Robust optimization for adversarial learning with finite sample complexity guarantees

André Bertolace[1], Konstantinos Gatsis[2], Kostas Margellos[1]

***Abstract*— Decision making and learning in the presence of uncertainty has attracted significant attention in view of the increasing need to achieve robust and reliable operations. In the case where uncertainty stems from the presence of adversarial attacks this need is becoming more prominent. In this paper we focus on linear and nonlinear classification problems and propose a novel adversarial training method for robust classifiers, inspired by Support Vector Machine (SVM) margins. We view robustness under a data driven lens, and derive finite sample complexity bounds for both linear and non-linear classifiers in binary and multi-class scenarios. Notably, our bounds match natural classifiers' complexity. Our algorithm minimizes a worst-case surrogate loss using Linear Programming (LP) and Second Order Cone Programming (SOCP) for linear and non-linear models. Numerical experiments on the benchmark MNIST and CIFAR10 datasets show our approach's comparable performance to state-of-the-art methods, without needing adversarial examples during training. Our work offers a comprehensive framework for enhancing binary linear and non-linear classifier robustness, embedding robustness in learning under the presence of adversaries.**

I. Introduction

Decision making and learning in the presence of uncertainty have considered significant attention in recent years, in particular due to the advancements in the machine learning literature that have opened the road for data driven considerations. However, adversaries may manipulate data to compromise model outcomes [1], and as such call for robust solutions. Adversarial attacks, particularly in neural networks [1], [2], [3], [4], [5], [6], [7], [4], [8], have become a significant concern for safety-critical applications like autonomous driving [9], [7]. Various attack methods such as Limited-Memory BFGS [2], Fast Gradient Sign Method [3], and Projected Gradient Descent [10] have been explored. To address these, research has focused on developing defense mechanisms like defensive distillation [11] and feature squeezing [4], yet these defenses often lack comprehensive guarantees, highlighting the need for further research towards unifying attacks and defense mechanisms through robust optimization frameworks [10].

The relationship between robust optimization and adversarial machine learning is notably strong. Recent studies have introduced a probabilistic framework that effectively balances average and worst-case scenarios [12]. This framework also has ties to research on the so called scenario approach [13], particularly in its applications to Support Vector Machines (SVM) [14], [15], [16] and learning in general [17], [18]. Furthermore, the Lipschitz constant for deep neural networks (DNNs) has emerged as a valuable tool in certifying the robustness of classifiers and analyzing the stability of systems equipped with reinforcement learning controllers [19], [20].

A. Our methodology and contribution

In this paper, we present a novel adversarial training method inspired by SVM [21] margin concepts for binary and multi-class linear and non-linear classifiers. Unlike prior approaches, we analyze manipulations through classifier margins. Our contributions:

- Establishing sample complexity bounds within a probably approximately correct (PAC)-learning framework for robust classifiers, leveraging input and parameter space norms. Notably, linear classifiers' sample complexity scales as $m \sim \mathcal{O}(\frac{1}{\epsilon^2}\log\frac{2}{\delta})$, where ϵ is a prespecified classsification accuracy level, and δ denotes the confidence.
- Introducing a data-driven optimization-based adversarial training procedure using linear programming (LP) for linear models and second-order cone programming (SOCP) for non-linear ones.
- Validating our approach on MNIST and CIFAR10 datasets, typically used as benchmarks in classification studies, demonstrating comparable performance to state-of-the-art methods achieving (probabilistic) robustness without the need to generate adversarial examples during training, thus reducing computational effort.

Our work offers a comprehensive framework for robustness enhancement, eliminating the need for fine-tuning penalization coefficients and specific adversarial examples.

B. Related work

Our sample complexity bounds match those in[22] but without assuming adversary tampering per input. They achieve $\mathcal{O}(\frac{1}{\epsilon^2}(k\log(k)\text{VC}(\mathcal{H})+\log\frac{1}{\delta}))$ using a zero-sum game framework extended to multi-class and real-valued cases. Another work [23] shows that adversarial Rademacher complexity for binary linear classifiers is never smaller than natural Rademacher complexity, consistent with our findings. Another result, [24] achieved $\mathcal{O}(\frac{1}{m})$ expected standard loss for linear classifiers under separable data assumptions, whereas our approach adds flexibility by accommodating real-world datasets often not meeting separability assumptions. And [25] studied tolerant adversarial PAC-learning with a larger

 [1] Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, U.K. E-mail: andre.bertolace@eng.ox.ac.uk, kostas.margellos@eng.ox.ac.uk

[2]Department of Engineering Science, University of Southampton, Southampton, UK. E-mail: k.gatsis@soton.ac.uk

perturbation radius, deriving sample complexity bounds based on VC-dimension.

On the algorithmic side, [10] unified attacks and defenses through robust optimization, shaping adversarial machine learning. TRADES [26] proposed a method to trade adversarial robustness for accuracy by leveraging natural error and boundary error decomposition. SMART [27] introduces a technique considering misclassification and differentiating between misclassified and correctly classified examples during training.

II. Learning in the presence of an adversary

Let $\big(\Omega, \mathscr{F}, \mathcal{P}\big)$ be a probability space, and $\omega : \big(\Omega, \mathscr{F}\big) \to \big(Z, \mathscr{Z}\big)$ be a measurable mapping. Note that since ω is measurable we can define the image probability measure of $\mathcal{P}$ through ω, defined over the Borel σ-algebra on Z, as

$$\mathbb{P}(z) = \big(\mathcal{P} \circ \omega^{-1}\big)(z) = \mathcal{P}\big(\omega^{-1}(z)\big), \forall z \in \mathscr{Z}.$$

In addition, let $\mathcal{H}$ be a class of hypotheses or models. Each hypothesis $h \in \mathcal{H}$ is a function mapping $X \to Y$, where X represents the domain of features and Y the domain of response variables. In classification context such a hypothesis can be simply termed as classifier.

We are concerned with the learning problem, in which the learner aims at finding the best hypothesis h that minimizes a certain risk, i.e., $\inf_{h \in \mathcal{H}} \mathcal{R}_{\mathbb{P}}\big[\ell(h(x), y)\big]$, where $\ell : Y \times Y \to \mathbb{R}_+$ is a loss function and, for a fixed probability measure $\mathbb{P}$ and $\mathcal{R}_{\mathbb{P}}\big[\cdot\big]$ is a functional quantifying the risk. Generally the risk is taken to be the expected value associated with $\mathbb{P}$, leading to, finding h that minimizes

$$\inf_{h \in \mathcal{H}} \mathbb{E}_{\mathbb{P}}\big[\ell(h(x), y)\big]. \tag{1}$$

A. Adversarial attacks and related approaches

Considerations of adversarial attacks after learning involve a common modeling assumption [28], [29], [10], [30], [31] which dictates that an adversary can manipulate data features within a certain vicinity of the original example, x. Formally, given a data perturbation, or adversarial power, ξ, the manipulated sample is denoted by $\tilde{x} \in \mathcal{B}_\xi(x)$, where,

$$\mathcal{B}_\xi(x) := \Big\{\tilde{x} : \ \|\tilde{x} - x\| \leq \xi\Big\}, \tag{2}$$

and the choice of the norm used to measure the distance can be arbitrary. For more details on state-of-the-art attacks, such as the Fast Gradient Sign Method (FGSM), Projected Gradient Descent (PGD), Carlini and Wagner (CW), and Deep Fool, please refer to Appendix II for more information on these attacks.

An effective approach for learning models to defend against adversarial examples is a procedure called adversarial training [3], [10]. The core idea is to expose the model, during the training process, to adversarial examples crafted to intentionally deceive it. As a result, adversarially trained models learn to better defend against attacks, leading to increased predictability and reliability during inference. The concept emerged by studying the adversarial robustness of neural networks through the lens of robust optimization [10]. More precisely, the authors examined the following parameterized $\min - \max$ problem,

$$\min_{\theta} \mathbb{E}_{\mathbb{P}}\big[\max_{\tilde{x} \in \mathcal{B}_\xi(x)} \ell(\theta, \tilde{x}, y)\big]. \tag{3}$$

This formulation enabled the authors to cast both attacks and defenses within a common theoretical framework, naturally encapsulating most prior work on adversarial examples. Specifically, to reliably train models that are robust to adversarial attacks, they propose the adversarial empirical risk minimization (AERM) paradigm, where the learner does not know the distribution $\mathbb{P}$ but has access to m independently and identically distributed examples $S = \big(z_1, \ldots, z_m\big) \in Z^m$. Setting $Z = X \times Y$ and $\omega = (x, y)$; then each point $z_i = (x_i, y_i)$ is sampled from the fixed but possibly unknown distribution $\mathbb{P}$. Note that S induces a probability over Z^m which we will denote by the product measure $\mathbb{P}^m$.

$$\min_{\theta} \frac{1}{m} \sum_{i=1}^{m} \max_{\tilde{x}_i \in \mathcal{B}_\xi(x_i)} \ell(\theta, \tilde{x}_i, y_i). \tag{4}$$

This approach has a clear impact on the result of the optimization problem. Take as an example the linear regression problem where, given a set of samples S, the traditional (non-adversarial) learner proceeds by deciding on the values of a, b by means of the following ERM procedure,

$$\min_{a,b \in R^d} \frac{1}{2m} \sum_{i=1}^{m} \|y_i - (a^T x_i + b)\|_2^2.$$

On the contrary, the AERM learner would formulate the following robust optimization counterpart

$$\min_{a,b \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^{m} \|y_i - (a^T x_i + b + \xi \|a\|_*)\|_2^2,$$

where $\|\cdot\|_*$ denotes the dual norm, that emanates through the reformulation of a $\min - \max$ robust program as in (3). We refer to Appendix III for more details on the linear (Appendix III-A) and logistic (Appendix III-B) regression problems.

In the upcoming sections, we will show that our approach distinguishes itself from AERM as we remove the need to solve the inner maximization problem. Instead, we focus on the robust counterpart of Equation 4, whose theoretical PAC-learning guarantee is exposed in Section III-A, with the resulting optimization algorithms detailed in Section III-B.

Also, consider a gradient ascent step towards solving the inner maximization problem in (3).

$$\tilde{x} = x + \xi \cdot \text{sign}(\nabla_x \ell(h_\theta(x, y)).$$

This update results in adversarial examples and constitutes an attack. In particular, such an attack is considered for an ℓ_∞-bounded adversary in [3], and is referred to as the FGSM attack. We will employ such an attack for the numerical results presented in the sequel.

B. Proposed approach: margin-inspired adversarial training

Typically, for a binary classifier, the learner minimizes the natural classification error, with the loss function $\ell(h(x), y) = \mathbb{1}_{\{x\in X:\ y\cdot h(x)<0\}}$, where the natural risk is

$$\mathcal{R}_{\text{nat}}[h] = \mathbb{E}_{\mathbb{P}}[\mathbb{1}_{\{x\in X:\ y\cdot h(x)\leq 0\}}] = \int_{\Omega} \mathbb{1}_{\{x\in X:\ y\cdot h(x)\leq 0\}} d\mathbb{P}. \quad (5)$$

Drawing inspiration from SVM's margin theory (see Appendix IV), we explore the concept of confidence margin for binary classification tasks. Given a real-valued function h that operates on a data point x labeled with y, the confidence margin is defined as $h'(x, y) = y \cdot h(x)$. Thus, a correct classification by h occurs when $h'(x, y) > 0$, signifying that x is classified accurately. Notably, $|h(x)|$ can be interpreted as the level of confidence in the prediction made by h.

Recall that in the presence of an adversary, the classifier might encounter $\tilde{x} \in \mathcal{B}_\xi(x)$ as defined in (2). Although this constraint is imposed on the feature space X, our goal is to ensure PAC learnability for the class of functions $\mathcal{H}$. To achieve this, we limit ourselves to working with well-behaved functions. Specifically, we consider only functions h that are Lipschitz continuous, which means that there exists a constant B such that for all $x_0, x_1 \in X$, the following inequality holds,

$$\|h(x_1) - h(x_0)\| \leq B\|x_1 - x_0\|. \quad (6)$$

Similar to the definition of $\tilde{x} \in \mathcal{B}_\xi(x)$, we can define the neighborhood of the decision boundary of h, namely $DB(h)$, as [26], i.e.,

$$\mathcal{B}_\xi(DB(h)) = \{x \in X : \exists \tilde{x} \in \mathcal{B}_\xi(x) | h(x) \cdot h(\tilde{x}) \leq 0\}. \quad (7)$$

This motivates the definition of the robust and the boundary classification errors respectively as

$$\mathcal{R}^{\xi}_{\text{rob}}[h] = \mathbb{E}_{\mathbb{P}}[\mathbb{1}_{\{x\in X:\ \exists \tilde{x}\in\mathcal{B}_\xi(x)|y\cdot h(\tilde{x})\leq 0\}}], \quad (8)$$

$$\mathcal{R}^{\xi}_{\text{bdy}}[h] = \mathbb{E}_{\mathbb{P}}[\mathbb{1}_{\{x\in\mathcal{B}_\xi(DB(h))|y\cdot h(x)>0\}}]. \quad (9)$$

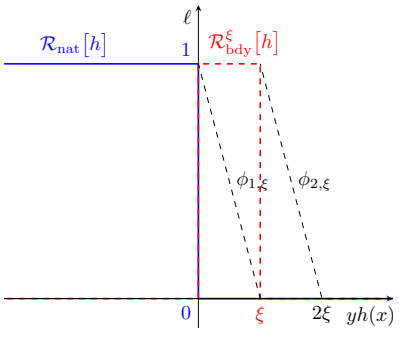

Fig. 1. Graphical representation of the decision boundary and errors: natural error (blue), boundary error (dashed-red), and robust error (dashed-black).

The boundary error measures the probability of points correctly classified but near the boundary, which might be misclassified by a powerful adversary. As a result of these definitions, the robust classification error can be decomposed into the natural classification error and the boundary classification error [26],

$$\mathcal{R}^{\xi}_{\text{rob}}[h] = \mathcal{R}_{\text{nat}}[h] + \mathcal{R}^{\xi}_{\text{bdy}}[h]. \quad (10)$$

In other words, by inspection of Fig. 1, $\mathcal{R}^{\xi}_{\text{bdy}}[h]$ constitutes a margin modification with respect to $\mathcal{R}_{\text{nat}}[h]$, to embed robustness towards example perturbations up to level ξ. Let $\phi_{\lambda,\xi}$ be the surrogate loss, as shown graphically in Fig. 1,

$$\phi_{\lambda,\xi}(z) = \left(\lambda - \frac{z}{\xi}\right)_+ = \max\left(0, \lambda - \frac{z}{\xi}\right). \quad (11)$$

Due to the dominance conditions induced by these surrogate loss functions, we then have that

$$\mathcal{R}_{\text{nat}}[h] \leq \mathbb{E}[\phi_{1,\xi}(y \cdot h(x))] \leq \mathcal{R}^{\xi}_{\text{rob}}[h] \leq \mathbb{E}[\phi_{2,\xi}(y \cdot h(x))].$$

Even though representing these sets in term of ξ is the most natural, it is easier to work directly with another constant $\zeta = B\xi$, where B is the Lipschitz constant of h as defined in (6). Thus, we can introduce the following set,

$$\mathcal{B}_\zeta(x) = \{\tilde{x} | \|h(\tilde{x}) - h(x)\| \leq \zeta\}. \quad (12)$$

Inspired by the definition of $\mathcal{B}_\xi(DB(h))$, in (7), we can also define,

$$\mathcal{B}_\zeta(DB(h)) = \{x \in X | \exists \tilde{x} \in \mathcal{B}_\zeta(x) \big| h(x) \cdot h(\tilde{x}) \leq 0\}. \quad (13)$$

It is worth mentioning that both sets $\mathcal{B}_\zeta(x)$ and $\mathcal{B}_\zeta(DB(h))$ contain the sets $\mathcal{B}_\xi(x)$ and $\mathcal{B}_\xi(DB(h))$ respectively.

This leads to a reformulation of the robust classification error,

$$\mathcal{R}^{\zeta}_{\text{rob}}[h] = \mathbb{E}_{\mathbb{P}}[\mathbb{1}_{\{x:\ \exists h(\tilde{x})\in\mathcal{B}_\zeta(x)|y\cdot h(\tilde{x})\leq 0\}}]. \quad (14)$$

We aim at designing PAC bounds for $\mathcal{R}^{\zeta}_{\text{rob}}[h]$, that as a result will constitute probabilistic classification statements. In the next section we show how to determine such bounds first for binary classifiers, and subsequently for multi-class ones.

III. Main Results

A. Sample complexity bounds

1) Binary classifiers: Theorem 3.1 below constitutes our main theoretical result. It provides PAC learning bounds on the robust classification error in (14). In particular, we provide complexity bounds for the sample size m, showing that the worst-case surrogate loss adversarial training method results in a learned classifier that can achieve a given classification accuracy of level ϵ, with confidence at least $1 - \delta$, for given $\epsilon, \delta \in (0, 1)$. The obtained sample size bounds provide explicit expressions for m as a function of ϵ, δ, the adversarial power ζ and the complexity of the class of hypotheses $\mathcal{H}$, represented by the Rademacher complexity, $\mathfrak{R}_m(\mathcal{H})$ (Definition 1.1 in Appendix I), that are of the same complexity with their non-adversarial counterparts up to the level of a constant. We show how to construct classifiers that enjoy such PAC properties in the following subsection.

Note that subsequent results involve considering an arbitrary ζ; this is effectively equivalent to fixing an arbitrary ξ.

Theorem 3.1: Binary classifier. Consider the hypothesis class $\mathcal{H}$ of Lipschitz continuous functions. Fix any $\zeta > 0$. We then have that, with probability at least $1-\delta$, for any $h \in \mathcal{H}$,

$$\mathcal{R}^{\zeta}_{\text{rob}}\big[h\big] \leq \frac{1}{m}\sum_{i=1}^{m}\phi_{2,\zeta}(y_i \cdot h(x_i)) + \frac{2}{\zeta}\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}.$$

The aforementioned bound holds uniformly, i.e., for $\gamma > 1$ and for any fixed $r > 0$, with probability at least $1-\delta$, for all $\zeta \in]0, r]$, and for any $h \in \mathcal{H}$,

$$\mathcal{R}^{\zeta}_{\text{rob}}\big[h\big] \leq \frac{1}{m}\sum_{i=1}^{m}\phi_{2,\zeta}(y_i \cdot h(x_i)) + \frac{2\gamma}{\zeta}\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log\log_{\gamma}\frac{\gamma r}{\zeta}}{m}} + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

Proof: Given the definitions (14) we use standard inequalities in the statistical learning theoretic literature and Talagrand's lemma to bound the Rademacher complexity of the loss functions by the complexity of the hypothesis class $\mathcal{H}$. In particular, we show the result holds for all $\zeta \in]0, r]$, by appropriately choosing series of $\zeta_k,\ \epsilon_k$ that converge uniformly. We refer to Appendix V-A for a complete proof. ∎

Remark 3.1: On the uniformity of the convergence statement. Uniform convergence not only strengthens the convergence notion but also empowers the learner to make informed decisions about the model's performance against various adversaries, making it a valuable tool in practical machine learning applications. For instance, the first inequality in Theorem 3.1 allows the learner to calculate the required sample size to achieve a desired level of accuracy and confidence against one adversary. However, when confronted with a stronger adversary, the learner is not be able to provide the same guarantees regarding the model's accuracy or confidence level. Instead, considering the uniform convergence case, the learner is able to ensure accuracy and confidence for a range of adversaries just after training. In other words, once the model has been trained, the learner can confidently assert its performance regarding accuracy and confidence against a range of adversaries.

When comparing both inequalities in Theorem 3.1, we notice that the statement with uniform convergence has only a small impact the sample complexity, as the order of the sample complexity remains unchanged, differing only by some constants.

Remark 3.2: Price of robustness. For the binary classifier, in a standard learning process without an adversary, by Theorem 1.2 we have that with probability at least $1-\delta$,

$$\mathcal{R}_{\text{nat}}\big[h\big] \leq \frac{1}{m}\sum_{i=1}^{m}\phi_{2,\zeta}(y_i \cdot h(x_i)) + \mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}.$$

For the sake of simplicity, consider a strong adversary and choose $\gamma = \zeta = r > 1$. In this case, the inequality stated in Theorem 3.1 takes a simpler form,

$$\mathcal{R}^{\zeta}_{\text{rob}}\big[h\big] \leq \frac{1}{m}\sum_{i=1}^{m}\phi_{2,\zeta}(y_i \cdot h(x_i)) + 2\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

This indicates that the sample complexity has the same order as the natural training procedure, only differing by constants influencing the Rademacher complexity.

Remark 3.3: Effect of the adversarial power. As the parameter ζ increases, the influence of the Rademacher complexity of the hypothesis class on the sample complexity diminishes. However, a higher ζ also leads to a more loose approximation of the desired loss by the surrogate loss as seen in Fig 1, indicating a deterioration in the quality of the approximation. In this case, it is intuitive that fewer observations are required to satisfy a more conservative inequality. Conversely, when ζ is small, we witness the opposite effect, the surrogate loss approximation becomes tighter, at the cost of a higher dependency on the Rademacher complexity of the hypothesis class.

In the following results we choose to state the uniform convergence version of the theorem. A non-uniform version is easily achievable by omitting the extra term in the equations.

2) Linear binary classifiers with bounded inputs: We now specialize attention to the case where $\mathcal{H}$ is the class of affine functions $a^T x + b$ in $\mathbb{R}^d$. For non-homogeneous half spaces in $\mathbb{R}^d$, The Rademacher complexity $\mathfrak{R}_m(\mathcal{H})$ for such classes can be bounded by a function of the VC dimension, which is in turn bounded by $d+1$ (see Theorem 1.4),

$$\mathcal{R}^{\zeta}_{\text{rob}}\big[h\big] \leq \frac{1}{m}\sum_{i=1}^{m}\phi_{2,\zeta}(y_i \cdot h(x_i)) + \sqrt{\frac{\log\log_{\gamma}\frac{\gamma r}{\zeta}}{m}} + \frac{2\gamma}{\zeta}\sqrt{\frac{2(d+1)\log\frac{em}{d+1}}{m}} + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

Compact spaces reduce the impact of dimensionality d on sample complexity, as demonstrated in the following lemma and corollary. They remove the reliance on the Rademacher complexity of class $\mathcal{H}$ by utilizing norm and dual-norm bounds in the input and parameter space. While this may appear as a stringent restriction, it is natural in image classification problems as pixels have maximum attainable values.

Lemma 3.1: Let $\mathcal{H}$ the hypothesis class of affine functions as defined above. Assume that, for all $x \in X$, $\|x'\| \leq u$, with $x' = [x^T, 1]^T$, and that $\|w\|_* \leq v$, where $w = [a^T, b]^T$. We then have that $\mathfrak{R}_m(\mathcal{H}) \leq \sqrt{\frac{u^2 v^2}{m}}$.

Proof: The proof bounds the Rademacher complexity through the use of the dual norm, which is typically employed to reformulate robust optimization programs. The complete proof is available at Appendix V-B. ∎

A linear classifier for this case can be computed by means of a linear optimization program as shown in Section III-B.1. However, prior to discussing this we show the probabilistic error classification guarantees that accompany such a classifier.

Corollary 3.1: Linear binary classifier with bounded inputs. Let $\mathcal{H} = \{x \to a^T x + b,\ a \in \mathbb{R}^d,\ b \in \mathbb{R}\}$. Assume that $\{x \in X | \|x'\| \leq u\}$ and that $\|w\|_* \leq v$, where $w = [a^T, b]^T$. We then have that for any $\gamma > 1$ and any $r > 0$, with probability at least $1-\delta$, for any $\xi \in]0, \frac{r}{v}]$, and for any linear

classifier (parameterized by a, b),

$$\mathcal{R}^{\zeta}_{\text{rob}}\big[h\big] \leq \frac{1}{m}\sum_{i=1}^{m}\phi_{2,v\xi}(y_i w^T x_i') + \frac{2\gamma}{\xi}\sqrt{\frac{u^2}{m}} + \sqrt{\frac{\log\log_\gamma \frac{\gamma r}{v\xi}}{m}} + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

Proof: The proof follows from Theorem 3.1 and Lemma 3.1 and is provided in Appendix V-C. ∎

3) Kernel-based non-linear binary classifiers: The so called "kernel" approach, commonly used for non-linear classifiers [32], embeds the input space into a higher-dimensional feature space and employs a linear classifier there. This enables non-linear classification in the original input space. However, applying this approach can be challenging due to potential infinite-dimensional feature spaces or the need for a large number of sample points to achieve desired accuracy. To address these challenges, kernel-based learning approaches provide a solution.

Definition 3.1: Kernel. Given an embedding $\psi : X \to \mathbb{H}$, mapping the domain space into some Hilbert space, we define the Kernel function as $K(x, x') = \langle \psi(x), \psi(x') \rangle$, for all $x, x' \in X$.

Definition 3.2: Positive Definite Symmetric (PDS) kernels. A kernel $K : X \times X \to \mathbb{R}$ is said to be posititive definite symmetric (PDS) if for any $x_1, \ldots, x_m \subseteq X$, the matrix $\mathbf{K} = [k(x_i, x_j)]_{ij} \in \mathbb{R}^{m\times m}$ is symmetric positive semidefinite.

Kernel-based classifiers can be computed by means of a second-order cone program as shown in Section III-B.2. However, prior to discussing this, the following corollary of Theorem 3.1 shows the probabilistic error classification guarantees that accompany such a classifier.

Corollary 3.2: Let $K : X \times X \to \mathbb{R}$ be a PDS kernel, $\mathbb{H}$ its corresponding RKHS, (Theorem1.5), equipped a the norm $\|\cdot\|_{\mathbb{H}}$, and $\psi : X \to \mathbb{H}$, the feature map associated with it. Let $\mathcal{H} = \{x \to w^T\psi(x), x \in X, w \in \mathbb{H}\}$. Assume that $\|w\|_{\mathbb{H}} \leq v$ and that $K(x, x) < u^2$. We then have that with probability at least $1 - \delta$,

$$\min_{w\in\mathbb{H}} \mathcal{R}^{\zeta}_{\text{rob}}\big[h\big] - \min_{\alpha^T\mathbf{K}\alpha \leq v^2} \frac{1}{m}\sum_{i=1}^{m}\phi_{2,\zeta}(y_i(\mathbf{K}\alpha)_i) \leq \frac{2\gamma}{\zeta}\sqrt{\frac{u^2v^2}{m}} + \sqrt{\frac{\log\log_\gamma \frac{\gamma r}{\zeta}}{m}} + \sqrt{\frac{\log\frac{2}{\delta}}{2m}},$$

where $\mathbf{K} = [K(x_i, x_j)]_{ij}$ is a symmetric positive semi-definite matrix and $\alpha \in X^m$.

Proof: The proof is provided in Appendix V-D. ∎

4) Multi-class classifiers: For multi-class classifiers with k classes, the preferred method involves employing scoring functions h, enabling the classifier to determine the class associated with the highest score. This approach establishes a mapping [33] between the input data and the class that yields the maximum score,

$$x \in \underset{y\in\{1,\ldots,k\}}{\text{argmax}}\ h(x, y).$$

Similar to the binary classification case, it is possible to generalize the concept of confidence margin, by defining

$$h'(x, y) = h(x, y) - \max_{y'\neq y} h(x, y'). \tag{15}$$

Note that if h misclassifies (x, y), then $h'(x, y) < 0$. In this case, the learner's goal is to minimize the natural classification error,

$$\mathcal{R}_{\text{nat}}\big[h\big] = \mathbb{E}_{\mathbb{P}}\big[\mathbb{1}_{\{x\in X:\ (h(x,y) - \max\limits_{y'\neq y} h(x,y'))\leq 0\}}\big]. \tag{16}$$

We focus on an adversary aiming to deceive the classifier by inducing a mistake, without targeting a specific class-to-class transformation. This involves manipulating the input to be classified as the closest class, without a particular target class. Similar to the binary classifier, we define $\mathcal{R}^{\zeta}_{\text{rob}}\big[h\big]$,

$$\mathcal{R}^{\zeta}_{\text{rob}}\big[h\big] = \mathbb{E}_{\mathbb{P}}\big[\mathbb{1}_{\{x:\ \exists h(\tilde{x})\in\mathcal{B}_\zeta(x)|h'(\tilde{x},y)\leq 0\}}\big]. \tag{17}$$

Note the similarity of this definition with proposed in (14).

Theorem 3.2: Upper bounded multi-class classifier: Consider the hypothesis class of scoring functions $\mathcal{H} = \{(x, y) \to h(x, y)\}$. We then have that for any $r > 0$ and any $\zeta \in]0, r]$, with probability at least $1 - \delta$,

$$\mathcal{R}^{\zeta}_{\text{rob}}\big[h\big] \leq \frac{1}{m}\sum_{i=1}^{m}\phi_{2,\zeta}(h'(x_i, y_i)) + \frac{2k\gamma}{\zeta}\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log\log_\gamma \frac{\gamma r}{\zeta}}{m}} + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

Proof: The proof follows the same steps as that of Theorem 3.1, only deviating on the bounding of the Rademacher complexity $\mathfrak{R}_m(\mathcal{H}')$. The proof is provided in Appendix V-E. ∎

The result is similar to Theorem 3.1, with only a scaling constant k difference. With more classes, a larger sample size is required for model training, and this dependency is linear with the number of classes.

B. Classifier computation

In this section we discuss how to compute linear and kernel-based classifiers using empirical data that enjoy the probabilistic guarantees of Corollaries 3.1 and 3.2, respectively.

1) Linear programming formulation for linear binary classifiers: Let $\|x'\|_\infty \leq u = r$, $\|w\|_1 \leq v = 1$, and $r = 1$, i.e., we normalize the input to its maximum value, and further consider a powerful adversary, such that $\xi = r = 1$. By Corollary 3.1, considering the explicit expression of the surrogate function, have that with confidence at least $1 - \delta$, for any linear classifier parameterized by $w \in \mathbb{R}^{d+1}$,

$$\min_{w} \mathcal{R}_{\text{rob}}\big[h\big] - \min_{\|w\|_1 \leq 1} \frac{1}{m}\sum_{i=1}^{m}\big(2 - y_i w^T x_i'\big)_+ \leq \sqrt{\frac{\log\frac{2}{\delta}}{2m}} + 1. \tag{18}$$

where we omitted the superscript ζ in $\mathcal{R}_{\text{rob}}\big[h\big]$ since $\zeta = 1$ based on the discussion above. We get this simplified bound

by setting $\gamma = \sqrt{m}/2$, while the requirement of $\gamma > 1$ holds for any $m > 4$.

A classifier that enjoys such guarantees can be constructed as the solution of the empirical minimization of the second term in the previous equation. This is a minimization subject to a first norm constraint. We could equivalently recast this as a linear program (LP) by introducing some additional decision variables. The resulting optimization program is given by

$$\begin{aligned} \min_{w,t,l,\in\mathbb{R}^{2(d+1)+m}} \quad & \frac{1}{m}\sum_{i=1}^{m} t_i \\ \text{s.t.} \quad & 2 - \frac{y_i w^T x_i'}{\xi} \le t_i, \\ & t_i, l_j \ge 0, \\ & w_j \le l_j,\ -w_j \le l_j, \\ & \sum_{j=1}^{d+1} l_j \le 1. \end{aligned} \tag{19}$$

This result is similar to SVM, however, it involves a different norm. We discuss these similarities in more detail in Appendix IV-A.

2) Second order cone programming formulation for kernel-based binary classifiers: By Corollary 3.2, we have that with probability at least $1-\delta$,

$$\min_{w\in\mathbb{H}} \mathcal{R}_{\text{rob}}[h] - \min_{\alpha^T\mathbf{K}\alpha \le v^2} \frac{1}{m}\sum_{i=1}^{m}\Big(2 - \frac{y_i(\mathbf{K}\alpha)_i}{\zeta}\Big)_+ \le \sqrt{\frac{\log\frac{2}{\delta}}{2m}} + 1, \tag{20}$$

where $\alpha \in \mathbb{R}^m$. Similarly to the case of linear classifier, we obtained this by letting $r = 1$, taking $u = v = \zeta = r = 1$ and setting γ as in the previous section.

The classifier that enjoys such classification guarantees can be obtained as the solution of the empirical minimization problem that appears as the second term in the previous equation. This can be equivalently written as a second-order cone program (SOCP), given by

$$\begin{aligned} \min_{\alpha,t_1,\ldots,t_m\in\mathbb{R}^{2m}} \quad & \frac{1}{m}\sum_{i=1}^{m} t_i \\ \text{s.t.} \quad & y_i(\mathbf{K}\alpha)_i \ge 2 - t_i, \\ & t_i \ge 0, \forall i \in 1,\ldots,m, \\ & \|\mathbf{L}\alpha\|_2 \le v^2, \end{aligned} \tag{21}$$

where $K = \mathbf{L}^T\mathbf{L}$, that is, $\mathbf{L}$ can be obtained through the Cholesky decomposition of $\mathbf{K}$.

IV. Numerical experiments

A. Simulation set-up

In our numerical examples, we conduct a series of experiments using a binary linear classifier applied to the MNIST and CIFAR10 data-sets. We refer to Appendix VI for the details about the parameters used in the numerical analysis and a reference to the the Github repository with the available code. We employ the proportion of correctly classified instances in an out-of-sample test set as our accuracy metric for performance evaluation. Our evaluation extends beyond the conventional test case, as all instances in the test set are manipulated with adversarial examples crafted to mislead the model. Despite this challenging scenario, our margin-based model showcases competitive performance, on par with state-of-the-art adversarial training techniques, across both datasets.

B. Simulation results

When evaluating the NIST dataset, our goal is to differentiate between two digit pairs (0/1 and 3/8) using a binary classifier. Figures 2 (a) and (b) show that standard training is highly vulnerable to adversarial perturbations. For the 0/1 case, both our margin-based approach and the FGSM adversarial training achieve excellent performance, with the latter slightly outperforming under stronger attacks. In the more challenging 3/8 case, our margin-based approach clearly outshines other adversarial training methods.

Data	ξ	Non-adv. [%]	Margin-based [%]
0/1	0.05	99.95 (2.17)	99.91 (3.07)
	0.10	99.95 (2.20)	99.85 (3.64)
	0.15	99.94 (2.24)	99.19 (8.33)
	0.20	99.94 (2.27)	99.52 (5.64)
3/8	0.05	96.74 (17.38)	95.96 (19.29)
	0.10	96.70 (17.15)	92.05 (26.14)
	0.15	96.67 (16.94)	85.61 (33.39)
	0.20	96.64 (16.71)	85.99 (31.22)
Plane/Dog	0.05	68.73 (46.20)	83.01 (36.89)
	0.10	68.73 (46.02)	79.00 (39.25)
	0.15	68.72 (45.87)	76.86 (39.42)
	0.20	68.69 (45.72)	71.58 (41.05)
Cat/Dog	0.05	51.70 (49.81)	61.21 (47.12)
	0.10	51.72 (49.68)	58.13 (45.77)
	0.15	51.74 (49.58)	57.07 (42.27)
	0.20	51.74 (49.49)	56.21 (37.76)

TABLE I

RoMa score: mean and standard deviation for non-adversarial and margin-based training methods.

In the context of the CIFAR10 dataset, it is crucial to note that a linear classifier exhibits low accuracy even without any adversarial influence. Specifically, in the cat/dog classification, a linear model struggles even without adversaries, performing no better than chance, as depicted in Figure 2 (d). However, our margin-based approach demonstrates robust classification, achieving notable accuracy even in challenging scenarios such as distinguishing between airplane/dog, outperforming all methods, and surpassing usual training in the cat/dog case, as shown in Figure 2 charts (c) and (d).

Robustness against adversarial inputs is a crucial factor, evaluated using the RoMA (Robustness Measurement and Assessment) procedure [34]. This method determines the probability of a random input perturbation causing a mis-classification, providing guarantees on the expected error frequency post-training [34]. In this metric (Table IV-B), both the proposed margin-based and conventional training methods show comparable robustness, especially in scenarios where adversaries find it challenging to execute attacks, as

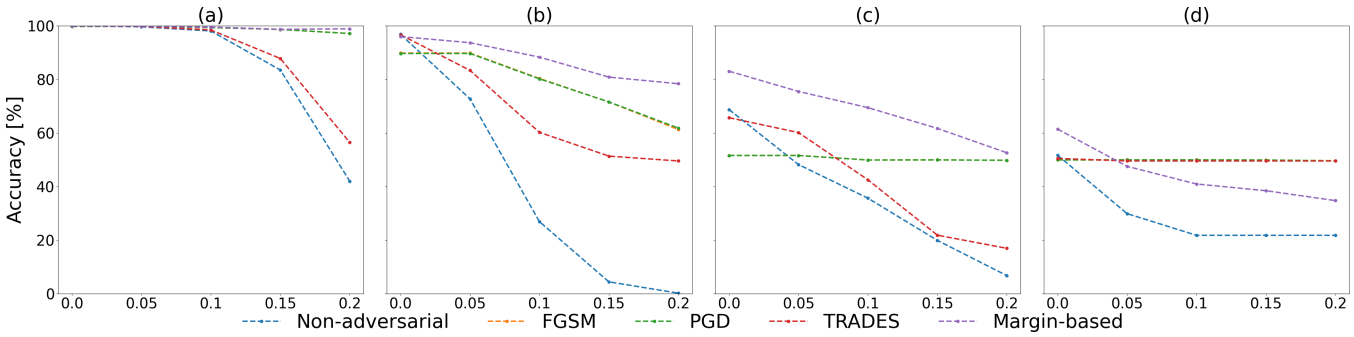

Fig. 2. Accuracy of linear classifiers using out-of-sample adversarial tampered data considering non-adversarial training, FGSM [3], PGD [10], TRADES [$\lambda = 1.0$] [26] and proposed margin-based approach. Datasets: (a) NIST 0/1, (b) NIST 3/8, (c) CIFAR10 Airplane/Dog and (d) CIFAR10 Cat/Dog.

observed in the NIST 0/1 dataset. However, the margin-based approach demonstrates significantly higher robustness on datasets with lower accuracy under conventional training, such as the CIFAR10.

V. Conclusion

We focused on robust classification under adversarial attacks and introduced a new method for adversarial training, inspired by SVM margin concepts. We established finite sample complexity bounds that accompany adversarialy trained classifiers with probabilistic error classification guarantees. Moreover, we showed that robust linear and kernel-based binary classifiers can be constructed by means of a linear and a second-order cone program respectively. Extensive numerical validation was provided. A distinctive feature of the proposed methodology is the ability to achieve high accuracy without generating adversarial examples during training.

References

[1] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma, "Adversarial classification," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, (New York, NY, USA), p. 99–108, Association for Computing Machinery, 2004.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2014.

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015.

[4] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317 – 331, 2018.

[5] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," *CoRR*, vol. abs/1810.00069, 2018.

[6] R. S. S. Kumar, D. R. O'Brien, K. Albert, S. Viljöen, and J. Snover, "Failure modes in machine learning systems," *CoRR*, vol. abs/1911.11034, 2019.

[7] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805–2824, 2019.

[8] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent advances in adversarial training for adversarial robustness," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21* (Z.-H. Zhou, ed.), pp. 4312–4321, International Joint Conferences on Artificial Intelligence Organization, 8 2021. Survey Track.

[9] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *CoRR*, vol. abs/1607.02533, 2016.

[10] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2019.

[11] 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)*, pp. 582–597, 2016.

[12] A. Robey, L. Chamon, G. J. Pappas, and H. Hassani, "Probabilistically robust learning: Balancing average and worst-case performance," in *Proceedings of the 39th International Conference on Machine Learning* (K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, eds.), vol. 162 of *Proceedings of Machine Learning Research*, pp. 18667–18686, PMLR, 17–23 Jul 2022.

[13] M. C. Campi and S. Garatti, *Introduction to the Scenario Approach*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2018.

[14] M. C. Campi and S. Garatti, "The exact feasibility of randomized solutions of uncertain convex programs," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1211–1230, 2008.

[15] S. Garatti and M. C. Campi, "Risk and complexity in scenario optimization," *Mathematical Programming*, vol. 191, pp. 243–279, Jan 2022[2019].

[16] M. C. Campi and S. Garatti, "A theory of the risk for optimization with relaxation and its application to support vector machines," *Journal of Machine Learning Research*, vol. 22, no. 288, pp. 1–38, 2021.

[17] K. Margellos, M. Prandini, and J. Lygeros, "A compression learning perspective to scenario based optimization," in *53rd IEEE Conference on Decision and Control*, pp. 5997–6002, 2014.

[18] M. C. Campi and S. Garatti, "Compression, generalization and learning," *Journal of Machine Learning Research*, vol. 24, no. 339, pp. 1–74, 2023.

[19] P. Pauli, A. Koch, J. Berberich, P. Kohler, and F. Allgöwer, "Training robust neural networks using lipschitz bounds," *IEEE Control Systems Letters*, vol. 6, pp. 121–126, 2022.

[20] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. Pappas, "Efficient and accurate estimation of lipschitz constants for deep neural networks," in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[21] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[22] I. Attias, A. Kontorovich, and Y. Mansour, "Improved generalization bounds for robust learning," in *Proceedings of the 30th International Conference on Algorithmic Learning Theory* (A. Garivier and S. Kale, eds.), vol. 98 of *Proceedings of Machine Learning Research*, pp. 162–183, PMLR, 22–24 Mar 2019.

[23] D. Yin, R. Kannan, and P. Bartlett, "Rademacher complexity for adversarially robust generalization," in *Proceedings of the 36th International Conference on Machine Learning* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 7085–7094, PMLR, 09–15 Jun 2019.

[24] R. Bhattacharjee, S. Jha, and K. Chaudhuri, "Sample complexity of robust linear classification on separated data," in *Proceedings of the 38th International Conference on Machine Learning* (M. Meila and T. Zhang, eds.), vol. 139 of *Proceedings of Machine Learning Research*, pp. 884–893, PMLR, 18–24 Jul 2021.

[25] H. Ashtiani, V. Pathak, and R. Urner, "Adversarially robust learning with tolerance," in *Proceedings of The 34th International Conference on Algorithmic Learning Theory* (S. Agrawal and F. Orabona, eds.),

vol. 201 of *Proceedings of Machine Learning Research*, pp. 115–135, PMLR, 20 Feb–23 Feb 2023.
[26] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. E. Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *Proceedings of the 36th International Conference on Machine Learning* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 7472–7482, PMLR, 09–15 Jun 2019.
[27] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu, "Improving adversarial robustness requires revisiting misclassified examples," in *ICLR*, 2020.
[28] D. Cullina, A. N. Bhagoji, and P. Mittal, "Pac-learning in the presence of evasion adversaries," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, (Red Hook, NY, USA), p. 228–239, Curran Associates Inc., 2018.
[29] U. Feige, Y. Mansour, and R. Schapire, "Learning and inference in the presence of corrupted inputs," in *Proceedings of The 28th Conference on Learning Theory* (P. Grünwald, E. Hazan, and S. Kale, eds.), vol. 40 of *Proceedings of Machine Learning Research*, (Paris, France), pp. 637–657, PMLR, 03–06 Jul 2015.
[30] J. Lee and M. Raginsky, "Minimax statistical learning with wasserstein distances," in *Advances in Neural Information Processing Systems* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), vol. 31, Curran Associates, Inc., 2018.
[31] Z. Tu, J. Zhang, and D. Tao, "Theoretical analysis of adversarial learning: A minimax approach," in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019.
[32] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. USA: Cambridge University Press, 2014.
[33] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. The MIT Press, 2nd ed., 2018.
[34] N. Levy and G. Katz, "Roma: A method for neural network robustness measurement and assessment," in *Proc. 29th Int. Conf. on Neural Information Processing (ICONIP)*, pp. 92–105, November 2022.
[35] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017.
[36] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2574–2582, 2016.
[37] B. Biggio, B. Nelson, and P. Laskov, "Support vector machines under adversarial label noise," in *Proceedings of the Asian Conference on Machine Learning* (C.-N. Hsu and W. S. Lee, eds.), vol. 20 of *Proceedings of Machine Learning Research*, (South Garden Hotels and Resorts, Taoyuan, Taiwain), pp. 97–112, PMLR, 14–15 Nov 2011.

APPENDIX I
DEFINITIONS AND THEOREMS

Definition 1.1: Empirical Rademacher complexity Let $\mathcal{H}$ be a family of functions mapping from Z to $[a,b]$ and $S = (s_1, s_2, \ldots, s_m)$ a fixed sample of size m with elements in Z. Then, the empirical Rademacher complexity of $\mathcal{H}$ with respect to the sample S is defined as,

$$\widehat{\mathfrak{R}}_S(\mathcal{H}) = \mathbb{E}_{\mathbb{B}}\big[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i h(s_i)\big],$$

where $\sigma_i, i = 1,\ldots,m$ are iid random variables following a symmetric Bernoulli (also called Rademacher) distribution, that is, σ takes values in $\{-1,1\}$ with probability $\frac{1}{2}$,

$$\mathbb{B}\{\sigma = 1\} = \mathbb{B}\{\sigma = -1\} = \frac{1}{2}.$$

Definition 1.2: Rademacher complexity Let $\mathbb{P}$ denote the distribution according to which samples are drawn. For any integer $m \geq 1$, the Rademacher complexity of $\mathcal{H}$ is the expectation of the empirical Rademacher complexity over all samples of size m according to $\mathbb{P}$,

$$\mathfrak{R}_m(\mathcal{H}) = \mathbb{E}_{\mathbb{P}^m}\big[\widehat{\mathfrak{R}}_S(\mathcal{H})\big].$$

Definition 1.3: Agnostic PAC-learning [33] A hypothesis class $\mathcal{H}$ is called agnostic PAC-learnable if there exists an algorithm M, that returns a hypothesis h_S given the training sample S and a polynomial function $p(.,.,.,.)$ such that for any $\epsilon, \delta > 0$, *for all distributions* $\mathcal{P}$ over $Z = X \times Y$, the following holds for any sample size $m \geq p(1/\epsilon, 1/\delta, dim(X))$:

$$\mathcal{P}^m\Big\{\mathbb{E}_{\mathcal{P}}\big[\ell(h_S(\chi),\upsilon)\big] - \min_{h\in\mathcal{H}} \mathbb{E}_{\mathcal{P}}\big[\ell(h(\chi),\upsilon)\big] \leq \epsilon\Big\} \geq 1-\delta.$$

Theorem 1.1: (Theorem 3.3 at [33]) Let $\mathcal{G}$ be a family of functions mapping from $X \to [0,1]$. Then, for any $\delta > 0$, with probability at least $1-\delta$ over the draw of any iid sample S of size m, each of the following holds for all $g \in \mathcal{G}$,

$$\mathbb{E}\big[g(x)\big] \leq \widehat{\mathbb{E}}_S\big[g(s_i)\big] + 2\mathfrak{R}_m(\mathcal{G}) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}$$

$$\mathbb{E}\big[g(x)\big] \leq \widehat{\mathbb{E}}_S\big[g(s_i)\big] + 2\widehat{\mathfrak{R}}_S(\mathcal{G}) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}},$$

with $\widehat{\mathbb{E}}_S\big[g(s_i)\big] = \frac{1}{m}\sum_{i=1}^{m} g(s_i)$.

Theorem 1.2: (Theorem 3.5 at [33]) Let $\mathcal{H}$ be a family of functions taking values in $\{-1,1\}$ and let $\mathcal{D}$ be the distribution over the input space X. Then, for any $\delta > 0$, with probability at least $1-\delta$ over a sample S of size m drawn according to $\mathcal{D}$, each of the following holds for any $h \in \mathcal{H}$:

$$R(h) \leq \hat{R}(h) + \mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}$$

$$R(h) \leq \hat{R}(h) + \widehat{\mathfrak{R}}_S(\mathcal{H}) + 3\sqrt{\frac{\log\frac{1}{\delta}}{2m}}.$$

Theorem 1.3: (Theorem 9.3 at [32]) The VC dimension of the class of non-homogeneous half-spaces in $\mathbb{R}^N$ is $N+1$

Theorem 1.4: (Corollary 3.19 at [33]) Let $\mathcal{H}$ be a family of functions taking values in $\{-1,+1\}$ with VC-dimension d. Then, for any $\delta > 0$, with probability at least $1-\delta$, the following holds for all $h \in \mathcal{H}$:

$$\mathcal{R}\big[h\big] \leq \widehat{\mathcal{R}}_S\big[h\big] + \sqrt{\frac{2d\log\frac{em}{d}}{m}} + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}$$

Lemma 1.1: Talagrand's lemma (Lemma 5.7 at [33]) Let $\Phi_1,\ldots,\Phi_m$ be l-Lipschitz functions from $\mathbb{R} \to \mathbb{R}$ and $\sigma_1,\ldots,\sigma_m$ be Rademacher random variables. Then, for any hypothesis set $\mathcal{H}$ of real-valued functions, the following inequality holds,

$$\frac{1}{m}\mathbb{E}_{\mathbb{Q}^m}\big[\sup_{h\in\mathcal{H}}\sum_{i=1}^{m}\sigma_i(\Phi_i\circ h)(z_i)\big] \leq$$

$$\frac{l}{m}\mathbb{E}_{\mathbb{Q}^m}\big[\sup_{h\in\mathcal{H}}\sum_{i=1}^{m}\sigma_i h(z_i)\big] = l\widehat{\mathfrak{R}}_S(\mathcal{H}).$$

In particular, if $\Phi_i = \Phi$ for all $i \in 1,\ldots,m$, then the following holds,

$$\widehat{\mathfrak{R}}_S(\Phi\circ\mathcal{H}) \leq l\widehat{\mathfrak{R}}_S(\mathcal{H}).$$

Theorem 1.5: Reproducing kernel Hilbert space (RKHS) Let $K : X \times X \to \mathbb{R}$ be a PDS kernel. Then, there exists a

Hilbert space $\mathbb{H}$ of functions f and a mapping $\psi : X \to \mathbb{H}$ such that:

$$\forall x, x' \in X,\ K(x, x') = \langle \psi(x), \psi(x') \rangle.$$

Furthermore, $\mathbb{H}$ has the following property known as the reproducing property:

$$\forall f \in \mathbb{H},\ \forall x \in X,\ f(x) = \langle f, K(x, \cdot) \rangle.$$

$\mathbb{H}$ is called the reproducing kernel Hilbert space (RKHS) associated to K.

Theorem 1.6: Representer theorem Let $K : X \times X \to \mathbb{R}$ be a PDS kernel and $\mathbb{H}$ its corresponding RKHS. Then for any non-decreasing function $G : \mathbb{R} \to \mathbb{R}$ and any loss function $L : \mathbb{R}^m \to \mathbb{R} \cup +\infty$, the optimization problem

$$\underset{f \in \mathbb{H}}{\operatorname{argmin}}\ F(h) = \underset{f \in \mathbb{H}}{\operatorname{argmin}}\ G(\|f\|_{\mathbb{H}}) + L(h(x_1), \ldots, h(x_m))$$

admits a solution of the form $f^* = \sum_{i=1}^{m} \alpha_i K(x_i, \cdot)$. If G is further assumed to be increasing, then any solution has this form.

APPENDIX II
ADVERSARIAL ATTACKS

A. Fast gradient sign method (FGSM)

FGSM is an attack for an ℓ_∞-bounded adversary [3] and computes an adversarial example as,

$$x + \xi \cdot \text{sign}(\nabla_x \ell(h_\theta(x, y)).$$

One interpretation is that this attack is a simple one-step scheme for maximizing the inner part of the adversarial problem.

B. Projected gradient descent (PGD)

A more powerfull attack [10] is a multi-step variant of the FGSM,

$$x_{t+1} = \Pi_{\mathcal{B}_\xi(x)}(x_t + \eta \cdot \text{sign}(\nabla_x \ell(h_\theta(x, y))).$$

C. Carlini and Wagner (CW)

This attack was proposed in [35] as a response to one approach to defend against adversarial attacks,

$$\begin{aligned} \min_{\xi} \quad & \|\xi\|_p + c f(x + \xi) \\ \text{s.t.} \quad & x + \xi \in [0, 1]^n. \end{aligned}$$

The $\|.\|_p$ measures the distance of the adversarial perturbation and the function f denotes a customized adversarial loss satisfying $f(x + \xi) \leq 0$. The constraint $x + \xi \in [0, 1]^n$ ensures that the image generated is a valid one.

D. DeepFool (DF)

DeepFool is based on an iterative linearization of the classifier to generate minimal perturbations that are sufficient to change classification labels [36]. Specifically, at each iteration, h is linearized around the current point x and the minimal perturbation of the linearized classifier is computed as,

$$\begin{aligned} \min_{\xi} \quad & \|\xi\|_2 \\ \text{s.t.} \quad & h(x) + \nabla_f(x)^T \xi = 0. \end{aligned}$$

APPENDIX III
ADVERSARIAL TRAINING EXAMPLES

A. Linear regression

Consider now the hypothesis class of linear functionals, $\mathcal{H} = \{h_{a,b} : x \to a^T x + b \ \big|\ a \in \mathbb{R}^d, b \in \mathbb{R}\}$. The learner wishes to find a, b that solves the following:

$$\min_{a,b \in R^{d+1}} \frac{1}{2} \mathbb{E}_{\mathbb{P}}\big[\|y - (a^T x + b)\|_2^2\big]$$

To train against adversarial attacks the learner considers the robust counterpart of the above problem. This is referred to adversarial risk minimization in the literature:

$$\begin{aligned} \min_{a,b \in R^{d+1}} \quad \max_{\delta} \quad & \frac{1}{2} \|y - (a^T(x + \delta) + b)\|_2^2 \\ \text{s.t.} \quad & \|\delta\| \leq \xi. \end{aligned}$$

1) ERM: Given samples $S = \big((x_1, y_1), \ldots, (x_m, y_m)\big)$ the learner proceeds by finding a, b that minimizes the quadratic empirical loss:

$$\min_{a,b \in R^d} \frac{1}{2m} \sum_{i=1}^{m} \|y_i - (a^T x_i + b)\|_2^2$$

One approach to solving this optimization problem is to rely on gradient descent methods. Yet, an alternative is to use OLS procedure. For this, let us fist write the problem in matrix form:

$$\min_{\theta \in R^d} \frac{1}{2m} \|Y - X\theta\|_2^2$$

with $X = [[x_1^T, 1], \ldots, [x_m^T, 1]]^T$, $Y = [y_1, \ldots, y_m]^T$ and and $\theta = [a^T, b]^T$.

Through OLS the optimizal solution is:

$$\theta^* = \begin{cases} (X^T X)^{-1} X^T Y \text{ , if } X^T X \text{ is invertible} \\ (X^T X)^{\dagger} X^T Y \text{ , otherwise} \end{cases}$$

2) AERM:

$$\min_{a,b \in R^{d+1}} \frac{1}{m} \sum_{i=1}^{m} \max_{\delta \in \mathbb{R}^d} \|y_i - (a^T(x_i + \delta) + b)\|_2^2 \quad \text{s.t.} \quad \|\delta\| \leq \xi.$$

The optimal δ^* that solves the inner maximization problem is the solution of either,

$$\begin{aligned} \max_{\delta \in \mathbb{R}^d} \quad & \|y_i - (a^T(x_i + \delta) + b)\| \\ \text{s.t.} \quad & \|\delta\| \leq \xi, \end{aligned}$$

or,

$$\begin{aligned} \min_{\delta \in \mathbb{R}^d} \quad & \|y_i - (a^T(x_i + \delta) + b)\| \\ \text{s.t.} \quad & \|\delta\| \leq \xi, \end{aligned}$$

and solving this is equivalent to solving the following problems:

$$y_i - (a^T x_i + b) + \max_{\delta \in \mathbb{R}^d} \quad - a^T \delta$$

$$\text{s.t.} \quad \|\delta\| \le \xi,$$

$$y_i - (a^T x_i + b) + \min_{\delta \in \mathbb{R}^d} \quad -a^T\delta$$
$$\text{s.t.} \quad \|\delta\| \le \xi.$$

By the dual norm definition, $\|y\|_* = \sup_x \left\{ y^T x \;\middle|\; \|x\| \le 1 \right\}$, the above can be rewritten as:

$$y_i - (a^T x_i + b) + \begin{cases} \xi\|a\|_* \\ -\xi\|a\|_* \end{cases}$$

or, in a simple form:

$$y_i - (a^T x_i + b) - \xi\|a\|_*$$

As a result, the AERM becomes:

$$\min_{a,b\in\mathbb{R}^d} \frac{1}{m}\sum_{i=1}^{m} \|y_i - (a^T x_i + b + \xi\|a\|_*)\|_2^2$$

B. Example: logistic binary classifier

In this case $y \in \{0, 1\}$. We will be working wiht the class of logistic regression models $\mathcal{H} = \{h_{a,b} : x \to 1/(1 + e^{-(a^T x+b)}) \mid a \in \mathbb{R}^d, b \in \mathbb{R}\}$.

1) ERM: In this case the learner wishes to find a, b that minimizes the cross entropy,

$$\min_{a,\, b \,\in\, R^{d+1}} \quad \mathbb{E}_{\mathbb{P}}\big[y f_{a,b}(x) + (1-y) g_{a,b}(x)\big],$$

where,

$$f_{a,b}(x) = \log(1 + e^{-(a^T x+b)}),$$

$$g_{a,b}(x) = \log(1 + e^{(a^T x+b)}).$$

.

The robust counterpart is,

$$\min_{a,b\in\mathbb{R}^{d+1}} \quad \max_{\delta \in \mathbb{R}^d} \quad \mathbb{E}_{\mathbb{P}}\big[y f_{a,b}(x+\delta) + (1-y) g_{a,b}(x+\delta)\big]$$
$$\text{s.t.} \quad \|\delta\| \le \xi.$$

2) AERM:

$$\min_{a,b\in\mathbb{R}^{d+1}} \frac{1}{m}\sum_{i=1}^{m} \max_{\delta \in \mathbb{R}^d} \quad y_i f_{a,b}(x_i+\delta) + (1-y_i) g_{a,b}(x_i+\delta)$$
$$\text{s.t.} \quad \|\delta\| \le \xi.$$

Let us focus first on the inner maximization problem,

$$\max_{\delta \in \mathbb{R}^d} \quad y_i f_{a,b}(x_i+\delta) + (1-y_i) g_{a,b}(x_i+\delta)$$
$$\text{s.t.} \quad \|\delta\| \le \xi.$$

If $y_i = 1$, the maximization can be simplified to,

$$\max_{\delta \in \mathbb{R}^d} \quad \log(1 + e^{-(a^T(x_i+\delta)+b)})$$
$$\text{s.t.} \quad \|\delta\| \le \xi.$$

The objective function in this case is monotonically decreasing, finding δ that solves the maximization is equivalent to finding δ that solves,

$$\min_{\delta \in \mathbb{R}^d} \quad (a^T(x_i+\delta) + b)$$
$$\text{s.t.} \quad \|\delta\| \le \xi,$$

$$a^T x_i + b + \min_{\delta \in \mathbb{R}^d} \quad a^T\delta$$
$$\text{s.t.} \quad \|\delta\| \le \xi.$$

Note that, $\|y\|_* = \sup_x \left\{ y^T x \;\middle|\; \|x\| \le 1 \right\}$, by the definition of the dual norm.

$$-\xi\|a\|_* = \min_{\delta \in \mathbb{R}^d} \quad a^T\delta$$
$$\text{s.t.} \quad \|\delta\| \le \xi.$$

The maximization becomes,

$$\max_{\delta \in \mathbb{R}^d} \quad \log(1 + e^{-(a^T(x_i+\delta)+b)})$$
$$\text{s.t.} \quad \|\delta\| \le \xi$$

$$= \log(1 + e^{-(a^T x_i+b+\xi\|a\|_*)}).$$

Now, if $y_i = 0$,

$$\max_{\delta \in \mathbb{R}^d} \quad \log(1 + e^{-(a^T(x_i+\delta)+b)})$$
$$\text{s.t.} \quad \|\delta\| \le \xi.$$

In this case, the objective function is monotonically increasing, and proceeding on a similar manner, leads to solving,

$$\max_{\delta \in \mathbb{R}^d} \quad (a^T(x_i+\delta) + b)$$
$$\text{s.t.} \quad \|\delta\| \le \xi,$$

$$a^T x_i + b + \max_{\delta \in \mathbb{R}^d} \quad a^T\delta$$
$$\text{s.t.} \quad \|\delta\| \le \xi.$$

And The maximization becomes,

$$\max_{\delta \in \mathbb{R}^d} \quad \log(1 + e^{(a^T(x_i+\delta)+b)})$$
$$\text{s.t.} \quad \|\delta\| \le \xi$$

$$= \log(1 + e^{(a^T x_i+b+\xi\|a\|_*)}).$$

Finally, the simplified form of the AERM is,

$$\min_{a,\, b \,\in\, \mathbb{R}^{d+1}} \quad \frac{1}{m}\sum_{i=1}^{m} y_i \log(1 + e^{-(a^T x_i+b+\xi\|a\|_*)}) +$$
$$(1-y_i)\log(1 + e^{a^T x_i+b+\xi\|a\|_*})$$

APPENDIX IV
SVM AND MARGIN THEORY

The SVM algorithm [21] had a profound impact on machine learning theory and applications. It was firstly introduced to solve a binary classification problem. It aims at finding the linear hyperplane that maximizes the distance between the closest training samples of the two classes, reducing the generalization error. In addition to performing linear classification, SVMs can efficiently perform non-linear classification by using the so called "kernel approach", which involves mapping the inputs into a higher dimension space.
Initially designed for separable data (hard-margin SVM), it was later extended to handle non-separable classification problems (soft-margin SVM). The goal of the SVM algorithm is to find the hyperplane (parameterized through w, b) that maximizes the geometric margin (equivalent to minimizing $\|w\|_2$), determined by the Euclidean distance from any point to the hyperplane. The hard-margin case is equivalent [33] to,

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2}\|w\|_2^2 \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1, \ \forall i \in 1 \dots m \end{aligned}$$

For the non-separable case, one needs to introduce the slack variables t_i and determine a trade-off between the margin maximization and the minimization of the slack variables penalty. In this case, the SVM takes the following form,

$$\begin{aligned} \min_{w,b,t_1,\dots,t_m} \quad & \frac{1}{2}\|w\|_2^2 + \lambda \sum_{i=1}^{m} t_i^p \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - t_i, \\ & t_i \geq 0, \ \forall i \in 1 \dots m. \end{aligned}$$

When considering a binary classification problem, we have two approaches for the classifier h'. One approach is to use $h'(x) = \text{sign}(h(x))$, where $h(x) = w^T x + b$. Another approach is to consider $h'(x, y) = yh(x)$, known as confidence margin. In this case, a correct classification by h occurs when $h'(x, y) > 0$, signifying that x is classified correctly. Notably, the magnitude of $h(x)$ can be interpreted as the level of confidence in the prediction made by h.

A. Similarities to SVM

Under similar assumptions, but taking the L_2 norm instead of the L_∞ norm, that is, $\|x'\|_2 \leq u = r = 1$, $\|w\|_2 \leq v = 1$ and assuming that $\xi = r = 1$, the same sample complexity guarantees work for,

$$\begin{aligned} \min_{w,t_1,\dots,t_m} \quad & \frac{1}{m}\sum_{i=1}^{m} t_i \\ \text{s.t.} \quad & y_i w^T x_i' \geq \xi(2 - t_i), \\ & t_i \geq 0, \forall i \in 1, \dots, m, \\ & \|w\|_2 \leq 1, \end{aligned}$$

which is equivalent to,

$$\begin{aligned} \min_{w,t_1,\dots,t_m} \quad & \nu\|w\|_2 + \frac{1}{m}\sum_{i=1}^{m} t_i \\ \text{s.t.} \quad & y_i w^T x_i' \geq \xi(2 - t_i), \\ & t_i \geq 0, \forall i \in 1, \dots, m, \end{aligned}$$

with ν being a Lagrange variable. Note that this formulation is quite similar to the soft-margin version of the SVM,

$$\begin{aligned} \min_{w,b,t_1,\dots,t_m} \quad & \frac{1}{2}\|w\|_2^2 + \lambda \sum_{i=1}^{m} t_i^p \\ \text{s.t.} \quad & y_i(w^T x_i + b) \geq 1 - t_i, \\ & t_i > 0, \ \forall i \in 1 \dots m. \end{aligned}$$

That being said, in another study [37], the robustness of SVMs against adversarial data manipulation was examined. The authors considered a scenario where the adversary has control over training data and aims to tamper with the SVM learning procedure. They proposed a strategy based on kernel matrix correction to enhance the SVMs' robustness to such manipulation.

APPENDIX V
PROOFS OF MAIN RESULTS

A. Proof of Theorem 3.1

Let us start by defining $\mathcal{H}' = \{(x, y) \to yh(x), \ h \in \mathcal{H}\}$, and $\mathcal{G} = \{\phi_{2,\zeta} \circ h', \ h' \in \mathcal{H}'\}$. By Theorem 1.1, we know that

$$\mathbb{E}\big[g(x)\big] \leq \widehat{\mathbb{E}}_S\big[g(s_i)\big] + 2\mathfrak{R}_m(\mathcal{G}) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}},$$

holds with probability at least $1 - \delta$ for all $g \in \mathcal{G}$. This can be rewritten as,

$$\begin{aligned} \mathbb{E}\big[\phi_{2,\zeta}(yh(x))\big] \leq \widehat{\mathbb{E}}_S\big[\phi_{2,\zeta}(y_i h(x_i))\big] + \\ 2\mathfrak{R}_m(\phi_{2,\zeta} \circ \mathcal{H}') + \sqrt{\frac{\log\frac{1}{\delta}}{2m}} \\ \mathcal{R}_{\text{rob}}^{\zeta}\big[h\big] \leq \widehat{\mathbb{E}}_S\big[\phi_{2,\zeta}(y_i h(x_i))\big] + \\ 2\mathfrak{R}_m(\phi_{2,\zeta} \circ \mathcal{H}') + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}. \end{aligned}$$

Through Talagrand's lemma, Lemma 1.1, we can further upper bound the Rademacher complexity and achieve,

$$\mathcal{R}_{\text{rob}}^{\zeta}\big[h\big] \leq \widehat{\mathbb{E}}_S\big[\phi_{2,\zeta}(y_i h(x_i))\big] + \frac{2}{\zeta}\mathfrak{R}_m(\mathcal{H}') + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}, \quad (22)$$

with probability at least $1 - \delta$, for a single, fixed ζ specified a-priori.
Furthermore, note that the Rademacher complexity is in terms of $\mathcal{H}'$, but we are interested in expressing it in terms of $\mathcal{H}$. To this end, it follows that the Rademacher complexity of $\mathcal{H}'$ is equal to the Rademacher complexity of $\mathcal{H}$,

$$\mathfrak{R}_m(\mathcal{H}') = \mathbb{E}_{\mathbb{P}^m}\big[\mathbb{E}_{\mathbb{Q}^m}\big[\sup_{h' \in \mathcal{H}'} \frac{1}{m}\sum_{i=1}^{m} \sigma_i h'(s_i)\big]\big]$$

$$= \mathbb{E}_{\mathbb{P}^m}\big[\mathbb{E}_{\mathbb{Q}^m}\big[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i y_i h(x_i)\big]\big]$$
$$= \mathbb{E}_{\mathbb{P}^m}\big[\mathbb{E}_{\mathbb{Q}^m}\big[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m}\sigma_i h(x_i)\big]\big]$$
$$= \mathfrak{R}_m(\mathcal{H}),$$

where the third equality follows by the symmetry in σ_i and because $y_i \in \{-1, 1\}$.
As a consequence,

$$\mathcal{R}^{\zeta}_{\text{rob}}\big[h\big] \leq \widehat{\mathbb{E}}_S\big[\phi_{2,\zeta}(y_i h(x_i))\big] + \frac{2}{\zeta}\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}, \quad (23)$$

holds, for all $h \in \mathcal{H}$, with probabiliy at least $1-\delta$, or, more precisely,

$$\mathbb{P}^m\Big(\mathcal{R}^{\zeta}_{\text{rob}}\big[h\big] - \widehat{\mathbb{E}}_S\big[\phi_{2,\zeta}(y_i h(x_i))\big] \leq \epsilon + \frac{2}{\zeta}\mathfrak{R}_m(\mathcal{H})\Big) \geq 1-\delta, \quad (24)$$

which is equivalent to,

$$\mathbb{P}^m\Big(\mathcal{R}^{\zeta}_{\text{rob}}\big[h\big] - \widehat{\mathbb{E}}_S\big[\phi_{2,\zeta}(y_i h(x_i))\big] > \epsilon + \frac{2}{\zeta}\mathfrak{R}_m(\mathcal{H})\Big) \leq \delta, \quad (25)$$

Notice that this holds for all $h \in \mathcal{H}$, and a given ζ specified beforehand. In particular it holds for h that results in the supremum of the left hand side of the inequality. Given that $\epsilon = \sqrt{\frac{\log\frac{1}{\delta}}{2m}}$, the right hand side be expressed just in terms of ϵ,

$$\mathbb{P}^m\Big(\sup_{h\in\mathcal{H}} \mathcal{R}^{\zeta}_{\text{rob}}\big[h\big] - \widehat{\mathbb{E}}_S\big[\phi_{2,\zeta}(y_i h(x_i))\big] > \epsilon + \frac{2}{\zeta}\mathfrak{R}_m(\mathcal{H})\Big) \leq e^{-2m\epsilon^2}. \quad (26)$$

In this part of the proof we generalize the result by showing that the bound holds uniformly for all $\zeta \in (0, r]$, with $r > 0$, at the cost of an extra term in (23).
Consider[1] now two sequences ζ_k, ϵ_k with $\epsilon_k \in]0, 1]$. It follows that (26) holds for any fixed $k \geq 1$:

$$\mathbb{P}^m\Big(\sup_{h\in\mathcal{H}} \mathcal{R}^{\zeta_k}_{\text{rob}}\big[h\big] - \widehat{\mathbb{E}}_S\big[\phi_{2,\zeta_k}(y_i h(x_i))\big] > \epsilon_k + \frac{2}{\zeta_k}\mathfrak{R}_m(\mathcal{H})\Big) \leq e^{-2m\epsilon_k^2}.$$

However, the probability of the union (due to the supremum with respect to k) is bounded by the sum of the probabilities, resulting in:

$$\mathbb{P}^m\Big(\sup_{k\geq 1, h\in\mathcal{H}} \mathcal{R}^{\zeta_k}_{\text{rob}}\big[h\big] - \widehat{\mathbb{E}}_S\big[\phi_{2,\zeta_k}(y_i h(x_i))\big] - \epsilon_k - \frac{2}{\zeta_k}\mathfrak{R}_m(\mathcal{H}) > 0\Big) \leq \sum_{k\geq 1} e^{-2m\epsilon_k^2}. \quad (27)$$

By choosing,

$$\epsilon_k = \epsilon + \sqrt{\frac{\log k}{m}}, \quad (28)$$

[1]This part follows closely the proof of Theorem 5.9 at [33]

the sum on the left-hand side of (27) has an upper bound,

$$\sum_{k\geq 1} e^{-2m\epsilon_k^2} = \sum_{k\geq 1} e^{-2m(\epsilon+\sqrt{\frac{\log k}{m}})^2} \leq \sum_{k\geq 1} e^{-2m\epsilon^2} e^{-2\log k} = \sum_{k\geq 1}\frac{1}{k^2} e^{-2m\epsilon^2} \leq 2e^{-2m\epsilon^2}.$$

Now, for $\gamma > 1$, choose $\zeta_k = \frac{r}{\gamma^k}$, and fix $\zeta_0 = r$. Then for any $\zeta \in]0, r]$, $\exists k \geq 1$, s.t., $\zeta \in]\zeta_k, \zeta_{k-1}]$. In addition, for this given k, $\zeta \leq \zeta_{k-1} = \gamma\zeta_k$, or, put differently,

$$\zeta_k \geq \frac{\zeta}{\gamma}, \quad (29)$$

which means that $\zeta \leq \gamma\frac{r}{\gamma^k}$, and thus,

$$\sqrt{\log k} \leq \sqrt{\log\log_\gamma \frac{\gamma r}{\zeta}}. \quad (30)$$

Substituting (28), (29) and (30) into (27) results in,

$$\mathbb{P}^m\Big(\sup_{\zeta\in]0,r], h\in\mathcal{H}} \mathcal{R}^{\zeta}_{\text{rob}}\big[h\big] - \widehat{\mathbb{E}}_S\big[\phi_{2,\zeta}(y_i h(x_i))\big] > \epsilon + \frac{2\gamma}{\zeta}\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log\log_\gamma\frac{\gamma r}{\zeta}}{m}}\Big) \leq 2e^{-2m\epsilon^2}, \quad (31)$$

for any $r > 0$ and $\gamma > 1$.
This is equivalent to,

$$\mathbb{P}^m\Big(\sup_{\zeta\in]0,r], h\in\mathcal{H}} \mathcal{R}^{\zeta}_{\text{rob}}\big[h\big] - \widehat{\mathbb{E}}_S\big[\phi_{2,\zeta}(y_i h(x_i))\big] \leq \sqrt{\frac{\log\frac{2}{\delta}}{2m}} + \frac{2\gamma}{\zeta}\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log\log_\gamma\frac{\gamma r}{\zeta}}{m}}\Big) \geq 1-\delta, \quad (32)$$

which concludes the proof.

B. Proof of Lemma 3.1

$$\widehat{\mathfrak{R}}_S(\mathcal{H}) = \mathbb{E}_{\mathbb{B}}\big[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i h(s_i)\big]$$
$$= \frac{1}{m}\mathbb{E}_{\mathbb{B}}\big[\sup_{h\in\mathcal{H}}\Big\{\sum_{i=1}^{m}\sigma_i w^T x_i' \;\Big|\; \|w\| \leq v\Big\}\big]$$
$$= \frac{1}{m}\mathbb{E}_{\mathbb{B}}\big[\sup_{h\in\mathcal{H}}\Big\{w^T\sum_{i=1}^{m}\sigma_i x_i' \;\Big|\; \|w\| \leq v\Big\}\big]$$
(by the definition of the dual norm)
$$= \frac{1}{m}\mathbb{E}_{\mathbb{B}}\big[v\|\sum_{i=1}^{m}\sigma_i x_i'\|_*\big]$$
(by Jensen's inequality)
$$\leq \frac{v}{m}\sqrt{\mathbb{E}_{\mathbb{B}}\big[\|\sum_{i=1}^{m}\sigma_i x_i'\|_*^2\big]}$$

$$= \frac{v}{m}\sqrt{\mathbb{E}_{\mathbb{B}}\big[\|\sum_{i,j=1}^{m}\sigma_i\sigma_j x_i' x_j'\|_*\big]}$$

(because σ_is are iid and follow a symmetric Bernoulli distribution)

$$= \frac{v}{m}\sqrt{\mathbb{E}_{\mathbb{B}}\big[\|\sum_{i}^{m} {x_i'}^2\|_*\big]}$$

$$\leq \sqrt{\frac{v^2u^2}{m}}$$

C. Proof of Corollary 3.1

From theorem 3.1 and lemma 3.1 we know that,

$$\mathcal{R}^{\zeta}_{\text{rob}}[h] \leq \frac{1}{m}\sum_{i=1}^{m}\max\big(0, 2 - \frac{y_i w^T x_i'}{\zeta}\big) + \frac{2\gamma}{\zeta}\sqrt{\frac{v^2u^2}{m}} + \sqrt{\frac{\log\log_\gamma \frac{\gamma r}{\zeta}}{m}} + \sqrt{\frac{\log\frac{2}{\delta}}{2m}},$$

holds with probability at least $1-\delta$.

Notice that $h(x') = w^T x'$ is continuous, and also $\|h(x') - h(x_0')\| \leq \|w^T\|\|x' - x_0'\|,\ \forall x, x_0 \in X$. In particular, given $\xi > 0$, if $\|x - x_0\| \leq \xi$, then $\|h(x) - h(x_0)\| \leq v\xi$. Taking $\zeta = v\xi$, concludes the proof.

D. Proof of Corollary 3.2

We know, from (Theorem 3.1), that the following holds with probability at least $1-\delta$,

$$\mathcal{R}^{\zeta}_{\text{rob}}[h] \leq \frac{1}{m}\sum_{i=1}^{m}\max\big(0, 2 - \frac{y_i h(x_i)}{\zeta}\big) + \frac{2\gamma}{\zeta}\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log\log_\gamma \frac{\gamma r}{\zeta}}{m}} + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

The learners's goal is to solve the following minimization problem,

$$\begin{aligned}\min_{w\,\in\,\mathbb{H}} \quad & \frac{1}{m}\sum_{i=1}^{m}\max\big(0, 2 - \frac{y_i w^T\psi(x_i)}{\zeta}\big)\\ \text{s.t.} \quad & \|w\|_{\mathbb{H}} \leq v,\end{aligned}$$

but note that this problem can be written as follows by introducing a Lagrange variable λ,

$$\min_{w\,\in\,\mathbb{H}} \quad \frac{1}{m}\sum_{i=1}^{m}\max\big(0, 2 - \frac{y_i w^T\psi(x_i)}{\zeta}\big) - \lambda\|w\|_{\mathbb{H}}.$$

This form is of particular interest because, given the representer theorem (Theorem 1.6), we know that the solution of this optimization has the form $w = \sum_{i=1}^{m}\alpha_i\psi(x_i)$, and hence, instead of solving the former optimization problem, in the Hilbert space $\mathbb{H}$, we can solve the following in the original space,

$$\min_{\alpha\,\in\,X^m} \quad \left\{\frac{1}{m}\sum_{i=1}^{m}\max\big(0, 2 - \frac{y_i}{\zeta}\sum_{j=1}^{m}\alpha_j\psi(x_j)\psi(x_i)\big) - \lambda\sqrt{\sum_{i,h=1}^{m}\alpha_i\alpha_j\psi(x_i)\psi(x_j)}\right\}$$

$$\min_{\alpha\,\in\,X^m} \quad \left\{\frac{1}{m}\sum_{i=1}^{m}\max\big(0, 2 - \frac{y_i}{\zeta}\sum_{j=1}^{m}\alpha_j K(x_i, x_j)\big) - \lambda\sqrt{\sum_{i,h=1}^{m}\alpha_i\alpha_j K(x_i, x_j)}\right\}$$

$$\min_{\alpha\,\in\,X^m} \quad \left\{\frac{1}{m}\sum_{i=1}^{m}\max\big(0, 2 - \frac{y_i(\mathbf{K}\alpha)_i}{\zeta}\big) - \lambda\sqrt{\alpha^T\mathbf{K}\alpha}\right\}$$

$$\begin{aligned}\min_{\alpha\,\in\,X^m} \quad & \frac{1}{m}\sum_{i=1}^{m}\max\big(0, 2 - \frac{y_i(\mathbf{K}\alpha)_i}{\zeta}\big)\\ \text{s.t.} \quad & \alpha^T\mathbf{K}\alpha \leq v^2,\end{aligned}$$

where $\mathbf{K} = [K(x_i, x_j)]_{ij}$ is the Gramiam matrix.

We start by first bounding the Rademacher complexity of the hypothesis class $\mathcal{H}$. This proof is very similar to the proof of 3.1:

$$\widehat{\mathfrak{R}}_S(\mathcal{H}) = \mathbb{E}_{\mathbb{B}}\big[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i h(x_i)\big]$$

$$= \frac{1}{m}\mathbb{E}_{\mathbb{B}}\big[\sup_{h\in\mathcal{H}}\Big\{\sum_{i=1}^{m}\sigma_i w^T\psi(x_i)\ \Big|\ \|w\|_{\mathbb{H}} \leq v\Big\}\big]$$

$$= \frac{1}{m}\mathbb{E}_{\mathbb{B}}\big[\sup_{h\in\mathcal{H}}\Big\{w^T\sum_{i=1}^{m}\sigma_i\psi(x_i)\ \Big|\ \|w\|_{\mathbb{H}} \leq v\Big\}\big]$$

(by the definition of the dual norm)

$$= \frac{1}{m}\mathbb{E}_{\mathbb{B}}\big[v\|\sum_{i=1}^{m}\sigma_i\psi(x_i)\|_{\mathbb{H}_*}\big]$$

(Jensen's inequality)

$$\leq \frac{v}{m}\sqrt{\mathbb{E}_{\mathbb{B}}\big[\|\sum_{i=1}^{m}\sigma_i\psi(x_i)\|^2_{\mathbb{H}_*}\big]}$$

$$= \frac{v}{m}\sqrt{\mathbb{E}_{\mathbb{B}}\big[\|\sum_{i,j=1}^{m}\sigma_i\sigma_j\psi(x_i)\psi(x_j)\|_{\mathbb{H}_*}\big]}$$

(because σ is iid)

$$= \frac{v}{m}\sqrt{\mathbb{E}_{\mathbb{B}}\big[\|\sum_{i}^{m}\psi(x_i)^2\|_{\mathbb{H}_*}\big]}$$

(because σ is iid)

$$= \frac{v}{m}\sqrt{\mathbb{E}_{\mathbb{B}}\big[\|\sum_{i}^{m}K(x_i, x_i)\|_{\mathbb{H}_*}\big]}$$

$$\leq \sqrt{\frac{v^2u^2}{m}}.$$

This means, (Theorem 3.1), that the following holds with probability at least $1-\delta$,

$$\mathcal{R}^{\zeta}_{\text{rob}}\big[h\big] \leq \frac{1}{m}\sum_{i=1}^{m}\max\big(0, 2-\frac{y_i(\mathbf{K}\alpha)_i}{\zeta}\big) + \frac{2\gamma}{\zeta}\sqrt{\frac{v^2u^2}{m}} + \sqrt{\frac{\log\log_{\gamma}\frac{\gamma r}{\zeta}}{m}} + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

E. Proof of Theorem 3.2

Let us first define $\mathcal{H}' = \{(x,y) \rightarrow h(x,y) - \max_{y'\neq y} h(x,y'),\ h\in\mathcal{H}\}$, and $\mathcal{G} = \{\phi_{2,\zeta}\circ h',\ h'\in\mathcal{H}'\}$, where $\phi_{2,\zeta}$ is a surrogate loss function.
We know, that,

$$\mathcal{R}^{\zeta}_{\text{rob}}\big[h\big] \leq \widehat{\mathbb{E}}_S\big[\phi_{2,\zeta}(h(x_i,y_i) - \max_{y'\neq y_i} h(x_i,y'))\big] + \frac{2}{\zeta}\mathfrak{R}_m(\mathcal{H}') + \sqrt{\frac{\log\frac{1}{\delta}}{2m}},$$

and the $\mathfrak{R}_m(\mathcal{H}')$ is given by,

$$\begin{aligned}
\mathfrak{R}_m(\mathcal{H}') &= \mathbb{E}_{\mathbb{P}^m}\big[\mathbb{E}_{\mathbb{Q}^m}\big[\sup_{h'\in\mathcal{H}'}\frac{1}{m}\sum_{i=1}^{m}\sigma_i h'(s_i)\big]\big] \\
&= \mathbb{E}_{\mathbb{P}^m}\big[\mathbb{E}_{\mathbb{Q}^m}\big[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i h(x_i,y_i) - \max_{y'\neq y_i} h(x_i,y')\big]\big] \\
&= \mathbb{E}_{\mathbb{P}^m}\big[\mathbb{E}_{\mathbb{Q}^m}\big[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i h(x_i,y_i)\big]\big] \\
&\quad + \mathbb{E}_{\mathbb{P}^m}\big[\mathbb{E}_{\mathbb{Q}^m}\big[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}-\sigma_i\max_{y'\neq y_i} h(x_i,y')\big]\big] \\
&= \mathbb{E}_{\mathbb{P}^m}\big[\mathbb{E}_{\mathbb{Q}^m}\big[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i h(x_i,y_i)\big]\big] \\
&\quad + \mathbb{E}_{\mathbb{P}^m}\big[\mathbb{E}_{\mathbb{Q}^m}\big[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i\max_{y'\neq y_i} h(x_i,y')\big]\big] \\
&= \mathbb{E}_{\mathbb{P}^m}\big[\mathbb{E}_{\mathbb{Q}^m}\big[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i(h(x_i,y)\mathbb{1}_{y=y_i} + \max_{y} h(x_i,y)\mathbb{1}_{y\neq y_i})\big]\big] \\
&\leq \mathbb{E}_{\mathbb{P}^m}\big[\mathbb{E}_{\mathbb{Q}^m}\big[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i k|h(x_i,y_i)|\big]\big] \\
&\quad \text{(by Talagrand's lemma)} \\
&\leq k\mathbb{E}_{\mathbb{P}^m}\big[\mathbb{E}_{\mathbb{Q}^m}\big[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i h(x_i,y_i)\big]\big] \\
&= k\mathfrak{R}_m(\mathcal{H})
\end{aligned}$$

And the bound follows by performing the same steps of the proof of Theorem 3.1.

APPENDIX VI
NUMERICAL EXAMPLES

The table below summarizes the number of training and test samples for various cases, impacting the reported empirical accuracy levels in the manuscript.

Data	Training samples	Test samples
NIST 0/1	12665	2115
NIST 3/8	11982	1984
CIFAR Cat/Dog	10000	2000
CIFAR Dog/Airplane	10000	2000

The code relies heavily on PyTorch and is publicly available at:

`https://github.com/f2cf2e10/advML`

For FGSM and PGD training/attacks we used SGD as optimization procedure with batch size 100, 10 epochs and seed torch seed set to 171. For our proposed approach we use a convex optimization solver for LP.

A. MNIST/CIFAR10 attacks

In this study, we examine the impact of an FGSM adversary on clean images, using examples of the number 3 and number 8 from the NIST dataset and cats and dogs from the CIFAR10 dataset.

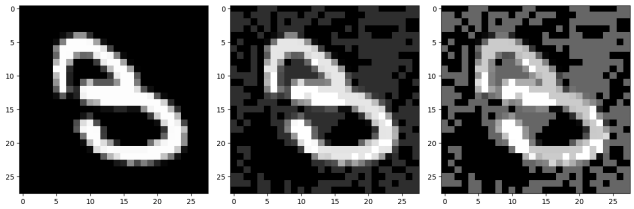

Fig. 3. NIST 3: clean (left), tampered $\xi = 0.1$ (middle) and tampered $\xi = 0.25$ (right)

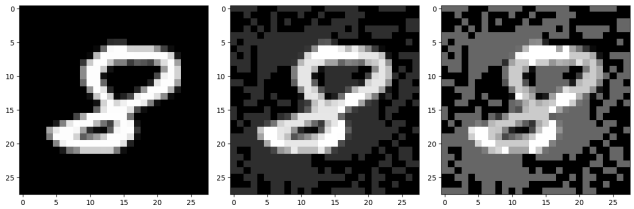

Fig. 4. NIST 8: clean (left), tampered $\xi = 0.1$ (middle) and tampered $\xi = 0.25$ (right)

We compare the original clean images with their corresponding adversarial versions at two power levels, $\xi = 0.1$ and $\xi = 0.25$. The purpose of this visual analysis is to illustrate and explore how the FGSM perturbations alter the visual appearance of the images and discuss the implications of such attacks on image recognition systems.

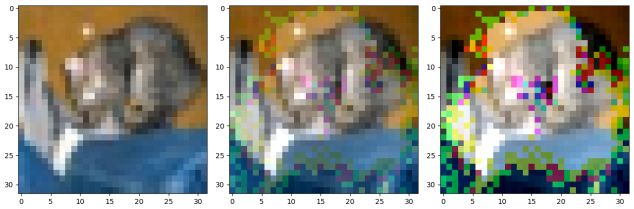

Fig. 5. CIFAR10 cat: clean (left), tampered $\xi = 0.1$ (middle) and tampered $\xi = 0.25$ (right)

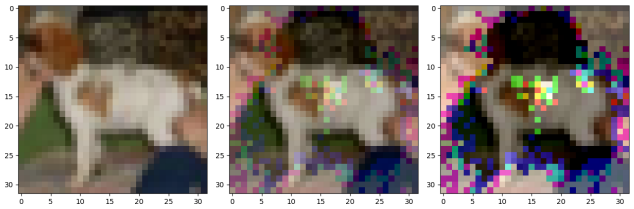

Fig. 6. CIFAR10 dog: clean (left), tampered $\xi = 0.1$ (middle) and tampered $\xi = 0.25$ (right)