Bayesian Learning with Information Gain Provably Bounds Risk for a Robust Adversarial Defense

Bao Gia Doan [1] **Ehsan Abbasnejad** [1] **Javen Qinfeng Shi** [1] **Damith C. Ranashinghe** [1]

Abstract

We present a new algorithm to learn a deep neural network model robust against adversarial attacks. Previous algorithms demonstrate an adversarially trained Bayesian Neural Network (BNN) provides improved robustness. We recognize the adversarial learning approach for approximating the multi-modal posterior distribution of a Bayesian model can lead to mode collapse; consequently, the model's achievements in robustness and performance are sub-optimal. Instead, we first propose preventing mode collapse to better approximate the multi-modal posterior distribution. Second, based on the intuition that a robust model should *ignore perturbations* and only consider the informative content of the input, we conceptualize and formulate an *information gain objective* to measure and force the information learned from both benign and adversarial training instances to be similar. Importantly. we prove and demonstrate that minimizing the information gain objective allows the adversarial risk to approach the conventional empirical risk. We believe our efforts provide a step toward a basis for *a principled method of adversarially training BNNs*. Our model demonstrate significantly improved robustness–up to 20%–compared with adversarial training (Madry et al., 2018) and Adv-BNN (Liu et al., 2019) under PGD attacks with 0.035 distortion on both CIFAR-10 and STL-10 datasets.

1. Introduction

Deep neural networks (DNNs) have demonstrated *impressive* performance on challenging tasks, such as image recognition (He et al., 2016) and natural language processing (Vaswani et al., 2017). Despite the impressive performance, DNNs are poor at quantifying the predictive uncertainty and tend to produce overconfident predictions. Consequently, DNNs are shown to be vulnerable to easily crafted perturbations added to the inputs—so-called adversarial examples (AEs) (Szegedy et al., 2014)—to significantly hinder their performance. In image classification tasks, these perturbations are *imperceptible* to human eyes (Goodfellow et al., 2015) but can drastically degrade a DNN's performance. There are various methods to find such perturbations in whitebox (Madry et al., 2018; Goodfellow et al., 2015; Carlini & Wagner, 2017; Papernot et al., 2016a; Yuan et al., 2021) and blackbox settings (Brendel et al., 2018; Cheng et al., 2020; Chen et al., 2020; Vo et al., 2022a;b). Alarmingly, these threats are also shown to be effective in the physical world (Kurakin et al., 2018; Eykholt et al., 2018) and effective in transferring across models to perform *black-box* attacks (Papernot et al., 2016a; 2017). Adversarial perturbations pose a realistic threat for DNN applications and motivate the need to develop robust DNNs.

Adversarial Training. Despite the immense effort to overcome threats posed by adversarial examples, training a DNN robust against these attacks is challenging. Athalye et al. (2018a) have shown that one of the most robust defenses against the threat is Adversarial Training (Madry et al., 2018). Now, a network is trained with adversarial examples to build robustness against input perturbations post model deployment. But, as mentioned by Ye & Zhu (2018), the adversarial training algorithm relies on the "*point estimate*" approach of a deep neural network—a fixed set of network parameters maps the input to the output. Essentially, a point estimate with a choice of parameters only defines a single decision boundary that could be easily manipulated with a stronger adversarial input beyond the pre-defined adversarial constraints, *e.g.* maximum norm of perturbations. Alternatively, we can use multiple decision boundaries from a distribution of model parameters and integrate out the effects of parameter choice in the model. That is the premise of Bayesian Deep Neural Network (BNN) learning methods (Welling & Teh, 2011) aiming to learn a distribution over the model parameters. Now, the output *predictive* distribution is obtained by integrating out the model parameters sampled from their distribution.

[1]School of Computer Science, University of Adelaide, SA, Australia. Correspondence to: Bao Gia Doan <giabao.doan@adelaide.edu.au>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

Bayesian Adversarial Training. Motivated by the intuition that removing the effects of the parameter choice can lead to more robust models, Liu et al. (2019) proposed adversarial training of BNNs and demonstrated impressive results. However, training BNNs pose a significant challenge; the exact solution of the posterior distribution (*i.e.* the model parameter distribution after observing the data) is *intractable*. Efforts devoted to developing a suitable inference approach to approximate the posterior involve either using Markov Chain Monte Carlo (MCMC; asymptotically accurate but slow; see *e.g.* (Welling & Teh, 2011) or variational inference (efficient but inaccurate; see *e.g.* (Blei et al., 2017). For instance, Liu et al. (2019) uses a variational method named Bayes by Backprop (BBB) (Blundell et al., 2015) to approximate the posterior with a unimodal Gaussian distribution. Whilst being an efficient learning algorithm, the challenge faced with such a learning algorithm is the difficulty of capturing the multi-modal aspect of the posterior distribution because the parameters sampled are in the proximity of the mode of the distribution.

Our Hypothesis. We are motivated to explore the potential robustness gains attainable from an adversarial training algorithm for a Bayesian Deep Neural Network capable of approximating the multi-modality of the posterior. We hypothesize a model (1) learning a better approximation of the parameter distribution that (2) gains the same information from the given input and its adversarial counterpart is more robust.

Our Contribution. In this paper, to achieve (1), we *combine* adversarial training with an inference approach to faithfully capture the posterior distribution of parameters. The learning of an approximate multi-modal posterior is not new, inspired by Liu & Wang (2016), we employ Stein Variational Gradient Descent (SVGD) that encourages *diverse* sampling from the posterior. By utilizing the SVGD approach, to achieving (2), we design an *Information Gain* (IG)[1] objective. We summarize our contributions below:

- We propose a novel method to learn a BNN robust against adversarial attacks by utilizing SVGD to generate parameter particles that are parallelly trained to be *as diverse as possible* whilst *maintaining the same measure of information content learned from benign and adversarial instances*. Our learning approach enables the model to both reduce the effect of single parameter choice and learn the invariant patterns common between the training dataset and its corresponding adversarial samples.

- To maintain the same measure of information content learned from both benign and adversarial training instances, we formulate an information gain (mutual information) objective. Our proposed objective reinforces the minimization of the empirical adversarial risk by forcing the information gained, learning from the benign and adversarial samples, to be similar.

- We prove, *minimizing the information gain objective* allows the adversarial risk to approach the empirical risk minimization bound. Simply, the risk of misclassification of an adversarial example is now the same as the risk of misclassifying a benign sample. This is the first time such a bound is formally derived; this is significant because it provides a theoretically justified approach to reducing the uncertainty associated with adversarial examples.

- Comprehensive evaluations on a set of neural architectures and datasets demonstrate our approach achieves significant improvement in robustness compared to previous methods.

[1] also known as *Mutual Information* (Houlsby et al., 2011; Gal et al., 2017)

2. Background & Related Work

Primer on Bayesian Learning. Given a dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$, a Bayesian Neural Network (BNN) aims to learn the *posterior* distribution: $p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})}$ given the prior distribution $p(\boldsymbol{\theta})$. However, the exact solution for the posterior is often *intractable* since the deep neural networks are complex distributions and infeasible due to the high dimensional integral of the denominator even for moderately sized networks in the context of deep learning (Blei et al., 2017). In addition, the true Bayesian posterior is usually a complex multimodal distribution (Izmailov et al., 2021) as illustrated in Figure 1.

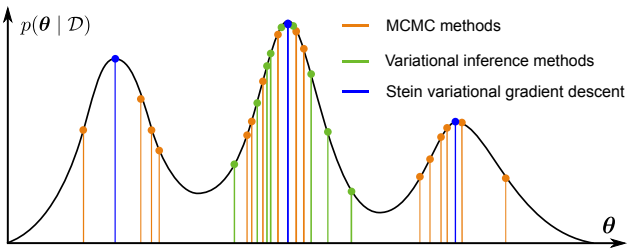

Figure 1: Different techniques to sample the posterior.

Variational inference, which relies on another parametric distribution, is too restrictive to resemble the true posterior and suffers from mode collapse (Izmailov et al., 2021). On the other hand, Wang & Liu (2019); Liu & Wang (2016) proposed a provable general purpose variational inference algorithm named Stein Variational Gradient Descent (SVGD) that transports a set of parameter particles, encouraged to be diverse, to fit the true posterior distribution; this approach can be beneficial for achieving higher performance and approximating the true posterior distribution. The visualization for different techniques to sample the posterior is

displayed in Figure 1.

Adversarial Attacks. Attackers can add carefully crafted noise (perturbations) to the input image to fool the classifier at the inference stage. In general, the goal of the attacker—described in Equation (1)—is to degrade the performance of a neural network by crafting $\boldsymbol{\delta}$, such that:

$$\max_{\|\boldsymbol{\delta}\|_p<\varepsilon_{\max}} \ell(f(\mathbf{x}+\boldsymbol{\delta};\boldsymbol{\theta}),y) \tag{1}$$

where, p is the norm, $\varepsilon_{\max}$ is the maximum attack budget (perturbation), ℓ is the loss function (typically cross-entropy), f is the network, $\mathbf{x}$ is the input, $\boldsymbol{\theta}$ is the network parameter, and y is the ground-truth label.

For a PGD (Madry et al., 2018) attack, an attacker starts from $\mathbf{x}^0 = \mathbf{x}_o$ and conducts projected gradient descent iteratively to update the adversarial example following the Equation (2):

$$\mathbf{x}^{t+1} = \Pi_{\varepsilon_{\max}}\left\{\mathbf{x}^t + \alpha \cdot \operatorname{sign}\left(\nabla_{\mathbf{x}}\ell\left(f\left(\mathbf{x}^t;\boldsymbol{\theta}\right),y_o\right)\right)\right\} \tag{2}$$

where $\Pi_{\varepsilon_{\max}}$ is the projection to the set $\left\{\mathbf{x} \mid \|\mathbf{x}-\mathbf{x}_o\|_\infty \le \varepsilon_{\max}\right\}$

Among all the attack methods, we decided to apply PGD in our experiments because: i) PGD (Madry et al., 2018) is regarded as the strongest attack in terms of the ℓ_∞ norm and ii) it gives us direct control over the distortion by changing ε_{max}.

Adversarial Defenses. Significant research efforts describe methods to mitigate this threat, such as distillation (Papernot et al., 2016b), input denoising (Song et al., 2017) or feature denoising (Xie et al., 2019), curious readers can find more from (Kurakin et al., 2018). Among these methods, adversarial training (Madry et al., 2018) and its variants are shown to be one of the most effective and popular methods to defend against adversarial attacks (Athalye et al., 2018a). The goal of adversarial training is to incorporate the adversarial search within the training process and, thus, realize robustness against adversarial examples at test time. This is achieved by solving the following optimization problem:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\mathbf{x},y)\sim D}\left\{\max_{\|\boldsymbol{\delta}\|_p<\varepsilon_{\max}} \ell(f(\mathbf{x}+\boldsymbol{\delta};\boldsymbol{\theta}),y)\right\} \tag{3}$$

where $\mathcal{D}$ is the training data. An approximate solution for the inner maximization can be realized by generating the PGD adversarial examples from Equation (2) and then minimizing the classification loss based on the generated PGD adversarial examples.

Recent works (Atsague et al., 2021; Zhu et al., 2020) also incorporate different variants of mutual information into their methods to realize a robust neural network. However, the mutual information is still utilized in a traditional "point-estimate" neural network setting; hence, the achieved robustness is marginal compared with traditional adversarial training. In contrast, this paper focuses on formulating mutual information (information gain) in a Bayesian neural network and theoretically prove that Bayesian adversarial learning with information gain allows the adversarial risk to approach the conventional risk.

Prior Art on Bayesian Defenses. Bayesian Neural Networks were proposed to detect adversarial attacks (Feinman et al., 2017; Smith & Gal, 2018). Recently, (Carbone et al., 2020) prove the robustness of BNNs to gradient-based adversarial attacks in the large data and overparameterized limit, while certified adversarial robustness on small $\varepsilon_{\max}$ was shown in (Wicker et al., 2021). On the other hand, Ye & Zhu (2018) and Liu et al. (2019) tried to combine Bayesian learning with adversarial training. Ye & Zhu (2018) present a method to jointly sample from the model's parameter posterior and the distribution of adversarial samples given the current parameter posterior to learn robust BNNs. Liu et al. (2019) further developed the direction proposed in Random Self-Ensemble (RSE) (Liu et al., 2018) to build an adversarially-trained Bayesian neural network method named Adv-BNN that can scales up to complex data by adding noise to each weight instead of input or hidden features as in RSE (Liu et al., 2018). Adv-BNN also incorporates adversarial training to learn a *variational posterior distribution* to further improve model robustness against strong adversarial examples with large $\varepsilon_{\max}$. However, using the variational inference method is likely to lead to mode collapse and limit the performance of the BNN (Izmailov et al., 2021) as we discussed earlier and demonstrate in our experiments in Section 4.

In this work, we propose exploring SVGD (Liu & Wang, 2016) as a Bayesian inference method to achieve a better approximation for the multi-modal posterior of a BNN. Using this approach, it is also easy to convert a traditional neural network to a Bayesian counterpart without much effort to modify the traditional neural network architecture. Further, by employing the *repulsive force* for encouraging exploration in the parameter space, we conceptualize the Information Gain in Bayesian learning to bound the difference of empirical risk versus the adversarial risk to further improve the robustness on strong adversarial examples.

3. Method

Our method combines adversarial training with an inference approach to faithfully capture the posterior distribution of parameters and formulate a new information gain objective in the setting to achieve a provably bounded adversarial risk to, hopefully, achieve a robust adversarial defense. We describe our formulation in what follows.

3.1. Bayesian Formulation for Adversarial Learning

In contrast to a point estimate learned in conventional deep learning models, in Bayesian learning, the posterior of the parameters is obtained using the Bayes rule *i.e.*:

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \prod_{(\mathbf{x},y)\sim\mathcal{D}} p(y \mid \mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta})/Z$$

where Z is the normalizer. Similarly, for the dataset of adversarials instances $\mathcal{D}_{\text{adv}}$, we obtain a corresponding posterior $p(\boldsymbol{\theta} \mid \mathcal{D}_{\text{adv}})$. We consider $p(y \mid \mathbf{x}_{\text{adv}}, \boldsymbol{\theta}) = \text{softmax}(f(\mathbf{x}_{\text{adv}}; \boldsymbol{\theta}))$ where f is a deep neural network. For adversarial dataset $\mathcal{D}_{\text{adv}}$, since adversarial examples can be generated from their corresponding benign instances, we can obtain $\mathcal{D}_{\text{adv}}$ during adversarial training by applying adversarial attacks such as PGD attacks. However, we acknowledge that PGD attacks cannot be directly applied in a BNN setting (Liu et al., 2019). Hence, to account for the uncertainty of BNNs, we utilize Expectation-over-Transformation (EoT) (Athalye et al., 2018b) approach to deploy an EoT PGD attack described in Equation (4); previously shown in Zimmermann (2019). This attack is more tailored for BNNs due to the fact that it achieves a more representative approximation to estimate the gradient and is formulated as:

$$\mathbf{x}^{t+1} = \Pi_{\varepsilon_{\max}} \left\{ \mathbf{x}^t + \alpha \cdot \text{sign}\left(\mathbb{E}_{\boldsymbol{\theta}} \left[\nabla_{\mathbf{x}} \ell\left(f\left(\mathbf{x}^t; \boldsymbol{\theta} \right), y_o \right) \right] \right) \right\}. \tag{4}$$

However, the posterior distribution, in general, is intractable and we need to resort to approximations. In particular, we propose utilizing Stein variational gradient descent (SVGD) (Liu & Wang, 2016) which provides an approach to learning multiple *particles* for parameters in parallel to approximate the true posterior. SVGD uses a repulsive factor to encourage the diversity of parameter particles to prevent mode collapse. This diversity enables learning multiple models to represent various patterns in the data. Collectively, the patterns are less vulnerable to adversarial attacks. Using n samples from the posterior (*i.e.* parameter particles) the variational bound is minimized when gradient descent is modified as:

$$\begin{aligned} \boldsymbol{\theta}_i = \boldsymbol{\theta}_i - \epsilon_i \hat{\boldsymbol{\phi}}^*(\boldsymbol{\theta}_i) \quad \text{with} \\ \hat{\boldsymbol{\phi}}^*(\boldsymbol{\theta}) = \sum_{j=1}^{n} \big[k(\boldsymbol{\theta}_j, \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}_j} \ell(f(\mathbf{x}_{\text{adv}}; \boldsymbol{\theta}_j), y) \\ - \frac{\gamma}{n} \nabla_{\boldsymbol{\theta}_j} k(\boldsymbol{\theta}_j, \boldsymbol{\theta}) \big]. \end{aligned} \tag{5}$$

Here, $\boldsymbol{\theta}_i$ is the ith particle, $k(\cdot,\cdot)$ is a kernel function that measures the similarity between particles, γ a hyperparameter and $\ell(\cdot,\cdot)$ is the cross entropy loss. Notably, the kernel function encourages the particles to be dissimilar to capture more diverse samples from the posterior and γ controls the trade-off between the diversity of the samples versus the minimization of the loss.

Further, given the test data point $\mathbf{x}^*$, we can approximate the posterior using the Monte Carlo samples as

$$\begin{aligned} p(y^* \mid \mathbf{x}^*, \mathcal{D}_{\text{adv}}) &= \int p(y^* \mid \mathbf{x}^*, \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}_{\text{adv}}) d\boldsymbol{\theta} \\ &\approx \frac{1}{n} \sum_{i=1}^{n} p(y^* \mid \mathbf{x}, \boldsymbol{\theta}_i), \quad \boldsymbol{\theta}_i \sim p(\boldsymbol{\theta} \mid \mathcal{D}_{\text{adv}}), \end{aligned}$$

where $\boldsymbol{\theta}_i$ is an individual parameter particle.

Importantly, in the adversarial setting, it is critical to take parameter samples that represent different modes of the distribution that may not have the same vulnerabilities towards perturbations. The adversarial instances are generally known to exploit the particular patterns learned by the parameters (Papernot et al., 2016c). When integrating out the parameters as in the Bayesian setting, especially under the diverse parameter particles in our approach, we implicitly remove the vulnerabilities that could arise from a single choice of a parameter.

3.2. Conceptualizing Information Gain for Bayesian Learning

Using the Bayesian setting we employ, we can formulate a notion of information gain that captures the impact of adding a new training instance to a dataset on the distribution of the parameters. The information gain can be defined as (see Appendix A):

$$\text{IG}(\mathbf{x}, y; \boldsymbol{\Theta}) = \mathbb{H}[\mathbb{E}_{\boldsymbol{\theta}}[y|\mathbf{x}, \mathcal{D}]] - \mathbb{E}_{\boldsymbol{\theta}}[\mathbb{H}[y|\mathbf{x}, \mathcal{D}]]. \tag{6}$$

This formulation quantifies an instance's informativeness for a model given the training set. Intuitively, the information gained from an instance is proportionate to the reduction in the expected entropy of the predictive distribution.

Our conjecture is that *a robust neural network quantifies the information gain from an observation the same as its adversarial counterpart i.e.* $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\text{IG}(\mathbf{x}, y; \boldsymbol{\Theta})] = \mathbb{E}_{(\mathbf{x}_{\text{adv}},y)\sim\mathcal{D}_{\text{adv}}}[\text{IG}(\mathbf{x}_{\text{adv}}, y; \boldsymbol{\Theta})]$. In other words, a robust model ignores the perturbations and only considers the informative content of the input. We will employ these concepts in the following learning formulation.

3.3. Formulate Learning a Robust Network Using Information Gain

We formulate the objective of our training to:

1. Learn the posterior from the *adversarial* dataset. Since we use SGVD, this corresponds to learning multiple parameter particles. This amounts to minimizing the loss subject to the repulsive constraint, *i.e.* $\mathbb{E}_{(\mathbf{x}_{\text{adv}},y)\sim\mathcal{D}_{\text{adv}}} \left[\mathbb{E}_{\boldsymbol{\theta}\sim\text{I}(\boldsymbol{\theta}|\mathcal{D}_{\text{adv}})} [\ell(f(\mathbf{x}_{\text{adv}}; \boldsymbol{\Theta}), y)] \right]$. Since the adversarial dataset is generated while training the model,

it depends on the particle chosen and its parameters. With SGVD, we ensure the samples are diverse, and each parameter particle explores a different pattern in the input.

2. Achieve comparable information gain from both the given dataset and the adversarials. Thus, ensuring: i) the information gained from data and adversarial examples is encouraged to be the same, *i.e.* $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\text{IG}(\mathbf{x},y;\boldsymbol{\Theta})] = \mathbb{E}_{(\mathbf{x}_{\text{adv}},y)\sim\mathcal{D}_{\text{adv}}}[\text{IG}(\mathbf{x}_{\text{adv}},y;\boldsymbol{\Theta})]$; ii) the model to be *not* biased towards learning from the adversarial instances; and iii) the receptive fields are active for similar and prominent features.

To this end, we formulate the problem as a constrained optimization:

$$\min_{\boldsymbol{\theta}} \quad \mathbb{E}_{(\mathbf{x}_{\text{adv}},y)\sim\mathcal{D}_{\text{adv}}}\left[L(\mathbf{x}_{\text{adv}},y;\boldsymbol{\Theta})\right] \tag{7}$$
$$\text{s.t.} \quad \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\text{IG}(\mathbf{x},y;\boldsymbol{\Theta})] = \mathbb{E}_{(\mathbf{x}_{\text{adv}},y)\sim\mathcal{D}_{\text{adv}}}[\text{IG}(\mathbf{x}_{\text{adv}},y;\boldsymbol{\Theta})]$$

where $L(\mathbf{x}_{\text{adv}},y;\boldsymbol{\Theta}) = \mathbb{E}_{\boldsymbol{\theta}\sim\text{I}(\boldsymbol{\theta}|\mathcal{D}_{\text{adv}})}[\ell(f(\mathbf{x}_{\text{adv}};\boldsymbol{\theta}),y)]$. Combining the above concepts using the Lagrangian method, we have the following objective:

$$L_{\text{IG}}(\boldsymbol{\Theta}) = L(\mathbf{x}_{\text{adv}},y;\boldsymbol{\Theta}) + \lambda|\text{IG}(\mathbf{x},y;\boldsymbol{\Theta}) - \text{IG}(\mathbf{x}_{\text{adv}},y;\boldsymbol{\Theta})| \tag{8}$$

where we use the Monte Carlo sampling using the particles to estimate the expectations. Subsequently, this learning objective $L_{\text{IG}}(\boldsymbol{\Theta})$ is optimized using the SVGD method in Equation (5) mentioned earlier in Section 3.1. Effectively, using this approach, we compute a posterior in a constrained space defined by the information gain criteria. Since the space is constrained, the likelihood of findings "particles" in the posterior that are more robust increases. We summarize our proposed robust Bayesian learning approach in Algorithm 1. Here, following Liu & Wang (2016), we use the RBF kernel $k(\boldsymbol{\theta},\boldsymbol{\theta}') = \exp\left(-\|\boldsymbol{\theta}-\boldsymbol{\theta}'\|^2/(2h^2)\right)$ and take the bandwidth h to be the median of the pairwise distances of the set of parameter particles at each iteration.

3.4. A Relation between Adversarial and Observational Training

A typical machine learning approach minimizes the empirical risk to learn. There are theoretical and empirical studies on the relation between the empirical risk and the true risk that measures the generalization ability of a learning algorithm. Generalization bounds such as Rademacher complexity or VC dimension for classical approaches or more recent studies for deep learning (see *e.g.* (Neyshabur et al., 2017)) underpin the theoretical framework for machine learning.

Notably, the relation between the risk when using samples from the *observational distribution* (*i.e.* the given dataset) versus when using their adversarial counterparts remains unexplored. *It is important, because, while adversarial training has been commonly used, the impact of using such an approach on generalization with respect to the true data distribution is unknown.* We consider a Bayesian model

Algorithm 1 Information Gain-BNN (IG-BNN)

Input: A set of initial parameter particles $\{\boldsymbol{\theta}_i^0\}_{i=1}^n$, observation data $\mathcal{D}$.
Output: A set of parameter particles $\boldsymbol{\Theta} := \{\boldsymbol{\theta}_i\}_{i=1}^n$ that approximates the true posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{D}_{\text{adv}})$
for $(\mathbf{x},y) \sim p(\mathcal{D})$ **do**
  $\mathbf{x}_{\text{adv}} \leftarrow \mathbf{x}$
  for $t = 1 \rightarrow T$ **do**
    $\mathbf{x}_{\text{adv}} = \Pi_{\varepsilon_{\max}}\{\, \mathbf{x}_{\text{adv}} + \alpha \cdot \text{sign}\,(\mathbb{E}_{\boldsymbol{\theta}}\left[\nabla_{\mathbf{x}}\ell\left(f\left(\mathbf{x}_{\text{adv}};\boldsymbol{\theta}_j\right),y\right)\right]\,)\,\}$
    {Generate Adversarial (Eq. (4))}
  end for
  for $i = 1 \rightarrow n$ **do**
    $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_i - \epsilon_i\hat{\phi}^*(\boldsymbol{\theta}_i,\boldsymbol{\theta}_j)$ with $\hat{\phi}^*(\boldsymbol{\theta}_i,\boldsymbol{\theta}_j) = \sum_{j=1}^n\left[k(\boldsymbol{\theta}_j,\boldsymbol{\theta}_i)\nabla_{\boldsymbol{\theta}_j}L_{\text{IG}}(\boldsymbol{\Theta}) - \frac{\gamma}{n}\nabla_{\boldsymbol{\theta}_j}k(\boldsymbol{\theta}_j,\boldsymbol{\theta}_i)\right]$
    ϵ_i is the step size at the current iteration, $k(\boldsymbol{\theta},\boldsymbol{\theta}')$ is a positive definite kernel that specifies the similarity between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, L_{IG} is the main objective (Eq. (8)), γ,λ is the weight to control the *repulsive force* that enforces the diversity among parameter particles and IG objective respectively, ℓ is the cross-entropy loss function.
  end for
end for

with no specific assumption on the distribution of either the adversarial examples or the perturbations to provide a generic defense approach. The only major assumption we make for the following adversarial risk bound is that the distribution of the data and the corresponding adversarial are sufficiently close. That is a mild assumption when we consider the adversarial instances are obtained from small perturbations of the given training dataset. Thus, we are interested in finding the bound of $|R_{\text{adv}} - R|$ where

$$R = \mathbb{E}_{\boldsymbol{\theta}}\left[\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{E}_{y'\sim p(y|\mathbf{x},\boldsymbol{\theta})}\left[\mathbb{I}(y=y')\right]\right]\right]$$

is the empirical risk, and

$$R_{\text{adv}} = \mathbb{E}_{\boldsymbol{\theta}}\left[\mathbb{E}_{(\mathbf{x}_{\text{adv}},y)\sim\mathcal{D}_{\text{adv}}}\left[\mathbb{E}_{y'\sim p(y|\mathbf{x}_{\text{adv}},\boldsymbol{\theta})}\left[\mathbb{I}(y=y')\right]\right]\right]$$

is the risk of the adversarial examples. Once we can obtain these, we can simply obtain the overall generalization and robustness bound. The following proposition summarizes our findings.

Proposition 1. *The risk of a classifier when trained on the observed training set denoted by R versus when trained with adversarials denoted by R_{adv} is bounded as*

$$|R_{adv} - R| \leq 1 - \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\Bigg[\exp\Bigg(\left(\mathbb{E}_{\boldsymbol{\theta}}[r_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{x}_{adv},y)] - \lambda\left|\mathbb{E}_{\boldsymbol{\theta}}[IG(\boldsymbol{x},y;\boldsymbol{\Theta})] - \mathbb{E}_{\boldsymbol{\theta}}[IG(\boldsymbol{x}_{adv},y;\boldsymbol{\Theta})]\right|\right)\Bigg)\Bigg],$$

where $r_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}_{adv}, y) = \sum_c^K p(y = c \mid \boldsymbol{x}, \boldsymbol{\theta}) \log(p(y = c \mid \boldsymbol{x}_{adv}, \boldsymbol{\theta}))$, $\lambda \geq 0$ *and* $\boldsymbol{x}_{adv}$ *denotes the adversarial example obtained from* $\boldsymbol{x}$.

Sketch of the Proof. We simplify the difference between the risks by considering that the difference between individual mistakes is smaller than their product, *i.e.*

$$\begin{aligned}
&\mathbb{E}_{y_1 \sim p(y|\mathbf{x}, \boldsymbol{\theta})} \left[\mathbb{E}_{y_2 \sim p(y|\mathbf{x}_{\text{adv}}, \boldsymbol{\theta})} \left[\mathbb{I}[y \neq y_1] - \mathbb{I}[y \neq y_2] \right] \right] \\
&\quad \leq \mathbb{E}_{y' \sim p(y|\mathbf{x}_{\text{adv}}, \boldsymbol{\theta})} \left[\mathbb{E}_{y' \sim p(y|\mathbf{x}_{\text{adv}}, \boldsymbol{\theta})} \left[\mathbb{I}[y_1 \neq y_2] \right] \right] \\
&\quad \leq 1 - \sum_{c=1}^{K} p(y = c \mid \mathbf{x}, \boldsymbol{\theta}) p(y = c \mid \mathbf{x}_{\text{adv}}, \boldsymbol{\theta}) \,.
\end{aligned}$$

We then use Jensen's inequality when using $\exp(\log(\cdot))$ to obtain the upper bound. The complete proof is provided in the Appendix B. We can see that the difference between the empirical risk and the adversarial risk is minimized when the upper bound is minimized. Hence, to minimize the upper bound, our main learning objectives are to:

1. *Minimize cross entropy for the adversarial examples*. This corresponds to matching the prediction from the adversarial data to that of the observations. Since $(\mathbf{x}, y)$ is given in the training, we simply minimize the entropy of the adversarial examples. This corresponds to using a cross-entropy loss in Eq. (7).

2. *Minimize the difference between the information gained from the dataset and its adversarial counterparts*. In addition to individual predictions, the information gained from each instance (*i.e.* the benign and its adversarial) has to have a similar impact in terms of how it changes the network parameters.

Notably, since we know $1 - \exp(-z) \leq z$, to avoid computational instabilities and gradient saturation, we consider minimizing the upper bound without the exponential function.

Our proposed algorithm is summarized in Algorithm 1.

4. Experimental Results

In this section, we verify the performance of our proposed method (IG-BNN) with other baselines in the literature on two popular and standard vision tasks. We use the CIFAR-10 (Krizhevsky et al.) dataset—a popular benchmark used to evaluate the robustness of a DNN in previous works (Madry et al., 2018; Athalye et al., 2018a). However, it is also known that adversarial training becomes increasingly hard for high-dimensional data (Schmidt et al., 2018). Therefore, to further evaluate the robustness of our method, we conduct an experiment on a high dimensional dataset—STL-10 (Coates et al., 2011) with 5,000 training images and 8,000 testing images with images of 96×96 pixels.

In all experiments, we utilized the same networks used in the adversarial training BNN method, Adv-BNN (Liu et al., 2019) to fairly compare the results. Specifically, we used the VGG-16 network architecture for CIFAR-10 and the smaller ModelA network for STL-10 used in Liu et al. (2019). The number of PGD steps and the attack budgets used for training and testing are also set to be the same for a fair comparison—see Appendix C Table 5. Notably, we also conduct the experiment with a larger number of PGD steps in the Appendix D, and Figure 5 confirm that 20-step is enough for the EoT PGD attack to reach its full strength.

Because our proposed method evaluates the robustness of a Bayesian learning method based on Adversarial Training, the traditional Adversarial Training (Adv. Training) (Madry et al., 2018) and adversarially trained Bayesian defense, Adv Bayesian Neural Network (Adv-BNN) (Liu et al., 2019) are good baselines for comparisons. In addition, we also compare our method with networks trained with no defenses (No Defense) as well as Bayesian Neural Networks trained for the tasks.

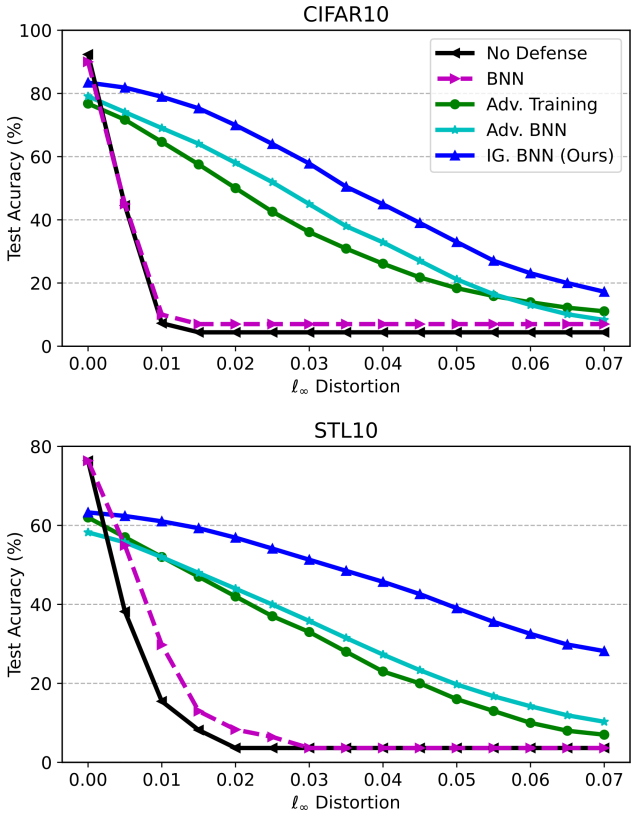

Figure 2: Accuracy under ℓ_∞-EoT PGD attack on different datasets. CIFAR-10 is trained on a VGG-16 network, and STL-10 is trained on ModelA–used in Adv-BNN (Liu et al., 2019).

4.1. Robustness Under White-box l_∞ Attacks

In this experiment, we compare the robustness of our models under the strong white-box l_∞-EoT PGD attack. Following the recent work in (Liu et al., 2019), we set the maximum l_∞ distortion to $\varepsilon_{\max} \in [0 : 0.07 : 0.005]$, adjust the PGD attacks for Bayesian methods as mentioned earlier– see Equation (4)—and report the accuracy on the test set (*robustness*). Overall, the results—shown in Figure 2— illustrate the improved robustness of our method compared with Adv. BNN (Liu et al., 2019), and the significantly better results compared to Adv. Training (Madry et al., 2018). We also provide detailed results in Table 1 where we show a marked increase in testing accuracy (benign) and robustness (against adversarial samples)—notably, IG-BNN achieves up to 17% at the distortion of 0.035 compared with Adv-BNN and 20% compared with Adv. Training on STL-10 dataset. These correspond to 13% on CIFAR-10 and 19% on STL-10, respectively. Although Adv-BNN helped improve robustness, we can see that the learning method is still below what could be achieved. On the other hand, IG-BNN achieved better results on both the testing data (*benign*) and adversarial examples (under increasing attack budgets).

Table 1: Comparing robustness under different levels of EoT PGD attacks (or attack budgets).

Data	*Defenses*	*0*	*0.015*	*0.035*	*0.055*	*0.07*
CIFAR-10	Adv. Training	80.3	58.3	31.1	15.5	10.3
	Adv-BNN	79.7	64.2	37.7	16.3	8.1
	IG-BNN (Ours)	**83.6**	**75.5**	**50.2**	**26.8**	**16.9**
STL-10	Adv. Training	63.2	46.7	27.4	12.8	7.0
	Adv-BNN	59.9	47.9	31.4	16.7	9.1
	IG-BNN (Ours)	**64.3**	**60.0**	**48.2**	**34.9**	**27.3**

4.2. Ablative Studies

In this section, we investigate the contribution of each of the formulations in our method. Particularly, we investigate: i) the contribution of the Bayesian inference method SVGD; and ii) the contribution of Information Gain (IG). We utilize the same network architecture and training parameters for the *higher resolution*, therefore more challenging, STL-10 dataset with the only difference being the ablative parameter to conduct the experiment.

Bayesian Inference Methods. We evaluate the network trained with the adversarial training using the Bayesian inference method proposed in Liu et al. (2019), that is Bayes by Backprop (Adv train + BBB), to compare with our proposed adversarially trained BNN using SVGD (Adv train + SVGD). The results are in Table 2. We can see that employing SVGD with the ability to capture a multi-model posterior contributed to improving the robustness of the Adversarial trained Bayesian Neural Networks.

Table 2: Ablative study on assessing the contribution of the Bayesian inference method under different levels of EoT PGD attacks (or attack budgets).

Defenses	*0*	*0.015*	*0.035*	*0.055*	*0.07*
Adv train + BBB	59.9	47.9	31.4	16.7	9.1
Adv train + SVGD	**63.6**	**54.2**	**36.6**	**24.3**	**19.4**

Information Gain. With the improvements in robustness achieved with the SVGD formulation for adversarial training, we conduct the ablative study on the network trained with SVGD inference method with and without IG to assess the impact of the IG objective on robustness. Notably, the trivial solution for the IG objective is that all parameter particles collapse to a single mode; hence, the IG objective and its effectiveness can be achieved with the inference methods encouraging diversity, such as SVGD.

Table 3: Ablative study on assessing the contribution of the Information Gain objective under different levels of EoT PGD attacks (or attack budgets).

Defenses	*0*	*0.015*	*0.035*	*0.055*	*0.07*
Adv train + SVGD	63.6	54.2	36.6	24.3	19.4
Adv train + SVGD + IG	**64.3**	**60.0**	**48.2**	**34.9**	**27.3**

As shown in Table 3, we can see that IG helped improve robustness further, up to 12%. We also empirically demonstrate the difference in empirical risk and the adversarial risk evaluated on the test set in Figure 3. Our empirical results demonstrate the impact of adding IG to tighten the bound and reduce the gap between conventional empirical risk and the adversarial risk; consequently, improving the robustness of the network.

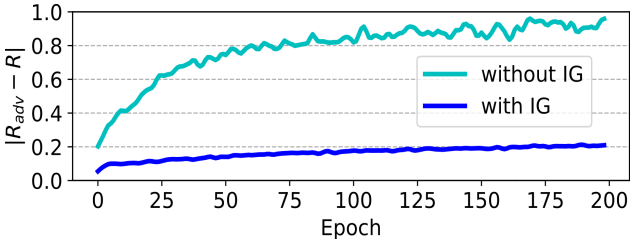

Figure 3: The difference between conventional empirical risk and adversarial risk $|R_{\text{adv}} - R|$ on the test set is tightened and minimized when training the BNN with Information Gain. Corroborating our proof, the empirical result further explains the improvement in robustness of the IG-BNN networks.

4.3. Evaluating the Obfuscated Gradient Effect

One possible failure mode of defense methods discussed in the literature is the obfuscated gradient effect (Athalye et al.,

2018a) where seemingly high adversarial accuracy is only superficial and creates false robustness. In this scenario, the network learns to obfuscate the gradients whilst showing seeming robustness by making it harder for the attack to find perturbations. However, an easy and effective way to verify this is to apply a black-box attack on the defense methods. The defense is considered to show an obfuscated gradient effect if the black-box attack is more successful than the white-box attack (*i.e.* the robustness is lower).

Following current practice, in this experiment, we deploy a black-box Square attack (Andriushchenko et al., 2020) on our IG-BNN models. Table 4 shows that our IG-BNN is also highly robust against the black-box attack and, more importantly, the robustness of the black-box attack is significantly higher than the white-box one. Particularly, the robustness against black-box attacks on CIFAR-10 at the distortion of 0.035 is 78.9% which is a 28% accuracy improvement compared with its white-box counterpart. On STL-10, at the same distortion, this improvement is 13%. These results demonstrate that our robustness is not simply the effect of obfuscated gradients.

Table 4: Blackbox attack to evaluate the obfuscated gradient effect.

Data	*Defenses*	*0*	*0.015*	*0.035*	*0.055*	*0.07*
CIFAR-10	IG-BNN (Ours)	83.6	75.5	50.2	26.8	16.9
	Black-box	-	82.3	78.9	71.0	63.2
STL-10	IG-BNN (Ours)	64.3	60.0	48.2	34.9	27.3
	Black-box	-	63.8	61.3	59.3	57.6

4.4. Transfer Attacks Among Parameter Particles

To further evaluate the robustness and illustrate the intuition for exploring diverse parameter particles, we conduct experiments on the transferability of the adversarial examples among parameter particles and evaluate the robustness at class-wise levels (i.e. the robustness in each class).

Specifically, we sample multiple different parameter particles for the experiment. For each parameter particle (*source particles*), we generate corresponding adversarial examples for that parameter particle. And then, using those adversarial examples generated from the source particles, attack and evaluate the robustness of other particles (*target particles*). We visualize the results as heatmaps with robustness as the measure (i.e. the ability to correctly identify the adversarial examples), and show the results in Figure 4— We provide comprehensive results in the Appendix G. Each row in the matrix shows the robustness of target particles against the adversarial examples generated from the source particles (with the attack budget $\epsilon = 0.015$).

As expected, we can observe that the adversarial examples are highly effective on their source particles with 0% robustness. However, other particles are able to recognize those adversarial examples correctly with high robustness. This further demonstrates the effectiveness of our learning algorithm where we encourage the parameter particles to be diverse and additionally bound the difference of empirical risk versus the adversarial risk in terms of the information gain formulation.

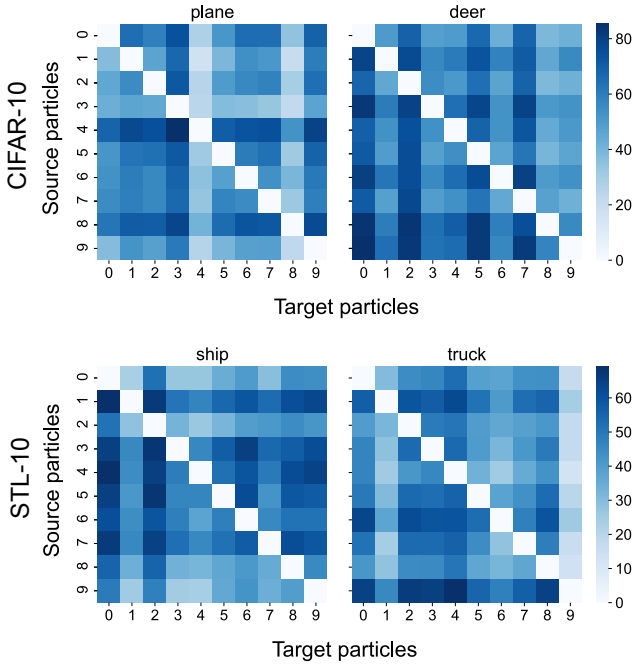

Figure 4: Diversity of parameter particles is demonstrated using the transferability of adversarial examples among particles. We provide comprehensive results in Appendix G.

5. Conclusion

In this study, we presented a novel method to learn a robust BNN against adversarial attacks. We demonstrate that, although an adversarially trained BNN improved robustness, the improvement is slight compared with the traditional adversarial training when using the EoT PGD attack tailored for BNNs. Our proposed IG-BNN learning method employing SVGD to encourage diverse parameter particles together with the formulated information gain objective under the Bayesian context provably bounds the difference of empirical risk versus adversarial risk to yield improved robustness. The empirical experiments demonstrate that learning a Bayesian neural network using our method tightens the gap between the empirical risk and the empirical adversarial risk; this, consequently leads to better robustness compared with previous adversarially trained Bayesian defense methods.

References

Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision (ECCV)*, 2020.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2018a.

Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. Synthesizing robust adversarial examples. In *International conference on machine learning (ICLR)*, 2018b.

Atsague, M., Fakorede, O., and Tian, J. A mutual information regularization for adversarial training. In *Asian Conference on Machine Learning*, pp. 188–203. PMLR, 2021.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 2017.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural network. In *International Conference on Machine Learning (ICML)*, 2015.

Brendel, W., Rauber, J., and Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations (ICLR)*, 2018.

Carbone, G., Wicker, M., Laurenti, L., Patane, A., Bortolussi, L., and Sanguinetti, G. Robustness of Bayesian Neural Networks to Gradient-Based Attacks. In *Advances in Neural Information Processing Systems NeurIPS*, 2020.

Carlini, N. and Wagner, D. Towards Evaluating the Robustness of Neural Networks. In *IEEE Symposium on Security and Privacy (S&P)*, 2017.

Chen, J., Jordan, M. I., and Wainwright, M. J. Hopskipjumpattack: A query-efficient decision-based attack. In *IEEE Symposium on Security and Privacy (S&P)*, 2020.

Cheng, M., Singh, S., Chen, P., Chen, P.-Y., Liu, S., and Hsieh, C.-J. Sign-opt: A query-efficient hard-label adversarial attack. In *International Conference on Learning Representations (ICLR)*, 2020.

Coates, A., Ng, A., and Lee, H. An Analysis of Single Layer Networks in Unsupervised Feature Learning. In *AISTATS*, 2011.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. Robust physical-world attacks on deep learning models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

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

Gal, Y., Islam, R., and Ghahramani, Z. Deep Bayesian Active Learning with Image Data. In *International Conference on Machine Learning (ICML)*, 2017.

Goodfellow, I., Shlens, J., and Szegedy, C. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations (ICLR)*, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

Houlsby, N., Huszár, F., Ghahramani, Z., and Lengyel, M. Bayesian Active Learning for Classification and Preference Learning. 2011.

Izmailov, P., Vikram, S., Hoffman, M. D., and Wilson, A. G. What are bayesian neural network posteriors really like? In *International Conference on Machine Learning (ICML)*, 2021.

Krizhevsky, A., Nair, V., and Hinton, G. Cifar-10 (canadian institute for advanced research). URL `http://www.cs.toronto.edu/~kriz/cifar.html`.

Kurakin, A., Goodfellow, I., Bengio, S., Dong, Y., Liao, F., Liang, M., Pang, T., Zhu, J., Hu, X., Xie, C., et al. Adversarial attacks and defences competition. In *The NIPS'17 Competition: Building Intelligent Systems*. 2018.

Liu, Q. and Wang, D. Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm. *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Liu, X., Cheng, M., Zhang, H., and Hsieh, C.-J. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

Liu, X., Li, Y., Chongruo, W., and Cho-Jui, H. ADV-BNN: Improved Adversarial Defense Through Robust Bayesian Neural Network. In *International Conference on Learning Representations (ICLR)*, 2019.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to

adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

Neyshabur, B., Bhojanapalli, S., Mcallester, D., and Srebro, N. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Papernot, N., McDaniel, P., and Goodfellow, I. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016a.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. The limitations of deep learning in adversarial settings. In *IEEE European symposium on security and privacy (EuroS&P)*, 2016b.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pp. 372–387. IEEE, 2016c.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security (Asia CCS)*, 2017.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Madry, A. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

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

Song, Y., Kim, T., Nowozin, S., Ermon, S., and Kushman, N. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems (NIPS)*, 2017.

Vo, V. Q., Abbasnejad, E., and Ranasinghe, D. C. Ramboattack: A robust query efficient deep neural network decision exploit. In *Network and Distributed Systems Security (NDSS) Symposium*, 2022a.

Vo, V. Q., Abbasnejad, E., and Ranasinghe, D. C. Query efficient decision based sparse attacks against black-box deep learning models. In *International Conference on Learning Representations (ICLR)*, 2022b.

Wang, D. and Liu, Q. Nonlinear stein variational gradient descent for learning diversified mixture models. In *International Conference on Machine Learning (ICML)*, 2019.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *International Conference on Machine Learning (ICML)*, 2011.

Wicker, M., Laurenti, L., Patane, A., Chen, Z., Zhang, Z., and Kwiatkowska, M. Bayesian Inference with Certifiable Adversarial Robustness. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics (AISTATS*, 2021.

Xie, C., Wu, Y., Maaten, L. v. d., Yuille, A. L., and He, K. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Ye, N. and Zhu, Z. Bayesian adversarial learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Yuan, M., Wicker, M., and Laurenti, L. Gradient-Free Adversarial Attacks for Bayesian Neural Networks. In *Advances in Approximate Bayesian Inference (AABI)*, 2021.

Zhu, S., Zhang, X., and Evans, D. Learning adversarially robust representations via worst-case mutual information maximization. In *International Conference on Machine Learning*, pp. 11609–11618. PMLR, 2020.

Zimmermann, R. S. Comment on" adv-bnn: Improved adversarial defense through robust bayesian neural network". *arXiv preprint arXiv:1907.00895*, 2019.

A. Definition of Information Gain

We first define our predictive distribution as:

$$p(y|\mathbf{x},\mathcal{D}) = \int p(y|\mathbf{x},\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}\,.$$

Following the definition of information gain, we have:

$$\begin{aligned}
\mathbb{E}[\mathrm{IG}(\mathbf{x},y;\boldsymbol{\Theta})] &= \sum_y p(y|\mathbf{x},\mathcal{D}))\int \frac{p(y|\mathbf{x},\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})p(y|\mathbf{x},\mathcal{D})}\log\left(\frac{p(y|\mathbf{x},\boldsymbol{\theta})}{p(y|\mathbf{x},\mathcal{D})}\right)d\boldsymbol{\theta}\\
&= \frac{1}{p(\mathcal{D})}\sum\int p(y|\mathbf{x},\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})\log\left(\frac{p(y|\mathbf{x},\boldsymbol{\theta})}{p(y|\mathbf{x},\mathcal{D})}\right)d\boldsymbol{\theta}\\
&= \frac{1}{p(\mathcal{D})}\sum\int p(y|\mathbf{x},\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})\log\left(\frac{p(y|\mathbf{x},\boldsymbol{\theta})}{p(y|\mathbf{x},\mathcal{D})}\right)d\boldsymbol{\theta}\\
&= \frac{1}{p(\mathcal{D})}\sum\int p(y|\mathbf{x},\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})\left[\log(p(y|\mathbf{x},\boldsymbol{\theta}))-\log(p(y|\mathbf{x},\mathcal{D}))\right]d\boldsymbol{\theta}\\
&= \frac{1}{p(\mathcal{D})}\sum\left[\int p(y|\mathbf{x},\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})\log(p(y|\mathbf{x},\boldsymbol{\theta}))d\boldsymbol{\theta}-\int p(y|\mathbf{x},\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})\log(p(y|\mathbf{x},\mathcal{D}))d\boldsymbol{\theta}\right]\\
&= \frac{1}{p(\mathcal{D})}\int p(\boldsymbol{\theta}|\mathcal{D})\sum p(y|\mathbf{x},\boldsymbol{\theta})\log(p(y|\mathbf{x},\boldsymbol{\theta}))d\boldsymbol{\theta}-\sum\int p(y|\mathbf{x},\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})\log(p(y|\mathbf{x},\mathcal{D}))d\boldsymbol{\theta}\\
&= \frac{1}{p(\mathcal{D})}\left(\mathbb{H}[\mathbb{E}_{\boldsymbol{\theta}}[y|\mathbf{x},\mathcal{D}]]-\mathbb{E}_{\boldsymbol{\theta}}[\mathbb{H}[y|\mathbf{x},\mathcal{D}]]\right)\\
&\propto \left(\mathbb{H}[\mathbb{E}_{\boldsymbol{\theta}}[y|\mathbf{x},\mathcal{D}]]-\mathbb{E}_{\boldsymbol{\theta}}[\mathbb{H}[y|\mathbf{x},\mathcal{D}]]\right)
\end{aligned}$$

where for the last line we assume $p(\mathcal{D}) \approx p(\mathcal{D}_{\text{adv}})$ as constant values. Since we are considering adversarial instances to be obtained from the observational one, this is a very mild assumption and is completely in alignment with current research.

B. Proof of the Objective

We have

$$\begin{aligned}
|R_{\text{adv}} - R| &= \left|\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{\theta}}\left[\sup\ \mathbb{E}_{y_1\sim p(y|\mathbf{x}_{\text{adv}})}\left[\mathbb{I}\left(y_1\neq y\right)\right]-\mathbb{E}_{y_2\sim p(y|\mathbf{x})}\left[\mathbb{I}\left(y_2\neq y\right)\right]\right]\right]\right|,\\
&= \left|\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{\theta}}\left[\sup\ \mathbb{E}_{y_1\sim p(y|\mathbf{x}_{\text{adv}}),y_2\sim p(y|\mathbf{x})}\left[\mathbb{I}\left(y_1\neq y\right)-\mathbb{I}\left(y_2\neq y\right)\right]\right]\right]\right|,\\
&\leq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{\theta}}\left[\sup\ \mathbb{E}_{y_1\sim p(y|\mathbf{x}_{\text{adv}}),y_2\sim p(y|\mathbf{x})}\left[\left|\mathbb{I}\left(y_1\neq y\right)-\mathbb{I}\left(y_2\neq y\right)\right|\right]\right]\right],\\
&\leq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{\theta}}\left[\sup\ \mathbb{E}_{y_1\sim p(y|\mathbf{x}_{\text{adv}}),y_2\sim p(y|\mathbf{x})}\left[\mathbb{I}\left(y_1\neq y_2\right)\right]\right]\right].
\end{aligned}$$

where we can upper bound the expected misclassification to have:

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{\theta}}\left[1-\sum_{c=1}^{K}p(y=c\mid\mathbf{x},\boldsymbol{\theta})p(y=c\mid\mathbf{x}_{\text{adv}},\boldsymbol{\theta})\right]\right].$$

Subsequently, we use Jensen's inequality and the fact that $\mathbf{x} = \exp(\log(\mathbf{x}))$ to have:

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{\theta}}\left[1-\exp(\underbrace{\log(\sum_{c=1}^{K} p(y=c\mid\mathbf{x},\boldsymbol{\theta})p(y=c\mid\mathbf{x}_{\text{adv}},\boldsymbol{\theta}))}_{\geq\sum_c^K p(y=c|\mathbf{x},\boldsymbol{\theta})\log(p(y=c|\mathbf{x}_{\text{adv}},\boldsymbol{\theta})}\right)\right]\right].$$

For a monotonically decreasing function, we know for $x \geq y$, $f(x) \leq f(y)$. Using Jensen's inequality we have $\log(\sum_{c=1}^{K} p(y=c\mid\mathbf{x},\boldsymbol{\theta})p(y=c\mid\mathbf{x}_{\text{adv}},\boldsymbol{\theta}))\geq \sum_c^K p(y=c\mid\mathbf{x},\boldsymbol{\theta})\log(p(y=c\mid\mathbf{x}_{\text{adv}},\boldsymbol{\theta})$. Since $1-\exp(z)$ is monotonically decreasing, we have:

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{\theta}}\left[1-\exp(\log(\sum_{c=1}^{K} p(y=c\mid\mathbf{x},\boldsymbol{\theta})p(y=c\mid\mathbf{x}_{\text{adv}},\boldsymbol{\theta}))\right]\right]\\
&\leq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{\theta}}\left[1-\exp\big(\sum_{c}^{K} p(y=c\mid\mathbf{x},\boldsymbol{\theta})\log(p(y=c\mid\mathbf{x}_{\text{adv}},\boldsymbol{\theta}))\big)\right]\right]\\
&= 1-\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{E}_{\boldsymbol{\theta}}\left[\exp\big(\sum_{c}^{K} p(y=c\mid\mathbf{x},\boldsymbol{\theta})\log(p(y=c\mid\mathbf{x}_{\text{adv}},\boldsymbol{\theta}))\big)\right]\right].
\end{aligned}$$

Thus we have the following bound:

$$|R_{\text{adv}}-R| \leq 1-\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\exp\left(\underbrace{\mathbb{E}_{\boldsymbol{\theta}}\left[\sum_{c}^{K} p(y=c\mid\mathbf{x},\boldsymbol{\theta})\log(p(y=c\mid\mathbf{x}_{\text{adv}},\boldsymbol{\theta}))\right]}_{r_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}_{\text{adv}},y)}\right)\right]. \tag{9}$$

This result demonstrates that the difference between the risks is bounded by the negative cross-entropy of the predictions. While informative, this bound expresses the relation between the predictions only and not how the model performs on each set (*i.e.* given dataset versus its corresponding adversarial).

From the definition of KL-divergence, we know

$$r_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}_{\text{adv}},y) = -\mathbb{H}(p(y=c\mid\mathbf{x},\boldsymbol{\theta}),p(y=c\mid\mathbf{x}_{\text{adv}},\boldsymbol{\theta})) = -\text{KL}(p(y=c\mid\mathbf{x},\boldsymbol{\theta})\|p(y=c\mid\mathbf{x}_{\text{adv}},\boldsymbol{\theta}) - \mathbb{H}(p(y=c\mid\mathbf{x},\boldsymbol{\theta}))$$

We can add and subtract $\mathbb{H}[\mathbb{E}_{\boldsymbol{\theta}}[p(y=c\mid\mathbf{x},\boldsymbol{\theta})]]$ and $\mathbb{E}_{\boldsymbol{\theta}}[\text{IG}(\mathbf{x}_{\text{adv}},y)]$ to have

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\theta}}[r_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}_{\text{adv}},y)] &= -\mathbb{E}_{\boldsymbol{\theta}}[\text{KL}(p(y=c\mid\mathbf{x},\boldsymbol{\theta})\|p(y=c\mid\mathbf{x}_{\text{adv}},\boldsymbol{\theta}))] - \mathbb{H}[\mathbb{E}_{\boldsymbol{\theta}}[p(y=c\mid\mathbf{x},\boldsymbol{\theta})]] + \mathbb{E}_{\boldsymbol{\theta}}[\text{IG}(\mathbf{x}_{\text{adv}},y)]\\
&\quad + \underbrace{(\mathbb{H}[\mathbb{E}_{\boldsymbol{\theta}}[p(y=c\mid\mathbf{x},\boldsymbol{\theta})]] - \mathbb{E}_{\boldsymbol{\theta}}[\mathbb{H}[p(y=c\mid\mathbf{x},\boldsymbol{\theta})]])}_{\mathbb{E}_{\boldsymbol{\theta}}[\text{IG}(\mathbf{x},y)]} - \mathbb{E}_{\boldsymbol{\theta}}[\text{IG}(\mathbf{x}_{\text{adv}},y)]\\
&= -\mathbb{E}_{\boldsymbol{\theta}}[\text{KL}(p(y=c\mid\mathbf{x},\boldsymbol{\theta})\|p(y=c\mid\mathbf{x}_{\text{adv}},\boldsymbol{\theta}))] - \underbrace{(\mathbb{E}_{\boldsymbol{\theta}}[\text{IG}(\mathbf{x}_{\text{adv}},y)] - \mathbb{E}_{\boldsymbol{\theta}}[\text{IG}(\mathbf{x},y)])}_{A}\\
&\quad + \underbrace{\mathbb{E}_{\boldsymbol{\theta}}[\text{IG}(\mathbf{x}_{\text{adv}},y)] - \mathbb{H}[\mathbb{E}_{\boldsymbol{\theta}}[p(y=c\mid\mathbf{x},\boldsymbol{\theta})]]}_{B} = -\mathbb{E}_{\boldsymbol{\theta}}[\text{KL}(p(y=c\mid\mathbf{x},\boldsymbol{\theta})\|p(y=c\mid\mathbf{x}_{\text{adv}},\boldsymbol{\theta}))] - A + B.
\end{aligned}$$

We consider two cases:

i) $A = 0$, then $\mathbb{E}_{\boldsymbol{\theta}}[r_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}_{\text{adv}},y)] = -\mathbb{E}_{\boldsymbol{\theta}}[\text{KL}(p(y=c\mid\mathbf{x},\boldsymbol{\theta})\|p(y=c\mid\mathbf{x}_{\text{adv}},\boldsymbol{\theta}))] - \mathbb{E}_{\boldsymbol{\theta}}[\mathbb{H}(p(y=c\mid\mathbf{x},\boldsymbol{\theta}))] \leq -\mathbb{E}_{\boldsymbol{\theta}}[\text{KL}(p(y=c\mid\mathbf{x},\boldsymbol{\theta})\|p(y=c\mid\mathbf{x}_{\text{adv}},\boldsymbol{\theta}))]$, because $\mathbb{E}_{\boldsymbol{\theta}}[\mathbb{H}(p(y=c\mid\mathbf{x},\boldsymbol{\theta}))] \geq 0$. Therefore, in this case, $-\mathbb{E}_{\boldsymbol{\theta}}[\text{KL}(p(y=c\mid\mathbf{x},\boldsymbol{\theta})\|p(y=c\mid\mathbf{x}_{\text{adv}},\boldsymbol{\theta}))]$ is an upper bound on $\mathbb{E}_{\boldsymbol{\theta}}[r_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{x}_{\text{adv}},y)]$.

ii) $A \neq 0$, then we have $-A+B = A(-1+B/A)$. We know $A \leq |A|$ for any value, then $A(-1+B/A) \leq |A|(-1+B/A)$. Setting $(-1+B/A) = -\lambda$, we have $\lambda = (1-B/A)$. In practice, we tune λ as detailed in the paper. As such, we have, $-A+B \leq -\lambda|A|$.

Thus, putting case (i) and (ii) together, we have:

$$\mathbb{E}_{\boldsymbol{\theta}}[r_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}_{\text{adv}}, y)] \leq -\mathbb{E}_{\boldsymbol{\theta}}[\text{KL}(p(y = c \mid \mathbf{x}, \boldsymbol{\theta}) \| p(y = c \mid \mathbf{x}_{\text{adv}}, \boldsymbol{\theta}))] - \lambda |\mathbb{E}_{\boldsymbol{\theta}}[\text{IG}(\mathbf{x}, y; \boldsymbol{\Theta})] - \mathbb{E}_{\boldsymbol{\theta}}[\text{IG}(\mathbf{x}_{\text{adv}}, y; \boldsymbol{\Theta})]| \tag{10}$$

and since $1 - \exp(\cdot)$ is monotonically decreasing, we are able to achieve a tighter bound for Eq. (9) with:

$$|R_{\text{adv}} - R| \leq 1 - \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\exp\left(-\big(\mathbb{E}_{\boldsymbol{\theta}}[\text{KL}(p(y = c \mid \mathbf{x}, \boldsymbol{\theta}) \| p(y = c \mid \mathbf{x}_{\text{adv}}, \boldsymbol{\theta}))] + \lambda |\mathbb{E}_{\boldsymbol{\theta}}[\text{IG}(\mathbf{x}, y; \boldsymbol{\Theta})] - \mathbb{E}_{\boldsymbol{\theta}}[\text{IG}(\mathbf{x}_{\text{adv}}, y; \boldsymbol{\Theta})]|\big)\right)\right]. \tag{11}$$

Then the difference between the empirical risk and the adversarial risk is minimized when the upper bound (the right-hand expression of the Eq. (11)) is minimized. Hence, the main learning objectives are to:

1. Minimize $\mathbb{E}_{\boldsymbol{\theta}}[\text{KL}(p(y = c \mid \mathbf{x}, \boldsymbol{\theta}) \| p(y = c \mid \mathbf{x}_{\text{adv}}, \boldsymbol{\theta}))]$: This corresponds to matching the prediction from the adversarial data to that of the observations. Since $(\mathbf{x}, y)$ is given in training, for minimizing this KL-divergence we simply convert the minimization of the KL term to minimization of the cross-entropy loss of the adversarial examples instead.

2. Minimize $|\mathbb{E}_{\boldsymbol{\theta}}[\text{IG}(\mathbf{x}, y; \boldsymbol{\Theta})] - \mathbb{E}_{\boldsymbol{\theta}}[\text{IG}(\mathbf{x}_{\text{adv}}, y; \boldsymbol{\Theta})]|$: In addition to individual predictions, the information gained from each instance has to have a similar impact on the network in terms of how it changes the parameters.

Notably, since we know $1 - \exp(-z) \leq z$, to avoid computational instabilities and gradient saturation, we consider minimizing the upper bound without the exponential function in our implementation.

C. Hyper-Parameters

These are hyper parameters used in our experiments. For a fair comparison with previous works, all of the training, testing parameters and attack budgets are identical to those in Liu et al. (2019).

Table 5: Hyper-parameters setting in our experiments

Name	*Value*	*Notes*
T'	20	#PGD iterations in attack at test time
T	10	#PGD iterations in adversarial training
$\varepsilon_{\max}$	8/255	Max l_∞-norm in adversarial training
α	2/255	Step size for each PGD iteration
γ	0.01	Weight to control the repulsive force
λ	CIFAR-10: 5, STL-10: 20	Weight to control IG objective
n	10	#Parameter particles #Forward passes when doing ensemble inference # Expectation over Transformation

D. Experiment with Increasing Number of EoT-PGD Steps

Following standard practice and due to the cost of running increasing numbers of EoT-PGD steps, the results in the main paper use 20 steps. In this section, we conduct experiments with an increasing number of EoT-PGD steps to demnonstrate that the robustness we evaluated in the main paper is on a so-called full-strength EoT-PGD. As shown in Figure 5, the robustness is significantly decreased in the first 20 steps. However, after that, robustness is maintained, *i.e.* the EoT-PGD attack has converged and reached its full strength.

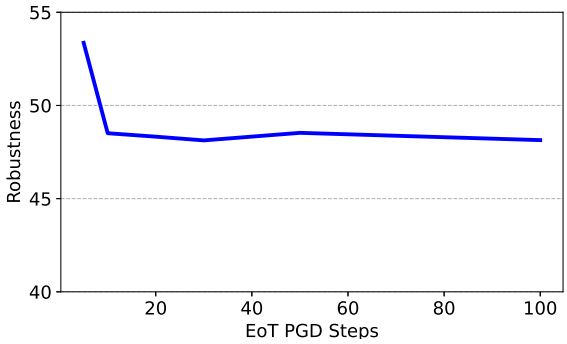

Figure 5: Robustness versus various numbers of EoT-PGD steps. EoT-PGD reaches its full strength after 20 steps. Further increasing PGD steps did not significantly improve the attack.

E. Transferability to Other Attacks

In this section, in order to extend the scope of the method and to show that our method is generic and applicable to other adversarial attacks, we conduct experiments to evaluate the robustness of networks trained on EoT-PGD ℓ_∞ against different attacks such as FGSM or ℓ_2-attack. Results in Table 6 show that our method's robustness is transferable to other attacks. The reason is that we utilized PGD in our method, and PGD is regarded as a "universal" adversary among first-order approaches, i.e. if a network is robust against PGD adversaries, it will be robust against a wide range of other attacks (Madry et al., 2018).

Table 6: Transferability. PGD ℓ_∞ trained IG-BNN robustness against different adversaries under different attack budgets.

Attacks on `CIFAR-10`	*0*	*0.015*	*0.035*	*0.055*	*0.07*
PGD ℓ_∞	83.6	75.5	50.2	26.8	16.9
FGSM	-	76.1	55.7	38.4	28.9
PGD ℓ_2	-	83.5	83.4	83.2	83.1

F. Validating Our Conjecture

Our method is built upon the conjecture: *a robust neural network quantifies the information gained from observation the same as its adversarial counterpart*. In this section, we further support this conjecture by conducting an evaluation where we assess the opposite conjecture. We make the BNN model 'inconsistent' under clean settings and adversarial settings. More specifically, instead of minimizing the Information Gain objective, we maximize to enforce the inconsistency. Figure 6 shows that this inconsistency leads to the deterioration of the network's performance. This experiment empirically validates our conjecture.

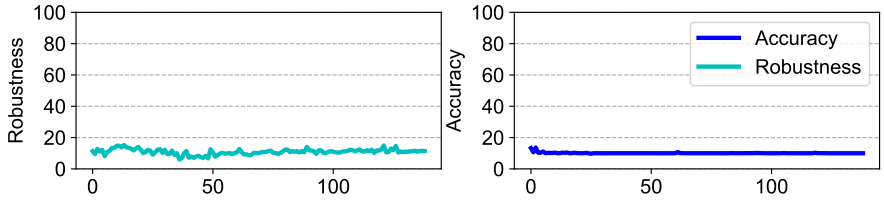

Figure 6: Accuracy and Robustness of the BNN network trained on STL-10 dataset where we enforce the model to be 'inconsistent' under clean settings and adversarial settings.

G. Details of Transfer Attacks of Adversarial Examples Among Parameter Particles

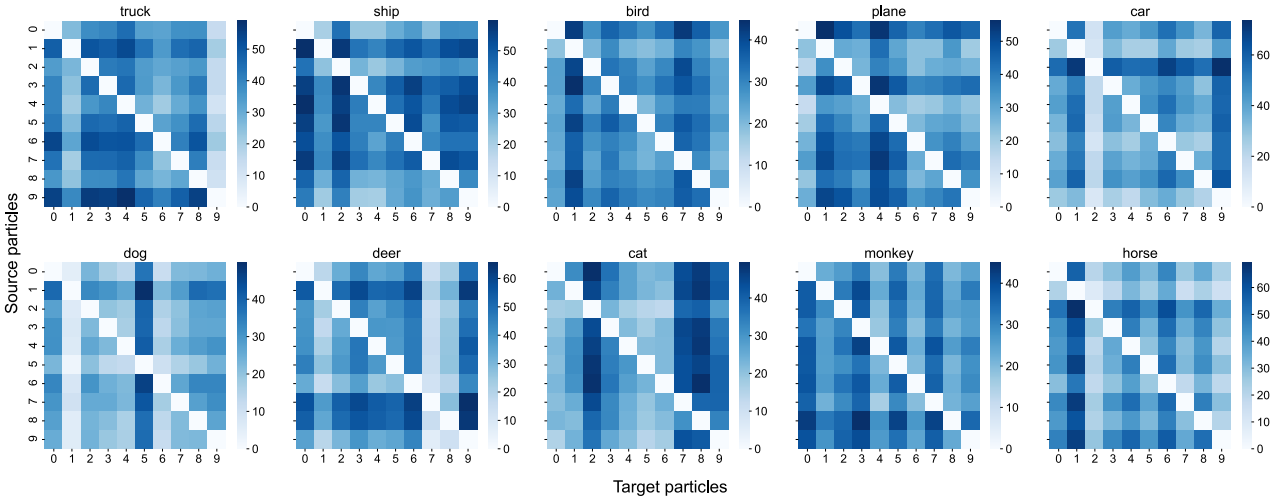

Figure 7: Transferability of adversarial examples among different particles on STL-10

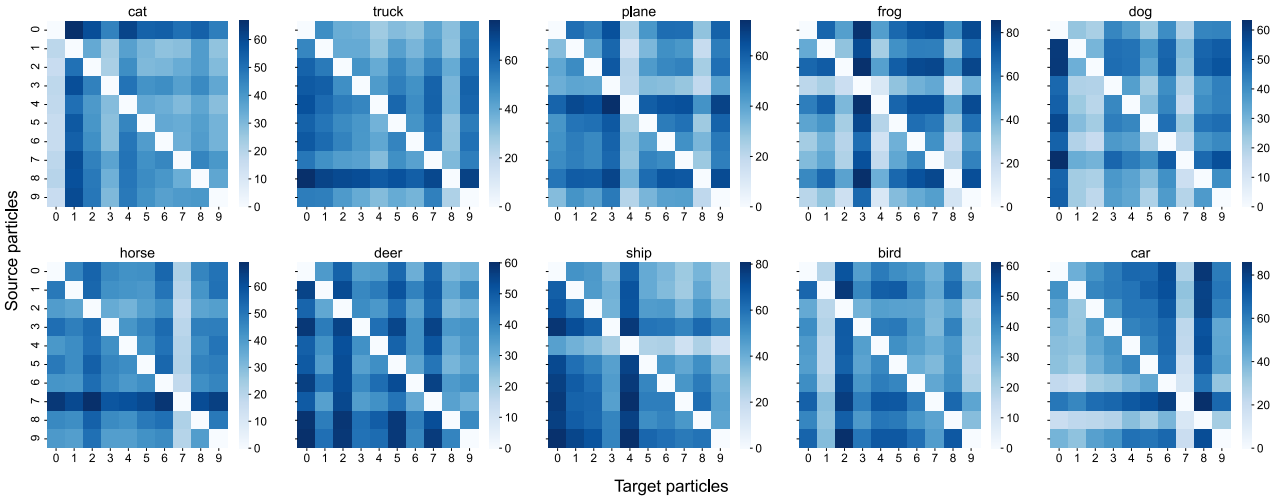

Figure 8: Transferability of adversarial examples among different particles on CIFAR-10