A PAC-Bayes Analysis of Adversarial Robustness

Guillaume Vidot, Paul Viallard, Amaury Habrard, Emilie Morvant

A PAC-Bayes Analysis of Adversarial Robustness

Guillaume Vidot*,1,2, Paul Viallard*,3, Amaury Habrard[3], and Emilie Morvant[3]

[1]Airbus Opération S.A.S
[2]University of Toulouse, Institut de Recherche en Informatique de Toulouse, France
eric-guillaume.vidot@airbus.com
[3]Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d Optique Graduate School,
Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France
firstname.name@univ-st-etienne.fr
*The authors contributed equally to this work

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[1] (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[2] 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 *averaged adversarial robustness risk* which corresponds to the probability that the model misclassifies a perturbed example (*i.e.*, 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 perturba-

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

[2]Majority vote learning is rather general since a lot of machine learning model can be expressed as a majority vote.

tion. 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 averaged to attacks; in other words, the minimization of bounds ensures that attacks will be ineffective on average. We empirically illustrate this behavior.

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 it soundness in Section 4.

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 less error 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 $\|\cdot\|$ is an arbitrary norm[3] and $b > 0$. 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

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

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 mechanisms (Ren et al., 2020).

2.2 Related Works

Adversarial Attacks/Defenses. Numerous methods exist to solve—or approximate—the optimization problem of Equation (3) . Among them, we can cite the Fast Gradient Sign Method (FGSM Goodfellow et al., 2015). This attack consists 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) have introduced (IFGSM) an iterative version of FGSM: at each iteration, one repeats FGSMand adds to the input x a noise, that corresponds to the sign of the gradient of the loss with respect to x. Following the same principle as IFGSM, Madry et al. (2018) have 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) considers finding adversarial example $x+\epsilon^*(x,y)$ the closest as possible to the original input 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

[3] In the literature, the most used norms are the ℓ_1-norm, the ℓ_2-norm and the ℓ_∞-norm.

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. Note that, in the binary setting, the latter one upper-bounds the Rademacher complexity with an unavoidable polynomial dependence on the dimension of the input. Furthermore Farnia et al. (2019) present margin-based bounds on the adversarial robust true risk for specific neural networks and attacks (such as FGSM or PGD). While they made use of a classical PAC-Bayesian bound (McAllester, 2003), their result is not directly a PAC-Bayesian analysis [4] on individual classifiers while we provide PAC-Bayesian bounds for general models expressed as majority votes. It is thus important to notice that their bounds are not directly comparable to ours.

3 Adversarial Robust PAC-Bayes

Although there exists few theoretical results, the majority of existing work 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 $\mathcal{H}$. In PAC-Bayes, $\mathcal{Q}$ is called the posterior distribution and is learned from S given $\mathcal{H}$ and a prior distribution $\mathcal{P}$ on $\mathcal{H}$ (defined before the observation of S). The learned weighted combination is called $\mathcal{Q}$-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.

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*

$$\begin{aligned} R_{\mathbf{D}}(H_{\mathcal{Q}}) &= \Pr_{((x,y),\epsilon)\sim\mathbf{D}}(H_{\mathcal{Q}}(x+\epsilon)\neq y) \\ &= \mathop{\mathbb{E}}_{((x,y),\epsilon)\sim\mathbf{D}} \mathbf{I}(H_{\mathcal{Q}}(x+\epsilon)\neq y). \end{aligned} \tag{6}$$

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 as an optimistic risk regarding $\epsilon^*(x,y)$ of Equation (5).

[4] Farnia et al. (2019)'s bounds are uniform-convergence bounds, see Nagarajan and Kolter (2019, Appendix J.) for more details.

Indeed, instead of taking the ϵ that maximizes 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 drawback, we propose an extension of this risk 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 is computed on a $m\times n$*-sample* $\mathbf{S}=\{((x_i,y_i),\boldsymbol{\mathcal{E}}_i)\}_{i=1}^m$ *is*

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

Concretely, for a particular example $(x,y)\sim D$, instead of checking if one perturbed example $x+\epsilon$ is an adversarial one, we sample n perturbed examples $x+\epsilon_1,\ldots,x+\epsilon_n$ and we check if at least one example is adversarial. Actually, we show in the following that $R_{\mathbf{D}}(H_{\mathcal{Q}})$, and $A_{\mathbf{D}^n}(H_{\mathcal{Q}})$ and the classical adversarial risk $R_D^{\texttt{ROB}}(H_{\mathcal{Q}})$ are related.

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, this result 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*

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

Proof. First, we prove $A_{\mathbf{D}^1}(H_{\mathcal{Q}})=R_{\mathbf{D}}(H_{\mathcal{Q}})$. We have

$$\begin{aligned}
A_{\mathbf{D}^1}(H_{\mathcal{Q}}) &= 1-\Pr_{((x,y),\boldsymbol{\mathcal{E}})\sim\mathbf{D}^1} \left(\forall\epsilon\in\boldsymbol{\mathcal{E}}, H_{\mathcal{Q}}(x+\epsilon)=y\right)\\
&= 1-\Pr_{((x,y),\boldsymbol{\mathcal{E}})\sim\mathbf{D}^1} \left(\forall\epsilon\in\{\epsilon_1\}, H_{\mathcal{Q}}(x+\epsilon)=y\right)\\
&= 1-\Pr_{((x,y),\boldsymbol{\mathcal{E}})\sim\mathbf{D}^1} \left(H_{\mathcal{Q}}(x+\epsilon_1)=y\right) = R_{\mathbf{D}}(H_{\mathcal{Q}}).
\end{aligned}$$

Then, we prove the inequality $A_{\mathbf{D}^{n'}}(H_{\mathcal{Q}}) \leq A_{\mathbf{D}^n}(H_{\mathcal{Q}})$ from the fact that the indicator function $\mathbf{I}(\cdot)$ is upper-bounded by 1. Indeed, from Definition 2 we have

$$\begin{aligned}
1-A_{\mathbf{D}^n}(H_{\mathcal{Q}}) &= \mathop{\mathbb{E}}_{(x,y)\sim D} \mathop{\mathbb{E}}_{\boldsymbol{\mathcal{E}}\sim\omega_{(x,y)}^n} \mathbf{I}\left(\forall\epsilon\in\boldsymbol{\mathcal{E}}, H_{\mathcal{Q}}(x+\epsilon)=y\right)\\
&= \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\prod_{i=1}^{n} \mathop{\mathbb{E}}_{\epsilon_i\sim\omega_{(x,y)}} \mathbf{I}\left(H_{\mathcal{Q}}(x+\epsilon_i)=y\right)\right]\\
&\leq \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\prod_{i=1}^{n'} \mathop{\mathbb{E}}_{\epsilon_i\sim\omega_{(x,y)}} \mathbf{I}\left(H_{\mathcal{Q}}(x+\epsilon_i)=y\right)\right]\\
&= \mathop{\mathbb{E}}_{(x,y)\sim D} \mathop{\mathbb{E}}_{\boldsymbol{\mathcal{E}}'\sim\omega_{(x,y)}^{n'}} \mathbf{I}\left(\forall\epsilon\in\boldsymbol{\mathcal{E}}', H_{\mathcal{Q}}(x+\epsilon)=y\right)\\
&= 1-A_{\mathbf{D}^{n'}}(H_{\mathcal{Q}}).
\end{aligned}$$

Lastly, to prove the rightmost inequality, we have to use the fact that the expectation over the set B is bounded by the maximum over the set B. We have

$$\begin{aligned}
&A_{\mathbf{D}^n}(H_{\mathcal{Q}})\\
&= \mathop{\mathbb{E}}_{(x,y)\sim D} \mathop{\mathbb{E}}_{\epsilon_1\sim\omega_{(x,y)}} \ldots \mathop{\mathbb{E}}_{\epsilon_n\sim\omega_{(x,y)}} \mathbf{I}(\exists\epsilon\in\{\epsilon_1,\ldots,\epsilon_n\}, H_{\mathcal{Q}}(x+\epsilon)\neq y)\\
&\leq \mathop{\mathbb{E}}_{(x,y)\sim D} \max_{\epsilon_1\in B}\ldots\max_{\epsilon_n\in B} \mathbf{I}(\exists\epsilon\in\{\epsilon_1,\ldots,\epsilon_n\}, H_{\mathcal{Q}}(x+\epsilon)\neq y)\\
&= \mathop{\mathbb{E}}_{(x,y)\sim D} \max_{\epsilon_1\in B}\ldots\max_{\epsilon_{n-1}\in B} \mathbf{I}(\exists\epsilon\in\{\epsilon_1,\ldots,\epsilon^*\}, H_{\mathcal{Q}}(x+\epsilon)\neq y)\\
&= \mathop{\mathbb{E}}_{(x,y)\sim D} \mathbf{I}(H_{\mathcal{Q}}(x+\epsilon^*)\neq y)\\
&= \mathop{\mathbb{E}}_{(x,y)\sim D} \max_{\epsilon\in B} \mathbf{I}(H_{\mathcal{Q}}(x+\epsilon)\neq y) \;=\; R_D^{\texttt{ROB}}(H_{\mathcal{Q}}).
\end{aligned}$$

Merging the three equations proves the claim. □

The left-hand side of Proposition 3's result confirms that the averaged adversarial risk $R_{\mathbf{D}}(H_{\mathcal{Q}})$ is optimistic regarding the classical adversarial risk $R_D^{\texttt{ROB}}(H_{\mathcal{Q}})$. 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*

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

$$\text{and}\quad \Pi(x',y') = \Pr_{(x,y)\sim D}\left[x+\epsilon^*(x,y)=x', y=y'\right]. \tag{8}$$

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

Proof. Deferred in Appendix A. □

From Equations (7) and (8), it is important to notice that $R_D^{\texttt{ROB}}(H_\mathcal{Q})$ and $R_\mathbf{D}(H_\mathcal{Q})$ can be rewritten (see Lemma 8 and Lemma 9 in Appendix A) respectively with Δ and Π as

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

Finally, by merging Propositions 3 and 4 we obtain

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

Hence, the smaller the TV distance $\mathrm{TV}(\Pi\|\Delta)$, the closer the averaged adversarial risk $R_\mathbf{D}(H_\mathcal{Q})$ is from $R_D^{\texttt{ROB}}(H_\mathcal{Q})$ and the more probable an example $((x,y),\epsilon)$ sampled from $\mathbf{D}$ would be adversarial.

In the next section, we introduce our PAC-Bayesian generalization bounds on our two risks $R_\mathbf{D}(H_\mathcal{Q})$ and $A_{\mathbf{D}^n}(H_\mathcal{Q})$.

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 the PAC-Bayesian framework (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

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

In the next theorem, we state the relationship between these surrogates and our risks, implying that a generalization bound for $\overline{R_\mathbf{D}}(H_\mathcal{Q})$ and, *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 distribution $\mathbf{D}$ on $(X\times Y)\times B$, for any distribution $\mathcal{Q}$ on $\mathcal{H}$, for any $n>1$, we have*

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

Proof. By the definition of the majority vote, we have

$$\frac{1}{2}R_\mathbf{D}(H_\mathcal{Q}) = \frac{1}{2}\Pr_{((x,y),\epsilon)\sim\mathbf{D}}\Big(y\mathop{\mathbb{E}}_{h\sim\mathcal{Q}} h(x+\epsilon)\leq 0\Big)$$
$$= \frac{1}{2}\Pr_{((x,y),\epsilon)\sim\mathbf{D}}\Big(1-y\mathop{\mathbb{E}}_{h\sim\mathcal{Q}} h(x+\epsilon)\geq 1\Big)$$
$$\leq \mathop{\mathbb{E}}_{((x,y),\epsilon)\sim\mathbf{D}} \frac{1}{2}\Big[1-y\mathop{\mathbb{E}}_{h\sim\mathcal{Q}} h(x+\epsilon)\Big] \quad \text{(Markov's ineq. on } y\,\mathbb{E}\,h(x+\epsilon)\text{)}.$$

Similarly we have

$$\frac{1}{2}A_{\mathbf{D}^n}(H_\mathcal{Q}) = \frac{1}{2}\Pr_{((x,y),\boldsymbol{\mathcal{E}})\sim\mathbf{D}^n}\Big(\exists\epsilon\in\boldsymbol{\mathcal{E}}, y\mathop{\mathbb{E}}_{h\sim\mathcal{Q}} h(x+\epsilon)\leq 0\Big)$$
$$= \frac{1}{2}\Pr_{((x,y),\boldsymbol{\mathcal{E}})\sim\mathbf{D}^n}\Big(\min_{\epsilon\in\boldsymbol{\mathcal{E}}}\Big(y\mathop{\mathbb{E}}_{h\sim\mathcal{Q}} h(x+\epsilon)\Big)\leq 0\Big)$$
$$= \frac{1}{2}\Pr_{((x,y),\epsilon)\sim\mathbf{D}}\Big(1-\min_{\epsilon\in\boldsymbol{\mathcal{E}}}\Big(y\mathop{\mathbb{E}}_{h\sim\mathcal{Q}} h(x+\epsilon)\Big)\geq 1\Big)$$
$$\leq \mathop{\mathbb{E}}_{((x,y),\epsilon)\sim\mathbf{D}} \frac{1}{2}\Big[1-\min_{\epsilon\in\boldsymbol{\mathcal{E}}}\Big(y\mathop{\mathbb{E}}_{h\sim\mathcal{Q}} h(x+\epsilon)\Big)\Big] \quad \text{(Markov's ineq. on } \min_{\epsilon} y\mathbb{E}h(x+\epsilon)\text{)}.$$

□

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 distribution $\mathcal{P}$ on $\mathcal{H}$, with probability at least $1-\delta$ over the random choice of $\mathbf{S}$, for all posterior distribution $\mathcal{Q}$ on $\mathcal{H}$, we have*

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

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

where $\mathrm{KL}(\mathcal{Q}\|\mathcal{P}) = \mathbb{E}_{h\sim\mathcal{P}} \ln\frac{\mathcal{P}(h)}{\mathcal{Q}(h)}$ *is the KL-divergence between $\mathcal{P}$ and $\mathcal{Q}$ and* $kl(a\|b) = a\ln\frac{a}{b} + (1-a)\ln\frac{1-a}{1-b}$, *and* $\overline{R_\mathbf{S}}(H_\mathcal{Q}) = \frac{1}{mn}\sum_{i=1}^m\sum_{j=1}^n \frac{1}{2}\left[1-y_i\mathbb{E}_{h\sim\mathcal{Q}} h(x_i+\epsilon_j^i)\right]$.

Proof. Let $\Gamma=(V,E)$ be the graph representing the dependencies between the random variables where *(i)* the set of vertices is $V=\mathbf{S}$, *(ii)* the set of edges E is defined such that $(((x,y),\epsilon),((x',y'),\epsilon')) \notin E \Leftrightarrow x\neq x'$. Then, applying Th. 8 of Ralaivola et al. (2010) with our notations gives

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

where $\chi(\Gamma)$ is the fractional chromatic number of Γ. From a property of Scheinerman and Ullman (2011), we have

$$c(\Gamma) \leq \chi(\Gamma) \leq \Delta(\Gamma)+1,$$

where $c(\Gamma)$ is the order of the largest clique in Γ and $\Delta(\Gamma)$ is the maximum degree of a vertex in Γ. By construction of

Γ, $c(\Gamma)=n$ and $\Delta(\Gamma)=n-1$. Thus, $\chi(\Gamma)=n$ and rearranging the terms proves Equation (9). Finally, by applying Pinsker's inequality (*i.e.*, $|a-b|\leq\sqrt{\frac{1}{2}\mathrm{kl}(a\|b)}$), we obtain Equation (10). □

Surprisingly, this theorem states results that are classic for the PAC-Bayes literature, especially it does not depend on the number n of perturbed examples while involving the usual trade-off between the empirical counterpart $\overline{R_{\mathbf{S}}}(H_{\mathcal{Q}})$ and $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$. Note that Equation (9) is under the form of a Seeger's bound (Seeger, 2002) and is tighter but less interpretable than Equation (10) which is under the form of a McAllester's bound (McAllester, 1998).

We now state our generalization bound $\overline{A_{\mathbf{D}^n}}(H_{\mathcal{Q}})$. Since this value depends on a minimum term, we cannot use the same trick as for Theorem 6. The trick to bypass this issue is based on the use of 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 and only if $\epsilon=\operatorname{argmax}_{\epsilon\in\boldsymbol{\mathcal{E}}_i}\frac{1}{2}\big[1-y_ih(x_i+\epsilon)\big]$, *i.e.*, if ϵ is the perturbation that maximizes 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 distribution* $\mathcal{P}$ *on* $\mathcal{H}$*, for any* n*, with probability at least* $1-\delta$ *over the random choice of* $\mathbf{S}$*, for all posterior distribution* $\mathcal{Q}$ *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*

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

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 $\mathrm{TV}(\rho\|\pi)=\mathbb{E}_{\epsilon\sim\pi}\frac{1}{2}\big|\big[\rho(\epsilon)/\pi(\epsilon)\big]-1\big|$

Proof. Let $L_{h,(x,y),\epsilon}=\frac{1}{2}\big[1-yh(x+\epsilon)\big]$ for the sake of readability. The losses $\max_{\epsilon\in\boldsymbol{\mathcal{E}}_1}L_{h,(x_1,y_1),\epsilon}$, $\cdots$ $\max_{\epsilon\in\boldsymbol{\mathcal{E}}_1}L_{h,(x_m,y_m),\epsilon}$ are *i.i.d.* for any $h\in\mathcal{H}$. Hence, we can apply Theorem 20 of Germain et al. (2015) and Pinsker's inequality (*i.e.*, $|q-p|\leq\sqrt{\frac{1}{2}\mathrm{kl}(q\|p)}$) to obtain

$$\mathop{\mathbb{E}}_{h\sim\mathcal{Q}}\mathop{\mathbb{E}}_{(x,y),\boldsymbol{\mathcal{E}})\sim\mathbf{D}^n}\max_{\epsilon\in\boldsymbol{\mathcal{E}}}L_{h,(x,y),\epsilon}\leq\mathop{\mathbb{E}}_{h\sim\mathcal{Q}}\frac{1}{m}\sum_{i=1}^{m}\max_{\epsilon\in\boldsymbol{\mathcal{E}}_i}L_{h,(x_i,y_i),\epsilon}+\sqrt{\frac{\mathrm{KL}(\mathcal{Q}\|\mathcal{P})+\ln\frac{2\sqrt{m}}{\delta}}{2m}}.$$

Then, we lower-bound the left hand side of the inequality with $\overline{A_{\mathbf{D}^n}}(H_{\mathcal{Q}})$, we have

$$\overline{A_{\mathbf{D}^n}}(H_{\mathcal{Q}})\leq\mathop{\mathbb{E}}_{h\sim\mathcal{Q}}\mathop{\mathbb{E}}_{((x,y),\boldsymbol{\mathcal{E}})\sim\mathbf{D}^n}\max_{\epsilon\in\boldsymbol{\mathcal{E}}}L_{h,(x,y),\epsilon}.$$

Finally, from the definition of ρ_i^h, and from Lemma 4 of Ohnishi and Honorio (2020), we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{h\sim\mathcal{Q}}\frac{1}{m}\sum_{i=1}^{m}\max_{\epsilon\in\boldsymbol{\mathcal{E}}_i}L_{h,(x_i,y_i),\epsilon}\\
&=\mathop{\mathbb{E}}_{h\sim\mathcal{Q}}\frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{\epsilon\sim\rho_i^h}L_{h,(x_i,y_i),\epsilon}\\
&\leq\mathop{\mathbb{E}}_{h\sim\mathcal{Q}}\frac{1}{m}\sum_{i=1}^{m}\mathrm{TV}(\rho_i^h\|\pi_i)+\mathop{\mathbb{E}}_{h\sim\mathcal{Q}}\frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{\epsilon\sim\pi_i}L_{h,(x_i,y_i),\epsilon}\\
&=\mathop{\mathbb{E}}_{h\sim\mathcal{Q}}\frac{1}{m}\sum_{i=1}^{m}\mathrm{TV}(\rho_i^h\|\pi_i)+\frac{1}{m}\sum_{i=1}^{m}\mathop{\mathbb{E}}_{\epsilon\sim\pi_i}\mathop{\mathbb{E}}_{h\sim\mathcal{Q}}L_{h,(x_i,y_i),\epsilon}\\
&\leq\mathop{\mathbb{E}}_{h\sim\mathcal{Q}}\frac{1}{m}\sum_{i=1}^{m}\mathrm{TV}(\rho_i^h\|\pi_i)+\overline{A_{\mathbf{S}}}(H_{\mathcal{Q}}).
\end{aligned}$$

□

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 point of 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 that minimizes $\mathbb{E}_{h\sim\mathcal{Q}}\mathrm{TV}(\rho_i^h\|\pi_i)$. Ideally, this term tends to 0 when *(i)* the distribution π_i is close[5] to ρ_i^h, *(ii)* all voters have their loss maximized by the same perturbation $\epsilon\in\boldsymbol{\mathcal{E}}_i$.

From a practical point of view, to learn a well-performing majority vote, one solution consists in minimizing the right-hand side of the bounds, meaning that we would like to find a good trade-off between *(i)* a small empirical risk $\overline{R_{\mathbf{S}}}(H_{\mathcal{Q}})$ or $\overline{A_{\mathbf{S}}}(H_{\mathcal{Q}})$ and *(ii)* a small divergence between the prior weights and the learned posterior ones $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})$.

4 Experimental Evaluation

In this section, we illustrate the soundness of our framework in the context of neural networks learning. First of all, we describe the learning method designed according to our theoretical results and used in our experiments.

4.1 From the Bounds to an Algorithm

Let $h_{\boldsymbol{w}'}$ be a neural network parametrized by the weight vector $\boldsymbol{w}'\in\mathbb{R}^d$. We consider that the weights $\boldsymbol{w}'$ are sampled from the posterior distribution $\mathcal{Q}=\mathcal{N}(\boldsymbol{w},\lambda\boldsymbol{I}_d)$ which

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

is a Gaussian distribution centered at $\boldsymbol{w}$ with covariance matrix $\lambda \boldsymbol{I}_d$ where $\boldsymbol{I}_d$ is the identity matrix of dimension $d \times d$. Then, the majority vote is defined by

$$H_{\mathcal{Q}}(x) = \text{sign}\left[\mathop{\mathbb{E}}_{\boldsymbol{w}' \sim \mathcal{Q}} h_{\boldsymbol{w}'}(x)\right].$$

Starting from a majority vote defined *a priori* with the prior distribution $\mathcal{P}=\mathcal{N}(\boldsymbol{v}, \lambda \boldsymbol{I}_d)$, we learn the majority vote $H_{\mathcal{Q}}$ by optimizing the bounds. This means that we need to minimize the risk and the KL divergence term $\text{KL}(\mathcal{Q}\|\mathcal{P})$. To do so, we consider in our experiments a data-dependent prior; this is a common approach in PAC-Bayes (Parrado-Hernández et al., 2012; Lever et al., 2013; Dziugaite and Roy, 2018; Dziugaite et al., 2020). For this purpose, we use a two-step learning process summarized in Algorithm 1 and that takes as input two disjoint learning sets $\mathcal{S}$ and $\mathcal{S}'$.

Step *(i)* We learn the prior $\mathcal{P}$. At each epoch t of the algorithm we learn from $\mathcal{S}'$ an "intermediate" prior $\mathcal{P}_t=\mathcal{N}(\boldsymbol{v}_t, \lambda \boldsymbol{I}_d)$ (Lines 2 to 12). At the end of the process (Line 13) we set $\mathcal{P}$ as the "intermediate" prior that leads to a good empirical performance on $\mathcal{S}$ estimated batch by batch. More precisely, we keep the one that minimizes (with the linear loss) $\mathbb{E}_{\mathbb{S}}\, R_{\mathbb{S}}(h_{\boldsymbol{v}_t^{\mathbb{S}}})$, where $\mathbb{E}_{\mathbb{S}}$ is the expectation over the batches that are sampled uniformly from a partition of $\mathcal{S}$, and $\boldsymbol{v}_t^{\mathbb{S}}$ are the weights sampled from $\mathcal{N}(\boldsymbol{v}_t, \lambda \boldsymbol{I}_d)$ for the batch $\mathbb{S}$. At a given epoch t, for each iteration of the optimizer 1) we sample from $P_{t-1}=\mathcal{N}(\boldsymbol{v}_{t-1}, \lambda \boldsymbol{I}_d)$ a weight vector $\boldsymbol{w}'$, and 2) we attack our sampled model to obtain $x+\epsilon$ a perturbed example, and 3) we forward in the sampled network the perturbed example and update the weights according to the linear loss (Line 10).

Step *(ii)*. Starting from the prior $\mathcal{P}$ and the learning set $\mathcal{S}$, we perform the same process as in **Step** *(i)* except that the loss considered corresponds to the desired bound to optimize (Line 23, denoted $\ell_{\texttt{bnd}}()$). For the sake of readability, we deferred in Appendix the definition of $\ell_{\texttt{bnd}}$ for Equation (9) and Equation (11).

Algorithm 1 Average Adversarial Training with Guarantee

Require: $\mathcal{S}, \mathcal{S}'$: disjoint learning sets
T, T': number of epochs
η, η': learning rates
$\texttt{attack}()$: the attack function
$\ell_{\texttt{bnd}}()$: the loss associated to a bound to minimize
$\boldsymbol{v}_0$: initial weights

Step *(i)*

2: **for** t from 1 to T **do**
3: $\quad \boldsymbol{v}_t \longleftarrow \boldsymbol{v}_{t-1}$
4: $\quad$ **for all** batches $\mathbb{S}'$ (from $\mathcal{S}'$) **do**
5: $\quad\quad$ **for all** $(x, y) \in \mathbb{S}'$ **do**
6: $\quad\quad\quad (x+\epsilon, y) \longleftarrow \texttt{attack}(x, y)$
7: $\quad\quad\quad$ Replace (x, y) by $(x+\epsilon, y)$ in $\mathbb{S}'$
8: $\quad\quad$ **end for**
9: $\quad\quad$ Sample the weights $\boldsymbol{v}'_t \sim \mathcal{N}(\boldsymbol{v}_t, \lambda \boldsymbol{I}_d)$
10: $\quad\quad \boldsymbol{v}_t \longleftarrow \boldsymbol{v}_t + \eta \mathop{\mathbb{E}}_{(x+\epsilon,y)\sim\mathbb{S}} \nabla_{\boldsymbol{v}_t} \frac{1}{2}\left[1 - y h_{\boldsymbol{v}'_t}(x+\epsilon)\right]$
11: $\quad$ **end for**
12: **end for**
13: $\mathcal{P} \leftarrow \mathcal{N}(\boldsymbol{v}_{best}, \lambda \boldsymbol{I}_d)$
with $\boldsymbol{v}_{best} \leftarrow \text{argmin}_{\boldsymbol{v} \in \{\boldsymbol{v}_t\}_{t=1}^{T}} \mathbb{E}_{\mathbb{S}}\, R_{\mathbb{S}}(h_{\boldsymbol{v}_t^{\mathbb{S}}})$

Step *(ii)*

15: $\boldsymbol{w} \leftarrow \boldsymbol{v}_{best}$
16: **for** t from 1 to T' **do**
17: $\quad$ **for all** batches $\mathbb{S}$ (from $\mathcal{S}$) **do**
18: $\quad\quad$ **for** all $(x, y) \in \mathbb{S}$ **do**
19: $\quad\quad\quad (x+\epsilon, y) \leftarrow \texttt{attack}(x, y)$
20: $\quad\quad\quad$ Replace (x, y) by $(x+\epsilon, y)$ in $\mathbb{S}$
21: $\quad\quad$ **end for**
22: $\quad\quad$ Sample the weights $\boldsymbol{w}' \sim \mathcal{N}(\boldsymbol{w}, \lambda \boldsymbol{I}_d)$
23: $\quad\quad \boldsymbol{w} \leftarrow \boldsymbol{w} + \eta' \mathop{\mathbb{E}}_{(x+\epsilon,y)\sim\mathbb{S}} \nabla_{\boldsymbol{w}} \ell_{\texttt{bnd}}(h_{\boldsymbol{w}'}, (x+\epsilon, y), \boldsymbol{v}_{best})$
24: $\quad$ **end for**
25: **end for**
26: **return** $\mathcal{Q} \leftarrow \mathcal{N}(\boldsymbol{w}, \lambda \boldsymbol{I}_d)$

4.2 Experimental Setting

General setting. We perform our experiment on MNIST. We decompose the learning set into two disjoint subsets $\mathcal{S}'$ of around $7,000$ examples (to learn the prior) and $\mathcal{S}$ of exactly $5,000$ examples (to learn the posterior). We select[6] the best prior on $\mathcal{S}$. We keep as test set the original set denoted $\mathcal{T}$, that contains $2,000$ examples. Note that, we consider the same architecture as Madry et al. (2018) but in a binary setting. To do so, we select pairs of classes that share similarities. We report here[7] the results on one task: 1vs7. We use a neural network consisting of a convolutional part (2 layers: 32 and 64 filters with Leaky ReLU and 2×2 max-pooling) and a fully connected part (1 layer of size $1,024$). We train all the models using Adam optimizer for 100 epochs with a learning rate at $1e^{-5}$ and a batch size at 64. To optimize the bound, we fix the confidence parameter $\delta=.05$.

Defenses/Attacks Setting. We stand in a white-box setting meaning that the attacker knows at least the architecture and the parameters of the model. We empirically study two attacks with 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 100 and the step size at .008 for PGD and IFGSM. One specificity of our setting is that we

[6] The "intermediate" priors does not depend on $\mathcal{S}$, since they are learned with $\mathcal{S}'$. The bounds are then still valid.

[7] Results for some other tasks are deferred in Appendix.

TABLE 1: The table shows the different test risks and bounds for **MNIST 1vs7** with n=50 perturbations for all the pairs (Defense,Attack). In **bold** are highlighted the most significant results. Note that, the results in *italic* corresponds to the baseline on the deterministic network $h_{\boldsymbol{w}}$: importantly, for the baseline, we did *not* sampled from the uniform distribution, but we put the results in the table as a reference.

		$b=0.1$					$b=0.3$				
		baseline	Algo.1		Algo.1		baseline	Algo.1		Algo.1	
		without U	with Eq. (9)		with Eq. (11)		without U	with Eq. (9)		with Eq. (11)	
Defense	Attack	$R_{\mathbf{T}}(h_{\boldsymbol{w}})$	$R_{\mathbf{T}_\mathrm{U}}(H_{\mathcal{Q}})$	Th. 6	$A_{\mathbf{T}_\mathrm{U}}(H_{\mathcal{Q}})$	Th. 7	$R_{\mathbf{T}}(h_{\boldsymbol{w}})$	$R_{\mathbf{T}_\mathrm{U}}(H_{\mathcal{Q}})$	Th. 6	$A_{\mathbf{T}_\mathrm{U}}(H_{\mathcal{Q}})$	Th. 7
—	PGD_U	*0.1244*	**0.2723**	**0.6850**	**0.2728**	**0.7355**	*0.5206*	0.5006	1.0626	0.5409	1.1355
—	$\mathrm{IFGSM}_\mathrm{U}$	*0.1110*	**0.2682**	**0.6817**	**0.2723**	**0.7349**	*0.5261*	0.5030	1.0729	0.5488	1.1565
UNIF	PGD_U	*0.1438*	**0.1854**	**0.5793**	**0.1937**	**0.6555**	*0.5229*	0.5212	1.1186	0.5224	1.1502
UNIF	$\mathrm{IFGSM}_\mathrm{U}$	*0.1382*	**0.1854**	**0.5785**	**0.1928**	**0.6540**	*0.5229*	0.5193	1.1163	0.5261	1.1522
PGD_U	PGD_U	*0.2386*	**0.1554**	**0.2857**	**0.1572**	**0.3365**	*0.3481*	0.4370	1.0641	0.4563	1.0872
PGD_U	$\mathrm{IFGSM}_\mathrm{U}$	*0.2228*	**0.1378**	**0.2598**	**0.1368**	**0.3115**	*0.3125*	0.4282	1.0368	0.4457	1.0596
$\mathrm{IFGSM}_\mathrm{U}$	PGD_U	*0.2205*	**0.1742**	**0.3583**	**0.1761**	**0.4015**	*0.5220*	0.4831	1.0772	0.5021	1.0998
$\mathrm{IFGSM}_\mathrm{U}$	$\mathrm{IFGSM}_\mathrm{U}$	*0.1937*	**0.1571**	**0.3194**	**0.1600**	**0.3660**	*0.2899*	0.4790	1.0627	0.4938	1.0849

deal with the perturbation distribution $\omega_{(x,y)}$. In consequence, we cannot use a single attack PGD or IFGSM as usually done. We propose thus to sample uniformly n perturbations between -0.01 and $+0.01$ to generate n examples from the attacked example: the associated methods are referred as PGD_U and $\mathrm{IFGSM}_\mathrm{U}$. Note that 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 forward through the network. We also consider a naive defense referred as UNIF that only adds a noise uniformly sampled between $-b$ and $+b$ (where b is the maximal allowed norm for the perturbation). Note that we also report for the sake of completeness the result of the classic baselines PGD and IFGSM of the literature which consist of running PGD_U and $\mathrm{IFGSM}_\mathrm{U}$without adding of a uniform noise. We run the baseline and our Algorithm 1 with Equations (9) and (11) in different scenarios of defense/attack. These scenarios correspond to all the pairs $(\text{Defense}, \text{Attack})$ belonging to the set $\{\text{—}, \text{UNIF}, \mathrm{PGD}_\mathrm{U}, \mathrm{IFGSM}_\mathrm{U}\}\times\{\mathrm{PGD}_\mathrm{U}, \mathrm{IFGSM}_\mathrm{U}\}$, where "—" means that we do not defend, *i.e.*, the attack returns the original example (in the case of the baseline PGD_U and $\mathrm{IFGSM}_\mathrm{U}$ are substituted by PGD and IFGSM). We report in Table 1, the risks and the bounds values computed with the test set $\mathcal{T}$ perturbed depending on the situation: $\mathbf{T}$ (for the baseline) is perturbed with PGD or IFGSM, and $\mathbf{T}_\mathrm{U}$ (for our algorithm) is perturbed[8] with PGD_U or $\mathrm{IFGSM}_\mathrm{U}$ taking n=50 perturbations.

Analysis of the results. 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 pessimistic than its averaged counterpart $R_{\mathbf{D}}(H_{\mathcal{Q}})$.

Second, we observe that the naive defense UNIF does not improve the risks $R_{\mathbf{T}}(H_{\mathcal{Q}})$ and $A_{\mathbf{T}}(H_{\mathcal{Q}})$, while PGD_U and $\mathrm{IFGSM}_\mathrm{U}$ are able to improve them. In the case of the maximum perturbation b=0.3, the risks are high (between .4 and .5) meaning that the models commit an error almost 40% of the time. Indeed, with a ℓ_∞-based attack, each grayscale pixel could be modified up to 30% and are thus strongly degraded making the task hard. Due to these strongly perturbed instances, the bounds are unsurprisingly not informative illustrating the difficulty of the task.

When considering a reduced level of noise by setting b=0.1, the task becomes more accessible. In this situation, if we focus on the results in bold in Table 1, we observe that all defense mechanisms provide better risks than when we use no defense. The bounds here are all informative (lower than 1) and give insightful guarantees for our models. An interesting fact is that the bounds we obtain when defending with PGD_U or $\mathrm{IFGSM}_\mathrm{U}$ are tighter than the bounds we get when defending in a naive way with UNIF, showing an improvement from 25% to 34%. This behavior confirms that we are able to learn models that are robust against the attacks tested with theoretical guarantees.

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 Ad-

[8] The perturbed set $\mathbf{T}_\mathrm{U}$ is generated by sampling a network from $\mathcal{P}$ and attacking this network with PGD_U or $\mathrm{IFGSM}_\mathrm{U}$. We provide more details in Appendix.

versarial 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 neural networks.

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, and Daniel M. Roy. On the role of data in PAC-Bayes bounds. *CoRR*, 2020.

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.

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*. OpenReview.net, 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.

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.

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.

David A. McAllester. Simplified PAC-Bayesian Margin Bounds. In *COLT*, 2003.

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.

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.

A PAC-Bayes Analysis of Adversarial Robustness
Supplementary Material

The supplementary material is structured as follows. In section A we provide a proof of Proposition 4. We discuss, in Section B, the validity of the bound when we select a prior with $\mathcal{S}$ and having a distribution on perturbations depending on this selected prior. Moreover, we detail how we optimize and compute our bounds in Section C. Finally, we present additional experiments in Section D.

Note we have a typing mistake in Equation (9), $\mathrm{kl}(\overline{R_{\mathbf{D}}}(H_Q)\|\overline{R_{\mathbf{S}}}(H_Q))$ must be $\mathrm{kl}(\overline{R_{\mathbf{S}}}(H_Q)\|\overline{R_{\mathbf{D}}}(H_Q))$. In consequence, this will be modified in the paper in case of acceptance.

A Proof of Proposition 4

In this section, we provide the proof of Proposition 4 that relies on Lemmas 8 and 9 which are also described and proved.

Lemma 8 shows that $R_{\mathbf{D}}(H_Q)$ is equivalent to $R_\Delta(H_Q)$.

Lemma 8. *For any distribution* $\mathbf{D}$ *on* $(X\times Y)\times B$ *and its associated distribution* Δ*, for any posterior* Q *on* $\mathcal{H}$*, we have*

$$R_{\mathbf{D}}(H_Q) = \Pr_{(x+\epsilon,y)\sim\Delta}[H_Q(x+\epsilon)\neq y] = R_\Delta(H_Q).$$

Proof. Starting from the averaged adversarial risk $R_{\mathbf{D}}(H_Q) = \mathbb{E}_{((x,y),\epsilon)\sim\mathbf{D}}\,\mathbf{I}\left[H_Q(x+\epsilon)\neq y\right]$, we have

$$\begin{aligned} R_{\mathbf{D}}(H_Q) &= \mathop{\mathbb{E}}_{(x'+\epsilon',y')\sim\Delta} \frac{1}{\Delta(x'+\epsilon',y')}\left[\Pr_{((x,y),\epsilon)\sim\mathbf{D}}[H_Q(x+\epsilon)\neq y, x'+\epsilon'=x+\epsilon, y'=y]\right] \\ &= \mathop{\mathbb{E}}_{(x'+\epsilon',y')\sim\Delta} \frac{1}{\Delta(x'+\epsilon',y')}\left[\mathop{\mathbb{E}}_{((x,y),\epsilon)\sim\mathbf{D}}\mathbf{I}[H_Q(x+\epsilon)\neq y]\,\mathbf{I}[x'+\epsilon'=x+\epsilon, y'=y]\right]. \end{aligned}$$

In other words, the double expectation only rearranges the terms of the original expectation: given an example $(x'+\epsilon',y')$, we gather probabilities such that $H_Q(x+\epsilon)\neq y$ with $(x+\epsilon,y)=(x'+\epsilon',y')$ in the inner expectation, while integrating over all couple $(x'+\epsilon', y') \in X\times Y$ in the outer expectation. Then, from the fact that when $x'+\epsilon'=x+\epsilon$ and that $y'=y$, $\mathbf{I}[H_Q(x+\epsilon)\neq y] = \mathbf{I}[H_Q(x'+\epsilon')\neq y']$, we have

$$\begin{aligned} R_{\mathbf{D}}(H_Q) &= \mathop{\mathbb{E}}_{(x'+\epsilon',y')\sim\Delta} \frac{1}{\Delta(x'+\epsilon',y')}\left[\mathop{\mathbb{E}}_{((x,y),\epsilon)\sim\mathbf{D}}\mathbf{I}[H_Q(x'+\epsilon')\neq y']\mathbf{I}[x'+\epsilon'=x+\epsilon, y'=y]\right] \\ &= \mathop{\mathbb{E}}_{(x'+\epsilon',y')\sim\Delta} \frac{1}{\Delta(x'+\epsilon',y')}\left[\mathbf{I}[H_Q(x'+\epsilon')\neq y']\mathop{\mathbb{E}}_{((x,y),\epsilon)\sim\mathbf{D}}\mathbf{I}[x'+\epsilon'=x+\epsilon, y'=y]\right]. \end{aligned}$$

Finally, by definition of $\Delta(x'+\epsilon',y')$, we can deduce that

$$\begin{aligned} R_{\mathbf{D}}(H_Q) &= \mathop{\mathbb{E}}_{(x'+\epsilon',y')\sim\Delta} \frac{1}{\Delta(x'+\epsilon',y')}[\mathbf{I}[H_Q(x'+\epsilon')\neq y']\,\Delta(x'+\epsilon',y')] \\ &= \mathop{\mathbb{E}}_{(x'+\epsilon',y')\sim\Delta}\mathbf{I}[H_Q(x'+\epsilon')\neq y'] = R_\Delta(H_Q). \end{aligned}$$

□

Similarly, Lemma 9 shows that $R^{\mathtt{ROB}}_D(H_Q)$ is equivalent to $R_\Pi(H_Q)$.

Lemma 9. *For any distribution D on $X \times Y$ and its associated distribution Π, for any posterior $\mathcal{Q}$ on $\mathcal{H}$, we have*

$$R_D^{\texttt{ROB}}(H_{\mathcal{Q}}) = \Pr_{(x+\epsilon,y)\sim\Pi}[H_{\mathcal{Q}}(x+\epsilon)\neq y] = R_\Pi(H_{\mathcal{Q}}).$$

Proof. The proof is similar to the one of Lemma 8. Indeed, starting from the definition of $R_D^{\texttt{ROB}}(H_{\mathcal{Q}}) = \mathbb{E}_{(x,y)\sim D}\,\mathbf{I}[H_{\mathcal{Q}}(x+\epsilon^*(x,y) \neq y]$, we have

$$\begin{aligned} R_D^{\texttt{ROB}}(H_{\mathcal{Q}}) &= \mathop{\mathbb{E}}_{(x'+\epsilon',y')\sim\Pi} \tfrac{1}{\Pi(x'+\epsilon',y')} \left[\mathop{\mathbb{E}}_{(x,y)\sim D} \mathbf{I}\left[H_{\mathcal{Q}}(x+\epsilon^*(x,y)) \neq y\right] \mathbf{I}[x'+\epsilon'=x+\epsilon^*(x,y), y'=y] \right] \\ &= \mathop{\mathbb{E}}_{(x'+\epsilon',y')\sim\Pi} \tfrac{1}{\Pi(x'+\epsilon',y')} \left[\mathop{\mathbb{E}}_{(x,y)\sim D} \mathbf{I}\left[H_{\mathcal{Q}}(x'+\epsilon') \neq y'\right] \mathbf{I}[x'+\epsilon'=x+\epsilon^*(x,y), y'=y] \right]. \end{aligned}$$

Finally, by definition of $\Pi(x'+\epsilon', y')$, we can deduce that

$$\begin{aligned} R_D^{\texttt{ROB}}(H_{\mathcal{Q}}) &= \mathop{\mathbb{E}}_{(x'+\epsilon',y')\sim\Pi} \tfrac{1}{\Pi(x'+\epsilon',y')} \left[\mathbf{I}\left[H_{\mathcal{Q}}(x'+\epsilon') \neq y'\right] \Pi(x'+\epsilon',y')\right] \\ &= \mathop{\mathbb{E}}_{(x'+\epsilon',y')\sim\Pi} \mathbf{I}\left[H_{\mathcal{Q}}(x'+\epsilon')\neq y'\right] = R_\Pi(H_{\mathcal{Q}}). \end{aligned}$$

□

We can now prove Proposition 4.

Proposition 10. *For any distribution $\mathbf{D}$ on $(X \times Y) \times B$, for any distribution $\mathcal{Q}$ on $\mathcal{H}$, we have*

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

Proof. From Lemmas 8 and 9, we have

$$R_{\mathbf{D}}(H_{\mathcal{Q}}) = R_\Delta(H_{\mathcal{Q}}), \quad \text{and} \quad R_D^{\texttt{ROB}}(H_{\mathcal{Q}}) = R_\Pi(H_{\mathcal{Q}}).$$

Then, we apply Lemma 4 of Ohnishi and Honorio (2020), we have

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

□

B Details on the Validity of the Bounds

In this section, we discuss about the validity of the bound when *(i)* generating perturbed sets such as $\mathbf{T}_{\mathrm{U}}$ or $\mathbf{S}$ from a distribution $\mathbf{D}$ dependent on the prior $\mathcal{P}$ *(ii)* selecting the prior $\mathcal{P}$ with $\mathcal{S}$.

Actually, computing the bounds implies perturbing examples, *i.e.*, generating examples from $\mathbf{D}$ that is defined as $\mathbf{D}((x,y),\epsilon) = D(x,y)\cdot\omega_{(x,y)}(\epsilon)$. However, in order to obtain valid bounds, $\omega_{(x,y)}$ must be defined *a priori*. Since the prior $\mathcal{P}$ is defined *a priori* as well, $\omega_{(x,y)}$ can be dependent on $\mathcal{P}$. Hence, $\omega_{(x,y)}$ boils down to generate perturbed example $(x{+}\epsilon,y)$ by attacking the prior network $\mathcal{P}$ with $\mathrm{PGD}_{\mathrm{U}}$ or $\mathrm{IFGSM}_{\mathrm{U}}$. Nevertheless, our selection of the prior $\mathcal{P}$ with $\mathcal{S}$ may seem like "cheating", but this remains a valid strategy when we perform a union bound.

We explain the union bound for Theorem 6, and the same technique can be applied for Theorem 7.
Let $\mathbf{D}_1,\ldots,\mathbf{D}_T$ be T distributions defined as $\mathbf{D}_1 = D(x,y)\cdot\omega^1_{(x,y)}(\epsilon),\ldots,\mathbf{D}_T = D(x,y)\cdot\omega^T_{(x,y)}(\epsilon)$ on $(X{\times}Y){\times}B$ where each distribution $\omega^t_{(x,y)}$ depends on the example (x,y) and possibly on the fixed prior $\mathcal{P}_t$. Furthermore, we denote as $(\mathbf{D}_t^n)^m$ the distribution on the perturbed learning sample constituted by m examples and n perturbations for each example. Then, for all distributions $\mathbf{D}_t$, we can derive a bound on the risk $\overline{R_{\mathbf{D}_t}}(H_{\mathcal{Q}})$ which holds with probability at least $1-\frac{\delta}{T}$, we have

$$\begin{aligned}&\Pr_{\mathbf{S}_t\sim(\mathbf{D}_t^n)^m}\left[\forall\mathcal{Q},\ \mathrm{kl}(\overline{R_{\mathbf{S}_t}}(H_{\mathcal{Q}})\|\overline{R_{\mathbf{D}_t}}(H_{\mathcal{Q}}))\leq\frac{1}{m}\left[\mathrm{KL}(\mathcal{Q}\|\mathcal{P}_t)+\ln\frac{T(m+1)}{\delta}\right]\right]\\ =&\Pr_{\mathbf{S}_1\sim(\mathbf{D}_1^n)^m,\ldots,\mathbf{S}_T\sim(\mathbf{D}_T^n)^m}\left[\forall\mathcal{Q},\ \mathrm{kl}(\overline{R_{\mathbf{S}_t}}(H_{\mathcal{Q}})\|\overline{R_{\mathbf{D}_t}}(H_{\mathcal{Q}}))\leq\frac{1}{m}\left[\mathrm{KL}(\mathcal{Q}\|\mathcal{P}_t)+\ln\frac{T(m+1)}{\delta}\right]\right]\geq 1-\tfrac{\delta}{T}.\end{aligned}$$

Then, from a union bound argument, we have

$$\begin{aligned}\Pr_{\mathbf{S}_1\sim(\mathbf{D}_1^n)^m,\ldots,\mathbf{S}_T\sim(\mathbf{D}_T^n)^m}\Bigg[&\forall\mathcal{Q},\ \mathrm{kl}(\overline{R_{\mathbf{S}_1}}(H_{\mathcal{Q}})\|\overline{R_{\mathbf{D}_1}}(H_{\mathcal{Q}}))\leq\frac{1}{m}\left[\mathrm{KL}(\mathcal{Q}\|\mathcal{P}_t)+\ln\frac{T(m+1)}{\delta}\right],\\ &\text{and }\ldots,\\ &\text{and }\mathrm{kl}(\overline{R_{\mathbf{S}_T}}(H_{\mathcal{Q}})\|\overline{R_{\mathbf{D}_T}}(H_{\mathcal{Q}}))\leq\frac{1}{m}\left[\mathrm{KL}(\mathcal{Q}\|\mathcal{P}_T)+\ln\frac{T(m+1)}{\delta}\right]\Bigg]\geq 1-\delta.\end{aligned}$$

Hence, we can select $\mathcal{P}{\in}\{\mathcal{P}_1,\ldots,\mathcal{P}_T\}$ with $\mathcal{S}$, and let $\mathbf{D}((x,y),\epsilon) = D(x,y)\cdot\omega_{(x,y)}(\epsilon)$ be the distributions on $(X{\times}Y){\times}B$ where $\omega_{(x,y)}(\epsilon)$ is dependent on $\mathcal{P}$ and on the example (x,y), we can say that

$$\Pr_{\mathbf{S}\sim(\mathbf{D}^n)^m}\left[\forall\mathcal{Q},\ \mathrm{kl}(\overline{R_{\mathbf{S}}}(H_{\mathcal{Q}})\|\overline{R_{\mathbf{D}}}(H_{\mathcal{Q}}))\leq\frac{1}{m}\left[\mathrm{KL}(\mathcal{Q}\|\mathcal{P})+\ln\frac{T(m+1)}{\delta}\right]\right]\geq 1-\delta.$$

Additionally, when applying the same process for Equation (11) in Theorem 7, we have

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

C Optimizing and Computing the Bounds

In this section, we explain how we compute and optimize the bounds obtained from Algorithm 1. Additionally, note that our losses $\ell_{\mathtt{bnd}}$ and our computed bounds are instantiated with the KL divergence between $\mathcal{P}=\mathcal{N}(\boldsymbol{v}_{best},\lambda\boldsymbol{I}_d)$ and $\mathcal{Q}=\mathcal{N}(\boldsymbol{w},\lambda\boldsymbol{I}_d)$, *i.e.*, we have $\mathrm{KL}(\mathcal{Q}\|\mathcal{P})=\frac{1}{2\lambda}\|\boldsymbol{w}{-}\boldsymbol{v}_{best}\|_2^2$. Furthermore, remark that the bounds involve the number of epochs T (see Section B for more details).

Optimizing the bound. The optimization differs when we optimize the bound of Theorem 6 or 7. Equation (9) is not directly optimizable since we upper-bound a deviation (the kl) between the empirical and true risk. Instead, we use the following loss $\ell_{\mathtt{bnd}}$

$$\ell_{\mathtt{bnd}}(h_{\boldsymbol{w}'},(x{+}\epsilon,y),\boldsymbol{v}_{best})=\frac{1-\exp\left(-C\frac{1}{2}\Big[1-yh_{\boldsymbol{w}'}(x{+}\epsilon)\Big]-\frac{1}{m}\left[\frac{1}{2\lambda}\|\boldsymbol{w}{-}\boldsymbol{v}_{best}\|_2^2+\ln\frac{T(m+1)}{\delta}\right]\right)}{1-\exp(-C)},$$

which involves another parameter C that is learned during the optimization (we set $C = 0.05$ at the initialization). The main advantage of this bound is that, when C is optimal, it will give the same upper-bound as Equation (9)[1]. On the contrary, the loss $\ell_{\texttt{bnd}}$ for Theorem 7 is crafted to minimize Equation 11. Indeed, we minimize

$$\ell_{\texttt{bnd}}(h_{\boldsymbol{w}'}, (x{+}\epsilon, y), \boldsymbol{v}_{best}) = \frac{1}{2}\Big[1{-}yh_{\boldsymbol{w}'}(x{+}\epsilon)\Big]{+}\sqrt{\frac{1}{2m}\left[\frac{1}{2\lambda}\|\boldsymbol{w}{-}\boldsymbol{v}_{best}\|_2^2{+}\ln\tfrac{2T\sqrt{m}}{\delta}\right]},$$

Note that we do not optimize the TV distance since this term is zero when we approximate the expectation over the voters $h \sim \mathcal{Q}$ with only one voter, *i.e.*, we can set $\pi_i = \rho_i^h$ for each example (x_i, y_i).

Computing the bounds. Concerning the computation of Equation (9), note that computing $\overline{R_{\mathbf{S}}}(H_{\mathcal{Q}})$ is not feasible in practice because of the expectation $\mathbb{E}_{h\sim\mathcal{Q}}\, h(x{+}\epsilon)$. Hence, we approximate this quantity with a Monte Carlo estimation, *i.e.*, we sample N networks (*i.e.*, N weights) $h_1, h_2, \ldots, h_N \sim \mathcal{Q}$ and we compute the approximated quantity

$$\overline{R_{\mathbf{S}}}(H_{\mathcal{Q}}) \approx \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\frac{1}{2}\left[1{-}y_i\sum_{k=1}^{N}h_k(x_i{+}\epsilon_j^i)\right].$$

From this estimation, we compute the worst possible true risk satisfying Equation (9), *i.e.*, we compute

$$\sup_{\overline{R_{\mathbf{S}}}(H_{\mathcal{Q}})\le r\le 1}\left\{r \;\middle|\; \mathrm{kl}(\overline{R_{\mathbf{S}}}(H_{\mathcal{Q}})\|r) \le \frac{1}{m}\left[\frac{1}{2\lambda}\|\boldsymbol{w}{-}\boldsymbol{v}_{best}\|_2^2{+}\ln\frac{T(m{+}1)}{\delta}\right]\right\}.$$

Similarly, for Equation (11), we approximate the risk $\overline{A_{\mathbf{S}}}(H_{\mathcal{Q}})$ and the expected value over h of the TV divergence term

$$\overline{A_{\mathbf{S}}}(H_{\mathcal{Q}}) \approx \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\Big[1{-}\min_{\epsilon\in\boldsymbol{\mathcal{E}}_i}\Big(y_i\sum_{k=1}^{N}h_k(x_i{+}\epsilon)\Big)\Big], \quad \text{and} \quad \mathop{\mathbb{E}}_{h\sim\mathcal{Q}}\mathrm{TV}(\rho_i^h\|\pi_i) \approx \frac{1}{N}\sum_{k=1}^{N}\mathrm{TV}(\rho_i^{h_k}\|\pi_i),$$

where $\pi_i(\epsilon) = \frac{1}{N}\sum_{k=1}^{N}\rho_i^{h_k}(\epsilon)$ for our approximation. Finally, the bound that is computed for Theorem 7 is

$$\overline{A_{\mathbf{S}}}(H_{\mathcal{Q}}) + \frac{1}{mN}\sum_{i=1}^{m}\sum_{k=1}^{N}\mathrm{TV}(\rho_i^{h_k}\|\pi_i) + \sqrt{\frac{1}{2m}\left[\frac{1}{2\lambda}\|\boldsymbol{w}{-}\boldsymbol{v}_{best}\|_2^2{+}\ln\tfrac{2T\sqrt{m}}{\delta}\right]}.$$

[1] See Theorem 3 in"Dichotomize and Generalize: PAC-Bayesian Binary Activated Deep Neural Networks" of Letarte *et al.*

D Additional Experimental Results

In this section, we report the results for **MNIST 4vs9** and **MNIST 5vs6**, respectively on Table 2 and Table 3. We train all the models using Adam optimizer for 100 epochs with a learning rate at $1e^{-4}$ and a batch size at 64. Moreover, we fix the confidence parameter δ=.05 for the bounds. For these tasks, we have a similar behavior than **MNIST 1vs7**. Indeed, using the attacks PGD_U and IFGSM_U as defense mechanism allows to obtain better risks and also tighter bounds. Compared to the bounds obtained with a defense based on UNIF(which is a naive defense), the bounds obtained with a defense based on PGD_U or IFGSM_U are improved from 17% to 31% for the task **MNIST 4vs9** (bold values in Table 2) and from 25% to 29% for the task **MNIST 5vs6** (bold values in Table 3). These results make sense since the defenses based on PGD_U and IFGSM_U are more elaborated than the one based on UNIF and therefore lead to better guarantees. When the perturbation is set to 0.3, in most cases, the bounds are non-informative (above 1) and the risks are around 0.5, (*i.e.*, almost half of the time the model predictions are false on perturbed examples). Moreover, we can notice that the risks of the baseline are also high, suggesting that the task becomes hard to learn when the perturbation is set to 0.3.

TABLE 2: The table shows the different test risks and bounds for **MNIST 4vs9** with n=50 perturbations for all the pairs (Defense, Attack). Tthe results in *italic* corresponds to the baseline on the deterministic network $h_{\boldsymbol{w}}$: importantly, for the baseline, we did *not* sampled from the uniform distribution, but we put the results in the table as a reference.

		$b = 0.1$					$b = 0.3$				
		baseline	Algo.1		Algo.1		baseline	Algo.1		Algo.1	
		without U	with Eq. (9)		with Eq. (11)		without U	with Eq. (9)		with Eq. (11)	
Defense	Attack	$R_{\mathbf{T}}(h_{\boldsymbol{w}})$	$R_{\mathbf{T}_\text{U}}(H_Q)$	Th. 6	$A_{\mathbf{T}_\text{U}}(H_Q)$	Th. 7	$R_{\mathbf{T}}(h_{\boldsymbol{w}})$	$R_{\mathbf{T}_\text{U}}(H_Q)$	Th. 6	$A_{\mathbf{T}_\text{U}}(H_Q)$	Th. 7
—	PGD_U	*0.1989*	0.2167	0.7362	0.2220	0.7462	*0.4917*	0.4905	1.0472	0.5073	1.0883
—	IFGSM_U	*0.1979*	0.2190	0.7369	0.2210	0.7479	*0.4937*	0.4921	1.0473	0.5043	1.0814
UNIF	PGD_U	*0.2215*	**0.2340**	**0.6717**	**0.2526**	**0.7453**	*0.4927*	0.4889	1.0662	0.5103	1.0921
UNIF	IFGSM_U	*0.2210*	**0.2375**	**0.6783**	**0.2557**	**0.7503**	*0.4902*	0.4834	1.0565	0.5048	1.0832
PGD_U	PGD_U	*0.3184*	**0.1919**	**0.4932**	**0.1934**	**0.5258**	*0.4807*	0.4932	1.0695	0.4932	1.0895
PGD_U	IFGSM_U	*0.2687*	**0.1838**	**0.4730**	**0.1858**	**0.5075**	*0.4571*	0.4932	1.0695	0.4932	1.0895
IFGSM_U	PGD_U	*0.2793*	**0.1535**	**0.5035**	**0.1542**	**0.4826**	*0.5058*	0.3778	0.9624	0.3852	0.9980
IFGSM_U	IFGSM_U	*0.1743*	**0.1415**	**0.4483**	**0.1437**	**0.4321**	*0.4425*	0.2900	0.7898	0.2928	0.8220

TABLE 3: The table shows the different test risks and bounds for **MNIST 5vs6** with n=50 perturbations for all the pairs (Defense, Attack). The results in *italic* corresponds to the baseline on the deterministic network $h_{\boldsymbol{w}}$: importantly, for the baseline, we did *not* sampled from the uniform distribution, but we put the results in the table as a reference.

		$b = 0.1$					$b = 0.3$				
		baseline	Algo.1		Algo.1		baseline	Algo.1		Algo.1	
		without U	with Eq. (9)		with Eq. (11)		without U	with Eq. (9)		with Eq. (11)	
Defense	Attack	$R_{\mathbf{T}}(h_{\boldsymbol{w}})$	$R_{\mathbf{T}_\text{U}}(H_Q)$	Th. 6	$A_{\mathbf{T}_\text{U}}(H_Q)$	Th. 7	$R_{\mathbf{T}}(h_{\boldsymbol{w}})$	$R_{\mathbf{T}_\text{U}}(H_Q)$	Th. 6	$A_{\mathbf{T}_\text{U}}(H_Q)$	Th. 7
—	PGD_U	*0.2265*	0.2720	0.6699	0.2719	0.6915	*0.4762*	0.4843	1.0351	0.5027	1.0796
—	IFGSM_U	*0.2346*	0.2748	0.6771	0.2762	0.6998	*0.4703*	0.4928	1.0185	0.5049	1.0637
UNIF	PGD_U	*0.2503*	**0.2564**	**0.6149**	**0.2627**	**0.6373**	*0.4784*	0.4657	1.0056	0.4741	1.0447
UNIF	IFGSM_U	*0.2503*	**0.2582**	**0.6170**	**0.2638**	**0.6404**	*0.4838*	0.4532	0.9864	0.4638	1.0191
PGD_U	PGD_U	*0.1162*	**0.1130**	**0.3364**	**0.1151**	**0.3817**	*0.3227*	0.2826	0.7840	0.2805	0.8047
PGD_U	IFGSM_U	*0.1162*	**0.1114**	**0.3249**	**0.1119**	**0.3713**	*0.2411*	0.2260	0.6337	0.2281	0.6584
IFGSM_U	PGD_U	*0.1481*	**0.1077**	**0.3626**	**0.1043**	**0.3965**	*0.5157*	0.4393	1.0104	0.4449	1.0430
IFGSM_U	IFGSM_U	*0.1211*	**0.1087**	**0.3585**	**0.1081**	**0.3950**	*0.2449*	0.1957	1.0477	0.1946	0.6347