found in (Zhang et al., 2024).

Proposition 3 *Suppose that $h(y)$ is L_0-Lipschitz continuous in y. Then, we have $|CVaR_{\alpha,y\sim\hat{\nu}_N}[h(y)] - CVaR_{\alpha,y\sim\nu}[h(y)]| \leq \frac{L_0}{\alpha}\epsilon_2(\beta)$, with probability at least $1-\beta$, where $\epsilon_2(\beta) = \rho(\mathcal{Y})(C^* N^{-\frac{1}{n}} + \sqrt{n}(2\ln\beta^{-1})^{\frac{1}{2}} N^{-\frac{1}{2}})$, $C^* = \sqrt{n}2^{(n-2)/2}\Big(\frac{1}{1-2^{1-n/2}} + 2\Big)$, $\rho(\mathcal{Y})$ is the diameter of the support of y.*

Proposition 3 states that the distance between the empirical CVaR value and the true CVaR value is related to the sample size N. Given a target certification range H with probability at least $1-\beta$, the required number of samples is given by $\Big(\frac{L_0\rho(\mathcal{Y})(C^*+\sqrt{n}(2\ln\beta^{-1})^{\frac{1}{2}})}{\alpha H}\Big)^n$, where n denotes the output dimension of the performance function. The sample size N is inversely proportional to H^n, indicating that as the certified bound tightness H decreases, N increases exponentially with an exponent related to n. Additionally, a smaller α, which focuses on the largest $\alpha\%$ of the distribution, necessitates a larger number of samples to accurately estimate the expected value within this specific portion of the distribution. Finally, N depends logarithmically on the confidence level β, meaning that achieving higher confidence (smaller β) necessitates an increase in the sample size.

5. Experiments

In this section, we present the experiment setup and evaluation results of the proposed approach.

Benchmark and Baseline. Following recent work (Adams et al., 2023), we evaluate our method on three regression tasks, including a 1D noisy sine dataset where the BNN is trained on samples from 1D sine function with additive noise, a 2D equivalent of Noisy Sine, and the Kin8nm dataset where the BNN is trained on a dataset of state-space readings for the dynamics of an eight-link robot arm. We further investigate the performance of our method on classification tasks where BNNs are trained on the MNIST and Fashion-MNIST datasets. For each experiment, we evaluate the risk-averse robustness of BNN models against different noise and attack settings by computing the certified CVaR values under a range of risk levels. Specifically, for risk level 1, we consider the state-of-the-art method BNN-DP in Adams et al. (2023) for robustness certification of BNNs and conduct performance comparison in terms of the tightness of the certified bounds and the computation overhead. All experiments are conducted on a cluster with Intel Xeon Gold 6252 2.1GHz CPU, and NVIDIA 2080Ti GPU.

Evaluation Metric. To evaluate the certification performance of our approach, we use γ-robustness (Adams et al., 2023), which computes the difference between the upper and lower bounds on the expectation outptus of BNNs with regard to a property formulated by the performance function. A smaller γ-robustness value implies a tighter certified bound computation.

5.1. Evaluation Results

5.1.1. RQ1: IS OUR APPROACH EFFECTIVE IN CHARACTERISING THE OUTPUT SET?

Consider again the specific case of the MNIST dataset under two distinct perturbation scenarios: rotation perturbation and noise perturbation. Figure 2 illustrates the relationship between two output variables of the BNN: y_c representing the output probability for the true class c, and y_t representing the output probability for the second-most likely class. The black dashed line $y_t = y_c$ serves as a threshold: if a data point lies below this line, the BNN classifies correctly, as the true class y_c has a higher probability than y_t. Points marked in red and blue correspond to samples affected by rotation and noise perturbations, respectively. Given a desired error 0.05 with a high confidence level of 95%, and using Proposition 2, we select $L = 16$ and compute the required number of samples to be 665.

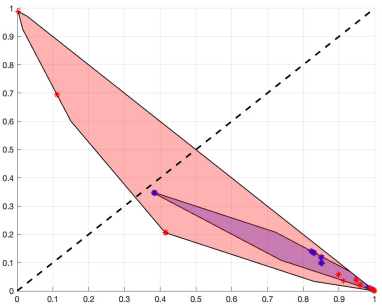

Figure 2: Output set computation for Bayesian neural networks when recognising images with different types of perturbations.

To systematically assess the BNN's robustness under such conditions, we propose computing the output support set, defined as the convex hull of all possible output pairs (y_t, y_c) under each perturbation type. The output support set provides a visual and quantitative measure of BNN performance. If the support set lies predominantly below the black line, the BNN demonstrates robustness against the perturbation. Conversely, if a large portion of the support set extends above the black line, it indicates vulnerability. Obviously, from Figure 2, the BNN shows robustness to noise perturbations but exhibits vulnerability to rotation perturbations. This geometric perspective enables a straightforward evaluation of the BNN's performance.

5.1.2. RQ2: IS OUR APPROACH EFFECTIVE IN CHARACTERISING RISK-AVERSE PERFORMANCE OF BNNS?

To answer this question, we evaluate the effectiveness of our approach in characterising certified CVaR bounds under a range of risk levels. A comparison of certification performance for the entire output distribution ($\alpha = 1$) with the baseline method *BNN-DP* is deferred to Section 5.1.3.

Table 1: Certified CVaR bounds under different confidence and risk levels for regression tasks.

Tasks	$\beta = 0.05$			$\beta = 0.01$		
	$\alpha = 1$	$\alpha = 0.5$	$\alpha = 0.25$	$\alpha = 1$	$\alpha = 0.5$	$\alpha = 0.25$
1D Noisy Sine	0.074 ± 0.1	0.146 ± 0.1	0.147 ± 0.1	0.058 ± 0.1	0.148 ± 0.1	0.165 ± 0.1
2D Noisy Sine	0.081 ± 0.1	0.173 ± 0.1	0.268 ± 0.1	0.077 ± 0.1	0.203 ± 0.1	0.272 ± 0.1
Kin8nm	0.080 ± 0.1	0.088 ± 0.1	0.097 ± 0.1	0.079 ± 0.1	0.088 ± 0.1	0.103 ± 0.1

Regression Benchmarks. We first evaluate the proposed method on BNNs trained with three regression datasets: 1D Noisy Sine, its 2D equivalent, and Kin8nm. To assess the effectiveness of our method, we compute certified CVaR bounds under different levels of risk ($\alpha = 0.25, 0.5, 1$) and confidence guarantees ($\beta = 0.05, 0.1$). Table 1 summarises the evaluation results for the regression tasks. For each configuration, the CVaR value represents the estimated expectation of the subset of the distribution of the performance function. The maximum deviation from the true value is constrained by a pre-defined limit of $H = 0.1$. For the 1D Noisy Sine task, we apply noise up to 0.01 around the input point $\pi/2$ and the property of interest is the deviation from the ground truth output, with the performance function defined as $h(y) = 1 - y$. As expected, when we focus on the worst-case outcomes, the CVaR values, which indicate the average deviation in the subset scenarios, show an increase. For the 2D Noisy Sine task, we use the same perturbation noise (up to 0.01), except that the perturbations are applied to two input features rather than one. The same performance function, $h(y) = 1 - y$, is used for evaluation. In this case, CVaR values show slight increases across all settings compared to the 1D dataset, largely due to the added noises to the 2D input space.

For the Kin8nm dataset, we simulate input perturbation noise up to 0.01 for eight input features and evaluate the output deviation from the ground truth, formulated using the performance function $h(y) = |y^* - y|$ where y^* indicates the ground-truth value. Notably, the estimated CVaR values demonstrate small variance across difference risk levels ($\alpha = 1, 0.5, 0.25$). As previously discussed, a consistent performance under varying risk levels is the ideal risk-averse robustness we aim to achieve for BNN certification.

Table 2: Certified CVaR bounds for different attacks on classification tasks.

Tasks	**CVaR Level**	L_∞ **noise**	**Rotation**	**Contrast**
MNIST	$\alpha = 1$	-0.999 ± 0.1	-0.998 ± 0.1	-0.997 ± 0.1
	$\alpha = 0.5$	-0.999 ± 0.1	-0.998 ± 0.1	-0.993 ± 0.1
	$\alpha = 0.25$	-0.999 ± 0.1	-0.997 ± 0.1	-0.986 ± 0.1
FASHION	$\alpha = 1$	-0.191 ± 0.1	-0.119 ± 0.1	-0.152 ± 0.1
	$\alpha = 0.5$	0.443 ± 0.1	0.525 ± 0.1	0.562 ± 0.1
	$\alpha = 0.25$	0.885 ± 0.1	0.964 ± 0.1	0.849 ± 0.1

Classification Benchmarks. Table 2 summarises the certified bounds for classification tasks under different levels of risk α. We evaluate the risk-averse robustness of BNNs against three types of attacks: L_∞ attack, where perturbation noise up to a specific limit is applied to all image pixels, and two geometric attacks, rotation (altering the image's orientation) and contrast (changing its brightness and contrast). For each configuration, the CVaR value represents the estimated expected value of the performance function, w.r.t. a property of interest, over (the subset of) the distribution.

For the MNIST dataset, the perturbation limit for the L_∞ attack is set to 0.1, the rotation range to $[-45°, 45°]$, and the contrast factor to 0.5. Robustness is evaluated by checking whether the predicted labels remain consistent with the ground truth label, denoted as c^* among the 10 classes. The corresponding performance function is defined as $h(y) = \max_{i \in [10] \setminus c^*} y_i - y_{c^*}$, where c^* is the ground-truth label. The BNN is robust to perturbations if $h(y) < 0$. For both L_∞ and *rotation* attacks, our approach reveals that the trained BNN achieves strong certified risk-averse robustness, with very low variance in performance across the entire output distribution and the worst 25% outcomes. Under the *contrast* attack, the BNN shows small variance when evaluating the overall expectation and the worst-performing 25% subset. Nonetheless, the expected values for the most challenging cases remain far below 0, demonstrating the BNN's risk-averse robustness across all attack types.

For the FASHION dataset, we set the perturbation limit for the L_∞ attack to 0.1, the rotation range to $[-15°, 15°]$, and the contrast factor to 0.9. As with the MNIST dataset, We investigate output label consistency, formulated by the performance function $h(y) = \max_{i \in [10] \setminus c^*} y_i - y_{c^*}$. Note that the expectation values over the full output distribution for all three attack types are negative, indicating overall certified robustness. However, when focusing on the worst-performing 50% and 25% subset of the output distribution, the expectation values are all positive, highlighting the necessity of risk-averse evaluation for BNNs. Compared with the BNN for the MNIST dataset, this BNN demonstrates a lack of robustness, especially in scenarios requiring a risk-averse approach.

Table 3: Comparison with SOTA in tightness of the certified bounds and computation time.

Method	1D Noisy Sine		Kin8nm		MNIST	
	γ-robustness	Time (s)	γ-robustness	Time (s)	γ-robustness	Time (s)
BNN-DP	**0.065**	6.148	0.137	7.886	1.572	24.956
RAC-BNN (0.1)	0.2	**0.815**	0.2	**0.984**	0.2	**1.064**
RAC-BNN (0.05)	0.1	1.640	**0.1**	1.893	**0.1**	4.099

5.1.3. RQ3: WHAT IS THE CERTIFICATION PERFORMANCE OF OUR APPROACH IN TIGHTNESS AND EFFICIENCY?

We perform comparison experiments with *BNN-DP* in computing certified bounds for the entire output distribution (corresponding to $\alpha = 1$). The evaluation focuses on two metrics: the tightness of the certified bounds (γ-robustness) and the computation time of the certification procedure. Table 3 summarises the results for both regression and classification tasks. In the evaluation, we present the performance results of our method (RAC-BNN) under two certification range settings, $H = 0.1$ and $H = 0.05$, and the γ-robustness is $2H$.

The evaluation results demonstrate that our method improves both the tightness and efficiency of computing certified bounds. For the $1D$ Noisy Sine dataset, while *BNN-DP* demonstrates com-

petitive performance in certification tightness, to achieve comparable tightness with our method requires further tightening of the certification range H. On the other hand, our method significantly reduces the computational overhead for certification. As the input dimension increases and the task gets more complex, the advantages of our method in certification performance become more significant than the baseline method. Specifically, when $H = 0.05$, our method surpasses the baseline in tightness for all remaining tasks. Particularly, for the MNIST dataset, our method improves the tightness of the certified bounds by 87.3% and 93.6% for $H = 0.1$ and $H = 0.05$, respectively. In terms of computation time, our method achieves a reduction of 76.0% and 83.6% for Kin8nm and MNIST under the $H = 0.05$ setting.

Table 4: Sample complexity for different hyper-parameters.

Task	γ**-robustness**	$\beta = 0.05$			$\beta = 0.01$		
		$\alpha = 1$	$\alpha = 0.5$	$\alpha = 0.25$	$\alpha = 1$	$\alpha = 0.5$	$\alpha = 0.25$
Kin8nm	$H = 0.1$	17648	69923	278354	26950	107129	427177
	$H = 0.05$	69923	278354	1110733	107129	427177	1706025
MNIST	$H = 0.1$	2520	9835	38844	3808	14986	59445
	$H = 0.05$	9835	38844	154379	14986	59445	236783

5.1.4. RQ4: WHAT IS THE SAMPLE COMPLEXITY OF OUR APPROACH?

In Section 4.2, Proposition 3 provides a theoretical relation between the sampling complexity and key hyper-parameters, including precision tightness, risk levels and confidence errors. For the benchmark tasks, the performance function $h(y)$ – which measures the output difference to the ground-truth value for regression tasks or label consistency for classification tasks – is a scalar random variable. With the dimension n reduced to 1, according to Proposition 3, the required sample size N is inversely proportional to H and the risk level α, while depending logarithmically on the confidence level β.

To demonstrate the practical sampling feasibility of our approach, we compute the required sample size under different configurations to evaluate the impact of individual parameters on sampling complexity. The results are summarised in Table 4. Across all configurations, the sampling complexity is manageable, with the largest required sample size reaching 10^6. This is the case when evaluating a narrow 25% worst-performing portion of the distribution with a tightness bound of 0.05 and a 99% confidence guarantee.

6. Conclusion

We introduce a novel risk-averse evaluation method for computing certified bounds of Bayesian neural networks using the coherent risk measure CVaR. By leveraging sampling and optimisation, our approach approximates the output distribution and bounds the CVaR values with probabilistic guarantees. We implement this method in a tool, RAC-BNN, and demonstrate its ability to compute sound approximations of output sets and quantify robustness under worst-performing conditions. The results show that RAC-BNN achieves improved certification tightness and better efficiency compared to existing methods. An interesting future direction is to extend our approach to the safety evaluation of closed-loop dynamical systems with BNN controllers.

Acknowledgments

MK and XZ received partial support from ELSA: European Lighthouse on Secure and Safe AI project (Grant No. 101070617 under UK guarantee) and the ERC under the European Union's Horizon 2020 research and innovation program (FUN2MODEL, Grant No. 834115).

References

Steven Adams, Andrea Patane, Morteza Lahijanian, and Luca Laurenti. BNN-DP: robustness certification of bayesian neural networks via dynamic programming. In *International Conference on Machine Learning, ICML 2023*, volume 202 of *Proceedings of Machine Learning Research*, pages 133–151. PMLR, 2023.

Alipanahi Babak, Andrew Delong, Matthew T. Weirauch, and Brendan J. Frey. Predicting the sequence specificities of dna- and rna-binding proteins by deep learning. *Nature Biotechnology*, 33(8):831–838, 2015. doi: https://doi.org/10.1038/nbt.3300.

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.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *CoRR*, abs/1604.07316, 2016. URL http://arxiv.org/abs/1604.07316.

Luca Cardelli, Marta Kwiatkowska, Luca Laurenti, Nicola Paoletti, Andrea Patane, and Matthew Wicker. Statistical guarantees for the robustness of Bayesian neural networks. *Proceedings of International Joint Conferences on Artificial Intelligence*, 2019.

Felipe Codevilla, Matthias Müller, Antonio M. López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation*, pages 1–9. IEEE, 2018. doi: 10.1109/ICRA.2018.8460487.

George E. Dahl, Jack W. Stokes, Li Deng, and Dong Yu. Large-scale malware classification using random projections and neural networks. In *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3422–3426. IEEE, 2013. doi: 10.1109/ICASSP.2013.6638293.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016.

Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety verification of deep neural networks. In *Proceedings of International Conference on Computer Aided Verification*, volume 10426 of *Lecture Notes in Computer Science*, pages 3–29. Springer, 2017. doi: 10.1007/978-3-319-63387-9_1.

Gregory Kahn, Adam Villaflor, Vitchyr Pong, Pieter Abbeel, and Sergey Levine. Uncertainty-aware reinforcement learning for collision avoidance. *arXiv preprint arXiv:1702.01182*, 2017.

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

Darius N Lakdawalla and Charles E Phelps. Health technology assessment with diminishing returns to health: the generalized risk-adjusted cost-effectiveness (grace) approach. *Value in Health*, 24 (2):244–249, 2021.

Rhiannon Michelmore, Matthew Wicker, Luca Laurenti, Luca Cardelli, Yarin Gal, and Marta Kwiatkowska. Uncertainty quantification with statistical guarantees in end-to-end autonomous driving control. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 7344–7350. IEEE, 2020.

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

Ted O'Donoghue and Jason Somerville. Modeling risk aversion in economics. *Journal of Economic Perspectives*, 32(2):91–114, 2018.

R Tyrrell Rockafellar, Stanislav Uryasev, et al. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–42, 2000.

Eui Chul Richard Shin, Dawn Song, and Reza Moazzezi. Recognizing functions in binaries with neural networks. In *Proceedings of the 24th USENIX Security Symposium*, pages 611–626. USENIX Association, 2015.

Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, pages 1–30, 2019.

Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Policy gradient for coherent risk measures. *Proceedings of Annual Conference on Advances in Neural Information Processing Systems*, 28, 2015.

Vincent Tjeng, Kai Yuanqing Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. In *Proceedings of International Conference on Learning Representations*. OpenReview.net, 2019.

Constantine Alexander Vitt, Darinka Dentcheva, and Hui Xiong. Risk-averse classification. *Annals of Operations Research*, pages 1–35, 2019.

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

Matthew Wicker, Luca Laurenti, Andrea Patane, Nicola Paoletti, Alessandro Abate, and Marta Kwiatkowska. Certification of iterative predictions in Bayesian neural networks. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference*

on Uncertainty in Artificial Intelligence, volume 161 of *Proceedings of Machine Learning Research*, pages 1713–1723. PMLR, 27–30 Jul 2021. URL https://proceedings.mlr.press/v161/wicker21a.html.

Matthew Wicker, Luca Laurenti, Andrea Patane, Nicola Paoletti, Alessandro Abate, and Marta Kwiatkowska. Probabilistic reach-avoid for bayesian neural networks. *Artificial Intelligence*, 334:104–132, 2024a. doi: https://doi.org/10.1016/j.artint.2024.104132. URL https://www.sciencedirect.com/science/article/pii/S0004370224000687.

Matthew Wicker, Andrea Patane, Luca Laurenti, and Marta Kwiatkowska. Adversarial robustness certification for bayesian neural networks. In *Proceedings of International Symposium on Formal Methods*, pages 3–28. Springer, 2024b.

Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. In *Proceedings of Annual Conference on Advances in Neural Information Processing Systems*, 2020.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *Proceedings of Annual Conference on Advances in Neural Information Processing Systems*, pages 4944–4953, 2018.

Xiyue Zhang, Zifan Wang, Yulong Gao, Licio Romao, Alessandro Abate, and Marta Kwiatkowska. Risk-averse certification of bayesian neural networks. 2024. URL https://drive.google.com/file/d/1H_RXVJSLtvx5irRNv3wtA_5AGsiBQipp/view?usp=sharing.