On the Minimal Adversarial Perturbation for Deep Neural Networks with Provable Estimation Error

Fabio Brau, Giulio Rossolini
Alessandro Biondi, *Member, IEEE* and Giorgio Buttazzo, *Fellow, IEEE*
Department of Excellence in Robotics and AI, Scuola Superiore Sant'Anna, Pisa, Italy

Abstract—Although Deep Neural Networks (DNNs) have shown incredible performance in perceptive and control tasks, several trustworthy issues are still open. One of the most discussed topics is the existence of adversarial perturbations, which has opened an interesting research line on provable techniques capable of quantifying the robustness of a given input. In this regard, the Euclidean distance of the input from the classification boundary denotes a well-proved robustness assessment as the minimal affordable adversarial perturbation. Unfortunately, computing such a distance is highly complex due the non-convex nature of DNNs. Despite several methods have been proposed to address this issue, to the best of our knowledge, no provable results have been presented to estimate and bound the error committed.

This paper addresses this issue by proposing two lightweight strategies to find the minimal adversarial perturbation. Differently from the state-of-the-art, the proposed approach allows formulating an error estimation theory of the approximate distance with respect to the theoretical one. Finally, a substantial set of experiments is reported to evaluate the performance of the algorithms and support the theoretical findings. The obtained results show that the proposed strategies approximate the theoretical distance for samples close to the classification boundary, leading to provable robustness guarantees against any adversarial attacks.

Index Terms—Adversarial Robustness, Deep Neural Networks, Trustworthy AI, Verification Methods

1 Introduction

In the last decade, deep neural networks (DNNs) achieved impressive performance on computer vision applications, such as image classification [1] and object detection [2].

Despite their excellent results, all those models are liable to adversarial attacks, defined as input perturbations intentionally designed to be undetectable to humans but causing the model to make a wrong output [3], [4]. Extensive studies have been conducted for improving these attacks through effective techniques that minimize the distance from the original input to make the resulting adversarial input imperceptible to humans.

Finding the closest adversarial example, or in other terms, the minimal perturbation capable of fooling the model, is a notorious hard problem, because it involves the solution of a non-convex optimization problem with highly-irregular constraints, due to the intrinsic nature of DNNs [4]–[7]. At present, this is still a hot research topic since it allows to deduce useful information on the robustness of the model under adversarial attacks.

Almost all the powerful attacks presented in the literature (e.g., [4]–[10]) rely on the loss function gradient to build up optimization methods for crafting those perturbations. In a nutshell, their idea is to move the adversarial perturbation towards the direction that mostly increases the loss function, thus increasing the probability of a misclassification.

Although the above methods provide an affordable empirical solution to the minimal perturbation problem, to the best of our records there is no theoretical analysis that estimates and bounds the error committed.

This paper. Inspired by the known strategies that aim at solving the minimal adversarial perturbation problem, this work aims at providing an approximate solution, supported by an analytical estimation of the error committed. The motivation behind this work is to leverage the approximate solution and the analytical findings to provide provable statements regarding the trustworthiness of the classification model with respect to a given input. In the following, we first discuss the minimal adversarial perturbation problem for a binary classifier and then we extend the analysis to a multi-class classifier. To solve the above problem, we propose two new strategies that leverage a root-finding paradigm for computing the distance from the boundary.

Differently from the previous work, aimed at solving the minimum perturbation problem, the proposed strategies allow formulating an *error estimation theory that quantifies the quality of the computed distance with respect to the theoretical optimum*. More specifically, Section 4 provides provable properties about the existence of a tubular neighborhood with radius σ, where the error between the approximate distance and the minimum distance from the classification boundary can be bounded. Figure 1 better clarifies the latter point by illustrating an example of binary classification. If x is the input vector and $f(x)$ is the classification function learned by the network, our formulation provides an estimation of the radius σ from the classification boundary $\mathcal{B} = \{f(x) = 0\}$ having some regularity property. The regularity is expressed in terms of the first and the second derivatives of the classifier and measures the linearity of the classification boundary.

Section 5 reports an extensive set of experiments carried

F.Brau, G.Rossolini, A. Biondi and G. Buttazzo are with the Department of Excellence in Robotics & AI, Scuola Superiore Sant'Anna `email:name.surname@santannapisa.it`.
This work has been partially supported by Huawei Technologies Co., Ltd.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3195616

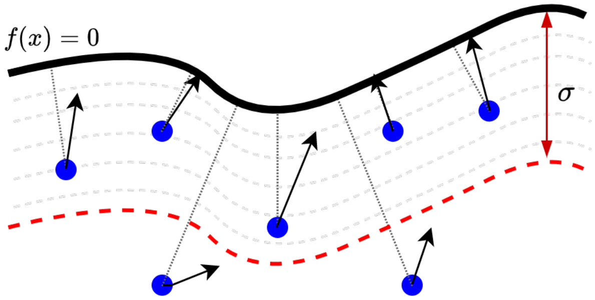

Fig. 1: Illustration of the addressed problem. The blue points are DNN inputs, while the black line $f(x) = 0$ is the classification boundary that distinguishes points belonging to the class -1 ($f(x) < 0$) and class 1 ($f(x) > 0$). The dotted line starting from each point is the unknown optimal perturbation, which is orthogonal to the classification boundary. The black arrows represent the gradient directions. Observe that the gradients computed on the points whose distance from the boundary is closer than σ provide a good approximation to the minimal adversarial distance.

out to validate the theoretical findings with a list of tests aimed at estimating the distance of an input from the classification boundary. The objective of such tests is to compare the distance computed by the proposed strategies with the approximate minimum distance obtained with a global-search method. Therefore, we validate the theoretical findings and we propose an empirical estimation of σ.

Another set of experiments exploits the theoretical findings presented in Section 4 to derive a lower bound on the magnitude of any adversarial perturbation for a given input. Such a lower bound is assessed by generating a set of powerful adversarial attacks and showing that they are not capable of finding adversarial examples of magnitude lower than the estimated distance derived by the proposed line-search methods. In summary, this paper makes the following contributions:

- It proposes two strategies based on a root-finding algorithm to solve the minimal adversarial perturbation problem close to the classification boundary.
- It presents an analytical estimation of the error committed by solving the minimal adversarial perturbation problem with the above strategies.
- It provides an analytical estimation of the neighborhood in which the previous analysis holds by leveraging a novel coefficient that measures the regularity of the classifier.
- It presents a rich set of experiments to validate the theoretical findings and a practical estimation of the radius σ that is used to deduce a provable robustness against any adversarial attack bounded in magnitude.

The remainder of this paper is organized as follows: Section 2 briefly reviews previous related work and the most effective adversarial perturbation techniques. Section 3 introduces the strategies to derive an approximate solution of the minimum adversarial perturbation problem. Section 4 provides the theoretical formulation of the error estimation. Section 5 shows the experimental results. Finally, Section 6 states the conclusion and proposes ideas for future works.

2 Background and Related Works

This section aims at presenting the problem of finding the closest adversarial example for a given input while discussing the most related papers on this topic.

2.1 Challenges in adversarial robustness

The literature related to adversarial robustness is quite vast. The problem of adversarial perturbations for DNNs was first introduced by Biggio et al. [3] and independently by Szegedy et al. [4]. Since then, a large number of works followed for proposing more powerful attacks [5], [6], [8], [9], detection mechanisms [11]–[13], and defense strategies [14]–[16]. Most adversarial attacks use a gradient based approach to craft adversarial perturbations. Although they generate impressive human undetectable adversarial examples, the reliability of the gradient direction is often taken for granted and no bound was ever provided on the error committed, with respect to the minimal theoretical perturbation.

2.2 Minimum adversarial perturbation problem

We consider a neural classifier with n inputs and C outputs, where C is the number of classes that can be recognized. Let $f : \mathbb{R}^n \to \mathbb{R}^C$ be a continuous function such that an input $x \in \mathbb{R}^n$ produces an output $f(x) \in \mathbb{R}^C$. For a given input x, the *predicted class* $\hat{k}(x)$ is defined as the index corresponding to the strictly highest component of $f(x)$; in formulas $\hat{k}(x)$ is such that $f_{\hat{k}(x)}(x) > f_k(x)$ for each $k \neq \hat{k}(x)$. If the maximum component is not unique, that is, $f_{\hat{k}(x)}(x) = \max_{k\neq\hat{k}(x)} f_k(x)$, then we define $\hat{k}(x) = 0$ meaning that the classification cannot be trusted.

It is also useful to define $R_j := \{x \in \mathbb{R}^n : \hat{k}(x) = j\}$ as the region of the input space corresponding to the class j, and $\mathcal{B}_j$ as the classification boundary for class j (or the frontier of R_j).

Let x be a correctly classified sample with label l. The problem of finding the minimal adversarial perturbation δ^*, such that $x + \delta^*$ is the closest adversarial example to x, can be obtained by solving the following minimization problem

$$\begin{aligned} d(x, l) = \min_{\delta\in\mathbb{R}^n} \quad & \|\delta\| \\ \text{s.t} \quad & \hat{k}(x + \delta) \neq l, \end{aligned} \tag{MP}$$

where $\|\cdot\|$ represents the Euclidean norm and the scalar value $d(x, l)$ represents the distance between x and the closest adversarial example $x + \delta^*$, or, equivalently, the distance of x from the classification boundary.

Note that, to practically apply the above formulation to computer vision, two additional constraints are required: *box-constraint* and *integer-constraint*. The box-constraint ensures that the adversarial example $x + \delta$ is such that $0 \leq x + \delta \leq 1$ (assuming images with pixel values normalized in $[0, 1]$). The integer-constraint ensures that each pixel x_i perturbed by δ_i is encoded into an integer with Q gray levels (e.g., $Q = 256$), that is, $Q \cdot (x_i + \delta_i) \in [0, Q - 1] \cap \mathbb{N}$.

Nevertheless, this work focuses on the unconstrained formulation, as done by Moosavi-Dezfooli et al. [5], since it is more compliant for the proposed analytical study. Note that this does not reduce generality, since the solution of MP provides a lower bound of the constrained problem.

Therefore, to reduce clutter, unless differently specified, the domain of the perturbation δ is equal to $\mathbb{R}^n$.

The following paragraphs review relevant state-of-the-art techniques for finding a practical solution of the previous minimum problem. For the sake of clarity, we group them into different categories depending on the approaches followed for solving MP.

2.3 Penalty Methods

A well known technique to solve a minimum constrained problem is given by the *Penalty Method* [17]. For instance, Szegedy et al. [4] and Carlini and Wagner [6] introduced a penalty term c and solved the following minimization problem:

$$\min_{\delta} \quad c \cdot \|\delta\| + \mathcal{L}(x+\delta, l) \tag{1}$$

where the hyper-parameter c is selected through a line search. The rationale of c is to balance the importance of the two terms in the cost function. The second term $\mathcal{L}$ represents a specific loss function that is positive in region R_l and zero in $\cup_{j\neq l} R_j$. Carlini and Wagner analyzed different loss functions finding that $\mathcal{L}(x,l) = (f_l(x) - \max_{j\neq l} f_j(x))^+$ produces the most effective results, where $f^+ = \max\{0, f\}$.

It is worth observing that in both works [4] and [6], a box constraint is added to achieve an adversarial perturbation that is feasible in the image domain. In particular, Szegedy et al. [4] exploited the L-BFGS-B optimizer [17] to directly solve the minimum problem with the box-constraint $0 \leq x+\delta \leq 1$, while Carlini and Wagner [6] introduced a change of variable to reduce to the solution of an unconstrained problem.

Although both the previous techniques allow crafting accurate perturbations, they turn out to be expensive in terms of memory usage and computational cost. Moreover, they require to repeat the optimization procedure over multiple choices of the penalty c, causing a large number of forward and backward network passes, thus resulting in a slow convergence.

2.4 Toward Faster Methods

A key contribution towards less expensive solutions of MP was given by the *Decoupling Direction and Norm* method (DDN) presented by Rony et al. [9] (recently extended by Pintor et al. [18] for different l_p norms), where the authors avoid searching for the best value of the penalty term c. Instead, they search for an adversarial example in the Euclidean ball centered in x with radius ε by performing some gradient descent steps with the loss function used to train the model and projecting the result on the sphere. Depending on whether the solution is an adversarial example, they adjust the radius of the sphere and iterate the procedure.

Another approach, named *Augmented Lagrangian Method for Adversarial Attack* (ALMA) [19], uses the same paradigm but avoids searching for the best penalty c through a line-search, by exploiting the Lagrangian duality theory [20]. Although both DNN and ALMA outperform the method by Carlini and Wagner in terms of execution time (by making less forwards and backwards passes), they do not provide a theoretical estimation of the goodness of the solution.

2.5 Distance Dependent Attacks

Much closer to this paper, DeepFool (DF) [5] is a famous fast method for finding a minimal adversarial perturbation. It leverages the geometrical properties of a specific distance (e.g., l_2) to quickly generate accurate solutions for MP.

In short, the method provides an approximate solution of MP by performing an iterative gradient based algorithm with variable step size at each iteration. To be compliant with the terminology used in Section 3, the problem solved by DF can be rewritten by considering the minimal solution of a list of less expensive minimum problems $d(x,l) = \min_{j\neq l} d_j(x)$, where $d_j(x,l)$:

$$\begin{aligned} d_j(x,l) = \min_{\delta} \quad & \|\delta\| \\ \text{s.t} \quad & f_l(x+\delta) \leq f_j(x+\delta). \end{aligned} \tag{2}$$

The main idea consists of building a sequence $x^{(1)}, x^{(2)}, \ldots, x^{(k)}, \ldots$ that converges to an approximate solution of MP, which lies in the adversarial region $\cup_{j\neq l} R_j$.

Given $x^{(k)}$, let $\tilde{f}_j(x)$ be the first order approximation of $(f_l(x) - f_j(x))$ in $x^{(k)}$. Then, the next element of the sequence $x^{(k+1)}$ is obtained by considering the minimal solution $d_j(x^{(k)}, l)$ of Problem 2 applied to $\tilde{f}_j(x)$ rather than $(f_l - f_j)(x)$. Since $\tilde{f}_j$ is an affine function, the problem has an exact solution of the form

$$x^{(k+1)} = x^{(k)} - \frac{\tilde{f}_j(x^{(k)})}{\|\nabla \tilde{f}_j(x^{(k)})\|} \frac{\nabla \tilde{f}_j(x^{(k)})}{\|\nabla \tilde{f}_j(x^{(k)})\|}. \tag{3}$$

The procedure turns out to reach convergence in $K \approx 3$ steps, resulting in $2CK$ forward and backward passes, if applied to a classifier with C classes. The comparative study reported in [9] empirically shows that the solution is close to the one found by more expensive methods, as Carlini and Wagner. However, it is crucial to point out that, since the iteration is stopped when the adversarial region is reached, there is no guarantee that the procedure provides a solution of MP. Indeed, the procedure just ensures that a feasible perturbation satisfying the constraint $\hat{k}(x+\delta) \neq l$, is found. In other words, to the best of our knowledge, there are no theoretical point-wise estimations of the approximation error, but only estimations of the average distance from the classification boundary [21].

2.6 Verification of Deep Neural Networks

Verification methods aim at establishing whether, given a sample x and a bound $\varepsilon \geq 0$, each sample in the l_p-ball centered in x with radius ε is classified with the same class of x. A verification can be performed by solving the following problem

$$\begin{aligned} \min \quad & c^T z \\ \text{s.t.} \quad & \|\delta\|_p \leq \varepsilon \quad , \\ & z = f(x+\delta) \end{aligned} \tag{VP}$$

where c depends on the task. Katz and Kochenderfer, [22], showed that such a problem is NP-complete for ReLU networks, and hence formal *complete verification* strategies are unfeasible for commonly large networks. Other works [23]–[26] bound the inner network activations to relax the constraints and provide an *incomplete verification*. However, being computationally expensive, these strategies do not scale to large networks and can be applied to multi-layer-perceptrons only or relatively small convolutional neural networks.

A different approach is given by [27], in which the authors search for the largest hyper-rectangle $\mathcal{S}_\varepsilon$, centered in x and with semi-sides of length $\varepsilon \in \mathbb{R}^n_+$, such that $\hat{k}(x+\delta) = l$ for each $\delta \in \mathcal{S}_\varepsilon(x)$. However, as in [23], solving such a problem requires estimating the bounds of the internal activations so that the method does not scale well to large networks.

More scalable approaches have been provided in [28], [29]. Krishnamurthy et al. [28] leverage dual optimization theory, which enables the verification of neural networks capable of accurately classifying images from the MNIST and CIFAR10 datasets. Recently, Wang et al. [29] proposed β-CROWN, which improves the computation of the inner activation bounds in the case of ReLU activation by splitting the verification problem in two easier problems based on the neuron outcome sign. Differently from this work, both the strategies are more suited for l_∞-bounded attacks and limit their analysis to ReLU activations only.

Another scalable verification method, named CLEVER, was proposed by Weng et al. [30]. CLEVER considers l_2-bounded attacks and provides a lower bound β_L of $d(x,l)$ (as defined in Problem (MP)) by evaluating the gradient of the network f on random samples in a neighborhood of x. However, the accuracy of the bound strongly depends on the number of gradient evaluations: the higher the number of evaluations, the narrower the bound. Hence, achieving accurate results requires a long computational time.

Cohen et al. [31] and its recent generalization [32] proposed a method for constructing a new “smoothed” classifier h_s from an arbitrary base classifier h. The classification of a sample x through h_s is performed by evaluating h using many perturbed versions of x. Classifier h_s is then proved to be robust against adversarial attacks with a certain magnitude. However, observe that the robustness is proved on the new classifier h_s and not on the black-box one h.

2.7 This work

Although the reviewed methods can craft accurate adversarial perturbations, they do not provide an estimation of the error committed with respect to the optimal distance. Differently from the methods described above, this work presents two methods for finding an approximate solution of MP that simplifies a complex global computation by treating it as a root-finding procedure. This allows formulating an error estimation theory that is formally illustrated in Section 4 and validated in Section 5. Moreover, a final test leverages the estimated error for deriving provable robustness guarantees of a given input x against any adversarial attack.

3 Boundary Distance via Root Algorithm

This section illustrates two main strategies that provide an approximate solution to problem MP by reducing it to a minimal root problem. A theoretical analysis for evaluating the approximation error is provided in Section 4. The most frequent symbols used throughout the paper are summarized in Table 1.

TABLE 1: Summary of the most frequent symbols.

Symb.	Dimensionality	Meaning
f	$: \mathbb{R}^n \to \mathbb{R}^C(\mathbb{R})$	classifier (binary classifier)
$d(x,l)$	$\in \mathbb{R}$	solution of MP
$t(x,l)$	$\in \mathbb{R}$	solution of RP
$\mathcal{B}$	$\subseteq \mathbb{R}^n$	classification boundary (binary classif.)
Ω_σ	$\subseteq \mathbb{R}^n$	tubular neighborhood of $\mathcal{B}$ of radius σ
ρ	$\in \mathbb{R}$	coefficient of Inequality 8
$\sigma(\rho)$	$\in \mathbb{R}$	radius in which Inequality 8 holds.
σ^*	$\in \mathbb{R}$	significant lower bound of σ
$\hat{\sigma}^*$	$\in \mathbb{R}$	empirical estimation of σ^*

Both strategies leverage two main observations: (i) the gradient of f suggests the fastest direction to reach the adversarial region; and (ii) due to the objective function, the minimal perturbation lays on the classification boundary. The two considerations above naturally bring to searching the minimal perturbation as the intersection between the classification boundary and the direction of the gradient ∇f.

3.1 The Case of Binary Classifiers

Differently from a multi-class classifier, a *binary classifier* can be modeled as a scalar function $f : \mathbb{R}^n \to \mathbb{R}$ that provides a classification based on its sign, i.e., for each $x \in \mathbb{R}^n$, $\hat{k}(x) = \operatorname{sgn}(f(x))$. Let x be a correctly predicted sample of class $l \in \{1, -1\}$. Due to the objective, the minimal perturbation δ^* that solves MP is such that the perturbed sample $x + \delta^*$ belongs to the classification boundary, i.e. $x + \delta^* \in \mathcal{B} = \{p \in \mathbb{R}^n \,:\, f(p) = 0\}$. This can easily be proved by contradiction by observing that, if δ^* is a solution of MP, but $\operatorname{sgn}(f(x)) \neq \operatorname{sgn}(f(x+\delta^*)) \neq 0$, then, due to the continuity of f, there exists $0 < t < 1$ such that $f(x + t\delta^*) = 0$, which is a contradiction because $\|t\delta^*\| < \|\delta^*\|$.

Based on this observation, we can replace the original problem with the following minimization problem with an equality constraint

$$\begin{aligned} d(x,l) = \min_{\delta} \quad & \|\delta\| \\ \text{s.t} \quad & f(x+\delta) = 0, \end{aligned} \tag{MP-Eq}$$

equivalent to a minimum distance problem from set $\mathcal{B}$.

It is worth observing that the gradient $\nabla f(p)$ is orthogonal to the boundary $\mathcal{B}$ for each $p \in \mathcal{B}$, and that, if x is close to the boundary, then $\nabla f(x) \approx \nabla f(p^*)$ (where $p^* = x + \delta^*$) provides the fastest direction to reach the boundary. Hence, it is reasonable to approximate MP-Eq with the following minimal root problem (a formal proof of this is reported in Section 4):

$$\begin{aligned} t(x,l) = \min_{t \in \mathbb{R}_+} \quad & t \\ \text{s.t} \quad & f(x + t\nu(x)) = 0 \end{aligned} \tag{RP}$$

where $\nu = -\operatorname{sgn}(f(x)) \frac{\nabla f(x)}{\|\nabla f(x)\|}$ represents the direction that best approximates $\nabla f(p^*)$ at the first order.

3.2 Extension to Multi-class Classifiers

The extension of the binary case to a multi-class classifier is not unique. This section presents two different strategies to tackle the problem.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3195616

3.2.1 The closest boundary

The *Closest Boundary* strategy (CB) leverages the idea that the minimum problem related to a classifier with C classes can be reduced to a list of minimum problems for binary classifiers.

In detail, let x be a sample, correctly classified by f with label $l \in \{1, \dots, C\}$, and let

$$\begin{aligned} d_j(x,l) = \min_{\delta} \quad & \|\delta\| \\ \text{s.t} \quad & f_l(x+\delta) \le f_j(x+\delta). \end{aligned} \tag{4}$$

Then, we observe that $d(x,l) = \min_{j \neq l} d_j(x,l)$, where $d(x,l)$ solves MP. This can be proved by reformulating the statement with the following inequalities

$$\min_{j \neq l} d_j(x,l) \le d(x,l) \le \min_{j \neq l} d_j(x,l).$$

Let $\delta^{(j)}$ be the solution of $d_j(x,l)$. The second inequality is a consequence from the fact that $\delta^{(j)}$ satisfies the constraint of MP and that, by construction, $d(x,l)$ is lower than $\|\delta\|$ for each feasible δ. The first inequality, instead, can be proved by observing that Problem MP is equivalent to

$$\begin{aligned} d(x,l) = \min_{\delta} \quad & \|\delta\| \\ \text{s.t} \quad & f_l(x+\delta) \le \max_{j \neq l} f_j(x+\delta). \end{aligned} \tag{5}$$

Hence, if δ^* is the solution of Problem MP and if $j^* \in \text{argmax}_{j \neq l} f_j(x+\delta^*)$, then, by construction, δ^* satisfies the constraint of Problem 4 for j^*, and so $\min_{j \neq l} d_j(x,l) \le d_{j^*}(x,l) \le d(x,l)$. In conclusion, if $t_j(x,l)$ is the solution of RP with $f(x) = f_l(x) - f_j(x)$, then $d(x,l)$ can be approximated by $t(x,l) = \min_{j \neq l} t_j(x,l)$. More informally, if $B_{jl} := \{p \in \mathbb{R}^n : f_l(x) = f_j(x)\}$ is the classification boundary of the binary classifier $f_l - f_j$, we can reduce MP to the problem of finding the closest intersection between the boundary B_{jl} and the straight line passing through x with the direction provided by the gradient of f.

A good aspect of this strategy is that it reduces to the solution of a sequence of minimum problems by preserving the regularity of f. In fact, it is important to anticipate that the regularity and the differentiability of f has a big impact on the accuracy of the approximation (see Section 4). For the sake of clarity, the procedure described above is summarized in Algorithm 1, where function `Zero`, called at Line 7, is any root finding algorithm for univariate functions that solves RP.

3.2.2 Fast outer boundary

The CB algorithm presented in the previous section can bring to a large computational cost for a classifier f that distinguishes a large number of classes. In fact, if O_j is the amount of forward and backward passes required to compute each $t_j(x,l)$, then the total cost O can be estimated as $\sum_{j \neq l} O_j$. The *Fast outer Boundary* strategy (FOB) is hence proposed here to contain the computational cost.

The minimum problem MP can be reduced to the minimal root problem RP by considering $L(x,l) = f_l(x) - \max_{j \neq l} f_j(x)$ and observing that L acts like a binary classifier that takes positive values in the region R_l and negative values in the outer region $\cup_{j \neq l} R_j$. Hence, the approximation of $d(x,l)$ can be deduced by solving the minimal root problem obtained by substituting f with L in Problem RP.

Algorithm 1: Pseudocode implementing the Closest Boundary strategy depending on a root-finding algorithm.

```
   Data: Zero                !The root-finding algorithm.
   Input: $x$, $l$, $f$      !The safe sample and the DNN.
   Output: $t$, $\nu$        !The distance and the direction
 1 t = $\infty$;
 2 for $j = 1, \dots, c$ and $j \neq l$ do
 3     $F(x) := f_l(x) - f_j(x)$;
 4     grad = $\nabla F(x)$;
 5     $\nu_j$ = -grad/$\|$grad$\|$;
 6     $g(t) := F(x + t \cdot \nu_j)$;
 7     $t_j$= Zero(g);
 8     if $t_j < t$ then
 9         $t = t_j$ ;
10         $\nu = \nu_j$;
11 return $t, \nu$;
```

Observe that, differently from the previous strategy, this one requires the solution of a single minimal root problem. The pseudocode formulation of the FOB strategy can easily be obtained as a variant of Algorithm 1 by replacing F with L and removing the `for` loop.

3.3 Root-Finding algorithms

In this work, the above strategies are tested by solving the root problem RP with a customized version of the *Bisection Algorithm* and the vanilla *Newton Algorithm*, which return the approximate distance $t(x,l)$ for each sample (x,l). The bisection method has been adapted to better fit the task. A more detailed illustration is provided below. In general, the bisection method allows finding a zero of a scalar univariate continuous function $g : [a,b] \to \mathbb{R}$ under the assumption that $g(a) > 0$ and $g(b) < 0$, without requiring the computation of the derivative of g. Note that in our case $a = 0$ because in Problem RP the variable t is positive.

Solving RP requires finding the minimal positive root of the g function, which, in general, is not a solution of the vanilla bisection algorithm. In fact, in the searching interval $[0,b]$, function g is not guaranteed to be monotone and it can change sign, from positive to negative and vice-versa. To tackle this issue, we apply a pre-processing to the initial searching interval $[0,b]$ that is inspired by Armijo rule for line search methods [17].

In details, given a maximum number of attempts R, we consider $\tilde{b} = b \cdot 2^{-\tilde{k}}$, where

$$\tilde{k} = \max \left\{ i \in \mathbb{N} \,:\, g(b \cdot 2^{-i}) < 0,\ i = 0, 1, \cdots, R \right\} \tag{6}$$

and we start the bisection in $[0, \tilde{b}]$.

The pseudocode that implements the Closest Boundary strategy is shown in Algorithm 2. Line 20 reduces the amount of forward passes of the model by stopping the inner iteration if the lower bound `t_curr_low` of the current label is higher than the actual overall minimal estimation `t`.

Algorithm 2: Pseudocode for bisection algorithm, with armijo-like upper bound estimation, applied to the Closest Boundary strategy.

```
Data: t_up, MaxIter, MaxAttempt
Input: $x$, $l$, $f$                 !The sample and the DNN
Output: $t$,$\nu$                    !The distance and the direction
Tol = 5e-5;
t = $\infty$;
for $j = 1, \dots, c$ and $j \neq l$ do
    $F(x) := f_l(x) - f_j(x)$;
    grad = $\nabla F(x)$;
    $\nu_j$ = −grad/‖grad‖;
    !Starting of the bisection algorithm;
    t_curr_low = 0;
    t_curr_up = Armijo(g, b=t_up);
    for step = 1,. . . , MaxIter do
        t_curr = (t_curr_low + t_curr_up )/2;
        x_curr = x + t_curr * $\nu_j$;
        out = F(x_curr);
        if out > 0 then
            t_low = t_curr;
            out_low = o;
        else
            t_up = t_curr;
            out_up = o;
        if t_curr_low > t then
            Break   !Reduce the amount of iterations;
        if 0 > o_up > -Tol then
            Convergence;
    if $t\_up < t$ then
        $t = t\_up$ ;
        $\nu = \nu_j$;
return $t$, $\nu$;
```

4 Bounding the Distance from the Classification Boundary

This section formally addresses the problem of estimating the Euclidean distance from the classification boundary. The case of a binary classifier is first considered, while multi-class classifiers are addressed later in Section 4.4.

The objective is to *leverage the error estimation to prove whether an input is far enough from the classification boundary*, hence guaranteeing that is provably safe with respect to adversarial perturbations bounded in Euclidean norm. To this end, this section provides an estimation of the error obtained by approximating the distance from the boundary $d(x, l)$, i.e., the solution of MP, with $t(x, l)$, i.e., the solution of the minimal root problem RP.

Formally, by adopting the notation from Section 3.1, given a radius $\sigma > 0$, let $\Omega_\sigma := \{x \in \mathbb{R}^n : d(x) < \sigma\}$ be the tubular neighborhood of $\mathcal{B}$ of radius σ, where

$$d(x) := \min_{p \in \mathcal{B}} \|x - p\|, \tag{7}$$

i.e., Ω_σ is the set of all samples whose distance from the classification border $\mathcal{B}$ is less than σ.

The proposed method provides an upper bound and a lower bound of $d(x, l)$ depending on $t(x, l)$ and a coefficient $\rho \geq 1$, which quantifies the quality of the estimation (the lower the better). In particular, we formally prove the existence of a radius $\sigma(\rho)$ such that the approximation error is bounded as follows, for each $\rho \in (\sqrt{2}, 2]$:

$$\frac{1}{\rho} t(x, l) < d(x, l) \leq t(x, l), \tag{8}$$

where the first inequality holds for each x in $\Omega_{\sigma(\rho)}$. Such an estimation is only valid in a neighborhood of $\mathcal{B}$ depending on the magnitude of ρ. However, the lower ρ the smaller the tubular neighborhood in which the inequality holds. In other words, the conditions under which the estimation error can be bounded become more and more difficult to be satisfied as the quality of the bound provided by Inequality (8) increases.

Given a distance $\varepsilon < \sigma(\rho)$, we say that f is an ε-robust classifier with respect to (x, l) if the sample x does not admit an adversarial perturbation of magnitude lower than ε, i.e., if for each perturbation δ with $\|\delta\| < \varepsilon$ then $\hat{k}(x) = \hat{k}(x + \delta)$.

Thus, by only computing $t(x, l)$, it is possible to deduce the robustness of a classifier with respect to a sample x according to the following rules:

- If $t(x, l) < \varepsilon$, then the classifier is not ε-robust with respect to (x, l).
- If $t(x, l) > \rho\varepsilon$, then the classifier is ε-robust with respect to (x, l).

4.1 Preliminaries

Before going deeper in the mathematical aspects, it is necessary to introduce three assumptions on the function f of the classifier.

Assumption A. The function f is of class $C^\infty(\mathbb{R}^n)$.

Assumption B. The function f is strictly positive outside some $B(0, M)$ (the open ball centered in 0 with radius M).

Assumption C. The gradient ∇f is not zero in $\mathcal{B}$ (i.e., 0 is a regular value of f).

Although the three assumptions above are not valid in general, they are not restrictive for a neural classifier. In particular, for a feed forward deep neural network with a one-dimensional output, Assumption B is not verified by f. However, being the samples of our interest always in some closed limited set K, we can theoretically substitute f in the following proofs with another function $\tilde{f}$ that coincides with f in the compact set K and that satisfies Assumption B. More details can be found in Appendix C. Similarly, Assumptions A and C are not valid in general, but we can assume that, in a practical domain, f is the quantized representation of another function $\tilde{f}$ that satisfies the conditions. Observe that Assumptions A and C ensure that $\mathcal{B}$ is a smooth manifold of dimension $n - 1$ (this can be proved by applying the implicit function theorem [33]). Assumption B, instead, ensures that $\mathcal{B}$ is a *compact set*.

Since $\mathcal{B}$ is a compact set, then the minimum distance problem formulated in Equation (7) admits a solution for each $x \in \mathbb{R}^n$. Nevertheless, there is no guarantee that for each $x \in \mathbb{R}^n$ there exists a unique closest point in $\mathcal{B}$. The following result ensures the existence of a unique solution in a tubular neighborhood of $\mathcal{B}$ (refer to [34] for more details).

Theorem 1 (Unique Projection [34])**.** *If $\mathcal{B} \subseteq \mathbb{R}^n$ is a compact manifold, then there exists a maximum distance σ_0 such that for each x in the open tubular neighborhood Ω_{σ_0} there exists a unique $\pi(x) \in \mathcal{B}$ that solves Equation (7). Moreover, d is differentiable in the neighborhood, and $\nabla d(x) = \frac{x-\pi(x)}{\|x-\pi(x)\|}$ for each $x \in \Omega_\sigma \setminus \mathcal{B}$.*

Following this result, the lemmas below explain in a formal fashion that, *close to the classification boundary, the gradient of f in x provides a fast direction to reach $\mathcal{B}$.*

Observe that this is the main idea behind all the gradient-based attacks and, in particular, DeepFool [5], which exploits the gradient of f to rapidly reach the adversarial region.

4.2 Bounding the estimation error

Let $B(x, r)$ be the open ball in the Euclidean norm centered in x with radius r. Furthermore, for each set $A \subseteq \mathbb{R}^n$, let $\overline{A}$ be the closure of A, i.e. the smallest closed set containing A.

Lemma 1. *Let σ_0 be the distance for which Theorem 1 holds. For each $x \in \Omega_{\sigma_0} \setminus \mathcal{B}$, the direction $x - \pi(x)$ is parallel to $\nabla f(\pi(x))$, where $\pi(x)$ is the unique closest point in $\mathcal{B}$ to x. In particular,*

$$\nabla d(x) = \frac{x - \pi(x)}{\|x - \pi(x)\|} = \operatorname{sgn}(f(x)) \frac{\nabla f(\pi(x))}{\|\nabla f(\pi(x))\|}. \tag{9}$$

Proof. By construction, $\pi(x)$ is the solution of the minimum problem on Eq. (7). Then, by the Necessary Condition Theorem in [17, p. 278], because of Assumption C, there exists $\lambda^* \in \mathbb{R}$ such that $\nabla \mathcal{L}(\pi(x), \lambda^*) = 0$, where $\mathcal{L}(p, \lambda) = \|x - p\| + \lambda f(p)$. Observe that $\nabla \mathcal{L}(\pi(x), \lambda^*) = 0$ implies that

$$\nabla d(x) = \frac{x - \pi(x)}{\|x - \pi(x)\|} = \lambda^* \nabla f(\pi(x)). \tag{10}$$

From the above equation, because $\|\nabla d(x)\| = 1$, we deduce that $|\lambda^*| = \frac{1}{\|\nabla f(\pi(x))\|}$. It remains to prove that $\operatorname{sgn}(\lambda^*) = \operatorname{sgn}(f(x))$. To prove this statement, we proceed in three steps: (i) we prove that the segment p_t that connects x to $\pi(x)$ is such that $\operatorname{sgn}(f(p_t)) = \operatorname{sgn}(f(x))$ for $t > 0$; (ii) we show that for $t \approx 0$, the sign of $\operatorname{sgn}(f(p_t))$ is equal to the sign of $\nabla f(\pi(x))^T (x - \pi(x))$; (iii) by leveraging identity Equation (10), we show that the sign of λ^* is equal to the sign of $\nabla f(\pi(x))^T (x - \pi(x))$.

Let $p_t := \pi(x) + t(x - \pi(x))$ where $t \in [0, 1]$. Observe that $\operatorname{sgn}(f(x)) = \operatorname{sgn}(f(p_t))$ for each $t \in (0, 1]$. In fact, by contradiction, if there exists τ with $\operatorname{sgn}(f(x)) \neq \operatorname{sgn}(f(p_\tau))$, then, by the Bolzano Theorem applied to function f, it would exist a $\tau_* \in (0, 1)$ such that $f(p_{\tau_*}) = 0$. This would imply that

$$\|x - p_{\tau_*}\| = \|(1 - \tau_*)(x - \pi(x))\| < \|x - \pi(x)\|,$$

which is a contradiction because $\|x - p_{\tau_*}\| < d(x)$ but $\pi(x)$ solves Problem 7.

Based on this fact, observe that, since f is differentiable in p_0, then

$$\begin{aligned} f(p_t) &= f(p_0) + \nabla f(p_0)^T (p_t - p_0) + o(p_t) \\ &= t \nabla f(\pi(x))^T (x - \pi(x)) + o(p_t), \end{aligned}$$

where $o(p_t)/t \to 0$ when $t \to 0$, from which we deduce that for small t, $\operatorname{sgn}(f(p_t)) = \operatorname{sgn}\left(\nabla f(\pi(x))^T (x - \pi(x))\right)$.

In conclusion, multiplying each term of Equation (10) by $\nabla f(\pi(x))^T$, we deduce that the sign of the first term of the equivalence is equal to $\operatorname{sgn}(\lambda^*)$, which proves the lemma. □

The above result can be seen as a particular case of the following lemma, which states that the angle between $\nabla f(x)$ and the optimal direction $\nabla d(x)$ can be bounded in a neighborhood of the boundary.

Lemma 2 (Angular Constraint)**.** *For each angle bound $\alpha \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$, there exists a distance $\sigma_1(\alpha)$, such that, for all $x \in \Omega_{\sigma_1(\alpha)}$, the following inequality holds*

$$\frac{\nabla f(x)^T \nabla f(\pi(x))}{\|\nabla f(x)\| \|\nabla f(\pi(x))\|} > \cos(\alpha), \tag{11}$$

where $\pi(x)$ is the unique projection of Theorem 1.

Proof. From Assumption A, we deduce the continuity of ∇f. From Assumption C and the compactness of $\mathcal{B}$, we deduce that there exists a distance δ such that $\|\nabla f(x)\| \neq 0$ in $\overline{\Omega}_\delta$ (the closure of Ω_δ), and so we deduce that $\frac{\nabla f}{\|\nabla f\|}$ is uniformly continuous in $\overline{\Omega}_\delta$. Hence, for each ε, there exists a distance $\sigma_\varepsilon \leq \delta$ such that, for each $x, y \in \Omega_\delta$ and $\|x - y\| < \sigma_\varepsilon$, the following inequality holds

$$\left\| \frac{\nabla f(x)}{\|\nabla f(x)\|} - \frac{\nabla f(y)}{\|\nabla f(y)\|} \right\| < \varepsilon. \tag{12}$$

By remembering that $\|v - w\|^2 = \|v\|^2 + \|w\|^2 - 2v^T w$ for each $v, w \in \mathbb{R}^n$ we can deduce the following inequality

$$1 - \frac{1}{2}\varepsilon^2 < \frac{\nabla f(x)^T \nabla f(y)}{\|\nabla f(x)\| \|\nabla f(y)\|}. \tag{13}$$

In conclusion, by taking $y = \pi(x)$ and by selecting $\varepsilon = \sqrt{2 - 2\cos(\alpha)}$, we deduce Equation (11) where $\sigma_1(\alpha) = \min(\sigma_0, \sigma_\varepsilon)$. □

Intuitively, by the geometrical properties of a manifold, a small portion of the boundary can be enclosed between two affine parallel hyperplanes. This is the aim of the following lemma.

Lemma 3 (Thickness Constraint)**.** *For each thickness factor $\beta \in (0, 1)$, there exists a maximum distance $\sigma_2(\beta)$ such that, for all $p \in \mathcal{B}$, the open set*

$$\Gamma_r(p) := \left\{ p + v \; : \; |v^T \nabla f(p)| < \beta r \|\nabla f(p)\|, \; v \in \mathbb{R}^n \right\}$$

contains $\mathcal{B} \cap B(p, r)$ for all $r < \sigma_2(\beta)$.

Proof. Let $p \in \mathcal{B}$. Because f is differentiable in p, there exists a radius δ_p such that for all points $q \in B(p, \delta_p) \cap \mathcal{B}$ the following identity holds

$$o(\|p - q\|) = f(q) - f(p) + (p - q)^T \nabla f(p) = (p - q)^T \nabla f(p)$$

and $o(\|p - q\|) / \|p - q\| \to 0$ for $\|p - q\| \to 0$. Observe that the same limit holds by dividing each term by $\|\nabla f(p)\|$, which is not zero due to Assumption C. By definition of limit, there exists $\sigma_p < \delta_p$ such that for each $q \in B(p, \sigma_p)$

$$\left| (p - q)^T \nabla f(p) \right| < \beta \|p - q\| \|\nabla f(p)\|, \tag{14}$$

with $\beta \in (0,1)$. This proves that for each $r \leq \sigma_p$, if $q \in B(p,r) \cap \mathcal{B}$, then $q \in \Gamma_r(p)$ by considering $v = q - p$ and observing that $\|p - q\| < r$ as $p, q \in B(p,r)$.

So far we proved that the statement holds locally, i.e. for each p there exits σ_p such that $B(p,r) \cap \mathcal{B} \subseteq \Gamma_r(p)$ for each $r \leq \sigma_p$. In conclusion, the thesis follows by observing that the existence of a global $\sigma_2(\beta)$ such that the condition above holds for all p and for all $r \leq \sigma_2(\beta)$ is a consequence of the compactness of $\mathcal{B}$. In fact, the family $\{B(p,\sigma_p)\}_{p \in \mathcal{B}}$ is an infinite cover of $\mathcal{B}$ that, by definition of a compact set, admits a finite sub-cover indexed by $p_1, \cdots, p_k$ such that $\mathcal{B} \subseteq \cup_{i=1}^{k} B(p_i, \sigma_{p_i})$. By taking $\sigma_2(\beta) = \min_i \sigma_{p_i}$ we deduce the thesis. □

The Lemma above shows that the boundary $\mathcal{B}$ can be locally bounded by the open set $\Gamma_r(p)$ for each point p and for each radius r not larger than $\sigma_2(\beta)$. Furthermore, the border $\mathcal{B}$ splits the set $B(p,r) \cap \Gamma_r(p)$ in a way that f keeps a constant sign in the two hyperplanes $R_\pm := \{p + v \,:\, v^T \nabla f(p) = \pm\beta r \|\nabla f(p)\|,\, v \in \mathbb{R}^n\}$ which coincide with the frontier of $\Gamma_r(p)$.

The geometrical intuition behind this statement is condensed in the following corollary of Lemma 3.

Corollary 1. *Let $\beta \in (0,1)$ and $\sigma_2(\beta)$ of Lemma 3. Let $x \in \Omega_{\sigma_2(\beta)}$, p such that $d(x) = \|x - p\|$ and $r = d(x)$, then the hyperplane*

$$R := \left\{p + v \,:\, v^T \nabla f(p) = -\mathrm{sgn}(f(x))\beta r \|\nabla f(p)\|,\, v \in \mathbb{R}^n\right\}$$

is such that

$$\forall y \in R \cap B(p,r), \quad \mathrm{sgn}(f(y)) = -\mathrm{sgn}(f(x)). \tag{15}$$

Proof. Let us prove the statement for $f(x) < 0$ first. The proof can be decomposed in two steps: (i) Prove that $p_+ := p + r\beta \frac{\nabla f(p)}{\|\nabla f(p)\|} \in R$ and $f(p_+) > 0$; (ii) Prove that if $y \in R \cap B(p,r)$, then $\mathrm{sgn} f(p_+) = \mathrm{sgn} f(y)$.

The first statement can be proved by using a procedure similar to the one adopted in Lemma 1. In particular, let $p_t := p + t\beta r \frac{\nabla f(p)}{\|\nabla f(p)\|}$ for $t \in [0,1]$ be the segment going from p to p_+; first, we prove that f takes positive values for small values of t; and then we prove that f does not change sign in p_+. Since f is differentiable in p, then

$$f(p_t) = tr \nabla f(p)^T \left(\frac{\nabla f(p)}{\|\nabla f(p)\|}\right) + o(p_t),$$

and because $o(p_t)/t \to 0$, we can deduce that $\mathrm{sgn}(f(p_t)) = \mathrm{sgn}(r\|\nabla f(p)\|) = 1$ for small t. Let us now prove by contradiction that if f changes sign in p_+, then Lemma 3 would be not valid in p. If $f(p_+) \leq 0$, then there exist $\tau^* \leq 1$ such that $f(p_{\tau^*}) = 0$. Hence, $\|p - p_{\tau^*}\| = |\tau^* r \beta|$, from which $p_{\tau^*} \in B(p, \tau^* r)$. Let us consider the smaller radius $r^* = \tau^* r$ and observe that $p_{\tau^*} \notin \Gamma_{r^*}(p)$. In fact, $\tau^* \beta r \frac{\nabla f(p)^T}{\|\nabla f(p)\|} \nabla f(p) = \beta r^* \|\nabla f(p)\|$ shows that p_{τ^*} lays on the topological border of the set $\Gamma_{r^*}(p)$. This brings to a contradiction for Lemma 3 being $p_{\tau^*} \in \mathcal{B} \setminus \Gamma_{r^*}(p)$.

Finally, if $y \in B(p,r) \cap R$, the second statement can be proved by contradiction observing that, if $f(y) \leq 0$, then there exists $p_0 \in R \cap B(p,r)$ for which $f(p_0) = 0$. Furthermore, this would implies that $p_0 \in \mathcal{B}$ and $p_0 \notin \Gamma_r(p)$, which brings to a contradiction by Lemma 3.

In conclusion, the case $f(x) > 0$ can be deduced by following the steps above, but considering $p_- := p - t\beta r \frac{\nabla f(p)}{\|\nabla f(p)\|}$, to prove that $f(p_-) < 0$. □

Lemma 2 and Lemma 3 are linked by the following intuitive connection. In a geometrical sense, $d(x)$ represents the length of the shortest path needed to reach the boundary, which is obtained by moving from x along $-\nabla d(x)$.

Similarly, let $t(x)$ be the length of the path (if there exists one) required to reach the boundary by following the direction $\nu(x) = -\mathrm{sgn}(f(x)) \frac{\nabla f(x)}{\|\nabla f(x)\|}$, in formulas $x + t(x)\nu(x) \in \mathcal{B}$. To ensure the existence of such a $t(x)$, we can leverage two conditions. If we admit that $\nu(x)$ is not similar to the optimal one (i.e., we assume a $\alpha \not\approx 0$ in Lemma 2), then the existence of $t(x)$ would only be guaranteed by an almost straight boundary $\mathcal{B}$, which requires a thickness factor close to zero, $\beta \approx 0$. Vice versa, if we admit a highly irregular boundary (i.e., $\beta \not\approx 0$), then the existence of $t(x)$ would only be guaranteed by a direction $\nu(x)$ close to the optimal one. This would require $\alpha \approx 0$.

This is the main idea of the following theorem, which, by balancing the two parameters α and β, ensures: (i) The existence of $t(x)$; and (ii) The estimation of $d(x)$ through $t(x)$ defined in Equation (8). A graphical idea of the proof is depicted in Figure 2.

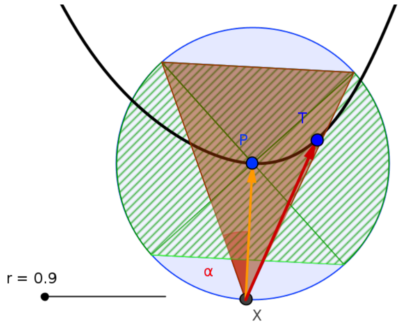

Fig. 2: A graphical proof of Theorem 2. Lemma 3 ensures that in $B(p,r)$ the boundary belongs in the green area. Lemma 2 ensures that $\nu(x)$ (in red) lays in the brown area. In conclusion, there exists a solution T of RP, i.e. an intersection between the boundary and the direction provided by the gradient.

Theorem 2 (Distance Estimation)**.** *For each angle $\alpha \in \left(-\frac{\pi}{4}, \frac{\pi}{4}\right)$ there exists a maximum distance $\sigma = \min\{\sigma_1(\alpha), \sigma_2(\cos(2\alpha))\}$ such that the error in approximating $d(x)$ with $t(x)$ can be bounded as*

$$\forall x \in \Omega_\sigma, \quad d(x) \leq t(x) \leq 2\cos(\alpha) d(x), \tag{16}$$

where $t(x) \in \mathbb{R}_+$ is the smallest value such that

$$x - t(x)\,\mathrm{sgn}(f(x)) \frac{\nabla f(x)}{\|\nabla f(x)\|} \in \mathcal{B}.$$

Proof. Let $\beta = \cos(2\alpha)$. Let $\sigma_1(\alpha)$ and $\sigma_2(\beta)$ be the maximum distances of Lemmas 2 and 3, respectively, and let $\sigma = \min(\sigma_1(\alpha), \sigma_2(\beta))$. Note that in this way Lemmas 2 and 3 hold for $x \in \Omega_\sigma$. Let $p = \pi(x) \in \mathcal{B}$ the closest projection, $r = \|p - x\| = d(x)$ the minimum distance from the boundary, and let $\varphi(t) = x + t\frac{\nabla f(x)}{\|\nabla f(x)\|}$ be the straight line passing through x with direction $\nabla f(x)$. Observe that, by definition of Ω_σ, it holds $r < \sigma$. Without loss of generality, we can assume that $f(x) < 0$.

The proof strategy consists in proving that the straight line $\varphi(t)$ intersects the hyperplane $R_+ := \{p + v \,:\, v^T\nabla f(p) = \beta r\|\nabla f(p)\|,\, v \in \mathbb{R}^n\}$ (which is one of the borders of the set $\Gamma_r(p)$ of Lemma 3) in a point $\varphi(t_*)$, in which f assumes a positive value. This would imply the existence of some point $\varphi(t(x))$ such that $f(\varphi(t(x)) = 0$.

Observe that the intersection between the support of φ and R_+ is realized for

$$t_* = \frac{\|\nabla f(x)\|}{\nabla f(x)^T\nabla f(p)}\left(r\beta\|\nabla f(p)\| - (x-p)^T\nabla f(p)\right). \tag{17}$$

Moreover, observe that, multiplying each term of Equation (10) in Lemma 1 by $\nabla f(p)^T$, we deduce that $(x - p)^T\nabla f(p) = -r\|\nabla f(p)\|$, from which, by substituting in the second term of Equation (17), we deduce that

$$t_* = \frac{\|\nabla f(x)\|\|\nabla f(p)\|}{\nabla f(x)^T\nabla f(p)}\,(1+\beta)\,r. \tag{18}$$

Note that with $\beta = \cos(2\alpha)$, the intersection $\varphi(t_*)$ is realized inside the closed ball $\overline{B(p,r)}$ (details can be found in Appendix D.1). From Lemma 2, $\frac{\|\nabla f(x)\|\|\nabla f(p)\|}{\nabla f(x)^T\nabla f(p)} < \frac{1}{\cos(\alpha)}$, thus by Equation (18) we deduce the right-hand side of the following inequality

$$d(x) \le t_* < \frac{1+\beta}{\cos(\alpha)}r = 2\cos(\alpha)d(x), \tag{19}$$

while the left-hand side is trivial by construction of $d(x)$.

In conclusion, by observing that $x = \varphi(0)$, if we prove that $f(\varphi(0)) < 0 < f(\varphi(t_*))$, we can deduce the existence of $t(x) < t_*$ such that $f(\varphi(t(x))) = 0$, which finally implies Equation (16). The condition $f(\varphi(0)) < 0$ holds by assumption. Moreover, by construction, $\varphi(t_*) \in R_+$ and so by Corollary 1 $f(\varphi(t_*))$ is strictly positive. Hence the theorem follows. □

4.3 A significant lower bound of σ

This section presents an analysis of the magnitude of the radius of the tubular neighborhood Ω_σ in which Equation (16) holds and provides a lower bound of the largest σ. In particular, the following lemmas provide an analytical estimation of two lower bounds $\tilde\sigma_1$ and $\tilde\sigma_2$ for σ_1 and σ_2, respectively, depending on the gradient of f and on the Hessian $\nabla^2 f$. Henceforth, we make use of the following notation

$$\|\nabla^2 f\|_{\mathcal{K}} := \max_{x\in\mathcal{K}}\|\nabla^2 f(x)\|,$$

where $\mathcal{K}$ is a compact set and $\|\nabla^2 f(x)\|$ is the operator norm of the matrix $\nabla^2 f(x)$ inducted by the euclidean norm.

Lemma 4 (Lower bound of σ_1)**.** *Let $\Omega = \Omega_\delta$ of Lemma 2. For each $\alpha \in \left(-\frac{\pi}{4}, \frac{\pi}{4}\right)$,*

$$\tilde\sigma_1(\alpha) := \frac{1}{2}\frac{\inf_{x\in\Omega}\|\nabla f(x)\|}{\|\nabla^2 f\|_{\overline{\Omega}}}(1-\cos(\alpha)) \le \sigma_1(\alpha) \tag{20}$$

where $\sigma_1(\alpha)$ is the same of Lemma 2.

Proof. See Appendix B.1 □

Lemma 5 (Lower bound of σ_2)**.** *For each $\beta \in (0,1)$*

$$\tilde\sigma_2(\beta) := 2\beta\cdot\frac{\inf_{p\in\mathcal{B}}\|\nabla f(p)\|}{\|\nabla^2 f\|_{\mathcal{B}}} \le \sigma_2(\beta) \tag{21}$$

where $\sigma_2(\beta)$ is the same of Lemma 3.

Proof. See Appendix B.2 □

The lemmas above provide a lower bound $\tilde\sigma$ of σ by considering $\tilde\sigma(\rho) = \min\{\tilde\sigma_1(\alpha), \tilde\sigma_2(\beta)\}$, where α and β are such that $\rho = 2\cos(\alpha)$ and $\beta = \cos(2\alpha)$.

Therefore, observe that the lower bounds $\tilde\sigma_1$ and $\tilde\sigma_2$ depend on two main parameters that measure the linearity of the function f. In fact, for an affine function $f(x) = w^Tx + b$, these bounds diverge to $+\infty$ due to the Hessian of f that is zero. This is in line with the properties of an affine classifier f, for which the direction provided by the gradient is parallel to the optimal direction needed to reach the boundary.

Moreover, for a highly irregular function, with many stationary points close to the boundary, the bound $\tilde\sigma$ could be close to zero, resulting in an extremely small tubular neighborhood for which the distance estimation holds.

In this section, we are interested in finding a value of ρ that provides the theoretically larger $\Omega_{\sigma(\rho)}$ for which Inequality (8) holds. In practice, this problem is hard to solve — it would require the complete knowledge of all the stationary points of f. However, the following observation brings to an interesting value ρ^* that provides a lower bound of the form

$$0 < \tilde\sigma_1(\rho^*) \le \max_{\sqrt{2}<\rho<2}\tilde\sigma(\rho) \le \max_{\sqrt{2}<\rho<2}\sigma(\rho), \tag{22}$$

where $\tilde\sigma_1(\rho^*)$ represents a lower bound of the largest σ for which Inequality (16) holds.

Observation 1 (Lower bound of largest σ)**.** Let $\Omega = \Omega_\delta$ of Lemma 2, and let α^* solving $\frac{1}{2}(1-\cos(\alpha^*)) = 2\cos(2\alpha^*)$. Then $\rho^* = 2\cos(\alpha^*)$ satisfies Equation (22).

Proof. Let $\Omega = \Omega_\delta$ of Lemma 2, let σ_1, σ_2 those in Lemmas 4 5, and let $\tilde\sigma(\rho) = \min\{\tilde\sigma_1(\alpha), \tilde\sigma_2(\beta)\}$, where α and β are such that $\rho = 2\cos(\alpha)$ and $\beta = \cos(2\alpha)$. Observe that, since $\mathcal{B} \subseteq \Omega_\delta$, then $\inf_{x\in\Omega}\|\nabla f(x)\| \le \inf_{x\in\mathcal{B}}\|\nabla f(x)\|$, and $\|\nabla^2 f\|_{\overline{\Omega}} \ge \|\nabla^2 f\|_{\mathcal{B}}$. Hence, we can consider the lower bound of $\tilde\sigma(\rho)$

$$\frac{\inf_{x\in\Omega}\|\nabla f(x)\|}{\|\nabla^2 f\|_{\overline{\Omega}}}\min\left(\frac{1}{2}(1-\cos(\alpha)), 2\cos(2\alpha)\right), \tag{23}$$

where $\rho = 2\cos(\alpha)$. And since function $u(\alpha) := \min(\frac{1}{2}(1-\cos(\alpha)), 2\cos(2\alpha))$ has maximum in α^*, the statement follows. □

In summary, the above results show that, given a neighborhood $\Omega = \Omega_\delta$ in which Equation (7) has unique solution (see Theorem 1) and there are no stationary points of classifier f (see Lemma 2), Inequality (8) holds for $\rho^* \approx 1.461$, and $\sigma^* := \sigma_1(\rho^*)$ is given by the observation above.

4.4 Error estimation for multi-class classifiers

The analysis above can be extended to a multi-class classifier by leveraging the two strategies presented in Section 3. In fact, if $f : \mathbb{R}^n \to \mathbb{R}^C$ is a classifier with C classes, both strategies reduce to a search for a solution of the minimal root problem RP for one or more binary classifiers in which the analysis above can be applied.

The Fast-Outer-Boundary strategy presented in Section 3.2.2 consists in solving Problem RP for a binary classifier of the form $L^{(l)} : \mathbb{R}^n \to \mathbb{R}$ where $L^{(l)}(x) := L(x,l) = f_l(x) - \max_{j\neq l} f_j(x)$. Thus, by applying Theorem 2 to $L^{(l)}$, we deduce the existence of a $\sigma^{(l)}(\rho)$ such that the estimation holds for each sample x with $\hat{k}(x) = l$. Therefore, by considering $\sigma(\rho) = \min_l \sigma^{(l)}(\rho)$, we obtain the same extension of Equation (8).

The Closest-Boundary strategy presented in Section 3.2.1 consists instead in solving Problem RP for a list of minimal root problems relative to binary classifiers of the form $f_{jl} = f_l - f_j$. In particular, for each ρ, Theorem 2 ensures the existence of a neighborhood with radius $\sigma_{jl}(\rho)$ such that the following inequalities holds

$$\frac{1}{\rho} t_j(x,l) \le d_j(x,l) \le t_j(x,l), \forall j,$$

where we keep the notation of Section 3.2.1. By taking the minimum over $j \neq l$ we deduce the estimation in Equation (8) for every x with $\hat{k}(x) = l$ and $x \in \Omega_{\sigma_l(\rho)}$, where $\sigma_l(x) = \min_{j\neq l} \sigma_{jl}(\rho)$. In conclusion, by considering $\sigma(\rho) = \min_{l,j\neq l} \sigma_{jl}(\rho)$, we deduce an extension of the desired inequality for the multi-class case.

5 Experiments

This section presents a set of experiments aimed at validating the strategies proposed in Section 3. They are executed on four neural classifiers, each trained on a different dataset. The approximate distances provided by the tested strategies are compared in Section 5.3 with the *Iterative Penalty* method (Section 5.1), which provides the ground-truth distance. Section 5.4 reports an empirical estimation of σ for three noticeable values of ρ. Finally, Section 5.5 discusses the case in which all the classifiers are attacked with different known methods. The magnitude of each attack is bounded to be lower than $t(x)/\rho^*$ in order to show that the attack success rate drops to zero for samples in $\Omega_{\hat{\sigma}^*}$, where $\hat{\sigma}^*$ is an estimation of σ^*.

5.1 Ground Truth Distance Estimation

In order to compare the approximate distances that solve Equation (RP), we need an accurate measure of the theoretical distance $d(x)$, which is practically unknown in the general case. To tackle this problem, based on the ideas presented in [35] and [6], we solve Equation (MP) by reducing to the following minimum problem with penalty analogous to Equation (1)

$$d(x,l;c) = \min_{\delta\in\mathbb{R}^n} \quad \|\delta\| + c \cdot L(x+\delta,l)^+, \tag{24}$$

where $L(x,l) = f_l(x) - \max_{j\neq l} f_j(x)$ and $L^+ = \max\{0, L\}$.

For each sample (x,l) and for each penalty value c, we perform a gradient descent with the Adam optimizer [36], with default parameters, up to 10^4 iterations, stopping the procedure when $-\mathtt{Tol} < L(x^{(k)},l) \le 0$, where the tolerance $\mathtt{Tol}$ is set to $5e^{-5}$. Note that this convergence criterion ensures that the solution lays close to the boundary and it is contained in the adversarial region $\cup_{j\neq l} R_j$.

Similarly to [6], the best penalty c is selected through a bisection-like search. In details, let $c_{\mathtt{low}} = 0$ and $c_{\mathtt{up}}$ such that $d(x,l;c_{\mathtt{low}}) = 0$ and $d(x,l;c_{\mathtt{up}})$ does not converge for all the samples x in the dataset. In our experiments, we discovered that $c_{\mathtt{up}} = 100$ is large enough to satisfy this definition. Then, through successive bisections, we can define $c_{\mathtt{curr}} = \frac{1}{2}\left(c_{\mathtt{low}} + c_{\mathtt{up}}\right)$ and either (i) set $c_{\mathtt{up}} = c_{\mathtt{curr}}$ (i.e., decreasing $c_{\mathtt{up}}$) if the optimization for $d(x,l;c_{\mathtt{curr}})$ does not converge, or (ii) set $c_{low} = c_{\mathtt{curr}}$ (i.e., increasing c_{low}) if it converges. We stop the search for c after 12 bisections. The whole procedure is implemented in batch mode to exploit GPU acceleration.

During the experiments, we noted that the Iterative Penalty (IP) method can provide, for a few of the tested samples, an estimation of $d(x)$ that is slightly higher than other global methods, such as DeepFool (DF) [5] and Decoupling Direction Norm (DDN) [9]. Thus, in order to adopt a more precise ground truth, we decided to consider for each sample x the ground-truth distance $d(x)$ as the minimum distance obtained with IP, DF, and DDN.

5.2 Experimental Settings

As done by Carlini and Wagner [37], the proposed techniques were evaluated on different datasets, each associated with a different neural network. In the following, we use the name of the dataset to refer to the experimental setting composed of the dataset itself and the corresponding network.

MNIST

The MNIST handwritten digits dataset [38] was used to train a vanilla LeNet [39] within a 2×2-MaxPool, 2 convolutional, and 3 fully connected layers, achieving a 1% error rate on the test set. The training was performed without data augmentation, using the Adam optimizer [36] (default hyperparameters) to minimize the Cross Entropy Loss with a 128 batch size for 5 epochs.

Fashion MNIST

This dataset includes 50,000 training images and 10,000 test images (28×28 greyscale pixels) grouped in 10 classes **[?]**. Compared to MNIST, this dataset is less trivial and requires a finer tuning to craft a model with a good accuracy. It was used to train a vanilla LeNet with the same structure of the previous one. The training was performed without data augmentation, by minimizing the Cross Entropy loss with the Adam optimizer for 30 epochs (with a batch size of 128) to achieve a 91% accuracy on the test set.

CIFAR10

This dataset contains 60,000 RGB images of size 32×32 pixels divided in 10 classes [40]. Inspired by [11], it was used to train a *Resnet32* model [41] over the first 50,000 images of the dataset with data augmentation, as described in the original paper. In details, the images were randomly cropped and horizontally flipped. The training was performed by minimizing the Cross Entropy loss for 182 epochs by the *stochastic gradient descent with Nesterov momentum* (SGD) [42] with a starting learning rate of 0.1, momentum of 0.9, and a weight decay of $1e-4$. The learning rate was decreased using a multiplicative factor of 0.1 after the 90th and the

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3195616

135th epoch, achieving a 8.8% error rate over the test set. This is in-line with the original results of [11].

GTSRB

The *German Traffic Sign Recognition Benchmark* [43] contains about 51,000 traffic signs RGB images of various shapes (from 15×15 to 250×250), grouped in 43 classes. It was used to train a *MicronNet* [44], a compact network similar to LeNet that classifies pixel-wise standardized 48×48 images. The training was performed over the first chunk of the dataset, containing $\approx$ 39,000 images with a data augmentation technique. During training, each image was randomly rotated by an angle in $\pm 5°$, translated towards a random direction with magnitude lower than 10%, and finally scaled with a factor between 0.9 and 1.1. Each transformed image was then scaled to have a dimension of 48 pixels per side. The model was trained to minimize the Cross Entropy loss by the SGD optimizer with a learning rate of $7e-3$, a momentum of 0.8, and a weight decay of $1e-5$, for 100 epochs. The learning rate was decreased every 10 epochs with a multiplicative factor of 0.9. We achieved a 1.2% error rate over the test set, which is comparable with the state-of-the-art classification performance with this dataset.

5.3 Comparing distances

This section focuses on comparing the estimated distances to the ground-truth distance for the four network models and corresponding data sets. For each sample (x, l), the approximate distances $t(x, l)$ are obtained by applying the zero finding algorithms (Bisection and Newton) to the strategies CB and FOB presented in Section 3. The ground-truth distance $d(x, l)$ is computed through the technique presented in Section 5.1.

Figure 3 shows a comparison between the approximate distance $t(x, l)$, computed by the Bisection CB strategy, and the ground-truth distance $d(x, l)$ for the four models considered in Section 5.2. For each sample x of label l, each dot in a graph represents the pair $(d(x, l), t(x, l))$. The dashed green line with slope 1 represents the points in which $d(x, l) = t(x, l)$. The other three lines have slopes $\sqrt{2}$, ρ^* and 2, where ρ^* is defined in Section 4, and represent the estimation of Equation (8) for different values of ρ. Observe that almost all the points close to the boundary (i.e., those with a small ground-truth distance to the boundary) are located above the green line and below the others, confirming that the estimation $t(x) \leq \rho d(x)$ holds.

Table 2 reports the average distances from the boundary for each dataset and for each tested strategy, and the average number of evaluations for a timing comparison. The statistics are computed over all the samples in the test set that satisfy the following conditions:

(i) the sample is correctly predicted by the model;
(ii) the algorithms reach the convergence;
(iii) the ground-truth distance is lower than 2.0 for MNIST and GTSRB, and lower than 0.5 for FMNIST and CIFAR10 (threshold values were selected to include a large part of the test set while still focusing on the region close to the boundary).

The amount of tested samples is detailed in Table 3. As one may expect, DF, DDN and Iterative Penalty (IP) provide lower distances with respect to our strategies CB and FOB. However, the distances computed by CB and FOB are associated with a bound on the approximation error relative to the theoretical distance $d(x)$.

The boxplot in Figure 4 provides a comparison of the approximate distances computed by the Bisection method applied to the CB strategy, DeepFool, IP, and DDN. The ground truth distance reported on the x-axis is partitioned, differently for each dataset, into four intervals, whose dimensions are summarized in Table 3.

Again, note that for points near the boundary, our method provides an accurate estimation of d, whereas, far from the boundary, a global techniques result to be more accurate, returning a better approximation of the ground-truth distance.

5.4 Estimation of $\sigma(\rho)$

Theoretically, Theorem 2 ensures that for each $\rho \in (\sqrt{2}, 2)$ there exists a $\sigma(\rho)$ for which Inequality (8) holds. In practice, however, for an arbitrary classifier f, such a $\sigma(\rho)$ cannot be deduced explicitly. Nevertheless, we can empirically estimate its value. In particular, given a data set $\mathcal{X}$, we can define $\hat{\sigma}(\rho)$, an estimation of $\sigma(\rho)$, as follows:

$$\hat{\sigma}(\rho) = \min \left\{ d(x, l) \, : \, \frac{t(x, l)}{d(x, l)} > \rho, \quad (x, l) \in \mathcal{X} \right\}, \tag{25}$$

which corresponds to the maximum distance for which Inequality (8) holds for the samples in $\mathcal{X}$.

Table 4 reports different estimations of σ for different values of ρ, in accordance with Section 5.3. For each ρ, the estimation $\hat{\sigma}(\rho)$ is deduced on a subset of the testset built by randomly sampling 60% of the images. Observe that the values of σ provided by CB are larger than or equal to those provided by FOB. In terms of algorithms, the customized bisection algorithm (augmented with the armijo-like rule) provides more reliable results with respect to the Newton method. We believe this is due to the fact that *there is no guarantee that the Newton algorithm provides the smallest positive zero of the function.*

These values can be seen as a measure of the regularity of the models: the higher $\hat{\sigma}$, the higher the regularity of the model (or the boundary). Also observe that these results are in line with Table 2, in which the model for FMNIST has an average distance that is lower than the one of the LeNet for MNIST (on which the images have the same dimension and have been normalized with same mean and standard deviation).

5.5 Adversarial robustness below $\hat{\sigma}$

This section evaluates the goodness of the empirical estimation $\hat{\sigma}^*$ of the theoretical σ^* (defined in Observation 1) to assess the model robustness against adversarial examples bounded in magnitude by $t(x)/\rho^*$.

In formulas, let $\tilde{x} = Adv_\varepsilon(x, l)$ an adversarial example crafted with an unknown attack technique Adv_ε that for each sample (x, l) provides a new sample $\tilde{x}$ (if exists) such that $\hat{k}(x) \neq l$ and $\|\tilde{x} - x\| \leq \varepsilon$. We want to empirically show that

$$\left\{ x \in \Omega_{\sigma^*} \, : \, \exists Adv_\varepsilon(x, \hat{k}(x); \varepsilon), \, \varepsilon < \frac{t(x)}{\rho^*} \right\} = \emptyset. \tag{26}$$

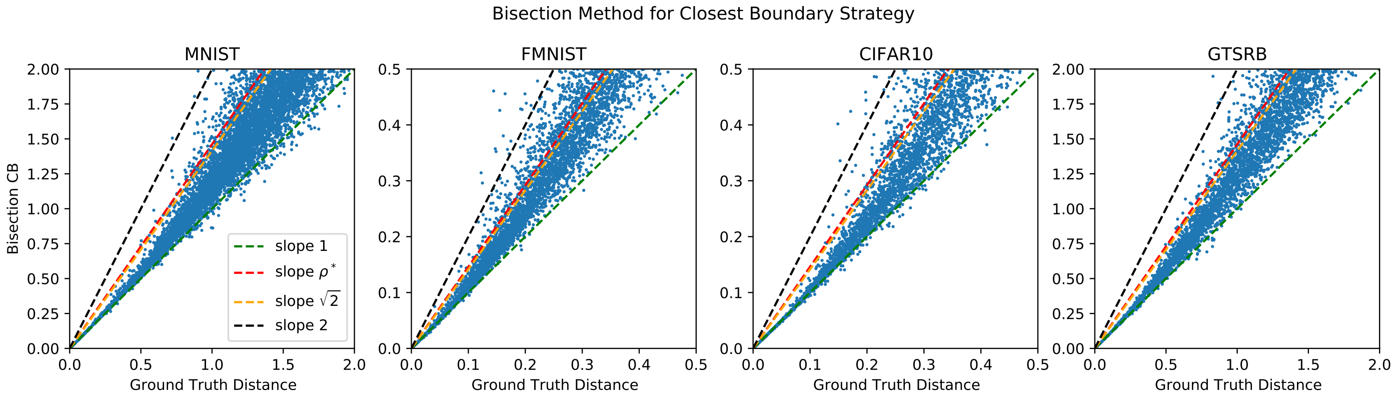

Fig. 3: Comparison of the approximate distance $t(x,l)$, computed by the Bisection CB strategy, and the ground-truth distance $d(x,l)$, for the four models considered in Section 5.2. Each dot represents the pair $(d(x,l), t(x,l))$ where x is a sample with label l. The region between the green line (slope 1) and the other lines (slope $\sqrt{2}$, ρ^*, 2), highlights the samples for which the Inequality 8 holds. Observe that, according to the theoretical results, the closer the boundary (small $d(x,l)$) the higher the number of dots in the region of interest.

		MNIST		FMNIST		CIFAR10		GTSRB	
		Avg.Dist.	# Evals	Avg.Dist.	# Evals	Avg.Dist.	# Evals	Avg.Dist.	# Evals
Strategy	Algorithm								
FOB	Bisection	1.645	17	0.455	16	0.508	17	2.221	17
CB	Bisection	1.467	17*†	0.385	16*†	0.483	17*†	1.667	17*†
FOB	Newton	1.641	3	0.442	3	0.496	4	2.169	4
CB	Newton	1.466	3*	0.385	3*	0.481	3*	1.668	3*
DF		1.526	2*	0.318	3*	0.346	3*	1.516	3*
DDN		1.287	1000	0.281	1000	0.338	1000	1.343	1000
IP		1.198	30162	0.261	32774	0.289	50609	1.262	34769
GT		1.172	-	0.255	-	0.283	-	1.204	-

* Average number of evaluations for each class of the datasets.
† Only one backward for each run. The remaining evaluations just perform forwards of the model.

TABLE 2: Average distance from the boundary and average number of evaluations of the models for the four datasets obtained with different methods. The behaviour of the tested methods for samples close to the boundary is detailed in Figure 4. Columns '# Evals' report the number of times the method requires a forward and a backward pass thorugh the model.

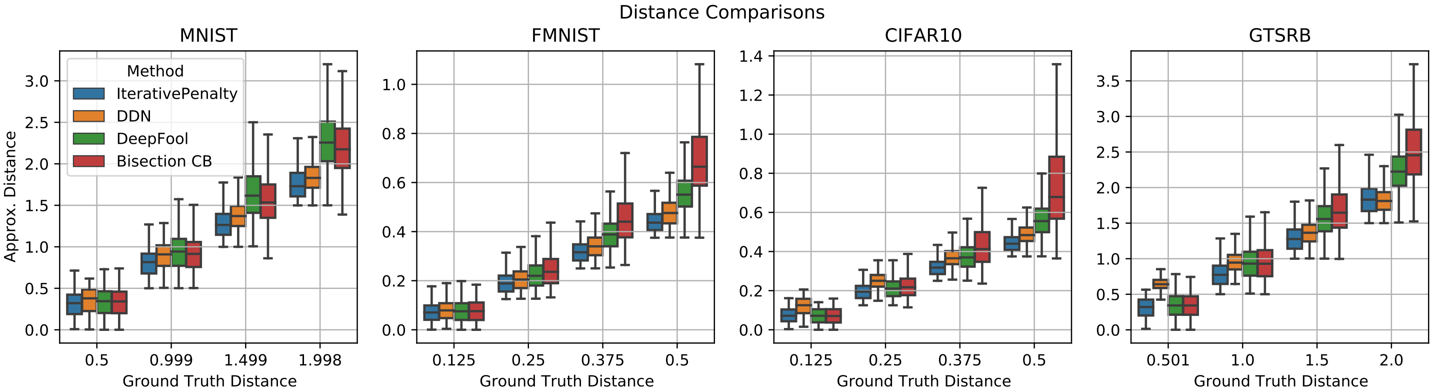

Fig. 4: Comparison of the distances computed by the Bisection CB strategy, DeepFool, Decoupling Direction Norm (DDN), and Iterative Penalty with respect to the ground-truth distance. For a clearer representation, the ground-truth distance is partitioned into four intervals that contains a number of samples summarized in Table 3. For each method, the lower and the upper side of each box represent the first and the fourth quartile Q_1 and Q_2, respectively; the lower and the upper whisker represent the quantiles $Q_1 - 1.5 \cdot I_q$ and $Q_3 + 1.5 \cdot I_q$, respectively, where I_q is the interquartile range.

In other words, we empirically assess that for each sample distant from the boundary less than $\hat{\sigma}^*$, there are no adversarial perturbations with a magnitude smaller than $t(x,l)/\rho^*$. For this purpose, we only test the approximation $t(x)$ provided by the CB strategy with the bisection method. In fact, higher values of $\hat{\sigma}$ represent a worst case to be tested, since there are more samples with a distance lower than $\hat{\sigma}$.

By using FoolBox [45], we generated adversarial examples for the four datasets with the following techniques: *Decoupling Norm Direction* (DDN) [9], *Deep Fool* (DF) [5], *Projected*

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3195616

	MNIST	FMNIST	CIFAR10	GTSRB
n_1	722	1198	849	576
n_2	2200	1723	1311	1347
n_3	3669	1767	1628	1696
n_4	2184	1350	1636	1866
Tot.	8775	6038	5424	5485

TABLE 3: Summary of the number of samples in each interval for each dataset. The MNIST and GTSRB are partitioned into intervals of length 0.5 from 0 to 2. The FMNIST and CIFAR10 are partitioned into intervals of length 0.125 from 0 to 0.5.

ρ	Algo.	Strategy	MNIST	FMNIST	CIFAR10	GTSRB
$\sqrt{2}$	B	FOB	0.34	0.06	0.04	0.15
		CB	0.37	0.06	0.13	0.58
	N	FOB	0.34	0.06	0.04	0.15
		CB	0.37	0.01	0.00	0.58
ρ^*	B	FOB	0.37	0.06	0.04	0.15
		CB	0.59	0.08	0.13	0.58
	N	FOB	0.37	0.06	0.04	0.15
		CB	0.59	0.01	0.00	0.58
2	B	FOB	0.37	0.12	0.11	0.49
		CB	0.72	0.12	0.17	0.87
	N	FOB	0.37	0.12	0.11	0.49
		CB	0.72	0.01	0.00	0.87

TABLE 4: Comparison of all the $\hat{\sigma}$ estimated by the different techniques.

Gradient Descent (PGD) [8], *Fast Gradient Method* (FGM) [45].

For each dataset $\mathcal{X}$, and for each sample $(x, l) \in \mathcal{X}$, we considered the `clipped` output of FoolBox that is guaranteed to have magnitude lower than ε, i.e. $\|\tilde{x} - x\| < \varepsilon$. Observe that in this test the magnitude of the attack ε is never computed by using the ground-truth distance $d(x, l)$, but by setting $\varepsilon = t(x, l)/\rho^*$.

The results of this experiment for the four datasets are shown in Figure 5, in which each graph reports the number of adversarial examples found with magnitude $t(x)/\rho^*$ as a function of the ground-truth distance $d(x, l)$. In detail, each stepped line reports, as a function of d, the cardinality of the set $\left\{(x, l) \in \mathcal{X} \,:\, \exists Adv_\varepsilon(x, l),\, \varepsilon = \frac{t(x)}{\rho^*},\, d(x, l) \leq d\right\}$ rescaled to be one for the maximum value of d, i.e, the fraction of points that are out of the bound for the tested attack. All graphs show that the higher d, the higher the number of samples that escapes the bounds (a sample escapes the bounds if $t(x, l)/\rho^*$ is higher than real distance from the boundary). In each plot, the values of $\hat{\sigma}^*$ computed in Table 4 are represented by the dashed red lines. It is important to observe that the estimation of $\hat{\sigma}^*$ was deduced as explained in the previous section, i.e., by applying Equation (25) without knowing the results of the attacks in advance.

The result of this test shows that the two datasets FMNIST and CIFAR10 have a different behavior with respect to MNIST and GTSRB. In particular, for MNIST and GTSRB, the estimation of σ^* is more selective, meaning that the estimation done by Inequality (8) holds for distances slightly larger than $\hat{\sigma}^*$. Moreover, for FMNIST and CIFAR10 datasets, the estimation of σ^* results to be less accurate, and for few samples (1 sample for each dataset) the attacks succeed even if the ground truth distance is lower than $\hat{\sigma}$, proving that the the estimation in Inequality 8 does not hold in a neighborhood of radius $\hat{\sigma}^*$ at least for one example.

5.6 Comparison with CLEVER

This section compares the estimation $t(x, l)/\rho^*$, obtained by the bisection method with the CB strategy, with the lower bound β_L obtained by CLEVER implemented by IBM in [46]. Note that both CLEVER and the CB strategy can provide estimates of different quality depending on the number of evaluations of the model. To fairly compare the quality of the results provided by the two methods, it is hence required to bound the maximum number of evaluations that they perform. In our experiments, this bound was set to 20, which was empirically selected by observing that the CB strategy requires only one gradient evaluation and (on average) at most 17 forward passes per class to converge. Note that the recommended amount of gradient evaluations of CLEVER is $500 * 1024$, which is clearly far from the bound we imposed. It is also worth remembering that, for a given sample, the Bisection method with CB strategy only performs *one* gradient evaluation at the first step (for each class), while all the following steps of the algorithm only require a forward pass of the model (and no gradient computations).

	Avg		Failures [%]		Evals [#]		$\#\{d \leq \hat{\sigma}^*\}$
	t/ρ^*	β_L	t/ρ^*	β_L	t/ρ^*	β_L	
Dataset							
MNIST	0.27	0.35	0.21	32.27	14.73	20.00	970
FMNIST	0.03	0.04	0.14	42.66	16.10	20.00	715
CIFAR10	0.05	0.07	0.12	48.94	17.04	20.00	852
GTSRB	0.28	0.39	0.54	66.58	15.40	20.00	745

TABLE 5: Comparison between the lower bound t/ρ^* and β_L of CLEVER. Only samples with ground truth distance lower than $\hat{\sigma}^*$ for each datasets are considered. Columns "Failures[%]" summarize the percentage of samples for which the estimated lower bounds are not lower than the ground-truth.

Table 5 provides a close comparison between the two lower bounds $t(x, l)/\rho^*$ and β_L. The metric named "Failure [%]" represents the percentage of samples for which the expected lower bounds are higher than $d(x, l)$. The metric named "Eval [#]" counts the mean number of evaluations of the models for each class. Observe that, for all the datasets, the amount of failures of the CB strategy is much lower than the one of CLEVER (β_L). To consider only the scenarios supported by our theoretical analysis from Section 4, only samples with a ground-truth distance lower than the corresponding empirical lower bound $\hat{\sigma}^*$ are considered.

6 Conclusions

This paper addressed the problem of estimating the minimal adversarial perturbation by presenting a novel strategy based on root-finding algorithms and providing theoretical guarantees on the goodness of the estimation. Indeed, differently from the state-of-the-art methods, which only focus on finding the minimal adversarial perturbation, the main contribution of this work is the derivation of a theoretical estimation of the error committed. Such a theoretical finding can be leveraged to verify the robustness of a classifier for a given input x close enough to the classification boundary. Furthermore, the approximate distance $t(x, l)$ obtained with

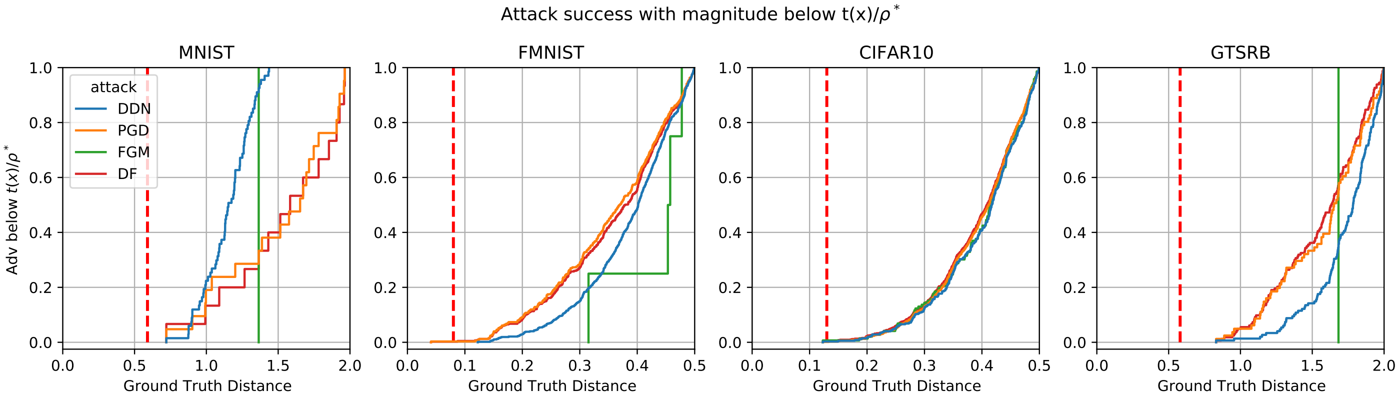

Fig. 5: Attack success rate cumulative curve for attacks bounded in magnitude less than $t(x)/\rho^*$ obtained with Bisection method and Closest Boundary strategy. The dashed red line represent $\hat{\sigma}^*$, which approximates σ^* of Theorem 2. For MNIST and GTSRB, none of the samples with distance from the boundary less than $\hat{\sigma}^*$ can be perturbed by the tested bounded attacks, in accordance with the theoretical results. For the FMNIST and the CIFAR10 dataset, instead, the estimation $\hat{\sigma}^*$ results to be less accurate, failing in a tiny portions of the tested samples (3 and 5 samples overall respectively).

the proposed approaches results to be less computationally expensive to compute than the distance $d(x, l)$ obtained with the aforementioned methods, enabling a fast verification of the ε-robustness of a classifier for the sample x.

The presented results open two interesting research directions to be addressed in future work. First, the estimated value $\hat{\sigma}$ only provides an empirical upper bound of the theoretical σ on a validation set, while there are no findings on the accuracy of such an empirical estimation with respect to the theoretical one. Second, as shown in Section 4.3, the theoretical bound σ^* depends on the first and the second derivatives of the model, which cannot be easily deduced for general DNN classifiers. Hence, future works should focus on leveraging σ^* to design more regular models, for which our analytical estimations hold for a larger amount of samples (i.e., for a larger Ω_σ) while preserving the classification accuracy. Towards this direction, promising recent works [47]–[49] focus on networks that have bounded $\|\nabla f\|$ by design. Based on this, we believe that a formal estimation of σ^* is a fundamental step for a fast and tight estimation of the boundary distance.

References

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, no. 3, pp. 211–252, 2015.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[3] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2013, pp. 387–402.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[5] S.-M. Moosavi-Dezfooli, A. Fawzi, and 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.

[6] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57.

[7] F. Croce and M. Hein, "Minimally distorted adversarial examples with a fast adaptive boundary attack," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2196–2205.

[8] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arxiv:1706.06083*, 2017.

[9] J. Rony, L. G. Hafemann, L. S. Oliveira, I. B. Ayed, R. Sabourin, and E. Granger, "Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4322–4330.

[10] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[11] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," *arXiv:1702.04267*, Feb 2017.

[12] G. Rossolini, A. Biondi, and G. Buttazzo, "Increasing the confidence of deep neural networks by coverage analysis," *IEEE Transactions on Software Engineering*, pp. 1–14, 2022.

[13] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. New York, NY, USA: Association for Computing Machinery, 2017, p. 3–14.

[14] F. Tramèr, A. Kurakin, N. Papernot, I. J. Goodfellow, D. Boneh, and P. D. McDaniel, "Ensemble adversarial training: Attacks and defenses." in *ICLR (Poster)*, 2018.

[15] F. Nesti, A. Biondi, and G. Buttazzo, "Detecting adversarial examples by input transformations, defense perturbations, and voting," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2021.

[16] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 582–597.

[17] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, no. 3, pp. 334–334, 1997.

[18] M. Pintor, F. Roli, W. Brendel, and B. Biggio, "Fast minimum-norm adversarial attacks through adaptive norm constraints," *Advances in Neural Information Processing Systems*, pp. 20 052–20 062, 2021.

[19] J. Rony, E. Granger, M. Pedersoli, and I. Ben Ayed, "Augmented lagrangian adversarial attacks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7738–7747.

[20] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, May 2014.

[21] A. Fawzi, O. Fawzi, and P. Frossard, "Analysis of classifiers' robustness to adversarial perturbations," *Machine Learning*, no. 3, pp. 481–508, 2018.

[22] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient smt solver for verifying deep neural networks," in *International conference on computer aided verification*. Springer, 2017, pp. 97–117.

[23] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *Proceedings of the 35th International Conference on Machine Learning*. PMLR, Jul 2018, p. 5286–5295.

[24] G. Singh, T. Gehr, M. Mirman, M. Püschel, and M. Vechev, "Fast and effective robustness certification," in *Advances in Neural Information Processing Systems*, 2018.

[25] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018.

[26] A. Boopathy, T.-W. Weng, P.-Y. Chen, S. Liu, and L. Daniel, "Cnn-cert: An efficient framework for certifying robustness of convolutional neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, no. 01, 2019, pp. 3240–3247.

[27] C. Liu, R. Tomioka, and V. Cevher, "On certifying non-uniform bounds against adversarial attacks," in *Proceedings of the 36th International Conference on Machine Learning*. PMLR, May 2019.

[28] K. Dvijotham, R. Stanforth, S. Gowal, T. A. Mann, and P. Kohli, "A dual approach to scalable verification of deep networks." in *UAI*, no. 2, 2018, p. 3.

[29] S. Wang, H. Zhang, K. Xu, X. Lin, S. Jana, C.-J. Hsieh, and J. Z. Kolter, "Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification," in *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2021, p. 29909–29921.

[30] L. Weng, H. Zhang, H. Chen, Z. Song, C.-J. Hsieh, L. Daniel, D. Boning, and I. Dhillon, "Towards fast computation of certified robustness for relu networks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5276–5285.

[31] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *Proceedings of the 36th International Conference on Machine Learning*. PMLR, May 2019, p. 1310–1320.

[32] K. Dvijotham, J. Hayes, B. Balle, J. Z. Kolter, C. Qin, A. György, K. Y. Xiao, S. Gowal, and P. Kohli, "A framework for robustness certification of smoothed classifiers using f-divergences," in *ICLR*, 2020.

[33] M. P. d. Carmo, *Differential geometry of curves and surfaces*. Prentice-Hall, OCLC: 1529515.

[34] L. Ambrosio and C. Mantegazza, "Curvature and distance function from a manifold," *The Journal of Geometric Analysis*, no. 5, 1998.

[35] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization." SIAM, 1995.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, Jan 2017.

[37] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, Nov 2017, p. 3–14.

[38] Y. LeCun, "The mnist database of handwritten digits," 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[39] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, no. 11, p. 2278–2324, Nov 1998.

[40] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[42] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning*. PMLR, May 2013, p. 1139–1147.

[43] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural networks*, pp. 323–332, 2012.

[44] A. Wong, M. J. Shafiee, and M. St. Jules, "Micronnet: A highly compact deep convolutional neural network architecture for real-time embedded traffic sign classification," *IEEE Access*, p. 59803–59810, 2018.

[45] J. Rauber, R. Zimmermann, M. Bethge, and W. Brendel, "Foolbox native: Fast adversarial attacks to benchmark the robustness of machine learning models in pytorch, tensorflow, and jax," *Journal of Open Source Software*, no. 53, p. 2607, 2020.

[46] "Adversarial robustness toolbox." [Online]. Available: https://github.com/Trusted-AI/adversarial-robustness-toolbox.git

[47] Q. Li, S. Haque, C. Anil, J. Lucas, R. B. Grosse, and J.-H. Jacobsen, "Preventing gradient attenuation in lipschitz constrained convolutional networks," *Advances in neural information processing systems*, 2019.

[48] A. Trockman and J. Z. Kolter, "Orthogonalizing convolutional layers with the cayley transform," in *International Conference on Learning Representations*, 2021.

[49] B. Kiani, R. Balestriero, Y. Lecun, and S. Lloyd, "projunn: efficient method for training deep networks with unitary matrices," *arXiv:2203.05483*, Mar 2022.

[50] G. Folland, "Higher-order derivatives and taylor's formula in several variables," *Preprint*, pp. 1–4, 2005.

[51] G. H. Golub and C. F. Van Loan, "Matrix computations. johns hopkins studies in the mathematical sciences." Johns Hopkins University Press, Baltimore, MD,, 1996.

Fabio Brau is a Ph.D. student at the Real-Time Systems (ReTiS) Laboratory of the Scuola Superiore Sant'Anna of Pisa with scolarship sponsored by Huawei Technologies Co.Ltd, under the supervision of Prof. Giorgio Buttazzo and A.P. Alessandro Biondi. He graduated (cum laude) in Mathematics at the University of Pisa with curriculum in Numerical Analysis. His current research topics are oriented to enhancing safety and trustworthiness of neural networks and machine learning algorithms with applications in safety critical systems.

Giulio Rossolini is a Ph.D. student at the Real-Time Systems (ReTiS) Laboratory of the Scuola Superiore Sant'Anna of Pisa. He graduated (cum laude) in Embedded Computing Systems Engineering, a Master's Degree jointly offered by the Scuola Superiore Sant'Anna of Pisa and University of Pisa. His current research interests include the design and implementation of software tools to support and increase the trustworthiness of machine learning algorithms used in computer vision applications and safety-critical systems.

Alessandro Biondi is associate professor at the Real-Time Systems (ReTiS) Laboratory of the Scuola Superiore Sant'Anna. He graduated (cum laude) in Computer Engineering at the University of Pisa, Italy, within the excellence program, and received a Ph.D. in computer engineering at the Scuola Superiore Sant'Anna under the supervision of Prof. Giorgio Buttazzo and Prof. Marco Di Natale. In 2016, he has been visiting scholar at the Max Planck Institute for Software Systems (Germany). His research interests include design and implementation of real-time operating systems and hypervisors, schedulability analysis, cyber-physical systems, synchronization protocols, and safe and secure machine learning. He was recipient of six Best Paper Awards, one Outstanding Paper Award, the ACM SIGBED Early Career Award 2019, and the EDAA Dissertation Award 2017.

Giorgio Buttazzo is full professor of computer engineering at the Scuola Superiore Sant'Anna of Pisa. He graduated in Electronic Engineering at the University of Pisa, received a M.S. degree in Computer Science at the University of Pennsylvania, and a Ph.D. in Computer Engineering at the Scuola Superiore Sant'Anna of Pisa. He has been Editor-in-Chief of Real-Time Systems, Associate Editor of the ACM Transactions on Cyber-Physical Systems, and IEEE fellow since 2012. He has authored 7 books on real-time systems and more than 300 papers in the field of real-time systems, robotics, and neural networks, receiving 13 best paper awards.