Approximate Manifold Defense Against Multiple Adversarial Perturbations

Jay Nandy
School of Computing
National University of Singapore
jaynandy@comp.nus.edu.sg

Wynne Hsu
School of Computing
National University of Singapore
whsu@comp.nus.edu.sg

Mong Li Lee
School of Computing
National University of Singapore
leeml@comp.nus.edu.sg

Abstract—**Existing defenses against adversarial attacks are typically tailored to a specific perturbation type. Using adversarial training to defend against multiple types of perturbation requires expensive adversarial examples from different perturbation types at each training step. In contrast, manifold-based defense incorporates a generative network to project an input sample onto the clean data manifold. This approach eliminates the need to generate expensive adversarial examples while achieving robustness against multiple perturbation types. However, the success of this approach relies on whether the generative network can capture the complete clean data manifold, which remains an open problem for complex input domain. In this work, we devise an approximate manifold defense mechanism, called RBF-CNN, for image classification. Instead of capturing the complete data manifold, we use an RBF layer to learn the density of small image patches. RBF-CNN also utilizes a reconstruction layer that mitigates any minor adversarial perturbations. Further, incorporating our proposed reconstruction process for training improves the adversarial robustness of our RBF-CNN models. Experiment results on MNIST and CIFAR-10 datasets indicate that RBF-CNN offers robustness for multiple perturbations without the need for expensive adversarial training.**

Index Terms—**Deep Learning, Adversarial attack, Robustness, Image classification, RBF filter, EM algorithm.**

I. INTRODUCTION

Despite the impeccable success of deep neural network (DNN)-based models in various applications, there is a growing awareness of their vulnerability against adversarial attacks [1], [2]. An adversary deliberately introduces minor perturbations that mislead the networks to produce wrong predictions for the perceptually identical inputs. The adversarial vulnerability of DNN models has led to concern about the safety and reliability of these models for real-world applications [3]–[5].

Several methods have been proposed to improve the robustness of DNN models against adversarial attacks. Two of the most successful defense frameworks against adversarial attacks are adversarial training and randomized smoothing. The *adversarial training* mechanism trains a model using adversarial examples of a specific ℓ_p perturbation type to achieve robustness for that perturbation type [6], [7]. This training process is expensive as it requires the generation of adversarial examples at each training iteration. Tramèr and Boneh (2019) [8] demonstrates that adversarial training can achieve robustness for multiple perturbation types only by incorporating different types of adversaries for training.

However, this raises the question of how many perturbation types one should include for training?

Randomized smoothing technique introduces run-time randomization that evaluates multiple noisy copies of a test image and returns the most probable class as their final prediction [9], [10]. This framework provides certified robustness for minor-ℓ_2 perturbation. However, it offers no guarantee for other perturbation types.

In contrast, *manifold-based defenses* incorporate a generative network to project an input image into the clean data manifold and have the potential to achieve robustness for multiple perturbation types [11]–[13]. However, due to the limited capacities of existing generative models, they often fail to capture the complete data manifold for complex image domains and project the adversarial images into the clean data manifold. To date, Schott et al. (2019) [13] provide the only effective manifold-based defense to achieve robustness for MNIST. Even then, they cannot train a single robust classifier for all $\ell_{p \geq 1}$ perturbation types.

In this work, we propose a novel manifold-based defense framework for image classification that can scale to complex data manifold and achieve robustness against any minor for $\ell_{p \geq 1}$ perturbation. Given an image, $\boldsymbol{x}$ and some perturbation bound ϵ_p, for all $p \geq 1$, the goal is to build a classifier $\mathcal{F}$ such that the prediction remains unchanged, that is,

$$\mathcal{F}(\boldsymbol{x}) = \mathcal{F}(\boldsymbol{x} + \boldsymbol{\delta}), \qquad ||\boldsymbol{\delta}||_p < \epsilon_p, \qquad \forall \, p \geq 1 \qquad (1)$$

To this end, we devise an approximate manifold defense mechanism called *RBF-CNN* that can achieve robustness for any minor $\ell_{p \leq 1}$ perturbation. Our RBF-CNN models consist of a radial basis function (RBF) layer and a reconstruction layer followed by a convolutional neural network (CNN) image classifier. An RBF layer consists of RBF filters and is utilized as a generative structure to capture the density of small image patches, instead of the distribution of full-sized training images [14]. Each RBF filter acts as a template matching function by producing similar match scores for two similar patterns in any $\ell_{p \geq 1}$-norms [2]. We show that our reconstruction layer utilizes this property to mitigate any minor perturbation in any $\ell_{p \geq 1}$-norm. Further, incorporating the reconstruction process for training allows us to improve the adversarial robustness of our RBF-CNN models.

Experimental results on MNIST and CIFAR-10 demonstrate that our RBF-CNN models achieve robustness against all ℓ_1, ℓ_2 and ℓ_∞ bounded adversarial attacks and provide certified robustness for ℓ_2 bounded perturbations. The saliency maps produced by RBF-CNN models conform to human interpretation and thereby ensure that our RBF and reconstruction layers do not cause gradient obfuscation [15]. Therefore, our framework does not provide a false sense of robustness by resisting attackers to generate optimal adversarial examples [16]. Further, these interpretable saliency maps provide visual evidence of robustness against adversarial attacks [17]. To the best of our knowledge, RBF-CNN is the first manifold based defense framework to achieve robustness for any $\ell_{p\geq1}$ perturbations and provides a desirable trade-off between robustness vs. accuracy at run-time.

II. RELATED WORK

Several defense models have been proposed to improve the robustness of DNN classifiers against adversarial perturbations. Among the existing defense frameworks, adversarial training provides the best empirical robustness against adversarial attacks. Madry et al. (2018) [6] achieve ℓ_∞-robustness by training the classifier using expensive ℓ_∞-bounded PGD adversaries. This framework can be generalized to any perturbation type, and the interpretable loss-gradients indicated that the classifiers do not cause gradient masking [15] [16]. Recent works such as TRADES [7], LLR [18] propose additional loss regularizer for adversarial training models. Shafahi et al. (2019) [19] propose a 'free'-adversarial training to improve the training efficiency by unfolding the PGD-adversary generation step. However, Tramèr and Boneh (2019) [8] demonstrates that it would require the generation of adversarial examples for different perturbation types to achieve robustness for multiple-perturbation, leading to a linear increase in the training time. Their findings have also raised concerns that adversarial training may not be the appropriate direction to improve the adversarial robustness for multiple perturbations [8] [13].

Input transformation techniques such as feature squeezing [20], pixel-deflection [21] have been proposed to heuristically alter the pixels in images. However, these techniques did not induce the correct amount of alteration and thus, could not improve the robustness [16]. Xie et al. (2018) [22] achieve ℓ_∞-robustness by incorporating adversarial training with their denoising network. In contrast, randomized smoothing defenses induce the correct amount of noise to achieve certified robustness [23] [9] [10]. However, their success is limited to minor-ℓ_2-perturbations.

Image quilting [24] is another type of input transformation technique. It replaces the original input patches using the clean patches, selected from a large set of $50,000$ clean patches. This defense has been broken by the adaptive BPDA attack [16]. Unlike the quilting technique, we reconstruct patches using a combination of patch samples, randomly drawn from a much smaller set of our RBF filters, that also inject noises to the images. Instead of using pre-trained weights for the CNN classifiers like [24], we train our model by incorporating the

reconstruction layer and achieve a robust classifier for any minor $\ell_{p\geq1}$ perturbations.

Our proposed defense can be categorized under manifold defense strategy where generative models are applied to project the input or the hidden activations onto the (learned) data manifold. Examples include DefenseGAN [11] using generative adversarial networks [25], Ape-GAN [26] using auto-embedding GAN [27], defense using Variational Autoencoder (VAE) [12], PixelDefend [28]. [13] train multiple VAEs for each class and chose the class with maximum-likelihood score as their prediction. Similar ideas using non-generative models include auto-encoders to project the input images into the 'known' data manifold [22], [29]. However, all these defenses, except [13], are found to be ineffective [16] [30].

The key reason why these defenses fail is that they rely on their generative model(s) to completely remove adversarial perturbations. This would require the generator to completely capture the underlying data manifold. However, due to the limited capacities of the existing generative (or non-generative) models, they fail to efficiently capture the complete data manifold and often over-estimate or under-estimate different modes of the underlying distribution. Further, the classifier used to classify these projected images, often remain adversarially vulnerable as before. Hence, these models are either broken or fail to scale to complex image domains. Here, we address these limitations by using an RBF layer to capture the density of smaller input image patches. Our reconstruction process allows us to train classifiers that is robust against any minor $\ell_{p\geq1}$-perturbations and remain effective on a more complex image dataset such as CIFAR-10.

The most recent work Croce and Hein (2020) [31] design a regularization scheme to provide robustness for small networks for all $\ell_{p\geq1}$ perturbations. However, their success remains limited for much smaller perturbation boundaries and at the cost of much reduced clean data classification accuracy.

III. PROPOSED FRAMEWORK

Figure 1 gives an overview of our proposed RBF-CNN framework. An input image is first passed through the RBF layer to obtain the activation maps of match scores. The reconstruction layer then combines the patch samples, drawn from the RBF filters and use the match scores to produce pseudo-clean images for classification.

a) RBF Layer: An RBF layer is similar to a convolutional layer, except that the convolutional filters are replaced by RBF filters. Given an input image $\boldsymbol{x}$, we slide a fixed-sized window to obtain the image patches $\boldsymbol{z}_i, i = \{1, 2, \cdots\}$. For each image patch $\boldsymbol{z}$, we compute the *match score* $s(\boldsymbol{z}, g)$ from an RBF filter $g(\boldsymbol{\mu}, \sigma)$ as follows:

$$s(\boldsymbol{z}, g) = \log\left(\frac{1}{\sqrt{2\pi}\sigma}\exp-\frac{||\boldsymbol{z}-\boldsymbol{\mu}||_2^2}{2\sigma^2}\right) \quad (2)$$

where the filter-mean $\boldsymbol{\mu}$ is of the same shape as the image patch $\boldsymbol{z}$, and $\sigma \in \mathbb{R}$ denotes the spread of the filter.
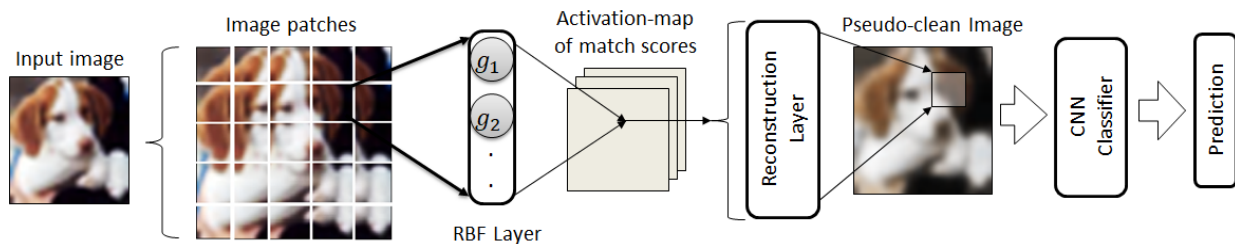
Fig. 1: Overview of the proposed RBF-CNN framework.

As we can see in Eqn. 2, an RBF filter produces a high match score $s(\boldsymbol{z}, g)$ for all the nearby patches $\boldsymbol{z}$ such that the Euclidean distance $||\boldsymbol{z} - \boldsymbol{\mu}||_2^2$ is small. Thus, an RBF filter $g(\boldsymbol{\mu}, \sigma)$ can be viewed as a template matching function where the mean $\boldsymbol{\mu}$ acts as a template to be matched with the input image patch $\boldsymbol{z}$ to produce the match score. Figure 2 visualizes a few RBF filter means of size (3×3) learned from clean CIFAR-10 training images.

Fig. 2: Visualization of a few RBF filter means of patch size 3×3 for the CIFAR-10 classification model.

However, learning the RBF filters, along with the CNN classifier, using the back-propagation algorithm is difficult as they may not be able to efficiently capture the density of image patches while minimizing classification loss. We overcome this by separately training the RBF layer in an unsupervised fashion to capture the density of the image patches. Here, we use a non-parametric variant of expectation maximization (EM) algorithm to learn the filter parameters and capture the density of the image patches [14].

The non-parametric EM is a hard clustering algorithm that automatically determines the required number of filters. We start with one filter $g_1(\boldsymbol{\mu}_1, \sigma_1)$ where the filter mean $\boldsymbol{\mu}_1$ and scaling parameter σ_1 are randomly initialized. New filters are then created subsequently in the following iterations consisting of E and M steps.

The E-Step computes the match score $s(\boldsymbol{z}_i, g_j)$ of the filter $g_j(\boldsymbol{\mu}_j, \sigma_j)$ for each clean training image patch $\boldsymbol{z}_i$. We assign patches with maximum match scores to the RBF filters. If the score is lower than some pre-determined threshold, we create a new filter for $\boldsymbol{z}_i$. The M-step updates the parameters of the RBF filters using the assigned patches as follows:

$$\boldsymbol{\mu}_j^{next} = \frac{\sum_{\boldsymbol{z}_i \in J} \boldsymbol{z}_i}{n_J}, \qquad \sigma_j^{next} = \sqrt{\frac{\sum_{\boldsymbol{z}_i \in J} ||\boldsymbol{z}_i - \boldsymbol{\mu}_j^{next}||_2^2}{n_J}} \quad (3)$$

where J is the set of patches assigned to the filter g_j and n_J denotes the cardinality of set J.

b) **Reconstruction Layer**: Our reconstruction layer does not contain any learn-able parameters. It reuses the RBF filters to separately reconstruct each input patch to produce a pseudo-clean image in the following two steps.

Step 1. Compute weight-vector from the match scores.
For each input patch $\boldsymbol{z}$, we obtain the match scores $s(\boldsymbol{z}, g_j) \; \forall j = 1, 2, \cdots$ and apply sigmoid activation to obtain the vector $\boldsymbol{v}$. The dynamic range of $\boldsymbol{v}$ is increased by applying an element-wise exponential function $\exp(\beta_1 \boldsymbol{v})$ where β_1 is a hyper-parameter. This is followed by a normalization operation to obtain the weight vectors $\boldsymbol{w}$.

Step 2. Draw samples from RBF filters.
We draw samples from all the RBF filters $g_j(\boldsymbol{\mu}_j, \sigma_j)$ as:

$$\tilde{\boldsymbol{z}}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \sigma_j \beta_2 I) = \boldsymbol{\mu}_j + \mathcal{N}(0, \sigma_j \beta_2 I) \quad (4)$$

where β_2 is a hyper-parameter that controls the amount of noise to be added to the patch $\tilde{\boldsymbol{z}}_j$, and I is the identity matrix.

We reconstruct the patch as the weighted sum of these samples $\boldsymbol{w}^T \boldsymbol{Z}$. The reconstructed image is obtained by reconstructing all the original input patches of image $\boldsymbol{x}$ and stitching them together by averaging the overlapping regions. Note that RBF-CNN is a randomized framework where the noise is injected during the sampling process in the reconstruction layer. This differs from the existing randomization smoothing techniques, where noise is injected in the input layer [9] [23].

The noise-level hyper-parameter β_2 controls the trade-off between robustness and accuracy during inference. We can pick an appropriate value for β_2 to obtain the desired level of robustness at run-time. To achieve high accuracy, we can set $\beta_2 = 0$ and obtain $\boldsymbol{z}_j = \boldsymbol{\mu}_j$ in Eqn. 4. In our experiments, we evaluate the robustness of RBF-CNN for different values of β_2, and demonstrate that our proposed framework remains robust for all ℓ_1, ℓ_2 and ℓ_∞ perturbations even for $\beta_2 = 0$. Alternatively, as we choose a higher value for β_2, the sample space of image patches for reconstruction is increased, thus improving the robustness of the classifier.

c) **Classification**: The RBF and reconstruction layers effectively mitigate any minor $\ell_{p \geq 1}$ perturbations before feeding the images into the CNN classifier. Since the reconstruction

	Clean Input	BPDA + EoT $\ell_1 = 20$	$\ell_2 = 1$	$\ell_\infty = 0.031$	Clean Input	EAD $\ell_1 = 20$	CW $\ell_2 = 1$	CW $\ell_\infty = 0.031$
Before Reconstruction								
Euclidean Dist.	-	0.6	1.0	1.4	-	0.6	1.0	1.4
After Reconstruction								
Euclidean Dist.	-	0.4	0.7	0.8	-	0.4	0.7	0.8

Fig. 3: Visualizing the effect of our reconstruction process to mitigate minor perturbations in any $\ell_{p \geq 1}$ norm.

process incorporates randomization, we obtain the final prediction as an average of m different runs as:

$$argmax \sum_{i=1}^{m} rCNN_{\beta_2}(\boldsymbol{x}) \qquad (5)$$

In our experiments, we use $m = 10$. In other words, for each test image, we create a batch of size m for the same image and execute them in parallel and obtain the final prediction by averaging these m prediction.

A. Analysis of the Reconstruction Process

An RBF filter acts as a template matching function that computes the match score as a function of the Euclidean distance between an input patch and the filter mean (see Eqn. 2). Hence, as an attacker chooses minor $\ell_{p \geq 1}$ adversarial perturbations, the match scores produced by the RBF filters change minimally. The reconstruction layer then combines the samples drawn from the RBF filters using the normalized match scores as the weight vectors to reconstruct pseudo-clean images, thereby mitigating minor perturbations.

Claim 1. *Reconstruction process mitigates the effect of minor $\ell_{p \geq 1}$ bounded adversarial perturbations and produces pseudo-clean images for classification.*

Proof. We provide the proof for only ℓ_∞-norm. The proof for other $\ell_{p \geq 1}$ can be obtained similarly. Let $\boldsymbol{x}'$ be an adversarial image, obtained from a clean image $\boldsymbol{x}$ by modifying the pixels as: $x'_i = x_i + \delta_i$. Let $\boldsymbol{z}'$ be the patch obtained from the adversarial image $\boldsymbol{x}'$. Suppose $g(\boldsymbol{\mu}, \sigma)$ is an RBF filter that produces a match score, $s(\boldsymbol{z}, g)$ for a clean patch $\boldsymbol{z}$ of $\boldsymbol{x}$ (recall Eqn. 2). Then we can express the match score $s(\boldsymbol{z}', g)$ as:

$$s(\boldsymbol{z}', g) = s(\boldsymbol{z}, g) - \frac{\sum_i \delta_i^2}{2\sigma^2} - \frac{\sum_i \delta_i(x_i - \mu_i)}{\sigma^2} \qquad (6)$$

We first establish that $s(\boldsymbol{z}', g) - s(\boldsymbol{z}, g)$ is bounded. For ℓ_∞ perturbations, we can bound $\sum_i \delta_i(x_i - \mu_i)$ as follows:

$$\left| \sum_i \delta_i(x_i - \mu_i) \right| \leq \delta_{max} \sum_i |(x_i - \mu_i)| \\ = \delta_{max} ||\boldsymbol{z} - \boldsymbol{\mu}||_1 \leq \delta_{max} \sqrt{n_z} ||\boldsymbol{z} - \boldsymbol{\mu}||_2, \qquad (7)$$

where $\delta_{max} = ||\boldsymbol{\delta}||_\infty$ and n_z is the number of pixels in $\boldsymbol{z}$.

Here, we use *Cauchy-Schwarz inequality* to get $||\boldsymbol{z} - \boldsymbol{\mu}||_1 \leq \sqrt{n_z} ||\boldsymbol{z} - \boldsymbol{\mu}||_2$. Since $\frac{\sum_i \delta_i^2}{2\sigma^2} > 0$, we combine Eqn. 6 and Eqn. 7 to obtain the bound as:

$$-\frac{n_z \delta_{max}^2}{2\sigma^2} - \frac{\delta_{max}\sqrt{n_z}||\boldsymbol{z} - \boldsymbol{\mu}||_2}{\sigma^2} \\ \leq s(\boldsymbol{z}', g) - s(\boldsymbol{z}, g) < \frac{\delta_{max}\sqrt{n_z}||\boldsymbol{z} - \boldsymbol{\mu}||_2}{\sigma^2} \qquad (8)$$

We show that this is insignificant to mitigate any minor ℓ_∞-perturbation. Since δ_{max} is small, the term $\frac{\sum_i \delta_{max}^2}{2\sigma^2} \to 0$ in Eqn. 8. If $g(\boldsymbol{\mu}, \sigma)$ produces a high match score for patch $\boldsymbol{z}$, then $||\boldsymbol{z} - \boldsymbol{\mu}||_2$ must be small. Hence, $\frac{\delta_{max}\sqrt{n_z}||\boldsymbol{z} - \boldsymbol{\mu}||_2}{\sigma^2} \to 0$. Thus, the lower bound of $s(\boldsymbol{z}', g)$ of $g(\boldsymbol{\mu}, \sigma)$ remains almost the same as $s(\boldsymbol{z}, g)$ and g still produces a high score.

In contrast, a low match score for $\boldsymbol{z}$ by an RBF filter $g'(\boldsymbol{\mu}', \sigma')$ implies $||\boldsymbol{z} - \boldsymbol{\mu}'||_2$ is large (recall Eqn. 2). We analyze the maximum value attained by $g'(\boldsymbol{\mu}', \sigma')$ and obtain:

$$s(\boldsymbol{z}', g') < \log \frac{1}{\sqrt{2\pi}\sigma'} - \frac{(||\boldsymbol{z} - \boldsymbol{\mu}'||_2 - 2\delta_{max}\sqrt{n_z})||\boldsymbol{z} - \boldsymbol{\mu}'||_2}{2(\sigma')^2} \qquad (9)$$

Hence, to achieve a high value of $s(\boldsymbol{z}', g')$, the term $(||\boldsymbol{z} - \boldsymbol{\mu}'||_2 - 2\delta_{max}\sqrt{n_z})$ should be small. However, since δ_{max} is small and $||\boldsymbol{z} - \boldsymbol{\mu}'||_2$ is large, we have $||\boldsymbol{z} - \boldsymbol{\mu}'||_2 >> 2\delta_{max}\sqrt{n_z}$. Hence, g' still produces low match score for $\boldsymbol{z}'$.

Since the difference between the match scores of clean images and the corresponding adversarial images remain insignificant, the weight vectors for the reconstruction process will hardly change. Hence, the reconstruction process mitigates their distance when reproducing images from the same set of samples, drawn from the RBF filters. $\square$

Figure 3 illustrates that our reconstruction process mitigates the distances between the images irrespective of all minor perturbation in any $\ell_{p \geq 1}$ norm. Hence, our reconstruction process would enforce the adversaries to choose larger perturbation bounds to circumvent the classifiers. However, we still need to robustly classify these reconstructed images with mitigated perturbations.

Incorporating the reconstruction process for training improves the adversarial robustness. Schmidt et al. (2018) [32] and Hendrycks et al. (2019) [33] have shown that adversarial robustness of a DNN classifier improves by incorporating

more training images. On the other hand, by following our Claim 1, we argue that our reconstruction process mitigates the distance between any two images within a small neighborhood in any $\ell_{p \geq 1}$ norm. In other words, it projects the original manifold of input images into a more *compact manifold* of reconstructed images. Hence, incorporating our proposed reconstruction process for training reduces the requirement of additional images to improve the robustness of our RBF-CNN models. Moreover, our Claim 1 implies that our reconstruction process would produce almost the same image for a set of input images in a small neighborhood of any $\ell_{p \geq 1}$-norm. Hence, by augmenting minor random noise to our training images, we can efficiently train our models to be more aware of the surroundings of the data manifold. Our experiments on MNIST and CIFAR-10 supports this observation.

IV. Performance Study

We carried out three sets of experiments to evaluate the robustness of our RBF-CNN models on MNIST [34] and CIFAR-10 [35][1]. First, we empirically evaluate the robustness against a wide range of ℓ_1, ℓ_2 and ℓ_∞ bounded attacks. We demonstrate that RBF-CNN models improve certified robustness for ℓ_2 perturbations. Next, we visualize that our RBF-CNN models produce interpretable saliency maps to ensure that our framework does not cause gradient obfuscation [15]. Finally, we demonstrate that our RBF-CNN models allow robustness vs. accuracy flexibility at run-time.

Experimental Setup. We use a 4-layer CNN for MNIST, and VGG-16 [36] for CIFAR-10. We train two sets of RBF-CNN models, denoted as $rCNN$ and $rCNN_+$. We use 3×3 filters for the RBF layer and train with the non-parametric EM that automatically learns 24 and 232 filters for MNIST and CIFAR-10 respectively. The CNN components of these models are trained using *label smoothing*. We use only clean images to train our $rCNN$ models. For $rCNN_+$ models, we use the clean images as well as noisy images, perturbed within a ℓ_∞ boundary of 0.3 and 0.03 for MNIST and CIFAR-10 respectively. Here, the noises for $rCNN_+$ is sampled from an isotropic Gaussian ($\mathcal{N}(0, 0.35)$ and $\mathcal{N}(0, 0.05)$ respectively) and clipped within those predefined ℓ_∞ boundaries. We also generate one set of PGD-adversarial examples of the training images at 25 and 200 epochs for MNIST and CIFAR-10 respectively. Then we obtain the *adversarial noises* by subtracting the original training images from the adversarial examples. We randomly add this adversarial noise to the clean training images for the rest of the training epochs.

During testing, β_1 is set to 25 for both datasets. The hyper-parameter β_2 controls the trade-off between accuracy and robustness for the classifiers. We do not inject any noise through the reconstruction layer during training. For testing, we set β_2 to 1.75 for both MNIST and CIFAR-10. For CIFAR-10, we choose β_2 such that the classification accuracy for clean images does not drop less than 85%.

[1]Code is available at //github.com/jayjaynandy/RBF-CNN.

A. Performance against Adversarial Attacks

The *robust accuracy* of a defense model is defined by their performance against the strongest adversarial attack within a specified ℓ_p boundary [16]. For RBF-CNN, we reconstruct images by drawing samples from the RBF filters. Noise is injected to the images during this sampling process using an isotropic Gaussian distribution (see Eqn. 4). The RBF filters and the reconstruction step are otherwise differentiable.

We apply the following wide range of adversarial attacks to evaluate the robustness of our models:

- Static Attacks. Adversarial examples are generated by removing the non-differentiable noise injection step from the reconstruction layer, i.e $z_j = \mu_j$ in Eqn. 4. Here, we consider single-step attack FGSM [2], iterative attacks such as PGD [6], MI-FGSM [37], CW [38], DAA [39], EAD [40].

- Adaptive Attacks. An adaptive attack is constructed after the defense model has been completely specified, such that the adversary can find the *optimal* adversarial perturbations within the specified perturbation boundaries [16]. For our RBF-CNN models, we design the adaptive attack as a combination of BPDA and EoT (Expectation of Transformations) [16], as follows. In the forward propagation, we do not change anything in the network. In the backward propagation, we choose a differentiable approximation for the reconstruction layer, by considering $z_j = \mu_j$ in Eqn. 4 to efficiently compute the gradients. Since the RBF filters are already differentiable, we do not need any approximation for the RBF layer. We compute the expected loss using a Monte-Carlo method with 50 simulations to find the strongest adversarial examples.

- Black-box Attacks. We also evaluate against gradient-free SPSA attack [41] to ensure that our framework is not giving any false sense of robustness using gradient-masking [16]. We apply *black-box transfer attacks* where the adversarial examples are generated by attacking a standard CNN classifier with no defense.

ℓ_∞ **Bounded Attacks.** Table I presents the performance of our $rCNN$ and $rCNN_+$ models against ℓ_∞ bounded attacks. We choose the perturbation boundaries of 0.3 and 0.031 for MNIST and CIFAR-10 respectively. We observe that for both MNIST and CIFAR-10, $rCNN_+$ achieves high robustness under ℓ_∞ perturbations bounds of 0.3 and 0.031 respectively.

One main advantage of RBF-CNN compared to the adversarial training frameworks is that it significantly reduces the training time required. Here, we compare the training times of our RBF-CNN models with Madry's models, that achieved robustness only for ℓ_∞ perturbations [6].

Table II shows the training times when executed on a GTX 1080Ti GPU. We observe that the training time of $rCNN_+$ models are more than $4.5\times$ and $7\times$ faster than the Madry models for MNIST and CIFAR-10 respectively. Even as our proposed approach significantly reduces the training time required, our RBF-CNN models are still able to achieve similar performances as Madry's models for ℓ_∞ perturbations in both

Attack Name	MNIST		CIFAR-10	
	$rCNN$	$rCNN_+$	$rCNN$	$rCNN_+$
	($\ell_\infty \leq 0.3$)		($\ell_\infty \leq 0.031$)	
Clean Test Data	99.6	99.5	85.0	85.1
FGSM	94.7	98.0	75.6	78.2
PGD	78.3	94.9	63.1	60.0
MI-FGSM	70.5	94.5	65.8	70.2
CW	83.1	91.1	46.6	46.2
DAA	41.5	94.8	63.8	62.8
BPDA+EoT	39.4	88.4	47.2	49.8
SPSA (Black-box)	61.4	92.9	62.4	67.6
FGSM (Black-box)	77.9	90.9	76.0	78.7
PGD (Black-box)	39.6	90.5	80.3	81.8
CW (Black-box)	63.0	95.5	76.5	79.4

TABLE I: Accuracy of RBF-CNN models under ℓ_∞ attacks.

	Defenses	#Epochs	Time/ Epoch	Overhead	Total
MNIST	Madry	85	112	0	9,520
	$rCNN$	100	4	600	1,000
	$rCNN_+$	100	11	900	**2,000**
CIFAR-10	Madry	205	1200	0	246,000
	$rCNN$	600	29	2700	20,100
	$rCNN_+$	600	51	3600	**34,200**

TABLE II: Training time comparison (in seconds).

	ℓ_∞-bounds	Madry's	$rCNN$	$rCNN_+$
MNIST	0.3	**88.6**	39.4	**88.4**
	0.35	42.9†	10.5	**75.8**
CIFAR-10	0.031	44.7	**46.6**	46.2
	0.05	25.7†	**37.1**	34.7

TABLE IV: Comparison of the robust accuracies at different perturbation boundaries. †Evaluated only against PGD attacks.

ℓ_1 Attacks		Static		Adaptive	Black-box	
		PGD	EAD	BPDA+EoT	PGD	EAD
MNIST ($\ell_1 = 15$)	Madry	**77.5**	90.4	-	97.9	98.2
	$rCNN$	89.8	**69.9**	83.1	98.4	99.3
	$rCNN_+$	95.7	**83.4**	90.5	98.9	99.3
CIFAR-10 ($\ell_1 = 20$)	Madry	**34.5**	35.5	-	86.2	86.1
	$rCNN$	69.2	61.0	**55.5**	82.0	81.1
	$rCNN_+$	76.1	**61.9**	66.8	82.9	82.3
ℓ_2 Attacks	Defenses	Static		Adaptive	Black-box	
		PGD	CW	BPDA+EoT	PGD	CW
MNIST ($\ell_2 = 2$)	Madry	**81.9**	91.0	-	95.6	97.2
	$rCNN$	80.0	72.3	**64.2**	88.3	97.0
	$rCNN_+$	92.6	**83.9**	87.5	96.5	98.6
CIFAR-10 ($\ell_2 = 1$)	Madry	**28.1**	43.3	-	85.2	85.1
	$rCNN$	62.1	47.9	**45.2**	79.1	76.8
	$rCNN_+$	67.3	**54.5**	54.8	81.1	79.5

TABLE V: Performance against ℓ_1 and ℓ_2 bounded attacks.

MNIST and CIFAR-10 as shown in Table III. Also, our best RBF-CNN model for MNIST significantly outperforms the only successful manifold defense model by Schott et al. (2019) [13]. We also compare with other input transformation [42] [24] and network randomization [43] techniques to show that unlike these defenses, our RBF-CNN models remain robust against all type of attacks.

Another limitation of adversarial training models is that they provide robustness only within a pre-specified boundary from where the adversarial examples were produced for their training, and offer no robustness guarantee slightly beyond these boundaries. Table IV shows that [6] achieve high robust accuracies within their specified ℓ_∞ bounds of 0.3 and 0.031 for MNIST and CIFAR-10 respectively. However, their robust accuracies drastically drop beyond those bounds when tested against PGD attack with $\ell_\infty \leq 0.35$ for MNIST and at

	Defenses	Robust Acc.	Strongest Attack
MNIST	Baseline (no defense)	0	PGD
	Madry et al., (2018) [6]	**88.6**	DAA
	Schott et al.,(2019) [13]	78.0	Deep-Fool
	$rCNN$	39.4	BPDA+EoT
	$rCNN_+$	**88.4**	BPDA+EoT
CIFAR-10	Baseline (no defense)	0	PGD
	Madry et al., (2018) [6]	44.7	DAA
	Buckman et al., (2018) [42]	30	BPDA
	Ma et al., (2018) [44]	5	CW
	Dhillon et al., (2018) [43]	0	EoT
	Song et al., (2018) [28]	9	BPDA
	Dezfooli et al., (2019) [45]	41.4	PGD
	$rCNN$	**46.6**	CW
	$rCNN_+$	46.2	CW

TABLE III: Comparison of robust accuracy against ℓ_∞ bounded adversarial attacks. Perturbation boundaries for MNIST and CIFAR-10 are set to 0.3 and 0.031 respectively.

$\ell_\infty \leq 0.05$ for CIFAR-10. In contrast, RBF-CNN models achieve significantly high accuracies at these ℓ_∞ boundaries.

ℓ_1 **and** ℓ_2 **Bounded Attacks.** For our experiments on ℓ_1 and ℓ_2 bounded attacks, we choose ℓ_1=15 and ℓ_2=2 for MNIST and ℓ_1=20 and ℓ_2=1 for CIFAR-10. For MNIST, we see that the perception of the adversarial images is changing around these bounds (see Fig. 4). Note that, such behavior cannot be observed when attacking a non-robust classifier [15].
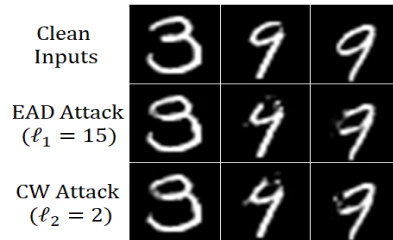

Fig. 4: Visual perception of MNIST images changed under ℓ_1 and ℓ_2 bounded attacks for RBF-CNN models.

Table V presents the performance of RBF-CNNs against different ℓ_1 and ℓ_2 bounded attacks. We observe that RBF-CNN models significantly outperform Madry's models, that are trained to provide robustness for ℓ_∞ perturbations.

Certification for ℓ_2-perturbations. Certification provides a lower bound guarantee of robust accuracy. Since the reconstruction process of our RBF-CNN models mitigate *any* minor perturbations, it also improves the classification performance against random Gaussian perturbed images. We apply the existing certification technique of randomized smoothing defenses [9] to further demonstrate that our RBF-CNN models

		$\ell_2 = 0.5$	$\ell_2 = 0.7$	$\ell_2 = 1.0$
	Baseline	92.7	84.4	0.0
MNIST	$rCNN$	94.7	85.3	0.0
	$rCNN_+$	**98.0**	**96.3**	**89.2**
		$\ell_2 = 0.25$	$\ell_2 = 0.3$	$\ell_2 = 0.35$
	Baseline	10.2	8.1	4.7
CIFAR-10	$rCNN$	37.2	32.3	25.3
	$rCNN_+$	**47.2**	**40.6**	**33.4**

TABLE VI: Certified robust accuracy for ℓ_2 perturbations.

also improve the certified robustness for ℓ_2 perturbations compared to the baseline models with no defense.

For each input x, we sample $50,000$ noisy samples from an isotropic Gaussian distribution $\epsilon \sim \mathcal{N}(0, \tau^2 I)$. For MNIST, we set τ to 0.2 for the baseline and $rCNN$ and 0.3 for $rCNN_+$. For CIFAR-10, we set $\tau = 0.1$ for all the models. Table VI shows the certified robust accuracies. We see that for CIFAR-10, the robustness of the CNN classifier is improved by adding RBF and reconstruction layers. The results for $rCNN_+$ suggests that the certified robustness of our models are further improved by augmenting minor noises the images for training. To summarize, our RBF-CNN models improves the empirical robustness against any minor $\ell_{p \geq 1}$ perturbations as well as certified robustness for ℓ_2 perturbations.

B. Interpretable Loss Gradients

Figure 5 visualizes the loss gradients of our RBF-CNN, $rCNN_+$ versus standard CNN. These loss gradients represent the most important pixels for the classifier. All these loss gradients are obtained in one single step by computing the losses with respect to the input pixels. For standard CNNs, these gradients appear noisy and incoherent patterns. In contrast, the loss gradients for RBF-CNN are aligned with human perception without any pre-processing other than scaling and clipping. This ensures that our RBF and reconstruction layers do not introduce gradient masking in our RBF-CNN framework. Further, Etmann et al. (2019) [17] demonstrate that only the robust classification models exhibit such interpretable saliency maps.

Fig. 5: Visualization of loss gradients.

A first-order adversary iteratively uses these loss gradients to minimize the salient features of the original class and maximize the salient features of a different class to fool a model. Thus, it is often not possible to fool a robust classifier

as such perturbation may not exist within a small ℓ_p neighborhood and the adversary would require larger perturbations. Consequently, as we allow a large ℓ_2 boundary and apply the static PGD attacks, we observe sharp salient features of a different class appear in the generated images for our robust RBF-CNN models (see Fig. 6). In contrast, the attack only able to produce a noisy version of the clean image for the non-robust standard CNN models.
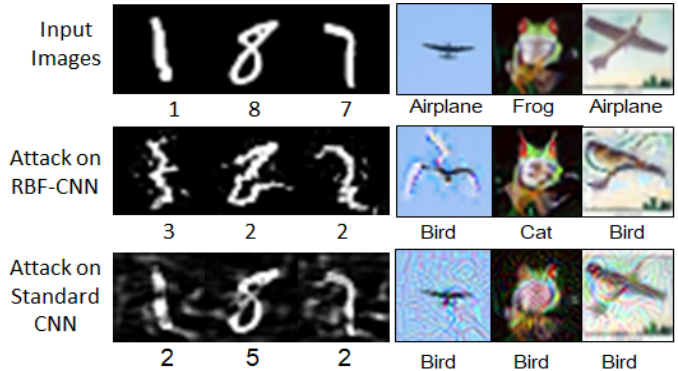

Fig. 6: Adversarial Images generated using PGD attack.

C. Robustness vs. Accuracy Trade-off at run-time

Ideally, a user should be able to select an optimal trade-off between robustness or accuracy performance at run-time. In RBF-CNN, we can vary the level of injected noise, β_2, during test time to achieve a different degree of robustness versus accuracy performance. Table VII presents the performance of $rCNN_+$ as we vary β_2. We see that $rCNN_+$ is able to achieve robustness for all ℓ_1, ℓ_2 and ℓ_∞ bounded attacks even when we remove the randomization step by setting $\beta_2 = 0$. Further, the robust accuracy doubles for CIFAR-10 when the noise level of $\beta_2 = 1.75$ is injected with minimum effect on its clean data accuracy. To the best of our knowledge, none of the existing methods allow such a trade-off between robustness versus accuracy at run-time.

Attack Name	MNIST			CIFAR-10		
β_2	0.0	1.0	1.75	0.0	1.5	1.75
Clean Test Data	99.5	99.5	99.4	89.2	87.2	85.1
	$\ell_\infty \leq 0.3$			$\ell_\infty \leq 0.031$		
PGD	91.6	93.1	94.9	51.1	59.4	60.0
CW	**77.7**	**85.8**	91.1	**20.9**	**44.3**	**46.2**
BPDA+EoT	-	88.4	**88.4**	-	50.6	49.8
PGD (black-box)	93.7	92.3	90.5	86.6	85.1	81.8
	$\ell_1 \leq 15$			$\ell_1 \leq 20$		
PGD	93.8	94.6	95.7	73.3	76.2	76.1
EAD	**80.6**	**82.0**	**83.4**	**41.1**	**61.1**	**61.9**
BPDA+EoT	-	90.2	90.5	-	67.9	66.8
	$\ell_2 \leq 2$			$\ell_2 \leq 1$		
PGD	88.4	90.5	92.6	63.8	66.8	67.3
CW	**80.4**	**82.2**	**83.9**	36.5	**52.7**	**54.5**
BPDA+EoT	-	87.1	87.5	-	56.4	54.8

TABLE VII: Robustness versus Accuracy tread-off at run-time for RBF-CNN models, $rCNN_+$ by varying the noise level hyper-parameter, β_2 in the reconstruction layer.

V. Conclusion

Existing successful defense models typically achieve robustness only for a specific perturbation type while providing no guarantee for other perturbation types. Towards this, we presented an "approximate manifold" defense called RBF-CNN that achieves robustness for all minor perturbations in any ℓ_p-norm with $p \geq 1$. Our RBF-CNN utilizes an RBF and a reconstruction layer. We propose to capture the density of small image patches, instead of capturing the complete data manifold to address the limitations of the existing manifold defenses. Our experimental results on MNIST and CIFAR-10 demonstrate that we can train a single RBF-CNN model to provide robustness for all ℓ_1, ℓ_2, and ℓ_∞ perturbations. While previously, the success of the only effective manifold-based defense remains limited to MNIST [13], we achieve robustness for a much complex image dataset, called CIFAR-10.

Even though our proposed RBF-CNN models achieve robustness against minor additive perturbations in-terms of any $\ell_{p\geq 1}$ norms, we do not necessarily provide any guarantee for other perturbation types such as spatial [46]–[48] or color transformations [49] or naturally occurring common perturbation types [50], [51]. While it is crucial to develop universally robust defense models for real-world applications to improve their reliability, it remains an open problem to the AI community.

References

[1] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *ICLR*, 2014.

[2] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.

[3] N. Papernot *et al.*, "The limitations of deep learning in adversarial settings," in *IEEE Euro S&P*, 2016.

[4] G. E. Dahl, J. W. Stokes, L. Deng, and D. Yu, "Large-scale malware classification using random projections and neural networks," in *IEEE ICASSP*, 2013.

[5] F. Tramèr, P. Dupré, G. Rusak, G. Pellegrino, and D. Boneh, "Adversarial: Defeating perceptual ad-blocking," *ACM CCS*, 2019.

[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *ICLR*, 2018.

[7] H. Zhang *et al.*, "Theoretically principled trade-off between robustness and accuracy," in *ICML*, 2019.

[8] F. Tramèr and D. Boneh, "Adversarial training and robustness for multiple perturbations," in *NeurIPS*, 2019.

[9] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *ICML*, 2019.

[10] B. Li, C. Chen, W. Wang, and L. Carin, "Certified adversarial robustness with additive noise," in *NeurIPS*, 2019.

[11] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," in *ICLR*, 2018.

[12] P. Ghosh, A. Losalka, and M. J. Black, "Resisting adversarial attacks using gaussian mixture variational autoencoders," in *AAAI*, 2019.

[13] L. Schott, J. Rauber, M. Bethge, and W. Brendel, "Towards the first adversarially robust neural network model on MNIST," in *ICLR*, 2019.

[14] J. Nandy, W. Hsu, and M. L. Lee, "Normal similarity network for generative modelling," in *IEEE ICIP*, 2018.

[15] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," in *ICLR*, 2019.

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

[17] C. Etmann, S. Lunz, P. Maass, and C.-B. Schönlieb, "On the connection between adversarial robustness and saliency map interpretability," in *ICML*, 2019.

[18] C. Qin *et al.*, "Adversarial robustness through local linearization," in *NeurIPS*, 2019.

[19] A. Shafahi *et al.*, "Adversarial training for free!" in *NeurIPS*, 2019.

[20] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," in *NDSS*, 2018.

[21] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer, "Deflecting adversarial attacks with pixel deflection," in *CVPR*, 2018.

[22] F. Liao *et al.*, "Defense against adversarial attacks using high-level representation guided denoiser," in *CVPR*, 2018.

[23] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, "Certified robustness to adversarial examples with differential privacy," *IEEE S&P*, 2018.

[24] C. Guo, M. Rana, M. Cisse, and L. Van Der Maaten, "Countering adversarial images using input transformations," in *ICLR*, 2018.

[25] I. Goodfellow *et al.*, "Generative adversarial nets," in *NIPS*, 2014.

[26] G. Jin, S. Shen, D. Zhang, F. Dai, and Y. Zhang, "APE-GAN: Adversarial perturbation elimination with gan," 2019.

[27] Y. Guo *et al.*, "Auto-embedding generative adversarial networks for high resolution image synthesis," *IEEE Transactions on Multimedia*, 2019.

[28] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," in *ICLR*, 2018.

[29] D. Meng and H. Chen, "Magnet: A two-pronged defense against adversarial examples," in *ACM CCS*, 2017.

[30] A. Jalal, A. Ilyas, C. Daskalakis, and A. G. Dimakis, "The robust manifold defense: Adversarial training using generative models," *arXiv*, 2019.

[31] F. Croce and M. Hein, "Provable robustness against all adversarial l_p-perturbations for $p \geq 1$," in *ICLR*, 2020.

[32] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Madry, "Adversarially robust generalization requires more data," in *NeurIPS*, 2018.

[33] D. Hendrycks, K. Lee, and M. Mazeika, "Using pre-training can improve model robustness and uncertainty," in *ICML*, 2019.

[34] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010.

[35] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Master's thesis, University of Toronto*, 2009.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[37] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *CVPR*, 2018.

[38] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *IEEE S&P*, 2017.

[39] T. Zheng, C. Chen, and K. Ren, "Distributionally adversarial attack," in *AAAI*, 2019.

[40] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, "EAD: Elastic-net attacks to deep neural networks via adversarial examples," in *AAAI*, 2018.

[41] J. Uesato, B. O'Donoghue, A. v. d. Oord, and P. Kohli, "Adversarial risk and the dangers of evaluating against weak attacks," *ICML*, 2018.

[42] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *ICLR*, 2018.

[43] G. S. Dhillon *et al.*, "Stochastic activation pruning for robust adversarial defense," in *ICLR*, 2018.

[44] X. Ma *et al.*, "Characterizing adversarial subspaces using local intrinsic dimensionality," in *ICLR*, 2018.

[45] S. M. Moosavi Dezfooli, A. Fawzi, J. Uesato, and P. Frossard, "Robustness via curvature regularization, and vice versa," in *CVPR*, 2019.

[46] C. Xiao *et al.*, "Spatially transformed adversarial examples," in *ICLR*, 2018.

[47] H. Zhang *et al.*, "The limitations of adversarial training and the blind-spot attack," in *ICLR*, 2019.

[48] E. Wong, F. Schmidt, and Z. Kolter, "Wasserstein adversarial examples via projected sinkhorn iterations," in *ICML*, 2019.

[49] C. Laidlaw and S. Feizi, "Functional adversarial attacks," in *NIPS*, 2019.

[50] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *ICLR*, 2019.

[51] R. Geirhos *et al.*, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," 2019.