Evaluating the Robustness of Bayesian Neural Networks Against Different Types of Attacks

Yutian Pang, Sheng Cheng, Jueming Hu, Yongming Liu*
Arizona State University, Tempe, AZ
{yutian.pang, scheng53, jueming.hu, yongming.liu}@asu.edu

Abstract

To evaluate the robustness gain of Bayesian neural networks on image classification tasks, we perform input perturbations, and adversarial attacks to the state-of-the-art Bayesian neural networks, with a benchmark CNN model as reference. The attacks are selected to simulate signal interference and cyberattacks towards CNN-based machine learning systems. The result shows that a Bayesian neural network achieves significantly higher robustness against adversarial attacks generated against a deterministic neural network model, without adversarial training. The Bayesian posterior can act as the safety precursor of ongoing malicious activities. Furthermore, we show that the stochastic classifier after the deterministic CNN extractor has sufficient robustness enhancement rather than a stochastic feature extractor before the stochastic classifier. This advises on utilizing stochastic layers in building decision-making pipelines within a safety-critical domain.

1. Introduction

Deep Neural Networks (DNNs) have been integrated into various safety-critical engineering applications (*e.g.* Unmanned Aerial Vehicle (UAV), Autonomous System (AS), Surveillance System (SS)). The prediction made by these algorithms needs to be reliable with sufficient robustness. A failed DNN can lead to potentially fatal collisions, especially for the solely camera-based autonomous systems. Several such real-world accidents have happened including ones that resulted in a fatality [21], where the image of the white-colored truck was classified as the cloud. On the other hand, it's widely known that the predicted labels of neural networks are vulnerable to adversarial samples [1, 12, 13]. The research on adversarial machine learning has focused on developing an enormous number of adversarial attack and defense methods [7, 8, 25]. Most of the adversarial attack/defense methods are developed towards the classical convolutional neural network (CNN) on image classification tasks. Typically, the defense requires adversarial training with adversarial samples that are used to perform adversarial attacks. In this way, the model can act more robust against these types of attacks. However, it's worth pointing out that the development of new attack methods never ends.

Bayesian NNs, with distributions over their weights, are gaining attention for their uncertainty quantification ability and high robustness from Bayesian regularization, while retaining the advantages of deterministic NNs [5]. The robustness gain of BNNs is not rigorous studied in the literature yet lacking quantified comparative experiments on a real-world dataset. In particular, we compare various types of Bayesian inference methods to NNs including Bayes By Backprop (BBB) [4] with (local) reparameterization [15, 19], Variational Inference (VI) [16], and Flipout approximation [23]. BNNs are evaluated against several types of input perturbations, white-box adversarial attacks, and black-box adversarial attacks without adversarial training. These attacks simulate the possible attacks toward a deployed NN system in the real world, intentionally or unintentionally. The adversarial samples are generated with the L_p threat models. In this paper, we adopt 6 input perturbation methods, 5 white-box adversarial attacks, 3 black-box attacks towards two open-source datasets (German Traffic Sign Recognition (GTSRB) [14] & Planes in Satellite Imagery (PlanesNet) [9]), both of which were in the safety-critical domains (AS & SS).

We have several exciting findings by analyzing the experiment results quantitatively. Firstly, we notice that BNN has limited robustness benefits against various input perturbations since the classical CNN has also demonstrated denoising capabilities Secondly, the Bayesian neural network shows significant robustness in the experiments in terms of classification accuracy, especially against constrained adversarial attacks [10]. Thirdly, we realize that both models will fail when dealing with unconstrained adversarial attacks. In this case, the attacks are obviously distinguishable by human visions. Furthermore, the stochasticity on the classifier can achieve comparative performance by putting weight uncertainties on both the convolutional extractor and the classifier, with comparative computation time consumption. These findings give advice on building robust stochastic image-based classifiers in real-world machine learning system applications. More discussions are presented in Sec 6.

2. Background

2.1. Bayesian neural networks

The formulation of Bayesian NN relies on Bayesian probabilistic modeling with i.i.d. distributions over network parameters. The Bayesian approach gives a space of parameters ω as a distribution $p(\omega)$ called the *prior*, and a likelihood distribution $p(Y|X,\omega)$, which is a probabilistic model of the model outputs given X and ω. The posterior is proportional to the likelihood and the prior and the prediction is simply $\mathbb{E}_{p(\omega|X,Y)}[p(Y^*|X^*,\omega)]$. X^* is the test input and Y^* is the prediction. However, the inference of the posterior and the prediction are both intractable [4].

Variational Inference (VI) [16], as an approximate probabilistic inference method, is used to resolve this. The objective is to minimize the distance between the approximate variational distribution $q_\theta(\omega)$ for the posterior $p(\omega|X,Y)$. The objective is further approximated as the negative Evidence Lower BOund (ELBO). In practice, ELBO is approximated by $\sum_{i=1}^{n}[q_{\omega^{(i)}}(\omega) - \log p(\omega^{(i)}) - \log p(Y|X,\omega^{(i)})]$, where $\omega^{(i)}$ is the i^{th} Monte Carlo sample from $q_\theta(\omega)$. ω is reparameterized into (μ,σ) for backpropagation [15, 19]. Sampling the network parameters stochastically during training is referred as *weight perturbations*. The recent advancements of weight perturbation method, Flipout, decorrelate the gradients within each batch of the data [23], while boosting the inference process of BNN.

2.2. Adversarial Attacks

Adversarial attacks are defined based upon the concept of threat model [6]. Denote $f(\cdot)$ as a classification model, with original input x and adversarial samples x^{adv}, and y denotes the ground-truth label. The adversarial attack is to attack the model f by adding small perturbation on the original inputs. This perturbation measured by the L_p norm is limited by the perturbation budget ε, that is $\|x - x^{adv}\|_p < \varepsilon$. Particularly, we use L_∞ in this paper which implies that the perturbation to each pixel in x can't be larger than ε. The generation of adversarial samples is formulated into two optimization problems depend on if ε presented. Eq. (1) is to generate an untargeted adversarial example by maximizing the cross-entropy loss function $\mathcal{L}$. The second strategy is to find the minimum perturbation as Eq. (2). x' is one proposed adversarial sample by the generation algorithm.

$$x^{adv} \leftarrow \underset{\|x-x'\|_p<\varepsilon}{\operatorname{argmax}} \mathcal{L}(x', y) \tag{1}$$

$$x^{adv} \leftarrow \underset{x'}{\operatorname{argmin}} \|x - x'\|_p \tag{2}$$

The concept of white-box attacks and black-box attacks build upon the level of adversary's knowledge. White-box attacks typically acquire the full knowledge of the model, including model architectures, parameters, loss formula. White-box attack methods generate perturbations based on NN gradients given the detailed knowledge of the model. Under Eq. (1), the Fast Gradient Sign Method (FGSM) [13] generates x^{adv} by an one-step update. Basic Iterative Method (BIM) [17] is an iterative version of FGSM with a multi-step update. Projected gradient descent method (PGD) [18] has a similar first-order setup but with random initials. DeepF [20] and Carlini & Wagner's method (C&W) [8] have been used to solve Eq. (2).

Black-box attacks have limit/partial knowledge of the target model. Depending on the portion of the knowledge to the model, it can be further categorized into transfer-based, score-based, and decision-based black-box attacks. Transfer-based attack uses *distillation as a defense* strategy by training a substitute model with the knowledge of the training data. Momentum Iterative Method (MIM) [11] gives guidance on update direction as an extension of BIM. Score-based black-box attacks only acquire the output probabilities. It estimates the gradients by gradient-free methods with limited queries. An example of a score-based attack is SPSA [22]. Decision-based black-box attacks solely acquire the hard-label predictions. The Square Attack [2] is an example based on a random search on the decision boundary.

2.3. Input perturbations

Input perturbations to x, as a similar concept to data augmentation, are also examined in this paper as they also exist in real-world cases. We adopt 6 types of input perturbation methods to simulate various user cases. Firstly, we use the recent advancement Random Erasing (RE) [26]. RE randomly masks a rectangular region with black color in x with several masking parameters to determine the size of the region. Furthermore, we generate RE with random color as masking of the inputs. A typical case is stickers on the stop sign and fails a self-driving car. We also adopt the salt-and-pepper noise [3], and speckle noise to simulate signal interference. This includes the electromagnetic interference (EMI) in unshielded twisted pairs (UTP) in Ethernet or adjacent-channel interference in frequency modulation (FM) systems. We use Gaussian/Possion blur on x to simulate the system with low data transmission speed, and/or bandwidth issues.

3. Experiments

3.1. Evaluated Datasets

We use the GTSRB [14] and PlanesNet [9] in this paper. The GTSRB images are classified into 43 classes where

Table 1: Performance Against Input Perturbations. Report Test Accuracy in %.

Dataset I: GTSRB(German Traffic Sign Recognition Benchmark)								
Methods		Clean	Gaussian	S&P	Poisson	RE	RE Colorful	Speckle
CNN	Baseline	96.28±0.69	96.19±0.72	76.91±4.37	96.29±0.69	88.76±0.79	72.50±4.52	15.18±2.33
F-BNN	Flipout	97.17±0.22	97.19±0.17	84.06±1.00	97.18±0.17	90.45±0.33	82.12±0.55	34.12±3.27
	BBB	97.25±0.16	97.27±0.16	80.81±1.61	97.26±0.21	89.81±0.92	79.69±1.99	26.89±2.81
BNN	Flipout	96.85±0.51	96.87±0.56	75.51±4.49	96.90±0.55	89.93±0.85	72.96±1.60	10.46±4.52
	BBB	96.93±0.32	96.94±0.35	79.42±1.93	96.93±0.34	88.45±1.32	73.65±2.71	17.48±2.97
	LRT	96.85±0.16	96.83±0.20	75.60±1.13	96.83±0.23	88.64±1.22	71.88±4.22	10.88±2.50
	VI	95.37±0.41	95.30±0.44	73.33±3.65	95.35±0.40	85.83±1.84	68.91±0.99	10.05±2.42

the training set contains 39,209 images and 12,630 in test set. The PlansNet has two class labels indicate *plane* or *no-plane* given a input image. We use 10% of the data for testing. The attack methods discussed in Sec. 2.2 and Sec. 2.3 are used in the experiment. The visualization of test images are shown in Appendix.

3.2. Evaluation Procedures

Firstly, we train the baseline CNN model, the F-BNN model, and the BNN model. We build all the models following the VGG-16 architecture, but with stochastic layers in the Bayesian formulation. F-BNN here refers to fully-Bayesian neural network where both feature extractor and classifier are stochastic, while only stochastic classifier presented in BNN model. Then, we generate different types of adversarial samples of the test data w.r.t. the baseline CNN model and test each of the trained model to get the classification accuracy against adversarial attacks. Repeating this procedure for 5 times and averaging the results to get the mean prediction accuracy and variance. The evaluation procedure for input perturbations are similar to this procedure. We report the quantitative results in Table. 1, Table. 2.

4. Results

Table. 1 lists the quantitative results of each model setups against input perturbations. Results with a clean test input show the CNN baseline is well-trained. Variational Bayes performs slightly worse. We analysis input perturbations by groups, a) The Gaussian/Possion noisy blur to the inputs won't affect the model performance. The reason is neural network has been proved to have denoising capabilities, especially for parameterized distributional noise. b) The F-BNN with the RE/Colorful RE input perturbations has better performance among all cases. c) The S&P/Speckle signal interference cases has the best attack performance.

In Table. 2, we report the results of different models against adversarial attacks on two datasets. For GTSRB dataset, We generate the untargeted adversarial samples using the L_∞ threat model with $\varepsilon = 0.10$ and $\varepsilon = 0.15$ on both of these two datasets. We perform attacks as previously discussed in Sec. 2.2. We also evaluate the L_∞ norm between adversarial samples and the original inputs for each run. We report the mean L_∞ distance between x and x^{adv} for different attack methods in the table (*e.g*. PGD/0.068).

We also perform training time analysis for BNN and F-BNN with various settings in Figure. 1. The model architecture, layer setups, and training procedure for different methods are kept identical to address a fair comparison. The minimum median time used for training BNN is the Bayes By Backprop method, with smaller interquartile range. This also holds true for F-BNN training where only BBB and Flipout are used. We observe that BNN requires less computer training time. The same pattern is discovered for both datasets.

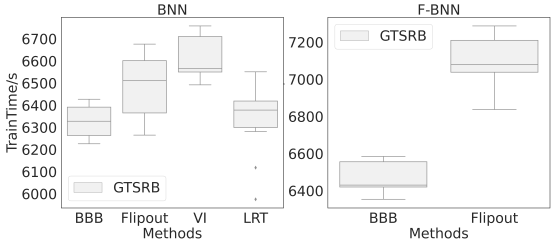


Figure 1: Training time comparison on GTSRB with stochastic BNN. *Left: BNN with only classifier as stochastic. Right: F-BNN with stochasticity on each neural network layer.*

5. Discussions

For adversarial attacks, the PGD achieves best attack performance towards our CNN baseline models with reasonable perturbations. However, BNN and F-BNN also exhibit significant robustness gain under PGD attacks. The classification accuracy rises to 80% in F-BNN for GTSRB data. The white-box adversarial samples generated with Eq. 2 show bipolarity. This indicates the generation algorithm needs specific parameter tuning. This is beyond the scope of this work. Bayesian NN also proves to be robust under these cases. The peer comparison between stochastic setting shows that BNN and F-BNN are analogous, in the

Table 2: Performance Against Adversarial Attacks with different ε and dataset. Report Test Accuracy in %.

Dataset I: GTSRB(German Traffic Sign Recognition Benchmark)									
$L_\infty(\epsilon = .10)$		White-box Attacks					Black-box Attacks		
Methods/Distance		PGD/0.056	FGSM/0.071	BIM/0.049	C&W/0.001	DeepF/0.283	SPSA/0.290	MIM/0.092	Square/0.088
CNN	Baseline	2.43 ± 0.50	28.79±1.69	28.35±1.65	83.80±6.19	3.97 ± 0.64	3.27 ± 0.12	2.26 ± 0.41	15.26±3.05
F-BNN	Flipout	77.86±2.57	58.86±1.68	68.08±2.53	95.59±0.27	20.58±1.73	3.75 ± 0.04	61.59±3.00	85.65±4.36
	BBB	78.78±3.27	59.96±2.01	69.32±2.88	95.76±0.23	20.57±1.90	3.78 ± 0.04	62.86±3.47	80.19±3.37
BNN	Flipout	75.55±2.93	55.71±2.39	65.77±2.42	94.99±0.61	18.57±1.88	3.78 ± 0.06	57.30±2.84	70.54±5.01
	BBB	76.47±1.68	57.13±1.61	66.69±1.58	95.44±0.53	19.47±2.65	3.76 ± 0.05	58.49±2.20	71.65±7.76
	LRT	76.78±2.04	57.10±1.86	67.25±1.89	95.55±0.35	19.33±1.42	3.77 ± 0.05	58.90±2.80	76.53±5.79
	VI	73.14±1.89	53.37±1.16	63.51±1.13	93.85±0.59	17.79±2.28	3.75 ± 0.08	53.97±2.18	67.86±8.42
Dataset I: GTSRB(German Traffic Sign Recognition Benchmark)									
$L_\infty(\epsilon = .15)$		White-box Attacks					Black-box Attacks		
Methods/Distance		PGD/0.061	FGSM/0.104	BIM/0.060	C&W/0.002	DeepF/0.285	SPSA/0.293	MIM/0.133	Square/0.132
CNN	Baseline	2.41 ± 0.51	28.29±1.77	28.32±1.66	79.77±2.49	3.99 ± 0.64	3.20 ± 0.09	2.21 ± 0.39	5.24 ± 1.66
F-BNN	Flipout	73.28±2.97	42.79±3.68	56.20±4.42	94.94±0.14	20.96±0.84	3.80 ± 0.02	42.86±4.41	73.10±5.62
	BBB	70.00±2.37	41.91±1.84	53.87±2.13	94.97±0.14	19.92±1.78	3.87 ± 0.04	39.76±1.79	74.51±5.97
BNN	Flipout	69.96±3.26	40.88±0.82	54.65±1.25	94.65±0.49	20.01±0.27	3.75 ± 0.04	37.58±2.41	62.82±4.08
	BBB	70.21±1.75	40.92±1.41	54.23±1.53	94.72±0.38	19.89±1.13	3.83 ± 0.04	38.54±1.89	58.18±9.47
	LRT	69.39±2.20	40.48±1.34	53.54±1.44	94.76±0.09	19.11±2.36	3.78 ± 0.07	36.34±1.81	60.43±11.1
	VI	67.35±3.84	39.89±1.89	53.28±2.45	93.23±0.42	18.07±1.75	3.79 ± 0.05	35.12±2.43	53.86±8.77
Dataset II: PlanesNet(Detect Aircraft in Planet Satellite Image Chips)									
$L_\infty(\epsilon = .10)$		White-box Attacks					Black-box Attacks		
Methods/Distance		PGD/0.068	FGSM/0.086	BIM/0.061	C&W/0.005	DeepF/0.349	SPSA/0.212	MIM/0.094	Square/0.080
CNN	Baseline	1.81 ± 0.37	45.44±2.40	13.50±2.28	68.21±2.13	43.77±4.83	49.65±0.91	1.83 ± 0.36	15.46±1.70
F-BNN	Flipout	24.51±3.70	57.69±2.37	36.78±4.35	91.13±0.41	54.90±2.84	60.88±0.79	23.14±2.71	75.36±6.81
	BBB	28.43±5.81	58.51±2.65	40.96±6.94	89.69±0.82	48.54±8.92	66.54±1.06	26.28±5.04	76.20±7.44
BNN	Flipout	23.54±2.67	62.77±1.30	40.31±3.33	91.33±0.76	68.30±1.33	65.58±2.68	24.07±2.25	75.89±2.65
	BBB	22.58±4.24	59.74±4.23	38.69±8.71	91.59±1.35	61.85±5.06	62.94±2.94	22.72±4.20	74.98±4.11
	LRT	25.22±1.80	62.59±1.09	42.23±3.93	91.47±1.42	69.33±2.28	65.57±2.16	25.83±2.42	77.79±1.59
	VI	20.72±5.56	55.79±3.38	33.92±6.17	91.19±1.70	65.89±4.76	60.30±2.49	19.91±5.57	72.56±4.02

sense of robustness against white-box adversarial attacks. The results shows inconsistency among different types of black-box attacks, especially adversarial samples based on Eq. 2. For instance, with $\varepsilon = 0.10$, SPSA shows better attack performance on GTSRB but MIM has the best attack performance on PlanesNet. SPSA fails every model for GTSRB, still due to the large perturbations to original inputs. Bayesian NN shows better performance against black-box attacks. In some cases, F-BNN has slightly better performance compare to BNN but others not. This peer comparison shows similar results with white-box attacks.

Overall, Bayesian NN shows remarkably robustness against all types of adversarial attacks except SPSA on GTSRB. This is due to the large, human-visible perturbations generated from SPSA. Larger adversarial perturbations cased by larger ε value makes the model perform worse, unless the model has already failed with small perturbations. BNN achieves comparable performance to F-BNN in the sense of classification accuracy, for both white-box and black-box attacks.

6. Conclusion

We highlight several discoveries here. Firstly, the Bayesian formulation of Neural Network can remarkably improve the performance of deep learning models, especially when dealing with constrained white-box adversarial attacks. Then, we notice that solely a Bayesian classifier is sufficient to improve model robustness. This decreases the time and space complexity with fewer parameter distributions. It's also insignificant when dealing with augmentation based input perturbations since classical CNN has already shows satisfactory denosing capabilities. Lastly, Bayesian neural network may fail. This happens when dealing images with human-visible modifications.

In the future, it worth looking at the possible reasons of good performance on image classification task. For instance, benefits of ensemble methods or the feature of Bayesian statistics. Also, it's interesting to look at the model robustness against attacks that are developed specifically toward BNNs (*e.g.* gradient-free adversarial attacks for BNN [24]). The Bayesian posterior observed from BNN can act as the safety precursor of ongoing malicious activities toward the deploy machine learning systems. This leads to the detection of adversarial samples in cybersecurity.

References

[1] Chirag Agarwal, Anh Nguyen, and Dan Schonfeld. Improving robustness to adversarial examples by encouraging discriminative features. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3801–3505. IEEE, 2019. 1

[2] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer, 2020. 2

[3] Jamil Azzeh, Bilal Zahran, and Ziad Alqadi. Salt and pepper noise: Effects and removal. *JOIV: International Journal on Informatics Visualization*, 2(4):252–256, 2018. 2

[4] 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. 1, 2

[5] Luca Cardelli, Marta Kwiatkowska, Luca Laurenti, Nicola Paoletti, Andrea Patane, and Matthew Wicker. Statistical guarantees for the robustness of bayesian neural networks. *arXiv preprint arXiv:1903.01980*, 2019. 1

[6] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019. 2

[7] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14, 2017. 1

[8] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017. 1, 2

[9] Planet's Open California dataset. Planes in satellite imagery, 2018. 1, 2

[10] Yinpeng Dong, Qi-An Fu, Xiao Yang, Tianyu Pang, Hang Su, Zihao Xiao, and Jun Zhu. Benchmarking adversarial robustness on image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 321–331, 2020. 1

[11] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 2

[12] Yinpeng Dong, Hang Su, Jun Zhu, and Fan Bao. Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv preprint arXiv:1708.05493*, 2017. 1

[13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2

[14] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288, 2013. 1, 2

[15] Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *arXiv preprint arXiv:1506.02557*, 2015. 1, 2

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1, 2

[17] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016. 2

[18] 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. 2

[19] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *International Conference on Machine Learning*, pages 2498–2507. PMLR, 2017. 1, 2

[20] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 2

[21] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th international conference on software engineering*, pages 303–314, 2018. 1

[22] Jonathan Uesato, Brendan O'donoghue, Pushmeet Kohli, and Aaron Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, pages 5025–5034. PMLR, 2018. 2

[23] Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386*, 2018. 1, 2

[24] Matthew Yuan, Matthew Wicker, and Luca Laurenti. Gradient-free adversarial attacks for bayesian neural networks. *arXiv preprint arXiv:2012.12640*, 2020. 4

[25] Huan Zhang, Hongge Chen, Zhao Song, Duane Boning, Inderjit S Dhillon, and Cho-Jui Hsieh. The limitations of adversarial training and the blind-spot attack. *arXiv preprint arXiv:1901.04684*, 2019. 1

[26] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 2

A. Appendix

A.1. Evaluation Results on PlanesNet Dataset

We list the classification results of PlanesNet Dataset in Table. 3 and Table. 4. The PlanesNet Dataset is open-source online. The objective is to classify the existence of aircrafts in surveillance satellite images. The PlanesNet has two class labels indicate *plane* or *not plane*. PlanesNet is another good example to demonstrate the potential benefits to the engineering applications in a safety-critical domain. Table. 3 is the results against input perturbations. Table. 4 is the results against adversarial attacks.

Table 3: Performance Against Input Perturbations. Report Test Accuracy in %.

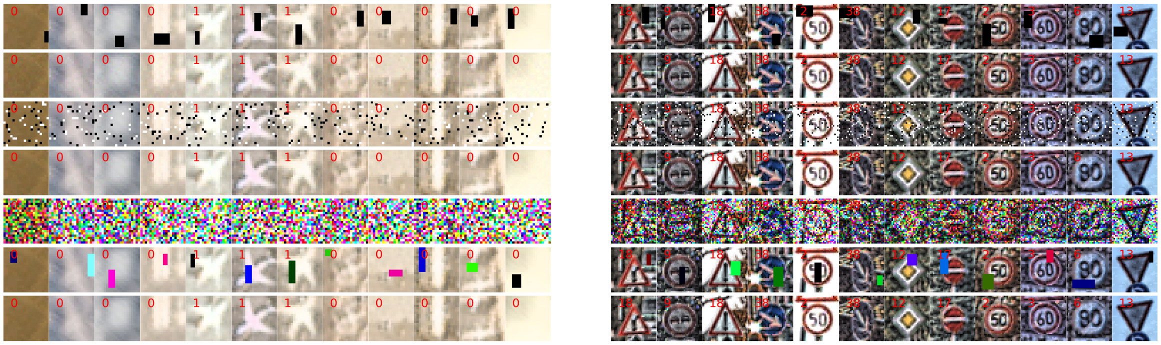
Dataset II: PlanesNet(Detect Aircraft in Planet Satellite Image Chips)								
Methods		Clean	Gaussian	S&P	Poisson	RE	RE Colorful	Speckle
CNN	Baseline	98.19±0.37	98.21±0.31	86.76±1.29	98.22±0.38	93.16±0.60	88.99±0.80	74.76±0.41
F-BNN	Flipout	97.83±0.12	97.79±0.21	91.91±0.41	97.86±0.15	93.06±0.24	89.43±0.75	74.80±0.90
	BBB	96.50±0.69	96.59±0.55	87.85±1.84	96.59±0.70	91.14±0.92	88.39±0.91	75.33±0.91
BNN	Flipout	98.55±0.28	98.44±0.28	84.06±3.02	98.53±0.26	92.21±0.57	89.44±0.52	74.79±0.23
	BBB	98.73±0.23	98.65±0.27	83.84±2.96	98.71±0.25	91.63±0.70	88.98±0.67	75.04±0.32
	LRT	98.88±0.12	98.87±0.11	83.00±1.88	98.89±0.11	92.30±0.55	89.04±0.21	74.78±0.12
	VI	96.41±2.72	96.23±2.82	82.73±3.41	96.34±2.67	89.86±2.31	87.01±1.33	72.96±3.25

Table 4: Performance Against Adversarial Attacks. Report Test Accuracy in %.

Dataset II: PlanesNet(Detect Aircraft in Planet Satellite Image Chips)									
$L_\infty(\epsilon = .10)$		White-box Attacks					Black-box Attacks		
Methods/Distance		PGD/0.068	FGSM/0.086	BIM/0.061	C&W/0.005	DeepF/0.349	SPSA/0.212	MIM/0.094	Square/0.080
CNN	Baseline	1.81 ± 0.37	45.44±2.40	13.50±2.28	68.21±2.13	43.77±4.83	49.65±0.91	1.83 ± 0.36	15.46±1.70
F-BNN	Flipout	24.51±3.70	57.69±2.37	36.78±4.35	91.13±0.41	54.90±2.84	60.88±0.79	23.14±2.71	75.36±6.81
	BBB	28.43±5.81	58.51±2.65	40.96±6.94	89.69±0.82	48.54±8.92	66.54±1.06	26.28±5.04	76.20±7.44
BNN	Flipout	23.54±2.67	62.77±1.30	40.31±3.33	91.33±0.76	68.30±1.33	65.58±2.68	24.07±2.25	75.89±2.65
	BBB	22.58±4.24	59.74±4.23	38.69±8.71	91.59±1.35	61.85±5.06	62.94±2.94	22.72±4.20	74.98±4.11
	LRT	25.22±1.80	62.59±1.09	42.23±3.93	91.47±1.42	69.33±2.28	65.57±2.16	25.83±2.42	77.79±1.59
	VI	20.72±5.56	55.79±3.38	33.92±6.17	91.19±1.70	65.89±4.76	60.30±2.49	19.91±5.57	72.56±4.02
Dataset II: PlanesNet(Detect Aircraft in Planet Satellite Image Chips)									
$L_\infty(\epsilon = .15)$		White-box Attacks					Black-box Attacks		
Methods/Distance		PGD/0.085	FGSM/0.127	BIM/0.084	C&W/0.007	DeepF/0.352	SPSA/0.217	MIM/0.139	Square/0.113
CNN	Baseline	1.81 ± 0.37	50.96±2.07	12.79±2.37	63.92±1.99	45.14±4.89	49.03±0.67	1.81 ± 0.36	12.53±2.90
F-BNN	Flipout	16.71±2.83	56.17±1.66	20.74±2.99	89.33±0.86	52.86±4.97	61.99±0.60	17.28±3.62	64.91±7.99
	BBB	21.33±5.90	57.48±0.57	23.98±3.64	87.85±1.27	52.00±3.38	68.57±1.92	19.77±4.45	68.76±7.46
BNN	Flipout	19.33±4.21	63.91±4.22	25.28±5.88	90.16±0.56	67.64±3.04	67.35±2.32	22.76±5.80	72.38±1.57
	BBB	16.85±3.95	63.92±3.16	24.04±3.61	90.32±0.84	65.43±2.05	67.89±2.97	19.54±4.39	72.58±2.80
	LRT	19.01±4.72	63.54±2.56	24.27±4.83	90.85±0.34	69.08±1.94	67.09±1.24	21.52±5.07	70.67±2.45
	VI	24.82±7.63	65.39±4.98	32.04±9.46	88.21±3.27	68.59±3.37	67.99±4.05	30.16±10.1	72.96±1.64

A.2. Visualization of the Data

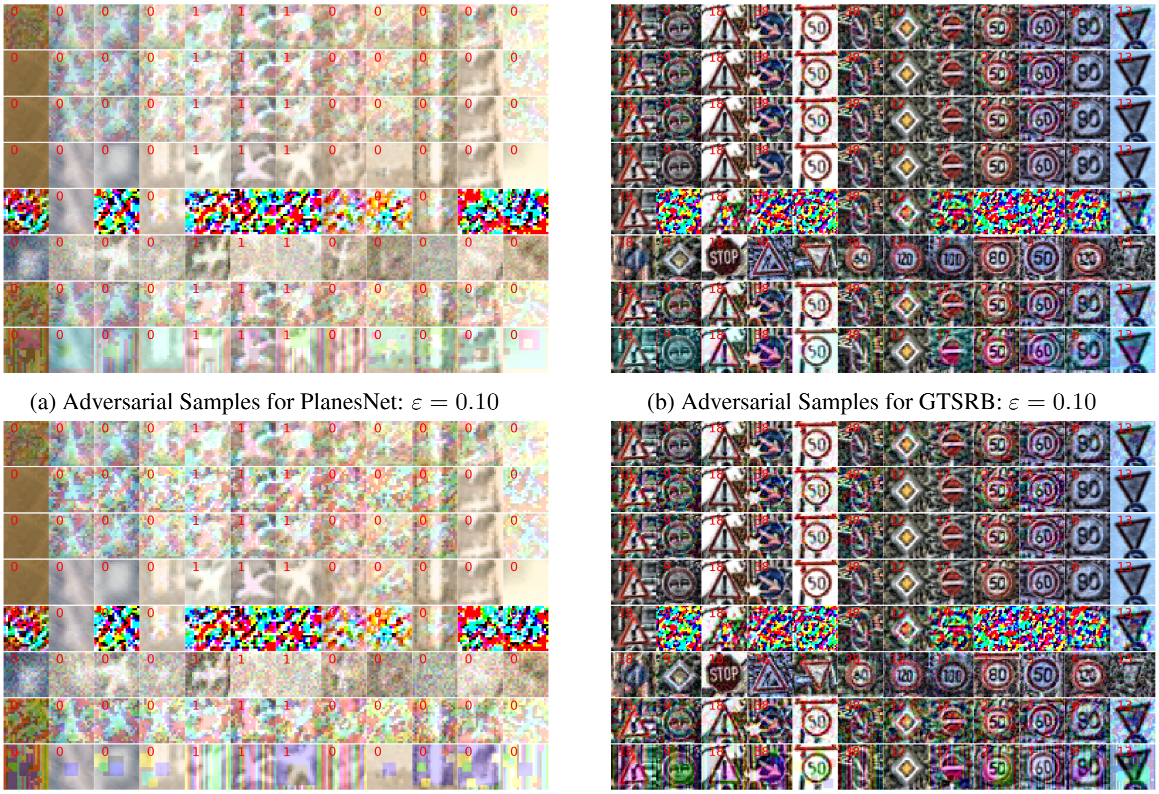
We visualize the data with input perturbations in Fig. 2. The first 12 samples in the test dataset are chosen with their correct class labels indicated at the top-left corner. The types of input perturbation are Random Erasing, Gaussian, Salt-and-Pepper, Possion, Speckle, Random Erasing Colorful, Clean. Similarly, the adversarial test samples are visualized in Fig. 3, with different perturbation budgets $\varepsilon = 0.10$ for (a) & (b), $\varepsilon = 0.15$ for (c) & (d).



(a) PlanesNet with Input Perturbations

(b) GTSRB with Input Perturbations

Figure 2: Visualization of applying input perturbations to PlanesNet and GTSRB. From top to bottom, the methods are: RE, Gaussian, S&P, Possion, Speckle, RE Colorful, Clean. The ground-truth for class labels are indicated at the top-left corner of each data plot. (a) PlanesNet. (b) GTSRB



(a) Adversarial Samples for PlanesNet: $\varepsilon = 0.10$

(b) Adversarial Samples for GTSRB: $\varepsilon = 0.10$

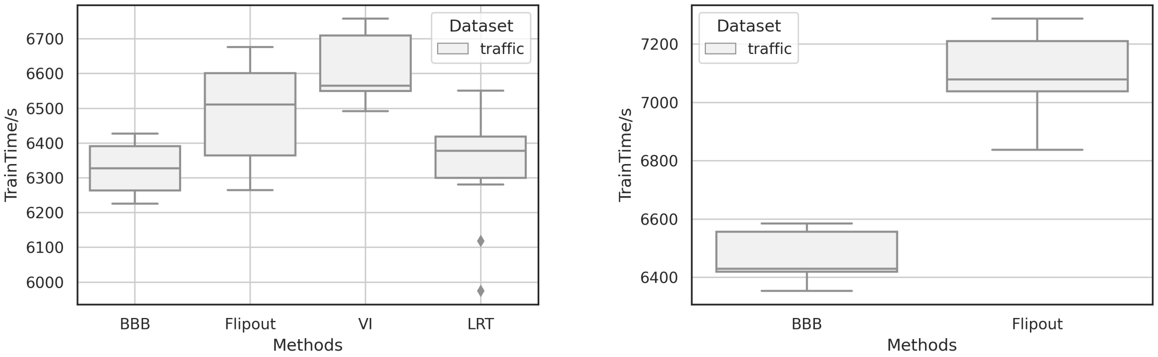
(c) Adversarial Samples for PlanesNet: $\varepsilon = 0.15$

(d) Adversarial Samples for GTSRB: $\varepsilon = 0.15$

Figure 3: Visualization of applying input perturbations to PlanesNet and GTSRB. From top to bottom, the methods are: PGD, FGSM, BIM, C&W, DeepF, SPSA, MIM, Square. The ground-truth for class labels are indicated at the top-left corner of each data plot. (a) PlanesNet. (b) GTSRB

A.3. Training Time Analysis

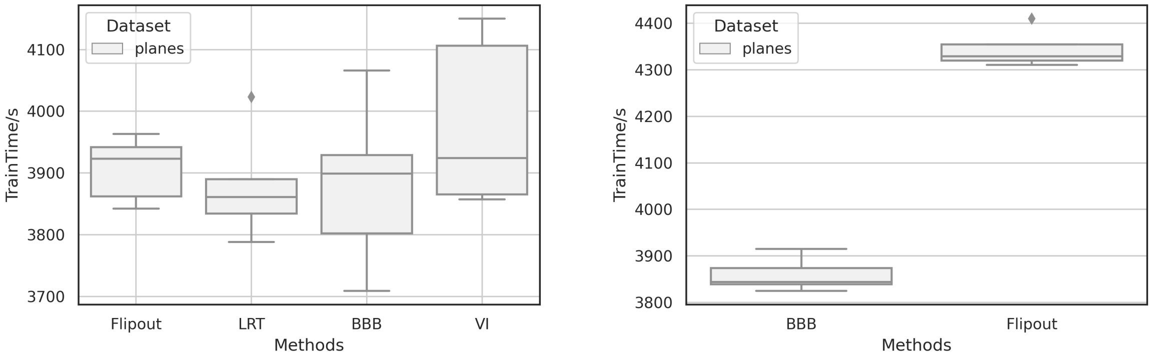
The training time of different methods on both two datasets are compared in Fig. 4 and Fig. 5. We can see BNNs require less training time.



(a) BNN Training time on GTSRB dataset

(b) F-BNN Training time on GTSRB dataset

Figure 4: Training time comparison with stochastic BNN. The figures for PlaneNet are reported in Fig. 5. (a) BNN with only classifier as stochastic. (b) F-BNN with stochasticity on every neural network layers.



(a) BNN on PlanesNet

(b) F-BNN on PlanesNet

Figure 5: Training time comparison with stochastic BNN. (a) BNN with only classifier as stochastic. (b) F-BNN with stochasticity on every neural network layers.