Identifying Untrustworthy Predictions in Neural Networks by Geometric Gradient Analysis*

Leo Schwinn[1] An Nguyen[1] René Raab[1] Leon Bungert[2] Daniel Tenbrinck[2] Dario Zanca[1] Martin Burger[2] Bjoern Eskofier[1]

[1]Department Artificial Intelligence in Biomedical Engineering, Univ. of Erlangen-Nürnberg, Germany
[2]Department Mathematics, Univ. of Erlangen-Nürnberg, Germany

Abstract

The susceptibility of deep neural networks to untrustworthy predictions, including out-of-distribution (OOD) data and adversarial examples, still prevent their widespread use in safety-critical applications. Most existing methods either require a retraining of a given model to achieve robust identification of adversarial attacks or are limited to out-of-distribution sample detection only. In this work, we propose a geometric gradient analysis (GGA) to improve the identification of untrustworthy predictions without retraining of a given model. GGA analyzes the geometry of the loss landscape of neural networks based on the saliency maps of their respective input. We observe considerable differences between the input gradient geometry of trustworthy and untrustworthy predictions. Using these differences, GGA outperforms prior approaches in detecting OOD data and adversarial attacks, including state-of-the-art and adaptive attacks.

1 INTRODUCTION

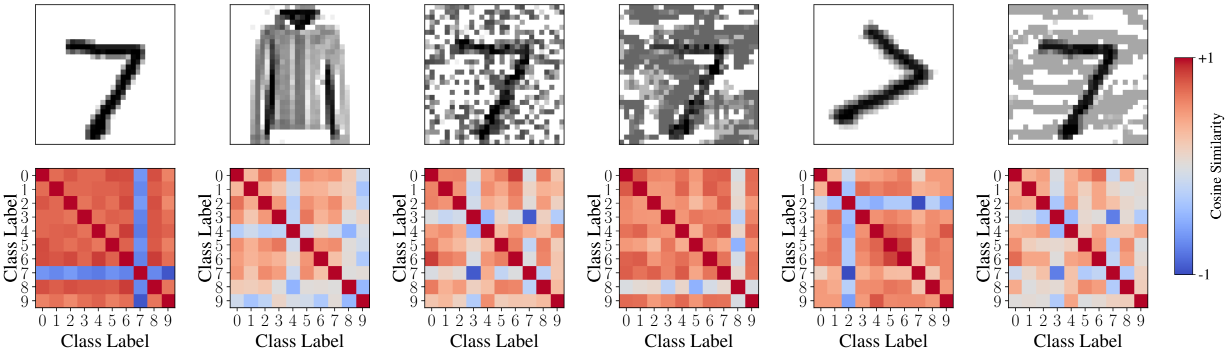
Deep neural networks (DNNs) are known to achieve remarkable results when the distributions of the training and test data are similar. However, this assumption is often violated in real-world scenarios where so-called out-of-distribution (OOD) data may be observed which are not covered by the training set. DNNs have been shown to make high-confidence predictions for OOD data even if it does not contain any semantic information, e.g., randomly generated noise [Hendrycks and Gimpel, 2017]. This behavior can lead to fatal outcomes in safety-critical applications, such as autonomous driving, where the algorithm might fail to call for human intervention when it is confronted with OOD data. In addition to the overconfidence of DNNs, it is widely recognized that most DNNs are vulnerable to imperceptible input perturbations called adversarial examples [Goodfellow et al., 2015, Madry et al., 2018]. These perturbations can lead to incorrect predictions by the neural network and therefore pose an additional security risk. Many approaches have been proposed to make neural networks more robust in terms of adversarial examples [Goodfellow et al., 2015, Madry et al., 2018, Gowal et al., 2020]. Nevertheless, there is still a wide gap between the accuracy on unperturbed data and adversarial examples.

An alternative to training robust DNNs is the early detection of attacks [Lee et al., 2018, Chen et al., 2020]. Identified attacks can then be forwarded for further human assessment. One line of research investigates geometric properties of neural networks in the input space to explain their classification decisions and detect adversarial attacks. Fawzi et al. [2017] demonstrate that the decision boundaries of neural networks are mostly flat around the training data and only show considerable curvature in very few directions. Jetley et al. [2018] illustrate that these high-curvature directions are mainly responsible for the final classification decision and thus can be exploited by adversarial attacks to induce misclassifications. However, Fawzi et al. [2017] only focus on detecting small adversarial perturbations and Jetley et al. [2018] restrict themselves to the theoretical analysis of the loss landscape.

In this work, we focus on the detection of two major problems of DNNs, namely OOD data and adversarial attacks. We propose a novel methodology inspired by the analysis of geometric properties in the input space of neural networks, which we name geometric gradient analysis (GGA). Here, we analyze and interpret the gradient of a neural network w.r.t. its input, in the following referred to as *saliency map*. More precisely, for a given input sample, we inspect the geometric relation among all possible saliency maps, which are calculated for each output class of the model. This is

[*] Code is available at: `https://github.com/mad-lab-fau/GGA-Identifying-Untrustworthy-Predictions`.

Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI 2021), PMLR 161:854–864.



(a) Clean (b) OOD (c) Noise (d) PGD (e) Rotation (f) B&B

Figure 1: Input samples of a neural network in the top row and the respective cosine similarity matrices in the bottom row (matrices which contain the pairwise cosine similarity between the saliency map w.r.t. every class-specific logit). Clean samples show high average cosine similarity (red) between the saliency maps of non-predicted classes, while untrustworthy samples (i.e., noisy images, adversarial examples, outliers) can be detected by saliency maps that are less aligned between non-predicted-classes (blue).

achieved by a pairwise calculation of the cosine similarity between saliency maps. GGA can be used with any pre-trained differentiable neural network and does not require any retraining of the model. Figure 1 shows input samples of a neural network in the top row and the respective cosine similarities between the saliency maps of every output class in the bottom row. Figure 1a exemplifies that if an input is correctly classified by the model, the saliency map of the predicted class (i.e., the digit 7) generally points in a direction that is opposite to the saliency maps of all other classes. This results in low average cosine similarity between these saliency maps in the rows and columns of class label 7 (blue-colored squares). Accordingly, the saliency maps of the other classes mostly align and display a high average cosine similarity (red-colored squares). In contrast, in the case of an OOD sample or adversarial attack, the saliency maps of the non-predicted classes point towards different directions and the cosine similarity is considerably lower on average.

The contributions of this paper can be summarized as follows. First, we demonstrate that the gradient of only the predicted class is not sufficient for the detection of untrustworthy predictions and thereby motivate to analyze the saliency maps of all possible output classes simultaneously. Subsequently, we demonstrate that for common OOD tasks, GGA is highly competitive compared to prior methods. Furthermore, we demonstrate that GGA successfully identifies a diverse variety of adversarial attacks and show that the geometric relation between gradients is difficult to compromise with adaptive attacks. Finally, we show that the computational overhead of GGA can be substantially reduced without considerable decreases in the detection performance.

2 RELATED WORK

Our proposed method combines ideas from several areas of neural network research. This includes out-of-distribution detection, adversarial attack detection, and model saliency. In this section, we briefly review prior work in these research areas.

Previous works have established that softmax-based neural networks tend to make overconfident predictions in the presence of misclassifications, OOD data, and adversarial attacks [Hendrycks and Gimpel, 2017, Liang et al., 2018, Jiang et al., 2018, Corbière et al., 2019]. Hendrycks and Gimpel [2017] propose a baseline method for detecting OOD data that utilizes the softmax output of a neural network. Depending on a pre-defined threshold based on the softmax score, they define samples as either in- or out-of-distribution. Liang et al. [2018] further enhance this baseline. They apply temperature scaling to the softmax scores and additionally add small perturbations to the input to increase the difference between in- and out-of-distribution samples. While both approaches have been shown to work on OOD data, they fail in the presence of adversarial attacks [Chen et al., 2020].

Lee et al. [2018] evaluate their detection framework both on OOD and adversarial samples. They calculate class-conditional Gaussian distributions from the pre-trained networks and discriminate samples based on the Mahalanobis distance between the distributions. Chen et al. [2020] proposed a combined framework for detecting OOD data and adversarial attacks as robust out-of-distribution (ROOD) detection. They extend the threat model to attacks on OOD data that aim to fool the adversarial detector as well as the classification model. They augment the training data of neural networks with both perturbed inlier and outlier data and demonstrate improved robustness compared to prior methods. Nevertheless, both methods require adversarial examples to train the respective detector.

Another line of research has found that as neural networks become more robust, the interpretability of their saliency maps increases [Tsipras et al., 2019, Etmann et al., 2019]. Gu and Tresp [2019] propose enhanced Guided Backpropagation and show that the classifications of adversarial images can be explained by saliency-based methods. Ye et al. [2020] demonstrate that the saliency maps of adversarial and benign examples exhibit different properties and utilize this behavior to detect adversarial attacks. However, Dombrowski et al. [2019] observe that explanation-based methods can be manipulated by adversarial attacks as well, which limits the robustness of these methods.

3 GEOMETRIC GRADIENT ANALYSIS

In this section, we first introduce the necessary mathematical notation and describe the proposed geometric gradient analysis (GGA) method. Then, necessary and sufficient conditions for local minima in the loss function using non-local gradient information are given to further motivate the geometrical gradient analysis.

Let (x, y) be a pair consisting of an input sample $x \in \mathbb{R}^d$ and its corresponding class label $y \in \{1, \dots, C\}$ in a supervised classification task. We denote by F_θ a neural network parametrized by the parameter vector $\theta \in \Theta$, and by $\hat{k}$ the class predicted by the neural network for a given sample x. We define $\mathcal{L}(F_\theta(x), y)$ as the loss function of the neural network. The GGA method can be summarized as follows. We first define $s_i(x) \in \mathbb{R}^d$ as the saliency map of the i-th class for a given sample x as

$$s_i(x) = \operatorname{sgn}\left(\nabla_x \mathcal{L}\left(F_\theta(x), i\right)\right), \qquad (1)$$

where sgn indicates the element-wise sign operation. As common for adversarial attacks [Goodfellow et al., 2015, Madry et al., 2018] we use the sign of the gradient instead of utilizing the gradient directly. This has shown to be effective for approximating the direction which will maximize the loss w.r.t. the respective class [Goodfellow et al., 2015, Madry et al., 2018] and has been more effective for GGA as well in our experiments. Omitting the dependency on x, the cosine similarity matrix CSM $\equiv$ CSM(x), for a given sample x is defined as

$$\text{CSM} = (c_{ij}) \in \mathbb{R}^{C \times C}, c_{ij} = \frac{s_i \cdot s_j}{|s_i||s_j|} \qquad (2)$$

where $i, j \in \{1, ..., C\}$ and c_{ij} represent the cosine similarity between the two saliency maps s_i and s_j. In contrast to previous methods, which rely solely on the saliency w.r.t. the predicted class, GGA takes into account the geometric properties between the saliency maps of all possible output classes. Considering multiple saliency maps simultaneously makes GGA more difficult to attack. To fool the trained neural network as well as the GGA detector, an attacker must cause a misclassification while simultaneously retaining the geometric properties between the saliency maps of all output classes. We observe that for clean samples, the saliency maps of non-predicted classes s_i, s_j, $i, j \neq \hat{k}$, all point in a similar direction and therefore show a high average cosine similarity. Simultaneously, the saliency map of the predicted class and the saliency maps of the non-predicted classes point in opposite directions and show a strongly negative cosine similarity. In contrast, adversarially attacked and OOD samples show less alignment between the different saliency maps and thus a lower average cosine similarity and more variance between the saliency maps. These observations are in line with prior work from Jetley et al. [2018], that showed that samples from the same class can be associated with specific directions in the input space of neural networks. We argue that samples that belong to the same distribution as the training data lie on these class-specific directions with high curvature. Thus, gradients at clean samples point either in the same or opposite of these class-specific directions. In contrast, samples that do not belong to the distribution of the training data deviate from these directions and show different properties. The described behavior of CSMs is exemplified in Figures 1 and 2.

The calculated CSMs can be used to differentiate between trustworthy and untrustworthy predictions in a classification pipeline. This could be done, for example, by training a classifier on CSMs from types of data (e.g., clean data, adversarial examples, OOD samples). To demonstrate that the CSMs can be used without training another neural network and without prior knowledge on the outlier data, we chose a simpler approach in this paper. We extract simple features from the CSMs and use them with a standard outlier detector to identify untrustworthy samples. Further, we only train the outlier detector on CSMs from clean data, which reduces the computational overhead. More details are given in Section 4.3.3.

4 EXPERIMENTS

In this section, we describe methods, evaluation metrics, data sets, and other settings of our experiments.

4.1 EXISTING METHODS

We compare GGA to two other methods, which also do not necessarily require any retraining of the neural network and do not utilize adversarial examples to train the outlier/adversarial detector. Namely, we consider the method proposed in [Hendrycks and Gimpel, 2017] (called *Baseline* in the following) and the *ODIN method* [Liang et al., 2018]. For the ODIN method we set the temperature scaling parameter $T = 1000$ and the perturbation bounds ϵ for the CIFAR10 ($\epsilon = 0.0014$) and CIFAR100 ($\epsilon = 0.002$) data sets were taken from the original paper. We found the best values for the MNIST, Fashion-MNIST, SVHN, and UCR ECG data set to be $\epsilon = 0.0014$, $\epsilon = 0.0014$, $\epsilon = 0.002$, and $\epsilon = 0.002$, respectively. We additionally consider the method proposed by Lee et al. [2018] (called *Maha* in the following), which requires the detector to be trained with adversarial examples. In this case, we directly used the implementation provided by the authors Lee et al. [2018].

4.2 EVALUATION METRICS

We use common evaluation metrics for the assessment of the OOD detection methods [Hendrycks and Gimpel, 2017, Liang et al., 2018]. The metrics are described below:

TNR at (95% **TPR):** This evaluation metric describes the true negative rate (TNR) for negative examples (i.e., adversarial attacks or OOD data) for a true positive rate (TPR) of 95% (i.e., clean data).

AUROC: The area under the receiver operating characteristic curve is a threshold independent metric [Davis and Goadrich, 2006]. It describes the relationship between false positive rate (FPR) and TPR and can be calculated by integrating over the ROC curve. The AUROC value can be interpreted as the probability that a positive example exhibits a higher detector score than a negative example [Fawcett, 2006].

AUPR: The area under precision-recall is also a threshold independent metric [Manning and Schütze, 1999]. In contrast to the AUROC, it adjusts for the frequency of the different classes and is thereby suitable for imbalanced problems [Hendrycks and Gimpel, 2017]. The AUPR can be calculated either with trustworthy predictions as the positive class (AUPR-IN) or with untrustworthy predictions as the positive class (AUPR-Out).

The best possible score for all described metrics is 100%.

4.3 SETUP

In the following, we give an overview of general hyperparameters used for the performed experiments. This includes a description of the data sets, neural network models, and the features we extract from the CSMs. Finally, we describe the threat model of the adversarial attacks.

4.3.1 Data and Architectures

We split each data set into predefined training and testing sets. Additionally, we used 10% of the training data as the validation set for self-trained models. All self-trained models were trained by minimizing the cross-entropy loss using SGD with Nesterov momentum (0.9) and a batch size of 128. We used a step-wise learning rate schedule that divides the learning rate by five at 30%, 60%, and 80% of the total training epochs. The following classification data sets were used to evaluate the proposed method.

MNIST [LeCun et al., 1998] consists of greyscale images of handwritten digits each of size $28 \times 28 \times 1$ ($60,000$ training and $10,000$ test) and is a common benchmark for outlier detection and adversarial robustness. We trained a basic CNN architecture as in prior work [Madry et al., 2018]. This architecture consists of four convolutional layers with 32, 64, and 128 filters and two fully-connected layers with 100 and 10 output units. We used ReLU for the activation functions between each layer. We used a learning rate of 0.1 and trained for 10 epochs, where the validation accuracy converged.

CIFAR10 [Krizhevsky, 2009] consists of RGB color images, each of size $32 \times 32 \times 3$, with 10 different labels ($50,000$ training and $10,000$ test). For CIFAR10 we used a ResNet56 [He et al., 2016]. All images from the CIFAR10 data set were standardized and random cropping and horizontal flipping were used for data augmentation during training as in [He et al., 2016].

CIFAR100 [Krizhevsky, 2009] has the same properties as CIFAR10 but is considerably more difficult as it contains 100 instead of only 10 classes. For CIFAR100, we used a pre-trained PreResNet164 [Xichen, 2019] and otherwise the same configurations as for CIFAR10.

UCR ECG (ID 49) [Dau et al., 2018] is a time series classification data set with 42 different classes (1800 training and 1965 test). It contains non-invasive electrocardiogram (ECG) recordings of fetuses with a length of 750 time steps each. We consider this data set in addition to the computer vision data sets for a basic benchmark of the proposed GGA method on time series classification tasks. We trained a basic CNN architecture consisting of three convolutional layers with 128, 256, and 128 filters and one fully-connected layer with 42 output units. We used batch normalization and a ReLU activation function between each layer. We used a learning rate of 0.01 and trained for 100 epochs.

4.3.2 Out-Of-Distribution Data Sets

We consider the respective test set of the training data as in-distribution data and suitable, realistic images from other data sets as OOD. Additionally, we create two synthetic noise data sets as OOD data for every data set as done in prior work [Hendrycks and Gimpel, 2017, Liang et al., 2018]. In the following list, the respective in-distribution data sets are put in brackets after the OOD data sets:

Fashion-MNIST (OOD for MNIST): [Xiao et al., 2017] consists of greyscale images of 10 different types of clothing, each of size $28 \times 28 \times 1$ ($10,000$ test).

SVHN (OOD for CIFAR10, CIFAR100): [Netzer et al., 2011] consists of color images of 10 different types of street view digit, each of size $32 \times 32 \times 3$ ($26,032$ test).

Uniform Noise (OOD for all data sets): the uniform noise data set consists of $10,000$ images, where each pixel value is drawn i.i.d. from a uniform distribution in $[0, 1]$.

Gaussian Noise (OOD for all data sets): the Gaussian noise data set consists of $10,000$ images, where each pixel value is drawn i.i.d. from a Gaussian distribution with unit variance.

4.3.3 Geometric Gradient Analysis Features and Prediction

To identify untrustworthy predictions with the GGA method, we first generate the respective CSM for a given sample x. Then, we compute simple features from the CSMs and use them for training a simple outlier detector. Let $\hat{k}$ be the index associated with the class predicted by the neural network F_θ for a given sample x. By exploiting the symmetry of the cosine similarity matrix CSM, and observing that the elements of the main diagonal are all equal to 1, we can restrict the analysis to the set S to the elements above the main diagonal, i.e., $S = \{c_{ij}\}_{i<j}$. We compute five basic statistical features (*mean, maximum, minimum, standard deviation, and energy*) separately for two different sets $S_1 = S \cap \{c_{ij}\}_{i,j \neq \hat{k}}$ and $S_2 = S \cap \{c_{ij}\}_{i=\hat{k} \vee j=\hat{k}}$. These statistics constitute the ten features f_{1-10} provided to the outlier detection model.

Figure 2 shows how the mean value of S_1 of a CSM can be used to differentiate between several data classes on the MNIST data set that are not discriminated by the softmax score alone. We exclude the softmax score from the GGA features for better comparison between the methods. For practical applications, the softmax score can be used as an additional feature. For all the remaining detection tasks, we train a lightweight on-line detector of anomalies (LODA) [Pevný, 2016] with the GGA features of the correctly classified samples of the training set. We chose LODA as it is designed to handle a large number of data points and does not add noticeable computational overhead to the classification pipeline. For LODA, we set the number of random cuts to 100 for all experiments. The number of random bins was set to 500 for the CIFAR data sets and 100 for MNIST and UCR ECG after an evaluation on the validation set.

4.3.4 Threat Model

Let $\delta \in \mathbb{R}^d$ be an adversarial perturbation. We use a variety of adversarial attacks with different attributes to generate untrustworthy predictions. We only consider successful adversarial attacks that change the classification result as untrustworthy and discard unsuccessful attacks. We employ attacks with different norm constraints (ℓ_2, ℓ_∞) such that the adversarial perturbation is smaller than some predefined perturbation budget $||\delta||_p \leq \epsilon$. We set the perturbation budget ϵ in the ℓ_∞-norm to $0.3, 8/255, 8/255, 0.1$ for MNIST, CIFAR10, CIFAR100, and UCR ECG, respectively, as in prior work [Madry et al., 2018, Fawaz et al., 2019]. For attacks in the ℓ_2-norm we multiply the allowed perturbation strength by 10. Furthermore, we use attacks that produce high and low confidence predictions as low confidence adversarial examples have shown to be effective against several detectors [Chen et al., 2020]. To create high confidence misclassifications we use Projected Gradient Descent (PGD) [Madry et al., 2018]. PGD is an iterative attack that tries to maximize the loss w.r.t. the original class and subsequently creates perturbations, which lead to wrong predictions with high certainty. For the PGD attack we used a step size of $\alpha = \frac{\epsilon}{4}$ and 70 attack iterations which lead to a success rate of 100% for all models. To create low confidence predictions we employ the B&B attack [Brendel et al., 2019], which creates perturbations at the decision boundary of the attack. For B&B we used a learning rate of 0.001 and 100 iterations. Finally, we consider random rotations between -45 and 45 degrees and uniform noise attacks in the ϵ-ball for non-gradient-based attacks (rotations are naturally omitted for the time-series classification task).

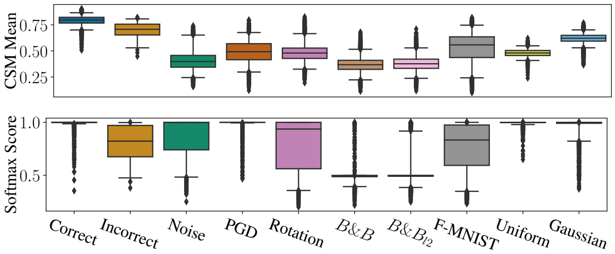
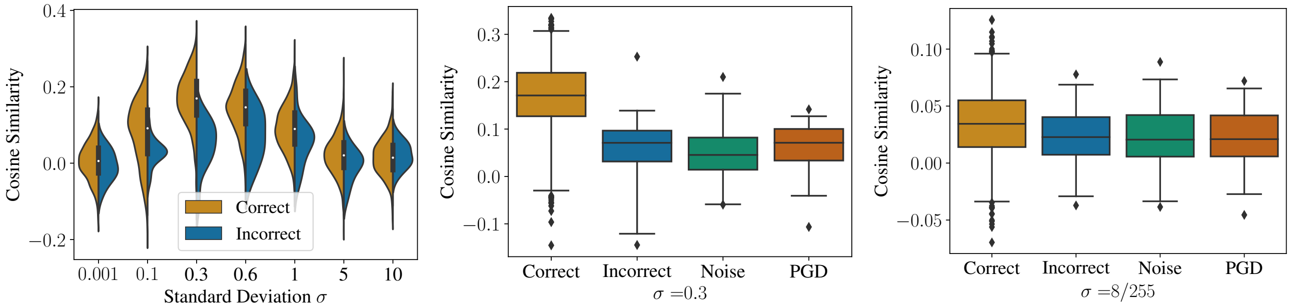


Figure 2: Box-plots of the mean value of the cosine similarity matrices and the softmax scores of the neural network for different data. The boxes show the quartiles of the data set while the whiskers extend to 95% of the distributions. The values are calculated on the **MNIST** validation set using the **MNIST** model.

We create several adaptive attacks that are designed to fool the proposed GGA method [Grosse et al., 2017, Carlini and Wagner, 2017]. With these attacks, we aim to obtain cosine similarity matrices which resemble those of correctly classified samples. This means that the cosine similarity between non-predicted classes in S_1 should be high on average while the cosine similarities in S_2 are small.

Targeted attacks: We use a targeted PGD-based attack to maximize the loss w.r.t. a random target class which is not the ground truth. We argue that such attacks could result in similar saliency maps for all other classes since the attack



(a) ζ values for different σ (MNIST)

(b) ζ values for different data (MNIST)

(c) ζ values for different data (CIFAR10)

Figure 3: (a) Violin plots of ζ_x estimates for the MNIST dataset for different values of the standard deviation σ. (b) and (c) show ζ estimates for different data types and constant σ for the MNIST and CIFAR10 data sets, respectively.

will optimize the input towards a local minimum of the loss landscape w.r.t. the target class. We employ this attack both with the mean squared error (T-MSE) and the categorical cross-entropy loss (T-SCE). We used the same step size of $\alpha = \frac{\epsilon}{4}$ and 100 attack iterations for all targeted attacks.

Cosine similarity attack (CSA): We use a PGD-based attack and additionally add a cosine similarity objective to optimize the perturbation such that the saliency maps of all non-predicted classes align. The loss of the cosine similarity objective is given by:

$$\mathcal{L}^{CSA}(x, \hat{k}) = \frac{1}{|S_1|} \sum_{c_{ij} \in S1} c_{ij}(x). \tag{3}$$

For this attack we exchange the ReLU activation functions [Agarap, 2018] with Softplus activations [Dugas et al., 2000]. This was shown to be an effective way to calculate a second-order gradient to attack the saliency maps of neural networks [Dombrowski et al., 2019]. We used the same step size and attack iterations as for the targeted PGD attacks. We achieved the highest success rate of the attack by weighting the CS objective in equation 3 by 0.8 and the cross-entropy objective with 0.2. As our cosine similarity attack needs the second-order gradient for optimization it may be prone to common pitfalls, such as gradient obfuscation [Athalye et al., 2018]. We encourage other researchers to create adaptive attacks that circumvent our method.

5 RESULTS AND DISCUSSION

In the following, we summarize and analyze the findings of the experiments used to evaluate the proposed GGA method.

5.1 LOSS LANDSCAPE ANALYSIS

In a preliminary experiment, we evaluated if it is necessary to analyze the geometry between multiple gradients in the input space to detect untrustworthy data or if the gradient w.r.t. the predicted class is sufficient for a geometrical analysis. To identify untrustworthy data with only the gradient of the predicted class, we introduce a property that lets us identify if a given data point lies on a local minimum of the loss landscape. First, we observe that the following holds.

Theorem 1. *Let $\zeta_x(\tilde{x})$ be defined by*

$$\zeta_x(\tilde{x}) := \frac{\langle -\nabla_x \mathcal{L}(F_\theta(\tilde{x}), i), x - \tilde{x} \rangle}{|\nabla_x \mathcal{L}(F_\theta(\tilde{x}), i)||x - \tilde{x}|}, \quad \tilde{x} \neq x. \tag{4}$$

The point x is a local minimum of $\mathcal{L}(F_\theta(\cdot), i)$ if and only if

$$0 \leq \liminf_{|x-\tilde{x}| \to 0} \zeta_x(\tilde{x}) \leq \limsup_{|x-\tilde{x}| \to 0} \zeta_x(\tilde{x}) \leq 1. \tag{5}$$

Proof. Can be found in the appendix. □

We empirically tested if the properties of ζ_x defined in equation 4 can be used to differentiate between trustworthy and untrustworthy predictions. Since ζ_x is a cosine similarity, it holds that $\zeta_x \in [-1, 1]$. To test our hypothesis, we estimate ζ_x in a neighborhood of a sample x and check whether it is non-negative. To generate points close to x we add i.i.d. Gaussian noise with standard deviation $\sigma > 0$. Following this procedure, we calculated the statistics of ζ_x in equation 4 with $1,000$ injections per sample on the MNIST and CIFAR10 validation set. Subfigure 3a shows the behavior of ζ_x for increasing values of σ for correct and incorrect classifications. In the direct vicinity of the original sample, the gradients are mostly orthogonal to noise for both correct and incorrect classifications with an average cosine similarity of zero, indicating that the corresponding samples lie in a relatively wide minimum of the loss. Incorrect classifications show a slightly lower values of ζ_x. For an increasing standard deviation σ, the difference of ζ_x values between correct and incorrect classifications increases. When the samples $\tilde{x}$ are too far from x the value ζ_x becomes normally distributed around zero. The optimal value to distinguish trustworthy

Table 1: True negative rate in [%] for different augmentations, OOD data, and attacks. All values are given for a true positive rate of 95%. Additionally the AUROC, AUPR-In, and AUPR-Out for all data types is shown.

Data set	Method	Noise	PGD	Rotation	B&B	B&B$_{\text{L2}}$	OOD	AUROC	AUPR-In	AUPR-Out
MNIST	**Baseline**	45.8	3.3	59.7	99.9	99.3	55.0	86.1	46.7	97.8
	ODIN	97.2	2.5	92.0	93.6	89.1	69.4	88.2	39.9	98.1
	Maha	**100.0**	12.4	93.5	98.4	99.0	97.9	92.4	88.3	**99.9**
	Ours	**100.0**	**98.9**	**98.2**	**100.0**	**100.0**	**98.1**	**99.5**	**97.7**	**99.9**
CIFAR10	**Baseline**	10.5	0.0	55.1	97.5	95.4	77.2	82.7	30.2	97.4
	ODIN	25.3	0.0	45.1	14.0	14.5	83.6	81.2	29.7	96.8
	Maha	93.1	85.9	73.2	90.8	91.3	**88.2**	90.0	72.1	99.0
	Ours	**95.6**	**92.6**	**84.5**	**93.2**	**93.3**	84.2	**96.3**	**83.7**	**99.4**
CIFAR100	**Baseline**	32.4	0.0	40.1	80.7	81.5	6.7	55.2	11.3	93.6
	ODIN	16.8	0.0	15.6	8.5	7.9	23.3	60.2	16.3	94.1
	Maha	93.7	81.9	77.3	52.1	55.3	**86.2**	68.2	47.1	98.4
	Ours	**95.1**	**98.5**	**95.1**	**98.1**	**97.9**	83.5	**98.0**	**87.5**	**99.7**
UCR ECG	**Baseline**	6.7	0.5	N/A	1.5	1.8	0.0	11.8	6.9	74.7
	ODIN	0.0	0.0	N/A	0.0	0.0	0.0	0.0	6.4	70.7
	Ours	**81.5**	**96.7**	N/A	**75.9**	**75.8**	**100.0**	**96.9**	**88.8**	**99.1**

and untrustworthy predictions was approximately the respective maximum perturbation magnitude ϵ for each data set. A summary of the results is displayed in Figure 3.

While the quantity in equation 4 can be used to distinguish untrustworthy and trustworthy samples for MNIST to some degree, it has major limitations. First, the calculation ζ_x is computationally expensive as it requires a considerable amount of sampling operations and model evaluations. Secondly, adaptive adversarial attacks could move a sample towards a local minimum of the loss landscape w.r.t. the predicted class and circumvent this approach in our experiments. Lastly, the method does not scale to more complicated data sets like CIFAR10. In the following experiments, we additionally consider the gradient directions w.r.t. other classes with the GGA method, which makes the detection of misclassifications induced by adversarial attacks substantially more robust.

5.2 OUT-OF-DISTRIBUTION DETECTION AND ADVERSARIAL ATTACKS

First, we studied the detection performance for OOD data and adversarial attacks as described in the previous section. The results are summarized in Table 1. As reported in prior work, the baseline and ODIN methods fail to identify adversarial attacks. In contrast, the proposed GGA shows high identification performance for all attacks. The augmentations which were the most difficult to detect for the computer vision tasks were rotations. UCR ECG noise and the B&B attacks were the most difficult to detect. For the detection of OOD data, the GGA method achieves worse results than the Mahalanobis distance-based approach (Maha) in some cases [Lee et al., 2018]. However, in contrast to GGA, Maha requires additional finetuning of the OOD and adversarial detector on OOD data and adversarial examples, respectively.

5.3 ADAPTIVE ADVERSARIAL ATTACKS

Next, we studied the detection performance for adaptive adversarial attacks that were specifically designed to fool the GGA detector. As seen in Table 2, the proposed GGA shows high identification performance on all adaptive attacks. The targeted PGD attacks (T-SCE, T-MSE) show a higher success rate than the untargeted PGD attack. Using the softmax cross-entropy loss for the targeted attack was more effective in our experiments. We observed that we can successfully increase the cosine similarity between non-predicted classes in the cosine similarity matrices with the cosine similarity attack (CSA). CSA achieves a considerably higher success rate than the standard PGD attack. However, combining this objective with the goal to induce misclassifications seems to be ineffective. A higher weight for the cosine similarity objective results in considerably fewer misclassifications and vice versa. The CSA attack was only able to induce misclassifications on 561, 1354, and 857 out of $10,000$ samples for the MNIST, CIFAR10, and CIFAR100 data sets, respectively. In contrast, the untargeted and targeted PGD attacks achieved 100% success rate and led to a misclassification on $10,000$ out of $10,000$ images on all data sets.

5.4 GRADIENT OBFUSCATION

Prior work demonstrates that defense mechanisms that are apparently robust to adaptive attacks can often be circumvented with another or simpler optimization objectives

Table 2: Identification accuracy [%] for different adaptive attacks for the proposed GGA. All values are given for a TPR of 95%

Data set	T-SCE	T-MSE	CSA
MNIST	95.4	97.3	72.1
CIFAR10	90.3	91.5	69.7
CIFAR100	96.9	97.8	71.6

[Athalye et al., 2018, Tramèr et al., 2020]. Complex objectives often result in noisy loss landscapes with unreliable gradient information. To evaluate if this phenomenon applies to the CSA attack, we further inspect the behavior of the CSM features over a wide variety of perturbed data points. In particular, we inspect the behavior of the objective in equation 3 along the direction of a successful adversarial perturbation (g) and a random orthogonal direction ($g^{\perp}$) originating from a clean sample. This results in a three-dimensional map where the x-axis g and y-axis $g^{\perp}$ describe the perturbation of the current data point, while the z-axis shows the value described in equation 3. A representative map for an individual sample of the MNIST data set is shown in Figure 4. The predicted label of the classifier is color-coded, where the upper plateau in orange corresponds to the ground truth class while the lower plateau in blue corresponds to the class predicted after the adversarial attack. Near the decision boundary, the CSM characteristics fluctuate as the gradient directions between saliency maps of different classes start to diverge. It can be seen that the mean value of the CSMs is a stable indicator of the classifier decision. Furthermore, the smoothness of the map indicates that the mean value can be utilized as an objective for an adaptive attack.

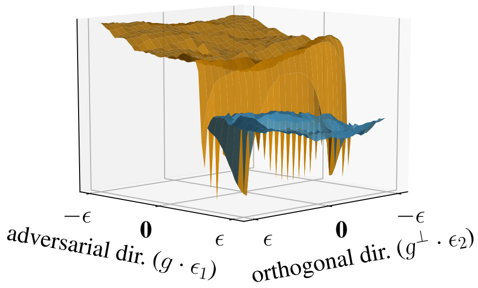


Figure 4: Landscape of the mean value of cosine similarity maps centered around a clean sample x. We calculate the loss value for sample $x + \epsilon_1 \cdot \gamma + \epsilon_2 \cdot \gamma^{\perp}$ where γ is the direction of a successful adversarial attack and $\gamma^{\perp}$ a random orthogonal direction. The different colors indicate the predicted class of the neural network, with orange indicating the correct class.

Table 3: Detection performance for cosine similarity maps which are calculated with only the top-N prediction of the classifier. AUROC, AUPR-In, and AUPR-Out in [%] for different augmentations, OOD data, and attacks are shown.

CIFAR100	AUROC	AUPR-In	AUPR-Out
top-100 (all)	98.0	87.5	99.7
top-10	98.0	86.9	99.7
top-5	97.7	85.6	99.7

5.5 ENHANCING THE EFFICIENCY

The main computational cost of GGA is given by the preliminary computation of the saliency maps for each reference class. Here, we show that it is possible to rely on a partial computation of the CSM with only the top-N predicted classes. This allows GGA to scale to data sets with a large number of output classes. Table 3 demonstrates the detection performance for the same adversarial attacks and outlier data as used in Section 5.2 for CSMs which are computed with the top-N predictions only. Even for the case of only 5% of the original saliency maps used to calculate the CSMs the performance degrades only marginally for all detection tasks. We observed that the cosine similarity between the gradients of the predicted class and the non-predicted classes is mostly sufficient for the detection of untrustworthy predictions. We argue that this enables the algorithm to perform well with largely reduced CSMs. In our experiments the partial CSMs performed similarly on the adaptive attacks as well. Note that the computation of the CSMs can be parallelized, which results in no time overhead between partial and full CSMs when sufficient memory is used for the calculation. In practice, the computational overhead can be adjusted based on the required detection performance and available resources.

6 CONCLUSION

In this paper, we propose a novel geometric gradient analysis (GGA) method, which is designed to identify out-of-distribution data and adversarial attacks in differentiable neural networks. The proposed method does not require retraining of the neural network model and can be used with any pre-trained differentiable model. We demonstrate the effectiveness of GGA for the detection of untrustworthy data with a simple framework. We first extract standard features from the calculated cosine similarity matrices (CSMs) and use them to identify untrustworthy data with a basic outlier detection method. We show that GGA achieves competitive performance for outlier detection without a complex classifier. Furthermore, we observe that GGA effectively detects state-of-the-art and adaptive adversarial attacks. Finally, we demonstrate how GGA can be efficiently implemented for data sets with a large number of output classes. Future work will explore the end-to-end training of a GGA-based detector with the CSMs.

Author Contributions

L.S. conceived the proposed method, conducted all experiments and wrote the initial draft of the paper. A.N. discussed the results. L.B. proposed the theorem and experiment described in Section 5.1 and reviewed the paper. All authors analyzed the results and contributed to the final manuscript.

Acknowledgements

Leo Schwinn was supported by the FAU Emerging Fields Initiative. An Nguyen gratefully acknowledges personal funding from Siemens Healthineers. Bjoern M. Eskofier gratefully acknowledges the support of the German Research Foundation (DFG) within the framework of the Heisenberg professorship programme (grant number ES 434/8-1).

A NECESSARY AND SUFFICIENT CONDITIONS FOR LOCAL MINIMA OF THE LOSS FUNCTION

The proof of Theorem 1 is divided in two steps. We first prove that equation 5 is necessarily met if the function $x \mapsto \mathcal{L}(F_\theta(x), i)$ attains a local minimum in x. This follows from

Lemma 1. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a C^1-function and let x be a local minimum of f. Then it holds*

$$
\begin{aligned}
0 &\leq \liminf_{|x-\tilde{x}|\to 0} \frac{\langle -\nabla f(\tilde{x}), x-\tilde{x}\rangle}{|\nabla f(\tilde{x})||x-\tilde{x}|} \\
&\leq \limsup_{|x-\tilde{x}|\to 0} \frac{\langle -\nabla f(\tilde{x}), x-\tilde{x}\rangle}{|\nabla f(\tilde{x})||x-\tilde{x}|} \leq 1.
\end{aligned}
$$

Proof. Taylor expanding around $\tilde{x}$ gives

$$f(x) = f(\tilde{x}) + \langle \nabla f(\tilde{x}), x-\tilde{x}\rangle + o(|x-\tilde{x}|),$$

which can be reordered to

$$\frac{f(\tilde{x})-f(x)}{|x-\tilde{x}|} = \frac{\langle -\nabla f(\tilde{x}), x-\tilde{x}\rangle}{|x-\tilde{x}|} + o(1).$$

If x is a local minimum, one obtains

$$0 \leq \liminf_{|x-\tilde{x}|\to 0} \frac{\langle -\nabla f(\tilde{x}), x-\tilde{x}\rangle}{|x-\tilde{x}|}$$

which directly implies the desired inequality. □

Non-negativity of the cosine similarity in equation 4 can also be brought into correspondence with positive semi-definiteness of the Hessian of f which follows from

Lemma 2. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a C^2-function. Then for all vectors $e \in \mathbb{R}^d$ with $|e| = 1$ it holds*

$$\lim_{r\to 0} \left\langle \frac{\nabla f(x+re) - \nabla f(x)}{r}, e \right\rangle = \langle Hf(x)e, e\rangle .$$

Proof. We compute

$$
\begin{aligned}
&\left\langle \frac{\nabla f(x+re) - \nabla f(x)}{r}, e \right\rangle \\
&= \left\langle \frac{1}{r}\int_0^r \frac{\mathrm{d}}{\mathrm{d}t}\nabla f(x+te)\mathrm{d}t, e \right\rangle \\
&= \left\langle \frac{1}{r}\int_0^r Hf(x+te)e\mathrm{d}t, e \right\rangle
\end{aligned}
$$

where $Hf = (\partial_i \partial_j f)_{i,j}$ denotes the Hessian matrix of f. Since f is a C^2-function, the integral $\frac{1}{r}\int_0^r Hf(x+te)\mathrm{d}t$ converges to $Hf(x)$ as $r \to 0$. Therefore, one obtains

$$\lim_{r\to 0} \left\langle \frac{1}{r}\int_0^r Hf(x+te)e\mathrm{d}t, e \right\rangle = \langle Hf(x)e, e\rangle .$$

□

We can now proceed to the proof of Theorem 1.

Proof of Theorem 1. Applying Lemma 1 to $f(x) := \mathcal{L}(F_\theta(x), i)$ shows that equation 5 is necessary for x to be a local minimum.

For the converse direction we argue as follows: First, we note that equation 5 implies that x is a critical point with $\nabla f(x) = 0$. Otherwise one could set $\tilde{x} = x - t\nabla f(x)$ with $t > 0$ and obtain

$$\frac{\langle -\nabla f(\tilde{x}), x-\tilde{x}\rangle}{|\nabla f(\tilde{x})||x-\tilde{x}|} = \frac{\langle -\nabla f(\tilde{x}), \nabla f(x)\rangle}{|\nabla f(\tilde{x})||\nabla f(x)|} \to -1,$$

as $|x-\tilde{x}| \to 0$, since ∇f is continuous. This is a contradiction to equation 5 and hence $\nabla f(x) = 0$.

This allows us to compute

$$
\begin{aligned}
&\frac{\langle -\nabla f(\tilde{x}), x-\tilde{x}\rangle}{|\nabla f(\tilde{x})||x-\tilde{x}|} \\
&= \frac{\langle \nabla f(x) - \nabla f(\tilde{x}), x-\tilde{x}\rangle}{|\nabla f(x) - \nabla f(\tilde{x})||x-\tilde{x}|} \\
&= \left\langle \frac{\nabla f(x) - \nabla f(\tilde{x})}{|x-\tilde{x}|}, \frac{x-\tilde{x}}{|x-\tilde{x}|} \right\rangle \frac{|x-\tilde{x}|}{|\nabla f(x) - \nabla f(\tilde{x})|}.
\end{aligned}
$$

Hence, if this expression is asymptotically non-negative for all $\tilde{x}$ converging to x, we can choose arbitrary $e \in \mathbb{R}^d$ with $|e| = 1$, define $\tilde{x} = x + re$ and apply Lemma 2 to get

$$0 \leq \lim_{r\to 0} \left\langle \frac{\nabla f(x+re) - \nabla f(x)}{r}, e \right\rangle = \langle Hf(x)e, e\rangle.$$

Since e was arbitrary, this means that x is a local minimum of the loss f. □

References

Abien Fred Agarap. Deep learning using rectified linear units (relu). *CoRR*, abs/1803.08375, 2018.

Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, pages 274–283, 2018.

Wieland Brendel, Jonas Rauber, Matthias Kümmerer, Ivan Ustyuzhaninov, and Matthias Bethge. Accurate, reliable and fast robustness evaluation. In *NeurIPS*, pages 12861–12871, 2019.

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

Jiefeng Chen, Yixuan Li, Xi Wu, Yingyu Liang, and Somesh Jha. Robust out-of-distribution detection in neural networks. *CoRR*, abs/2003.09711, 2020.

Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. In *NeurIPS*, pages 2898–2909, 2019.

Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. The ucr time series classification archive, October 2018. `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/`.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and ROC curves. In *ICML*, volume 148 of *ACM International Conference Proceeding Series*, pages 233–240. ACM, 2006.

Ann-Kathrin Dombrowski, Maximilian Alber, Christopher J. Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. In *NeurIPS*, pages 13567–13578, 2019.

Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. In *NeurIPS*, pages 472–478, 2000.

Christian Etmann, Sebastian Lunz, Peter Maass, and Carola Schönlieb. On the connection between adversarial robustness and saliency map interpretability. In *ICML*, pages 1823–1832, 2019.

Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Adversarial attacks on deep neural networks for time series classification. In *IJCNN*, pages 1–8. IEEE, 2019.

Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Stefano Soatto. Classification regions of deep neural networks. *CoRR*, abs/1705.09552, 2017.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy A. Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *CoRR*, abs/2010.03593, 2020.

Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick D. McDaniel. On the (statistical) detection of adversarial examples. *CoRR*, abs/1702.06280, 2017.

Jindong Gu and Volker Tresp. Saliency methods for explaining adversarial attacks. *CoRR*, abs/1908.08413, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, pages 770–778, 2016.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.

Saumya Jetley, Nicholas A. Lord, and Philip H. S. Torr. With friends like these, who needs adversaries? In *NeurIPS*, pages 10772–10782, 2018.

Heinrich Jiang, Been Kim, Melody Y. Guan, and Maya R. Gupta. To trust or not to trust A classifier. In *NeurIPS*, pages 5546–5557, 2018.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, pages 7167–7177, 2018.

Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Christopher D. Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT Press, 1999.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS workshop on deep learning and unsupervised feature learning*, 2011.

Tomás Pevný. Loda: Lightweight on-line detector of anomalies. *Mach. Learn.*, 102(2):275–304, 2016. doi: 10.1007/s10994-015-5521-0. URL `https://doi.org/10.1007/s10994-015-5521-0`.

Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *CoRR*, abs/2002.08347, 2020.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *ICLR*. OpenReview.net, 2019.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. `https://github.com/zalandoresearch/fashion-mnist`, 2017.

Aaron Xichen. pytorch-playground. `https://github.com/aaron-xichen/pytorch-playground`, 2019. [Online; accessed 30-11-2020].

Dengpan Ye, Chuanxi Chen, Changrui Liu, Hao Wang, and Shunzhi Jiang. Detection defense against adversarial attacks with saliency map. *CoRR*, abs/2009.02738, 2020.