A PAC-Bayes Analysis of Adversarial Robustness

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose the first general PAC-Bayesian generalization bounds for adversarial robustness, that estimate, at test time, how much a model will be invariant to imperceptible perturbations in the input. Instead of deriving a worst-case analysis of the risk of a hypothesis over all the possible perturbations, we leverage the PAC-Bayesian framework to bound the averaged risk on the perturbations for majority votes (over the whole class of hypotheses). Our theoretically founded analysis has the advantage to provide general bounds *(i)* independent from the type of perturbations (*i.e.*, the adversarial attacks), *(ii)* that are tight thanks to the PAC-Bayesian framework, *(iii)* that can be directly minimized during the learning phase to obtain a robust model on different attacks at test time.

1 Introduction

While machine learning algorithms are able to solve a huge variety of tasks, Szegedy et al. [2014] pointed out a crucial *weakness*: The possibility to generate samples similar to the originals (*i.e.*, with no or insignificant change recognizable by the human eyes) but with a different outcome from the algorithm. This phenomenon known as adversarial robustness contributes to the impossibility to ensure the safety of machine learning algorithms for safety-critical applications such as aeronautics functions (*e.g.*, vision-based navigation), autonomous driving or medical diagnosis. Adversarial robustness is thus a critical issue in machine learning that studies the ability of a model to be robust or invariant to perturbations of its input; We talk about *adversarial examples*. In other words, an adversarial example can be defined as an example that has been modified by an imperceptible noise (or that does not exceed a threshold) but which leads to a misclassification. One line of research is referred to as adversarial robustness verification [see, *e.g.*, Gehr et al., 2018, Huang et al., 2017, Singh et al., 2019, Tsuzuku et al., 2018], where the objective is to formally checking whether the neighborhood of each sample does not contain any adversarial examples. This kind of method comes with some limitations such as scalability, overapproximation, etc [Gehr et al., 2018, Katz et al., 2017, Singh et al., 2019]. In this paper we stand in another setting called adversarial attack/defense [see, *e.g.*, Papernot et al., 2016, Goodfellow et al., 2015, Madry et al., 2018, Carlini and Wagner, 2017, Zantedeschi et al., 2017, Kurakin et al., 2017]. An adversarial attack consists in finding perturbed examples that defeat machine learning algorithms while the adversarial defense techniques enhance their adversarial robustness to make the attacks useless. While a lot of methods exist, adversarial robustness suffers from a lack of general theoretical understandings (see Section 2.2).

To tackle this issue, we propose in this paper to formulate the adversarial robustness in the lens of a well-founded statistical machine learning theory called PAC-Bayes introduced by Shawe-Taylor and Williamson [1997], McAllester [1998]. This theory has the advantage to provide tight generalization bounds in average over the set of hypothesis considered (leading to bounds for a weighted majority vote over this set), in contrast to other theories such as VC-dimension or Rademacher-based approaches that give worst-case analysis, *i.e.*, for all the hypotheses. We start by defining our setting called *adversarial robust PAC-Bayes*. The idea consists in considering an

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

averaged adversarial robustness risk which corresponds to the probability that the model misclassifies a perturbed example (this can be seen as an averaged risk over the perturbations). This measure can be too optimistic and not enough informative, since for each example we sample only one perturbation. Thus we also define an *averaged-max adversarial risk* as the probability that there exists at least one perturbation (taken in a set of sampled perturbations) that leads to a misclassification. These definitions have the advantages *(i)* to be suitable to majority vote classifiers, and *(ii)* to be related to the classical adversarial robustness risk. Then, we derive a PAC-Bayesian generalization bound for each of our adversarial risks that have the advantage to be independent from the kind of attacks considered. From an algorithmic point of view, these bounds can be directly minimized in order to learn a majority vote robust in average to attacks. We empirically illustrate that our framework is able to provide generalization guarantees with non-vacuous bounds for the adversarial risk while ensuring efficient protection to adversarial attacks.

The paper is organized as follows. Section 2 recalls some basics on the classical adversarial robustness setting. We state our new adversarial robustness PAC-Bayesian setting along with our theoretical results in Section 3, and we empirically show its soundness in Section 4. Note that all the proofs of the results presented in the paper are deferred in Appendix.

2 Basics on adversarial robustness

2.1 General setting

We stand in binary classification where the input space is $X \subseteq \mathbb{R}^d$ and the output space is $Y = \{-1, +1\}$. We assume D a fixed but unknown distribution on $X \times Y$. An example is denoted by $(x, y) \in X \times Y$. Let $S = \{(x_i, y_i)\}_{i=1}^m$ be the learning sample constituted by m examples *i.i.d.* from D. We denote the distribution of such a m-sample by D^m. Let $\mathcal{H}$ be a set of real-valued functions from X to $[-1, +1]$ called voters or hypothesis. Usually, given S, a learner aims at finding the best hypothesis h from $\mathcal{H}$ that commits as few errors as possible on unseen data from D. In other words, one wants to find $h \in \mathcal{H}$ that minimizes the true risk $R_D(h)$ on D defined as

$$R_D(h) = \mathop{\mathbb{E}}_{(x,y)\sim D} \ell\left(h, (x, y)\right), \tag{1}$$

where $\ell : \mathcal{H} \times X \times Y \to \mathbb{R}^+$ is the loss function. In practice since D is unknown we cannot compute $R_D(h)$, we usually deal with the empirical risk $R_S(h)$ estimated on S and defined as $R_S(h) = \frac{1}{m} \sum_{i=1}^m \ell(h, (x_i, y_i))$. From a classic ideal machine learning standpoint, we are able to learn a well-performing classifier with strong guarantees on unseen data, and even to measure how much the model will be able to generalize on D (*e.g.*, with generalization bounds).

However in real-life applications at classification time, an imperceptible perturbation of the input (due to malicious attacks or noise for instance) can have a bad influence on the classification performance on unseen data [Szegedy et al., 2014]: the usual guarantees do not stand anymore. Such imperceptible perturbation can be modeled by a (relatively small) noise in the input. The set of possible noise is defined by $B = \{\epsilon \in X \mid \|\epsilon\| \leq b\}$, where $b > 0$ and $\|\cdot\|$ is an arbitrary norm (the most used norms are the ℓ_1, ℓ_2 and ℓ_∞-norms). The learner objective is then to find an *adversarial robust* classifier that is in average robust to all noises in B over $(x, y) \sim D$. More formally, one wants to minimize the adversarial robust true risk $R_D^{\texttt{ROB}}(h)$ defined as

$$R_D^{\texttt{ROB}}(h) = \mathop{\mathbb{E}}_{(x,y)\sim D} \max_{\epsilon \in B} \ell\left(h, (x+\epsilon, y)\right) \tag{2}$$

Similarly as in the classic setting, since D is unknown, $R_D^{\texttt{ROB}}(h)$ cannot be directly computed, and then one usually deals with the empirical adversarial risk $R_S^{\texttt{ROB}}(h) = \frac{1}{m} \sum_{i=1}^m \max_{\epsilon \in B} \ell(h, (x_i+\epsilon, y_i))$. That being said, a learned classifier h should be robust to *adversarial attacks* that aim at finding an *adversarial example* $x+\epsilon^*(x,y)$ to fool h for given example (x, y), where $\epsilon^*(x,y)$ is defined as

$$\epsilon^*(x,y) \in \operatorname{argmax}_{\epsilon \in B} \ \ell(h, (x_i+\epsilon, y_i)). \tag{3}$$

In consequence, *adversarial defense* mechanisms often rely on the adversarial attacks by replacing the original examples by the adversarial ones during the learning phase; this procedure is referred to as adversarial training. Even if there exists other defenses, adversarial training appears to be one of the most efficient defense mechanism [Ren et al., 2020].

2.2 Related works

Adversarial Attacks/Defenses.[1] Numerous methods exist to solve–or approximate–the optimization of Equation (3). Among them, Fast Gradient Sign Method (FGSM [Goodfellow et al. [2015]]) is an attack consisting in generating a noise ϵ in the direction of the gradient of the loss function with respect to the input x. [Kurakin et al. [2017]] introduced IFGSM, an iterative version of FGSM: At each iteration, one repeats FGSM and adds to x a noise, that is the sign of the gradient of the loss with respect to x. Following the same principle as IFGSM, [Madry et al. [2018]] proposed a method based on Projected Gradient Descent (PGD) that includes a random initialization of x before the optimization. Another technique known as the *Carlini and Wagner Attack* [Carlini and Wagner, 2017] aims at finding adversarial example $x{+}\epsilon^*(x,y)$ the closest as possible to the original x, *i.e.*, they want an attack being the most imperceptible as possible. However, producing such imperceptible perturbations leads to a high-running time in practice. Contrary to the most popular techniques that look for a model with low adversarial robust risk of Equation (2), our work stands in another line of research where the idea is to relax this worst-case risk measure by considering an *averaged* adversarial robust risk over the noise instead of a $\max$-based formulation [see, *e.g.*, Zantedeschi et al., 2017, Hendrycks and Dietterich, 2019]. Our averaged formulation is introduced in the next section.

Generalization Bounds. Recently, few generalization bounds for adversarial robustness have been introduced. Among them, we can mention Rademacher complexity-based bounds [Khim and Loh, 2018, Yin et al., 2019]. Khim and Loh's result is based on a surrogate of the adversarial robust true risk, and Yin et al. have obtained bounds in the specific case of neural networks and linear classifiers. For binary classification, the latter one upper-bounds the Rademacher complexity with an unavoidable polynomial dependence on the dimension of the input. Moreover, Farnia et al. [2019] present margin-based bounds on the adversarial robust risk for specific neural networks and attacks (such as FGSM or PGD). While they made use of a classical PAC-Bayes bound, their result is not a PAC-Bayesian analysis and stands in the family of uniform-convergence bounds (see [Nagarajan and Kolter, 2019, Ap.J] for details). In this paper, we provide PAC-Bayes bounds for general models expressed as majority votes, their bounds are thus not directly comparable to ours.

3 Adversarial robust PAC-Bayes

Although there exists few theoretical results, the majority of existing works comes either without theoretical guarantee or with very specific theoretical justifications. In the following, we aim at giving a different point of view on adversarial robustness based on the so-called PAC-Bayesian framework. By leveraging this framework, we derive a general generalization bound for adversarial robustness based on an averaged notion of risk that allows us to learn robust models at test time. We introduce below our new setting referred to as adversarial robust PAC-Bayes.

3.1 Adversarially robust majority vote

The PAC-Bayesian framework provides practical and theoretical tools to analyze majority vote classifiers. Assuming the voters' set $\mathcal{H}$ and a learning sample S defined as in Section 2, our goal is not anymore to learn one classifier in $\mathcal{H}$ but to learn a well-performing weighted combination of the voters involved in $\mathcal{H}$, the weights being modeled by a distribution $\mathcal{Q}$ on $\mathcal{H}$. This distribution is called the posterior distribution and is learned from S given a prior distribution $\mathcal{P}$ on $\mathcal{H}$. The learned weighted combination is called a $\mathcal{Q}$-weighted majority vote and is defined by

$$\forall x \in X, \quad H_{\mathcal{Q}}(x) = \operatorname{sign}\left[\mathop{\mathbb{E}}_{h\sim\mathcal{Q}} h(x)\right]. \tag{4}$$

In the rest of the paper, we consider the 0-1 loss function classically used for majority votes in PAC-Bayes and defined as $\ell(h,(x,y)){=}\mathbf{I}\left(h(x)\neq y\right)$ with $\mathbf{I}(a){=}1$ if a is true, and 0 otherwise. In this context, the adversarial perturbation related to Equation (3) becomes

$$\epsilon^*(x,y) \;\in\; \operatorname{argmax}_{\epsilon\in B} \mathbf{I}(H_{\mathcal{Q}}(x{+}\epsilon)\neq y). \tag{5}$$

Optimizing this problem is intractable due to the non-convexity of $H_{\mathcal{Q}}$ induced by the sign function. Therefore, all the attacks based on that definition give only an approximation of the exact solution.

[1] The reader can refer to Ren et al. [2020] for a survey on adversarial attacks and defenses.

Hence, instead of searching for the noise that maximizes the chance of fooling the algorithm, we propose to model the noise perturbation according to an example-dependent distribution. First let us define $\omega_{(x,y)}$ a distribution on B that is dependent on an example $(x,y) \in X \times Y$. Then we denote as $\mathbf{D}$ the distribution on $(X \times Y) \times B$ defined as $\mathbf{D}((x,y),\epsilon) = D(x,y) \cdot \omega_{(x,y)}(\epsilon)$ which further permits to generate *perturbed examples*. For estimating the risks on a sample, for each example (x_i, y_i) sampled from D, we consider a set of n perturbations sampled from $\omega_{(x_i,y_i)}$ denoted by $\boldsymbol{\mathcal{E}}_i = \{\epsilon_j^i\}_{j=1}^n$. Then we consider as a learning set the $m \times n$-sample $\mathbf{S} = \{((x_i,y_i),\boldsymbol{\mathcal{E}}_i)\}_{i=1}^m \in (X \times Y)^m \times B^n$. In other words, each $((x_i,y_i),\boldsymbol{\mathcal{E}}_i) \in \mathbf{S}$ is sampled from a distribution that we denote by $\mathbf{D}^n$ such that

$$\mathbf{D}^n((x_i,y_i),\boldsymbol{\mathcal{E}}_i) = D(x_i,y_i) \cdot \prod_{j=1}^{n} \omega_{(x_i,y_i)}(\epsilon_j^i).$$

Then, inspired by the works of Zantedeschi et al. [2017], Hendrycks and Dietterich [2019], we define our *robustness averaged adversarial risk* as follows.

Definition 1 (Averaged Adversarial risk)**.** *For any distribution* $\mathbf{D}$ *on* $(X \times Y) \times B$*, for any distribution* $\mathcal{Q}$ *on* $\mathcal{H}$*, the averaged adversarial risk of* $H_{\mathcal{Q}}$ *is defined as*

$$R_{\mathbf{D}}(H_{\mathcal{Q}}) \;=\; \Pr_{((x,y),\epsilon)\sim\mathbf{D}} \left(H_{\mathcal{Q}}(x+\epsilon) \neq y\right) \;=\; \mathop{\mathbb{E}}_{((x,y),\epsilon)\sim\mathbf{D}} \mathbf{I}(H_{\mathcal{Q}}(x+\epsilon) \neq y).$$

The empirical averaged adversarial risk is computed on a $m \times n$*-sample* $\mathbf{S} = \{((x_i,y_i),\boldsymbol{\mathcal{E}}_i)\}_{i=1}^m$ *is*

$$R_{\mathbf{S}}(H_{\mathcal{Q}}) = \frac{1}{nm} \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{I}(H_{\mathcal{Q}}(x_i + \epsilon_j^i) \neq y_i).$$

As we will show in Proposition 3, the risk $R_{\mathbf{D}}(H_{\mathcal{Q}})$ can be seen optimistic regarding $\epsilon^*(x,y)$ of Equation (5). Indeed, instead of taking the ϵ maximizing the loss, a unique ϵ is drawn from a distribution. Hence, it can lead to a non-informative risk regarding the occurrence of adversarial examples. To overcome this, we propose an extension that we refer as *averaged-max adversarial risk*.

Definition 2 (Averaged-Max Adversarial Risk)**.** *For any distribution* $\mathbf{D}$ *on* $(X \times Y) \times B$*, for any distribution* $\mathcal{Q}$ *on* $\mathcal{H}$*, the averaged-max adversarial risk of* $H_{\mathcal{Q}}$ *is defined as*

$$A_{\mathbf{D}^n}(H_{\mathcal{Q}}) = \Pr_{((x,y),\boldsymbol{\mathcal{E}})\sim\mathbf{D}^n} \left(\exists\, \epsilon \in \boldsymbol{\mathcal{E}}, H_{\mathcal{Q}}(x+\epsilon) \neq y\right).$$

The empirical averaged-max adversarial risk computed on a $m \times n$*-sample* $\mathbf{S} = \{((x_i,y_i),\boldsymbol{\mathcal{E}}_i)\}_{i=1}^m$ *is*

$$A_{\mathbf{S}}(H_{\mathcal{Q}}) = \frac{1}{m} \sum_{i=1}^{m} \max_{\epsilon \in \boldsymbol{\mathcal{E}}_i} \mathbf{I}(H_{\mathcal{Q}}(x_i+\epsilon) \neq y_i).$$

For an example $(x,y) \sim D$, instead of checking if one perturbed example $x+\epsilon$ is adversarial, we sample n perturbed examples $x+\epsilon_1, \ldots, x+\epsilon_n$ and we check if at least one example is adversarial.

3.2 Relations between the adversarial risks

Proposition 3 below shows the intrinsic relationships between the classical adversarial risk $R_D^{\texttt{ROB}}(H_{\mathcal{Q}})$ and our two relaxations $R_{\mathbf{D}}(H_{\mathcal{Q}})$ and $A_{\mathbf{D}^n}(H_{\mathcal{Q}})$. In particular, the result below shows that the larger n the number of perturbed examples, the higher is the chance to get an adversarial example and then to be close to the adversarial risk $R_D^{\texttt{ROB}}(H_{\mathcal{Q}})$.

Proposition 3. *For any distribution* $\mathbf{D}$ *on* $(X \times Y) \times B$*, for any distribution* $\mathcal{Q}$ *on* $\mathcal{H}$*, for any* $n, n' \in \mathbb{N}$*, with* $n \geq n' \geq 1$*, we have*

$$R_{\mathbf{D}}(H_{\mathcal{Q}}) \leq A_{\mathbf{D}^{n'}}(H_{\mathcal{Q}}) \leq A_{\mathbf{D}^n}(H_{\mathcal{Q}}) \leq R_D^{\texttt{ROB}}(H_{\mathcal{Q}}). \tag{6}$$

The left-hand side of Equation (6) confirms that the averaged adversarial risk $R_{\mathbf{D}}(H_{\mathcal{Q}})$ is optimistic regarding the classical $R_D^{\texttt{ROB}}(H_{\mathcal{Q}})$. Proposition 4 estimates how close $R_{\mathbf{D}}(H_{\mathcal{Q}})$ can be to $R_D^{\texttt{ROB}}(H_{\mathcal{Q}})$.

Proposition 4. *For any distribution* $\mathbf{D}$ *on* $(X \times Y) \times B$*, for any distribution* $\mathcal{Q}$ *on* $\mathcal{H}$*, we have*

$$R_D^{\texttt{ROB}}(H_{\mathcal{Q}}) - \mathrm{TV}(\Pi \| \Delta) \leq R_{\mathbf{D}}(H_{\mathcal{Q}}).$$

where Δ and Π are distributions on $X\times Y$; and $\Delta(x',y')$, respectively $\Pi(x',y')$, corresponds to the probability of drawing a perturbed example $(x+\epsilon)$ with $((x,y),\epsilon)\sim\mathbf{D}$, respectively an adversarial example $(x+\epsilon^*(x,y),y)$ with $(x,y)\sim D$, we have

$$\Delta(x',y')=\Pr_{((x,y),\epsilon)\sim\mathbf{D}}\left[x+\epsilon=x',y=y'\right],\quad \textit{and}\quad \Pi(x',y')=\Pr_{(x,y)\sim D}\left[x+\epsilon^*(x,y)=x',y=y'\right],\tag{7}$$

and $\mathrm{TV}(\Pi\|\Delta)=\mathop{\mathbb{E}}_{(x',y')\sim\Delta}\frac{1}{2}\left|\frac{\Pi(x',y')}{\Delta(x',y')}-1\right|$, *is the Total Variation (TV) distance between Π and Δ.*

From Equation (7), it is important to notice that $R_D^{\mathtt{ROB}}(H_\mathcal{Q})$ and $R_\mathbf{D}(H_\mathcal{Q})$ can be rewritten (see Lemmas 12 and 13 in Appendix A) respectively with Δ and Π as

$$R_\mathbf{D}(H_\mathcal{Q})=\Pr_{(x',y')\sim\Delta}\left[H_\mathcal{Q}(x')\neq y'\right],\quad\text{and}\quad R_D^{\mathtt{ROB}}(H_\mathcal{Q})=\Pr_{(x',y')\sim\Pi}\left[H_\mathcal{Q}(x')\neq y'\right].$$

Finally, Propositions 3 and 4 relate the adversarial risk $R_\mathbf{D}(H_\mathcal{Q})$ to the "standard" adversarial risk $R_D^{\mathtt{ROB}}(H_\mathcal{Q})$. Indeed by merging the two propositions we obtain

$$R_D^{\mathtt{ROB}}(H_\mathcal{Q})-\mathrm{TV}(\Pi\|\Delta)\leq R_\mathbf{D}(H_\mathcal{Q})\leq A_{\mathbf{D}^n}(H_\mathcal{Q})\leq R_D^{\mathtt{ROB}}(H_\mathcal{Q}).\tag{8}$$

Hence, the smaller the TV distance $\mathrm{TV}(\Pi\|\Delta)$, the closer the averaged adversarial risk $R_\mathbf{D}(H_\mathcal{Q})$ is from $R_D^{\mathtt{ROB}}(H_\mathcal{Q})$ and the more probable an example $((x,y),\epsilon)$ sampled from $\mathbf{D}$ would be adversarial, *i.e.*, when our "averaged" adversarial example looks like a "specific" adversarial example. Moreover, Equation (8) justifies that the PAC-Bayesian point of view makes sense for adversarial learning with theoretical guarantees: the PAC-Bayesian guarantees we derive in the next section for our adversarial risks give also some guarantees on the "standard risk" $R_D^{\mathtt{ROB}}(H_\mathcal{Q})$.

3.3 PAC-Bayesian bounds on the adversarially robust majority vote

First of all, since $R_\mathbf{D}(H_\mathcal{Q})$ and $A_{\mathbf{D}^n}(H_\mathcal{Q})$ risks are not differentiable due to the indicator function, we propose to use a common surrogate in PAC-Bayes (known as the Gibbs risk): instead of considering the risk of the $\mathcal{Q}$-weighted majority vote, we consider the expectation over $\mathcal{Q}$ of the individual risks of the voters involved in $\mathcal{H}$. In our case, we define the surrogates with the linear loss as

$$\overline{R_\mathbf{D}}(H_\mathcal{Q})=\mathop{\mathbb{E}}_{((x,y),\epsilon)\sim\mathbf{D}}\frac{1}{2}\left[1-y\mathop{\mathbb{E}}_{h\sim\mathcal{Q}}h(x+\epsilon)\right],\text{ and }\overline{A_{\mathbf{D}^n}}(H_\mathcal{Q})=\mathop{\mathbb{E}}_{((x,y),\boldsymbol{\epsilon})\sim\mathbf{D}^n}\frac{1}{2}\left[1-\min_{\epsilon\in\boldsymbol{\epsilon}}\left(y\mathop{\mathbb{E}}_{h\sim\mathcal{Q}}h(x+\epsilon)\right)\right].$$

The next theorem relates these surrogates to our risks, implying that a generalization bound for $\overline{R_\mathbf{D}}(H_\mathcal{Q})$, *resp.* for $\overline{A_{\mathbf{D}^n}}(H_\mathcal{Q})$, leads to a generalization bound for $R_\mathbf{D}(H_\mathcal{Q})$, *resp.* $A_{\mathbf{D}^n}(H_\mathcal{Q})$.

Theorem 5. *For any distributions $\mathbf{D}$ on $(X\times Y)\times B$ and $\mathcal{Q}$ on $\mathcal{H}$, for any $n>1$, we have*

$$R_\mathbf{D}(H_\mathcal{Q})\leq 2\overline{R_\mathbf{D}}(H_\mathcal{Q}),\qquad\textit{and}\qquad A_{\mathbf{D}^n}(H_\mathcal{Q})\leq 2\overline{A_{\mathbf{D}^n}}(H_\mathcal{Q}).$$

Theorem 6 below presents our PAC-Bayesian generalization bounds for $\overline{R_\mathbf{D}}(H_\mathcal{Q})$. Before that, it is important to mention that the empirical counterpart of $\overline{R_\mathbf{D}}(H_\mathcal{Q})$ is computed on $\mathbf{S}$ in which the samples are not identically independently distributed, meaning that a "classical" proof process is not applicable. The trick here is to make use of a result of Ralaivola et al. [2010] that provides a *chromatic PAC-Bayes bound*, *i.e.*, a bound which supports non-independent data.

Theorem 6. *For any distribution $\mathbf{D}$ on $(X\times Y)\times B$, for any set of voters $\mathcal{H}$, for any prior $\mathcal{P}$ on $\mathcal{H}$, with probability at least $1-\delta$ over the random choice of $\mathbf{S}$, for all posterior $\mathcal{Q}$ on $\mathcal{H}$, we have*

$$\mathrm{kl}(\overline{R_\mathbf{S}}(H_\mathcal{Q})\|\overline{R_\mathbf{D}}(H_\mathcal{Q}))\leq\frac{1}{m}\left[\mathrm{KL}(\mathcal{Q}\|\mathcal{P})+\ln\frac{m+1}{\delta}\right],\tag{9}$$

$$\textit{and}\quad\overline{R_\mathbf{D}}(H_\mathcal{Q})\leq\overline{R_\mathbf{S}}(H_\mathcal{Q})+\sqrt{\frac{1}{2m}\left[\mathrm{KL}(\mathcal{Q}\|\mathcal{P})+\ln\frac{m+1}{\delta}\right]},\tag{10}$$

Surprisingly, this theorem states results that are classic in PAC-Bayes, especially it does not depend on the number of perturbed examples n while involving the usual trade-off between the empirical risk

$\overline{R_{\mathbf{S}}}(H_{\mathcal{Q}})$ and $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$. Note that Equation (9) takes the form of a Seeger [2002]'s bound and is tighter but less interpretable than Equation (10) which takes the form of a McAllester [1998]'s bound.

We now state a generalization bound for $\overline{A_{\mathbf{D}^n}}(H_{\mathcal{Q}})$. Since this value involves a minimum term, we cannot use the same trick as for Theorem 6. To bypass this issue, we use the TV distance between two "artifical" distributions on $\boldsymbol{\mathcal{E}}_i$. Given $((x_i, y_i), \boldsymbol{\mathcal{E}}_i) \in \mathbf{S}$, let π_i be an arbitrary distribution on $\boldsymbol{\mathcal{E}}_i$, and given $h \in \mathcal{H}$, let ρ_i^h be a Dirac distribution on $\boldsymbol{\mathcal{E}}_i$ such that $\rho_i^h(\epsilon)=1$ if $\epsilon = \mathrm{argmax}_{\epsilon\in\boldsymbol{\mathcal{E}}_i} \frac{1}{2}\big[1-y_i h(x_i+\epsilon)\big]$ (*i.e.*, if ϵ is the perturbation maximizing the linear loss), and 0 otherwise.

Theorem 7. *For any distribution* $\mathbf{D}$ *on* $(X\times Y)\times B$*, for any set of voters* $\mathcal{H}$*, for any prior* $\mathcal{P}$ *on* $\mathcal{H}$*, for any* n*, with probability at least* $1-\delta$ *over the random choice of* $\mathbf{S}$*, for all posterior* $\mathcal{Q}$ *on* $\mathcal{H}$*, for all* $i \in \{1,\ldots,m\}$*, for all distribution* π_i *on* $\boldsymbol{\mathcal{E}}_i$ *independent from a voter* $h \in \mathcal{H}$*, we have*

$$\overline{A_{\mathbf{D}^n}}(H_{\mathcal{Q}}) \le \overline{A_{\mathbf{S}}}(H_{\mathcal{Q}}) + \frac{1}{m}\sum_{i=1}^{m} \mathop{\mathbb{E}}_{h\sim\mathcal{Q}} \mathrm{TV}(\rho_i^h\|\pi_i) + \sqrt{\frac{1}{2m}\left[\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) + \ln\tfrac{2\sqrt{m}}{\delta}\right]}, \tag{11}$$

where $\overline{A_{\mathbf{S}}}(H_{\mathcal{Q}}) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\Big[1-\min_{\epsilon\in\boldsymbol{\mathcal{E}}_i}\Big(y_i \mathop{\mathbb{E}}_{h\sim\mathcal{Q}} h(x_i+\epsilon)\Big)\Big]$*, and* $\mathrm{TV}(\rho\|\pi) = \mathop{\mathbb{E}}_{\epsilon\sim\pi}\frac{1}{2}\left|\left[\frac{\rho(\epsilon)}{\pi(\epsilon)}\right]-1\right|$.

Unusually, this bound involves an additional term $\frac{1}{m}\sum_{i=1}^{m}\mathbb{E}_{h\sim\mathcal{Q}}\mathrm{TV}(\rho_i^h\|\pi_i)$. From an algorithmic view, an interesting behavior is that the bound stands for all distributions π_i on $\boldsymbol{\mathcal{E}}_i$. This suggests that given (x_i, y_i), we want to find π_i minimizing $\mathbb{E}_{h\sim\mathcal{Q}}\mathrm{TV}(\rho_i^h\|\pi_i)$. Ideally, this term tends to 0 when π_i is close[2] to ρ_i^h and all voters have their loss maximized by the same perturbation $\epsilon \in \boldsymbol{\mathcal{E}}_i$.

To learn a well-performing majority vote, one solution is to minimize the right-hand side of the bounds, meaning that we would like to find a good trade-off between a low empirical risk $\overline{R_{\mathbf{S}}}(H_{\mathcal{Q}})$ or $\overline{A_{\mathbf{S}}}(H_{\mathcal{Q}})$ and a low divergence between the prior weights and the learned posterior ones $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$.

4 Experimental evaluation on differentiable decision trees

In this section, we illustrate the soundness of our framework in the context of (differentiable) decision trees learning. First of all, we describe our learning procedure designed from our theoretical results.

4.1 From the bounds to an algorithm

We consider a finite hypothesis set $\mathcal{H}$ constituted of (differentiable) decision trees [Kontschieder et al., 2016] where each $h \in \mathcal{H}$ are parametrized by a weight vector $\boldsymbol{w}^h$. Inspired by Masegosa et al. [2020], we learn the decision trees of $\mathcal{H}$ and a data-dependent prior distribution $\mathcal{P}$ from a first learning set $\mathcal{S}'$; this is a common approach in PAC-Bayes [Parrado-Hernández et al., 2012, Lever et al., 2013, Dziugaite and Roy, 2018, Dziugaite et al., 2021]. Then, the posterior distribution is learned on the second learning set $\mathcal{S}$ by minimizing the bounds. This means that we need to minimize the risk and the KL divergence term $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$. Our two-step learning precedure is summarized in Algorithm 1.

Step 1. Starting from an initial prior $\mathcal{P}_0$ and an initial set of voters $\mathcal{H}_0$ where each voter h is parametrized by a weight vector $\boldsymbol{w}_0^h$, the objective of this step is to construct the hypothesis set $\mathcal{H}$ and the prior distribution $\mathcal{P}$ to give as input to Step 2 for minimizing the bound. To do so, at each epoch t of the Step 1, we learn from $\mathcal{S}'$ an "intermediate" prior $\mathcal{P}_t$ on an "intermediate" hypothesis set $\mathcal{H}_t$ constitued of voters h parametrized by the weights $\boldsymbol{w}_t^h$; Note that the optimization in Line 9 is done with respect to $\boldsymbol{w}_t = \{\boldsymbol{w}_t^h\}_{h\in\mathcal{H}_t}$. At each iteration of the optimizer, from Lines 4 to 7 for each (x, y) of the current batch $\mathbb{S}'$, we attack the majority vote $H_{\mathcal{P}_t}$ to obtain a perturbed example $x+\epsilon$, then in Lines 8 and 9 we perform a forward pass in the majority vote with the perturbed examples and update the weights $\boldsymbol{w}_t$ and the prior $\mathcal{P}_t$ according to the linear loss. To sum up Lines 11 to 20, at the end of Step 1, the prior $\mathcal{P}$ and the hypothesis set $\mathcal{H}$ constructed for Step 2 are the ones associated to the best epoch $t^* \in \{1,\ldots,T'\}$ that permits to minimize $\overline{R_{\mathcal{S}_t}}(H_{\mathcal{P}_t})$, where $\mathcal{S}_t = \{\texttt{attack}(x, y) \mid (x, y) \in \mathcal{S}\}$ is the perturbed set obtained by attacking the majority vote $H_{\mathcal{P}_t}$.

Step 2. Starting from the prior $\mathcal{P}$ on $\mathcal{H}$ and the learning set $\mathcal{S}$, we perform the same process as in

[2] Note that, since ρ_i^h is a Dirac distribution, we have $\mathbb{E}_h\,\mathrm{TV}(\rho_i^h\|\pi_i) = \frac{1}{2}\left[1-\mathbb{E}_h\,\pi_i(\epsilon_h^*) + \mathbb{E}_h\sum_{\epsilon\neq\epsilon_h^*}\pi_i(\epsilon)\right]$, with $\epsilon_h^* = \mathrm{argmax}_{\epsilon\in\boldsymbol{\mathcal{E}}_i}\frac{1}{2}\big[1-y_i h(x_i+\epsilon)\big]$.

Algorithm 1 Average Adversarial Training with Guarantee

Require: $\mathcal{S}, \mathcal{S}'$: disjoint learning sets – T, T': number of epochs – $\mathcal{P}_0$: initial prior – $\mathcal{H}_0$ (with $\boldsymbol{w}_0$): initial hypothesis set – $\texttt{attack}()$: the attack function – $B()$: the objective function associated to a bound

Step 1 – prior and voters' set construction

1: **for** t from 1 to T' **do**
2: $\quad \mathcal{P}_t \leftarrow \mathcal{P}_{t-1}$ and $\mathcal{H}_t \leftarrow \mathcal{H}_{t-1}$ ($\boldsymbol{w}_t \leftarrow \boldsymbol{w}_{t-1}$)
3: $\quad$ **for all** batches $\mathbb{S}'$ (from $\mathcal{S}'$) **do**
4: $\quad\quad$ **for all** $(x, y) \in \mathbb{S}'$ **do**
5: $\quad\quad\quad (x{+}\epsilon, y) \leftarrow \texttt{attack}(x, y)$
6: $\quad\quad\quad \mathbb{S}' \leftarrow (\mathbb{S}' \setminus \{(x, y)\}) \cup \{(x{+}\epsilon, y)\}$
7: $\quad\quad$ **end for**
8: $\quad\quad$ Update $\mathcal{P}_t$ with $\nabla_{\mathcal{P}_t} \overline{R_{\mathbb{S}'}}(H_{\mathcal{P}_t})$
9: $\quad\quad$ Update $\boldsymbol{w}_t$ with $\nabla_{\boldsymbol{w}_t} \overline{R_{\mathbb{S}'}}(H_{\mathcal{P}_t})$
10: $\quad$ **end for**
11: $\quad \mathcal{S}_t \leftarrow \emptyset$
12: $\quad$ **for all** $(x, y) \in \mathcal{S}$ **do**
13: $\quad\quad (x{+}\epsilon, y) \leftarrow \texttt{attack}(x, y)$
14: $\quad\quad \mathcal{S}_t \leftarrow \mathcal{S}_t \cup \{(x{+}\epsilon, y)\}$
15: $\quad$ **end for**
16: $\quad t^* \leftarrow \operatorname{argmin}_{t' \in \{1,\ldots,t\}} \overline{R_{\mathcal{S}_{t'}}}(H_{\mathcal{P}_{t'}})$
17: $\quad \mathcal{P} \leftarrow \mathcal{P}_{t^*}$
18: $\quad \mathcal{H} \leftarrow \mathcal{H}_{t^*}$
19: **end for**
20: **return** $(\mathcal{P}, \mathcal{H})$

Step 2 – bound minimization

21: $(\mathcal{P}, \mathcal{H}) \leftarrow$ Output of **Step 1**
22: $\mathcal{Q}_0 \leftarrow \mathcal{P}$
23: **for** t from 1 to T **do**
24: $\quad$ **for all** batches $\mathbb{S}$ (from $\mathcal{S}$) **do**
25: $\quad\quad \mathcal{Q}_t \leftarrow \mathcal{Q}_{t-1}$
26: $\quad\quad$ **for all** $(x, y) \in \mathbb{S}$ **do**
27: $\quad\quad\quad (x{+}\epsilon, y) \leftarrow \texttt{attack}(x, y)$
28: $\quad\quad\quad \mathbb{S} \leftarrow (\mathbb{S} \setminus \{(x, y)\}) \cup \{(x{+}\epsilon, y)\}$
29: $\quad\quad$ **end for**
30: $\quad\quad$ Update $\mathcal{Q}_t$ with $\nabla_{\mathcal{Q}_t} B_{\mathbb{S}}(H_{\mathcal{Q}_t})$
31: $\quad$ **end for**
32: $\quad \mathcal{S}_t \leftarrow \emptyset$
33: $\quad$ **for all** $(x, y) \in \mathcal{S}$ **do**
34: $\quad\quad (x{+}\epsilon, y) \leftarrow \texttt{attack}(x, y)$
35: $\quad\quad \mathcal{S}_t \leftarrow \mathcal{S}_t \cup \{(x{+}\epsilon, y)\}$
36: $\quad$ **end for**
37: $\quad t^* \leftarrow \operatorname{argmin}_{t' \in \{1,\ldots,t\}} B_{\mathcal{S}_{t'}}(H_{\mathcal{Q}_{t'}})$
38: $\quad \mathcal{Q} \leftarrow \mathcal{Q}_{t^*}$
39: **end for**
40: **return** $(\mathcal{Q}, \mathcal{H})$

Step 1 except that the considered objective function corresponds to the desired bound to optimize (Line 30, denoted $B()$). For the sake of readability, we deferred in Appendix the definition of $B()$ for Equations (9) and (11). Note that the "intermediate" priors do not depend on $\mathcal{S}$, since they are learned from $\mathcal{S}'$: The bounds are then valid.

4.2 Experiments

In this section we empirically illustrate that our PAC-Bayesian framework for adversarial robustness is able to provide generalization guarantees with non-vacuous bounds for the adversarial risk.

Setting. We stand in a white-box setting meaning that the attacker knows the voters set $\mathcal{H}$, the prior distribution $\mathcal{P}$ and the posterior one $\mathcal{Q}$. We empirically study 2 attacks with the ℓ_2-norm and ℓ_∞-norm: The Projected Gradient Descent (PGD, Madry et al. [2018]) and the iterative version of FGSM (IFGSM, Kurakin et al. [2017]). We fix the number of iterations at $k{=}20$ and the step size at $\frac{b}{k}$ for PGD and IFGSM (where $b{=}1$ for ℓ_2-norm and $b{=}0.1$ for ℓ_∞-norm). One specificity of our setting is that we deal with the perturbation distribution $\omega_{(x,y)}$. Thus, we propose to sample uniformly $n{=}100$ perturbations between -10^{-2} and $+10^{-2}$ to generate n examples from the attacked example: The associated methods are referred to as PGD_U and IFGSM_U. We set $n{=}1$ when these attacks are used as defense mechanism in Algorithm 1. Indeed since the adversarial training is iterative, we do not need to sample numerous perturbations for each example: We sample a new perturbation each time the example is forwarded through the decision trees. We also consider a naive defense referred to as UNIF that only adds a noise uniformly such that the ℓ_p-norm of the added noise is lower than b. We study the following scenarios of defense/attack. These scenarios correspond to all the pairs (Defense, Attack) belonging to the set $\{$—, UNIF, PGD, IFGSM$\}\times\{$—, PGD, IFGSM$\}$ for the baseline, and $\{$—, UNIF, PGD_U, $\text{IFGSM}_\text{U}\}\times\{$—, PGD_U, $\text{IFGSM}_\text{U}\}$, where "—" means that we do not defend, *i.e.*, the attack returns the original example (note that PGD_U and IFGSM_U when "Attack without U" refers to PGD and IFGSM for computing the classical adversarial risk $R^{\texttt{ROB}}()$).

Datasets and algorithm description. We perform our experiment on 6 binary classification tasks from MNIST [LeCun et al., 1998] (1vs7, 4vs9, 5vs6) and Fashion MNIST [Xiao et al., 2017] (Coat vs Shirt, Sandal vs Ankle Boot, Top vs Pullover). We decompose the learning set into two disjoint subsets $\mathcal{S}'$ of around $7,000$ examples (to learn the prior and the voters) and $\mathcal{S}$ of exactly $5,000$ examples (to learn the posterior). We keep as test set $\mathcal{T}$ the original test set that contains around

Table 1: Test risks and bounds for **MNIST 1vs7** with n=100 perturbations for all pairs (Defense,Attack) with the two voters' set $\mathcal{H}$ and $\mathcal{H}^{\text{SIGN}}$. The results in **bold** correspond to the best values between results for $\mathcal{H}$ and $\mathcal{H}^{\text{SIGN}}$. To quantify the gap between our risks and the classical definition we put in *italic* the risk of our models against the classical attacks: we replace PGD_U and IFGSM_U by PGD or IFGSM (*i.e.*, we did *not* sample from the uniform distribution).

ℓ_2-norm $b=1$		Algo 1 with Eq. (9)						Algo 1 with Eq. (11)					
		Attack without U						Attack without U					
		$R^{\text{ROB}}_{\mathcal{T}}(H_{\mathcal{Q}})$		$R_{\mathbf{T}}(H_{\mathcal{Q}})$		Th. 6		$R^{\text{ROB}}_{\mathcal{T}}(H_{\mathcal{Q}})$		$A_{\mathbf{T}}(H_{\mathcal{Q}})$		Th. 7	
Defense	Attack	$\mathcal{H}^{\text{SIGN}}$	$\mathcal{H}$	$\mathcal{H}^{\text{SIGN}}$	$\mathcal{H}$	$\mathcal{H}^{\text{SIGN}}$	$\mathcal{H}$	$\mathcal{H}^{\text{SIGN}}$	$\mathcal{H}$	$\mathcal{H}^{\text{SIGN}}$	$\mathcal{H}$	$\mathcal{H}^{\text{SIGN}}$	$\mathcal{H}$
—	—	*.005*	*.005*	.005	.005	**.017**	.019	*.005*	*.005*	.005	.005	**.099**	.100
—	PGD_U	*.315*	***.255***	**.263**	.276	.577	**.448**	*.315*	***.229***	**.325**	.326	**.801**	1.667
—	IFGSM_U	*.117*	***.110***	**.066**	.080	**.170**	.185	*.117*	***.113***	**.106**	.110	**.356**	1.431
UNIF	—	*.005*	*.005*	.005	.005	**.018**	.019	*.005*	*.005*	.005	.005	**.099**	.100
UNIF	PGD_U	***.151***	*.174*	**.151**	.158	.355	**.292**	***.151***	*.178*	.190	**.189**	**.531**	1.620
UNIF	IFGSM_U	*.071*	***.066***	**.031**	.035	**.088**	.114	*.071*	***.070***	.056	**.054**	**.248**	1.405
PGD_U	—	***.006***	*.007*	**.006**	.007	**.023**	.024	***.006***	*.007*	**.006**	.007	**.102**	.103
PGD_U	PGD_U	***.028***	*.030*	**.021**	.025	.065	**.064**	*.027*	***.025***	**.025**	.028	**.143**	1.389
PGD_U	IFGSM_U	***.020***	*.021*	**.013**	.016	**.043**	.045	*.022*	*.022*	**.018**	.019	**.125**	1.362
IFGSM_U	—	***.006***	*.007*	**.006**	.007	**.019**	.021	***.006***	*.007*	**.006**	.007	**.100**	.102
IFGSM_U	PGD_U	***.040***	*.041*	**.033**	.035	**.086**	.094	*.040*	***.037***	.040	**.038**	**.184**	1.368
IFGSM_U	IFGSM_U	*.021*	*.021*	**.013**	.014	**.039**	.049	***.021***	*.022*	**.018**	.021	**.131**	1.329

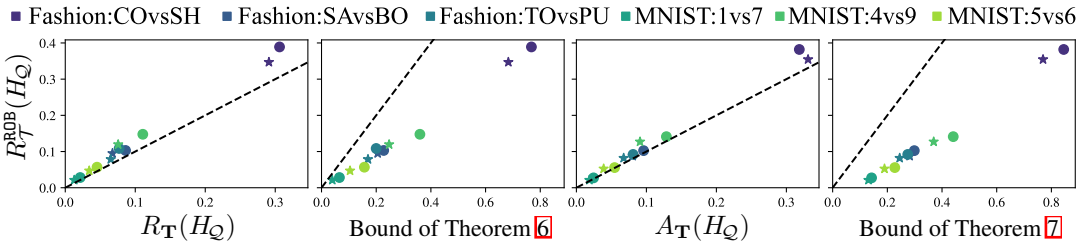

Figure 1: Visualization of the risk and bound values when Defense=Attack when the set of voters is $\mathcal{H}^{\text{SIGN}}$. Results obtained with the PGD_U, *resp.* IFGSM_U, defense are represented by a star ★, *resp.* a circle ● *(reminder:* $R^{\text{ROB}}_{\mathcal{T}}(H_{\mathcal{Q}})$ *is computed with a* PGD*, resp.* IFGSM*, attack).* The dashed line corresponds to bisecting line y=x. For $R_{\mathbf{T}}(H_{\mathcal{Q}})$ and $A_{\mathbf{T}}(H_{\mathcal{Q}})$, the closer the datasets are to the bisecting line, the more accurate our relaxed risk is compared to the classical adversarial risk $R^{\text{ROB}}_{\mathcal{T}}(H_{\mathcal{Q}})$. For the bounds, the closer the datasets are to the bisecting line, the tighter the bound.

$2,000$ examples. Moreover we need of a perturbed test set, denoted by $\mathbf{T}$, to computed our averaged(-max) adversarial risks. Depending on the scenario, $\mathbf{T}$ is constructed from $\mathcal{T}$ by attacking the prior model $H_{\mathcal{P}}$ with PGD_U or IFGSM_U with n=100 (we provide more details in Appendix). We run our Algorithm 1 for Equation (9) (Theorem 6), *resp.* Equation (11) (Theorem 7), and we compute our risk $R_{\mathbf{T}}(H_{\mathcal{Q}})$, *resp.* $A_{\mathbf{T}}(H_{\mathcal{Q}})$, the bound value and the usual adversarial risk associated to the model learned $R^{\text{ROB}}_{\mathcal{T}}(H_{\mathcal{Q}})$. For Step 1, the initial prior P_0 is fixed to the uniform distribution, the initial set of voters $\mathcal{H}_0$ is constructed with weights initialized with Xavier Initializer [Glorot and Bengio, 2010] and bias intialized at 0 (more details are given in Appendix). During Step 2, to optimize the bound, we fix the confidence parameter δ=.05, and we consider as the set of voters $\mathcal{H}$ two settings: $\mathcal{H}$ as it is output by Step 1, and the set $\mathcal{H}^{\text{SIGN}}=\{h'(\cdot)=\text{sign}(h(\cdot)) \mid h \in \mathcal{H}\}$ for which the theoretical results are still valid (we will see that in this latter situation we are able to better minimize the TV term of Theorem 7). For the two steps, we use Adam optimizer [Kingma and Ba, 2015] for T=T'=20 epochs with a learning rate at 10^{-2} and a batch size at 64.

Analysis of the results. For the sake of readibility, we exhibit the detailed results for one task (MNIST:1vs7) and all the pairs (Defense,Attack) with ℓ_2-norm in Table 1; the detailed results on the other tasks are reported in Appendix. We also provide in Figure 1 an overview of the results we obtained on all the tasks for the pairs (Defense,Attack) where Defense=Attack and with $\mathcal{H}^{\text{SIGN}}$.

First of all, note that, from Table 1 the bounds of Theorem 6 are tighter than the one of Theorem 7: This is an expected result since we showed that the averaged-Max adversarial risk $A_{\mathbf{D}^n}(H_{\mathcal{Q}})$ is more

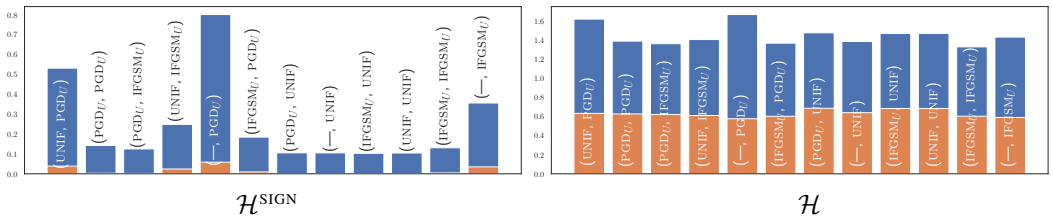

Figure 2: Visualization of the impact of the TV term in the bound of Theorem 7. The left *resp.* right bar plot show the bounds for the set of voters $\mathcal{H}^{\text{SIGN}}$, *resp.* $\mathcal{H}$. We plot the bounds for all the scenarios of Table 1 that use the TV distance, *i.e.*, all except for the pairs $(\cdot, \text{—})$. In orange we represent the value of the TV distance while in blue we represent all the remaining terms of the bound.

pessimistic than its averaged counterpart $R_{\mathbf{D}}(H_Q)$.
Moreover the bounds with $\mathcal{H}^{\text{SIGN}}$ are all informative (lower than 1) and give insightful guarantees for our models. For Theorem 7 with $\mathcal{H}$, while the risk is comparable to the risk obtained with $\mathcal{H}^{\text{SIGN}}$, the bound is greater than 1, meaning that we have no more guarantee on the model learned. As we can observed in Figure 2, this is due to the TV term involved in the bound. Considering $\mathcal{H}^{\text{SIGN}}$ when optimizing $A()$ helps to control the TV term. Even if the bounds are non-vacuous for Theorem 6 with $\mathcal{H}$, the best models with the best guarantees are obtained with $\mathcal{H}^{\text{SIGN}}$. This is also confirmed by the columns $R^{\texttt{ROB}}_{\mathcal{T}}(H_Q)$ that are always worse than $R_{\mathbf{T}}(H_Q)$ and mostly worse than $A_{\mathbf{T}}(H_Q)$ with $\mathcal{H}^{\text{SIGN}}$. The performance obtained with $\mathcal{H}^{\text{SIGN}}$ can be explained by the fact that the sign "saturates" the output of the voters which makes the majority vote more robust to noises. Thus, we focus in the rest of the analysis on results obtained with $\mathcal{H}^{\text{SIGN}}$.
Second, we observe that the naive defense UNIF is able to improve the risks $R_{\mathbf{T}}(H_Q)$ and $A_{\mathbf{T}}(H_Q)$, but the improvement with the defenses based on PGD_U and IFGSM_U is much more significant specifically against PGD_U attack (up to 13 times better). We observe the same phenomenon for both bounds (Theorems 6 and 7). This is an interesting fact because this behavior confirms that we are able to learn models that are robust against the attacks tested with theoretical guarantees.
Lastly, from Figure 1 and Table 1 it is important to notice that the gap between the classical risks and the relaxed risks is small, meaning that our relaxation is not pessimistic. Despite the pessimism of the classical risk $R^{\texttt{ROB}}_{\mathcal{T}}(H_Q)$, it remains consistent with our bounds, *i.e.*, it is lower than the bounds. In other words, in addition to give upper bounds for our risks $R_{\mathbf{T}}(H_Q)$ and $A_{\mathbf{T}}(H_Q)$, our bounds give non-vacuous guarantees on the classical risks $R^{\texttt{ROB}}_{\mathcal{T}}(H_Q)$.

5 Conclusion

Our work is the first one that studies from a general standpoint adversarial robustness through the lens of the PAC-Bayesian framework. We have started by formalizing a new adversarial robustness setting (for binary classification) specialized for models that can be expressed as a weighted majority vote; we referred to this setting as Adversarial Robust PAC-Bayes. This formulation allowed us to derive PAC-Bayesian generalization bounds on the adversarial risk of general majority votes. We illustrated the usefulness of this setting on the training of (differentiable) decision trees. Our contribution is mainly theoretical and it does not appear to directly lead to potentially negative social impact.
This work gives rise to many interesting questions and line of future research. Some perspectives will focus on extending our results to other classification settings such as multiclass or multilabel. Another line of research could focus on taking advantages of other tools of the PAC-Bayesian literature. Among them, we can make use of other bounds on the risk of the majority vote that take into consideration the diversity between the individual voters; for example, the C-bound [Lacasse et al., 2006], or more recently the tandem loss [Masegosa et al., 2020]. Last but not least, in real-life applications, one often wants to combine different input sources (from different sensors, cameras, etc). Being able to combine these sources in an effective way is then a key issue. We believe that our new adversarial robustness setting can offer theoretical guarantees and well-founded algorithms when the model we learn is expressed as a majority vote, whether for ensemble methods with weak voters (*e.g.*, Roy et al. [2011], Lorenzen et al. [2019]), or for fusion of classifiers (*e.g.*, Morvant et al. [2014]), or for multimodal/multiview learning (*e.g.*, Sun et al. [2017], Goyal et al. [2019]).

References

Nicholas Carlini and David A. Wagner. Towards Evaluating the Robustness of Neural Networks. In *IEEE SP*, 2017.

Gintare Karolina Dziugaite and Daniel M. Roy. Data-dependent PAC-Bayes priors via differential privacy. In *NeurIPS*, 2018.

Gintare Karolina Dziugaite, Kyle Hsu, Waseem Gharbieh, Gabriel Arpino, and Daniel Roy. On the role of data in PAC-Bayes. In *AISTATS*, 2021.

Farzan Farnia, Jesse M. Zhang, and David Tse. Generalizable Adversarial Training via Spectral Normalization. In *ICLR*, 2019.

Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin T. Vechev. AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation. In *IEEE SP*, 2018.

Pascal Germain, Alexandre Lacasse, François Laviolette, Mario Marchand, and Jean-Francis Roy. Risk Bounds for the Majority Vote: From a PAC-Bayesian Analysis to a Learning Algorithm. *JMLR*, 16(26):787–860, 2015.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *ICLR*, 2015.

Anil Goyal, Emilie Morvant, Pascal Germain, and Massih-Reza Amini. Multiview Boosting by Controlling the Diversity and the Accuracy of View-specific Voters. *Neurocomputing*, 2019.

Dan Hendrycks and Thomas G. Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *ICLR*, 2019.

Xiaowei Huang, Marta Kwiatkowska, Sen Wang, and Min Wu. Safety Verification of Deep Neural Networks. In *CAV*, 2017.

Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In *CAV*, volume 10426 of *Lecture Notes in Computer Science*, pages 97–117. Springer, 2017.

Justin Khim and Po-Ling Loh. Adversarial Risk Bounds for Binary Classification via Function Transformation. *CoRR*, abs/1810.09519, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

Peter Kontschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Bulò. Deep Neural Decision Forests. In *IJCAI*, 2016.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial Machine Learning at Scale. In *ICLR*, 2017.

Alexandre Lacasse, François Laviolette, Mario Marchand, Pascal Germain, and Nicolas Usunier. PAC-Bayes Bounds for the Risk of the Majority Vote and the Variance of the Gibbs Classifier. In *NIPS*, pages 769–776, 2006.

Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. THE MNIST DATASET of handwritten digits, 1998. URL `http://yann.lecun.com/exdb/mnist/`.

Guy Lever, François Laviolette, and John Shawe-Taylor. Tighter PAC-Bayes bounds through distribution-dependent priors. *Theoretical Computer Science*, 2013.

Stephan S Lorenzen, Christian Igel, and Yevgeny Seldin. On pac-bayesian bounds for random forests. *Machine Learning*, 108(8):1503–1522, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR*, 2018.

Andrés R. Masegosa, Stephan Sloth Lorenzen, Christian Igel, and Yevgeny Seldin. Second Order PAC-Bayesian Bounds for the Weighted Majority Vote. In *NeurIPS*, 2020.

David A. McAllester. Some PAC-Bayesian Theorems. In *COLT*, pages 230–234, 1998.

Emilie Morvant, Amaury Habrard, and Stéphane Ayache. Majority Vote of Diverse Classifiers for Late Fusion. In *S+SSPR*, 2014.

Vaishnavh Nagarajan and J. Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. In *NeurIPS*, 2019.

Yuki Ohnishi and Jean Honorio. Novel Change of Measure Inequalities and PAC-Bayesian Bounds. *CoRR*, 2020.

Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The Limitations of Deep Learning in Adversarial Settings. In *IEEE EuroS&P*, 2016.

Emilio Parrado-Hernández, Amiran Ambroladze, John Shawe-Taylor, and Shiliang Sun. Pac-bayes bounds with data dependent priors. *J. Mach. Learn. Res.*, 13:3507–3531, 2012. URL http://dl.acm.org/citation.cfm?id=2503353.

Liva Ralaivola, Marie Szafranski, and Guillaume Stempfel. Chromatic PAC-Bayes Bounds for Non-IID Data: Applications to Ranking and Stationary β-Mixing Processes. *JMLR*, 2010.

Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. Adversarial Attacks and Defenses in Deep Learning. *Engineering*, 6(3):346 – 360, 2020.

Jean-Francis Roy, François Laviolette, and Mario Marchand. From PAC-Bayes bounds to quadratic programs for majority votes. In Lise Getoor and Tobias Scheffer, editors, *ICML*, pages 649–656, 2011.

Edward R. Scheinerman and Daniel H. Ullman. *Fractional Graph Theory: A Rational Approach to the Theory of Graphs*. 2011.

Matthias Seeger. PAC-Bayesian generalisation error bounds for gaussian process classification. *Journal of machine learning research*, 3(Oct):233–269, 2002.

John Shawe-Taylor and Robert C. Williamson. A PAC Analysis of a Bayesian Estimator. In *COLT*, 1997.

Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin T. Vechev. Boosting Robustness Certification of Neural Networks. In *ICLR*, 2019.

Shiliang Sun, John Shawe-Taylor, and Liang Mao. Pac-bayes analysis of multi-view learning. *Information Fusion*, 35:117–131, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-Margin Training: Scalable Certification of Perturbation Invariance for Deep Neural Networks. In *NeurIPS*, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Dong Yin, Kannan Ramchandran, and Peter L. Bartlett. Rademacher Complexity for Adversarially Robust Generalization. In *ICML*, 2019.

Valentina Zantedeschi, Maria-Irina Nicolae, and Ambrish Rawat. Efficient Defenses Against Adversarial Attacks. In *ACM Workshop on Artificial Intelligence and Security, AISec@CCS*, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] In section 3.2, we show that our risks are optimistic compared to the "classical" adversarial risk. However, in Section 4, we show empirically that our risks are close to the "classical" one.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We discuss it in conclusion.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] The proofs are deferred in Appendix due to the limited number of pages and for readibility purposes.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code will be available in case of acceptance.
   (b) Did you specify all the training details (*e.g.*, data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (*e.g.*, with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (*e.g.*, type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (*e.g.*, code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]