Non-vacuous PAC-Bayes bounds for Models under Adversarial Corruptions

Waleed Mustafa [1] **Philipp Liznerski** [1] **Dennis Wagner** [1] **Puyu Wang** [2] **Marius Kloft** [1]

Abstract

PAC-Bayes generalization bounds have been shown to provide non-vacuous performance certificates for several Machine Learning models. However, under adversarial corruptions, these bounds often fail to maintain their non-vacuous nature due to the increased empirical risk. In this work, we address this limitation by deriving and computing the first non-vacuous generalization bounds for models operating under adversarial conditions. Our approach combines the PAC-Bayes and Adversarial Smoothing frameworks to derive generalization bounds for randomly smoothed models. We empirically demonstrate the efficacy of our bounds in providing robust population risk certificates for stochastic Convolution Neural Networks (CNN) operating under L_2-bounded adversarial corruptions for both MNIST and CIFAR-10.

1. Introduction

Deep neural networks (DNN) are known to outperform other models in complex applications (LeCun et al., 2015). However, it is difficult to justify their use in modern safety-critical applications, as DNN models are generally susceptible to various security threats (Szegedy et al., 2013; Papernot et al., 2016), particularly adversarial examples (Biggio et al., 2013; Szegedy et al., 2013).

Past attempts to predict the robustness of trained models from training data (Yin et al., 2019; Awasthi et al., 2020; Khim and Loh, 2018; Mustafa et al., 2022; Gao and Wang, 2021; Farnia et al., 2018) inherit the limitations of uniform convergence bounds in explaining the generalization of DNNs (Nagarajan and Kolter, 2019). The resulting bounds for modern models are vacuous, thus, while providing valuable theoretical insights, are of little practical use.

[1]University of Kaiserslautern-Landau, Kaiserslautern, Germany [2]City University of Hong Kong, Hong Kong. Correspondence to: Waleed Mustafa <mustafa@cs.uni-kl.de>.

Appearing in the *PAC-Bayes Meets Interactive Learning Workshop at ICML 2023*, Honolulu, Hawaii, USA.

Dziugaite and Roy (2017) introduced the concept of non-vacuous bounds on the population risk of stochastic DNNs, leading to the emergence of self-certified DNNs. Self-certified DNNs refer to models or algorithms that provide population risk certificates based solely on the training data (Pérez-Ortiz et al., 2021). These risk certificates play a crucial role in deploying DNNs in sensitive scenarios. Such certificates, however, are lacking in adversarial settings.

On the other hand, existing robustness verification methods are capable of providing robustness certificates against adversarial examples for individual test samples. For instance, exact verification methods (Katz et al., 2017a; Ehlers, 2017; Tjeng et al., 2017) provide deterministic robustness certificates to a given test sample. Yet, they suffer from high computational complexity, particularly for large models. In contrast, randomized smoothing techniques (Cohen et al., 2019) have been proposed to scale up to deeper models. These algorithms, however, are primarily focused on test-time verification and cannot provide adversarial risk certificates based solely on training data.

To overcome these limitations, our objective is to derive and compute non-vacuous generalization bounds for the adversarial loss. Our approach is designed for stochastic DNNs and enables the computation of population risk certificates for the adversarial loss. Although we primarily focus on the L_2 measure of robustness in this work, our approach can be extended to other measures as well.

2. Non-vacuous generalization bounds in an adversarial setting

Problem setting We start by introducing the notation and problem setting. Let $\mathcal{X} \subset \mathbb{R}^d$ denote the input space and $\mathcal{Y} \subset \{0,1\}^K$ the output space (one-hot encoding of K classes). The joint input-output space $\mathcal{X} \times \mathcal{Y}$ is endowed with an unknown probability measure P. We consider a stochastic classification setting using classifiers $h : (W, X) \mapsto h(W; x)$ parameterized by vectors $W \in \mathcal{W} \subset \mathbb{R}^p$, where the classifier is represented by a probability measure $Q \in \mathcal{M}(\mathcal{W})$ on the set of parameters $\mathcal{W}$. Here $\mathcal{M}(\mathcal{W})$ is the set of all probability measures on $\mathcal{W}$. We measure the prediction quality with the loss $\ell : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}^{\mathcal{X}} \to [0,1]$. For instance, the 0-1 loss $\ell^{01}(x, y, h(W;\cdot)) = \mathbb{I}(h(W;x) = y)$. The risk associated

with the stochastic prediction Q is defined as: $L(Q,\ell) := \mathbb{E}_{W\sim Q}\left[\mathbb{E}_{(x,y)\sim P}\ell(x,y,h(W;\cdot))\right]$ Our goal is to learn Q by minimizing the risk (i.e., $Q^* = \arg\min_Q L(Q,\ell)$), but L depends on the unknown population distribution P. Thus, we resort to minimizing the empirical risk: $\widehat{L}(Q,S,\ell) := \mathbb{E}_{W\sim Q}[\frac{1}{n}\sum_{i=1}^n \ell(x_i,y_i,h(W;\cdot))]$, where $S := \{(x_i,y_i) \sim P \mid i \in [n]\}$ is an i.i.d training sample. Here, $[n] = \{1,\dots,n\}$. Evaluating and optimizing $\widehat{L}(Q,S,\ell)$ directly is often computationally intractable due to the expectation with respect to the probability measure Q. To address this, we resort to Monte Carlo approximation of Q using m i.i.d. samples $\{W_j \sim Q \mid j \in [m]\}$, resulting in an unbiased estimate: $\widehat{L}(\widehat{Q},S,\ell) \approx \frac{1}{mn}\sum_{j=1}^m\sum_{i=1}^n \ell\big(x_i,y_i,h(W_j;\cdot)\big)$.

We consider an attack model where an adversary manipulates the input x by adding noise to it to disrupt the classifier's prediction. That is, the adversary's goal is to find an altered input $\tilde{x}$ deviating from the original input x by a certain Euclidean distance while incurring a maximal loss. In other words, the adversary seeks to solve the optimization problem $\tilde{x} = \arg\max_{\tilde{x}:\|x-\tilde{x}\|_2<R}\ell(\tilde{x},y,h(W;\cdot))$. The adversarial loss is defined as $\ell_{\text{adv}}(x,y,h(W;\cdot)) := \max_{\tilde{x}:\|x-\tilde{x}\|_2<R}\ell(\tilde{x},y,h(W;\cdot))$.

2.1. PAC-Bayesian bounds for neural networks in adversarial environments

Now we introduce our approach for computing non-vacuous generalization bounds of adversarial deep learning. We start by considering an idealized setting. In Section 2.2, we extend the results to compute practical certificates.

We first apply the classical PAC-Bayes theorem (Langford and Caruana, 2001; McAllester, 1999) (see Theorem E.7 in the appendix) to the adversarial loss to get, with probability at least $1-\delta$, $L(Q,\ell_{\text{adv}})$

$$\leq \text{KL}^{-1}\left(\widehat{L}(Q,S,\ell_{\text{adv}}), \frac{\text{KL}(Q||Q^0)+\ln(\frac{2\sqrt{n}}{\delta})}{n}\right). \quad (1)$$

where $\text{KL}(\cdot||\cdot)$ is the Kullback-Leibler (KL) divergence, $\text{KL}(p,q)$ for $p,q \in [0,1]$ is the KL-divergence between Bernouli distributions with parameters p and q, $\text{KL}^{-1}(p,c) := \sup\{q\in[0,1] : \text{KL}(p,q)\leq c\}$. Computing the bound (1) is computationally challenging due to the intractability of evaluating the expectation with respect to Q in $\widehat{L}(Q,S,\ell_{\text{adv}})$ and solving $\max_{\tilde{x}:\|x-\tilde{x}\|<R}\ell(\tilde{x},y,h)$ for DNNs (Madry et al., 2017). To address the first challenge, we utilize Monte Carlo sampling to approximate Q (Langford and Caruana, 2001) (see Lemma E.8). To address the problem of evaluating ℓ_{adv} we resort to adversarial verification methods. Exact verification methods for computing an upper bound on ℓ_{adv} are computationally prohibitive, particularly for larger models (Xiao et al., 2018; Katz et al., 2017b). Furthermore, these methods require the given model to be robust, which is often difficult to satisfy for a subset of $\mathcal{W}$ with a large probability measure under Q. To overcome these issues, we employ *Randomized Smoothing* (RS) (Cohen et al., 2019), which allows us to derive efficient and scalable upper bounds on ℓ_{adv} without the robustness assumption of the original model.

RS transforms a classifier $h(W;\cdot)$ into a provably robust classifier $g(W;\cdot)$ by applying the operator $\mathcal{T}_\sigma$ defined by

$$g(W;x) = \mathcal{T}_\sigma h(W;x) := \arg\max_{y\in\mathcal{Y}} \Pr[h(W;x+\epsilon)=y],$$

for $x\in\mathcal{X}$, $W\in\mathcal{W}$. Here, $\epsilon\sim\mathcal{N}(0,\sigma I)$ represents a random noise vector and $\sigma>0$ determines the level of smoothing. The smoothed classifier $g(W;\cdot)$ selects the output class y that maximizes the probability of the original classifier $h(W;\cdot)$ producing the same output class for perturbed inputs $x+\epsilon$. This smoothing process makes the classifier more robust to small input perturbations (Cohen et al., 2019). The lemma below presents an upper bound on the empirical adversarial loss of the randomized smoothing classifier. The detailed proof can be found in Appendix E.

Lemma 2.1. *Let $\epsilon\sim\mathcal{N}(0,\sigma I)$ with $\sigma>0$, and let $\overline{p},\underline{p}: \mathcal{W}\times\mathcal{X}\to[0,1]$ such that, for all $(x,y)\in S$ and $W\in\mathcal{W}$,*

$$P_g(W;x) \geq \overline{p}(W,x) \geq \underline{p}(W,x) \geq P_{\neg g}(W;x),$$

where $P_g(W;x) := \Pr(h(W;x+\epsilon)=g(W;x))$ and $P_{\neg g}(W;x) := \max_{c\neq g(W;x)}\Pr(h(W;x+\epsilon)=c)$. Then,

$$\widehat{L}(Q,S,\ell_{\text{adv}}) \leq \widehat{L}(Q,S,\tilde{\ell}), \text{ where, } \tilde{\ell}(x,y,g(W;\cdot)) :=$$

$$\max\{\ell(x,y,g(W;\cdot)), \mathbb{I}(\Phi^{-1}(\overline{p}(W,x)) - \Phi^{-1}(\underline{p}(W,x)) \geq \tfrac{2R}{\sigma})\}$$

and Φ is the CDF of a standard normal distribution.

Note that $\overline{p}(W,x)$ serves as a lower bound on the probability of the class predicted by the smoothed classifier $g(W;x)$, while $\underline{p}(W,x)$ represents an upper bound on the probability of any other class. The loss function $\tilde{\ell}$ captures the idea that as the difference between $\overline{p}(W,x)$ and $\underline{p}(W,x)$ increases, the robustness of $g(W,x)$ also improves. This means that when the gap between $\overline{p}(W,x)$ and $\underline{p}(W,x)$ is large, indicating high confidence in the predicted class, $\tilde{\ell}$ coincides with the natural loss $\ell(x,y,g(W;\cdot))$. However, when the difference between $\overline{p}(W,x)$ and $\underline{p}(W,x)$ is small, suggesting a lack of robustness, $\tilde{\ell}$ takes on its maximum value, indicating the potential presence of an adversarial example for $g(W;x)$. In this way, the loss function $\tilde{\ell}$ provides a measure of the robustness of the smoothed classifier $g(W,x)$ based on the probability bounds $\underline{p}(W,x)$ and $\overline{p}(W,x)$.

Lemma 2.1 provides an upper bound on $\widehat{L}(Q,S,\ell_{\text{adv}})$ using $\widehat{L}(Q,S,\tilde{\ell})$, allowing us to replace the computation of ℓ_{adv} with that of $\tilde{\ell}$. Notably, in $\tilde{\ell}$, there is no need to solve the intractable adversarial optimization problem

$\max_{\|\tilde{x}-x\|_2<R} \ell(\tilde{x}, y, g(W;\cdot))$. Although the exact computation of $\overline{p}(W,x)$ and $\underline{p}(W,x)$ within $\tilde{\ell}$ is also intractable, we can efficiently estimate them using sampling-based techniques (Cohen et al., 2019) (See Section 2.2). The final bound is summarized in the following theorem, which is derived from Lemma 2.1 and the bound (1). The detailed proof can be found in Appendix E.

Theorem 2.2. *Let $Q^0 \in \mathcal{M}(\mathcal{W})$ be a prior probability measure. Then, with probability $1-\delta$ over the randomness of the training sample S, simultaneously for all $Q \in \mathcal{M}(\mathcal{W})$, we have $L(Q, \ell_{\text{adv}}) \leq$*

$$\mathrm{KL}^{-1}\left(\widehat{L}(Q,S,\tilde{\ell}), \frac{\mathrm{KL}(Q||Q^0) + \ln(\frac{2\sqrt{n}}{\delta})}{n}\right). \quad (2)$$

2.2. Computing the certificate

In this section, we provide a tractable upper bound on (2) that holds with high probability. The main computational challenge lies in computing $\widehat{L}(Q,S,\tilde{\ell})$. To address this challenge, we first efficiently approximate the expectation with respect to Q by Monte Carlo sampling (Langford and Caruana, 2001) (see Lemma E.8) to get the bound, with probability at least $1-\delta$ over the randomness in $\widehat{Q}$,

$$\widehat{L}(Q,S,\tilde{\ell}) \leq \mathrm{KL}^{-1}\left(\widehat{L}(\widehat{Q},S,\tilde{\ell}), \frac{\ln(\frac{2}{\delta})}{n}\right). \quad (3)$$

Algorithm 1 Estimate an upper bound on $\widehat{L}(\widehat{Q},S,\tilde{\ell})$

Input : S, N_0, N, σ, $\{W_i\}_{i=1}^m$, α, R
Output : An estimate of $\widehat{L}(\widehat{Q},S,\tilde{\ell})$

```
1  errors_count ← 0
2  for (x, y) ∈ S do
3      for j ← 1 : m do
4          {ε_t}_{t=1}^{N_0} ← Sample N_0 samples from N(0, σI)
5          counts ← Σ_{t=1}^{N_0} h(W_j; x + ε_t)
6          c_A ←= arg max_{k∈[K]} counts_k
7          {ε_t}_{t=1}^{N} ← Sample N sample from N(0, σI)
8          counts ← Σ_{t=1}^{N} h(W_j; x + ε_t)
9          p_A ← BinomialLowerConf(counts_{c_A}, N, 1 − α)
10         if p_A ≤ 1/2 or c_A ≠ y or Φ^{-1}(p_A) < R/σ then
11             errors_count ← errors_count +1
12         end
13     end
14 end
15 return errors_count /m|S|
```

It remains now to estimate an upper bound on $\widehat{L}(\widehat{Q},S,\tilde{\ell})$. We proceed by addressing the difficulty of evaluating $g(W;\cdot)$, which is computationally intractable. To overcome this, we leverage the `CERTIFY` algorithm proposed by Cohen et al. (2019), which provides a sampling-based approach to approximate $g(W;\cdot)$. Algorithm 1 presents an algorithm to estimate the empirical error $\widehat{L}(\widehat{Q},S,\tilde{\ell})$ based on `CERTIFY`. We first estimate the prediction $c_A \approx g(W_j;x)$ by sampling N_0 instances from $h(W_j;x+\epsilon)$ (lines 4-6). Next, we proceed to estimate a lower bound on $\Pr(h(W_j;x+\epsilon)=c_A)$ that holds with probability at least $1-\alpha$. Specifically, in lines 7-8, we count the number of times $h(W_j,x+\epsilon_t)$ predicts c_A in N trials, $\sum_{t=1}^N \mathbb{I}(h(W_j;x+\epsilon_t)=c_A)$. In line 9, we estimate the lower bound p_A on $\Pr(h(W_j;x+\epsilon)=c_A)$ using the confidence interval estimation procedure BinomialLowerConf (Brown et al., 2001), which ensures a lower bound with probability at least $1-\alpha$. Finally, lines 10-11 compute the upper bound on $\tilde{\ell}(x,y,g(W;\cdot))$. The following lemma assesses the quality of Algorithm 1.

Lemma 2.3. *Let $S \subset \mathcal{X}\times\mathcal{Y}$ and $\mathcal{W} := \{W_i\}_{i=1}^m$ be a set of weights. Let* Alg1 *be the output of Algorithm 1, then with probability at least $1-\delta$ over the randomness of the algorithm,*

$$\widehat{L}(\widehat{Q},S,\tilde{\ell}) \leq B(\widehat{Q},S,\delta),$$

where $B(\widehat{Q},S,\delta) := \left(\text{Alg1} + \alpha + \sqrt{\frac{2\alpha(1-\alpha)\ln(\frac{1}{\delta})}{m|S|}} + \frac{\ln(\frac{1}{\delta})}{3m|S|}\right)$.

We obtain the final certificate by combining Theorem 2.2, Eq.(3), and Lemma 2.3:

Theorem 2.4. *Let $Q^0 \in \mathcal{M}(\mathcal{W})$ be prior distribution and* Alg1 *is the output of algorithm 1. Then with probability at least $1-\delta-\delta'-\delta''$, simultaneously for $Q \in \mathcal{M}(\mathcal{W})$, the adversarial risk $L(Q,\ell_{\text{adv}})$ is upper-bounded by*

$$\mathrm{KL}^{-1}\left(\mathrm{KL}^{-1}\left(B(\widehat{Q},S,\delta''), \frac{\ln(\frac{2}{\delta'})}{n}\right), \frac{\mathrm{KL}(Q||Q^0)+\ln(\frac{2\sqrt{n}}{\delta})}{n}\right)$$

2.3. Training a certifiable network

In this section, we consider training self-certified stochastic models, that is, models for which the bound in Theorem 2.4 is non-vacuous. Recall that the goal of training is to find a posterior distribution $Q \in \mathcal{M}(\mathcal{W})$ that minimizes the adversarial PAC-Bayes bound (1), which can be formulated as selecting $\widehat{Q}$ that attains the minimum

$$\min_{Q\in\mathcal{M}(\mathcal{W})} \mathrm{KL}^{-1}\left(\widehat{L}(Q,S,\ell_{\text{adv}}), \frac{\mathrm{KL}(Q||Q^0)+\ln(\frac{2\sqrt{n}}{\delta})}{n}\right).$$

Evaluating and minimizing the adversarial PAC-Bayes bound directly is challenging. Therefore, we aim to derive a surrogate objective that is amenable to optimization, particularly using SGD-based algorithms. Since the KL^{-1} term does not have a closed-form solution, several upper bounds have been proposed in the literature (McAllester, 1999; Pérez-Ortiz et al., 2021; Tolstikhin and Seldin, 2013; Thiemann et al., 2017). In an extensive empirical study, Pérez-Ortiz et al. (2021) observed that the bound *PAC-Bayes-quadratic* (Rivasplata et al., 2020) outperformed the other

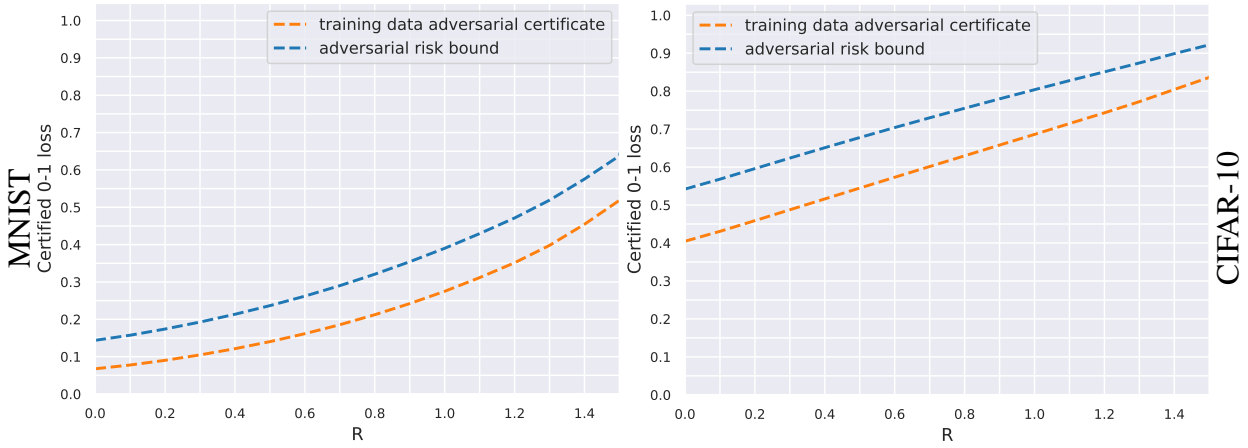


Figure 1. Robust risk certificates for MNIST and CIFAR-10.

candidates across various settings. Therefore, we focus on this bound in our experiments. The *PAC-Bayes-quadratic* bound takes the following form: $f_q(Q, Q^0, S, \ell_{\text{adv}}) :=$

$$\left(\sqrt{\widehat{L}(Q, S, \ell_{\text{adv}}) + \frac{\text{KL}(Q||Q^0)+\ln(\frac{2\sqrt{n}}{\delta})}{2n}} + \sqrt{\frac{\text{KL}(Q||Q^0)+\ln(\frac{2\sqrt{n}}{\delta})}{2n}} \right)^2 .$$

Now we proceed to estimate the gradient of $f_q(Q, Q^0, S, \ell_{\text{adv}})$. It is computationally efficient to compute the KL-divergence and its gradients when using normal distributions with diagonal covariance for both the prior Q^0 and the posterior Q. To estimate the gradients of $\widehat{L}(Q, S, \ell_{\text{adv}})$ with respect to the parameters of Q (i.e. μ and Σ), we use the pathwise gradient estimator (Price, 1958; Jankowiak and Obermeyer, 2018; Pérez-Ortiz et al., 2021). This approach addresses the computational challenges in evaluating the expectation with respect to Q. Specifically, we consider the approximation $\nabla_{(\mu,\Sigma)}\widehat{L}(Q, S, \ell_{\text{adv}}) \approx \nabla_{(\mu,\Sigma)} \frac{1}{n}\sum_{i=1}^{n} \ell_{\text{adv}}(x_i, y_i, g(W;\cdot))$, where $W := \mu + \Sigma^{\frac{1}{2}}V$, $V \sim \mathcal{N}(0, I)$. Next, we focus on the classifier $g(W; x)$. During the training process, we employ the empirical version of the classifier, which is given by $\frac{1}{M}\sum_{t=1}^{M} h(W; x + \epsilon_t)$, where ϵ_t are i.i.d. samples from the normal distribution $\mathcal{N}(0, \sigma I)$. To approximate the adversarial loss, we utilize adversarial training techniques (Madry et al., 2017; Tramèr et al., 2017). Specifically, we adopt the SMOOTHADV approach proposed by Salman et al. (2019), in which Projected Gradient Descent (PGD) is used to find an adversarial example for each training sample. The gradients of the inputs required by PGD are approximated by $\nabla_x \ell\big(\frac{1}{M}\sum_{t=1}^{M} h(W; x + \epsilon_t)\big)$. It is important to note that during the training process, the cross-entropy loss is used as a surrogate for the 0-1 loss. Algorithm 2 in the appendix provides a summary of the training procedure.

3. Experiments

In this section, we demonstrate the practical applications of our self-certified model training and evaluation techniques, showcasing their utility and efficacy. We compute the empirical certificates (Algorithm 1) and the adversarial risk bound (Theorem 2.4) for various training settings on the well-established MNIST and CIFAR-10 datasets.

Our model achieves robust risk certificates across all settings Overall, our method consistently achieves robust certificates across all datasets. Figure 1 shows the results of our experiments. Orange lines depict the bound on $\widehat{L}(\widehat{Q}, S, \tilde{\ell})$ as presented in Algorithm 1, while blue lines depict the bound on $L(Q, \ell_{\text{adv}})$ as given in Theorem 2.4. The computed bounds are indeed non-vacuous for both MNIST on the small network and CIFAR-10 on the deep network.

Our approach scales well with deeper networks When evaluating a deeper network on CIFAR-10, we observed that despite the network's larger size (41M parameters vs. 4.8M parameters), the generalization gap did not significantly change. This finding emphasizes that KL-divergence is a superior measure of complexity for DNNs compared to relying solely on the number of parameters. Other measures with similar implications have been proposed in the literature, such as the distance to initialization (Bartlett et al., 2017; Arora et al., 2019). For further insights, a detailed set of additional experiments is available in the Appendix.

4. Conclusion

In this work, we have addressed the challenge of adversarial attacks in machine learning models by developing novel generalization bounds and practical methodologies for computing robustness certificates. Our approach leverages the PAC-Bayesian and randomized smoothing frameworks and demonstrates the ability to provide non-vacuous certificates for neural networks in adversarial settings. Through rigorous experiments on benchmark datasets, we have validated the effectiveness of our method in yielding robust certificates for stochastic convolutional neural networks.

Acknowledgements

WM, PL, DW, and MK acknowledge support by the Carl-Zeiss Foundation, the DFG awards KL 2698/2-1, KL 2698/5-1, KL 2698/6-1 and KL 2698/7-1, and the BMBF awards 03|B0770E, and 01|S21010C.

References

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The Limitations of Deep Learning in Adversarial Settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387. IEEE, mar 2016. ISBN 978-1-5090-1751-5. doi: 10.1109/EuroSP.2016.36. URL http://ieeexplore.ieee.org/document/7467366/.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 387–402. Springer, 2013.

Dong Yin, Ramchandran Kannan, and Peter Bartlett. Rademacher complexity for adversarially robust generalization. In *International Conference on Machine Learning*, pages 7085–7094. PMLR, 2019.

Pranjal Awasthi, Natalie Frank, and Mehryar Mohri. Adversarial learning guarantees for linear hypotheses and neural networks. In *International Conference on Machine Learning*, pages 431–441. PMLR, 2020.

Justin Khim and Po-Ling Loh. Adversarial risk bounds via function transformation. *arXiv preprint arXiv:1810.09519*, 2018.

Waleed Mustafa, Yunwen Lei, and Marius Kloft. On the generalization analysis of adversarial learning. In *International Conference on Machine Learning*, pages 16174–16196. PMLR, 2022.

Qingyi Gao and Xiao Wang. Theoretical investigation of generalization bounds for adversarial learning of deep neural networks. *Journal of Statistical Theory and Practice*, 15(2):1–28, 2021.

Farzan Farnia, Jesse Zhang, and David Tse. Generalizable adversarial training via spectral normalization. In *International Conference on Learning Representations*, 2018.

Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

María Pérez-Ortiz, Omar Rivasplata, John Shawe-Taylor, and Csaba Szepesvári. Tighter risk certificates for neural networks. *The Journal of Machine Learning Research*, 22(1):10326–10365, 2021.

Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10426 LNCS, pages 97–117, feb 2017a. ISBN 9783319633862. doi: 10.1007/978-3-319-63387-9_5. URL http://arxiv.org/abs/1702.01135.

Ruediger Ehlers. Formal verification of piece-wise linear feed-forward neural networks. In *Automated Technology for Verification and Analysis: 15th International Symposium, ATVA 2017, Pune, India, October 3–6, 2017, Proceedings 15*, pages 269–286. Springer, 2017.

Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. *arXiv preprint arXiv:1711.07356*, 2017.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR, 2019.

John Langford and Rich Caruana. (not) bounding the true error. *Advances in Neural Information Processing Systems*, 14, 2001.

David A McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 164–170, 1999.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *CoRR*, abs/1706.06083, 2017.

Kai Y Xiao, Vincent Tjeng, Nur Muhammad Shafiullah, and Aleksander Madry. Training for faster adversarial robustness verification via inducing relu stability. *arXiv preprint arXiv:1809.03008*, 2018.

Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Towards Proving the Adversarial Robustness of Deep Neural Networks. *Electronic Proceedings in Theoretical Computer Science*, 257:19–26, sep 2017b. ISSN 2075-2180. doi: 10.4204/EPTCS.257.3. URL http://arxiv.org/abs/1709.02802http://dx.doi.org/10.4204/EPTCS.257.3http://arxiv.org/abs/1709.02802v1.

Lawrence D Brown, T Tony Cai, and Anirban DasGupta. Interval estimation for a binomial proportion. *Statistical science*, 16(2):101–133, 2001.

Ilya O Tolstikhin and Yevgeny Seldin. Pac-bayes-empirical-bernstein inequality. *Advances in Neural Information Processing Systems*, 26, 2013.

Niklas Thiemann, Christian Igel, Olivier Wintenberger, and Yevgeny Seldin. A strongly quasiconvex pac-bayesian bound. In *International Conference on Algorithmic Learning Theory*, pages 466–492. PMLR, 2017.

Omar Rivasplata, Ilja Kuzborskij, Csaba Szepesvári, and John Shawe-Taylor. Pac-bayes analysis beyond the usual bounds. *Advances in Neural Information Processing Systems*, 33:16833–16845, 2020.

Robert Price. A useful theorem for nonlinear devices having gaussian inputs. *IRE Transactions on Information Theory*, 4(2):69–72, 1958.

Martin Jankowiak and Fritz Obermeyer. Pathwise derivatives beyond the reparameterization trick. In *International conference on machine learning*, pages 2235–2244. PMLR, 2018.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in Neural Information Processing Systems*, 32, 2019.

Peter Bartlett, Dylan Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in Neural Information Processing Systems*, 30:6241–6250, 2017.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR, 2019.

Felix Biggs and Benjamin Guedj. Non-vacuous generalisation bounds for shallow neural networks. In *International Conference on Machine Learning*, pages 1963–1981. PMLR, 2022.

Greg Yang, Tony Duan, J Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*, pages 10693–10705. PMLR, 2020.

Idan Attias, Aryeh Kontorovich, and Yishay Mansour. Improved generalization bounds for robust learning. In *Algorithmic Learning Theory*, pages 162–183. PMLR, 2019.

Yue Xing, Qifan Song, and Guang Cheng. On the algorithmic stability of adversarial training. *Advances in Neural Information Processing Systems*, 34, 2021.

Jiancong Xiao, Yanbo Fan, Ruoyu Sun, and Zhi-Quan Luo. Adversarial Rademacher complexity of deep neural networks. 2021.

Yunwen Lei, Ürün Dogan, Ding-Xuan Zhou, and Marius Kloft. Data-dependent generalization bounds for multi-class classification. *IEEE Transactions on Information Theory*, 65(5):2995–3021, 2019.

Waleed Mustafa, Yunwen Lei, Antoine Ledent, and Marius Kloft. Fine-grained generalization analysis of structured output prediction. In *IJCAI 2021*, 2021.

Gintare Karolina Dziugaite and Daniel M Roy. Data-dependent pac-bayes priors via differential privacy. *Advances in neural information processing systems*, 31, 2018.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33: 2958–2969, 2020.

Zifan Wang, Nan Ding, Tomer Levinboim, Xi Chen, and Radu Soricut. Improving robust generalization by direct pac-bayesian bound minimization. *arXiv preprint arXiv:2211.12624*, 2022.

Paul Viallard, Eric Guillaume VIDOT, Amaury Habrard, and Emilie Morvant. A pac-bayes analysis of adversarial robustness. *Advances in Neural Information Processing Systems*, 34:14421–14433, 2021.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International conference on machine learning*, pages 5286–5295. PMLR, 2018.

Krishnamurthy Dvijotham, Robert Stanforth, Sven Gowal, Timothy A. Mann, and Pushmeet Kohli. A dual approach to scalable verification of deep networks. *CoRR*, abs/1803.06567, 2018.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. *CoRR*, abs/1801.09344, 2018.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pages 1802–1811. PMLR, 2019.

Pranjal Awasthi, George Yu, Chun-Sung Ferng, Andrew Tomkins, and Da-Cheng Juan. Adversarial robustness across representation spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7608–7616, 2021.

Alberto Bietti, Grégoire Mialon, Dexiong Chen, and Julien Mairal. A kernel perspective for regularizing deep neural networks. *arXiv preprint arXiv:1810.00363*, 2018.

Moustapha Cissé, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *ICML*, 2017.

Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *NIPS*, 2017.

Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Waleed Mustafa, Robert A Vandermeulen, and Marius Kloft. Input hessian regularization of neural networks. In *Workshop on "Beyond first-order methods in ML systems" at the 37th International Conference on Machine Learning*, 2020.

Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373*, 2018.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573*, 2019.

Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

Stéphane Boucheron, Gábor Lugosi, and Olivier Bousquet. Concentration inequalities. In *Summer school on machine learning*, pages 208–240. Springer, 2003.

Gintare Karolina Dziugaite, Kyle Hsu, Waseem Gharbieh, Gabriel Arpino, and Daniel Roy. On the role of data in pac-bayes bounds. In *International Conference on Artificial Intelligence and Statistics*, pages 604–612. PMLR, 2021.

A. Limitations and future work

One limitation of the proposed approach is its applicability only to stochastic DNNs. However, the work Biggs and Guedj (2022) provides a promising direction to extend the bounds to the deterministic DNN case. Additionally, the current bounds are specific to L_2 robustness, but we aim to generalize them to L_p robustness using techniques proposed by (Yang et al., 2020). Another limitation is the computational complexity, as our bounds require a large sample of both input noise and network parameters. However, in practice, we observed that the empirical average of errors converges relatively quickly, indicating the potential for exploring bounds of a faster convergence rate.

B. Related work

In this section, we briefly discuss the related work.

Adversarial generalization on deterministic DNNs Attias et al. (2019) utilized the VC dimension of the hypothesis class to derive adversarial generalization bounds. Some studies assume that the attacker's strategy is known in advance (Gao and Wang, 2021; Farnia et al., 2018), which is a strong assumption as real-world attackers can utilize a variety of attack techniques. Xing et al. (2021) employed algorithmic stability techniques to analyze the generalization of adversarial training. Several works have employed the Rademacher complexity to study the generalization of ℓ_p-additive-perturbation attacks (Khim and Loh, 2018; Yin et al., 2019; Awasthi et al., 2020; Xiao et al., 2021). Mustafa et al. (2022) utilized covering numbers arguments (Lei et al., 2019; Mustafa et al., 2021) to derive generalization bounds for general attacks beyond ℓ_p-additive attacks. These bounds, however, are numerically vacuous when applied to modern DNNs and datasets.

Non-vacuous bounds on stochastic DNNs Dziugaite and Roy (2017) were the first to compute non-vacuous bounds on stochastic DNNs using techniques from Langford and Caruana (2001). Dziugaite and Roy (2018) utilized differential privacy to train data-dependent priors. Pérez-Ortiz et al. (2021) performed an extensive study on optimizing several PAC-Bayes bounds and computed the state-of-the-art risk certificate in the natural settings. Biggs and Guedj (2022) brought non-vacuous PAC-Bayes bounds to deterministic shallow networks by a carefully designed architecture. These bounds, however, do not apply to adversarial settings.

Practical algorithms inspired by PAC-Bayes bounds Wu et al. (2020) draw insight from PAC-Bayes bounds to derive a scheme of adversarial training in which both the input and network weights are attacked. Wang et al. (2022) proposed minimizing an upper bound on a PAC-Bayes bound by using the trace of the Hessian of the empirical loss. Viallard et al. (2021) proposed to optimize a PAC-Bayes bound of a lower bound on the adversarial loss. They give tightness guarantees on this lower bound by a total variation between the random and adversarial noise distributions. This quantity, however, is very hard to estimate in practice. These methods, while showing practical success in the empirical evaluation of robustness, do not provide any guarantees on the population adversarial risk.

Adversarial verification methods Based on Mixed Integer Linear Programming (MILP) and Satisfiability Modulo Theories (SMT), exact verifiers (Katz et al., 2017a; Ehlers, 2017; Tjeng et al., 2017) are *complete*-verifiers, that is, they will report adversarial examples when they exist. MILP verifiers do not scale well to large networks (Cohen et al., 2019). Conservative verifiers (Wong and Kolter, 2018; Dvijotham et al., 2018; Raghunathan et al., 2018) use relaxation and duality techniques to verify a given input. These, however, tend to flag robust inputs as adversarial for expressive networks (Salman et al., 2019). Randomized smoothing (Cohen et al., 2019) are probabilistic verification methods that showed to scale to large DNNs and datasets. They transform a given classifier into a robust one by adding Gaussian noise to its inputs. The resulting classifier is provably robust to L_2 attacks. Yang et al. (2020) extends randomized smoothing to provide general guarantees to general L_p norms. These methods, however, concern test time verification, without any guarantees on their generalization properties.

Adversarial attacks Adversarial attacks are usually categorized as *white-box* (Carlini and Wagner, 2017) or *black-box* (Brendel et al., 2017), depending on the information available to the attacker. Most commonly, the attacker is constrained to alter the input by additive noise from an ℓ_p-ball. Recently, further (non-additive) attack models have been considered. In which the adversary manipulates the input by a non-linear transformation, either in the input space (e.g., rotation of an input image; Engstrom et al., 2019) or in a semantic representation space (e.g., in the frequency domain of an image; Awasthi et al., 2021).

Practical Defenses In response to such attacks, several defense mechanisms have been developed, for instance, based on regularizing the model's Lipschitz constant (Bietti et al., 2018; Cissé et al., 2017), input gradient (Hein and Andriushchenko, 2017; Ross and Doshi-Velez, 2018), or input Hessian (Mustafa et al., 2020) at training. The most widely used defense mechanism against adversarial attacks is *adversarial training* (Madry et al., 2017) and its variants (Kannan et al., 2018; Zhang et al., 2019). Its key idea is to replace clean training samples with their adversarial counterparts while maintaining their correct labels. Systematic studies have shown that the resulting models are robust and can withstand a large number of attacks (Athalye et al., 2018).

C. Details on the experimental setup

In this section, we provide more details on the experimental setup used in the main manuscript and, if not mentioned otherwise, in the appendix. In Algorithm 2, line 1, the prior is randomly initialized. However, following (Pérez-Ortiz et al., 2021), we found that *learning* the prior mean via ERM yields consistently stronger bounds. We use 70% of the training data to learn the prior for CIFAR-10 and 50% of the data for MNIST. The remaining data is utilized to learn the posterior and compute the certificates. During training, we fix the number of ϵ_i samples for smoothing to $M = 4$, use 10 steps for the PGD adversarial attack, a KL regularization for the posterior training with a factor of $\lambda_{KL} = 0.1$, a batch size of 256, SGD optimization with a momentum of 0.9 for learning the prior and 0.95 for learning the posterior, and train for 100 epochs both for the prior and posterior. We tested different learning rate schedulers and used a linear learning rate decrease of one-tenth at the 60th epoch for CIFAR-10 and every 20 epochs for MNIST. The smoothing variance for training (Algorithm 2, line 11) and computing the certificates (Algorithm 1, lines 4 and 7) is set to $\sigma_\epsilon = 0.5$. The final attacker capacity during training (Algorithm 2, line 13) is set to $R_{train} = 1.0$. We implemented a "warm-up" where we gradually increase R_{train} during the first 10 epochs of prior and posterior training until it matches the final attacker capacity. For the empirical certificate computation, we utilize 100 Monte Carlo samples for selection ($N_0 = 100$) and 10000 samples for estimation ($N = 10000$). We set $\delta = \delta' = \delta'' = 0.01$, $\alpha = 0.001$, and $p_{min} = 10^{-5}$ (see Theorem 3). 300 Monte Carlo samples (m=300) are used for the adversarial risk bound.

DNN architectures For MNIST, we use a simple CNN architecture (∼4.8M parameters) consisting of two convolutional layers with 32 and 64 filters, respectively. They are followed by two fully connected layers with 128 and 10 output neurons, respectively. We use ReLU activation and a dropout of 50% for each but the final layer. For CIFAR-10, we adopt a VGG-like (Simonyan and Zisserman, 2014) deep CNN (∼41M parameters) following Pérez-Ortiz et al. (2021). This architecture comprises 13 convolutional layers with up to 512 filters. The final prediction is computed using three fully connected layers with 1024, 512, and 10 output neurons, respectively. Similar to MNIST, ReLU activation, and dropout are applied to all layers except the final one. Additionally, we observed that incorporating Batch Normalization (BatchNorm) (Ioffe and Szegedy, 2015) facilitates faster prior learning. We exclude the learnable affine transformation that scale and shift the normalized data, and we freeze the running statistics after learning the prior to ensure that the network is fully parameterized by its weights.

Our model is sensitive to hyperparameters Due to computational constraints–training and computing the certificates on CIFAR-10 takes approximately 30 hours on 8 A100 GPUs–we performed a greedy grid search of the hyperparameters. For CIFAR-10, picking from learning rates in {5e-4, 1e-3, 5e-3, 1e-2, 5e-2}, we determined that 1e-3 produced the best results for the posterior, while 5e-3 was optimal for the prior. Overall the model displayed robustness to the choice of learning rate. However, we found that the stochastic network is sensitive to the choice of the prior covariance Σ_0. Any value above $\Sigma_0 = 0.015I$ makes the posterior training not converge, while any value below $\Sigma_0 = 0.01I$ showed no further improvement. Additionally, the prior tends to overfit, prompting us to search for an optimal dropout rate. Among the values tested (0.1, 0.2, 0.3, 0.5), a dropout rate of 0.2 proved to be the most effective. Our data preprocessing involved standardizing all data and applying simple data augmentation techniques, i.e., random resizing with padding of four and random horizontal flips. For MNIST, we searched within the same set of hyperparameters as for CIFAR-10. We picked 5e-2 as a learning rate for the posterior and 1e-3 for the prior. The prior Σ_0 was selected to be $0.03I$. We set the dropout parameter to 0.5.

D. Summary of the training proceadure

The following algorithm summarizes the training procedure derived in Section 2.3.

Algorithm 2 Adversarial PAC-Bayes

Input : Training set S, number of iteration T, batch size B

Output : Posterior distribution model Q

Randomly initialize μ_0, ρ_0

$\mu \leftarrow \mu_0$

$\rho \leftarrow \rho_0$

for $t \leftarrow 1 : T$ **do**

  $S_b \leftarrow$ Sample a batch from S with batch size B

  $V \leftarrow$ Sample from $\mathcal{N}(0, I)$

  $\Sigma_\rho \leftarrow \ln(1 + \exp(\rho))$

  $W \leftarrow \mu + \Sigma_\rho^{\frac{1}{2}} V$

  $\tilde{S}_b \leftarrow []$

  for $(x, y) \in S_b$ **do**

    $\{\epsilon_i\}_{i=1}^M \leftarrow$ Sample m i.i.d samples from $\mathcal{N}(0, \sigma^2 I)$

    Generate adversarial examples for samples (x, y) by solving:

    $\hat{x} = \arg\max_{\|\hat{x}-x\|_2 \leq R} \ell\left(\tilde{x}, y, \frac{1}{M}\sum_{t=1}^M h(W; \cdot + \epsilon_t)\right)$

    Append $\{(\hat{x} + \epsilon_i, y)\}_{i=0}^M$ to the adversarial training set $\tilde{S}_b$.

  end

  Update μ and ρ by SGD/ADAM with gradients $\nabla_\mu f_q(\mathcal{N}(\mu_0, \Sigma_0), \mathcal{N}(\mu, \Sigma_\rho), \widehat{L}(\delta_W, \tilde{S}_b, \ell))$ and $\nabla_\rho f_q(\mathcal{N}(\mu_0, \Sigma_0), \mathcal{N}(\mu, \Sigma_\rho), \widehat{L}(\delta_W, \tilde{S}_b, \ell))$, where δ_W is the Dirac measure.

end

return $\mathcal{N}(\mu, \rho)$

E. Proofs of theorems

In this section, we present the missing proofs in the main manuscript.

E.1. Proofs of section 3.2

In this section, we give the missing proofs for section 3.2 in the main manuscript. We use Theorem 1 in (Cohen et al., 2019) in the proof of Lemma 1.

Theorem E.1 (Theorem 1 in (Cohen et al., 2019))**.** *Let $h : \mathcal{X} \to \mathcal{Y}$ be a given function, ϵ be a random variable with a Gaussian distribution $\mathcal{N}(0, \sigma I)$, where I is the identity matrix. Define $g = \mathcal{T}_\sigma h$. Suppose $c_A \in \mathcal{Y}$, and let $p_A, p_B \in [0, 1]$ be defined such that*

$$\Pr(h(x + \epsilon) = c_A) \geq p_A \geq p_B \geq \max_{c \neq c_A} \Pr(h(x + \epsilon) = c).$$

Then we have

$$g(\tilde{x}) = c_A, \quad \textit{for all} \quad \|\tilde{x} - x\|_2 < R,$$

where

$$R = \frac{\sigma}{2}(\Phi^{-1}(p_A) - \Phi^{-1}(p_B)),$$

where Φ is the CDF of a standard normal distribution.

We are now ready to present the proof of Lemma 1 in the main manuscript.

Lemma E.2 (Restated)**.** *Let $S := \{(x_1, y_1), \ldots, (x_n, y_n)\} \subset \mathcal{X} \times \mathcal{Y}$ be a given dataset, $Q \in \mathcal{M}(\mathcal{W})$ be a probability measure on the set of weights $\mathcal{W}$. Further, let $\epsilon \sim \mathcal{N}(0, \sigma I)$ with some $\sigma > 0$, and let $p_A, p_B : \mathcal{W} \times \mathcal{X} \to [0, 1]$ such that, for all $(x, y) \in S$ and $W \in \mathcal{W}$,*

$$\Pr(h(W; x + \epsilon) = g(W; x)) \geq p_A(W, x) \geq p_B(W, x) \geq \max_{c \neq g(W;x)} \Pr(h(W; x + \epsilon) = c).$$

Then the following statements are true:

$$\widehat{L}(Q, S, \ell_{\text{adv}}) \leq \widehat{L}(Q, S, \tilde{\ell}), \tag{4}$$

$$\widehat{L}(\widehat{Q}, S, \ell_{\text{adv}}) \leq \widehat{L}(\widehat{Q}, S, \tilde{\ell}), \tag{5}$$

$$L(Q, \ell_{\text{adv}}) \leq L(Q, \tilde{\ell}), \tag{6}$$

where

$$\tilde{\ell}(x, y, g(W; \cdot)) := \begin{cases} \ell(x, y, g(W; \cdot)) & \textit{if} \quad \Phi^{-1}(p_A(W, x)) - \Phi^{-1}(p_B(W, x)) \geq \frac{2R}{\sigma} \\ 1 & \textit{otherwise}, \end{cases}$$

and Φ is the CDF of a standard normal distribution.

Proof. appendix We commence the proof by first observing the stability property of function g as delineated in Theorem E.1. Subsequently, through the careful construction of $\tilde{\ell}$, we demonstrate its capacity to provide an upper bound for the adversarial loss ℓ_{adv}.

Let $W \in \mathcal{W}$ and consider an arbitrary $(x, y) \in \mathcal{X} \times \mathcal{Y}$. By Theorem E.1 and the definitions of $p_A(W, x)$ and $p_B(W, x)$, we establish that

$$g(W; \tilde{x}) = g(W; x) \quad \text{for all} \quad \|\tilde{x} - x\|2 < R',$$

where

$$R' = \frac{\sigma}{2}(\Phi^{-1}(p_A(W, x)) - \Phi^{-1}(p_B(W, x))).$$

Thus, we can deduce that

$$\ell_{\text{adv}}(x, y, g(W; \cdot)) = \max_{\|\tilde{x} - x\|2 < R} \ell(\tilde{x}, y, g(W; \cdot)) = \ell(x, y, g(W; \cdot)),$$

whenever $R' \geq R$. In instances where $R' \leq R$, as per the definition of $\tilde{\ell}$, we ascertain that the loss assumes its maximum value of 1. Consequently, we arrive at the conclusion that

$$\ell_{\text{adv}}(x, y, g(W; \cdot)) \leq \tilde{\ell}(x, y, g(W; \cdot)).$$

Since W, x, and y are arbitrarily chosen, the above inequality holds for all $(x, y, W) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{W}$. By the monotonicity property of expectations, Equations (4) to (6) follow. Thus, we conclude the proof.

□

Next, we present the proof of Theorem 2.

Theorem E.3 (Restated)**.** *Let $Q^0 \in \mathcal{M}(\mathcal{W})$ be a prior probability measure. Then, with probability $1-\delta$ over the randomness of the training sample S, simultaneously for all $Q \in \mathcal{M}(\mathcal{W})$, we have*

$$L(Q, \ell_{\text{adv}}) \leq \text{KL}^{-1}\left(\widehat{L}(Q, S, \tilde{\ell}), \frac{\text{KL}(Q||Q^0) + \ln(\frac{2\sqrt{n}}{\delta})}{n}\right). \tag{7}$$

Proof. Recall that by Lemma E.2 we have $\widehat{L}(Q, S, \ell_{\text{adv}}) \leq \widehat{L}(Q, S, \tilde{\ell})$. Since $\text{KL}^{-1}(\cdot, \cdot)$ is monotonically increasing in the first argument, the result holds by Eq. (2) in the main manuscript.

□

E.2. Proofs of section 3.3

In this section, we present the missing proofs of Section 3.3 in the main manuscript. We first present Bernstein's inequality, a useful result that we utilize in the proof of Lemma 3.

Lemma E.4 (Bernstein's Inequality (Boucheron et al., 2003))**.** *Let $X_1, \ldots, \mathcal{X}_n$ be i.i.d real-valued random variables with $X_i \leq 1$, and $\mathbb{E}[X_i] = 0$, for $i \in [n]$. Further, let*

$$\frac{1}{n}\sum_{i=1}^{n} \text{Var}(X_i) \leq \nu.$$

Then with probability at least $1 - \delta$,

$$\frac{1}{n}\sum_{i=1}^{n} X_i \leq \sqrt{\frac{2\nu \ln(\frac{1}{\delta})}{n}} + \frac{\ln(\frac{1}{\delta})}{3n} \tag{8}$$

Next, we give the proof of Lemma 3.

Lemma E.5 (Restated)**.** *Let $S := \{(x_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ and $\mathcal{W} := \{W_i\}_{i=1}^m$ be a set of weights. Let* Alg1 *be the output of Algorithm 1, then with probability at least $1 - \delta$ over the randomness of the algorithm, we have*

$$\widehat{L}(\widehat{Q}, S, \tilde{\ell}) \leq \left(\text{Alg1} + \alpha + \sqrt{\frac{2\alpha(1-\alpha)\ln(\frac{1}{\delta})}{mn}} + \frac{\ln(\frac{1}{\delta})}{3mn}\right).$$

Proof. Firstly, let us consider $(x_i, y_i) \in S$ for $i \in [n]$ and $W_j \in \mathcal{W}$ where $j \in [m]$. In Algorithm 1, lines 5 and 6 provide an estimation for the predicted class c_A of $g(W_j; x_i)$. This estimation relies on the empirical estimate $\hat{g}(W_j; x_i)$ of the function $g(W_j; x_i)$. For the sake of simplicity, let us define $\tilde{\ell}_{ij} := \tilde{\ell}(x_i, y_i, g(W_j; \cdot))$ and $\hat{\ell}_{ij} := \tilde{\ell}(x_i, y_i, \hat{g}(W_j; \cdot))$. The objective of Algorithm 1 is to utilize $\hat{\ell}_{ij}$ as a substitute for computing $\tilde{\ell}_{ij}$. To ensure the validity of this substitution, it is imperative that $\tilde{\ell}_{ij} \leq \hat{\ell}_{ij}$. Thus, we proceed by quantifying the frequency with which this condition is not satisfied. Let $Z_{ij} := \mathbb{I}(\tilde{\ell}_{ij} > \hat{\ell}_{ij})$, where Z_{ij} is a random variable indicating whether the surrogate loss is smaller than the original loss. Consequently, we have the following inequality:

$$\widehat{L}(\widehat{Q}, S, \tilde{\ell}) := \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} \tilde{\ell}_{ij} \leq \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}(\hat{\ell}_{ij} + Z_{ij}). \tag{9}$$

We now proceed to establish an upper bound for $\frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m} Z_{ij}$. Let $p_{c_A} = \Pr(h(W; x+\epsilon) = c_A)$ and $\hat{p}_c$ be the lower $1-\alpha$ confidence interval estimate of p_c based on a finite sample of size N as computed in line 9. Consequently, we have:

$$\Pr(p_c < \hat{p}_c) \leq \alpha.$$

According to the definition of $\tilde{\ell}$, $\tilde{\ell}_{ij} \leq \hat{\ell}_{ij}$ only if $p_c < \hat{p}_c$. Thus, the variables Z_{ij} are independent Bernoulli random variables with a success probability less than α, a mean $\mathbb{E}[Z_{ij}] \leq \alpha$, and a variance $\mathrm{Var}(Z_{ij}) \leq \alpha(1-\alpha)$. Let $Z = \frac{1}{mn}\sum_i\sum_j(Z_{ij} - \mathbb{E}[Z_{ij}])$. Then we know that Z is a random variable bounded by 1 with zero mean and a variance less than $\alpha(1-\alpha)$.

By applying Bernstein's inequality (Lemma E.4), we obtain, with a probability of at least $1-\delta$, the following inequality:

$$\begin{aligned} Z &\leq \frac{\sqrt{2\alpha(1-\alpha)\ln(\frac{1}{\delta})}}{\sqrt{mn}} + \frac{\ln(\frac{1}{\delta})}{3mn}, \\ \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m} Z_{ij} &\leq \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbb{E}[Z_{ij}] + \frac{\sqrt{2\alpha(1-\alpha)\ln(\frac{1}{\delta})}}{\sqrt{mn}} + \frac{\ln(\frac{1}{\delta})}{3mn}, \\ &\leq \alpha + \frac{\sqrt{2\alpha(1-\alpha)\ln(\frac{1}{\delta})}}{\sqrt{mn}} + \frac{\ln(\frac{1}{\delta})}{3mn}. \end{aligned} \tag{10}$$

By noting that $\mathrm{Alg1} = \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}\hat{\ell}_{ij}$ and combining Eq.(9) and Eq.(10), we arrive at the final result.

□

Finally, we present the proof of Theorem 3.

Theorem E.6 (Restated)**.** *Let $Q^0 \in \mathcal{M}(\mathcal{W})$ be prior distribution. Then with probability at least $1-\delta-\delta'-\delta''$, simultaneously for $Q \in \mathcal{M}(\mathcal{W})$, the adversarial risk $L(Q, \ell_{\mathrm{adv}})$ is upper-bounded by*

$$\mathrm{KL}^{-1}\left(\mathrm{KL}^{-1}\left(\left(\mathrm{Alg1} + \alpha + \sqrt{\frac{2\alpha(1-\alpha)\ln(\frac{1}{\delta''})}{mn}} + \frac{\ln(\frac{1}{\delta''})}{3mn}\right), \frac{\ln(\frac{2}{\delta'})}{m}\right), \frac{\mathrm{KL}(Q||Q^0) + \ln(\frac{2\sqrt{n}}{\delta})}{n}\right).$$

Proof. The proof follows by combining Lemma 3 with Eq.(4) and Theorem 2 in the main manuscript with the fact that KL^{-1} is monotonic in the first argument.

□

E.3. Helper Lemmas

Theorem E.7 (Classical PAC-Bayes bound (Langford and Caruana, 2001; McAllester, 1999))**.** *Let $Q^0 \in \mathcal{M}(\mathcal{W})$ be a prior probability measure on $\mathcal{W}$. For any $\delta \in (0,1)$, with probability at least $1-\delta$ over the randomness of the training sample S, simultaneously for all distributions $Q \in \mathcal{M}(\mathcal{W})$,*

$$\mathrm{KL}(\widehat{L}(Q,S,\ell), L(Q,\ell)) \leq \frac{\mathrm{KL}(Q||Q^0) + \ln(\frac{2\sqrt{n}}{\delta})}{n}.$$

Lemma E.8 ((Langford and Caruana, 2001))**.** *Let $t_1, \ldots, t_m \sim \mathcal{B}(\lambda)$ be independent Bernoulli variables with $\lambda \in [0,1]$. Then with probability at least $1-\delta$,*

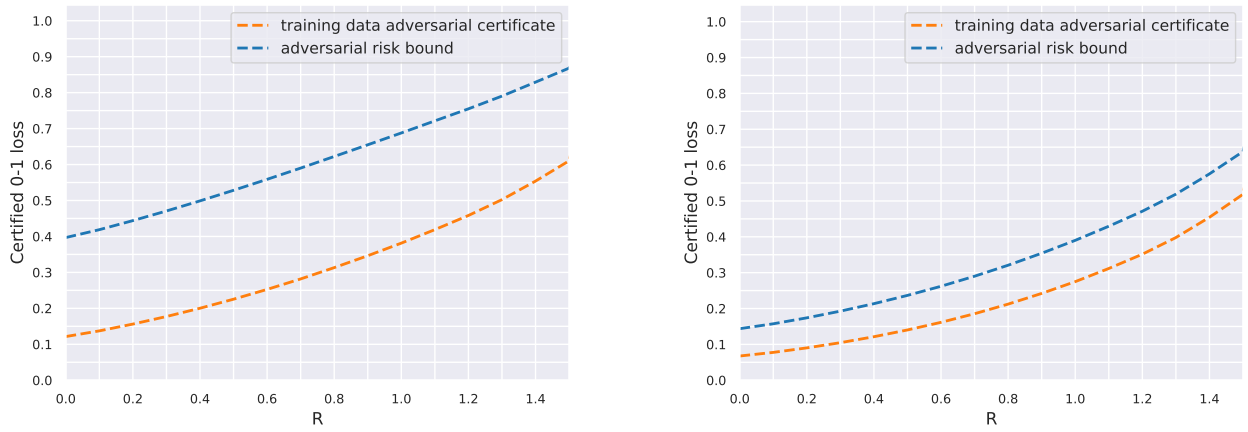
$$\mathrm{KL}\left(\frac{1}{m}\sum_{j=1}^{m} t_j \middle\| \lambda\right) \leq \frac{\ln(\frac{2}{\delta})}{n}.$$

F. Additional experiments

In the following sections, we provide ablation studies that focus on sensitive hyperparameters and highlight the usefulness of smoothing and adversarial learning. First, we demonstrate the effect of learning the prior mean via ERM.

F.1. Learning the prior yields stronger certificates

In this section, we investigate the impact of training a data-dependent prior. On MNIST, our findings align with the expectations, as learning the prior significantly reduces the generalization gap, as demonstrated in Figure 2. Moreover, our observations reveal notable enhancements in the empirical training certificates when the prior mean is learned. However, when considering CIFAR-10, training with data-independent priors proved challenging. The posterior did not outperform random guessing. These results strongly emphasize the importance of learning the prior to obtaining non-vacuous certificates. This was also highlighted in the literature (Dziugaite and Roy, 2018; Dziugaite et al., 2021; Pérez-Ortiz et al., 2021).



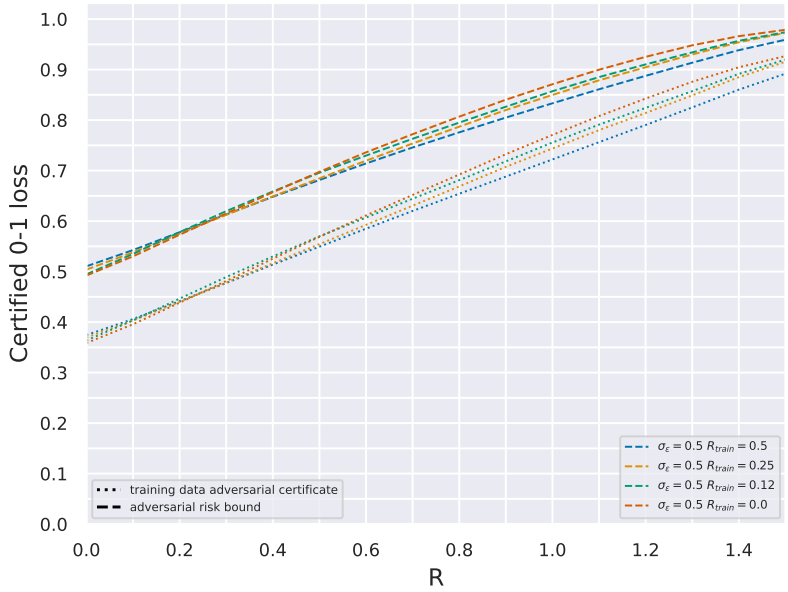
(a) Without Prior Training

(b) With Prior Training

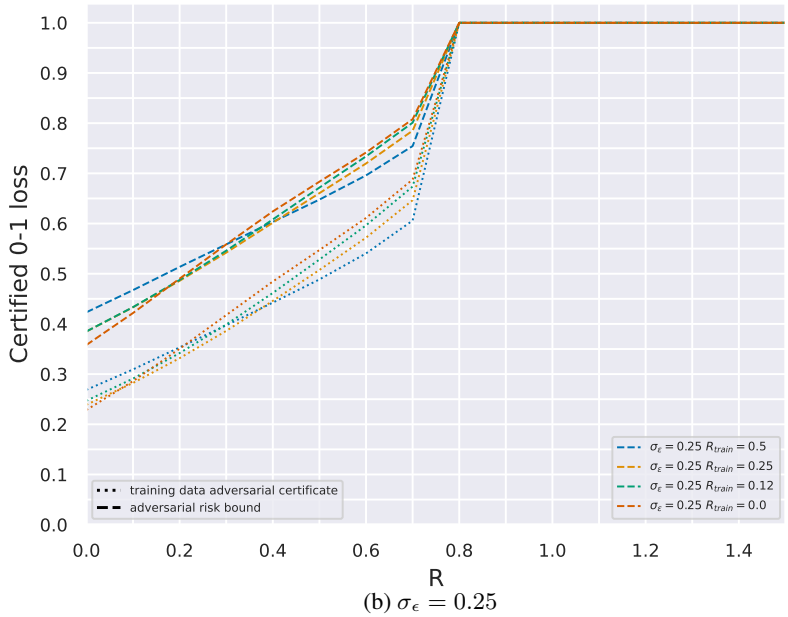
Figure 2. Effect of data-dependent prior. Shown are adversarial risk bounds for a DNN trained with and without prior training on MNIST. Each subfigure plots the 0-1 loss over increasing attacker capacities (i.e., R). The blue curve represents the risk bound, while the orange curve represents the empirical certified robustness of the training data.

F.2. Smoothing and adversarial training improves model robustness

In this experiment, we investigate the effects of varying the random smoothing and adversarial attack hyperparameters in the training algorithm. In particular, we vary the smoothing variance (Algorithm 1, lines 4 and 7; Algorithm 2, line 11) and the attacker capacity for adversarial learning. As shown in Figure 3, we observe that models are more robust when confronted with stronger adversarial attacks during training but achieve inferior bounds in a weak adversarial setup ($R < 0.2$). Decreasing the variance of smoothing has a similar effect. While it improves the bounds for weaker adversarial setups, it makes the bounds collapse at $R \geq 0.7$.



(a) $\sigma_\epsilon = 0.5$



(b) $\sigma_\epsilon = 0.25$

Figure 3. Effect of varying the adversarial capacity and smoothing parameter during training. Each color corresponds to a network trained with varying attacker capacities (i.e., R_{train}). Each subfigure depicts the 0-1 loss as the attacker's capacity during inference (R) increases. The dashed curve represents the risk bound, while the dotted curve represents the empirically certified robustness of the training data.

F.3. Adversarial training improves model robustness

We consider two settings for our experiments. Firstly, we report results for adversarial training as outlined in Algorithm 2. In the second setting, we omit the adversarial training step, i.e. omitting line 27 in Algorithm 2. Adversarial training imposes a harder constraint on the models. Consequently, we anticipated that while it would enhance the empirical certificate (orange line), it could potentially widen the generalization gap, thus leading to inferior risk certificates. Surprisingly, adversarial training increased the model robustness without significantly affecting the generalization gap. This observation suggests that the KL regularization is not at odds with adversarial training when applied to smoothed classifiers.

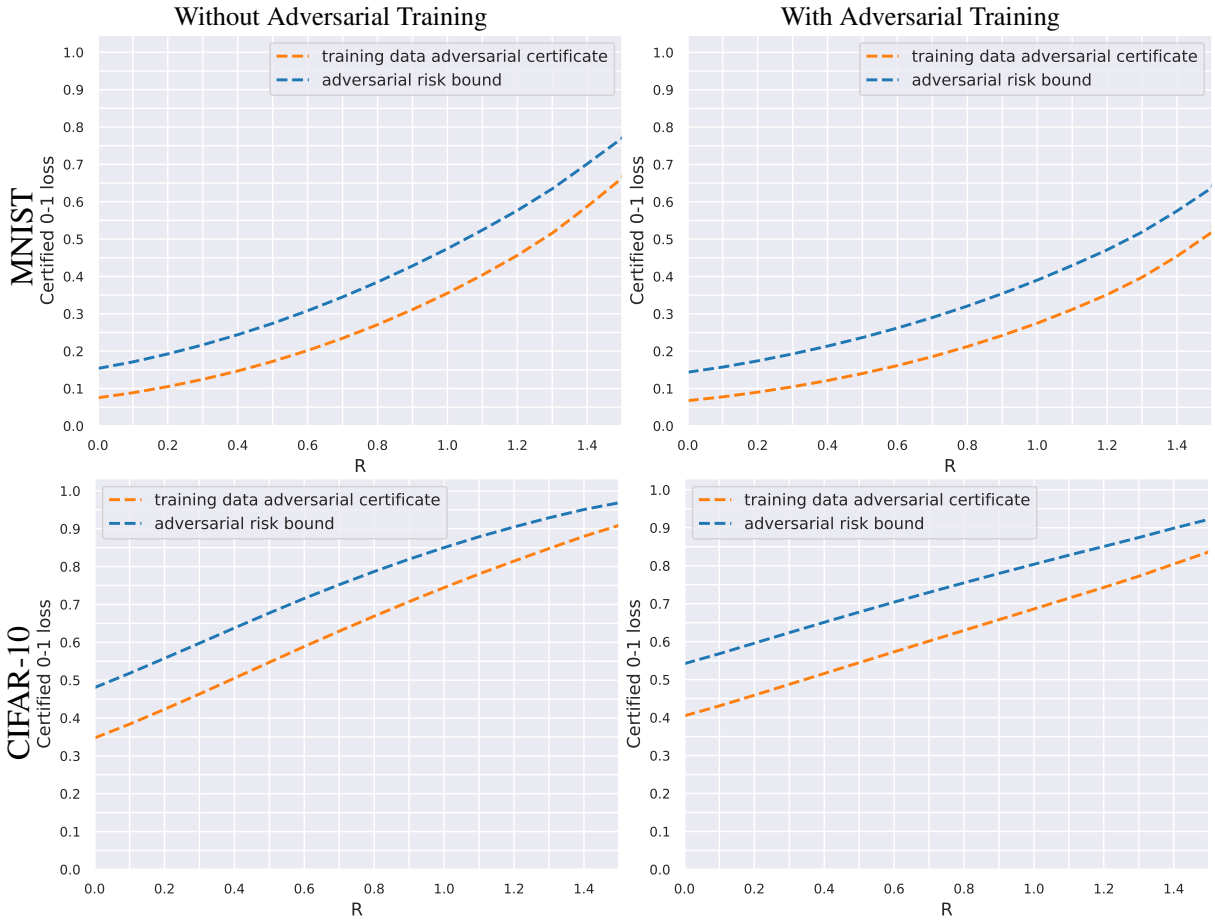
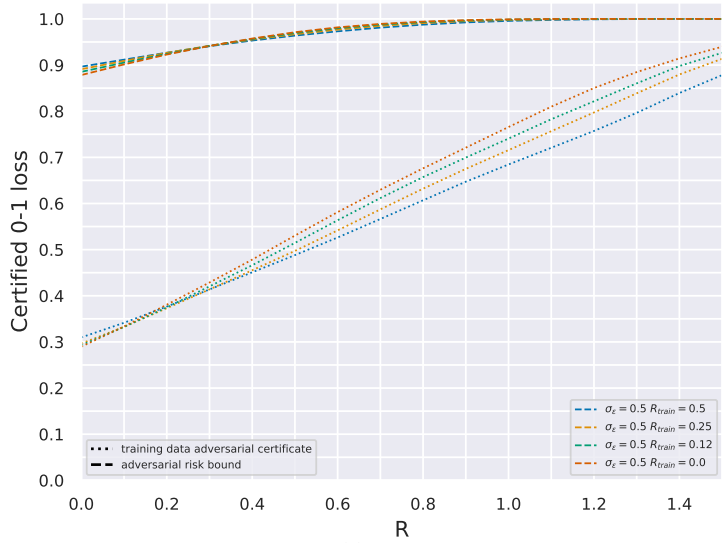


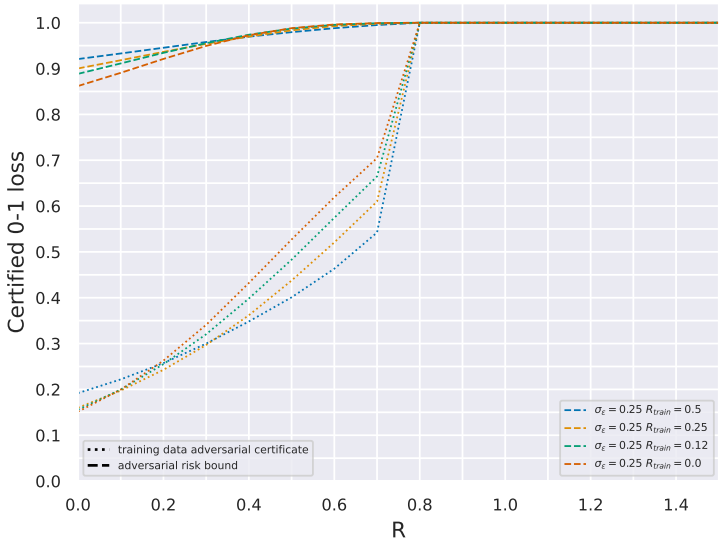
Figure 4. The figure shows adversarial risk bounds for a DNN trained with (left) and without (right) adversarial training on MNIST (top) and CIFAR-10 (bottom). Each subfigure depicts the 0-1 loss over increasing attacker capacities (i.e., R). The blue curve represents the risk bound, while the orange curve represents the empirical certified robustness of the training data.

F.4. The generalization gap explodes without KL regularization

In this section, we shift our focus toward investigating the impact of Kullback-Leibler (KL) divergence regularization. To assess its influence on the production of certifiable models, we conduct experiments without KL-regularization, specifically setting λ_{KL} to 0.



(a) $\sigma_\epsilon = 0.5$



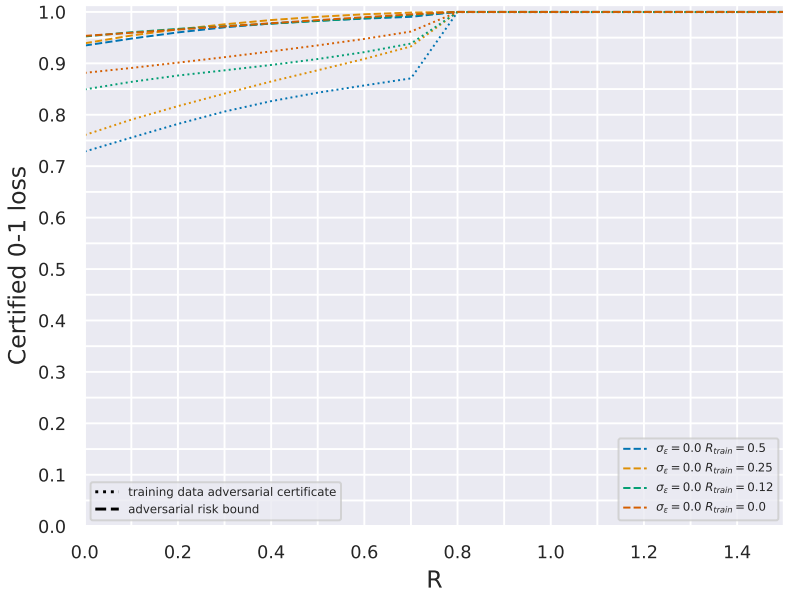
(b) $\sigma_\epsilon = 0.25$

Figure 5. Effect of omitting KL regularization. This figure shows adversarial risk bounds for a DNN trained on CIFAR-10 with two different smoothing variances σ_ϵ. Each color corresponds to a network trained with varying attacker capacities (i.e., R_{train}). Each subfigure depicts the 0-1 loss as the attacker's capacity during inference (R) increases. The dashed curve represents the risk bound, while the dotted curve represents the empirically certified robustness of the training data.

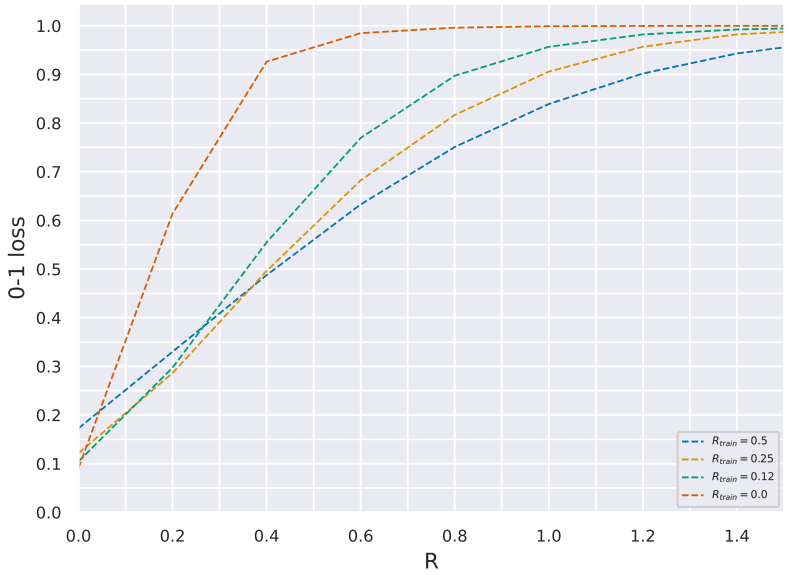
As expected, the absence of KL-regularization leads to an improvement in the empirical training error. However, the resulting adversarial risk certificates are found to be vacuous (see Figure 5). This observation underscores the significance of optimizing the PAC-Bayes bound to computing non-vacuous generalization bounds.

F.5. No smoothing deteriorates robustness significantly

In this section, we investigate the effect of smoothing in an extreme scenario. We completely remove smoothing during training; i.e., set $\sigma_\epsilon = 0$ in line 11 of Algorithm 2.



(a) smoothing variance $\sigma_{epsilon} = 0.25$ for computing the certificate



(b) empirical performance under PGD attack

Figure 6. Effect of adversarial smoothing. This figure shows (a) adversarial risk bounds and (b) empirical performance under PGD attack for a DNN trained on CIFAR-10 with no smoothing (i.e., $\sigma_\epsilon = 0$). Each color corresponds to a network trained with varying attacker capacities (i.e., R_{train}). The figure depicts the 0-1 loss as the attacker capacity during inference (R) increases. In (a), the dashed curve represents the risk bound, while the dotted curve represents the empirically certified robustness of the training data.

We first investigate the significance of training a smoothed classifier vs. smoothing a naturally or adversarially trained one. To this end, we compute certificates to classifiers that are trained under natural or adversarial conditions and are subsequently smoothed by a smoothing parameter $\sigma_\epsilon = 0.25$. Figure 6a shows the adversarial risk bounds and training data certificates for different adversarial training settings. Interestingly, the figures demonstrate that naturally trained classifiers fail to provide reasonable robustness certificates even after smoothing, emphasizing the significance of using randomized smoothing during training.

While the aforementioned experiment highlights the importance of randomized smoothing during training, it raises an intriguing question: Are the trained models robust, despite the absence of certifiability through smoothing techniques? To address that question, we subject these models to PGD attacks, thereby establishing a lower bound on empirical adversarial risk. Evidently in Figure 6b, the model's robust performance decreased significantly even for adversarially-trained models. This provides evidence that it is challenging to obtain robustness for a set of models with large probability as measured by the posterior, underscoring the effectiveness of randomized smoothing in obtaining such a set of robust models.

F.6. Training the prior is prone to overfitting

In our early experiments without dropout, we noticed that, while the prior often achieves a training error of close to 0%, the posterior fails to follow. It is stuck at around 50% training error. Even though we use the common data augmentation, we hypothesize that the prior overfits on the training data. Figure 7 shows adversarial risk bounds for varying dropout rates when training the prior mean via ERM. We use $R_{train} = 0.5$. It can be seen that the model achieves the best bounds with a dropout rate of roughly 20%. Without dropout, the model seems to overfit and produces far inferior bounds. On the other hand, larger dropout rates seem to cause the model to underfit as the bounds deteriorate.

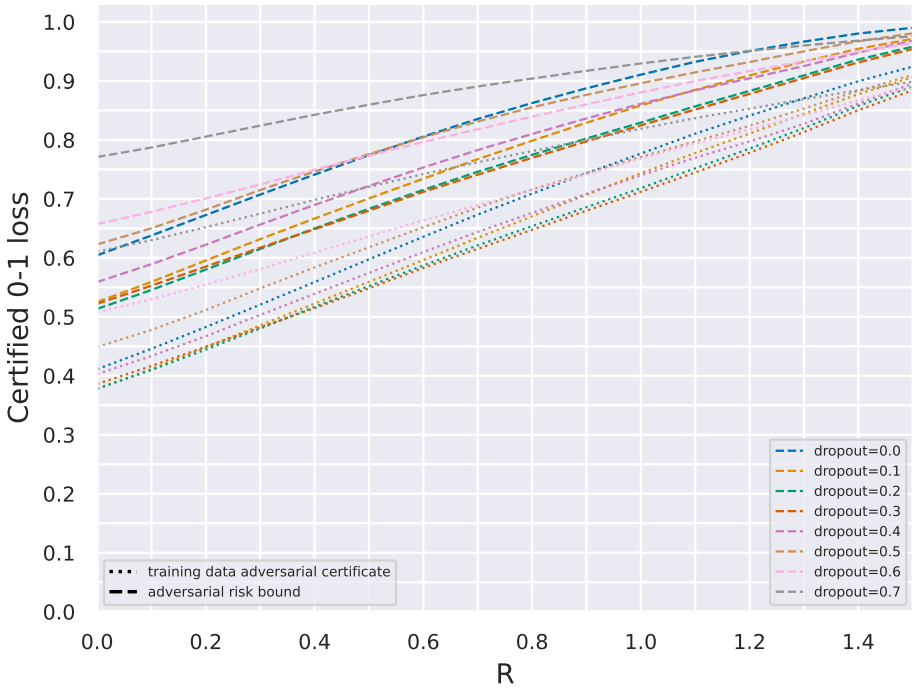


Figure 7. Effect of dropout in prior training. This figure shows adversarial risk bounds for a DNN trained on CIFAR-10. Each color corresponds to a network trained with varying dropout rates during prior training. The figure depicts the 0-1 loss as the attacker capacity during inference (R) increases. The dashed curve represents the risk bound, while the dotted curve represents the empirically certified robustness of the training data.

F.7. Training the posterior is sensitive to the prior variance

As shown in Appendix F.1, learning the prior mean via ERM improves the model performance significantly. This prompts us to investigate the impact of the prior covariance on the posterior performance. Figure 8 shows adversarial risk bounds for varying prior covariances Σ_0 and fixed $R_{train} = 0.5$. We find that the model is sensitive to this hyperparameter. Increasing Σ_0 above 0.015 deteriorates the bounds drastically. Decreasing Σ_0 below 0.01 seems to have no significant effect.

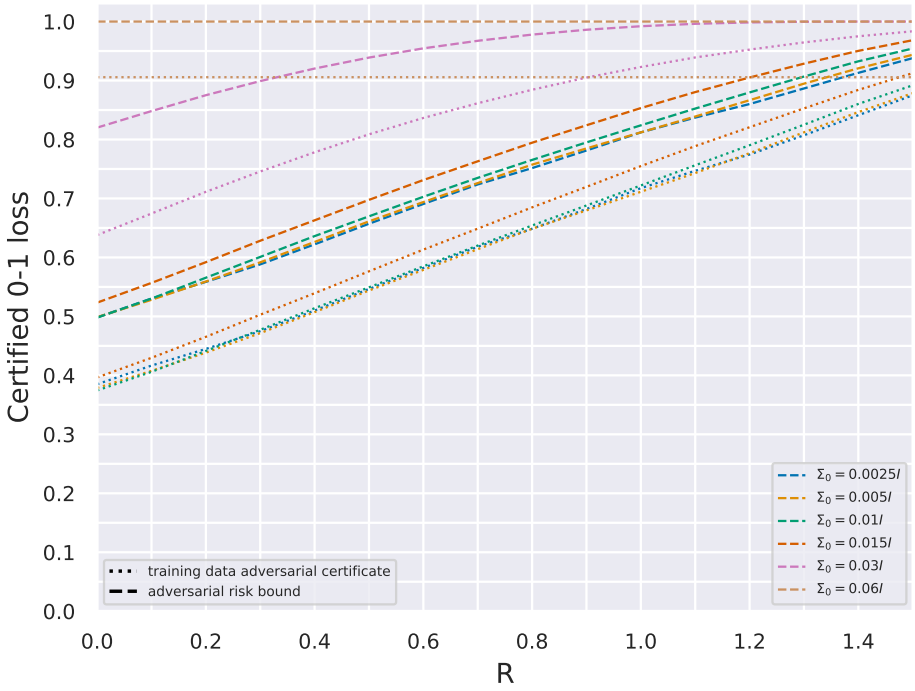


Figure 8. Effect of prior variance. This figure shows adversarial risk bounds for a DNN trained on CIFAR-10. Each color corresponds to a network trained with varying prior covariances Σ_0. The figure depicts the 0-1 loss as the attacker capacity during inference (R) increases. The dashed curve represents the risk bound, while the dotted curve represents the empirically certified robustness of the training data.

G. Approximating KL^{-1}

In this section, for completeness, we present the numerical algorithm to approximate the inverse Kullback-Leibler divergence KL^{-1} (Dziugaite and Roy, 2017). In order to approximate $\mathrm{KL}^{-1}(p, c) = \sup\{q \in [0, 1] \colon \mathrm{KL}(p||q) \leq c\}$, we leverage Newton's method for finding the roots of the function $f(q; p, c) = \mathrm{KL}(p||q) - c$. This approach is effective since the proximity of q to the supremum in the definition of KL^{-1} corresponds to the closeness of f to zero at q. Newton's method utilizes iterative updates of the form $q_{n+1} = q_n - f(q_n)(\frac{df}{dq}\Big|_{q=qn})^{-1}$ to converge towards a root of f. For Bernoulli distributions, the Kullback-Leibler divergence is expressed as $\mathrm{KL}(p, q) = p \ln \frac{p}{q} + (1 - p) \ln \frac{1-p}{1-q}$, and its derivative with respect to q is $\frac{\partial \mathrm{KL}}{\partial q} = \frac{1-p}{1-q} - \frac{p}{q}$. Thus, we can utilize updates in the following form:

$$q_{n+1} = q_n - \frac{p \ln \frac{p}{q_n} + (1 - p) \ln \frac{1-p}{1-q_n} - c}{\frac{1-p}{1-q_n} - \frac{p}{q_n}}$$

to approximate $\mathrm{KL}^{-1}(p, c)$.

To initialize the process (setting q_0), we employ the simple upper bound $\mathrm{KL}^{-1}(p, c) \leq p + \sqrt{\frac{c}{2}}$ (Dziugaite and Roy, 2017) and ensure that the initial estimate falls within the domain $[0, 1]$ by setting:

$$q_0 = \min\left\{1, p + \sqrt{\frac{c}{2}}\right\}.$$