Ali *et al.*
EURASIP Journal on Information Security (2023) 2023:3
https://doi.org/10.1186/s13635-023-00137-0

EURASIP Journal on Information Security

RESEARCH

Open Access

Gaussian class-conditional simplex loss for accurate, adversarially robust deep classifier training

Arslan Ali, Andrea Migliorati*, Tiziano Bianchi and Enrico Magli

Abstract

In this work, we present the Gaussian Class-Conditional Simplex (GCCS) loss: a novel approach for training deep robust multiclass classifiers that improves over the state-of-the-art in terms of classification accuracy and adversarial robustness, with little extra cost for network training. The proposed method learns a mapping of the input classes onto Gaussian target distributions in a latent space such that a hyperplane can be used as the optimal decision surface. Instead of maximizing the likelihood of target labels for individual samples, our loss function pushes the network to produce feature distributions yielding high inter-class separation and low intra-class separation. The mean values of the learned distributions are centered on the vertices of a simplex such that each class is at the same distance from every other class. We show that the regularization of the latent space based on our approach yields excellent classification accuracy. Moreover, GCCS provides improved robustness against adversarial perturbations, outperforming models trained with conventional adversarial training (AT). In particular, our model learns a decision space that minimizes the presence of short paths toward neighboring decision regions. We provide a comprehensive empirical evaluation that shows how GCCS outperforms state-of-the-art approaches over challenging datasets for targeted and untargeted gradient-based, as well as gradient-free adversarial attacks, both in terms of classification accuracy and adversarial robustness.

Keywords Deep neural networks, Classification, Adversarial robustness, Adversarial training, Adversarial defense

1 Introduction

Over the course of the last few years, deep learning has been applied to several multimedia, scientific, and industrial applications thanks to its ability to generalize well over unseen data. The performance of these techniques has reached similar or even greater accuracy levels than humans in multiple and complex visual and classification tasks [1, 2]. More recently, deep networks have also shown remarkable performance at learning complex mappings for image translation and segmentation [3–5]. However, the ever-growing use of neural networks in our society raises serious concerns in the matter of *security*, as they can be targeted by malevolent *adversaries*.

In particular, many barriers affect the use of deep neural networks in applications where security is of key importance, such as medical diagnostics and autonomous driving [6, 7]. One of the most severe threats to deep learning is represented by *adversarial perturbations*, a collection of methods that are designed to interfere with neural networks' input data to produce undesired outputs, shift the expected outcome, or more in general cause algorithm malfunctions and performance reductions. This happens in the face of modifications that are very difficult to detect, to the extent that they are often undetectable to the human eye.

*Correspondence:
Andrea Migliorati
andrea.migliorati@polito.it
DET - Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy

Lately, numerous methods have been devised to apply successful adversarial perturbations [8, 9]. Against such techniques, neural networks can be considered unreliable when employed in applications involving security and safety. Despite many solutions have been proposed, the community has not yet been able to establish a conclusive defense mechanism against the wide range of adversarial perturbations. In particular, recent works such as [10–13] consider the importance of performing correct evaluations of adversarial attacks and defense algorithms, introducing a rigorous benchmark method to evaluate adversarial robustness on image classification tasks and to produce certified defenses against adversarial attacks. The main factor hindering the robustness of deep networks is that the learned boundaries in which the feature space is partitioned are highly complex and non-linear. Works such as [14, 15], specifically addressing this problem, observed that most of the mass of data points is actually positioned very close to the learned decision boundaries, in such a way that the robustness of the classifier is heavily affected by adversarial perturbations. Among the most widespread techniques employed for improving robustness figures *adversarial training* [16], which consists in augmenting with adversarial samples the training data fed to the network, also when working with large datasets [17].

However, adversarial training is a very time-consuming process. To generate an adversarial image, each stochastic gradient descent (SGD) iteration requires multiple gradient computations in addition to those needed to update the network parameters. On medium or large datasets such as ImageNet, CIFAR-10, and CIFAR-100, this is a burdensome task. To this end, several alternatives have been developed. Some recent works include [17, 18], where authors study ensemble adversarial training and generative adversarial perturbations for large-scale datasets such as ImageNet. These methods lead to improved robustness, but at the cost of reducing the classification accuracy in the case of no attacks. Further, recent works that tackle the robustness problem try to devise efficient adversarial defense methods such as curvature regularization [19], logits regularization [20], injection of noise [21], and unlabeled data [22] at inference time, while others focus on the use of randomized classifiers for improved robustness [23, 24].

In this paper, we address the adversarial robustness problem to devise a method that can protect from both gradient-based and gradient-free attacks without requiring computationally intensive training. To this end, we design a novel loss function for the classifier that enables the learning of features maximizing inter-class separation and minimizing intra-class separation. The proposed loss function provides state-of-the-art classification accuracy while ensuring a large degree of adversarial robustness without any other specific training trick. In the proposed method, the network learns a mapping of input data onto well-defined, Gaussian target distributions in a regularized latent space. The employed loss maximizes linear separability of the learned distributions in the latent space, enabling straightforward classification based on simple thresholding of the decision variable. Ultimately, this is a general-purpose technique that improves on the traditional cross-entropy loss function. We show that the proposed method achieves higher classification accuracy with respect to cross-entropy even when no attacks are applied while exhibiting notably improved robustness against multiple adversarial attacks. Indeed, our design requires significantly stronger attacks to induce the same classification errors than other methods, thanks to the maximized distances between adjacent decision regions. Furthermore, the proposed loss function can be generally employed in the fine-tuning of any pre-trained classifier, allowing for improvements in robustness and accuracy while also reducing the training overhead.

1.1 Contributions

In this work, we extend and improve on our previous work [25]. We provide an in-depth analysis of how GCCS improves the classification accuracy by correctly mapping the samples that are typically misclassified by competing approaches, while at the same time showing the motivation behind such higher misclassification. We include thorough ablation experiments to evaluate the effects of the GCCS loss parameters on classification accuracy. Further, we show that GCCS achieves higher robustness compared to conventional adversarially trained models with no additional cost with respect to natural training. In particular, we show that adversarial training combines well with our proposed method, which proves to be more robust against a broad range of attacks than competing methods even when coupled with AT. We report a thorough investigation and experimental evaluation of the proposed method for classification over widespread datasets such as MNIST [26], FMNIST [27], SVHN [28], and the very challenging CIFAR10 and CIFAR100 [29]. Also, we assess the performance of our technique showing that it outperforms the state-of-the-art defence methods by a considerable margin in terms of robustness against targeted (TGSM [30], JSMA [31]) and untargeted (DeepFool [32], FGSM [33], and PGD [16]) gradient-based adversarial attacks. Also, we evaluate the adversarial robustness of our method when dealing with the very challenging gradient-free SPSA [34] attack. Finally, we analyze the behavior of our method when used together with adversarial training via a detailed graphical representation of the latent space, showing that the combination further

improves model robustness. In particular, to assess adversarial robustness, we refer to the guidelines presented by [13, 35] and [36].

2 Related works

The idea of manipulating data seeking to disrupt a classifier's performance was first formally investigated in works such as [37, 38], where the authors were interested in countering adversaries in applications like surveillance and spam, intrusion, and fraud detection. In the following years, many works [33, 39–41] analyzed security aspects of deep neural networks. They showed how easily they could be led into misclassification by receiving as input *adversarially* altered data, i.e., inputs specifically modified according to the sign of the gradients of the cost function.

The effectiveness of adversarial perturbations extends both to the virtual and the physical realms, as reported in [42] and [30, 43], where deep networks are respectively shown to be fooled in recognizing faces by specifically devised adversarial glasses, and in classifying images that are printed on paper or 3D-printed after being tampered with adversarial perturbations. Also, Ross and Doshi-Velez [44] showed that adversarial attacks devised to meddle with one specific network model can easily transfer to all models trained on the same dataset, which are greatly prone to be affected by the perturbation. Adversarial attacks can be crafted and performed with remarkable results also without any prior knowledge of the targeted image, as thoroughly examined in [45, 46]. In conclusion, despite being one of the most effective defense algorithms against perturbations, adversarial training suffers from a large gap between training and test accuracy [47].

While some of the latest findings suggest that the existence of adversarial perturbations is a structural weakness of deep neural networks [48] and that the strength of the applied adversarial attack directly bounds the expected robustness of the classifier [49], there is no universal agreement on the trade-off between adversarial robustness and generalization. Some authors argue that defense methods will inevitably decrease the classification accuracy [50], whereas others claim that both adversarially robust and general models are indeed possible [51]. This situation is reflected in the large variety of methods for defending against such attacks.

Authors in [52], after formally characterizing the robustness of a classifier against attacks, compute the minimum perturbation required to affect classification, identifying instance-specific lower bounds dependent on the strength of the attack, and then proceed to define the Cross-Lipschitz Regularization (CLR) functional, whose goal is to constrain differences at data points of the classifier functions to be as constant as possible. Ross and Doshi-Velez [44] instead offer a method called Input Gradient Regularization (IGR) that is applied at training time and forces the gradients of the network to be as smooth as desired; the underlying motivation is that models trained with gradients that exhibit a smaller set of spike values tend to respond in more understandable ways to adversarial attacks while also retailing greater robustness. Finally, Jakubovitz et al. [53] rely on the Frobenius norm of the Jacobian of the gradients (JR) to craft a low-complexity method that is applied to already trained networks to improve robustness. This method has also been implemented in a computationally effective way [54]. Finally, works such as [32, 55, 56] propose the idea of introducing adversarial samples in the training data as a form of augmentation, so that the network could learn how to cope with adversarial perturbation at training time. However, as shown in [16, 17, 35], adversarial training does not prevent adversaries to interfere with classification accuracy, to the extent that universal adversarial attacks can actually be devised to be highly effective on different datasets and network structures [30, 57–59]. Furthermore, adversarial training also comes with the extra cost required to generate adversarial samples that consequently increase the size of the training set and hence training time [60].

Approaches in the literature mainly focus on learning suitable classification boundaries. Instead, in this paper, we propose a method that directly learns a mapping of the input onto output target distributions in a regularized latent space. Despite being a passive defense method, the proposed design inherently enjoys high classification accuracy and robustness against adversarial attacks, with almost no training overhead. In particular, through the use of an encoder, features of an arbitrary number of classes are mapped from their input class distributions onto Gaussian distributions whose expected values lie on the vertices of a simplex in the latent space such that inter-class separability is maximized and intra-class separability minimized at training time. Previously, other works have considered a learned mapping onto a well-behaved latent space, such as techniques based on variational and adversarial autoencoders [61, 62], or even discriminant analysis methods that aim at dimensionality reduction [63]. Examples include the approach by Stuhlsatz et al. [64] that consists of a generalized version of Linear Discriminant Analysis (LDA) relying on deep networks, and [65] that first performs LDA on top of a neural network and then maximizes eigenvalues on the final hidden representation. These approaches, however, while showing great effectiveness at increasing the distance of the centers of distributions for well-separated classes, appear to perform poorly in terms of inter-class

separability when it comes to neighboring classes with blurred borders.

Lately, latent space regularization has been used also in [66–68]. In these works, deep learning is employed in the field of biometric authentication to separate authorized users from non-authorized ones. These methods work by regularizing a 2D latent space via a cost function derived from a simplified version of the Kullback-Leibler divergence. While relevant to the topic, these approaches do not scale for higher-dimensional classification, which is instead the focus of the proposed GCCS method.

Finally, works such as Gaussian Mixture loss (L-GM [69]) and Max-Mahalanobis center loss (MMC [70]) formally showed the inherent property of the cross-entropy loss and its variations to learning feature vectors that are sparsely spread over the feature space, causing the models trained in such fashion to be prone to suffer adversarial attacks.

3 Proposed method

The goal of the proposed GCCS method is to learn the most discriminative features of the input and map them onto target distributions with the mean values centered on the vertices of a simplex, as shown in Fig. 1: a deep network that consists of a feature extractor and a latent space mapper receives as input labeled training data ($\mathbf{X}$) to tackle a multi-class classification problem. The feature extractor learns nonlinear functions that transform arbitrary input data distributions and produces discriminative, highly separable features. The subsequent latent space mapper is composed of one or more fully connected layers that map the output decision variable ($\mathbf{z}$) onto desired target distributions that lie in a D-dimensional latent space, where D denotes the number of classes. In particular, we do not employ any nonlinear activation in the mapper's last layer. Also, it is important to notice that our proposed GCCS technique is not dependent on a particular feature extraction design. Hence, any deep architecture might be employed.

In the following, we describe the three fundamental components of the proposed approach, namely, a target model for the feature distributions in the latent space, a cost function devised to ensure that such distributions could be obtained, and last, a decision rule for classification. Ablation studies for the relevant parameters are reported in Section 4.

3.1 Model for the target distributions

The GCCS method projects high-dimensional data belonging to D classes onto a lower-dimensional space. This is done by employing a latent space that has the same number (D) of dimensions as the number of classes so that each dimension corresponds to a class-conditional statistical distribution.

First, we denote the desired target distribution for class $\mathcal{C}_i$, $i = 1, \dots D$ as $\mathbb{P}_i$. In particular, we choose $\mathbb{P}_i$ to be a D-variate Gaussian distribution, such that $\mathbb{P}_i = \mathcal{N}(\mu_{Ti}, \mathbf{\Sigma}_T)$, where $\mathcal{N}$ indicates the Gaussian distribution. In particular, μ_{Ti} and $\mathbf{\Sigma}_T$ indicate the target mean and standard deviation vectors, i.e., $\mu_{Ti} = \mu_T \mathbf{e}_i$ and $\mathbf{\Sigma}_T = \sigma_T^2 \mathbb{I}_D$, where $\mathbf{e}_i$ refers to the ith standard unit vector, while $\mathbb{I}_D$ is the $D \times D$ identity matrix. Hence, the target statistical distributions are centered on the vertices of a regular $(D-1)$-simplex at $\mu_T \mathbf{e}_i$, such that every distribution $\mathbb{P}_i$ has a mean value in the direction determined by $\mathbf{e}_i$. The μ_T and σ_T parameters, referring to the characteristics of the target distributions we want to enforce at training time, are defined arbitrarily keeping in mind that, to avoid mixing of the classes as D increases and to have more separable distribution, the parameters should take values such that $\mu_T/\sigma_T > \sqrt{2D}$. In other words, the choice of μ_T and σ_T influences the inter-class/intra-class separability at the classification stage.

The defined target model exhibits several advantages. First, assuming the simplex to be regular ensures that each of the classes is equidistant from all others. For this reason, there is no specific class that is "weaker" than

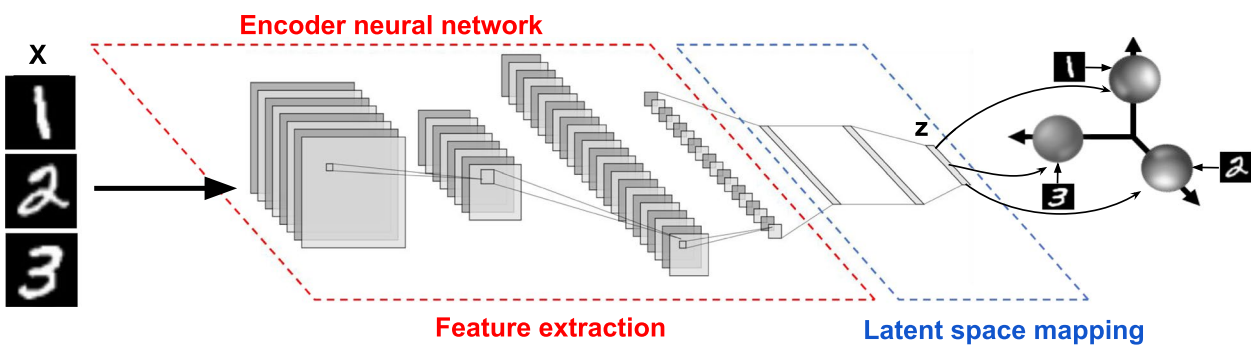


Fig. 1 The proposed GCCS architecture learns a mapping of features onto output Gaussian target distributions lying in the latent space

the others from an adversarial point of view. Moreover, the parameters μ_T and σ_T are chosen so as to maximize inter-class separability, leading to improved robustness. Secondly, since we choose Gaussian distributions as learning target, the optimal decision boundaries are simple hyperplanes. This design leads to better accuracy and greater robustness, in contrast to the typical behavior of neural networks which learn very complex decision boundaries.

3.2 Loss function

At this point, it is necessary to define a loss function that allows us to minimize a suitable distance metric between the latent output distributions and the target ones. Such a loss function is intended to simultaneously constrain the classes' distributions in order to maximize separability. Given $\mathbf{x} \in \mathbb{R}^n$, the input data belonging to D different classes, we refer to the output of the network as $\mathbf{z} = H(\mathbf{x})$, where $[z_1, \ldots, z_D] \in \mathbb{R}^D$, and H indicates the concatenation of the feature extractor and the latent space mapper (i.e., the whole network architecture). In other words, our method learns an encoding function of the input data $\mathbf{z} = H(\mathbf{x})$ such that $\mathbf{z} \sim \mathbb{P}_i$ if $\mathbf{x} \in \mathcal{C}_i$. During training, the network receives as input single b-sized batches of samples $\mathbf{X} \in \mathbb{R}^{b \times n}$, and outputs encoded data $\mathbf{Z} \in \mathbb{R}^{b \times D}$. In particular, we can estimate their first and second-order statistics for each class as the sample mean μ_{Oi} and sample covariance $\mathbf{\Sigma}_{Oi}$. Once target distributions and batch statistics are known, we can define a suitable loss to compute how distant the batch statistics are from the target distributions. To this end, we employ the Kullback-Leibler (KL) divergence. The KL divergence with respect to Gaussian target distributions, for the sample distribution of any class $\mathcal{C}_i$, can be calculated as:

$$\mathcal{L}_i = \log \frac{|\mathbf{\Sigma}_T|}{|\mathbf{\Sigma}_{Oi}|} - D + \operatorname{tr}(\mathbf{\Sigma}_T^{-1}\mathbf{\Sigma}_{Oi}) + (\mu_{Ti} - \mu_{Oi})^\intercal \mathbf{\Sigma}_T^{-1}(\mu_{Ti} - \mu_{Oi}) \tag{1}$$

By extension, we can compute the cumulative loss for all classes such as $\mathcal{L} = \sum_{i=1}^{D} \mathcal{L}_i$. In particular, $\mathcal{L}$ reaches its minimum value when the sample statistics of the D distributions exactly match the target ones. For a small batch size, however, it is difficult to control how the tails of the distributions behave relying only on KL. Hence, we also employ the Kurtosis $\mathcal{K}_{i,j}$ [71] of the j-th component of the i-th target distribution, defined as $\mathcal{K}_{i,j} = \left(\frac{z_{i,j} - \mu_{Oi,j}}{\sigma_{Oi,j}}\right)^4$. When enforcing multiple i.i.d. univariate normal distributions at training time, the target Kurtosis for each of the classes is $\mathcal{K}_{i,j} = 3$. This term can then be added to the cumulative loss, leading to the final proposed loss $\mathcal{L}^{\text{GCCS}}$ defined as

$$\mathcal{L}^{\text{GCCS}} = \sum_{i=1}^{D} [\mathcal{L}_i + \lambda(\mathcal{K}_i - 3)], \tag{2}$$

where $\mathcal{K}_i = 1/D \cdot \sum_j \mathcal{K}_{i,j}$. λ is a parameter that balances the effect of the Kurtosis term with respect to the effect of the KL divergence.

3.3 Decision rule

Last, when the training converges properly using the GCCS loss, the proposed method allows us to define optimal decision boundaries in the learned latent space. In particular, for the chosen target distributions, optimal boundaries are defined by the partition of the latent space into Voronoi regions such that all the points are respectively closer to their region centroid, i.e., the mean vector $\boldsymbol{\mu}_{Ti}$, than any other in the $(D-1)$-simplex. For this reason, the resulting decision rule requires the computation of the distance between output feature points and all centers of the regions and consequently classifies the sample as belonging to the class at the minimum distance:

$$\widehat{y} = \arg\max_i z_i. \tag{3}$$

In particular, given a feature extractor output z_i, Eq. 3 outputs the index of the predicted class for the correspondent input test sample.

4 Experiments

In this section, we first consider the case of no attacks, while specifically assessing adversarial robustness later on. We assess our method's pure classification accuracy as compared to the three considered state-of-the-art defense approaches Jacobian Regularization [53], Input Gradient Regularization [44], and Cross Lipschitz Regularization [52], as well as compared to a plain structure trained with a cross-entropy loss (*No Defense*). We first validate the choice of the distribution parameters, then analyze the quality of the features in terms of separation and classification accuracy. In more detail, we consider two different training settings in each case, namely *regular training*, where the network is trained with plain SGD, and *fine-tuning*, where a model that has been already pre-trained with a standard cross-entropy loss is further trained for 100 epochs with the proposed GCCS loss. The analysis of this second case is aimed at showing that, potentially, our proposed method could efficiently be employed on top of an already-trained network, allowing for improved adversarial robustness with little fine-tuning effort. The complete analysis of the robustness against adversarial attacks is then presented in Section 5.

4.1 Datasets and training parameters

To evaluate the performance of GCCS and other methods, the five datasets MNIST [26], FMNIST [27], SVHN [28], CIFAR-10, and CIFAR-100 [29] have been employed. For the less complex MNIST, FMNIST, and SVHN datasets, we employed ResNet-18 [72] as a feature extraction network. For the challenging CIFAR-10 and CIFAR-100, the Shake-Shake-96 and Shake-Shake-112 [73] regularization networks were used respectively, using a widen factor equal to 6 for the former and 7 for the latter. A fully connected layer that outputs a vector with dimension D follows the last layer of the encoder. The same baseline architectures were then used to compare our results with competing methods. After choosing fixed μ_T and σ_T values, as better explained in the following, the networks are trained for a total of 1800 epochs. For better network convergence, we employed cosine learning rate decay [74] with an initial value of 0.01 as well as weight decay with a rate set to 0.001. To avoid over-fitting, we apply dropout regularization [75] with a 0.8 keep probability to all the fully connected layers in the network. For every experiment, if not differently specified, we use a training batch size equal to 200, as better detailed in Section 4.4. The additional 100-epoch training we perform in the *fine-tuning* configuration are carried out with a learning rate of $1e^{-3}$, without learning rate decay.

4.2 Target distributions parameters

We explore the behavior of the target distributions by setting different target mean and variance values. We fix the mean μ_T and variance σ_T values for the target distributions so that they are centered on the vertices of a regular $(D-1)$-simplex. For this reason, the main parameter affecting our design is the μ_T/σ_T ratio, i.e., how far apart the distributions are with respect to the chosen variance.

In the experiment, we set $\sigma_T = 1$ so that the target distributions can be written as $\mathbb{P}_i = \mathcal{N}(\mu_T \mathbf{e}_i, \mathbb{I}_D)$. Then, we compute the classification accuracy as a function of $\mu_T \in [5, 300]$. Figure 2 shows the accuracy as a function of μ_T/σ_T on the MNIST, SVHN, and CIFAR10 dataset. It can be observed that in the $\mu_T \geq 30$ region the accuracy is even higher than that obtained with the cross-entropy loss. In the following, assuming unitary standard deviation for the target distributions ($\sigma_T = 1$), we set $\mu_T = 70$ which corresponds to the maximum test accuracy on the CIFAR10, SVHN, and MNIST datasets, as shown in Fig. 2. As expected, small values of μ cause the learned distributions to be too close in the latent space, leading to difficult class separability and poor classification performance. On the other hand, it can be noticed how, for more challenging datasets like CIFAR10 and SVHN, the classification accuracy gain ensured by GCCS increases for the chosen value $\mu_T = 70$. Specifically, we hypothesize that values around $\mu_T = 70$ ensure the best trade-off between the separability of the different distributions in the latent space and the enforceability of the target distributions during training. On the other hand, greater μ_T/σ_T ratios are more likely to cause instability during training due to harder constraints on the target distributions, resulting in worse classification accuracy. Our empirical findings lead us to the assumption that the μ_T

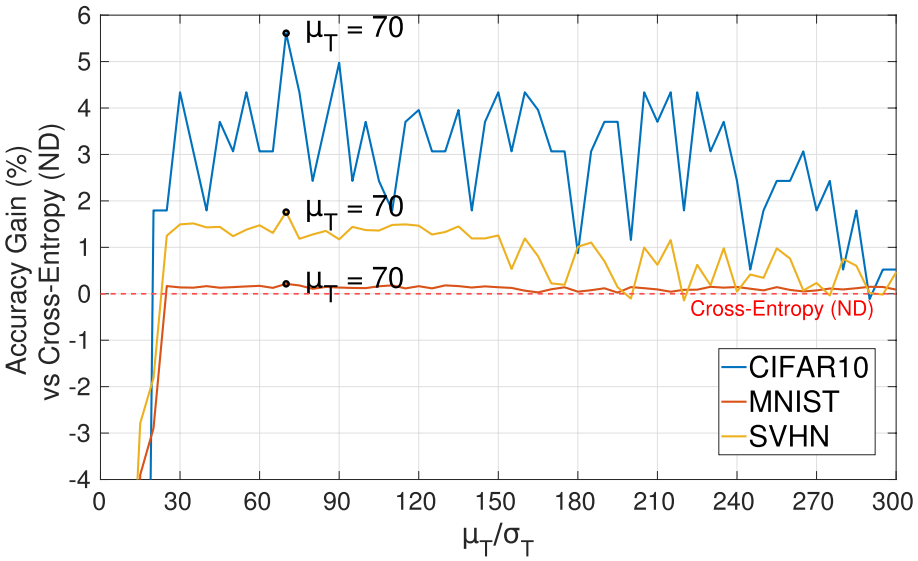


Fig. 2 Classification accuracy gain (%) for GCCS with respect to Cross-Entropy with ResNet-18, as a function of μ_T/σ_T ($\sigma_T = 1$), on the CIFAR10, MNIST, and SVHN datasets

value could be tuned by training just once on a specific dataset and the obtained value used also with datasets with the same number of classes, without the need for an exhaustive and expensive search based on multiple training experiments.

4.3 Kurtosis factor

In this section, we analyzed the effects of the Kurtosis factor in the loss formulation over the classification accuracy. We set $\mu_T = 70$ and investigate performance as a function of the balancing parameter λ, taking values in the range $\lambda \in [0, 1]$ on the CIFAR10, SVHN, and MNIST datasets. By looking at Table 1, one can observe a significant performance drop when the Kurtosis factor is not included in the loss (i.e., when $\lambda = 0$). For this reason, throughout the whole experimental evaluation, we set the value to $\lambda = 0.2$ as we experimentally verified it ensures the greatest class separability and hence classification accuracy.

4.4 Training batch size

Keeping fixed the value of the μ_T, σ_T, and λ parameters to $\mu_T = 70$, $\sigma_T = 1$, and $\lambda = 0.2$, we now investigate the effect of different training batch sizes on the accuracy of our GCCS method. Similarly to Table 1, Table 2 reports the GCCS classification accuracy as a function of the training batch size on the CIFAR10 and MNIST datasets. Table 2 shows that, for smaller batch sizes, the training does not converge to high accuracy. This happens because the batch size is too small to allow the network to learn distributions that are Gaussian and therefore well-separable in the latent space. On the other hand, as the training batch size increases, the accuracy does not significantly improve. While it might be possible to reach slightly better accuracy by further increasing the training batch size, it is worth noticing that, as the batch size increases, so does the complexity of the training, and with it the training time. For this reason, we choose to keep the training batch size equal to 200 for all our experiments, as it represents a sweet spot in the trade-off between high accuracy and problem size.

4.5 Features extraction and separation

Feature selection is of key importance for the classification accuracy and robustness of a neural network. Often in the literature, inter-class and intra-class separation are considered independently, while, as said, our approach yields high inter-class and intra-class separation at the same time.

Table 2 Classification accuracy (%) for GCCS with ResNet-18 as a function of the training batch size, evaluated on CIFAR10 and MNIST

Training Batch Size	50	100	200	400
CIFAR10	77.33	82.11	82.97	82.93
MNIST	95.14	99.35	99.58	99.56

4.5.1 Latent space analysis

When dealing with a multiclass classification problem, it is difficult to evaluate latent space features as dimensionality increases. Without loss of generality, Fig. 3 illustrates the distributions in the latent space $\mathbf{z}$ for three classes of MNIST, where points represent samples of a particular class (green, blue, and red respectively). Figure 3a–e show the distribution of the decision variable with different methods when no adversarial attack is applied. Figure 3a reports the latent space distribution for GCCS: one can observe that features are mapped onto distributions that are centered on the vertices of a simplex with a fixed variance and also that classes exhibit the same inter-class and intra-class distances. Moreover, the overlap between samples of different classes is negligible for GCCS, while it is not for the cross-entropy loss and the other considered methods, as in Fig. 3b–e. Further, GCCS ensures remarkably low intra-class variance as compared to the other cases, enabling much more robust decision boundaries. In fact, thanks to the GCCS loss that ensures both high separability and uniformity of the target distributions, we can observe the lack of a short path between neighboring distributions and the resulting lower misclassification rate with respect to competing methods, where instead class distributions are not uniformly optimized.

Further, while the figure is only meant to offer a qualitative insight into why GCCS significantly outperforms competing methods in the considered scenarios, it offers precious insight nonetheless. Specifically, we

Table 1 Classification accuracy (%) for GCCS with ResNet-18 as a function of λ (Eq. 2), evaluated on CIFAR10, MNIST, and SVHN

λ	0.0	0.1	0.2	0.3	0.4	0.5
CIFAR10	79.33	81.89	82.97	81.40	82.05	82.10
SVHN	94.10	95.06	95.58	94.82	94.89	95.32
MNIST	99.20	99.51	99.58	99.40	99.43	99.38

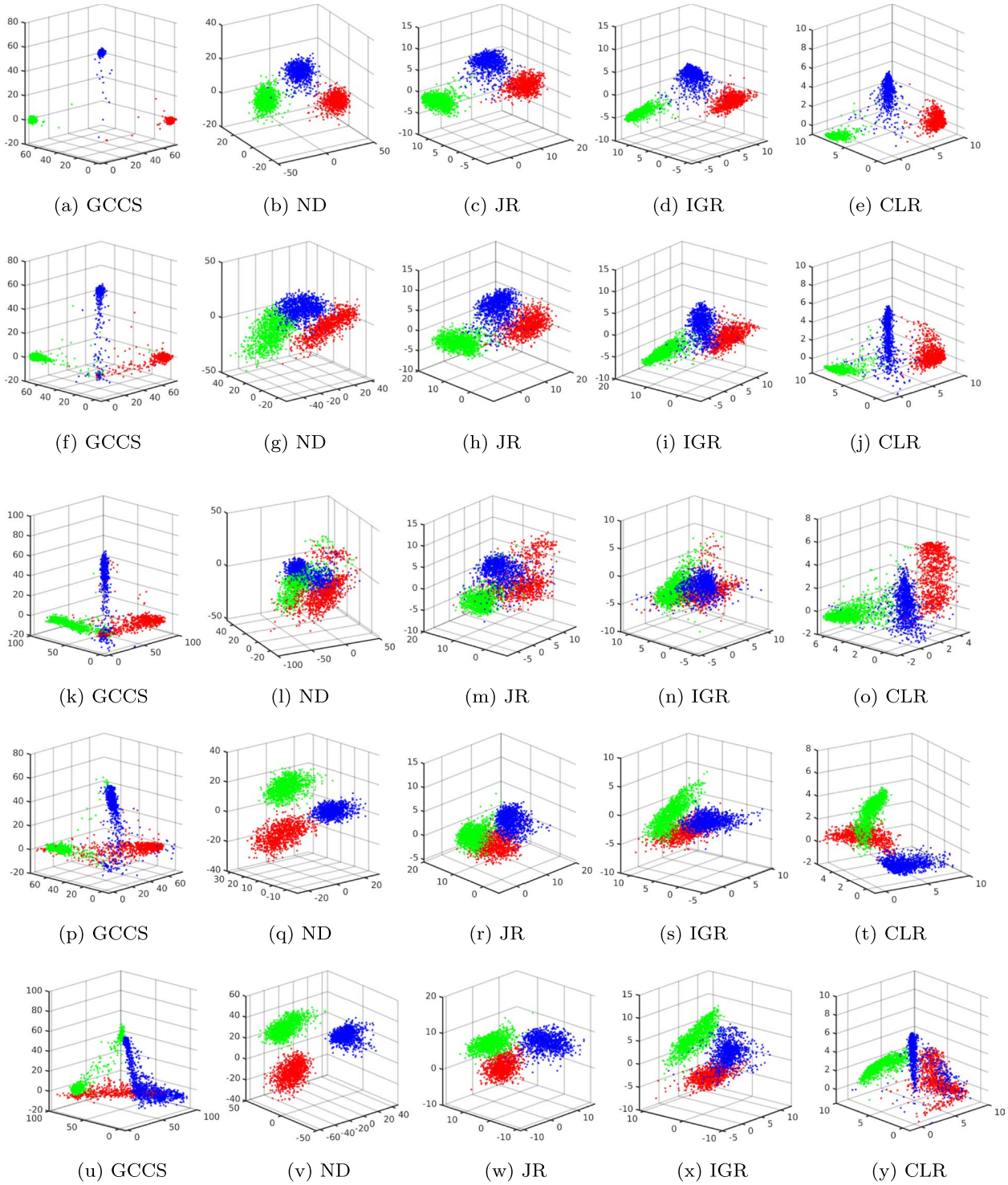


Fig. 3 Visual representation of the output distributions in the latent space on the three MNIST classes [0, 1, 9] (*red, green, and blue colors respectively*) without adversarial training for GCCS, cross-entropy (ND), Jacobian Regularization [53] (JR), Input Gradient Regularization [44] (IGR), and Cross Lipschitz Regularization [52] (CLR) methods : **a**–**e** when no adversarial attack is applied; **f**–**j** when applying FGSM; **k**–**o** when applying PGD (5 iterations); **p**–**t** when applying TGSM (5 iterations); **u**–**y** when applying JSMA (200 iterations, 1 pixel). To better understand the measure in which our method outperform the competitors, one should also consider the scale on the axis of each individual plot, which clearly show how the latent distributions for GCCS are significantly more distant in the latent space, hence separable

chose to visualize the latent distributions of MNIST classes [0, 1, 9] (red, green, and blue colors respectively) to offer a representation of how the learned distributions of potentially similar classes of digits (0 and 9) would behave in the latent space compared to the distributions of geometrically dissimilar classes (0 and 1). What can be observed from Fig. 3a compared for with Fig. 3b–e is that the learned distributions for similar digits in the GCCS case are not closer to each other than the distributions for the dissimilar ones. The same cannot be said for ND and the other competing methods. Instead, in these cases, the red and blue distributions (i.e., the learned distributions for the 0 and 9 classes respectively) appear to be closer to each other compared with the distance from the green distribution (i.e., class 1). Finally, we analyze the remaining Fig. 3f–y in Section 5.

4.5.2 Classification accuracy and inter-class/intra-class separation

We can now examine the quality of the features extracted by measuring the distance of each sample with respect to every other sample. We evaluate separability by measuring the average inter-class/intra-class distances in the latent space when no adversarial attack is performed. With n samples in the class A, n latent space feature vectors $\{z_1^A \cdots z_i^A \cdots z_j^A \cdots z_n^A\}$ are generated for each class, where vector z_i^A represents the i-th sample in class A. Hence, we define the intra-class distance of class A as:

$$d_{intra} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\left(z_i^A - z_j^A\right)^2}, \tag{4}$$

such that the distance between a sample and itself in the latent space is zero [76]. On the other hand, we can define the inter-class distance between n samples in class A and m samples in class B as:

$$d_{inter} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m}\left(z_i^A - z_j^B\right)^2}. \tag{5}$$

Finally, we can define the ratio of the inter-class and intra-class distance between two classes A and B as:

$$r_{AB} = d_{inter}/d_{intra}. \tag{6}$$

According to the r_{AB} definition, higher ratios correspond to higher robustness and lower misclassification rate thanks to greater separability and simpler decision boundaries. Table 3 shows the average r_{AB} ratio for all possible pairs of classes over different datasets and different methods, together with the corresponding confidence intervals.

It can be observed that GCCS leads to much higher rates than the cross-entropy loss and the considered competing methods when no attacks are carried out. This directly translates to excellent classification accuracy both for the *regular training* and *fine-tuning*, as reported in Table 4 which shows the maximum accuracy over multiple datasets for the different methods. In more detail, it can be seen that, in some cases, other techniques might even cause a drop in classification accuracy with respect to the standard cross-entropy loss function, especially for challenging datasets such as CIFAR-10 and CIFAR-100 where instead GCCS improves the plain cross-entropy loss testing accuracy.

Table 3 Ratio r_{AB} of inter-class to intra-class distance obtained through *regular-training* vs *fine-tuning* over different benchmark datasets with different competing techniques when no adversarial attack is performed

Method	MNIST ResNet-18	FMNIST ResNet-18	SVHN ResNet-18	CIFAR-10 ResNet-18	CIFAR-10 Shake-Shake-96	CIFAR-100 Shake-Shake-112
GCCS (regular training)	18.50 ± 1.45	7.64 ± 5.42	5.37 ± 0.65	2.89 ± 1.01	9.21 ± 3.52	2.34 ± 0.87
GCCS (fine-tuning)	18.11 ± 1.59	8.29 ± 5.69	7.33 ± 0.85	2.91 ± 1.22	9.57 ± 3.16	2.16 ± 0.51
No Defense (cross-entropy loss)	3.12 ± 0.71	3.13 ± 1.24	1.94 ± 0.21	1.71 ± 0.31	2.71 ± 0.48	1.62 ± 0.26
Jacobian Reg. (regular training) [53]	3.35 ± 1.01	3.64 ± 1.22	1.94 ± 0.21	2.03 ± 0.68	-	-
Jacobian Reg. (fine-tuning)[53]	4.09 ± 0.81	3.71 ± 1.24	2.43 ± 0.26	2.35 ± 0.63	-	-
Input Gradient Reg. (regular training) [44]	3.70 ± 0.92	2.71 ± 0.85	2.12 ± 0.24	1.57 ± 0.24	2.70 ± 0.55	1.63 ± 0.27
Input Gradient Reg. (fine-tuning) [44]	3.65 ± 0.97	3.18 ± 1.03	2.11 ± 0.27	1.68 ± 0.31	2.97 ± 0.59	1.60 ± 0.25
Cross Lipschitz (regular training) [52]	4.43 ± 1.07	4.44 ± 3.24	2.40 ± 0.29	1.91 ± 0.41	-	-
Cross Lipschitz (fine-tuning) [52]	6.72 ± 2.23	4.42 ± 3.09	2.62 ± 0.28	1.85 ± 0.31	-	-

Table 4 Maximum test accuracy obtained through *regular training* vs *fine-tuning* over different benchmark datasets with different competing techniques when no adversarial attack is performed

Method	MNIST ResNet-18	FMNIST ResNet-18	SVHN ResNet-18	CIFAR-10 ResNet-18	CIFAR-10 Shake-Shake-96	CIFAR-100 Shake-Shake-112
GCCS (regular training)	99.58	92.69	94.20	82.97	96.19	76.53
GCCS (fine-tuning)	99.64	93.83	95.58	81.52	97.06	77.48
No Defense (cross-entropy loss)	99.35	91.91	94.12	78.59	95.78	76.30
Jacobian Reg. (regular training) [53]	98.99	91.79	94.11	70.09	-	-
Jacobian Reg. (fine-tuning) [53]	98.53	92.43	93.54	82.09	-	-
Input Gradient Reg. (regular training) [44]	97.98	88.45	93.77	78.32	96.50	74.89
Input Gradient Reg. (fine-tuning) [44]	99.11	92.55	93.17	76.15	96.90	75.68
Cross Lipschitz (regular training) [52]	96.78	92.54	91.42	80.10	-	-
Cross Lipschitz (fine-tuning) [52]	98.77	92.41	93.50	79.39	-	-

The missing entries in the tables for the CIFAR-10 and CIFAR-100 datasets account for the cases in which it was not possible to apply the particular adversarial defense method because of very high computational requirements.

4.5.3 F-score

Further, we assess the performance of our method by computing the *F*-score directly from the confusion matrix and then compare the results with state-of-the-art methods. Also in this second set of experiments, we only deal with the case in which no adversarial attack is performed. The *F*-score, defined as the harmonic mean of precision and recall, is considered to be a more precise measure of the classification accuracy for uneven datasets such as SVHN. Table 5 illustrates the *F*-scores for all the considered datasets, and it shows that GCCS outperforms the state-of-the-art also according to this metric. As anticipated, the gap is more significant for SVHN than for the other datasets.

5 Adversarial robustness evaluation

To assess the robustness of our proposed method, we follow the guidelines proposed in [13, 35, 36] to provide a rigorous comparison with existing methods when applying different adversarial attacks. First, we consider gradient-based attacks applied in a white-box scenario, e.g., assuming the attacker has access to the model parameters and the loss function. Specifically, we carry out large-scale experimentation with non-targeted (DeepFool, FGSM, and PGD) and targeted (TGSM, JSMA) gradient-based attacks. Then, we evaluate robustness against the gradient-free SPSA attack, which does not exploit knowledge of the gradients with respect to the loss. In all the experiments, we have employed adversarial attacks on a batch of images of a given size equal to 100. In other words, we simulate an attacker trying to

Table 5 *F*-score obtained through *regular training* vs *fine-tuning* over different benchmark datasets with different competing techniques when no adversarial attack is performed

Method	MNIST ResNet-18	FMNIST ResNet-18	SVHN ResNet-18	CIFAR-10 ResNet-18	CIFAR-10 Shake-Shake-96	CIFAR-100 Shake-Shake-112
GCCS (regular training)	99.58	92.66	94.17	82.93	96.18	76.49
GCCS (fine-tuning)	99.64	93.80	95.28	81.46	97.05	77.72
No Defense (cross-entropy loss)	99.35	91.88	93.7	78.59	95.77	76.55
Jacobian Reg. (regular training) [53]	98.98	91.73	93.68	69.32	-	-
Jacobian Reg. (fine-tuning) [53]	98.51	92.41	93.24	82.2	-	-
Input Gradient Reg. (regular training) [44]	97.96	88.51	93.26	78.70	96.58	76.24
Input Gradient Reg. (fine-tuning) [44]	99.08	92.38	92.62	76.39	96.98	75.59
Cross Lipschitz (regular training) [52]	96.64	92.52	90.55	80.15	-	-
Cross Lipschitz (fine-tuning) [52]	98.75	92.39	92.97	79.22	-	-

maximize the loss over the whole batch with respect to the true class.

We evaluate classification accuracy as a function of a tunable parameter ϵ that indicates how strong is the applied attack, such that $\|\mathbf{n}\|_\infty / \|\mathbf{x}\|_\infty \leq \epsilon$, where $\mathbf{n}$ is the added noise vector, and $\mathbf{x}$ is the input signal. In more detail, given an attack model $A_\epsilon(\mathbf{x})$ dependent on the input $\mathbf{x}$, a classifier C, and the expected output $\mathbf{y}$, the accuracy of the classifier against the attack is computed as:

$$Acc\ (C, A_\epsilon(\mathbf{x})) = \frac{1}{N} \sum_{i=1}^{N} [C(A_\epsilon(x_i)) = y_i]. \tag{7}$$

In more detail, we first evaluate the robustness of GCCS against attacks without employing any adversarial training. Our goal is to show that the proposed regularization of the latent space offers an advantage with respect to adversarial training since it is not trained for a specific attack. Hence, the proposed technique performs better than other defense methods even when adversarial training is employed at the same time. Then, since actually the proposed method and adversarial training are not mutually exclusive, we apply GCCS together with adversarial training and show even greater combined robustness. To perform adversarial training, as explained in the following, we consider FGSM perturbations with $\epsilon = 0.03$, averaging the plain GCCS loss with the adversarial one as done in the standard approaches [44]. In particular, we employ adversarial training by fine-tuning an already trained model for a total 100 epochs. Without loss of generality, we fine-tune models that have been trained with the *regular training* setting (as defined in Section 4), but our findings can be extended straightforwardly also to the *fine-tuning* configuration.

5.1 Gradient-based attacks

5.1.1 Non-targeted attacks

The general goal of non-targeted attacks is to cause a misclassification in labeling the input, so that the output decision differs from the actual class the input belongs to.

DeepFool attack: We start by evaluating GCCS performance when applying the gradient-based DeepFool attack [32]. This particular attack uses a first-order approximation of the decision boundaries, altering the input towards the closest decision boundary in order to cause misclassification. When dealing with DeepFool, robustness is usually measured with ρ, which is a parameter that indicates the average robustness against the attack. ρ is dependent on the estimated minimal perturbation caused by DeepFool and also on the cardinality of the considered images dataset, as explained in [32]. Table 6 reports the robustness of the considered methods over multiple datasets, showing that GCCS provides a very significant improvement in robustness compared to different methods, both for regular and adversarial training. The greater robustness obtained by GCCS can be explained by the fact that the learned inter-class boundaries are equidistant, so it is more difficult for the attack to efficiently alter them and cause misclassification.

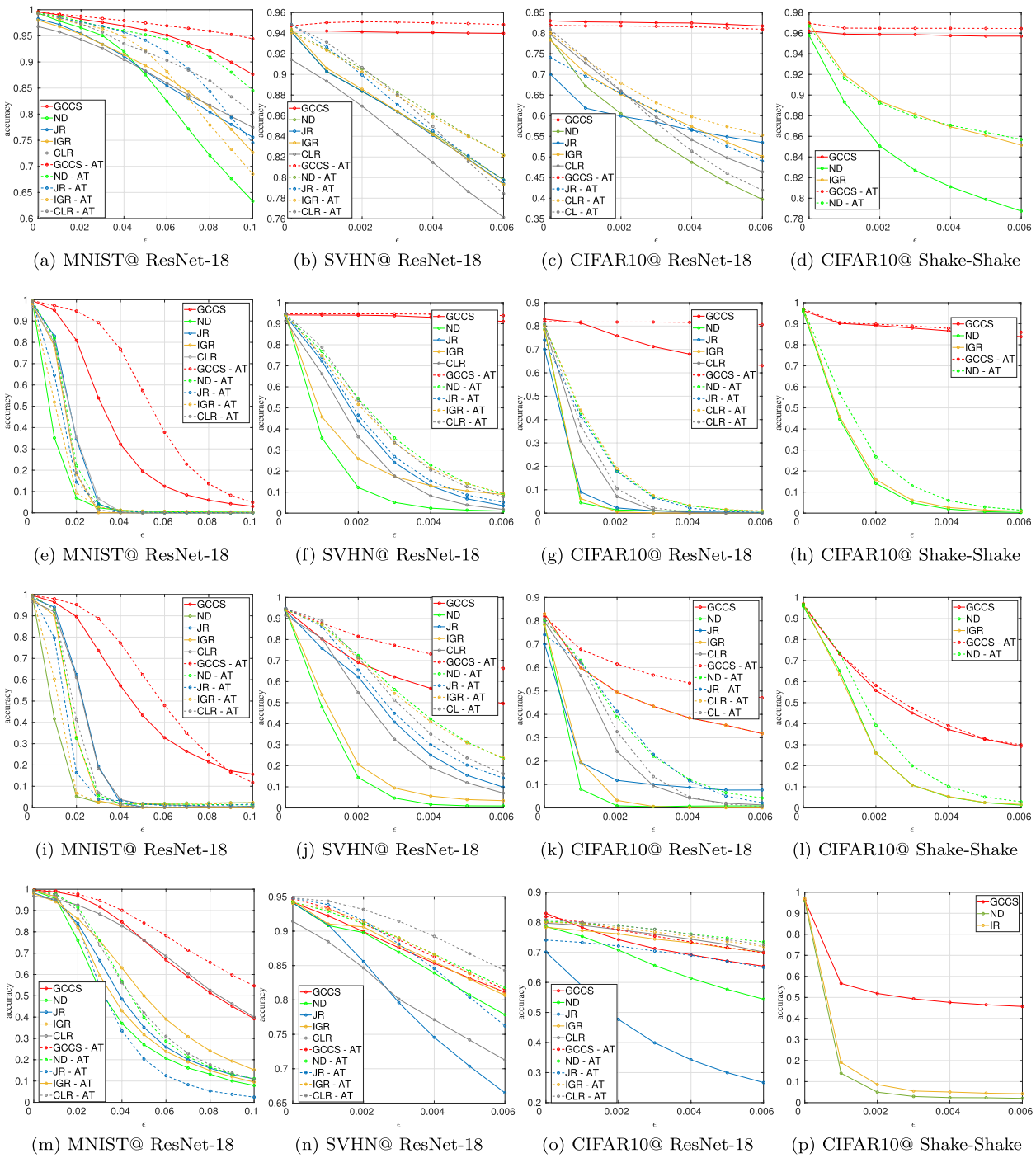
FGSM and PGD attacks: Further, we analyze the robustness of GCCS against the FGSM [33] and PGD [30] gradient-based attacks on MNIST, SVHN, CIFAR-10, and CIFAR-100. While the former is a single-step attack that adds noise in the direction of the gradient of the loss function with respect to the input data, the latter is an iterative version of FGSM in which noise is added in multiple iterations, resulting in the strongest adversarial attack that exploits first-order local information about the trained model. In the PGD attack, the number of iterations K plays an important role in defining the strength

Table 6 Robustness to DeepFool attack obtained through *regular* vs *adversarial training* over different benchmark datasets with different competing techniques, measured with ρ [32]

Method	MNIST ResNet-18	FMNIST ResNet-18	SVHN ResNet-18	CIFAR-10 ResNet-18	CIFAR-10 Shake-Shake-96	CIFAR-100 Shake-Shake-112
GCCS (regular training)	3.79	4.98	1.85	2.53	0.46	0.0034
GCCS (adversarial training)	3.93	4.91	3.05	2.42	0.94	0.0051
No Defense (cross-entropy loss)	1.83	0.55	1.73	1.43	0.27	0.0011
No Defense (adversarial training)	1.10	0.45	0.52	0.48	0.06	0.0013
JR (regular training) [53]	0.55	0.31	1.73	0.32	-	-
JR (adversarial training) [53]	1.60	0.39	0.53	0.17	-	-
IGR (regular training) [44]	0.48	0.37	1.54	0.83	0.05	0.0013
IGR (adversarial training) [44]	1.79	0.46	0.58	0.46	0.06	0.0017
CLR (regular training) [52]	0.31	0.29	0.33	0.26	-	-
CLR (adversarial training) [52]	0.85	0.39	0.36	0.26	-	-

of the attack and the time needed to generate the corresponding adversarial examples. In this work, we consider a 5-iteration PGD attack (PGD-5) as done in [49, 77, 78].

In the case of MNIST, we set $0 \leq \epsilon \leq 0.10$, while we chose $0 \leq \epsilon \leq 0.06$ for SVHN, CIFAR-10, and CIFAR-100. Figure 4 a–d show that even the regularly trained GCCS model enjoys higher classification accuracy with respect to competing methods over all the datasets and through all the ϵ range; indeed, in many cases and for sensible values of ϵ, GCCS without



(a) MNIST@ ResNet-18 (b) SVHN@ ResNet-18 (c) CIFAR10@ ResNet-18 (d) CIFAR10@ Shake-Shake

(e) MNIST@ ResNet-18 (f) SVHN@ ResNet-18 (g) CIFAR10@ ResNet-18 (h) CIFAR10@ Shake-Shake

(i) MNIST@ ResNet-18 (j) SVHN@ ResNet-18 (k) CIFAR10@ ResNet-18 (l) CIFAR10@ Shake-Shake

(m) MNIST@ ResNet-18 (n) SVHN@ ResNet-18 (o) CIFAR10@ ResNet-18 (p) CIFAR10@ Shake-Shake

Fig. 4 Test accuracy as a function of ϵ under gradient-based adversarial attacks, both targeted and untargeted (AT indicates also adversarial training is employed): **a–d** FGSM attack; **e**–**h** PGD attack (5 iterations); **i**–**l** TGSM attack (5 iterations); **m–p** JSMA attack (200 iterations, 1 pixel)

adversarial training is almost as good as GCCS with adversarial training. This highlights that, despite not explicitly enforcing robustness to a specific attack, GCCS achieves excellent robustness to *any* tested attack (dotted lines). The performance gap is especially strong in the case of the challenging CIFAR datasets.

In order to gain a better understanding of why the proposed method works much better than the others, we refer to Fig. 3f–j that shows a visual representation of the distributions in the latent space *after* an FGSM attack ($\epsilon = 0.2$) has been applied. In the GCCS case (Fig. 3f), even if the tails of the output distributions become heavier, the classes are still clearly separated, allowing for improved classification accuracy and robustness. Our findings are valid also in the case of the PGD-5 attack, as highlighted in Fig. 4e–h. When tampered with this very strong attack, the accuracy ensured by competing methods very quickly collapses to low values, while GCCS enables robust classification even for high ϵ. Similarly to FGSM, Fig. 3k–o depict the latent space after the PGD-5 attack is applied. The effect of the attack is much more prominent compared to FGSM to the extent that the class distributions are significantly overlapped for all the competing methods, and classification fails when no defense against adversarial perturbations is applied. Our proposed GCCS method leads instead to remarkably lower misclassification rates and more separable distributions. As a matter of example, for $\epsilon \geq 0.04$ GCCS is still able to reach around 85% classification accuracy on CIFAR-10, while other methods collapse to random-guess performance.

5.1.2 Targeted attacks

In this second case, the attack works by creating an adversarial sample that is crafted to be misclassified to the desired target class, such that $C(x^{adv}) = y^*$. Again, we provide robustness curves for both regular and adversarial training as in Section 5.

TGSM attack: In the gradient-based TGSM [30] attack, the input samples are perturbed by adding noise in the direction of the negative gradient with respect to a selected target class, such that the targeted output class is y_{l+1} when the true one is y_l. Figure 4i–l present the results for a challenging 5-iteration TGSM attack (TGSM-5) over the considered datasets.

It can be observed from Fig. 4i–l that GCCS yields significantly higher performance compared to the other methods, throughout different datasets and with different attack strength ϵ both for the regular and adversarial training. Figure 3p–t show a visual representation of the distributions in the latent space *after* the TGSM-5 attack with $\epsilon = 0.02$ is applied. Figure 3q clearly shows the effectiveness of the attack when no defense mechanism is employed. Specifically, it can be observed that the distributions' center of mass shifted their position in the latent space to fall exactly onto the position of the distribution that has been chosen by the targeted TGSM attack we are considering. In other words, the output distributions are shifted so as to replace the output distribution of the next class, leading to very strong misclassification. In the GCCS case instead, as shown in Fig. 3p, it is much more difficult for the attacker to successfully swap the positions of the target distributions with the ones of the neighboring classes, as testified by the distributions in the latent space, which partially spread out as per the effect of the targeted attack, but at the same time they preserve their center of mass in the same positions with respect to the no-attack case (Fig. 3a) even after the considered amount of TGSM iterations. This indeed ensures improved robustness and separability for our GGCS method.

JSMA attack: The other gradient-based targeted attack we consider is JSMA [31], which consists of iteratively computing the Jacobian matrix of the network function to form a saliency map. This map is used at every iteration to choose which pixels to tamper with so that the likelihood of changing the output class towards a selected one is increased. In our case, we consider JSMA-200 to make the attack more challenging for better performance comparison, with a 1-pixel saliency map. Similarly to the TGSM case, Fig. 4m–p shows the classification accuracy for increasing attack strength ϵ. The proposed method confirms its robustness even against the JSMA attack, achieving better robustness than other methods, especially on the challenging CIFAR datasets. Once again, Fig. 3u–y show how resilient the proposed GCCS method is against the considered adversarial attack as compared to competing methods where the position of the classes' distributions after the JSMA attack is completely swapped, as already previously observed for the TGSM targeted attack.

5.2 Gradient-free SPSA attack

Finally, we evaluate robustness also against the gradient-free Simultaneous Perturbation Stochastic Approximation attack (SPSA [34]) based on [79], which is a method to approximate analytic gradients with finite difference estimates in random directions. SPSA is an iterative attack whose strength is determined by three parameters: the perturbation strength ϵ as defined in Section 5, the number of attack iterations (*IT*), and batch size (*BS*), which indicated the number of finite difference estimates used before applying each gradient estimate. We tested the robustness of our GCCS method against the traditional cross-entropy loss (no defense) with and without adversarial training by measuring test accuracy as a function of ϵ on the CIFAR10 dataset.

First, we analyzed the robustness by varying the IT and BS parameters, and keeping fixed the adversarial perturbation strength to $\epsilon = 0.02$. In general, Uesato et al. [34] show that the higher the values, the strongest the attack. However, the computational cost of applying SPSA increases very strongly for high BS values. For this reason, we are interested in observing how robustness decreases by varying the strength of the attack for different (IT, BS) combinations. Table 7 shows that, roughly after $IT = 50$ iterations and for $BS > 1024$, test accuracy does not significantly decrease for the higher (IT, BS) combinations appearing on the right side of the table. Hence, we assume our working point to be ($IT = 50, BS = 1024$) without loss of generality, as also confirmed by [34] which shows that, in many cases, the maximum strength of SPSA is reached even when doing fewer than ten iterations of the attack.

Figure 5 shows how the proposed method exhibits much greater robustness than the traditional cross-entropy loss, even when applying adversarial training to the models. Since SPSA is generally considered to be the most effective gradient-free adversarial attack, we are able to conclude that GCCS is robust also against attacks carried out without the knowledge of the gradients.

Table 7 Test accuracy on CIFAR10 (%) when attacking the GCCS (bold) and cross-entropy (No defence) models with SPSA [34] with parameters (BS, IT) and fixed perturbation strength $\epsilon = 0.02$

IT / BS	512	1024	2048	8192
50	**58.57**	**57.89**	**57.49**	**57.01**
	22.85	21.81	21.07	20.46
100	**58.32**	**57.80**	**57.29**	**56.89**
	22.74	21.70	21.01	20.39

6 Training time

To complete the evaluation of the proposed method, we also considered computational requirements. Looking at Table 8, one can observe that the impact of GCCS in

Table 8 Training time for *regular* vs *adversarial* training over the MNIST dataset (1800 epochs). The numbers in brackets indicate the training time percentage increase caused by the considered defense method as compared to the corresponding plain cross-entropy training case (No Defense)

Applied defense method	Training time (minutes)
No Defense (cross-entropy loss)	312
No Defense (adversarial training)	687
GCCS (regular training)	318 (+1.92%)
GCCS (adversarial training)	694 (+1.02%)
JR (regular training) [53]	512 (+64.10%)
JR (adversarial training) [53]	1073 (+56.18%)
IGR (regular training) [44]	617 (+97.75%)
IGR (adversarial training) [44]	1297 (+88.79%)
CLR (regular training) [52]	687 (+120.2%)
CLR (adversarial training) [52]	1456 (+111.9%)

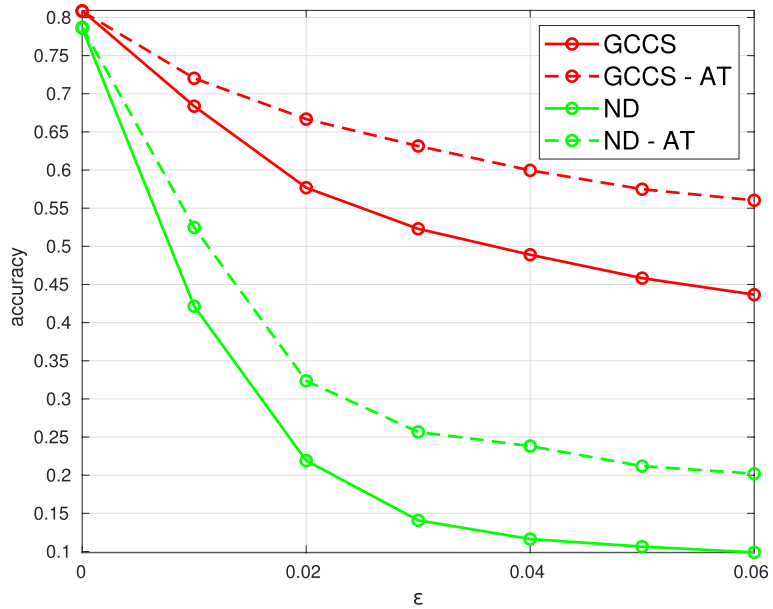


Fig. 5 Test accuracy (%) for GCCS and cross-entropy (ND) on CIFAR10 with and without AT when applying SPSA with parameters ($IT = 50, BS = 1024$), as a function of ϵ

terms of training time overhead is negligible with respect to standard cross-entropy training (No Defense), as it is in the order of a few added minutes when training for a considerable number of epochs (1800). As highlighted in works such as [80], this feature is of great importance because it allows to efficiently improve the robustness of any deep architecture (even pre-trained) at the expense of a negligible additional training effort.

7 Conclusions

We have presented an approach that employs a loss function promoting class separability and robustness by learning a mapping of the decision variables onto Gaussian distributions. The proposed GCCS loss enjoys high classification accuracy and robustness against adversarial attacks, with negligible training overhead. Experiments on different multi-class datasets show excellent performance of the classifiers trained using the GCCS loss, outperforming existing state-of-the-art methods both when used to train from scratch and when applied as a fine-tuning step on pre-trained networks. Also, performance is investigated both for targeted and non-targeted gradient-based adversarial attacks in a white-box scenario.

The analysis of the distributions in the latent space for the proposed GCCS method shows that the different classes tend to remain well separated even in the presence of gradient-based targeted attacks, whereas a similar attack strength invariably mixes the distributions achieved by competing methods.

Also, we have shown that GCCS, when used in combination with adversarial training, can further improve the model robustness while maintaining high classification accuracy. Finally, we illustrated how the proposed method ensures greater robustness when employing the challenging gradient-free SPSA [34] attack, which does not rely on the loss gradient.

Abbreviations

GCCS	Gaussian Class-Conditional Simplex loss
AT	Adversarial training
CLR	Cross-Lipschitz Regularization
IGR	Input Gradient Regularization
JR	Jacobian Regularization
ND	No Defense (Cross-Entropy Loss)
SPSA	Simultaneous Perturbation Stochastic Approximation attack
IT	SPSA attack iterations
BS	SPSA attack batch-size

Acknowledgements

This research has been carried out in collaboration with Sony R &D Center Europe, Stuttgart, Laboratory 1.

Authors' contributions

AA performed the initial set of experiments with the proposed loss and drafted the first version of the manuscript. AM tested the robustness of the approach against the SPSA attack. AM, TB, and EM updated the manuscript to the final version. All authors read and approved the final manuscript.

Availability of data and materials

Model evaluation was performed on publicly available datasets such as MNIST [26], FMNIST [27], SVHN [28], CIFAR10, and CIFAR100 [29]. The implementation of the considered attacks is available following the links in the original papers (TGSM [30], JSMA [31], DeepFool [32], FGSM [33], PGD [16], and SPSA [34]).

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 11 October 2022 Accepted: 11 February 2023
Published online: 10 March 2023

References

1. A. Kirzhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks. Adv. Neural. Inf. Process. Syst, **25**, 1097–1105 (2012)
2. L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, R. Fergus, Regularization of neural networks using dropconnect. In International conference on machine learning (2013) pp. 1058–1066. PMLR
3. J. Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, & E. Shechtman, Toward multimodal image-to-image translation. Adv. Neural. Inf. Process. Syst, **30** (2017)
4. A. Gonzalez-Garcia, J. Van De Weijer, Y. Bengio, Image-to-image translation for cross-domain disentanglement. Adv. Neural. Inf. Process. Syst. **31**, 1287–1298 (2018)
5. M. Uricár, P. Krizek, D. Hurych, I. Sobh, S. Yogamani, P. Denny,Yes, we gan: Applying adversarial techniques for autonomous driving. arXiv preprint. (2019). arXiv:1902.03442
6. K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, ... D. Song, Robust physical-world attacks on deep learning visual classification. In Proceedings of the IEEE conference on computer vision and pattern recognition. (2018), pp. 1625–1634
7. S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L., Beam, I. S. Kohane, Adversarial attacks on medical machine learning. Science, **363**(6433), 1287-1289 (2019)
8. T. Zheng, C. Chen, K. Ren, Distributionally adversarial attack. In Proceedings of the AAAI Conference on Artificial Intelligence, **33**(01), 2253–2260) (2019)
9. H. Zhang, H. Chen, Z. Song, D. Boning, I.S. Dhillon, C.J. Hsieh, The limitations of adversarial training and the blind-spot attack. (2019). arXiv preprint arXiv:1901.04684
10. A. Raghunathan, J. Steinhardt, P. Liang, Certified defenses against adversarial examples. (2018). arXiv preprint arXiv:1801.09344
11. E. Wong, Z. Kolter, Provable defenses against adversarial examples via the convex outer adversarial polytope. In International conference on machine learning. (2018), pp. 5286–5295. PMLR
12. M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, S. Jana, Certified robustness to adversarial examples with differential privacy. In 2019 IEEE Symposium on Security and Privacy (SP). (2019), p. 656–672. IEEE
13. Y. Dong, Q.A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, J. Zhu, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Benchmarking adversarial robustness on image classification (2020), pp. 321–331
14. A. Fawzi, S. Moosavi-Dezfooli, P. Frossard, The robustness of deep networks: A geometrical perspective. IEEE Signal Process. Mag. **34**(6), 50–62 (2017). https://doi.org/10.1109/MSP.2017.2740965
15. M. Khoury, D. Hadfield-Menell, On the geometry of adversarial examples. (2018). arXiv preprint arXiv:1811.00525
16. A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks. (2017). arXiv preprint arXiv:1706.06083
17. F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, P. McDaniel, Ensemble adversarial training: Attacks and defenses. (2017). arXiv preprint arXiv:1705.07204

18. O. Poursaeed, I. Katsman, B. Gao, S. Belongie, Generative adversarial perturbations. In Proceedings of the IEEE conference on computer vision and pattern recognition. (2018), pp. 4422–4431
19. S. M.Moosavi-Dezfooli, A. Fawzi, J. Uesato, P. Frossard, Robustness via curvature regularization, and vice versa. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019), pp. 9078–9086
20. C. Summers, M.J. Dinneen, Improved adversarial robustness via logit regularization methods. (2019). arXiv preprint arXiv:1906.03749
21. R. Pinot, L. Meunier, A. Araujo, H. Kashima, F. Yger, C. Gouy-Pailler, J. Atif, Theoretical evidence for adversarial robustness through randomization. Adv. Neural. Inf. Process. Syst, **32**, pp. 11838–11848 (2019)
22. Y. Carmon, A. Raghunathan, L. Schmidt, J.C. Duchi, P.S. Liang, in *Advances in Neural Information Processing Systems*. Unlabeled data improves adversarial robustness (2019), pp. 11190–11201
23. A. Araujo, L. Meunier, R. Pinot, B. Negrevergne, Robust neural networks using randomized adversarial training. (2019). arXiv preprint arXiv:1903.10219
24. R. Pinot, R. Ettedgui, G. Rizk, Y. Chevaleyre, J. Atif, Randomization matters. How to defend against strong adversarial attacks. (2020). arXiv preprint arXiv:2002.11565
25. A. Ali, A. Migliorati, T. Bianchi, E. Magli, Beyond cross-entropy: learning highly separable feature distributions for robust and accurate classification. In 2020 25th International Conference on Pattern Recognition (ICPR). (2021), pp. 9711–9718). IEEE
26. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition. Proc. IEEE, **86**(11), 2278–2324 (1998)
27. H. Xiao, K. Rasul, R. Vollgraf, Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. (2017). arXiv preprint arXiv:1708.07747
28. Y. Netzer, T. Wang, A. Coates, A., Bissacco, B. Wu, A. Y. Ng, Reading digits in natural images with unsupervised feature learning. (2011)
29. A. Krizhevsky, G. Hinton et al., Learning multiple layers of features from tiny images. Technical report, Citeseer, (2009)
30. A. Kurakin, I. Goodfellow, S. Bengio, Adversarial examples in the physical world. (2016). arXiv preprint arXiv:1607.02533
31. N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, A. Swami, The limitations of deep learning in adversarial settings. In 2016 IEEE European symposium on security and privacy (EuroS&P). (2016), pp. 372–387. IEEE
32. S. M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition. (2016), pp. 2574–2582
33. C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks. (2013). arXiv preprint arXiv:1312.6199
34. J. Uesato, B. O'donoghue, P. Kohli, A. Oord, Adversarial risk and the dangers of evaluating against weak attacks. In International Conference on Machine Learning. (2018), pp. 5025-5034. PMLR
35. N. Carlini, D. Wagner, Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the 10th ACM workshop on artificial intelligence and security. (2017), pp. 3–14
36. N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, A. Kurakin, On evaluating adversarial robustness. (2019). arXiv preprint arXiv:1902.06705
37. N. Dalvi, P. Domingos, S. Sanghai, D. Verma, Adversarial classification. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. (2004), pp. 99–108
38. D. Lowd, C. Meek, Adversarial learning. In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. (2005), pp. 641–647
39. M. Barreno, B. Nelson, A. D. Joseph, J. D. Tygar, The security of machine learning. Mach. Learn, **81**, 121–148 (2010)
40. A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, A. Madry, in *Advances in Neural Information Processing Systems*, Adversarial examples are not bugs, they are features (2019)
41. J. Cohen, E. Rosenfeld, Z. Kolter, Certified adversarial robustness via randomized smoothing. In international conference on machine learning. (2019), pp. 1310–1320. PMLR
42. M. Sharif, S. Bhagavatula, L. Bauer, M. K. Reiter, Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In Proceedings of the 2016 acm sigsac conference on computer and communications security. (2016), pp. 1528-1540
43. A. Athalye, L. Engstrom, A. Ilyas, K. Kwok, Synthesizing robust adversarial examples. In International conference on machine learning. (2018), pp. 284–293. PMLR
44. A. Ross, F. Doshi-Velez, Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In Proceedings of the AAAI Conference on Artificial Intelligence **32**(1), (2018)
45. T.B. Brown, D. Mané, A. Roy, M. Abadi, J. Gilmer, Adversarial patch. (2017). arXiv preprint arXiv:1712.09665
46. S. Bhambri, S. Muku, A. Tulasi, A. Balaji Buduru, A survey of black-box adversarial attacks on computer vision models. (2019). arXiv
47. S. Lee, H. Lee, S. Yoon, Adversarial vertex mixup: Toward better adversarially robust generalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020), pp. 272–281
48. A. Athalye, N. Carlini, D. Wagner, Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In International conference on machine learning. (2018), pp. 274–283. PMLR
49. A. Shafahi, W.R. Huang, C. Studer, S. Feizi, T. Goldstein, Are adversarial examples inevitable? (2018). arXiv preprint
50. D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, A. Madry, Robustness may be at odds with accuracy. (2018). arXiv preprint arXiv:1805.12152.
51. D. Stutz, M. Hein, B. Schiele, Disentangling adversarial robustness and generalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019), pp. 6976–6987
52. M. Hein, M. Andriushchenko, Formal guarantees on the robustness of a classifier against adversarial manipulation. Advances in neural information processing systems, **30**, 6976–6987 (2017)
53. D. Jakubovitz, R. Giryes, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Improving dnn robustness to adversarial attacks using jacobian regularization (2018), pp. 514–529
54. J. Hoffman, D.A. Roberts, S. Yaida, Robust learning with jacobian regularization. (2019). arXiv preprint arXiv:1908.02729
55. I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples. (2014). arXiv preprint arXiv:1412.6572
56. R. Huang, B. Xu, D. Schuurmans, C. Szepesvári, Learning with a strong adversary. (2015). arXiv preprint arXiv:1511.03034
57. N. Papernot, P. McDaniel, X. Wu, S. Jha, A. Swami, Distillation as a defense to adversarial perturbations against deep neural networks. In 2016 IEEE symposium on security and privacy (SP). (2016), pp. 582–597. IEEE
58. Y. Liu, X. Chen, C. Liu, D. Song, Delving into transferable adversarial examples and black-box attacks. (2016). arXiv preprint
59. S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, Universal adversarial perturbations. In Proceedings of the IEEE conference on computer vision and pattern recognition. (2017), pp. 1765–1773.
60. A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, ... T. Goldstein, Adversarial training for free! Adv. Neural. Inf. Process. Syst, **32**, (2019)
61. D.P. Kingma, M. Welling, Auto-encoding variational bayes. (2013). arXiv preprint arXiv:1312.6114
62. A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, B. Frey, Adversarial autoencoders. (2015). arXiv preprint arXiv:1511.05644
63. J. Ye, S. Ji, *Discriminant analysis for dimensionality reduction: An overview of recent developments. Biometrics: Theory, Methods, and Applications* (Wiley-IEEE Press, New York, 2010)
64. A. Stuhlsatz, J. Lippel, T. Zielke, Feature extraction with deep neural networks by a generalized discriminant analysis. IEEE Trans. Neural Netw. Learn. Syst. **23**(4), 596–608 (2012)
65. M. Dorfer, R. Kelz, G. Widmer, Deep linear discriminant analysis. (2015). arXiv preprint arXiv:1511.04707
66. M. Testa, A. Ali, T. Bianchi, E. Magli, Learning mappings onto regularized latent spaces for biometric authentication. In 2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP). (2019), pp. 1-6. IEEE
67. Ali, A., Testa, M., Bianchi, T., & Magli, E. (2019). Authnet: Biometric authentication through adversarial learning. In 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP) (pp. 1–6). IEEE.
68. Ali, A., Testa, M., Bianchi, T., & Magli, E. (2020). Biometricnet: deep unconstrained face verification through learning of metrics regularized onto gaussian distributions. In Computer Vision–ECCV 2020: 16th European

Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16 (pp. 133–149). Springer International Publishing.
69. W. Wan, Y. Zhong, T. Li, J. Chen, Rethinking feature distribution for loss functions in image classification. In Proceedings of the IEEE conference on computer vision and pattern recognition. (2018), pp. 9117–9126
70. T. Pang, K. Xu, Y. Dong, C. Du, N. Chen, J. Zhu, Rethinking softmax cross-entropy loss for adversarial robustness. (2019). arXiv preprint arXiv:1905.10626
71. D. N. Joanes, C. A. Gill, Comparing measures of sample skewness and kurtosis. J. R. Stat. Soc.: Series D (The Statistician), **47**(1), 183–189 (1998)
72. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770–778).
73. X. Gastaldi, Shake-shake regularization. (2017). arXiv preprint arXiv:1705.07485
74. A. Gotmare, N.S. Keskar, C. Xiong, R. Socher, A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. (2018). arXiv preprint arXiv:1810.13243
75. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res, **15**(1), 1929–1958 (2014)
76. Y. Luo, Y. Wong, M. Kankanhalli, Q. Zhao, G-softmax: improving intraclass compactness and interclass separability of features. IEEE Trans. Neural Netw. Learn. Syst, **31**(2), 685–699 (2019)
77. T. Zheng, C. Chen, K. Ren, Is pgd-adversarial training necessary? Alternative training via a soft-quantization network with noisy-natural samples only. (2018). arXiv preprint arXiv:1810.05665
78. T. Davchev, T. Korres, S. Fotiadis, N. Antonopoulos, S. Ramamoorthy, An empirical evaluation of adversarial robustness under transfer learning. (2019). arXiv preprint arXiv:1905.02675
79. J.C. Spall et al., Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Trans. Autom. Control **37**(3), 332–341 (1992)
80. H. Zhang, Y. Yu, J. Jiao, E.P. Xing, L.E. Ghaoui, M.I. Jordan, Theoretically principled trade-off between robustness and accuracy. (2019). arXiv preprint arXiv:1901.08573

Publisher's Note