

Beating Attackers At Their Own Games: Adversarial Example Detection Using Adversarial Gradient Directions

Yuhang Wu, Sunpreet S. Arora, Yanhong Wu, and Hao Yang

Visa Research

{yuhawu, sunarora, yanwu, haoyang}@visa.com

Abstract

Adversarial examples are input examples that are specifically crafted to deceive machine learning classifiers. State-of-the-art adversarial example detection methods characterize an input example as adversarial either by quantifying the magnitude of feature variations under multiple perturbations or by measuring its distance from estimated benign example distribution. Instead of using such metrics, the proposed method is based on the observation that the directions of adversarial gradients when crafting (new) adversarial examples play a key role in characterizing the adversarial space. Compared to detection methods that use multiple perturbations, the proposed method is efficient as it only applies a single random perturbation on the input example. Experiments conducted on two different databases, CIFAR-10 and ImageNet, show that the proposed detection method achieves, respectively, 97.9% and 98.6% AUC-ROC (on average) on five different adversarial attacks, and outperforms multiple state-of-the-art detection methods. Results demonstrate the effectiveness of using adversarial gradient directions for adversarial example detection.

Introduction

Deep neural networks (DNNs) are being widely used in classification systems because of their exceptional performance on a wide range of practical problems from fraud detection to biometrics. However, recent research in adversarial machine learning has highlighted a major security concern with use of the DNNs in practical applications. Researchers have shown that an adversary can add human-imperceptible malicious perturbations to input examples to cause incorrect predictions (Szegedy et al. 2014; Carlini and Wagner 2017). Such input examples are termed *adversarial examples*.

State-of-the-art adversarial example detection methods primarily exploit the following two observed properties of adversarial examples: (i) adversarial examples are comparatively more sensitive to perturbations in the input space than benign examples, and (ii) the distance of an adversarial example to the data distribution of benign examples is anomalous.

The methods presented in (Wang et al. 2018; Xu, Evans, and Qi 2018; Tian, Yang, and Cai 2018) are based on observation (i). These methods transform an input example

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

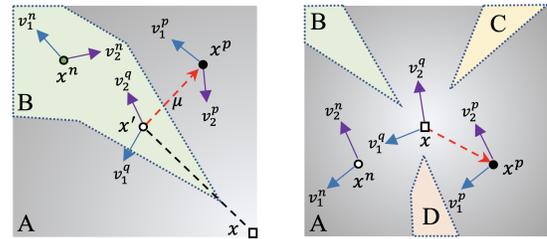


Figure 1: Schematic illustration of the proposed approach in the 2D embedding space (L) of adversarial example x' , and (R) of benign example x . Assume that an attacker crafts an adversarial example x' from a benign example x of class A, and aims to deceive a deep network into misclassifying x' in class B. Given an input example x^q , the proposed method perturbs x^q using a random perturbation μ to obtain x^p , and retrieves x^q 's nearest neighbor prototype x^n from a reference database, which shares the same predicted class with x^q . Following this, adversarial gradients of x^q , x^p , and x^n (denoted as v_k^i) corresponding to their top K predicted classes (here $K=2$) are calculated. Pairwise angular similarity between $\{v_k^q, v_k^p\}$, $\{v_k^q, v_k^n\}$, and $\{v_k^p, v_k^n\}$ is computed ($k = \{1, 2\}$) and used to determine if x^q is adversarial. Note that adversarial gradient directions in (R) are comparatively consistent and have larger angular similarities than (L).

using geometrical transformations (*e.g.*, rotation, translation) and/or filtering operations (*e.g.* median filtering). The changes in prediction probabilities of the classifier before and after transformation are aggregated and compared to determine if the input example is adversarial. Despite promising results, one limitation of these methods is the use of multiple random transformations which, in turn, increases the computational complexity. The proposed method, on the other hand, applies a single random perturbation to an input example to determine if the example is adversarial by using adversarial gradient directions.

Instead of relying on observation (i), the method described in (Feinman et al. 2017) is based on observation (ii) and uses kernel density estimation to identify low probability adversarial sub-spaces. The method reported in (Ma et al. 2018) characterizes the space-filling capability of the region surrounding an adversarial example using local intrinsic di-

mensions, and the one presented in (Lee et al. 2018) models the likelihood of an input example being adversarial (out-of-distribution) using class-conditional Gaussian distributions. A key assumption in these approaches is the availability of sufficient predicted class examples in the reference database to estimate the distribution of benign examples. In contrast, the proposed method uses a single benign example from the predicted class to determine whether an input example is adversarial. This is quite useful in the few shot learning setting, especially when the number of predicted class examples in the reference database is limited.

The proposed method is inspired by the research presented in (Roth, Kilcher, and Hofmann 2019), which shows that for an adversarial input, adding random perturbations to the input space induces directional feature variations. On the contrary, the feature variations induced by random perturbations on a benign input are non-directional. Roth *et al.* (Roth, Kilcher, and Hofmann 2019) modeled the statistics of directional feature variations by measuring the changes in logits between each pair of predicted class examples for different random perturbations. In contrast, our approach characterizes feature variations in a more intuitive manner. Our findings show that the gradient directions used to craft “adversarial examples” are sufficiently discriminative to characterize the adversarial input space. We observe that benign examples usually have similar adversarial gradient directions (AGDs) before and after a random perturbation (RP), while the difference in AGDs of adversarial examples before and after a RP is significant. We also observe that AGDs strongly depend on the original class of adversarial examples, and that the examples that are closer on the data manifold share similar AGDs. Given these observations, we compute AGDs of the nearest predicted class example to the input example (prototype). The AGDs of the input (query) example are compared with the AGDs of the prototype as well as the randomly perturbed example as depicted in Fig. 1. Finally, a classifier is trained using the angular similarities between each pair of the computed AGDs to detect adversarial examples.

The main contributions of this work are as follows:

- Use of Adversarial Gradient Directions (AGDs) for adversarial example detection. To the best of our knowledge, this is the first work that uses AGDs for adversarial example detection. AGDs are typically used for crafting adversarial examples.
- Demonstration of adversarial example detection using a single random perturbation and a single example from the predicted class.
- State-of-the-art adversarial example detection performance in the gray-box and white-box setting on CIFAR-10 and ImageNet databases.

Related Work

Transformation-based defenses use one or more transformations to counter adversarial patterns in input examples. For example, (Kurakin, Goodfellow, and Bengio 2017) use JPEG compression, and (Li and Li 2017) use median-filtering. One of the most effective ways to improve the ro-

bustness of transformation-based defense methods is to introduce randomness. The method proposed in (Yang et al. 2019) randomly deletes pixels from an input example and reconstructs the deleted pixels using matrix estimation to suppress adversarial patterns. A more recent approach (Raff et al. 2019) employs a set of random transformations to defend against strong adversarial attacks generated by a potential adversary with reasonable compute power. The method described in (Xu, Evans, and Qi 2018) applies median smoothing, non-local mean, and bit quantization to an input example and measures the changes in the model’s prediction to determine if the input example is adversarial. The approach presented in (Tian, Yang, and Cai 2018) uses random rotation and scaling operations to improve adversarial example detection accuracy in white-box setting. Compared to these approaches that use multiple transformations, the proposed method simply applies a single random transformation on the input example to determine if it is adversarial.

Neighbor-based defenses exploit peer wisdom to characterize adversarial examples. Most neighbor-based methods can be categorized into either class-independent or class-conditional methods. Class-independent approaches select K -nearest benign examples of an adversarial example from a reference database and use majority class voting (Papernot and McDaniel 2018), pixel-wise relations (Svoboda et al. 2019) or local intrinsic dimension (Ma et al. 2018) to detect adversarial examples. On the other hand, class-dependent approaches use multiple examples of the predicted class (usually greater than ten) to estimate the kernel density (Feinman et al. 2017) or construct a valid data distribution (Lee et al. 2018). An input example that is an outlier with respect to the estimated distribution is labelled adversarial.

Prototype-based methods use a few examples, called prototypes, distilled from the training or reference database for classification. This approach is useful in the few shot learning setting, especially when the number of examples in the reference database is limited. It has been shown that measurement of similarities between test examples and prototypes is quite effective in understanding a model’s behavior. Snell *et al.* (Snell, Swersky, and Zemel 2017) select prototype examples for each class, and use the nearest-class prototype for classification. Arik and Pfister (Arik and Pfister 2019) use prototypes to detect out-of-distribution examples, and provide efficient confidence metrics. Inspired by these methods, we use a prototype from the predicted class of an input example to determine if the input example belongs to the predicted class, and consequently if the input example is adversarial.

Adversarial Example Detection

Problem Definition

We demonstrate the effectiveness of the proposed adversarial example detection method in the multi-class image classification setting. Given a labeled training set \mathbb{D} containing N examples and C classes such that $\mathbb{D} = \{(\mathbf{I}_1, y_1), \dots, (\mathbf{I}_N, y_N)\}$, with labels $y \in \mathbb{Z}_C$, a classifier (e.g., deep neural network) f is trained on \mathbb{D} to classify an input example (image) \mathbf{I} into one of C classes: $f(\mathbf{I}) \rightarrow \mathbb{Z}_C$.

The loss function for the classifier f is $\phi(f(\mathbf{I}), y)$. The adversary aims to create an adversarial example \mathbf{I}' that maximizes ϕ by using a query example \mathbf{I}^q . An additional constraint for the adversary is that the distance between \mathbf{I}' and \mathbf{I}^q should be less than ϵ :

$$\mathbf{I}' = \arg \max_{d(\mathbf{I}^q, \mathbf{I}') < \epsilon} \phi(f(\mathbf{I}'), y), \quad (1)$$

with l_∞ as the assumed distance metric d , which provides the adversary maximum flexibility to craft adversarial examples. Furthermore, the adversary's goal is *misclassification* of the crafted adversarial example \mathbf{I}' i.e. $f(\mathbf{I}') \neq f(\mathbf{I}^q)$.

To secure f from adversarial examples, our goal is to design an adversarial example detector $\tau(\mathbf{I}^q) \rightarrow [0, 1]$, such that $\tau(\cdot)$ outputs a score that indicates whether a query example \mathbf{I}^q is adversarial or benign.

Transformation-based Detection

Let \star denote a generic image transformation operator, and $\mathbf{I}^q \star \mathbf{T}_l$ represent the application of an image transformation \mathbf{T}_l on \mathbf{I}^q , and result in a perturbed (transformed) image: $\mathbf{I}_l^p = \mathbf{I}^q \star \mathbf{T}_l$. Transformation \mathbf{T}_l (e.g., median filtering, scaling, rotation, Gaussian filtering) where $l \in \{1 : L\}$ denotes the l^{th} transformation used in the adversarial example detection method. Let $f^m(\cdot)$ indicate the vectorized output of layer m of deep network f . The transformation-based detection methods in (Xu, Evans, and Qi 2018; Wang et al. 2018; Hu et al. 2019) can be represented as $\tau(\mathbf{I}^q) = S(\{r_0, r_1, \dots, r_L\})$, where the result of feature variations r_l is defined as $r_l = \|f^m(\mathbf{I}_l^p) - f^m(\mathbf{I}^q)\|_1$. The method S aggregates the distances between features of the transformed input examples and the original input example and compares it with a predefined threshold to determine if \mathbf{I}^q is adversarial. Because \mathbf{T}_l is random, it introduces uncertainties that ideally should be compensated by averaging over a large number or type of transformations.

Proposed Method

To detect adversarial examples with high accuracy using a small number of transformations, the use of a significantly discriminative characteristic of the adversarial space is important. To this end, instead of introducing complex transformations in \mathbf{T}_l that further increase computational complexity, the proposed method is based on how an attacker typically generates an adversarial example.

Adversarial gradient direction: When gradient-based attacks (e.g., FGSM (Szegedy et al. 2014) and PGD (Madry et al. 2018)) use an input example \mathbf{I} to generate an adversarial example to cause mis-classification, they optimize Eq. 1 using gradients computed with respect to \mathbf{I} :

$$\mathbf{I}_t = \mathbf{I} + \varepsilon_0 \text{sign}(\nabla_{\mathbf{I}} \phi(\mathbf{I}, a)) \quad (2)$$

Here, a is the predicted class label, and ε_0 is the step size of the gradient update. In each iteration, gradient direction $\nabla_{\mathbf{I}} \phi(\mathbf{I}, a)$, abbreviated as $\nabla \phi(\mathbf{I}, a)$, plays a key role in determining the generated adversarial example. The impact of $\nabla \phi(\mathbf{I}, a)$ on vectorized layer output $f^m(\cdot)$ can be quantified by measuring:

$$\Delta f^m(\mathbf{I}, a) = f^m(\mathbf{I}_t, a) - f^m(\mathbf{I}, a) \quad (3)$$

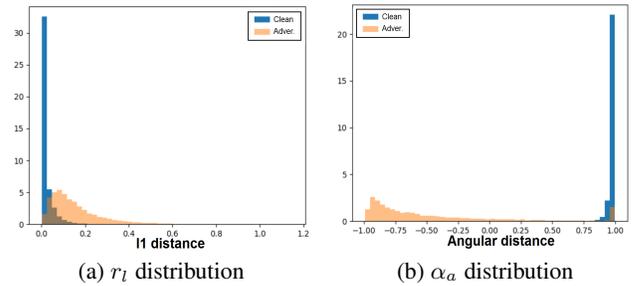


Figure 2: Distributions of (a) the traditional feature variation score r_l and (b) the proposed score α_a for benign and adversarial examples (generated using FGSM attack) computed for ImageNet database in identical experimental settings.

We refer to the direction of the vector $\Delta f^m(\mathbf{I}, a)$ as **Adversarial Gradient Direction (AGD)** targeted at class a . We observe that the direction of $\Delta f^m(\mathbf{I}, a)$ changes significantly if \mathbf{I} corresponds to adversarial example \mathbf{I}' . However, the direction remains consistent when \mathbf{I} corresponds to a benign example. We exploit this property of AGD to detect adversarial examples:

Given a query image \mathbf{I}^q , we compute the following score:

$$\alpha_a = \langle \Delta f^m(\mathbf{I}^q, a), \Delta f^m(\mathbf{I}^p, a) \rangle \quad (4)$$

In Eq. 4, $\langle \cdot, \cdot \rangle$ computes the angular similarity between two vectors. Fig. 2(a) and 2(b) respectively show the distributions of the traditional feature variation score r_l and the proposed score α_a under the same random pixel perturbation μ . Note the significantly less overlap between adversarial and benign example distributions of α_a compared to corresponding distributions of r_l .

To improve the overall detection accuracy, AGDs can be calculated not only for the predicted class a with the highest probability but for each class in the training set. One suggested strategy is to compute AGDs for the top K classes that yield the highest probability, and choose K based on the trade-off between the desired performance and required computation. Eq. 4 is thus extended as follows:

$$\alpha_k = \langle \Delta f^m(\mathbf{I}^q, k), \Delta f^m(\mathbf{I}^p, k) \rangle \quad (5)$$

The detection performance is found to saturate after $K = 4$ (see Table 4) for both CIFAR-10 and ImageNet databases.

Use of predicted class prototype: Because the perturbation applied on an input example is random, a transformed example may not always exhibit the desired property of AGDs used for adversarial example detection. In other words, a transformed example created from an adversarial example may have similar AGD to the adversarial example, while it may have different AGD compared to its benign neighbors.

To handle this anomaly, we use a prototype benign example \mathbf{I}^n that belongs to the predicted class a from a reference database \mathbb{D}' . Recall that AGD of \mathbf{I}^q is jointly determined by target class k and $f^m(\mathbf{I}^q)$. Similarly, let $\Delta f^m(\mathbf{I}^n, k)$ correspond to the prototype benign example \mathbf{I}^n from class a

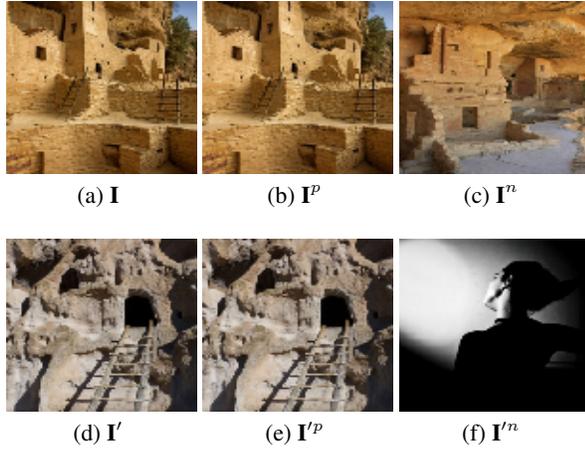


Figure 3: Sample benign and adversarial examples (a) and (d) for ImageNet database. (b) and (e) show the corresponding perturbed (transformed) examples, and (c) and (f) show the corresponding nearest neighbors from the predicted class for (a) and (d).

that is sufficiently close to \mathbf{I}^q on the manifold. In this work, the nearest neighbor of the query example in the reference database \mathbb{D}' based on the embedding distance is selected as the prototype.

To capture the relationship between AGDs of \mathbf{I}^q and \mathbf{I}^n , we define a score β_k :

$$\beta_k = \langle \Delta f^m(\mathbf{I}^q, k), \Delta f^m(\mathbf{I}^n, k) \rangle, \quad (6)$$

such that β_k should be high if $\mathbf{I}^q = \mathbf{I}$ but low if $\mathbf{I}^q = \mathbf{I}'$. This is because an adversarial example is typically crafted by applying perturbation on a query example in the direction of a decision plane relative to the query example. Therefore, if two examples are neighbors in the feature space, their corresponding directions towards a decision plane (defined by k) should be similar. Specifically, because \mathbf{I}^n is a benign example from class a , AGD of \mathbf{I}^n should act a prototype for \mathbf{I}^q and encapsulates the AGD of the local manifold. However, a crafted adversarial example intrinsically belongs to a different class. Therefore, the gradient update path used by the adversarial example generation process is different.

It is straightforward to derive the third score γ_k which measures the similarity between AGD of the transformed example \mathbf{I}^p and \mathbf{I}^n :

$$\gamma_k = \langle \Delta f^m(\mathbf{I}^p, k), \Delta f^m(\mathbf{I}^n, k) \rangle \quad (7)$$

Because $f^m(\cdot)$ is trained to be robust to small perturbations in benign input examples and enforce low intra-class variability, it is expected that $f^m(\mathbf{I}^p)$ and $f^m(\mathbf{I}^n)$, and their corresponding AGDs will be similar to each other if $\mathbf{I}^q = \mathbf{I}$, and quite different if $\mathbf{I}^q = \mathbf{I}'$.

Figure 3 shows benign and adversarial examples and their corresponding neighbors from ImageNet database. The distributions of β_a and γ_a are shown in Fig. 4. Also visualized are the distributions of β_{c_2} and γ_{c_2} when AGDs are computed for the 2nd most probable class (noted as subscript

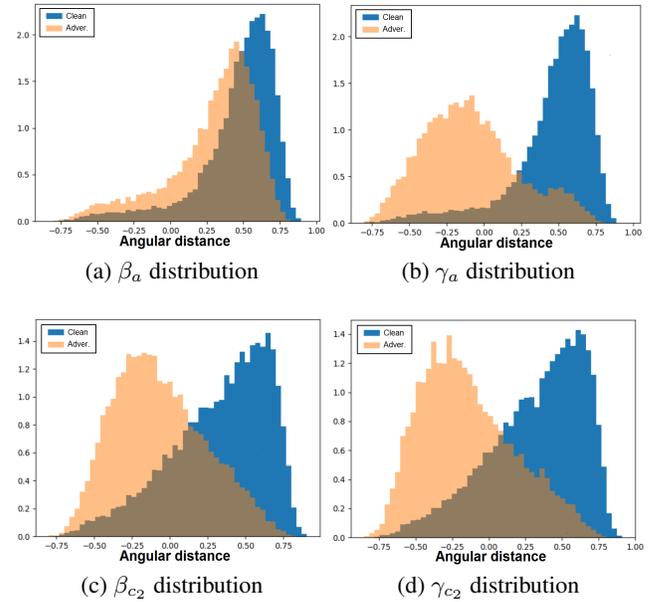


Figure 4: Distributions of $\beta_a, \gamma_a, \beta_{c_2}$, and γ_{c_2} for benign and adversarial examples computed for ImageNet database using the same random transformation.

c_2). Note that the distributions of $\beta_{c_2}, \gamma_a, \gamma_{c_2}$ have significantly less overlap compared to β_a . This is due to overfitting of the adversarial example to target class a resulting in an adversarial example exhibiting similar characteristic AGD properties as its nearest neighbor from class a .

Detection using adversarial gradient directions: Thus far, we discussed the key elements (α_a, β_a , and γ_a) of the proposed adversarial example detector. The performance of the detector primarily depends on the following two observed properties of AGDs:

- (i) **Transformation consistency:** AGDs of adversarial examples change quite significantly after a single random transformation compared to AGDs of benign examples.
- (ii) **Neighborhood smoothness:** Benign examples and the nearest neighbors that belong to the same class share similar AGDs, while adversarial examples share different AGDs with their benign neighbors from the predicted class.

The proposed method is summarized in Algorithm 1. The output τ is obtained using the trained classifier $S^*\{\cdot\}$:

$$\tau(\mathbf{I}^q) = S^*(\{\alpha_a, \beta_a, \gamma_a, \alpha_{c_1}, \beta_{c_1}, \gamma_{c_1}, \dots, \alpha_K, \beta_K, \gamma_K\}) \quad (8)$$

Computational complexity: Instead of L transformations, the proposed method uses a single transformation. Furthermore, the number of classes used in our method is a fixed parameter $K \ll N$; N is the total number of classes. Compared to state-of-the-art methods (e.g., (Roth, Kilcher, and Hofmann 2019)) that use a large L and N and have computational complexity of $O(LN)$, the computational complexity of the proposed method is $O(K)$.

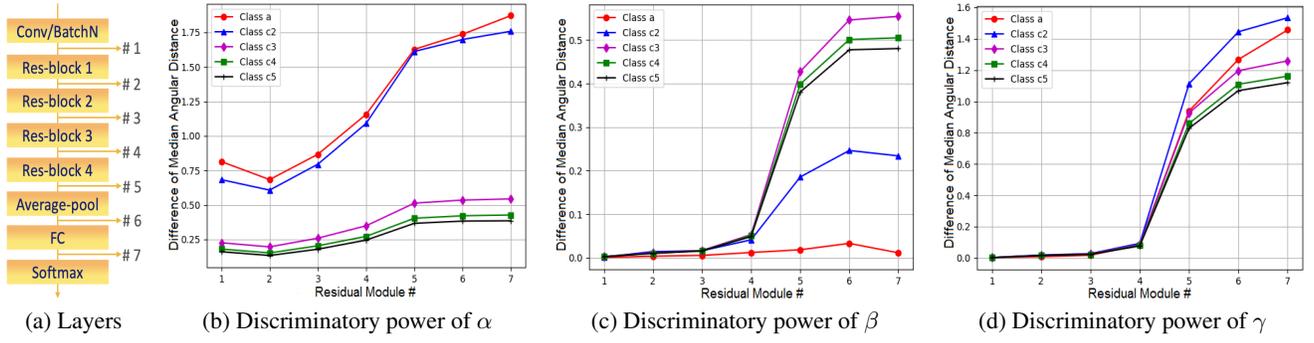


Figure 5: Discriminatory power of (b) α , (c) β , and (d) γ measured using difference of median angular distances between benign and adversarial examples at different ResNet-18 network layers m shown in (a).

Experimental Evaluation

The proposed method is evaluated as a defense mechanism for four widely used deep network architectures, ResNet-18, ResNet-50 (He et al. 2016), GoogleNet (Szegedy et al. 2016), and DenseNet (Huang et al. 2017), against five different state-of-the-art attacks including three gradient-based attacks: FGSM (Szegedy et al. 2014), CW (Carlini and Wagner 2017), PGD (Madry et al. 2018), a decision-based attack: Boundary (Brendel, Rauber, and Bethge 2018), as well as an image deformation-based attack: Adef (Alaifari, Alberti, and Gauksson 2019). Experiments are conducted on two different databases, CIFAR-10 (Krizhevsky 2009) and ImageNet (Krizhevsky, Sutskever, and Hinton 2017). In experiments conducted on CIFAR-10 database, 20,000 images are used to train the baseline residual network, 20,000 are used as reference data, 10,000 are used to generate adversarial examples and learn the parameters of the proposed method, and the remaining 10,000 testing images are used for evaluation. The experiments on ImageNet database follow the protocol in (Krizhevsky, Sutskever, and Hinton 2017). The entire validation set of ImageNet containing 50,000 images is used. 10,000 images are used for neighborhood retrieval, 20,000 to train the model parameters, and the remaining 20,000 for testing. We used the pre-trained network parameters for ResNet-50 and DenseNet-121 on ImageNet. The proposed method is compared with state-of-the-art adversarial example detection methods that are transformation-based (Feature Squeeze (Xu, Evans, and Qi 2018) and (Szegedy et al. 2016)), and nearest neighbor-based (Mahalanobis (Lee et al. 2018) and DkNN (Papernot and McDaniel 2018)). Each method is trained and tested on identical training and testing set partitions so that the reported results are comparable.

Experimental settings: Adversarial examples are generated under the l_∞ norm constraint. The maximum permitted perturbation is set to 8 and 16 pixels, respectively, for CIFAR-10 and ImageNet databases. The step size ϵ_0 is fixed to 0.0013. A single random perturbation μ is permitted on a maximum of three pixels for input transformation. A random forest classifier (30 trees) is used as $S^*\{\cdot\}$. Area Under the Curve (AUC) of the Receiver Operating Characteristics (ROC) curve is used as the evaluation metric. Following the

Algorithm 1: Detection using AGDs

Input: $\mathbf{I}^q, f(\cdot), m, \mathbf{T}, K, \epsilon_0, \mathbb{D}',$ and $S^*(\cdot)$
Output: $\tau(\mathbf{I}^q)$

- 1 Compute the predicted class $a = f(\mathbf{I}^q)$;
- 2 Save top K class indices to $\mathbb{C} = \{a, c_2, \dots, c_k\}$;
- 3 Retrieve \mathbf{I}^n of \mathbf{I}^q from \mathbb{D}' based on a ;
- 4 Obtain a transformed image \mathbf{I}^p with $\mathbf{I}^p = \mathbf{I}^q \star \mathbf{T}$;
- 5 Initialize an empty feature vector $\mathbf{v} = \{\}$;
- 6 **for** $c_k \in \mathbb{C}$ **do**
- 7 Run FGSM: $\mathbf{I}_k^q = \mathbf{I}^q + \epsilon_0 \text{sign}(\nabla \phi(\mathbf{I}^q, c_k))$;
- 8 Run FGSM: $\mathbf{I}_k^p = \mathbf{I}^p + \epsilon_0 \text{sign}(\nabla \phi(\mathbf{I}^p, c_k))$;
- 9 Run FGSM: $\mathbf{I}_k^n = \mathbf{I}^n + \epsilon_0 \text{sign}(\nabla \phi(\mathbf{I}^n, c_k))$;
- 10 $\Delta f^m(\mathbf{I}_k^q, c_k) = f^m(\mathbf{I}_k^q) - f^m(\mathbf{I}^q)$;
- 11 $\Delta f^m(\mathbf{I}_k^p, c_k) = f^m(\mathbf{I}_k^p) - f^m(\mathbf{I}^p)$;
- 12 $\Delta f^m(\mathbf{I}_k^n, c_k) = f^m(\mathbf{I}_k^n) - f^m(\mathbf{I}^n)$;
- 13 $\alpha_k = \langle \Delta f^m(\mathbf{I}_k^q, c_k), \Delta f^m(\mathbf{I}_k^p, c_k) \rangle$;
- 14 $\beta_k = \langle \Delta f^m(\mathbf{I}_k^q, c_k), \Delta f^m(\mathbf{I}_k^n, c_k) \rangle$;
- 15 $\gamma_k = \langle \Delta f^m(\mathbf{I}_k^p, c_k), \Delta f^m(\mathbf{I}_k^n, c_k) \rangle$;
- 16 $\mathbf{v} = \mathbf{v} \cup \{\alpha_k, \beta_k, \gamma_k\}$
- 17 $\tau(\mathbf{I}^q) = S^*(\mathbf{v})$.

protocol in (Lee et al. 2018), examples that are: (i) classified correctly, and (ii) can be perturbed to successfully attack the deep network, are used for training and testing.

Parameter Analysis

Network layers used for detection: The goal of this experiment is to determine the deep network layer(s) m that are useful for computing AGDs. Experiments are conducted on CIFAR-10 database using the ResNet-18 architecture and ImageNet database using DenseNet-121 architecture to detect the FGSM attack. The difference between median angular similarities of AGDs, α_k, β_k and γ_k , of benign and adversarial example distributions are computed after each residual module in the ResNet-18 pipeline (see Fig. 5). It is observed that adversarial and benign example distributions are comparatively separable after residual block #5. Given this, α_k, β_k , and γ_k are computed for the embedding (layer output #6) and logit layer (layer output #7), and combined

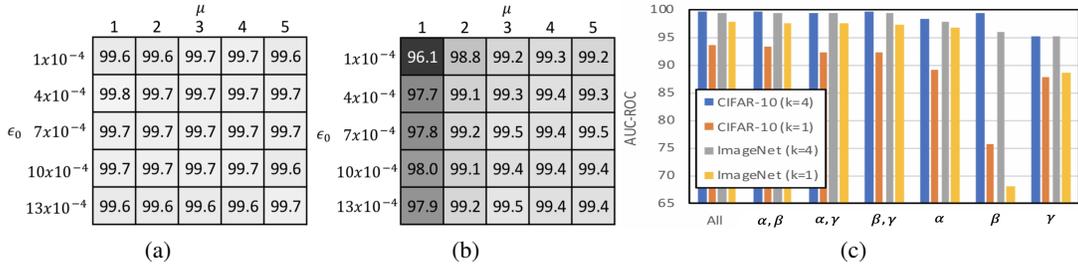


Figure 6: Adversarial example detection performance (AUC-ROC (%)) on (a) CIFAR-10 and (b) ImageNet database for different combinations of random perturbation μ and step size ϵ_0 . Shown in (c) is the detection performance on the two databases for different combinations of the proposed angular similarity scores α, β , and γ for predicted classes under $K = 1$ and $K = 4$.

Class number (K)	1	2	3	4	5
CIFAR-10	94.1	97.5	99.7	99.8	99.8
ImageNet	98.2	99.2	99.4	99.5	99.5

Table 1: Adversarial example detection performance (AUC-ROC in %) for top K predicted classes.

for the final decision. Note that the discriminatory power of α_k is maximal for the top two most confident classes, while that of β_k and γ_k is maximal for the 3rd and 2nd most confident class, respectively. This observation indicates that it is necessary to take into account the results of AGDs for multiple classes to improve the overall detection performance.

AGD parameters: The objective of this experiment is to determine the optimal combination of parameters that maximizes the overall detection performance. The parameters include (i) number of predicted classes K , (ii) added random perturbation μ , and (iii) the step size ϵ_0 . A parameter search is performed on the parameter space with K ranging from one to five, and μ and ϵ_0 as shown in Fig. 6. Parameter values that minimize the required perturbation on an input example (for imperceptibility) are considered optimal.

Table 1 shows the detection performance for different values of K . The performance saturates after $K = 4$. For $K = 4$, the best performing combinations of μ and ϵ_0 are shown in Figs. 6 (a) and 6 (b).

Number of random transformations: To measure the impact of the number of random transformations on our method, different number of random transformations (upto five) are used to detect the FGSM attack. The angular similarity scores for the selected transformations are concatenated and fed to the classifier in Eq. 8.

Table 2 shows that only a slight increase in detection performance is obtained using more than one random transformation. Hence, a single random transformation is used in other experiments.

Ablation Study

The objective of this experiment is to determine the impact of each angular similarity score α_k, β_k , and γ_k on the final performance. Experiments are conducted for top K predicted classes, $K = 1$ and $K = 4$, and the obtained per-

Random trans. number	1	2	3	4	5
CIFAR-10	99.6	99.7	99.7	99.7	99.7
ImageNet	99.5	99.5	99.6	99.7	99.7

Table 2: Adversarial example detection performance (AUC-ROC in %) for different number of random transformations.

formance for different score combinations is visualized in Fig. 6 (c).

Fig. 6 (c) shows that the performance obtained using any combination of two scores is lower than the performance obtained by aggregating all three scores. Using only one similarity score is not recommended since it may not appropriately capture either of the two AGD properties.

Comparison With State-Of-The-Art

Comparison of the proposed method with state-of-the-art adversarial example detection methods is shown in Table 3. For transformation-based methods, we compare the proposed method to (Hu et al. 2019) and (Xu, Evans, and Qi 2018). These two methods use a series of transformations to transform the input example. For a fair comparison with our method, we limit the comparison to two fundamental but most effective transformations. The first is one-time Gaussian random perturbation used in the first stage of (Hu et al. 2019) (denoted as Rand-1) and the second is the 2×2 median filter used in Xu *et al.* (Xu, Evans, and Qi 2018) (denoted as Median). The l_1 score used in these methods is used for benchmarking. For neighbor-based defenses, we benchmark against DkNN (Papernot and McDaniel 2018) and MA (Lee et al. 2018). Each of these methods uses over 20 reference examples to model benign example distribution for adversarial example detection. The parameters of each method are optimized to maximize performance on CIFAR-10 and ImageNet databases. The performance is reported in two different settings: (i) training and testing with the same attack algorithm, (ii) training on adversarial samples generated by FGSM and testing on other attacks. The results shown in Table 3 indicate that the proposed approach significantly outperforms state-of-the-art methods on CIFAR and ImageNet databases. The proposed method also scales reasonably well when the number of classes increase from 10 (CIFAR-10) to

Network (Dataset)	Method	FGSM	ADef.	CW	PGD	Boundary	ADef.	CW	PGD	Boundary
ResNet-18 (CIFAR-10)	Rand-1	78.8	82.8	88.8	92.0	74.3	78.7	88.7	92.0	74.3
	Median	91.7	86.5	91.6	93.0	92.2	86.4	91.6	93.0	92.1
	MA.	96.8	94.0	94.1	96.4	98.3	93.9	94.1	96.4	98.4
	DkNN	99.2	93.7	93.4	99.3	99.5	93.7	93.4	99.3	99.5
	Ours	99.6	94.7	96.4	99.7	99.6	90.3	93.1	99.7	99.4
GoogleNet (CIFAR-10)	Rand-1	58.4	53.1	52.5	51.7	55.7	58.4	52.5	51.7	55.7
	Median	60.0	54.4	52.8	52.0	74.3	54.4	52.8	52.0	74.3
	MA.	95.8	94.1	91.4	95.1	97.9	94.1	91.4	95.1	94.8
	DkNN	99.0	97.1	90.4	99.0	99.1	97.1	90.4	99.0	99.1
	Ours	98.0	97.2	98.2	98.5	96.6	96.9	98.1	98.4	92.1
ResNet-50 (ImageNet)	Rand-1	50.4	51.4	54.1	69.3	50.0	50.4	54.1	69.3	50.0
	Median	85.7	87.2	91.2	89.0	86.7	87.2	91.2	89.0	86.7
	MA.	58.4	58.0	58.9	56.3	77.9	58.0	55.8	56.5	77.4
	DkNN	67.2	63.7	61.9	62.7	81.6	63.7	61.9	62.7	81.6
	Ours	99.9	99.9	99.7	99.9	99.8	99.9	99.6	99.9	99.8
DenseNet-121 (ImageNet)	Rand-1	50.1	51.0	53.5	66.3	50.0	50.1	53.5	66.3	50.0
	Median	86.2	83.3	90.0	86.7	85.5	83.3	90.0	87.7	85.5
	MA.	72.3	70.3	70.8	71.4	70.7	71.4	71.7	71.4	44.3
	DkNN	63.7	60.6	59.4	60.3	80.0	60.6	59.4	60.3	80.0
	Ours	99.5	99.1	91.0	99.5	98.1	97.4	83.7	99.5	96.1

Table 3: Comparison of the proposed adversarial example detection method with state-of-the-art methods on the ImageNet database. Performance is reported using area under curve (AUC) (%) of the ROC curve. The top performing algorithms are highlighted. (L) Training and testing on the same attack, and (R) training using FGSM attack and testing on other attacks.

1,000 (ImageNet).

White-box attack setting: In the earlier experiments (gray-box attack setting), it was assumed that the attacker has no knowledge of the adversarial detection approach used. In this experiment, it is assumed that the attacker has complete knowledge of the adversarial detection method. The adaptive attack setting described in (Hu et al. 2019) is used. A subset of 1,000 images from ImageNet database are selected. PGD attack is used to jointly optimize $\lambda\phi(f(\mathbf{I}), y)$ and maximize all similarity scores in $\{\alpha_a, \beta_a, \gamma_a, \alpha_{c_2}, \beta_{c_2}, \gamma_{c_2}, \dots, \alpha_k, \beta_k, \gamma_k\}$, simultaneously. The parameter λ is set to two. The attack success rate which indicates if an adversarial example is able to fool the deep network, and detection AUC-ROC for $K = \{1, 4\}$ are reported in Table 4. The performance is compared with Rand-1, where $r = \|f^m(\mathbf{I}^p) - f^m(\mathbf{I}^q)\|_1$ and $\lambda\phi(f(\mathbf{I}), y)$ are jointly minimized. The maximum permitted perturbation l_∞ is set to 0.1 of pixel value range, and the step size for PGD is set to 100. The results for two different learning rates, 0.1 and 0.03, are compared. Experimental results show that our method significantly reduces the attack success rate when $\lambda\phi(f(\mathbf{I}), y)$ and detection scores are optimized jointly. For $K = 4$, 64.1% AUC-ROC is obtained for our method while

Target	Attack Success Rate		Detection AUC	
	LR=0.1	LR=0.03	LR=0.1	LR=0.03
Rand-1	79.4	78.0	53.0	53.1
Ours (K=1)	74.9	70.9	53.3	60.8
Ours (K=4)	53.5	60.3	64.1	62.5

Table 4: White-box attack performance (AUC-ROC in %). LR denotes the learning rate used while conducting the attack.

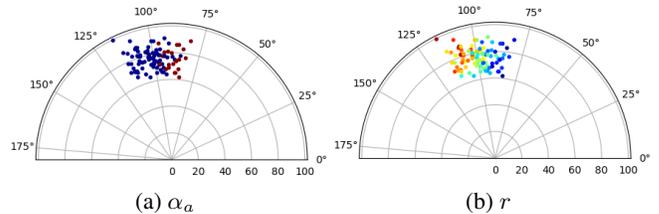


Figure 7: Variations in detection scores obtained for 100 different random transformations on an adversarial example. Scores α_a generated by our method are more consistent than traditional l_1 -based scores r .

Rand-1 only performs slightly better than random (50% AUC-ROC). In summary, the proposed AGD-based solution outperforms the traditional l_1 score-based solution (Rand-1) in identical white box attack settings.

Detection visualization: To investigate the consistency of the proposed score α_a is compared to the detection score r used in Rand-1, we visualize their variations for 100 different random transformations on a random adversarial example in Fig. 7. The color of dots correspond to min-max normalized score values. The angle is indicative of the cosine distance between vectors $f^m(\mathbf{I}^p) - f^m(\mathbf{I}^q)$ and $\Delta f^m(\mathbf{I}^p, a)$. The distance of each dot from the origin corresponds to the magnitude of vector $\Delta f^m(\mathbf{I}^p, a)$. The plot shows that the proposed scores: α_a is more consistent than $r = \|f^m(\mathbf{I}^p) - f^m(\mathbf{I}^q)\|_1$ under multiple random transformations, and the majority of the samples have low similarities.

Conclusions

We propose the use of adversarial gradient directions for adversarial example detection. The proposed approach uses a single transformation of the input example and a single example of the predicted class from a reference database. Despite its simplicity, the proposed method has significant discriminative power and outperforms existing state-of-the-art adversarial example detection methods on CIFAR-10 and ImageNet databases in both gray-box and white-box settings. We encourage the research community to further explore the utility of adversarial gradient directions for adversarial example detection.

Acknowledgements

The authors would like to thank Dr. Yizhen Wang for his feedback on an earlier version of this paper.

References

- Alaifari, R.; Alberti, G. S.; and Gauksson, T. 2019. ADef: an Iterative Algorithm to Construct Adversarial Deformations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.
- Arik, S. Ö.; and Pfister, T. 2019. Attention-Based Prototypical Learning Towards Interpretable, Confident and Robust Deep Neural Networks. *CoRR* abs/1902.06292.
- Brendel, W.; Rauber, J.; and Bethge, M. 2018. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.
- Carlini, N.; and Wagner, D. A. 2017. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, 39–57.
- Feinman, R.; Curtin, R. R.; Shintre, S.; and Gardner, A. B. 2017. Detecting Adversarial Samples from Artifacts. *CoRR* abs/1703.00410.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 770–778.
- Hu, S.; Yu, T.; Guo, C.; Chao, W.; and Weinberger, K. Q. 2019. A New Defense Against Adversarial Images: Turning a Weakness into a Strength. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d’Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, 1633–1644.
- Huang, G.; Liu, Z.; van der Maaten, L.; and Weinberger, K. Q. 2017. Densely Connected Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2261–2269.
- Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. In *Technical report*. Department of Computer Science, University of Toronto.
- Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. ImageNet classification with deep convolutional neural networks. *Commun. ACM* 60(6): 84–90.
- Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2017. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*.
- Lee, K.; Lee, K.; Lee, H.; and Shin, J. 2018. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. In *Advances in Neural Information Processing Systems, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, 7167–7177.
- Li, X.; and Li, F. 2017. Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 5775–5783.
- Ma, X.; Li, B.; Wang, Y.; Erfani, S. M.; Wijewickrema, S. N. R.; Schoenebeck, G.; Song, D.; Houle, M. E.; and Bailey, J. 2018. Characterizing Adversarial Subspaces Using Local Intrinsic Dimensionality. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.
- Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.
- Papernot, N.; and McDaniel, P. D. 2018. Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning. *CoRR* abs/1803.04765.
- Raff, E.; Sylvester, J.; Forsyth, S.; and McLean, M. 2019. Barrage of Random Transforms for Adversarially Robust Defense. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 6528–6537.
- Roth, K.; Kilcher, Y.; and Hofmann, T. 2019. The Odds are Odd: A Statistical Test for Detecting Adversarial Examples. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 5498–5507. PMLR.
- Snell, J.; Swersky, K.; and Zemel, R. S. 2017. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems*, 4077–4087.
- Svoboda, J.; Masci, J.; Monti, F.; Bronstein, M. M.; and Guibas, L. J. 2019. PeerNets: Exploiting Peer Wisdom Against Adversarial Attacks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.
- Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2818–2826. IEEE Computer Society.
- Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.
- Tian, S.; Yang, G.; and Cai, Y. 2018. Detecting Adversarial Examples Through Image Transformation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*, 4139–4146.
- Wang, J.; Sun, J.; Zhang, P.; and Wang, X. 2018. Detecting Adversarial Samples for Deep Neural Networks through Mutation Testing. *CoRR* abs/1805.05010.

Xu, W.; Evans, D.; and Qi, Y. 2018. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society.

Yang, Y.; Zhang, G.; Xu, Z.; and Katabi, D. 2019. ME-Net: Towards Effective Adversarial Robustness with Matrix Estimation. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 7025–7034. PMLR.