Robust Bayesian Neural Networks by Spectral Expectation Bound Regularization

Jiaru Zhang[1] Yang Hua[2] Zhengui Xue[1] Tao Song[1*] Chengyu Zheng[1] Ruhui Ma[1] Haibing Guan[1†]
[1]Shanghai Jiao Tong University [2]Queen's University Belfast
{jiaruzhang, zhenguixue, songt333, zhengcy, ruhuima, hbguan}@sjtu.edu.cn, Y.Hua@qub.ac.uk

Abstract

Bayesian neural networks have been widely used in many applications because of the distinctive probabilistic representation framework. Even though Bayesian neural networks have been found more robust to adversarial attacks compared with vanilla neural networks, their ability to deal with adversarial noises in practice is still limited. In this paper, we propose Spectral Expectation Bound Regularization (SEBR) to enhance the robustness of Bayesian neural networks. Our theoretical analysis reveals that training with SEBR improves the robustness to adversarial noises. We also prove that training with SEBR can reduce the epistemic uncertainty of the model and hence it can make the model more confident with the predictions, which verifies the robustness of the model from another point of view. Experiments on multiple Bayesian neural network structures and different adversarial attacks validate the correctness of the theoretical findings and the effectiveness of the proposed approach.

1. Introduction

Bayesian neural networks [8, 29] provide a probabilistic view of deep learning frameworks by treating the model weights as random variables. One of the profound advantages of a Bayesian neural network is that it can provide both the aleatoric uncertainty and the epistemic uncertainty estimations because of the probabilistic representation of the model. In contrast, a vanilla deep neural network only models the aleatoric uncertainty by a certain probability distribution. Thus, Bayesian neural networks are successfully applied in many tasks to model uncertainties and build a more reliable and robust system, including but not limited to computer vision tasks [17, 20, 30] and natural language processing tasks [39].

Neural network models without particular settings [2, 14] are sensitive and vulnerable to adversarial attacks in testing. Defenses against adversarial attacks are difficult. The Lipschitz constant serves as an evaluation metric of the adversarial robustness of a model by providing a worst-case bound [18, 37]. Many previous methods enhance the model robustness by constricting the Lipschitz constant [10, 23, 31]. These methods have made a significant improvement in both theoretical analysis and practical applications. However, they cannot be used in Bayesian neural networks directly because of the probabilistic representations of model parameters.

Bayesian neural networks, on the other hand, are useful for defending adversarial noises compared with vanilla neural networks. Because of the probabilistic representations of model parameters and predictions, Bayesian neural networks can be applied to detect adversarial samples from normal samples [5, 24, 34]. Moreover, Bayesian neural networks have been found to have adversarial robustness naturally. Y. Gal *et al.* [13] and Carbone *et al.* [9] reveal that any gradient-based adversarial attacks are invalid on Bayesian neural networks under some extremely idealized conditions, *e.g.*, idealized architecture [13], sufficient data and sampling times [9]. Nonetheless, these studies all have certain limitations. In many practical scenarios, predictions on adversarial samples are still necessary even though they have been detected. Additionally, the idealized conditions are almost impossible in practice. Therefore, there is still a vast space for further improvement of the robustness of Bayesian neural networks.

This paper presents a method, Spectral Expectation Bound Regularization (SEBR), to enhance the robustness of Bayesian neural networks. The model trained with SEBR has a smaller expectation of the spectral norm of the training parameter matrices. As a result, the improvement on the adversarial robustness of Bayesian neural networks is guaranteed based on theoretical derivation in this paper. Moreover, the impact of SEBR on the epistemic uncertainty of the output of Bayesian models is also studied theoretically and it further verifies the robustness of the proposed method. Ex-

*Tao Song is the corresponding author.

†This work was supported in part by National NSF of China (NO. 61872234, 61732010, 61525204) and Shanghai Key Laboratory of Scalable Computing and Systems.

periments are carried out to validate both the correctness of the theoretical findings and the improvement on the robustness of the models in a variety of actual scenarios.

In summary, the main contributions are listed as follows:

- This paper proposes Spectral Expectation Bound Regularization (SEBR), which applies the Lipschitz constraint in Bayesian neural networks efficiently. According to the theoretical analysis, it can improve the robustness of the Bayesian neural network models.
- It is proved that SEBR training reduces the uncertainty of the model effectively in theoretical analysis, which provides another explanation of the model robustness.
- Experiments on multiple Bayesian neural network structures verify the theory and the effectiveness of the proposed method. The codes are available in https://github.com/AISIGSJTU/SEBR.

2. Related Work

Robustness on Bayesian Neural Networks. Bayesian neural networks, where the model weights are treated as random variables, provide a probabilistic view of deep learning models [29]. Many previous methods investigated the robustness of Bayesian neural networks. It has been shown that Bayesian neural networks are effective in detecting adversarial samples [5, 24, 34], and it is observed that models tend to make wrong predictions on adversarial samples where the model outputs have high uncertainties [24]. X. Liu *et al*. applied adversarial training in Bayesian neural networks and gained an obvious robustness improvement [25]. From another point of view, Y. Gal *et al*. [13] revealed that idealized Bayesian neural networks can even avoid adversarial attacks. As the sufficient conditions in [13] are difficult to achieve in practice, Carbone *et al*. [9] further demonstrated that Bayesian neural networks are robust to gradient-based adversarial attacks in the large-data, over-parameterized limit. However, as the idealized conditions are almost impossible, Bayesian neural networks do not perform perfectly on defending against adversarial attacks in real tasks.

Lipschitz Constraint in Neural Networks. Methods about Lipschitz continuity are widely used to enhance the robustness and other targets in deep learning models. Yoshida and Miyato [40] proposed the spectral norm regularization to maintain the Lipschitz continuity by penalizing the sum of spectral norms of the parameter weight matrices. Further, Gouk *et al*. generalized the spectral norm regularization to non-l_2 norms and convolution layers [15]. On the other hand, Miyato *et al*. [28] proposed spectral normalization, where the spectral norms are normalized so that the Lipschitz constraint $Lip(f) = 1$ is satisfied. It is added into the discriminator in a generative adversarial network and the quality of generated samples gets improved. Jens Behrmann *et al*. [4] proved that the ResNet is invertible if its Lipschitz constant is restricted to $Lip(f) < 1$ on the residual blocks. Many other papers [3, 6, 12, 19, 32] apply the Lipschitz constraint and spectral norm in deep learning to enhance the generalizability and robustness. However, these existing methods are not suitable for Bayesian neural networks because of the probabilistic representations of parameters. Our method is the first to apply the Lipschitz constraint in Bayesian neural networks.

3. Background

3.1. Variational Inference in Bayesian Neural Networks

Suppose we have observations $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \dots\}$. A Bayesian neural network parameterized by $\mathbf{W}$ uses a variational distribution $Q(\mathbf{W})$ to approximate the true posterior probability $P(\mathbf{W}|\mathcal{D})$. For simplicity, we consider Bayesian neural networks with Gaussian priors, and parameters are represented as Gaussian distributions. The Bayesian neural network minimizes the Kullback–Leibler (KL) divergence

$$\begin{aligned} KL(Q(\mathbf{W})||P(\mathbf{W}|\mathcal{D}))) &= -\int Q(\mathbf{W}) \log \frac{P(\mathbf{W}|\mathcal{D})}{Q(\mathbf{W})} d\mathbf{W} \\ &= \log P(\mathcal{D}) - \int Q(\mathbf{W}) \log \frac{P(\mathbf{W}, \mathcal{D})}{Q(\mathbf{W})} d\mathbf{W}. \end{aligned} \tag{1}$$

Since $\log P(\mathcal{D})$ is a constant for given observations $\mathcal{D}$, minimizing the KL divergence is equivalent to minimizing

$$\begin{aligned} \mathcal{L} &= -\int Q(\mathbf{W}) \log \frac{P(\mathbf{W}, \mathcal{D})}{Q(\mathbf{W})} d\mathbf{W} \\ &= -\mathbb{E}_{\mathbf{W}} \ \log P(\mathcal{D}|\mathbf{W}) + KL(P(\mathbf{W})||Q(\mathbf{W}))). \end{aligned} \tag{2}$$

Note that $-\mathcal{L}$ is a lower bound of $logP(\mathbf{X})$, thus $\mathcal{L}$ is usually called the Evidence Lower Bound (ELBO) loss [7]. In practical Bayesian neural networks, the first term is usually estimated for each sample $(\mathbf{x}, \mathbf{y})$ in the observations by the following Monte Carlo sampling

$$-\mathbb{E}_{\mathbf{W}} \ \log P(\mathcal{D}|\mathbf{W}) \approx -\frac{1}{K} \sum_{k=1}^{K} \log p(\mathbf{y}|\mathbf{x}, \mathbf{W}_k), \mathbf{W}_k \sim Q(\mathbf{W}), \tag{3}$$

where $\log p(\mathbf{y}|\mathbf{x}, \mathbf{W}_k)$ can be calculated by the cross-entropy loss in classification tasks. The second term $KL(P(\mathbf{W})||Q(\mathbf{W})))$ is directly computed analytically with a presumed prior distribution.

3.2. Lipschitz Continuity for Neural Networks

Lipschitz continuity is a significant property of a function in mathematical analysis. A function $f : X \to Y$, is said to be Lipschitz continuous if there exists a real constant $\alpha \geq 0$ such that, for $\forall \mathbf{x}_1, \mathbf{x}_2 \in X$, we have

$$d_Y\left(f\left(\mathbf{x}_1\right), f\left(\mathbf{x}_2\right)\right) \leq \alpha \cdot d_X\left(\mathbf{x}_1, \mathbf{x}_2\right), \tag{4}$$

where d_X and d_Y denote the distance metrics on set X and Y, respectively. The smallest α that satisfies this condition is referred to as the Lipschitz Constant of function f. In the context of a deep neural network, the function f is a composite function composed by multiple functions:

$$f(\mathbf{x}) = \left(\phi_L \circ \phi_{L-1} \cdots \circ \phi_1\right)(\mathbf{x}), \tag{5}$$

where each ϕ_l is the mapping function of each layer $l = 1, \cdots, L$. For the convenience of expression, we let $Lip(\cdot)$ represent the Lipschitz constant of a function. According to the composition property of Lipschitz continuity, we have

$$Lip(f) \leq \prod_{l=1}^{L} Lip(\phi_l). \tag{6}$$

Hence, to constraint the Lipschitz constant of the whole function f, it is sufficient to bound the Lipschitz constants for the mapping functions ϕ_l of each layer $l = 1, \cdots, L$.

3.3. Uncertainty Estimation

Uncertainty estimation is one of the significant functions of Bayesian neural networks, which is also essential in the evaluation of the robustness of a deep learning model [16, 24, 27]. For a classification model with parameters $\mathbf{W}$, input $\mathbf{x}$ and output y in classes $C = \{c_1, c_2, \ldots, c_m\}$, following previous work [1, 11], we model the uncertainty in the prediction by its predictive entropy

$$H(y|\mathbf{x}, \mathbf{W}) = \sum_{i=1}^{m} p(y = c_i|\mathbf{x}, \mathbf{W}) \log p(y = c_i|\mathbf{x}, \mathbf{W}). \tag{7}$$

It contains both aleatoric uncertainty H_a and epistemic uncertainty H_e. The aleatoric uncertainty H_a is given by

$$H_a(y|\mathbf{x}, \mathbf{W}) = \mathbb{E}_{\mathbf{W}} H(y|\mathbf{x}, \mathbf{W}) \approx \frac{1}{K} \sum_{k=1}^{K} H(y|\mathbf{x}, \mathbf{W}_k), \tag{8}$$

which implies that it can be estimated by K Monte Carlo samplings. The epistemic component H_e is given by the difference between the total uncertainty H and the aleatoric uncertainty H_a, i.e.,

$$H_e(y|\mathbf{x}, \mathbf{W}) = H(y|\mathbf{x}, \mathbf{W}) - \mathbb{E}_{\mathbf{W}} H(y|\mathbf{x}, \mathbf{W}). \tag{9}$$

In a regression task, the output becomes a vector $\mathbf{y}$ instead of a class, which means the predictive entropy cannot be used to measure the uncertainty. Instead, the uncertainty can be measured by the following variance of the Gaussian mixture distribution over outputs [1]:

$$\begin{aligned} H(\mathbf{y}|\mathbf{x}, \mathbf{W}) = \sigma^2(\mathbf{y}|\mathbf{x}, \mathbf{W}) = \underbrace{\frac{1}{K} \sum_{k=1}^{K} \sigma^2(\mathbf{y}|\mathbf{x}, \mathbf{W}_k)}_{\text{aleatoric uncertainty}} \\ + \underbrace{\frac{1}{K} \sum_{k=1}^{K} \mu(\mathbf{y}|\mathbf{x}, \mathbf{W}_k)^2 - \left(\frac{1}{K} \sum_{k=1}^{K} \mu(\mathbf{y}|\mathbf{x}, \mathbf{W}_k)\right)^2}_{\text{epistemic uncertainty}}. \end{aligned} \tag{10}$$

The aleatoric uncertainty H_a is usually used to model the uncertainty caused by the noise in data, while the epistemic uncertainty H_e corresponds to the uncertainty in model parameters and model structures [1].

4. Spectral Expectation Bound Regularization

Here we consider an L-layer feed-forward Bayesian neural network to explain our method. A layer with the mapping function $f_{\mathbf{W}}(\mathbf{x}) = f(W\mathbf{x} + \mathbf{b})$ accepts $\mathbf{x} \in \mathbb{R}^m$ as the input. Here $f(\cdot)$ represents an activation function, *e.g.* `relu` and `sigmoid`, and $\mathbf{W}$ represents all trainable parameters of the function, including W and $\mathbf{b}$. The elements in parameter matrices W and $\mathbf{b}$ are all random variables in the Bayesian framework. Therefore, when we forward the function multiple times, the output vector $\mathbf{y}$ is sampled from a probabilistic distribution determined by input $\mathbf{x}$ and parameters $\mathbf{W}$.

In the following section, we will consider how to make the function robust to a given perturbation. The following theorem presents that the expectation of disturbance of the output in a layer is bounded by the expectation of the spectral norm of parameter matrix $\mathbb{E}\|W\|_2$, the length of the perturbation vector $\|\xi\|$, and the Lipschitz constant of the activation function $Lip(f)$.

Theorem 1. *Consider function $f_{\mathbf{W}}(\mathbf{x}) = f(W\mathbf{x} + \mathbf{b})$, where the activation function $f(\cdot)$ is Lipschitz continuous with Lipschitz constant $Lip(f)$. For any perturbation $\boldsymbol{\xi}$ with norm $\|\boldsymbol{\xi}\|$, we have*

$$\mathbb{E}_{\mathbf{W}} \left\| f_{\mathbf{W}}(\mathbf{x} + \boldsymbol{\xi}) - f_{\mathbf{W}}(\mathbf{x}) \right\| \leq Lip(f) \cdot \mathbb{E}\|W\|_2 \cdot \|\boldsymbol{\xi}\|, \tag{11}$$

where $\|W\|_2$ represents the spectral norm of matrix W, and it is defined as

$$\|W\|_2 = \max_{\boldsymbol{\xi} \in \mathbb{R}^n, \boldsymbol{\xi} \neq \mathbf{0}} \frac{\|W\boldsymbol{\xi}\|}{\|\boldsymbol{\xi}\|}. \tag{12}$$

The proof of this theorem is given in the Supplementary Material. Note that the Lipschitz constant of the activation function $f(\cdot)$ is fixed for a given Bayesian neural network structure. Besides, the Lipschitz constant of many popularly used activation function, *e.g.*, `relu` and `sigmoid`, is bounded by 1. Therefore, the expectation of the spectral norm of the weight matrix can influence the sensitivity of a Bayesian neural network model. The model will become more robust if $\mathbb{E}\|W\|_2$ of each layer get restricted.

Similar to the spectral norm regularization in vanilla neural networks [40], a simple method to restrict $\mathbb{E}\|W\|_2$ in a Bayesian neural network model is to add it to the loss as a regularization term, i.e.,

$$\underset{\mathbf{W}}{\text{minimize}} \quad \mathcal{L} + \frac{\lambda}{2} \sum_{l=1}^{L} (\mathbb{E}\|W^l\|_2)^2, \tag{13}$$

where the expectation is estimated by Monte Carlo sampling and the spectral norm is calculated by the Power Iteration method. However, this method has a very high computational complexity. We denote the times of Monte Carlo sampling as K and the iterations of Power Iteration as N. Then, the time complexity of such calculation is $O(KN)$. To accelerate the training process, we propose a method to fast estimate the upper bound of $\mathbb{E}\|W\|_2$ analytically, instead of directly estimating the expectation for W by Monte Carlo sampling and Power Iteration.

Theorem 2 gives an upper bound of the expectation of the spectral norm of a Gaussian random matrix.

Theorem 2. *Consider a Gaussian random matrix $W \in \mathbb{R}^{m\times n}$, where $W_{ij} \sim N(M_{ij}, A_{ij}^2)$ with $M, A \in \mathbb{R}^{m\times n}$. Suppose $G \in \mathbb{R}^{m\times n}$ is a zero-mean Gaussian random matrix with the same variance,* i.e.*, $G_{ij} \sim N(0, A_{ij}^2)$. We have*

$$\begin{aligned} &\mathbb{E}\|W\|_2 \\ \le &\|M\|_2 + c\left(\max_i \|A_{i,:}\| + \max_j \|A_{:,j}\| + \mathbb{E}\max_{i,j} |G_{ij}|\right), \end{aligned} \tag{14}$$

where c is a constant independent of W.

The proof for this theorem is shown in the Supplementary Material. With Theorem 2, we do not need to directly optimize $\mathbb{E}\|W\|_2$. We can optimize the upper bound of $\mathbb{E}\|W\|_2$. Specifically, the Power Iteration method is utilized to estimate the spectral norm of $\|M\|_2$. Monte Carlo sampling is adopted to estimate the last term $\mathbb{E}\max_{i,j}|G_{ij}|$. The remaining term, $\max_i \|A_{i,:}\| + \max_j \|A_{:,j}\|$ can be directly calculated on given A. Therefore, the time complexity is reduced from $O(KN)$ to $O(K+N)$ successfully. The constant c can be simply ignored because it is independent of the input and our target is to minimize the whole algebraic expression. Therefore, we can add the upper bound as a regularisation term into the loss function. Consequently, we consider the following empirical risk minimization problem:

Algorithm 1: Variational Inference with Spectral Expectation Bound Regularization

Compute the ELBO loss $\mathcal{L}$ with Equation (2).
$\mathcal{L}_S = 0$
for *l = 1 to L* **do**
  Define M^l, A^l following Theorem 2.
  $\mathcal{L}_l = 0$
  `// First:` $\|M\|_2$
  sample $u \sim N(\mathbf{0}, \mathbf{1})$
  for *Sufficient iterations N* **do**
    $v = (M^l)^T u / \|(M^l)^T u\|$
    $u = M^l v / \|M^l v\|$
  end
  $\mathcal{L}_l = \mathcal{L}_l + u^T M^l v$
  `// Second:` $\max_i \|A^l_{i,:}\| + \max_j \|A^l_{:,j}\|$
  $\mathcal{L}_l = \mathcal{L}_l + \max_i \|A^l_{i,:}\| + \max_j \|A^l_{:,j}\|$
  `// Third:` $\sum_{k=1}^{K} \max_{i,j} \left|\epsilon \cdot A^l_{ij}\right|^2$
  $sum = 0$
  for *Sufficient MC simulation times K* **do**
    sample $\epsilon_k \sim N(0,1)$
    $sum = sum + \max_{i,j} \left|\epsilon_k \cdot A^l_{ij}\right|$
  end
  $\mathcal{L}_l = \mathcal{L}_l + sum/t$
  $\mathcal{L}_S = \mathcal{L}_S + \frac{1}{2}\mathcal{L}_l^2$
end
$\mathcal{L} = \mathcal{L} + \lambda \cdot \mathcal{L}_S$
Update with the gradient on minimizing $\mathcal{L}$.

$$\underset{\mathbf{W}}{\text{minimize}}\, \mathcal{L} + \lambda \cdot \mathcal{L}_{\mathcal{S}}. \tag{15}$$

Here $\mathcal{L}$ is the ELBO loss as defined in Equation (2). The notation $\mathcal{L}_{\mathcal{S}}$ represents the SEBR loss:

$$\begin{aligned} \mathcal{L}_{\mathcal{S}} = \frac{1}{2}\sum_{l=1}^{L} (\|M^l\|_2 + \max_i \|A^l_{i,:}\| + \max_j \|A^l_{:,j}\| + \\ \sum_{k=1}^{K} \max_{i,j} \left|\epsilon_k \cdot A^l_{ij}\right|)^2, \epsilon_k \sim N(0,1). \end{aligned} \tag{16}$$

The parameter λ is a regularization factor, which controls the trade-off between the robustness and the expressive power of the model. We refer to this method as Spectral Expectation Bound Regularization (SEBR). The algorithm to apply SEBR together with variational inference in practice is provided in Algorithm 1.

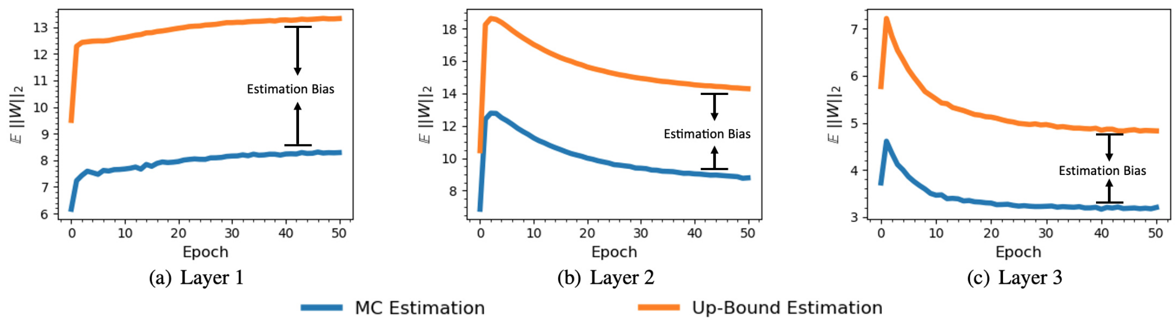


Figure 1. The variation trends of both Monte Carlo estimation and the estimated upper bound of $\mathbb{E}\|W\|_2$ in a 3-layer Bayesian neural network during training. *Best viewed in color.*

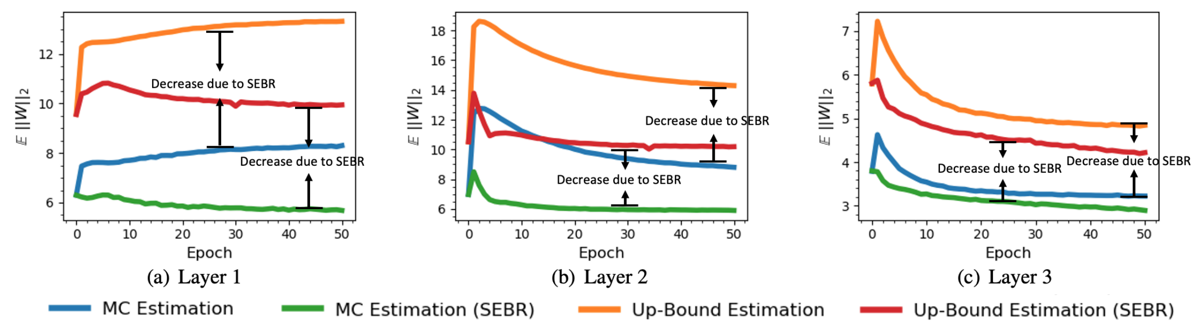


Figure 2. The comparison of the change of the Monte Carlo estimation and the estimated upper bound of $\mathbb{E}\|W\|_2$ between the original model and the model trained with the SEBR method. *Best viewed in color.*

Method	Avg. time per epoch
Reg. on $\mathbb{E}\|W\|_2$	1654.8 (s)
SEBR	410.5 (s)

Table 1. Time cost comparison between SEBR and the direct regularization on $\mathbb{E}\|W\|_2$.

5. Influence of SEBR on Uncertainty

In this section, we show that our SEBR method can reduce the epistemic uncertainty on the output of a Bayesian neural network model.

The following theorem shows the epistemic uncertainty of the output of a one-layer Bayesian neural network decreases after one step gradient descent with SEBR.

Theorem 3. *Consider a Bayesian neural network with only a linear layer $f_{\mathbf{W}}(\mathbf{x}) = W\mathbf{x} + \mathbf{b}$, where $\mathbf{x} \in \mathbb{R}^n$, $W \in \mathbb{R}^{m\times n}$. Denote the epistemic uncertainty (following the definition in Equation (10)) of the output after one step gradient descent without SEBR as H_e, and the epistemic uncertainty after one step gradient descent with SEBR as H'_e. With sufficient sampling times, we have*

$$H'_e \leq H_e. \tag{17}$$

Proof. With sufficient sampling times, the epistemic uncertainty the function $f_{\mathbf{W}}(\mathbf{x}) = W\mathbf{x} + b$ estimates is the variance of $\mu(\mathbf{y}|\mathbf{x}, \mathbf{W})$. Since $\mathbf{x}$ is a constant vector and all elements of W are independent Gaussian variables, we have

$$\begin{aligned} H_e = \sigma^2(\mu(\mathbf{y}|\mathbf{x}, \mathbf{W})) &= \sigma^2(\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} W_{ij}x_j) \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{m} x_j^2\sigma^2(W_{ij}) \\ &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{m} x_j^2 A_{ij}^2. \end{aligned} \tag{18}$$

Here A is the standard deviation matrix of W following the definition in Theorem 2. Compared with normal training, the one step gradient descent with SEBR additionally optimize the SEBR loss $\mathcal{L}_S$. Let A and A' be the standard deviation matrices corresponding to training with SEBR and without SEBR respectively. For each $p = 1, 2, \ldots, m$ and $q = 1, 2, \ldots, n$,

$$\begin{aligned} A'_{pq} = &A_{pq} - \alpha\sqrt{2\mathcal{L}_S}(\frac{\partial\|M\|_2}{\partial A_{pq}} + \frac{\partial \max_i \|A_{i,:}\|}{\partial A_{pq}} + \\ &\frac{\partial \max_j \|A_{:,j}\|}{\partial A_{pq}} + \frac{\partial \sum_{k=1}^{K} \max_{i,j} |\epsilon_k \cdot A_{ij}|}{\partial A_{pq}}), \end{aligned} \tag{19}$$

Model	Dataset	Attack	Noise	ℓ_∞ norm	Acc. w/o. SEBR (%)	Acc. w. SEBR (%)	Δ (%)
Bayesian MLP	MNIST	/	0	0	97.05 ± 0.38	96.83 ± 0.48	-0.22
		FGSM	small	0.04	83.83 ± 0.51	85.74 ± 0.64	**+ 1.91**
			medium	0.16	8.97 ± 0.28	43.69 ± 5.92	**+ 34.72**
			large	0.3	5.06 ± 0.21	24.54 ± 8.65	**+ 19.48**
		PGD	small	0.04	81.99 ± 1.05	83.67 ± 0.67	**+ 1.68**
			medium	0.16	4.20 ± 0.84	9.54 ± 2.82	**+ 5.34**
			large	0.22	1.55 ± 0.35	3.18 ± 1.52	**+ 1.63**
Bayesian CNN	MNIST	/	0	0	98.88 ± 0.27	98.70 ± 0.04	-0.18
		FGSM	small	0.04	85.64 ± 2.52	86.14 ± 2.76	**+ 0.50**
			medium	0.08	55.98 ± 4.40	60.27 ± 8.65	**+ 4.29**
			large	0.14	18.16 ± 0.57	22.55 ± 11.23	**+ 4.39**
		PGD	small	0.04	82.91 ± 2.63	85.10 ± 2.96	**+ 2.19**
			medium	0.08	36.53 ± 5.85	49.20 ± 10.75	**+ 12.67**
			large	0.14	9.88 ± 2.02	12.33 ± 5.31	**+ 2.45**
Bayesian MLP	Fashion MNIST	/	0	0	84.38 ± 0.37	78.75 ± 0.83	-5.63
		FGSM	small	0.04	60.96 ± 0.24	62.06 ± 1.15	**+ 1.10**
			medium	0.1	24.29 ± 1.16	31.65 ± 1.25	**+ 7.36**
			large	0.2	1.99 ± 0.57	4.59 ± 0.75	**+ 2.60**
		PGD	small	0.04	59.86 ± 0.34	61.80 ± 1.13	**+ 1.94**
			medium	0.1	19.18 ± 1.01	29.67 ± 1.22	**+ 10.49**
			large	0.2	0.44 ± 0.14	2.71 ± 0.60	**+ 2.27**
Bayesian CNN	Fashion MNIST	/	0	0	87.45 ± 0.57	84.83 ± 0.33	-2.62
		FGSM	small	0.04	40.82 ± 1.86	46.03 ± 4.22	**+ 5.21**
			medium	0.08	15.89 ± 0.97	18.96 ± 5.00	**+ 3.07**
			large	0.1	10.24 ± 0.31	11.97 ± 3.95	**+ 1.73**
		PGD	small	0.04	32.81 ± 1.70	39.92 ± 3.25	**+ 7.11**
			medium	0.06	15.03 ± 2.03	20.87 ± 4.00	**+ 5.84**
			large	0.08	5.62 ± 0.73	9.27 ± 1.62	**+ 3.65**

Table 2. Comparison on the Robustness of Models without SEBR and with SEBR. The mean value and maximum deviation of three runs are reported.

where $\alpha > 0$ is the learning rate.

It is obvious that the first term $\frac{\partial \|M\|_2}{\partial A_{ij}} = 0$ since the mean matrix M is unrelated to A_{ij}. The second term $\frac{\partial \max_i \|A_{i,:}\|}{\partial A_{pq}} > 0$ when $p = \operatorname{argmax}_i \|A_{i,:}\|$; otherwise, $\frac{\partial \max_i \|A_{i,:}\|}{\partial A_{pq}} = 0$. Similarly, the third term $\frac{\partial \max_j \|A_{:,j}\|}{\partial A_{pq}}$ is non-negative. The last term satisfies

$$\frac{\partial \sum_{k=1}^{K} \max_{i,j} |\epsilon_k \cdot A_{ij}|}{\partial A_{pq}} = \sum_{k=1}^{K} \frac{\partial \max_{i,j} |\epsilon_k| \cdot A_{ij}}{\partial A_{pq}} \geq 0. \tag{20}$$

Therefore, we have: $\forall p, q, A'_{pq} \leq A_{pq}$. By substituting it into Equation (18), we can get the result of the theorem. $\square$

Theorem 3 states that our SEBR naturally reduces the epistemic uncertainty on the output of the Bayesian neural network effectively. On the other hand, it is obvious that the aleatoric uncertainty can also be reduced because of the optimization on the spectral norm of the mean matrix $\|M\|_2$. The reduction on both the epistemic uncertainty and the aleatoric uncertainty is observed and verified in the following experiments, which enables our model to be robust and confident on the predictions.

6. Experiments

In this section, we empirically verify our theoretical findings and investigate the effectiveness of the proposed SEBR method. We train a variety of Bayesian neural networks, including Bayesian MLPs and Bayesian CNNs, on MNIST dataset [22], Fashion-MNIST dataset [38], CIFAR-10 dataset, and CIFAR-100 dataset [21]. In Section 6.1,

Model	Dataset	Attack	Noise	ℓ_∞ norm	Acc. w/o. SEBR (%)	Acc. w. SEBR (%)	Δ (%)
Bayesian MLP + Adv. Training	MNIST	/	0	0	97.22 ± 0.27	96.94 ± 0.39	-0.28
		FGSM	small	0.04	92.87 ± 0.27	92.08 ± 0.12	-0.79
			medium	0.16	54.56 ± 1.71	57.63 ± 1.08	**+ 3.07**
			large	0.3	9.94 ± 0.13	33.09 ± 8.23	**+ 23.15**
		PGD	small	0.04	92.57 ± 0.40	91.87 ± 0.26	-0.70
			medium	0.16	40.05 ± 5.32	40.66 ± 4.18	**+ 0.61**
			large	0.22	11.15 ± 5.70	16.47 ± 3.57	**+ 5.32**
Bayesian CNN + Adv.Training	MNIST	/	0	0	98.89 ± 0.19	98.77 ± 0.08	-0.12
		FGSM	small	0.04	96.23 ± 0.40	95.96 ± 0.23	-0.27
			medium	0.2	62.34 ± 4.70	63.20 ± 4.10	**+ 0.86**
			large	0.44	11.36 ± 2.17	14.18 ± 0.82	**+ 2.82**
		PGD	small	0.04	95.98 ± 0.40	95.79 ± 0.24	-0.19
			medium	0.2	26.17 ± 4.39	30.06 ± 3.92	**+ 3.89**
			large	0.44	6.85 ± 1.67	8.78 ± 1.07	**+ 1.93**

Table 3. Comparison on the Robustness of Adversarial trained Models without SEBR and with SEBR. The mean value and maximum deviation of three runs are reported.

we experimentally present the variation of the models after adding the SEBR and validate the theoretical motivation. In Section 6.2, we discuss the improvement on the robustness of defending adversarial attacks on Bayesian neural networks with SEBR. In Section 6.3, we analyze the uncertainty variation caused by SEBR and verify Theorem 3 experimentally .

6.1. Variation of Models after adding SEBR

We present the experiment results to verify that the upper bound of $\mathbb{E}\|W\|_2$ is a suitable estimation and it can reflect the change trend of the real value of $\mathbb{E}\|W\|_2$. The experiments are done on a Bayesian MLP with three layers on the MNIST dataset [22]. The values of $\mathbb{E}\|W\|_2$ and our estimated upper bound are recorded for each layer during a 50-epoch training. Figure 1 shows the results. Even though there is an obvious gap between the upper bound and the un-biased estimated value by the Monte-Carlo estimation, the difference between them keeps stable and their variation trends are synchronous. This validates the rationality of utilizing the upper bound in our method.

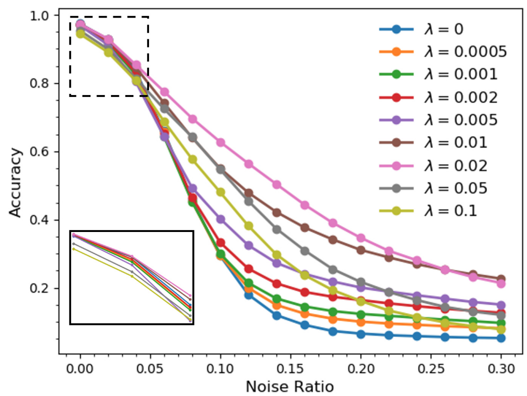
We further investigate how our SEBR method influences $\mathbb{E}\|W\|_2$. The parameter λ is set to be 0.01 in the Bayesian neural network. The experiment results are shown in Figure 2. The added constraint on the upper bound from SEBR not only reduces the upper bound itself but also reduces the un-biased evaluated value estimated by the Monte Carlo estimation, which validates the effectiveness of SEBR.

To verify that SEBR indeed reduces the time cost in practice, we compare the time cost for SEBR and the direct regularization method on the expectation of the spectral norms shown in Equation (13). The simulation times of Monte Carlo sampling and iteration times of Power Iteration are set as 10. According to the experimental results shown in Table 1, the direct optimization on $\mathbb{E}\|W\|_2$ makes the training very slow because it needs sufficient times for both Monte Carlo sampling and Power Iteration and the calculation of the expectation is necessary in every forward propagation. The training with SEBR significantly reduces the amount of time cost for training compared with the direct optimization method. Hence, it enhances the feasibility of the method in practice.

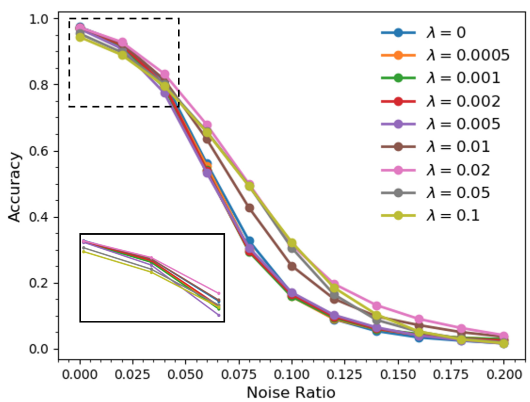
6.2. Improvements on Adversarial Robustness

The Fast Gradient Sign Method (FGSM) [14] is one of the most commonly used attack methods. The Projected Gradient Descent method (PGD) [26] is a more sophisticated and powerful adversarial attack method. To evaluate the impact of different settings of λ for the SEBR method shown in Equation (15), we measure the change in robustness with varying λ on defending the FGSM and the PGD attacks. The results are presented in Figure 3, where the accuracy is used as the evaluation metric. In the absence of adversarial noise, our SEBR causes a slight decrease on performance. It is normal because of the trade-off between clean accuracy and adversarial accuracy [35, 41]. With the increase of λ from 0 (i.e., model without SEBR) to 0.02, the model becomes more robust on defending noises, even though there is a subtle performance decrease on data without adversarial noise. On the other hand, when we continue increasing λ, the model performs worse because of the poorer fitting ability. Therefore, using a suitable λ is important to achieve fairly competent performance.

Table 2 provides the comparisons of robustness of the

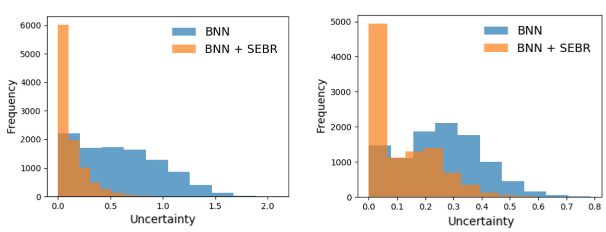


(a) FGSM attack



(b) PGD attack

Figure 3. Change in robustness on defending FGSM and PGD attacks with different λ in SEBR. *Best viewed in color.*



(a) Aleatoric Uncertainty (b) Epistemic Uncertainty

Figure 4. Uncertainties measured by Bayesian neural networks on data with adversarial noises. Models trained with SEBR have lower uncertainty on the predictions. *Best viewed in color.*

models without SEBR and with SEBR, where both the Bayesian MLP models and the Bayesian CNN models are tested on the MNIST [22] dataset and the Fashion MNIST dataset [38]. We continue using the 3-layer neural network in the MLP model, and we use LeNet as the CNN architecture here. The hyper-parameter settings and the implementation details are reported in the Supplementary Material. We present the accuracy of the models on defending adversarial attacks of different norms. Since different adversarial attacks are not of the same attack power and the robustness of different baseline models are also different, different absolute noise ℓ_∞ norms are adopted for different models to reflect the robustness of the model in various situations as fully as possible. The models with SEBR are more robust on defending all of small, medium, large noises compared with the original Bayesian neural network models. To verify that SEBR is also effective on more modern architectures and larger datasets, we show more experiment results about SEBR of Bayesian CNN with VGG [33] architecture on CIFAR10 and CIFAR100 datasets in the Supplementary Material. SEBR keeps effective on the larger diverse datasets and more complex network architecture.

We also implement the model adversarially trained with FGSM as a higher baseline. It utilizes the information from model gradients and input data, and hence it is among the most effective defense techniques [14, 25, 36]. The results are shown in Table 3. It makes models robust on defensing data with small noise. Nonetheless, our SEBR method further improves the model robustness obviously on defensing larger adversarial noise, which further verifies the universality and effectiveness of SEBR.

6.3. Uncertainty Variation

To further verify the robustness of the models trained with SEBR, we measure the aleatoric uncertainty and the epistemic uncertainty on Bayesian neural networks trained with SEBR and without SEBR. Figure 4 presents the measured uncertainties on data with small FGSM adversarial noises ($\ell_\infty = 0.1$). More experimental results on clean data and other noises can be found in the Supplementary Material. All of the experiments show that the models trained with SEBR has lower uncertainties, including both the aleatoric uncertainty and the epistemic uncertainty. Therefore, our SEBR makes the models more confident on the predictions and improves the robustness.

7. Conclusion

In this paper, we propose the SEBR method that restricts the expectation of the Lipschitz constant on Bayesian neural networks. The theoretical analysis demonstrates that SEBR improves the robustness of defending against adversarial noises. The relationship between SEBR training and the output uncertainty variation is also discussed. It is proved that SEBR reduces the uncertainty on the model outputs. We verify our proposals by experiments on both the Bayesian MLP model and the Bayesian CNN model in defending FGSM and PGD attacks. Further experiments validate that models trained with SEBR have lower uncertainties, which verifies the robustness from another side.

References

[1] Javier Antorán. Understanding uncertainty in bayesian neural networks. Master's thesis, University of Cambridge, 2019. 3

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018. 1

[3] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *NeurIPS*, 2017. 2

[4] Jens Behrmann, Will Grathwohl, Ricky T. Q. Chen, David Duvenaud, and Joern-Henrik Jacobsen. Invertible residual networks. In *ICML*, 2019. 2

[5] Artur Bekasov and Iain Murray. Bayesian adversarial spheres: Bayesian inference and adversarial examples in a noiseless setting. *arXiv preprint arXiv:1811.12335*, 2018. 1, 2

[6] Alberto Bietti, Grégoire Mialon, and Julien Mairal. On regularization and robustness of deep neural networks. 2018. 2

[7] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. 2

[8] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *ICML*, 2015. 1

[9] Ginevra Carbone, Matthew Wicker, Luca Laurenti, Andrea Patane, Luca Bortolussi, and Guido Sanguinetti. Robustness of bayesian neural networks to gradient-based attacks. *arXiv preprint arXiv:2002.04359*, 2020. 1, 2

[10] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *ICML*, 2017. 1

[11] Stefan Depeweg. *Modeling Epistemic and Aleatoric Uncertainty with Bayesian Neural Networks and Latent Variables*. PhD thesis, Technische Universität München, 2019. 3

[12] Farzan Farnia, Jesse Zhang, and David Tse. Generalizable adversarial training via spectral normalization. In *ICLR*, 2018. 2

[13] Yarin Gal and Lewis Smith. Sufficient conditions for idealised models to have no adversarial examples: a theoretical and empirical study with bayesian neural networks. *arXiv preprint arXiv:1806.00667*, 2018. 1, 2

[14] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 1, 7, 8

[15] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael Cree. Regularisation of neural networks by enforcing lipschitz continuity. *arXiv preprint arXiv:1804.04368*, 2018. 2

[16] Kathrin Grosse, David Pfaff, Michael Thomas Smith, and Michael Backes. The limitations of model uncertainty in adversarial settings. *arXiv preprint arXiv:1812.02606*, 2018. 3

[17] Fredrik K Gustafsson, Martin Danelljan, and Thomas B Schön. Evaluating scalable bayesian deep learning methods for robust computer vision. In *CVPR Workshops*, 2020. 1

[18] Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *NeurIPS*, 2017. 1

[19] Kui Jia, Dacheng Tao, Shenghua Gao, and Xiangmin Xu. Improving training of deep neural networks via singular value bounding. In *CVPR*, 2017. 2

[20] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, 2017. 1

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[22] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 6, 7, 8

[23] Qiyang Li, Saminul Haque, Cem Anil, James Lucas, Roger B Grosse, and Joern-Henrik Jacobsen. Preventing gradient attenuation in lipschitz constrained convolutional networks. In *NeurIPS*, 2019. 1

[24] Yingzhen Li and Yarin Gal. Dropout inference in bayesian neural networks with alpha-divergences. In *ICML*, 2017. 1, 2, 3

[25] Xuanqing Liu, Yao Li, Chongruo Wu, and Cho-Jui Hsieh. Adv-BNN: Improved adversarial defense through robust bayesian neural network. In *ICLR*, 2019. 2, 8

[26] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 7

[27] Lu Mi, Hao Wang, Yonglong Tian, and Nir Shavit. Training-free uncertainty estimation for neural networks. *arXiv preprint arXiv:1910.04858*, 2019. 3

[28] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018. 2

[29] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012. 1, 2

[30] Buu Truong Phan. Bayesian deep learning and uncertainty in computer vision. Master's thesis, University of Waterloo, 2019. 1

[31] Haifeng Qian and Mark N. Wegman. L2-nonexpansive neural networks. In *ICLR*, 2019. 1

[32] Hanie Sedghi, Vineet Gupta, and Philip M Long. The singular values of convolutional layers. In *ICLR*, 2018. 2

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 8

[34] Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. In Amir Globerson and Ricardo Silva, editors, *UAI*, 2018. 1, 2

[35] Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy? - A comprehensive study on the robustness of 18 deep image classification models. In *ECCV*, 2018. 7

[36] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *ICML*, 2019. 8

[37] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. In *ICLR*, 2018. 1

[38] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 6, 8

[39] Yijun Xiao and William Yang Wang. Quantifying uncertainties in natural language processing tasks. In *AAAI*, 2019. 1

[40] Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017. 2, 4

[41] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019. 7