Weight-Covariance Alignment for Adversarially Robust Neural Networks

Panagiotis Eustratiadis [1] **Henry Gouk** [1] **Da Li** [1 2] **Timothy Hospedales** [1 2]

Abstract

Stochastic Neural Networks (SNNs) that inject noise into their hidden layers have recently been shown to achieve strong robustness against adversarial attacks. However, existing SNNs are usually heuristically motivated, and often rely on adversarial training, which is computationally costly. We propose a new SNN that achieves state-of-the-art performance without relying on adversarial training, and enjoys solid theoretical justification. Specifically, while existing SNNs inject learned or hand-tuned isotropic noise, our SNN learns an anisotropic noise distribution to optimize a learning-theoretic bound on adversarial robustness. We evaluate our method on a number of popular benchmarks, show that it can be applied to different architectures, and that it provides robustness to a variety of white-box and black-box attacks, while being simple and fast to train compared to existing alternatives.

1. Introduction

It has been shown that deep convolutional neural networks, while displaying exceptional performance in computer vision problems such as image recognition (He et al., 2016), are vulnerable to input perturbations that are imperceptible to the human eye (Szegedy et al., 2014). The perturbed input images, known as adversarial examples, can be generated by single-step (Goodfellow et al., 2015) and multi-step (Madry et al., 2018; Kurakin et al., 2017; Carlini & Wagner, 2017) updates using both gradient-based optimization methods and derivative-free approaches (Chen et al., 2017). This vulnerability raises the question of how one can go about ensuring the security of machine learning systems, thus preventing a malicious entity from exploiting instabilities (Biggio et al., 2013). In order to tackle this problem, many adversarial defense algorithms have been proposed in the literature. Among them, Stochastic Neural Networks (SNNs) that inject fixed or learnable noise into their hidden layers have shown promising results (Liu et al., 2018; 2019; He et al., 2019; Jeddi et al., 2020; Yu et al., 2021).

In this paper, we identify three limitations of the current state-of-the-art stochastic defense methods. First, most contemporary adversarial defense methods use a mixture of clean and adversarial (or even purely adversarial) samples during training, i.e., adversarial training (Goodfellow et al., 2015; Madry et al., 2018; Liu et al., 2019; Mustafa et al., 2019; He et al., 2019; Jeddi et al., 2020). However, generating strong adversarial examples during training leads to significantly higher computational cost and longer training time. Second, many existing adversarial defenses (Mustafa et al., 2019), and especially stochastic defenses (Jeddi et al., 2020) are heuristically motivated. Although they may be empirically effective against existing attacks, they lack theoretical support. Third, the noise incorporated by existing stochastic models is *isotropic* (i.e., generated from a multivariate Gaussian distribution with a diagonal covariance matrix), meaning that it perturbs the learned features of different dimensions independently. Our theoretical analysis will show that this is a strong assumption and best performance is expected from *anisotropic* noise.

We address the aforementioned limitations and propose an SNN that makes use of learnable anisotropic noise. We theoretically analyse the margin between the clean and adversarial performance of a stochastic model and derive an upper bound on the difference between these two quantities. This novel theoretical insight suggests that the anisotropic noise covariance in an SNN should be optimized to align with the classifier weights, which has the effect of tightening the bound between clean and adversarial performance. This leads to an easy-to-implement regularizer, which can be efficiently optimized on clean samples alone without need for adversarial training. We show that our method, called Weight-Covariance Alignment (WCA), can be applied to architectures of varied depth and complexity (namely, LeNet++ and ResNet-18), and achieves state-of-the-art robustness across several widely used benchmarks, including CIFAR-10, CIFAR-100, SVHN and F-MNIST. Moreover, this high level of robustness is demonstrated for

[1] University of Edinburgh [2] Samsung AI Center, Cambridge. Correspondence to: Panagiotis Eustratiadis <p.eustratiadis@ed.ac.uk>, Henry Gouk <henry.gouk@ed.ac.uk>, Da Li <dali.academic@gmail.com>, Timothy Hospedales <t.hospedales@ed.ac.uk>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

both white-box and black-box attacks. We name our proposed model WCA-Net.

The contributions of our paper are summarized as follows:

- While the majority of existing stochastic defenses are heuristically motivated, our proposed method is derived by optimizing a learning theoretic bound, providing solid justification for its robust performance.
- To the best of our knowledge, we are the first to propose a stochastic defense with learned anisotropic noise.
- WCA only requires clean samples for training, unlike most of the current state-of-the art defenses that depend on costly adversarial training.
- We demonstrate the state-of-the-art performance of our method on various benchmarks and provides resilience to both white- and black-box attacks.

2. Related Work

2.1. Adversarial Attacks

We consider the standard threat model, where the attacker can construct norm-bounded perturbations to a clean input. First-order white-box adversaries use the gradient with respect to the input image to perturb it in the direction that increases misclassification probability. The attack can also be targeted or untargeted, depending on whether a specific misclassification is required (Goodfellow et al., 2015; Kurakin et al., 2017; Madry et al., 2018; Carlini & Wagner, 2017). By default, we consider the untargeted variants of these attacks. The simplest first-order adversary is the Fast Gradient Sign Method (FGSM), proposed in Goodfellow et al. (2015). The attack adds a small perturbation to the input in the direction indicated by the sign of the gradient of the classification loss, $\mathcal{L}$, w.r.t. the input, $\vec{x}$, controlled by an attack strength ϵ,

$$\vec{x}' = \vec{x} + \epsilon \cdot \mathrm{sign}(\nabla_{\vec{x}} \mathcal{L}(h(\vec{x}), y)),$$

where h is the target model. Kurakin et al. (2017) upgraded this single-step attack to a multi-step version named Basic Iterative Method (BIM) with iterative updates and smaller step size at each update. Though BIM works effectively, Madry et al. (2018) demonstrated that randomly initializing the perturbation generated by BIM, and then making multiple attempts to construct an adversarial example results in a stronger adversarial attack known as Projected Gradient Descent (PGD). Another white-box attack of slightly different nature is the C&W attack (Carlini & Wagner, 2017), which aims to find an input perturbation δ that maximizes the following objective:

$$\begin{aligned} &\mathcal{L}(h(\vec{x} + \delta), y) - ||\delta||_p \\ &s.t. \quad \vec{x} + \delta \in [0, 1]^n, \end{aligned}$$

where p is commonly chosen from $\{0, 2, \infty\}$.

Different from the white-box attacks, black-box attacks assume the details of the targeted model are unknown, and one can only access the model through queries. Therefore, in order to attack a target model in this case, one typically trains a substitute of it (Papernot et al., 2017) and generates an attack using the queried prediction of the target model and the local substitute. Also, instead of training a substitute for the target model, zero-order optimization methods (Chen et al., 2017; Su et al., 2019) have been proposed to estimate the gradients of the target model directly. In this paper, we demonstrate that our proposed method is robust against both white- and black-box attacks.

2.2. Stochastic Adversarial Defense

Recent work has shown that SNNs yield promising performance in adversarial robustness. This can be achieved by injecting either fixed (Liu et al., 2018) or learnable (He et al., 2019; Jeddi et al., 2020; Yu et al., 2021) noise into the models.

The idea behind Random Self Ensemble (RSE) (Liu et al., 2018) is that one can simulate an ensemble of virtually infinite models while only training one. This is achieved by injecting additive spherical Gaussian noise into various layers of a network and performing multiple forward passes at test time. Though simple, it effectively improves the model robustness in comparison to a conventional deterministic model. RSE treats the variance of the injected noise as a hyperparameter that is heuristically tuned, rather than learned in conjunction with the other network parameters. In contrast, He et al. (2019) propose Parametric Noise Injection (PNI), where a fixed spherical noise distribution is controlled by a learnable "intensity" parameter, further improving model robustness. The authors show that the noise can be incorporated into different locations of a neural network, i.e., it is applicable to both feature activations and model weights. The injected noise is trained together with the model parameters via adversarial training. Learn2Perturb (L2P) (Jeddi et al., 2020) is a recent extension of PNI. Instead of learning a single spherical noise parameter, L2P learns a set of parameters defining an isotropic noise perturbation-injection module. The parameters of the perturbation-injection module and the model are updated alternatingly in a manner named "alternating back-propagation" by the authors, using adversarial training. Finally, SE-SNN (Yu et al., 2021) introduces fully-trainable stochastic layers, which are trained for adversarial robustness by adding a regularization term to the objective function that maximizes the entropy of the learned noise distribution. Unlike the other SNNs, but similarly to ours, SE-SNN only requires clean training samples.

Although conceptually related to the aforementioned stochastic defense methods, WCA-Net differs in several important aspects: WCA-Net is the first stochastic model to inject learnable *anisotropic* noise into the latent features. Our approach is derived from from optimization of a learning theoretic bound on the adversarial generalisation performance of SNNs, which motivates the use of anisotropic noise. WCA-Net does not require adversarial training and can be optimized with clean samples alone, and is therefore simpler and more efficient to train.

Another class of stochastic defenses apply noise to the input images, rather than injecting noise to intermediate activations (Pinot et al., 2019; Cohen et al., 2019; Li et al., 2019; Lee et al., 2019). From a theoretical point of view, this can be seen as "smoothing" the function implemented by the neural network in order to reduce the amount the output of the network can change when the input is changed only slightly. This type of defense can be considered a black-box defense, in the sense that it does not actually involve regularizing the weights of the network — it only modifies the input. While interesting, it has primarily been applied in scenarios where one is using a model-as-a-service framework, and cannot be sure if the model was trained with some sort of adversarial defense or not (Cohen et al., 2019).

3. Methods

Based on theoretical analysis of how the injected noise can impact generalisation performance, further expanded in Section 3.1, we propose a weight-covariance alignment loss term that encourages the weight vectors associated with the final linear classification layer to be aligned with the covariance matrix of the injected noise. Consequently, our theory leads us to use anisotropic noise, rather than the isotropic noise typically employed by previous approaches.

Our method fits into the family of SNNs that apply additive noise to the penultimate activations of the network. Consider the function, $f(\vec{x})$, which implements the feature extractor portion of the network i.e., everything except the final classification layer. Our WCA-Net architecture is defined as

$$h(\vec{x}) = W(f(\vec{x}) + \vec{z}) + \vec{b}, \quad \vec{z} \sim \mathcal{N}(0, \Sigma),$$

where W and $\vec{b}$ are the parameters of the final linear layer, $\vec{z}$ is the vector of additive noise. The objective function used to train this model is

$$\mathcal{L} = \mathcal{L}_{\text{C}} - \mathcal{L}_{\text{WCA}}, \tag{1}$$

where $\mathcal{L}_{\text{C}}$ and $\mathcal{L}_{\text{WCA}}$ represent the classification loss (e.g. softmax composed with cross entropy) and weight-covariance alignment term respectively. We describe each of our technical contributions in the remainder of this Section.

3.1. Weight-Covariance Alignment

Non-stochastic methods for defending against adversarial examples typically try to guarantee that the prediction for an input image cannot be changed. In contrast, a defense that is stochastic should aim to minimize the probability that the prediction can be changed. In this Section, we present a theoretical analysis of the probability that the prediction of an SNN will be changed by an adversarial attack. For simplicity, we restrict our analysis to the case of binary classification.

Denoting a feature extractor as f, we define an SNN h, trained for binary classification as

$$h(\vec{x}) = \vec{w}^T(f(\vec{x}) + \vec{z}) + b, \quad \vec{z} \sim \mathcal{N}(0, \Sigma),$$

where $\vec{w}$ is the weight vector of the classification layer and b is the bias. We denote the non-stochastic version of h, where the value of $\vec{z}$ is always a vector of zeros, as $\tilde{h}$. The margin of a prediction is given by

$$m_h(\vec{x}, y) = yh(\vec{x}),$$

for $y \in \{-1, 1\}$. It is positive if the prediction is correct and negative otherwise.

The quantity in which we are interested is the difference in probabilities of misclassification when the model is and is not under adversarial attack δ, which is given by

$$\begin{aligned} G^h_{p,\epsilon}(\vec{x}, y) = \max_{\vec{\delta}:\|\vec{\delta}\|_p \leq \epsilon} & P(m_h(\vec{x} + \delta, y) \leq 0) \\ & -P(m_h(\vec{x}, y) \leq 0). \end{aligned} \tag{2}$$

Our main theoretical result, given below, shows how one can take an adversarial robustness bound, $\Delta^{\tilde{h}}_p(\vec{x}, \epsilon)$, for the deterministic version of a network, and transform it to a bound on G for the stochastic version of the network.

Theorem 1. *The quantity $G^h_{p,\epsilon}(\vec{x}, y)$, as defined above, is bounded as*

$$G^h_{p,\epsilon}(\vec{x}, y) \leq \frac{\Delta^{\tilde{h}}_p(\vec{x}, \epsilon)}{\sqrt{2\pi\vec{w}^T\Sigma\vec{w}}},$$

where the robustness of the deterministic version of h is known to be bounded as $|\tilde{h}(\vec{x}) - \tilde{h}(\vec{x} + \vec{\delta})| \leq \Delta^{\tilde{h}}_p(\vec{x}, \epsilon)$ for any $\|\vec{\delta}\|_p \leq \epsilon$.

The proof is provided in the supplemental material. We can see from Theorem 1 that increasing the bi-linear form, $\vec{w}^T\Sigma\vec{w}$, of the noise distribution covariance and the classifier reduces the gap between clean and robust performance. As such, we define the loss term,

$$\mathcal{L}_{\text{WCA}} = \sum_{i=1}^{C} \ln(\vec{w}_i^T\Sigma\vec{w}_i), \tag{3}$$

where C is the number of classes in the classification problem, and $\vec{w}_i$ is the weight vector of the final layer that is associated with class i. We found that including the logarithm results in balanced growth rates between the $\mathcal{L}_C$ and $\mathcal{L}_{\text{WCA}}$ terms in Eq. 1 as training progresses, hence improving the reliability of training loss convergence.

The key insight of Theorem 1, operationalized by Eq. 3, is that the noise and weights should co-adapt to align the noise and weight directions. We call this loss Weight-Covariance Alignment (WCA) because it is maximized when each $\vec{w}_i$ is well-aligned with the eigenvectors of the covariance matrix.

This WCA loss term runs into the risk of maximizing the magnitude of $\vec{w}$, rather than encouraging alignment or increasing the scale of the noise. To avoid the uncontrollable scaling of network parameters, it is common practice to penalize large weights by means of ℓ^2 regularization:

$$\mathcal{L} = \mathcal{L}_{\mathcal{C}} - \mathcal{L}_{WCA} + \lambda \vec{w}^T \vec{w},$$

where λ controls the strength of the penalty. In our case, we apply the ℓ^2 penalty when updating the parameters of the classification layer and the covariance matrix. Another approach to limiting parameter magnitude would be to enforce norm constraints on $\vec{w}$ and Σ, e.g., using a projected subgradient method at each update. We provide more details of this alternative in the supplementary material. Empirically, we found that the penalty-based approach outperformed the constraint-based approach, so we focus on the former by default.

3.2. Injecting Anisotropic Noise

In contrast to previous work that only considers injecting isotropic Gaussian noise (Liu et al., 2019; He et al., 2019; Jeddi et al., 2020; Yu et al., 2021), we make use of anisotropic noise, providing a richer noise distribution than previous approaches. Crucially, it also means that the principal directions in which the noise is generated no longer have to be axis-aligned. I.e., prior work suffers from the inability to simultaneously optimise alignment between noise and weights (required to minimise the adversarial gap bounded by Theorem 1), and freedom to place weight vectors off the axis (required for good clean performance). Our use of anisotropic noise in combination with WCA encourages alignment of the weight vectors with the covariance matrix eigenvectors, while allowing non-axis aligned weights, thus providing more freedom about where to place the classification decision boundaries.

Previous approaches are able to train the variance of each dimension of the isotropic noise via the use of the “reparameterization trick” (Kingma & Welling, 2014), where one samples noise from a distribution with zero mean and unit variance, then rescales the samples to get the desired variance. Because the rescaling process is differentiable, this allows one to learn variance jointly with the other network parameters with backpropagation. In order to sample anisotropic noise, one can instead sample a vector of zero mean unit variance and multiply this vector by a lower triangular matrix, L. This lower triangular matrix is related to the covariance matrix as

$$\Sigma = L \cdot L^T.$$

This guarantees that the covariance matrix remains positive semi-definite after each gradient update.

4. Experiments

In this Section we present the experiments that demonstrate the efficacy of our model and verify our theoretical analysis.

4.1. Experimental Setup

Datasets For comparison against the current state-of-the-art and for our ablation study we use four benchmarks: CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), SVHN (Netzer et al., 2011) and Fashion-MNIST (Xiao et al., 2017). CIFAR-10 and CIFAR-100 contain 60K 32x32 color images, 50K for training and 10K for testing, evenly spread across 10 and 100 classes respectively. SVHN can be considered a more challenging version of MNIST (LeCun et al., 2010); it contains almost 100K 32x32 color images of digits (0-9) collected from Google’s Street View imagery, with roughly 73K for training and 26K for testing. Fashion-MNIST is a collection of 70K 28x28 grayscale images of clothing, 60K for training and 10K for testing, also spread across 10 classes.

Models For all benchmarks except F-MNIST we use a ResNet-18 (He et al., 2016) backbone, while for F-MNIST, being a relatively simpler dataset, we use LeNet++ (Wen et al., 2016). After the backbone we add a penultimate layer for dimensionality reduction; this enables us to always train a reasonably-sized covariance matrix regardless of the original dimensionality of the feature extractor[1]. The only restriction for the dimensionality of the penultimate layer is that it needs to be a number greater or equal to the number of classes in the task, so as to allow the covariance matrix to align with at least one classifier vector. The two hyperparameters of note across all of our experiments are the learning rate and ℓ^2 penalty (i.e., weight decay), the exact values of which are provided in the supplementary material.

4.1.1. Attacks

We evaluate our method using three white-box adversaries: FGSM (Goodfellow et al., 2015), PGD (Madry et al., 2018) and C&W (Carlini & Wagner, 2017), and one black-box

[1] 32x32 for the benchmarks with 10 classes, 256x256 for the benchmarks with 100 classes.

attack: the One-Pixel attack (Su et al., 2019).

We parameterize the attacks following the literature (He et al., 2019; Jeddi et al., 2020). More specifically, FGSM and PGD are set with an attack strength of $\epsilon = 8/255$ for CIFAR-10, CIFAR-100 and SVHN, and $\epsilon = 0.3$ for F-MNIST. PGD has a step size of $\alpha = \epsilon/10$ and number of steps $k = 10$ for all benchmarks as per He et al. (2019). C&W has a learning rate of $\alpha = 5 \cdot 10^{-4}$, number of iterations $k = 1000$, initial constant $c = 10^{-3}$ and maximum binary steps $b_{\max} = 9$ same as Jeddi et al. (2020).

For the parameters of the One-Pixel attack we tried to replicate the experimental setup described in the supplementary material of Jeddi et al. (2020) for attack strengths of 1, 2 and 3 pixels. We followed their setup with population size $\mathrm{N} = 400$ and maximum number of iterations $\mathrm{k}_{\max} = 75$. However, we noticed that the more pixels we added to our attack the weaker the attack became, which is counterintuitive. We attribute that to the small number of iterations; every added pixel substantially increases the search space of the differential evolution algorithm, and 75 iterations are no longer enough to converge when the number of pixels is 2 and 3. Therefore we maintain a population size of $\mathrm{N} = 400$, but increase the number of iterations to $\mathrm{k}_{\max} = 1000$. For reproducibility purposes, we further clarify that for the differential evolution algorithm we use a crossover probability of $\mathrm{r} = 0.7$, a mutation constant of $\mathrm{m} = 0.5$, and the following criterion for convergence:

$$\sqrt{\mathrm{Var}(\mathcal{E}(X))} \leq \left| \frac{1}{100N} \sum_{x \in X} \mathcal{E}(x) \right|,$$

where X denotes the population, $\mathcal{E}(X)$ the energy of the population and $\mathcal{E}(x)$ the energy of a single sample.

Expectation over Transformation Due to the noise injected by SNNs, the gradients used by white-box attacks are stochastic (Athalye et al., 2018). As a result, the true gradients cannot be correctly estimated for attacks that use only one sample to compute the perturbation. To avoid this issue, we apply Expectation over Transformation (EoT) following Athalye et al. (2018). When generating an attack, we compute gradients of multiple forward passes using Monte-Carlo sampling and perturb the inputs using the averaged gradient at each update. We empirically found that a reliable number of MC samples is 50 (as we observed performance begins to saturate from around 35 and converges at 40); thus, we use 50 across all experiments.

4.2. Comparison to Prior Stochastic Defenses

Competitors We compare the performance of WCA-Net to three recent state-of-the-art stochastic defenses to verify its efficacy. **AdvBNN** (Liu et al., 2019): adversarially trains a Bayesian neural network for defense. **PNI** (He et al., 2019): learns an "intensity" parameter to control the variance of their SNN. **Learn2Perturb (L2P)** (Jeddi et al., 2020): improves PNI by learning an isotropic perturbation injection module. Furthermore, there are partial comparisons against **SE-SNN** (Yu et al., 2021) and **IAAT** (Xie et al., 2019). All experiments use a ResNet-18 backbone and are conducted on CIFAR-10 for fair comparison.

Table 1. Comparison of state-of-the-art SNNs for FGSM and PGD attacks on CIFAR-10 and CIFAR-100 with a ResNet-18 backbone. Performance of Adv-BNN, PNI and L2P extracted from Jeddi et al. (2020).

	CIFAR-10			CIFAR-100		
Method	Clean	FGSM	PGD	Clean	FGSM	PGD
Adv-BNN	82.2	60.0	53.6	$\sim$ 58.0	$\sim$ 30.0	$\sim$ 27.0
PNI	87.2	58.1	49.4	$\sim$ 61.0	$\sim$ 27.0	$\sim$ 22.0
L2P	85.3	62.4	56.1	$\sim$ 50.0	$\sim$ 30.0	$\sim$ 26.0
SE-SNN	92.3	74.3	-	-	-	-
IAAT	-	-	-	63.9	-	18.5
WCA-Net	**93.2**	**77.6**	**71.4**	**70.1**	**51.5**	**42.7**

Table 2. Comparison of state-of-the-art SNNs for white box C&W attack and black box n-Pixel attack on CIFAR-10 with a ResNet-18 backbone. Performance of competing methods extracted from Jeddi et al. (2020).

	Attack Strength	Adv-BNN	PNI	L2P	WCA-Net
	Clean	82.2	87.2	85.3	**93.2**
C&W	$\kappa = 0.1$	78.1	66.1	84.0	**89.4**
	$\kappa = 1$	65.1	34.0	76.4	**78.4**
	$\kappa = 2$	49.1	16.0	66.5	**71.9**
	$\kappa = 5$	16.0	0.08	34.8	**55.0**
n-Pixel	1 pixel	68.6	50.9	64.5	**90.8**
	2 pixels	64.6	39.0	60.1	**85.5**
	3 pixels	59.7	35.4	53.9	**81.2**
	5 pixels	-	-	-	64.3

4.2.1. White-box Attacks

We first compare our proposed WCA-Net to the existing state-of-the-art methods in the white-box attack setting. From the results in Table 1, we can see that our WCA-Net shows noticeable improvement of $\sim 15\%$ over the strongest competitor, L2P. Moreover, we find that our method does not sacrifice its performance on clean data to afford such strong robustness.

An important aspect of WCA that needs to be assessed is its potential to scale with the number of classes. For this reason we conduct experiments on CIFAR-100, comparing against our previously mentioned competitors, plus IAAT (Xie et al., 2019), all of which use a ResNet-18 backbone in their architectures. From Table 1 we can see that the adversarial robustness of WCA-Net outperforms the other methods.

Table 3. Comparison of WCA-Net to recent state-of-the-art, both stochastic and non-stochastic, on CIFAR-10. All competitors evaluate their models on the untargeted PGD attack, with attack strength $\epsilon = 8/255$, and number of iterations $k \in \{7, 10, 20\}$. Some results are extracted from He et al. (2019). AT: Use of adversarial training.

Defense	Architecture	AT	Clean	PGD
RSE (Liu et al., 2018)	ResNext	✗	87.5	40.0
DP (Lécuyer et al., 2019)	28-10 Wide ResNet	✗	87.0	25.0
TRADES (Zhang et al., 2019)	ResNet-18	✓	84.9	56.6
PCL (Mustafa et al., 2019)	ResNet-110	✓	91.9	46.7
PNI (He et al., 2019)	ResNet-20 (4x)	✓	87.7	49.1
Adv-BNN (Liu et al., 2019)	VGG-16	✓	77.2	54.6
L2P (Jeddi et al., 2020)	ResNet-18	✓	85.3	56.3
MART (Wang et al., 2020)	ResNet-18	✓	83.0	55.5
BPFC (Addepalli et al., 2020)	ResNet-18	✗	82.4	41.7
RLFLAT (Song et al., 2020)	32-10 Wide ResNet	✓	82.7	58.7
MI (Pang et al., 2020)	ResNet-50	✗	84.2	64.5
SADS (S. & Babu, 2020)	28-10 Wide ResNet	✓	82.0	45.6
WCA-Net	ResNet-18	✗	**93.2**	**71.4**

We also present the evaluation of our method against the C&W attack in Table 2. Here, the confidence level κ indicates the attack strength. Our WCA-Net achieves the best performance, with the accuracy degrading gracefully as the confidence increases.

4.2.2. Black-box Attacks

To further verify the robustness of our WCA-Net, we conduct experiments on a black-box attack, the One-Pixel attack (Su et al., 2019). This attack is derivative-free and relies on evolutionary optimization, and its attack strength is controlled by the number of pixels it compromises. We follow Jeddi et al. (2020) and consider pixel numbers in $\{1, 2, 3\}$. Additionally, we report results for a stronger 5-pixel attack. From Table 2, we can see that our method demonstrates the strongest robustness in all cases, showing $\sim 13\%$ to $\sim 22\%$ improvement over the best competitor Adv-BNN. Importantly, these results show that the robustness of our method does not rely on stochastic gradients.

4.2.3. Stronger Attacks

In addition, we evaluate WCA-Net against two stronger attacks that are, in general, common among recent adversarial robustness literature, but are not mentioned in the stochastic defenses we outline as direct competitors. These are: (i) PGD$_{100}$; a stronger variant of PGD with 100 random restarts and (ii) the Square Attack (Andriushchenko et al., 2020); a black-box attack that compromises the attacked image in small localized square-shaped updates. We present the results of our evaluation in Table 4.

Table 4. Evaluation of WCA-Net with a ResNet-18 backbone on CIFAR-10, against the white-box PGD$_{100}$ and black-box Square Attack, for different values of attack strength ϵ.

	$\epsilon/255$	Clean	1	2	4	8	16	32	64	128
PGD$_{100}$	No Def.	93.3	45.3	14.6	0	0	0	0	0	0
	WCA	93.2	73.2	72.2	72.1	71.2	69.7	56.4	28.2	10.5
Square	No Def.	93.3	32.9	31.7	12.4	6.0	1.2	0	0	0
	WCA	93.2	51.7	51.7	50.4	49.0	48.8	44.3	36.9	28.6

4.3. Comparison to State of the Art

Direct comparison to a wider range of competitors is difficult due to the variety of backbones and settings used. Nevertheless, Table 3 provides comparison to recent state of the art stochastic and non-stochastic defenses. We can see that WCA-Net achieves excellent performance including comparing to methods that use bigger backbones and make the stronger assumption of adversarial training.

4.4. Further Analysis

Ablation Study We perform an ablation study on four benchmarks, CIFAR-10, CIFAR-100, SVHN and F-MNIST, to investigate the contribution of anisotropic noise, as shown in Table 5. For each benchmark, we evaluate a "clean" baseline architecture, consisting only of the backbone and the classification layer. We then evaluate a variant of WCA-Net with isotropic, and one with anisotropic noise. We observe that our anisotropic noise provides consistent benefit to adversarial robustness.

Another important observation is that there is no trade-off between the robust and clean performance of our models; both the isotropic and anisotropic variants of WCA-Net maintain the clean performance of the baseline defenseless model.

All the FGSM and PGD attacks in Table 5 use attack strength $\epsilon = 8/255$. For completeness, we report the performance of all the variants above against FGSM and PGD with various attack strengths $\epsilon = 2^n$, $n \in \{0...7\}$ on CIFAR-10 shown in Figure 1. From these results, we can see the overall trend here is consistent with the observations in Table 5. Also, we can see that the performance of our variants degrades more gracefully than the defenseless baseline.

Large-scale, high-resolution We are further interested to show that our WCA-Net can handle high-resolution images and more challenging datasets. For that purpose we evaluate our method on two additional benchmarks: (i) Imagenette[2], a subset of ImageNet with 10 classes and full-resolution images, and (ii) mini-ImageNet (Vinyals et al., 2016), a large subset of ImageNet with 100 classes and 84x84 images, designed to be more challenging than CIFAR-100. The

[2] https://github.com/fastai/imagenette

Table 5. Ablation study for FGSM and PGD attacks on CIFAR-10, CIFAR-100, SVHN and F-MNIST. For CIFAR-10, CIFAR-100 and SVHN we use a ResNet-18, and for F-MNIST a LeNet++ backbone.

	CIFAR-10			CIFAR-100			SVHN			F-MNIST		
Model	Clean	FGSM	PGD	Clean	FGSM	PGD	Clean	FGSM	PGD	Clean	FGSM	PGD
No Defense	93.3	14.9	3.9	72.2	12.3	1.2	93.4	55.6	23.5	90.8	26.4	12.0
WCA-Net Isotropic	93.1	60.7	55.9	70.1	27.5	21.8	93.4	45.0	40.1	90.1	63.5	37.2
WCA-Net Anisotropic	93.2	77.6	71.4	70.1	51.5	42.7	93.4	87.6	85.7	90.1	65.2	48.5

Table 6. Control experiments on CIFAR-10 for further analysis. See Sec. 4.4. AT: Training purely with adversarial examples. CT+AT: Training with a mix of clean and adversarial examples.

Experiment	Clean	FGSM	PGD
No Defense	93.3	14.9	3.9
WCA-Net (Penalty regularizer)	93.2	77.6	71.4
WCA-Net (Constraint regularizer)	92.2	62.9	53.2
E1: Test without EoT	93.2	82.9	75.1
E2: Average multiple noise samples	93.2	70.3	68.8
E3: Noise trained independently	93.1	45.0	41.6
WCA-Net: AT	88.1	75.4	70.4
WCA-Net: CT+AT	90.0	75.6	70.7

Table 7. Comparison between the undefended ResNet-18 baseline and WCA-Net with a ResNet-18 backbone for Imagenette (high-res, 10 categories) and mini-ImageNet (large-scale, 100 categories) under PGD attack.

	Imagenette			mini-ImageNet		
Model	Clean	FGSM	PGD	Clean	FGSM	PGD
No Defense	75.5	8.4	0	51.9	5.0	0
WCA-Net	74.2	59.3	48.7	51.3	41.6	30.4

results presented in Table 7 demonstrate that our method generalizes quite well to both high-resolution images as well as more challenging datasets.

Norm-constrained architecture As explained in Section 3.1, we control the magnitude of the weights in our architecture by means of ℓ^2 regularization. Another option to achieve the same effect is to apply norm constraints to the classification vectors $\vec{w}_i$ and covariance matrix Σ. A detailed explanation of how we apply these norm constraints is given in the supplementary material. In Table 6 we report results of a WCA-Net variant with a norm-constrained regularizer. Constraint-based regularization still provides good robustness, but is weaker than the ℓ^2 penalty-based variant.

E1: Importance of EoT To show the impact of EoT, we also evaluate the test performance without it. Table 6 shows that the test performance increases without using EoT. This makes sense as critiqued in Athalye et al. (2018); one gradient sample is not enough to construct an effective attack.

E2: Average multiple noise samples at test time Our model's forward pass performs the following: (i) Extract features from the penultimate layer of the backbone, (ii) inject additive noise, and (iii) compute the logits. By default we draw a single noise sample as suggested by our theory. In this experiment, we sample from the distribution multiple times and average the final logits. The more noise samples we average, the more we expect the additive noise to lose its regularization effect. The experimental results in Table 6 confirm that using more ($n = 10$) samples degrades performance.

E3: Train noise and model independently In this experiment, we first train the model without injecting any noise. Then, keeping the model parameters frozen we train the noise independently. In Table 6 we can see that this variant achieves an elementary level of robustness that is better than the defenseless baseline shown in Table 5, however, not as strong as the isotropic baseline. As mentioned in Section 3.1, a key insight of Theorem 1 is that the noise and weights should co-adapt. As expected, keeping the weight vectors $\vec{w}_i$ frozen, overall limits the ways the WCA term (see Eq. 3) can inflate, thus never realizing its full potential.

Adversarial training Our proposed method only requires clean data for training. To show this, we adversarially train our anisotropic WCA-Net in two settings: (i) purely with adversarial examples and (ii) with a mix of clean and adversarial examples. We train with a PGD attack with $\epsilon = 8/255$ and $k = 10$. From the results in Table 6, we can see that incorporating adversarial training harms our performance on clean data as expected (Goodfellow et al., 2015); while providing no consistent benefit for adversarial defense.

4.5. Inspection of Gradient Obfuscation

Athalye et al. (2018) proposed a set of criteria to inspect whether a stochastic defense method relies on obfuscated gradients. Following He et al. (2019), we summarize these criteria as a checklist. If any item in this checklist holds true, the stochastic defense is deemed unreliable. The following analysis verifies that our model's strong robustness is not

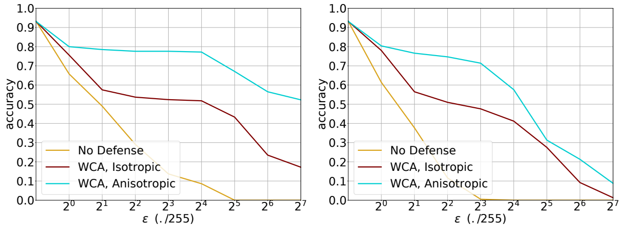

Figure 1. Evaluation of our model variants (see Table 5) for different attack strengths $\epsilon = 2^n,\ n \in \{0...7\}$, specifically for the FGSM (left) and PGD (right) attacks on CIFAR-10. Best viewed in color.

caused by gradient obfuscation.

Criterion 1: One-step attacks perform better than iterative attacks.

Refutation: Knowing that PGD is an iterative variant of FGSM, we use our existing evaluation to refute this criterion. From the results in Tables 1, 5 and 6, we can see that our WCA-Net performs consistently better against FGSM than against PGD.

Criterion 2: Black-box attacks perform better than white-box attacks.

Refutation: From Tables 1 and 2 we observe that FGSM and PGD outperform the 1-pixel attack. In Figure 1 we see the effect of increasing the attack strength on both white-box attacks, and they still outperform the stronger 2-, 3- and 5-pixel attacks.

Criterion 3: Unbounded attacks do not reach 100% success.

Refutation: For fair comparison to previous work, FGSM and PGD in this paper are parameterized following He et al. (2019). However, for this check we deliberately increase the attack strength of PGD to $\epsilon = 255/255$ and number of iterations to $k = 20$. We evaluate all of our models against this attack, and they achieve an accuracy of 0%.

Criterion 4: Random sampling finds adversarial examples.

Refutation: To assess this, we hand-pick 100 CIFAR-10 test images that our model successfully classifies during standard testing (100% accuracy), but misclassifies under FGSM with $\epsilon = 8/255$ (0% accuracy). For each of these test images, we randomly sample 1,000 perturbed images within the same ϵ-ball, and replace the original image if any of the samples result in misclassification. We then evaluate our model on these 100 images to get a performance of 98%.

Criterion 5: Increasing the distortion bound doesn't increase success.

Refutation: Figure 1 shows that increasing the distortion bound increases the attack's success.

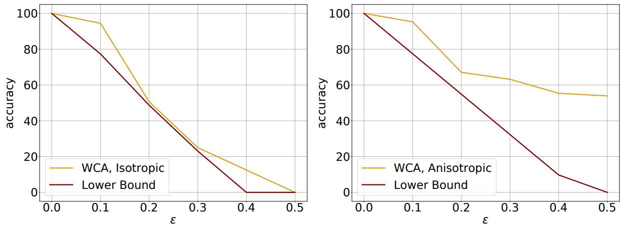

Figure 2. Evaluating our bound. Plots of the test set accuracy of SVMs trained on the zero and one digits found in MNIST. We report the performance of models trained with isotropic (left) and anisotropic (right) noise, and the worst-case performance according to Theorem 1. The anisotropic model provides a more robust bound than the isotropic model as well as better empirical performance. Best viewed in color.

4.6. Empirical Evaluation of Theorem 1

To evaluate the tightness of our bound presented in Theorem 1, we train linear Support Vector Machines (SVM) on the zero and one digits found in the MNIST dataset. Using a linear model allows us to compute the numerator using the technique of Gouk & Hospedales (2020),

$$\Delta_{\infty}^{\tilde{h}}(\vec{x}, \epsilon) = \epsilon \|\vec{w}\|_1,$$

where $\vec{w}$ is the weight vector of the SVM. We use principal components analysis to reduce the images to 32 dimensions, and apply learned isotropic and anisotropic noise to these reduced features before classification with the SVM. The covariance matrix and SVM weights are found by minimizing the hinge loss plus the WCA loss term using gradient descent. Results of attacking these models with PGD, and the lower bound on performance as computed by Theorem 1, are given in Figure 2. From these plots we can see: (i) the bound is not violated at any point, corroborating our analysis; (ii) as the strength of the adversarial attack is increased, the bound remains non-vacuous for reasonable (i.e., likely imperceptible) values of the attack strength; and (iii) the model with anisotropic noise is more robust than the model with isotropic noise. This last finding is particularly interesting because in the linear model regime PGD attacks are able to find globally optimal adversarial examples.

4.7. Empirical Observations about WCA

Figure 3 shows the effect of our regularization methods with a bivariate Gaussian, by plotting the contours of the distribution against the weight vectors of the classification layer. These visualizations are obtained by training our WCA-Net variants with a LeNet++ backbone on F-MNIST, with a 2-dimensional bottleneck and 2x2 covariance matrix.

We show the following: (i) First, in the left of Figure 3, we can see that the learned noise is axis-aligned since the injected noise is isotropic. Further, we can see that the

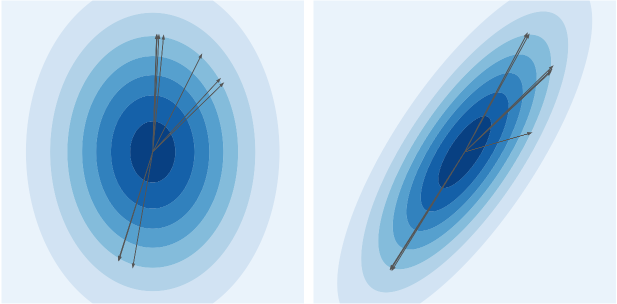

Figure 3. Visualisation of our models on F-MNIST with a 2D bottleneck. Contours and arrows indicate noise covariance Σ and weights $\vec{w}_i$. Left: WCA-Net with isotropic noise. Right: WCA-Net with anisotropic noise. Evidently, our WCA-Net with anisotropic noise allows covariance to be aligned with off-axis weights.

weight vectors are near-axis-aligned, as WCA pushes them to align with the learned noise. (ii) Then, in the right Figure, due to the combination of anisotropic noise and WCA, our model has weight-aligned noise, and the weights are free to be non-axis-aligned. Overall, we observe better alignment between the learned weight vectors and the eigenvectors of the covariance matrix in our proposed anisotropic WCA-Net.

5. Conclusions

In this paper we contribute the first stochastic model for adversarial defense that features fully-trained, anisotropic Gaussian noise, is hyperparameter free, and does not rely on adversarial training. We provide both theoretical support for the core ideas behind it, and experimental evidence of its excelling performance. We extensively evaluate WCA-Net on a variety of white-box and black-box attacks, and further show that its high performance is not a result of stochastic (obfuscated) gradients. Thus, we consider the proposed model to push the boundary of adversarial robustness.

6. Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/S000631/1; and the MOD University Defence Research Collaboration (UDRC) in Signal Processing.

References

Addepalli, S., S., V. B., Baburaj, A., Sriramanan, G., and Babu, R. V. Towards achieving adversarial robustness by enforcing feature consistency across bit planes. In *CVPR*, 2020.

Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: A query-efficient black-box adversarial attack via random search. In *ECCV*, 2020.

Athalye, A., Carlini, N., and Wagner, D. A. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

Biggio, B., Fumera, G., and Roli, F. Security evaluation of pattern classifiers under attack. *IEEE transactions on knowledge and data engineering*, 26(4):984–996, 2013.

Carlini, N. and Wagner, D. A. Towards evaluating the robustness of neural networks. In *SP*, 2017.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., and Hsieh, C.-J. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *ACM*, 2017.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320, 2019.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Gouk, H. and Hospedales, T. M. Optimising network architectures for provable adversarial robustness. In *SSPD*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

He, Z., Rakin, A. S., and Fan, D. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *CVPR*, 2019.

Jeddi, A., Shafiee, M. J., Karg, M., Scharfenberger, C., and Wong, A. Learn2perturb: An end-to-end feature perturbation learning to improve adversarial robustness. In *CVPR*, 2020.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *ICLR*, 2014.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *Toronto.edu [Online]. Available: https://www.cs.toronto.edu/ kriz*, 2009.

Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial examples in the physical world. In *ICLR*, 2017.

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2010.

Lécuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. Certified robustness to adversarial examples with differential privacy. In *SP*, 2019.

Lee, G.-H., Yuan, Y., Chang, S., and Jaakkola, T. Tight certificates of adversarial robustness for randomly smoothed classifiers. In *Advances in Neural Information Processing Systems*, pp. 4910–4921, 2019.

Li, B., Chen, C., Wang, W., and Carin, L. Certified adversarial robustness with additive noise. In *Advances in Neural Information Processing Systems*, pp. 9464–9474, 2019.

Liu, X., Cheng, M., Zhang, H., and Hsieh, C. Towards robust neural networks via random self-ensemble. In *ECCV*, 2018.

Liu, X., Li, Y., Wu, C., and Hsieh, C. Adv-bnn: Improved adversarial defense through robust bayesian neural network. In *ICLR*, 2019.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Mustafa, A., Khan, S. H., Hayat, M., Goecke, R., Shen, J., and Shao, L. Adversarial defense by restricting the hidden space of deep neural networks. In *ICCV*, 2019.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. *Stanford.edu [Online]. Available: http://ufldl.stanford.edu/housenumbers*, 2011.

Pang, T., Xu, K., and Zhu, J. Mixup inference: Better exploiting mixup to defend adversarial attacks. In *ICLR*, 2020.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *ACM*, 2017.

Pinot, R., Meunier, L., Araujo, A., Kashima, H., Yger, F., Gouy-Pailler, C., and Atif, J. Theoretical evidence for adversarial robustness through randomization. In *Advances in Neural Information Processing Systems*, pp. 11838–11848, 2019.

S., V. B. and Babu, R. V. Single-step adversarial training with dropout scheduling. In *CVPR*, 2020.

Song, C., He, K., Lin, J., Wang, L., and Hopcroft, J. E. Robust local features for improving the generalization of adversarial training. In *ICLR*, 2020.

Su, J., Vargas, D. V., and Sakurai, K. One pixel attack for fooling deep neural networks. *IEEE Trans. Evol. Comput.*, 2019.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In *ICLR*, 2014.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., and Wierstra, D. Matching networks for one shot learning. In *NeurIPS*, 2016.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020.

Wen, Y., Zhang, K., Li, Z., and Qiao, Y. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

Xie, C., Wu, Y., van der Maaten, L., Yuille, A. L., and He, K. Feature denoising for improving adversarial robustness. In *CVPR*, 2019.

Yu, T., Yang, Y., Li, D., Hospedales, T., and Xiang, T. Simple and effective stochastic neural networks. In *AAAI*, 2021.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.