Non-vacuous Generalization Bounds for Adversarial Risk in Stochastic Neural Networks

Waleed Mustafa[1*] **Philipp Liznerski**[1*] **Antoine Ledent**[2†]

Dennis Wagner[1] **Puyu Wang**[3] **Marius Kloft**[1]

[1] RPTU, Kaiserslautern, Germany; [2] Singapore Management University, Singapore
[3] Hong Kong Baptist University, Hong Kong, China
[*] Equal contribution; [†] Correspondence to: aledent@smu.edu.sg

Abstract

Adversarial examples are manipulated samples used to deceive machine learning models, posing a serious threat in safety-critical applications. Existing safety certificates for machine learning models are limited to individual input examples, failing to capture generalization to unseen data. To address this limitation, we propose novel generalization bounds based on the PAC-Bayesian and randomized smoothing frameworks, providing certificates that predict the model's performance and robustness on unseen test samples based solely on the training data. We present an effective procedure to train and compute the first non-vacuous generalization bounds for neural networks in adversarial settings. Experimental results on the widely recognized MNIST and CIFAR-10 datasets demonstrate the efficacy of our approach, yielding the first robust risk certificates for stochastic convolutional neural networks under the L_2 threat model. Our method offers valuable tools for evaluating model susceptibility to real-world adversarial risks. Our code is publicly available[1].

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; Liznerski et al., 2024), particularly adversarial examples (Szegedy et al., 2013). The ability of DNNs to generalize to unseen data has been the subject of extensive research in recent years (Bartlett et al., 2017; Kawaguchi et al., 2017; Wei and Ma, 2019; Cao and Gu, 2019; Nagarajan and Kolter, 2019; Dziugaite and Roy, 2017; Pérez-Ortiz et al., 2021), but how to simultaneously achieve robustness against adversarial attacks is still an important open question (Bai et al., 2021). Past attempts to quantify the robustness of trained models (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 (i.e., > 1.0)[2], thus, while providing valuable theoretical insights, are of little practical use.

The pioneering work of Dziugaite and Roy (2017) computed the first non-vacuous bounds (i.e., < 1.0) on the population risk of 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.

In this paper, we develop the first non-vacuous population risk bounds applicable to models deployed in an adversarial setting.

Our novel approach involves the incorporation of randomized smoothing Cohen et al. (2019) to transform

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

[1] https://github.com/waleedamustafa/nonadvgenaistats

[2] See for example Figure 2, in Graf et al. (2022).

the class of DNNs into a class of smoothed functions. Members of such a class admit an efficient approach to empirical robustness evaluation (i.e., robust performance evaluation on the training set). Subsequently, we apply PAC-Bayes analysis to effectively bound the adversarial population risk (i.e., robust performance on unseen data) of stochastic and smoothed DNNs in terms of empirical robust performance. Our approach offers a method to evaluate the robustness of models using only training data, eliminating the requirement for a separate test set while also providing a mechanism to guide the training of such models.

2 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 function 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. Xiao et al. (2023) considered a PAC-Bayes approach to prove bound for adversarial loss in terms of the product of spectral norms of weight matrices. Again, these bounds are vacuous for modern setups.

Non-vacuous Bounds On Stochastic DNNs Dziugaite and Roy (2017) were the first to compute non-vacuous bounds on stochastic DNNs. They employed the techniques outlined in Langford and Caruana (2001), commonly applied in the analysis of classical Bayesian methods (Wenzel et al., 2017). 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 specific 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., 2017; 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 are shown 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.

3 NON-VACUOUS GENERALIZATION BOUNDS IN AN ADVERSARIAL SETTING

Here, we present our approach. We start by introducing the notation and problem setting in Section 3.1. Next, in Section 3.2, we present the main bounds, focusing on an idealized setting to establish a strong theoretical foundation. Section 3.3 is dedicated to deriving practical algorithms to compute these bounds, enabling their application in real-world scenarios. In Section 3.4 we describe the training process for models that exhibit non-vacuous bounds, ensuring their practical relevance.

3.1 Problem Setting

We start by introducing the notation and problem settings. 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 0-1 loss $\ell(x, y, h(W; \cdot)) = \mathbb{I}(h(W; x) = y)$. 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 not exceeding $R > 0$ while incurring a maximal loss. In other words, the adversary seeks to solve the optimization problem $\tilde{x} = \arg\max_{\tilde{x}:\|x-\tilde{x}\|_2<R} \ell(\tilde{x}, y, h(W; \cdot))$, where $\ell_{\text{adv}}(x, y, h(W; \cdot)) := \max_{\tilde{x}:\|x-\tilde{x}\|_2<R} \ell(\tilde{x}, y, h(W; \cdot))$ is the adversarial loss.

We are interested in bounding the adversarial risk associated with the stochastic prediction Q:

$$L(Q, \ell_{\text{adv}}) := \mathbb{E}_{W\sim Q}\left[\mathbb{E}_{(x,y)\sim P}\ell_{\text{adv}}(x, y, h(W; \cdot))\right].$$

However, we only have access to the empirical risk

$$\widehat{L}(Q, S, \ell_{\text{adv}}) := \mathbb{E}_{W\sim Q}[\frac{1}{n}\sum_{i=1}^{n}\ell_{\text{adv}}(x_i, y_i, h(W; \cdot))],$$

where $S := \{(x_i, y_i) \sim P \mid i \in [n]\}$ is an i.i.d. training sample. Here, $[n] = \{1, \dots, n\}$.

Our main goal is derive an upper bound on $L(Q, \ell_{\text{adv}})$ in terms of $\widehat{L}(Q, S, \ell_{\text{adv}})$ and the properties of Q that is less than the trivial 1.0

In our approach, we employ Randomized Smoothing (RS) (Cohen et al., 2019). RS transforms a given classifier $h(W; \cdot)$ into a provably robust classifier $g(W; \cdot)$ by applying the operator $\mathcal{T}_{\sigma_x}$ defined by

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

for $x \in \mathcal{X}$, $W \in \mathcal{W}$. Here, $\epsilon \sim \mathcal{N}(0, \sigma_x^2 I)$ represents a random noise vector and $\sigma_x > 0$ 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).

3.2 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 to clearly describe the idea. In Section 3.3, we extend the results to compute practical certificates.

Our approach entails substituting the loss ℓ by an extended robust version $\tilde{\ell}$ which we define below.

Definition 1 (Robust Loss of a Smoothed Classifier)**.** We define the robust loss $\tilde{\ell}$ as $\tilde{\ell}(x, y, g(W; \cdot)) :=$

$$\begin{cases} \ell(x, y, g(W; \cdot)) & \text{if } \Phi^{-1}(\overline{p}_{W,x}) - \Phi^{-1}(\underline{p}_{W,x}) \geq \frac{2R}{\sigma} \\ 1 & \text{otherwise,} \end{cases}$$

where $\overline{p}_{W,x}$ and $\underline{p}_{W,x}$ are defined such that,

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

$P_g(W; x) := \Pr(h(W; x + \epsilon) = g(W; x))$, $P_{\neg g}(W; x) := \max_{c\neq g(W;x)} \Pr(h(W; x + \epsilon) = c)$, and Φ is the CDF of the standard normal distribution.

The definition of the robust loss $\tilde{\ell}$ extends the 0-1 loss ℓ by incorporating the uncertainty level of the smooth classifier g into the loss. To see this, 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 (i.e., g predicts with high confidence), the robustness of $g(W; x)$ also improves. On the other hand, 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 presence of an adversarial example for $g(W; x)$. Thus, the loss $\tilde{\ell}$ inherently provides a measure of robustness for the smoothed classifier $g(W; x)$ based on the probability bounds $\underline{p}_{W,x}$ and $\overline{p}_{W,x}$.

It is noteworthy that the exact computation of $\overline{p}_{W,x}$ and $\underline{p}_{W,x}$ within $\tilde{\ell}$ can be intractable, however, we can efficiently estimate them using sampling-based techniques (Cohen et al., 2019). Further details on the computational methods can be found in Section 3.3.

We are now ready to present the main result of this section. Theorem 1 introduces a bound on the adversarial risk of smoothed classifiers by their empirical risk induced by the robust loss $\tilde{\ell}$ and the Kullback-Leibler (KL) divergence between a prior and a posterior distributions of the model parameters. Recall that the KL divergence between two distributions Q and Q' on $\mathcal{W}$ is defined as $\text{KL}(Q||Q') := \int_{\mathcal{W}} \ln(\frac{dQ}{dQ'})dQ$

This work	Xiao et al. (2023)	Mustafa et al. (2022)
$\widetilde{O}\left(\sqrt{\frac{\frac{1}{\sigma_w^2}\sum_{l=1}^{L}\\|W_l^0 - W_l\\|_2^2}{n}}\right)$	$\widetilde{O}\left(\sqrt{\frac{(B+R)^2 L^2 \omega \prod_{l=1}^{L}\\|W_l\\|_\sigma^2 \sum_{l=1}^{L}\frac{\\|W_l - W_l^0\\|_2^2}{\\|W_l\\|_\sigma^2}}{n}}\right)$	$\widetilde{O}\left(\sqrt{\frac{(B+R)^2 L^2 d \prod_{l=1}^{L}\\|W_l\\|_\sigma^2 \sum_{l=1}^{L}\frac{\\|W_l - W_l^0\\|_{2,1}^2}{\\|W_l\\|_\sigma^2}}{n}}\right)$

Table 1: Norm-based generalization bounds.

if the Radon-Nikodym derivative $\frac{dQ}{dQ'}$ exists and ∞ otherwise. When Q and Q' are Bernoulli distributions with parameters $q, q' \in [0, 1]$, we use the shorthand $\mathrm{KL}(q, q') := \mathrm{KL}(Q||Q')$.

Theorem 1 (Main Result)**.** *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*

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

where $\mathrm{KL}^{-1}(q, c) := \sup\{p \in [0, 1] : \mathrm{KL}(q, p) \leq c\}$.

The theorem presents an upper bound on the adversarial risk $L(Q, \ell_{\text{adv}})$ in terms of the empirical risk $\widehat{L}(Q, S, \tilde{\ell})$ and the complexity term of the posterior $\mathrm{KL}(Q\|Q^0)$. Note that the empirical risk is defined in terms of the robust loss function $\tilde{\ell}$ rather than ℓ_{adv}. While computing ℓ_{adv} is intractable for many practical models including DNNs, due to the maximum operator, $\tilde{\ell}$ admits a tractable estimation via sampling based techniques (see Section 3.3 for more details). Therefore, Theorem 1 implies an efficient method to evaluate the adversarial risks of classifiers via the bound (1).

We briefly sketch the proof of Theorem 1 here (Details can be found in Appendix B). First, we leverage the careful definition of $\tilde{\ell}$ to establish an upper bound for the adversarial loss ℓ_{adv}. Indeed, when g classifies with a low margin (i.e., $\Phi^{-1}(P_g(W; x)) - \Phi^{-1}(P_{\neg g}(W; x)) < \frac{2R}{\sigma}$), then $\tilde{\ell}$ yields the maximum loss 1, therefore $\ell_{\text{adv}} \leq \tilde{\ell}$. Conversely, when g classifies with high margin (i.e., $\Phi^{-1}(P_g(W; x)) - \Phi^{-1}(P_{\neg g}(W; x)) \geq \frac{2R}{\sigma}$), then ℓ_{adv} coincides with the non-adversarial loss ℓ and by extension $\tilde{\ell}$, that is $\ell_{\text{adv}} = \tilde{\ell}$. Indeed, this the case since, as shown in Salman et al. (2019), the function $\Phi^{-1}(P_g(W; x))$ is $\frac{1}{\sigma}$-Lipschitz in x. Therefore, we establish $L(Q, \ell_{\text{adv}}) \leq L(Q, \tilde{\ell})$. Next, we bound $L(Q, \tilde{\ell})$ via the PAC-Bayes analysis (Langford and Caruana, 2001) to get the final result.

Comparison To Related Work Existing generalization bounds of adversarial learning largely focus on deterministic networks, rendering direct comparisons challenging. A relaxed comparison, however, provides insights into the mechanisms of non-vacuous bounds. To facilitate the comparison, we consider deterministic and stochastic DNNs, that is $h(W; x) := \mathrm{Softmax}(W_L\sigma(\ldots\sigma(W_1\sigma(W_0x))))$. Let W_l^0 denote weights at initialization and W_l for trained networks. The posterior distribution of the stochastic DNN is defined as $\widetilde{W_l} \sim \mathcal{N}(W_l, \sigma_w^2 I)$[3] and the prior is defined as $\widetilde{W_l} \sim \mathcal{N}(W_l^0, \sigma_w^2 I)$. Further assume that $\|x\|_2 \leq B$, and the maximum number of neurons per layer is ω. Table 1 summarises existing bounds in comparison to the bound (1). Notably, the dominant factor of the bounds Xiao et al. (2023) and Mustafa et al. (2022) is the product of spectral norms $\prod_{l=1}^{L}\|W\|_\sigma$. This term arises as an estimate of the Lipschitz constant of a DNN. Interestingly, the corresponding term in our bound is the inverse of the standard deviation $\frac{1}{\sigma_w}$. We argue that these two terms are closely related through a randomized smoothing view of $G(W) := \mathbb{E}_{\widetilde{W} \sim \mathcal{N}(W, \sigma I)}[\mathbb{E}_{(x,y)\sim P}[\ell_{\text{adv}}(x, y, h(\widetilde{W}; \cdot))]]$. Indeed, while Mustafa et al. (2020) and Xiao et al. (2023) showed bounds on $H(W) := \mathbb{E}_{(x,y)\sim P}[\ell_{\text{adv}}(x, y, h(W; \cdot))]$, the bound (1) is on $\mathbb{E}_{\widetilde{W} \sim \mathcal{N}(W, \sigma I)}[H(\widetilde{W})] = G(W)$. Thus, G can be perceived as a smoothed variant of the H, where the smoothing is conducted w.r.t. to the weights W. Consequently, we suggest that the inherent smoothing effect on the expected (adversarial) risk of a stochastic network could be the primary factor behind the efficacy of these methods in achieving non-trivial bounds.

On The Importance Of Randomized Smoothing We consider stochastic DNNs, therefore, a non-vacuous bound is achieved only if there exists a posterior distribution Q over classifiers such that robustness to adversarial examples is achieved with high probability (over Q). The following proposition shows, even for the linear binary classification case, that the expected adversarial loss of a stochastic DNN is always larger than that of a deterministic model. The proof of the proposition can be found in Appendix B.

Proposition 1. *Consider binary classification with a linear model. Suppose we are given a classifier w^* and an input example $(x, y) \in \mathcal{X} \times \{-1, 1\}$. The quality of w^* is measured by the hinge-loss $\ell(t, y) = \phi(-yt)$, where $\phi(t) := \max(0, t)$. Then we have*

$$\max_{\|\delta\| \leq \epsilon} \ell(\langle w^*, x + \delta \rangle, y) = \phi(\|w^*\|\epsilon - y\langle w^*, x \rangle).$$

[3] For simplicity we assume an isotropic covariance matrix for both the prior and posterior with parameter σ_w

Furthermore, for any σ, there exists $\Gamma > 0$ such that when $d \geq 6$

$$\mathbb{E}_{\tilde{w}\sim\mathcal{N}(0,\sigma^2 I)}\left[\max_{\|\delta\|\leq\epsilon} \ell(\langle w^* + \tilde{w}, x+\delta\rangle, y)\right] \\ \geq \phi\left(\|w^*\|\epsilon - y\langle w^*, x\rangle + \epsilon\Gamma\right).$$

Note that since $\Gamma > 0$, the lower bound on the expected loss is larger than the loss on the mean w^* for any $w^* \in \mathbb{R}^d$. Notably, the randomized smoothing approach ensures that the model is $\frac{1}{\sigma}$-Lipschitz independently of the model weights. Therefore, we posit that randomized smoothing can help alleviate this. This is also experimentally verified in Section 4.3.

3.3 Computing The Certificate

In the previous section, we considered an idealized setting, assuming tractability of the expectations with respect to Q and the smoothing variable ϵ. In this section, we provide a tractable upper bound on (1) that holds with high probability. The main computational challenge lies in computing $\widehat{L}(Q, S, \tilde{\ell})$. To address this, we follow Langford and Caruana (2001) to utilize the Monte Carlo approximation of Q using m i.i.d. samples $\{W_j \sim Q \mid j \in [m]\}$, resulting in an unbiased estimate

$$\widehat{L}(\widehat{Q}, S, \ell) = \frac{1}{mn}\sum_{j=1}^{m}\sum_{i=1}^{n} \ell\big(x_i, y_i, h(W_j; \cdot)\big),$$

where $\widehat{Q}$ is the empirical distribution $\frac{1}{m}\sum_{j=1}^{m}\delta_{W_j}$.

To estimate an upper bound on $\widehat{L}(\widehat{Q}, S, \tilde{\ell})$, we need to address the difficulty of evaluating $g(W; \cdot)$, which is computationally intractable. Algorithm 1 summerizes an estimation procedure for $\widehat{L}(\widehat{Q}, S, \tilde{\ell})$. 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 BLC (Brown et al., 2001), which ensures a lower bound with probability at least $1-\alpha$. Finally, lines 10-12 compute $\tilde{\ell}(x, y, g(W; \cdot))$.

While Algorithm 1 provides an estimate to $\widehat{L}(\widehat{Q}, S, \tilde{\ell})$, its output is not an upper bound on $L(Q, \ell_{\text{adv}})$. The following theorem presents such a bound that holds with high probability in terms of the output of Algorithm 1.

Algorithm 1: Estimate an upper bound on $\widehat{L}(\widehat{Q}, S, \tilde{\ell})$. Based on CERTIFY (Cohen et al., 2019).

```
Input   : $S$, $N_0$, $N$, $\sigma$, $\{W_i\}_{i=1}^{m}$, $\alpha$, $R$
Output : An estimate of $\widehat{L}(\widehat{Q}, S, \tilde{\ell})$
errors_count ← 0
for $(x, y) \in S$ do
    for $j \leftarrow 1 : m$ do
        $\{\epsilon_t\}_{t\in[N_0]} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma I)$
        counts ← $\sum_{t\in[N_0]} h(W_j; x+\epsilon_t)$
        $c_A \leftarrow \arg\max_{k\in[K]} \text{counts}_k$
        $\{\epsilon_t\}_{t\in[N]} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma I)$
        counts ← $\sum_{t\in[N]} h(W_j; x+\epsilon_t)$
        $p_A \leftarrow \text{BLC}(\text{counts}_{c_A}, N, 1-\alpha)$
        if $p_A \leq \frac{1}{2}$ or $c_A \neq y$ or $\Phi^{-1}(p_A) < \frac{R}{\sigma}$ then
            errors_count ← errors_count +1
        end
    end
end
return Alg1 := errors_count $/m|S|$
```

Theorem 2. *Let $Q^0 \in \mathcal{M}(\mathcal{W})$ be prior distribution. Then with probability at least $1-\delta-\delta'-\delta''$, simultaneously for all $Q \in \mathcal{M}(\mathcal{W})$, the adversarial risk $L(Q, \ell_{\text{adv}})$ is upper-bounded by*

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

where Alg1 *is the output of Algorithm 1.*

The theorem provides a computationally tractable upper bound on the adversarial risk. The proof is found in Appendix B. The key step is to utilize Bernstein's inequality to bound $\widehat{L}(\widehat{Q}, S, \ell)$ combined with the observation that KL^{-1} is monotonically increasing in the first argument.

3.4 Training A Certifiable Network

In this section, we utilize Theorem 1 to train stochastic DNNs, for which the bound (2) is non-vacuous. The goal is then to find a posterior distribution $\widehat{Q} \in \mathcal{M}(\mathcal{W})$ that minimizes the adversarial PAC-Bayes bound (1). As discussed earlier, 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. First, we consider the evaluation of the function KL^{-1}. It does not have a closed-form solution, and back-propagating through the numerical algorithm for KL^{-1} is computationally expensive. While several upper bounds have been proposed in the literature (McAllester, 1999; Pérez-Ortiz et al., 2021; Tolstikhin and Seldin, 2013; Thiemann et al., 2017), we employ the *PAC-Bayes-quadratic* (Rivasplata et al., 2020) as it experimentally outperformed other surrogates. It is defined as $f_q(Q, Q^0, S, \ell_{\mathrm{adv}}) :=$

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

Now we proceed to estimate the gradient of $f_q(Q, Q^0, S, \ell_{\mathrm{adv}})$. 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. It remains to estimate the gradients of $\widehat{L}(Q, S, \ell_{\mathrm{adv}})$ with respect to the parameters of Q (i.e. μ and Σ). We employ the pathwise gradient estimator (Price, 1958) $\nabla_{\mu,\Sigma} \frac{1}{n} \sum_{i=1}^{n} \ell_{\mathrm{adv}}(x_i, y_i, g(W; \cdot))$, where $W := \mu + \Sigma^{\frac{1}{2}} V$, where V is sampled from $\mathcal{N}(0, I)$ [4]. This approach addresses the computational challenges in evaluating the expectation with respect to Q.

Next, we focus on approximating classifier $g(W; x)$ and the adversarial loss ℓ_{adv}. During the training process, we employ the empirical version of the classifier, which is given by $\frac{1}{M} \sum_{t \in [M]} h(W; x + \epsilon_t)$, where $\epsilon_t \overset{\mathrm{iid}}{\sim} \mathcal{N}(0, \sigma I)$. 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 provides a summary of the training procedure, outlining the steps involved in training the self-certified stochastic model. This algorithm incorporates the techniques mentioned above to optimize the model parameters and enhance its robustness against adversarial attacks.

Algorithm 2: Adversarial PAC-Bayes

```
Input  : Training set $S$, number of iteration $T$,
         batch size $B$, prior $Q^0$
Output : Posterior distribution model $Q$
Initialize $\mu_0$, $\rho_0$ from prior $Q^0$
$\mu \leftarrow \mu_0$
$\rho \leftarrow \rho_0$
for $t \leftarrow 1 : T$ do
    $S_b \leftarrow$ Sample a batch from $S$ with batch size $B$
    $V \sim \mathcal{N}(0, I)$
    $\Sigma_\rho \leftarrow \log(1 + \exp(\rho))$
    $W \leftarrow \mu + \Sigma_\rho^{\frac{1}{2}} V$
    $\tilde{S_b} \leftarrow []$
    for $(x, y) \in S_b$ do
        $\{\epsilon_i\}_{i=1}^{M} \overset{\mathrm{iid}}{\sim} \mathcal{N}(0, \sigma^2 I)$
        $\hat{x} = \arg\max_{\|\hat{x}-x\|_2 \le R} \ell\left(\tilde{x}, y, \frac{1}{M} \sum_{t \in [M]} h(W; \cdot + \epsilon_t)\right)$
        Append $\{(\hat{x} + \epsilon_i, y)\}_{i=0}^{M}$ to $\tilde{S_b}$.
    end
    $f_q(\mu, \rho) = f_q(\mathcal{N}(\mu_0, \Sigma_0)\mathcal{N}(\mu, \Sigma_\rho), \widehat{L}(\delta_W, \tilde{S_b}, \ell))$
    $\mu \leftarrow$ SGD/ADAM($\nabla_\mu f_q$)
    $\rho \leftarrow$ SGD/ADAM($\nabla_\rho f_q$)
end
return $\mathcal{N}(\mu, \rho)$
```

4 EXPERIMENTS

In this section, we demonstrate the practical applications of our self-certified model training and evaluation techniques, showcasing their utility and efficacy in computing a non-vacuous generalization bound. We compute the empirical certificates (Algorithm 1) and the adversarial risk bound (Theorem 2) for various settings on the established MNIST and CIFAR-10 datasets.

4.1 Experimental Setup

For all our experiments, we use a shared setup of evaluation parameters, which we report in the Appendix. However, most of the experimental parameters vary depending on the context. For example, we use different network architectures for MNIST and CIFAR-10.

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 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. Additionally, we observed that incorporating Batch Normalization (BatchNorm) (Ioffe and

[4] To ensure positivity of Σ during training we use the reparameterization $\Sigma = \log(1 + \exp(\rho))$ (Pérez-Ortiz et al., 2021).

Szegedy, 2015) facilitates faster prior learning. However, 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.

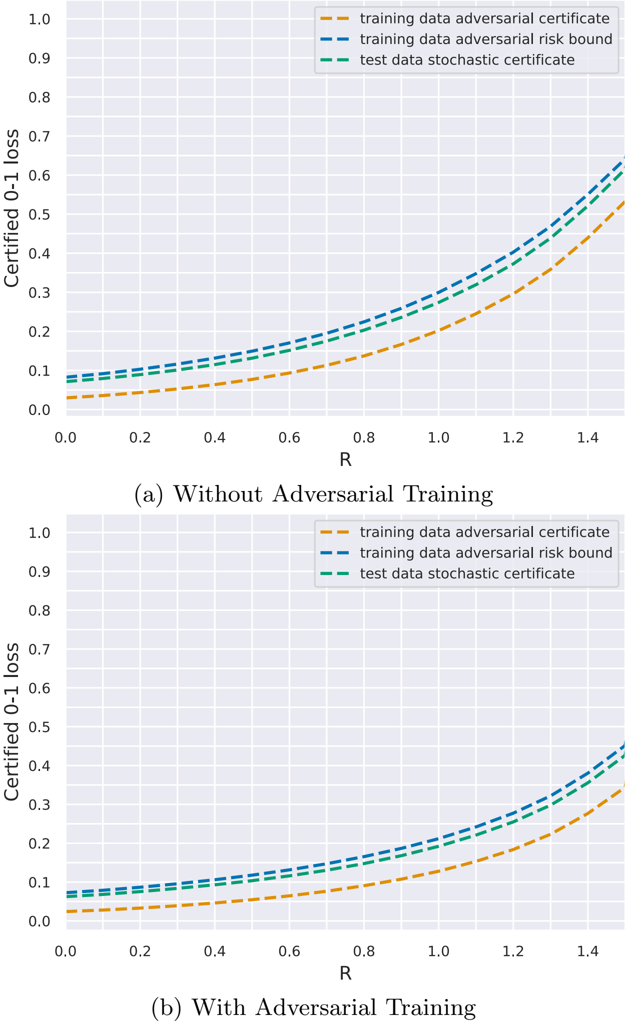
Our Model Is Sensitive To Hyperparameters Due to computational constraints–training and computing the certificates on CIFAR-10 takes approximately 30 hours on 8 A100 GPUs–we performed a greedy grid search of the hyperparameters. Overall the model displayed robustness to, e.g., the choice of learning rate. However, we found that, for CIFAR-10, 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 fail to converge, while any value below $\Sigma_0 = 0.01I$ showed no further improvement. Additionally, the prior exhibits a tendency 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. For MNIST, we searched within the same set of hyperparameters as for CIFAR-10. The prior Σ_0 was selected to be $0.03I$. We set dropout to 0.5.

4.2 Our Model Achieves Robust Risk Certificates Across All Settings

First of all, the most important experiment is to show empirically that our computed bounds are indeed non-vacuous and that the method remains robust across different settings. In Figures 1 and 2, we investigate the MNIST and CIFAR-10 setup.

The orange lines depict the bound on $\widehat{L}(\widehat{Q}, S, \tilde{\ell})$, while the blue lines depict the bound on $L(Q, \ell_{\text{adv}})$ as given in Theorem 2. The green line shows the empirical risk on a hold-out test dataset (i.e., the proportion of adversarially certified test sets' samples). Overall, our method consistently achieves robust certificates across all datasets. The computed generalization bounds are non-vacuous for both MNIST on the small network and CIFAR-10 on the deep network. They are also tight when compared to the stochastic test certificate, as the bounds barely exceed the test certificates.

Adversarial Training Leads To Better Certificates 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 12 in Algorithm 2. Adversarial training imposes a harder constraint on the models. Consequently, we anticipated that while it would enhance the empirical certificate (orange lines), it could potentially widen the generalization gap, thus leading to inferior risk certificates (blue lines). 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.

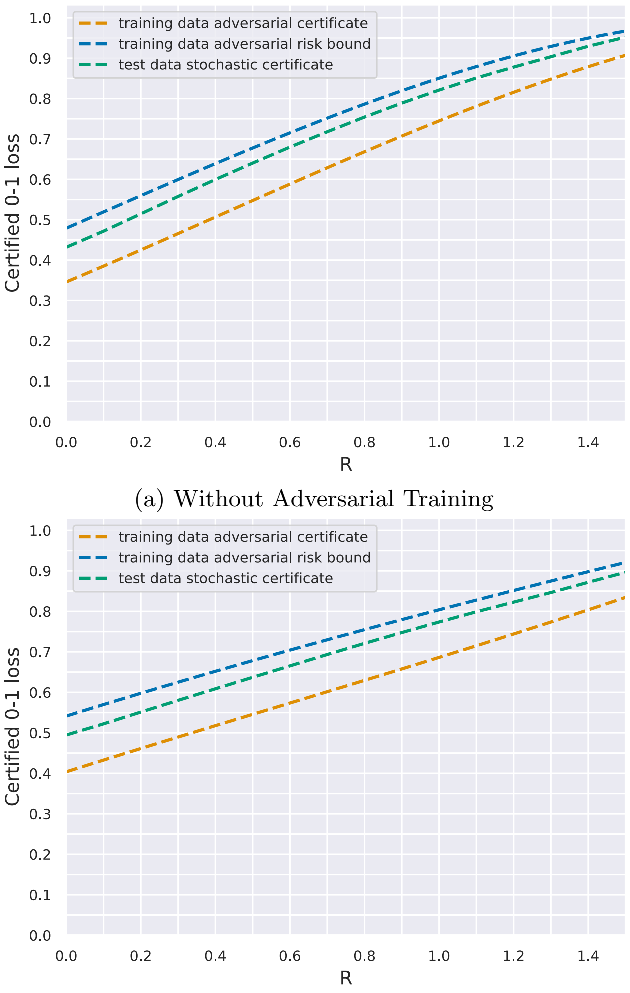


(a) Without Adversarial Training

(b) With Adversarial Training

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

Our Approach Scales Well With Deeper Networks When evaluating a much deeper network on CIFAR-10, we observed that despite the network's larger size (41 million parameters vs. 4.8 million parameters), the generalization gap did not significantly change. This finding emphasizes that the KL-divergence is a superior measure of complexity for deep neural networks compared to relying solely on the num-



(a) Without Adversarial Training

(b) With Adversarial Training.

Figure 2: Shown are adversarial risk bounds for a DNN trained with and without adversarial training on CIFAR-10. Each subfigure depicts the 0-1 loss over increasing attacker capacities (i.e., R). The blue curve is the risk bound, the orange curve is the empirical certified robustness of the training data, and the green curve is the empirical loss on hold-out data.

ber 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).

4.3 Smoothing Is Vital For Robust Classification

In our previous experiments, we observed that adversarial training combined with smoothed networks yields consistently strong empirical adversarial certificates across various settings. Smoothing is a core ingredient for achieving this strong performance. While we have already given an intuition for this in the methodology section, we here demonstrate this in an extended experiment on CIFAR-10. Figure 3 shows both adversarial certificates for smoothed networks (dashed lines) and empirical performance under PGD attack (solid lines) for non-smoothed deterministic networks.

Note that adversarial certificates are upper bounds on the empirical adversarial risk, while the empirical performance under PGD attack is a lower bound on the empirical adversarial risk. We find that the empirical performance for deterministic networks is consistently worse than the certificate for smoothed networks. 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.

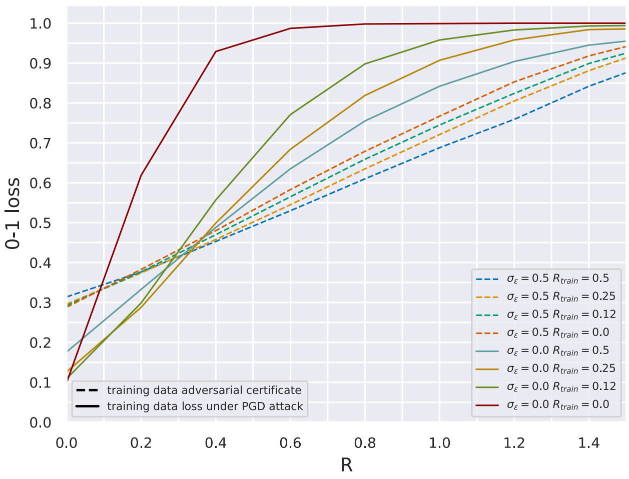
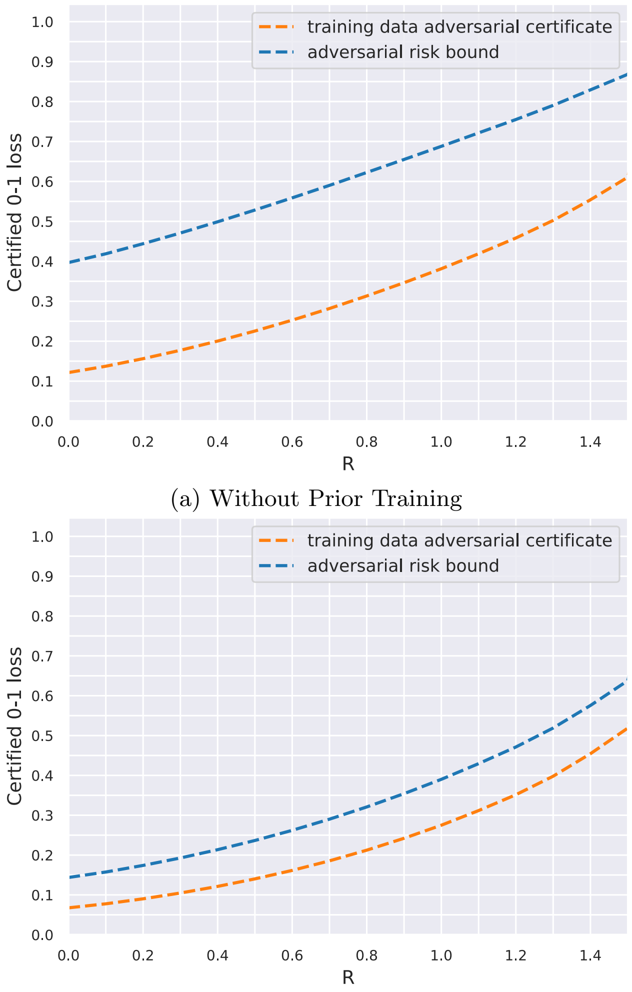


Figure 3: This figure shows the effect of adversarial smoothing on CIFAR-10. It shows empirical adversarial certificates (dashed lines) for smoothed networks (i.e., $\sigma_\epsilon = 0.5$) and empirical performance under PGD attack (solid lines) for a DNN trained with no smoothing (i.e., $\sigma_\epsilon = 0$). Each color corresponds to a network trained with varying attacker capacities (i.e., R_{train}). Overall, the figure depicts the 0-1 loss as the attacker capacity during inference (R) increases.

4.4 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 4. 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 4: Effect of a 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.

4.5 Further Insights Are In The Appendix

In further experiments, we find that varying the smoothing variance (σ_ϵ) for a range of attacker capacities substantiates the usefulness of smoothing. Additionally, as expected, without KL regularization, the generalization gap between the certificates and the adversarial risk bound explodes. Training–especially training the prior–is particularly prone to overfitting. We require careful tuning of the hyperparameters. There are more details on these findings and further experiments with interesting observations in the Appendix.

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

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.

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

Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356*, 2021.

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

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

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

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

Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. *Advances in neural information processing systems*, 32, 2019.

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

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.

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.

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

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.

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.

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

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.

Florian Graf, Sebastian Zeng, Bastian Rieck, Marc Niethammer, and Roland Kwitt. On measuring excess capacity in neural networks. *Advances in Neural Information Processing Systems*, 35:10164–10178, 2022.

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.

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 2017. ISBN 9783319633862. doi: 10.1007/978-3-319-63387-9_5. URL http://arxiv.org/abs/1702.01135.

Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. *arXiv preprint arXiv:1710.05468*, 2017.

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

Andrea Laforgia and Pierpaolo Natalini. On some inequalities for the gamma function. *Advances in Dynamical Systems and Applications*, 8(2):261–267, 2013.

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

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

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.

Philipp Liznerski, Saurabh Varshneya, Ece Calikus, Sophie Fellenz, and Marius Kloft. Reimagining anomalies: What if anomalies were normal? *arXiv preprint arXiv:2402.14469*, 2024.

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

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

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.

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

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.

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.

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/.

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.

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

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

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.

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.

Marvin Kenneth Simon. *Probability distributions involving Gaussian random variables: A handbook for engineers and scientists*. Springer, 2002.

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

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.

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.

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

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

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.

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.

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.

Colin Wei and Tengyu Ma. Data-dependent sample complexity of deep neural networks via lipschitz augmentation. *Advances in Neural Information Processing Systems*, 32, 2019.

Florian Wenzel, Théo Galy-Fajou, Matthäus Deutsch, and Marius Kloft. Bayesian nonlinear support vector machines for big data. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part I 10*, pages 307–322. Springer, 2017.

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.

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

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

Jiancong Xiao, Ruoyu Sun, and Zhi-quan Luo. Pac-bayesian spectrally-normalized bounds for adversarially robust generalization. *arXiv preprint arXiv:2310.06182*, 2023.

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

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.

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

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]

   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:

   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:

   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A 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 can be 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. 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. For MNIST, we picked 5e-2 as a learning rate for the posterior and 1e-3 for the prior. 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 12) 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 2). 300 Monte Carlo samples (m=300) are used for the adversarial risk bound. 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.

A.1 Time Complexity Of Algorithms

We discuss the time complexity of Algorithms 1 and 2 in this section. Algorithm 1 runs in $O(nmN)$, where n is the number of samples, m the number of MC-samples from Q, and N is the MC-sample size for estimating g. Here we assume a fixed architecture; that is, the forward path takes constant time. For Algorithm 2 the time complexity is $O(TBT_{\text{adv}}M)$, where T is the number of iteration for the algorithm, B is the batch size, M is the number of samples to estimate g, and T_{adv} is the number of iterations of the adversarial examples subroutine (e.g., number of steps for PGD).

B MISSING PROOFS

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

B.1 Proof Of Theorem 1

We first present the proof of Theorem 1.

Theorem 3 (Theorem 1 (restated))**.** *Let $Q^0 \in \mathcal{M}(\mathcal{W})$ be a prior probability measure on the set of weights $\mathcal{W}$. Consider the smoothed classifier $g := \mathcal{T}_\sigma h$. Then, with probability $1-\delta$ over the randomness of the training sample S, simultaneously for all $Q \in \mathcal{M}(\mathcal{W})$, we have*

$$L(Q, \ell_{\text{adv}}) \leq \mathrm{KL}^{-1}\left(\widehat{L}(Q, S, \tilde{\ell}), \frac{\mathrm{KL}(Q||Q^0) + \log(\frac{2\sqrt{n}}{\delta})}{n}\right), \tag{3}$$

where L and $\widehat{L}$ are defined for the smoothed classifier g.

Proof. We start by showing that $L(Q, \ell_{\text{adv}}) \leq L(Q, \tilde{\ell})$ in the context of the smoothed classifiers g. The following lemma summarizes this result.

Lemma 1. *Let $S := \{(x_1, y_1), \dots, (x_n, y_n)\} \subset \mathcal{X} \times \mathcal{Y}$ 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:

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

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

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

Proof. We commence the proof by first observing the stability property of function g as delineated in Theorem 5. 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 5 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.

□

Now it remains to bound $L(Q, \tilde{\ell})$. Note that $\tilde{\ell}$ is bounded by 1.0, therefore by the classical PAC-Bayes bound (see Theorem 6) we have with probability at least $1 - \delta$ over the randomness of S, we have

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

Combined with (6) concludes the proof.

□

B.2 Proof Of Theorem 2

We now state the proof of Theorem 2.

Theorem 4 (Theorem 2 (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 $L(Q, \ell_{\text{adv}})$ is upper-bounded by*

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

Proof. We structure the proof in three steps.

- First, we establish the bound $\widehat{L}(Q, S, \tilde{\ell}) \leq \widehat{L}(\widehat{Q}, S, \tilde{\ell})$. By applying Lemma 4 to $\frac{1}{m}\sum_{j=1}^{m}\tilde{\ell}(x, y, g(W_j; \cdot))$ and $\mathbb{E}_{W\sim Q}[\tilde{\ell}(x, y, g(W; \cdot))]$, we obtain the following bound (Langford and Caruana, 2001), with probability at least $1-\delta'$:
$$\widehat{L}(Q, S, \tilde{\ell}) \leq \mathrm{KL}^{-1}\left(\widehat{L}(\widehat{Q}, S, \tilde{\ell}), \frac{\log(\frac{2}{\delta'})}{n}\right). \tag{8}$$

- Second, we establish a bound on $\widehat{L}(\widehat{Q}, S, \tilde{\ell})$ by the output Alg1 of Algorithm 1. We summarize this result in the following lemma.

  Lemma 2. *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*
  $$\widehat{L}(\widehat{Q}, S, \tilde{\ell}) \leq \left(\text{Alg1} + \alpha + \sqrt{\frac{2\alpha(1-\alpha)\log(\frac{1}{\delta})}{mn}} + \frac{\log(\frac{1}{\delta})}{3mn}\right). \tag{9}$$

  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:
  $$\widehat{L}(\widehat{Q}, S, \tilde{\ell}) := \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\tilde{\ell}_{ij} \leq \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}(\hat{\ell}_{ij} + Z_{ij}). \tag{10}$$

  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 5), we obtain, with a probability of at least $1-\delta$, the following inequality:
  $$\begin{aligned} Z &\leq \frac{\sqrt{2\alpha(1-\alpha)\log(\frac{1}{\delta})}}{\sqrt{mn}} + \frac{\log(\frac{1}{\delta})}{3mn}, \\ \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m} Z_{ij} &\leq \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbb{E}[Z_{ij}] + \frac{\sqrt{2\alpha(1-\alpha)\log(\frac{1}{\delta})}}{\sqrt{mn}} + \frac{\log(\frac{1}{\delta})}{3mn}, \\ &\leq \alpha + \frac{\sqrt{2\alpha(1-\alpha)\log(\frac{1}{\delta})}}{\sqrt{mn}} + \frac{\log(\frac{1}{\delta})}{3mn}. \end{aligned} \tag{11}$$

  By noting that Alg1 $= \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}\hat{\ell}_{ij}$ and combining Eq.(10) and Eq.(11), we arrive at the final result. □

- Finally, by combining Equations (1), (8), and (9) and noting that KL^{-1} is monotonic in the first term, we achieve the final result and conclude the proof.

□

B.3 Proof Of Proposition 1

Proposition 2 (Proposition 1 (restated))**.** *Consider binary classification with a linear model. Suppose we are given a classifier w^* and an input example $(x, y) \in \mathcal{X} \times \{-1, 1\}$. Consider the loss $\ell(t, y) = \phi(-yt)$, where $\phi(t) := \max(0, t)$. Then we have,*

$$\max_{\|\delta\|\leq\epsilon} \ell(\langle w^*, x+\delta\rangle, y) = \phi(\|w^*\|\epsilon - y\langle w^*, x\rangle).$$

Furthermore, for any σ, as long as $d \geq 6$, we have:

$$\mathbb{E}_{\tilde{w}\sim\mathcal{N}(0,\sigma^2 I)}\left[\max_{\|\delta\|\leq\epsilon} \ell(\langle w^* + \tilde{w}, x+\delta\rangle, y)\right] \geq \phi\left(\|w^*\|\epsilon - y\langle w^*, x\rangle + \epsilon\Gamma\right),$$

where $\Gamma := \min\left(\frac{1}{8}\sigma(d-1)\left(\frac{\sigma}{\|w^*\|}\right), \|w^*\|\gamma\right)$ (with $\gamma = \sqrt{\frac{6(d-1)^2}{d(d+5)}} - 2$).

Before we prove the above result, we will need the following lemma:

Lemma 3. *As long as $d \geq 6$, we have:*

$$\mathbb{E}_{\tilde{\mathrm{w}}\sim\mathcal{N}(0,\sigma^2\mathrm{I})}\|w^* + \tilde{w}\| \geq \|w^*\| + \min\left(\frac{1}{8}\sigma(d-1)\left(\frac{\sigma}{\|w^*\|}\right), \|w^*\|\gamma\right), \tag{12}$$

where $\gamma := \sqrt{\frac{6(d-1)^2}{d(d+5)}} - 2 > 0$.

Proof. Note that by Taylor's Theorem, we have, for any positive x:

$$\sqrt{1+x} \geq 1 + \frac{1}{2}x - \frac{1}{8}x^2. \tag{13}$$

Indeed, the third derivative of the function $f(x) = \sqrt{1+x}$ is $\frac{3}{8\sqrt{1+x}^3}$, which is always positive.

Without loss of generality, we can assume that $w^* = (\|w^*\|, 0, \ldots, 0)^\top = (s, 0, \ldots, 0)^\top$ and $\tilde{w} = (a_1, a_2, \ldots, a_d)^\top$. Then we have $\mathbb{E}_{\tilde{\mathrm{w}}\sim\mathcal{N}(0,\sigma^2\mathrm{I})}(\|w^* + \tilde{w}\|) =$

$$\begin{aligned}\mathbb{E}\left(\sqrt{(s+a_1)^2 + \sum_{i=2}^{d} a_i^2}\right) &= \mathbb{E}\left(\sqrt{s^2 + A + 2sa_1}\right) = s\mathbb{E}\left(\sqrt{1 + \underbrace{\frac{A + 2sa_1}{s^2}}_{:=x}}\right) \\ &\geq s\mathbb{E}\left(1 + \frac{1}{2}x - \frac{1}{8}x^2\right) \geq s + \frac{1}{2}s\mathbb{E}(x) - \frac{1}{8}s\mathbb{E}\left(x^2\right),\end{aligned} \tag{14}$$

where $x := \frac{A+2sa_1}{s^2}$ and $A = \sum_{i=1}^{d} a_i^2 = \|\tilde{w}\|^2$. Next, we can calculate:

$$\begin{aligned}\mathbb{E}(x) &= \frac{\sigma^2 d}{s^2} \\ \mathbb{E}\left(x^2\right) &= \frac{1}{s^4}\left(\mathbb{E}\left(A^4\right) + 4s\mathbb{E}a_1^3 + 4s^2\mathbb{E}(a_1^2)\right) \\ &= \frac{1}{s^4}\left(3d\sigma^4 + \frac{d(d-1)}{2}\sigma^4 + 4s^2\sigma^2\right) \\ &= \frac{2}{2}\frac{d(d+5)}{2}\left(\frac{\sigma}{s}\right)^4 + 4\left(\frac{\sigma}{s}\right)^2.\end{aligned} \tag{15}$$

Plugging this back into equation (14), we obtain:

$$\begin{aligned}\mathbb{E}_{\tilde{\mathrm{w}}\sim\mathcal{N}(0,\sigma^2\mathrm{I})}(\|w^* + \tilde{w}\|) &\geq s + \frac{1}{2}sd\left(\frac{\sigma}{s}\right)^2 - s\cdot\frac{d(d+5)}{16}\left(\frac{\sigma}{s}\right)^4 - \frac{1}{2}s\left(\frac{\sigma}{s}\right)^2 \\ &= s + \frac{1}{2}s(d-1)\left(\frac{\sigma}{s}\right)^2 - s\frac{d(d+5)}{16}\left(\frac{\sigma}{s}\right)^4\end{aligned} \tag{16}$$

We now split into two cases:

Case 1: $s \neq 0$ **and** $\frac{d\sigma^2}{s^2} \leq 6\frac{d-1}{(d+5)}$:

$$\begin{aligned}
\mathbb{E}_{\tilde{\mathrm{w}}\sim\mathcal{N}(0,\sigma^2\mathrm{I})}(\|w^* + \tilde{w}\|) &\geq s + \frac{1}{2}s(d-1)\left(\frac{\sigma}{s}\right)^2 - s\frac{d(d+5)}{16}\left(\frac{\sigma}{s}\right)^4 \\
&\geq s + \frac{1}{8}s(d-1)\left(\frac{\sigma}{s}\right)^2 && (17) \\
&= \|w^*\| + \frac{1}{8}\sigma(d-1)\left(\frac{\sigma}{\|w^*\|}\right) && (18)
\end{aligned}$$

where at equation (17) we have used the assumption that $\frac{d\sigma^2}{s^2} \leq 6\frac{d-1}{(d+5)}$.

Case 2: $d\sigma^2 > 6s^2\frac{d-1}{(d+5)}$

In this case, $\sigma\sqrt{d-1} \geq s\sqrt{d-1}\sqrt{6\frac{d-1}{d(d+5)}} = s(2+\gamma)$, where $\gamma := \sqrt{\frac{6(d-1)^2}{d(d+5)}} - 2 > 0$ (recall $d \geq 6$).

Thus we have in this case:

$$\begin{aligned}
\mathbb{E}_{\tilde{\mathrm{w}}\sim\mathcal{N}(0,\sigma^2\mathrm{I})}(\|w^* + \tilde{w}\|) &\geq -\|w^*\| + \mathbb{E}_{\tilde{\mathrm{w}}\sim\mathcal{N}(0,\sigma^2\mathrm{I})}(\|\tilde{w}\|) \\
&= -\|w^*\| + \frac{\sqrt{2}\sigma\Gamma(d+1/2)}{\Gamma(d/2)} && (19) \\
&\geq -\|w^*\| + \sigma\sqrt{d-1} && (20) \\
&\geq \|w^*\| + \|w^*\|\gamma && (21)
\end{aligned}$$

where at line (20) we have used the following inequality for ratios of Gamma functions (valid for all $x > 0$ and $0 < \lambda < 1$), from Laforgia and Natalini (2013):

$$x^{1-\lambda} \leq \frac{\Gamma(x+1)}{\Gamma(x+\lambda)} \leq (x+1)^{(1-\lambda)}, \tag{22}$$

used with $x = \frac{d-1}{2}$ and $\lambda = \frac{1}{2}$. At line (19), we have used the explicit formula for the expectation of the norm of a Chi-squared distribution (cf. Simon (2002)). The lemma is proved when putting together equations (21) and (18).

□

Now, we can go back to the proof of Lemma 1:

Proof of Lemma 1. Note that ϕ is monotonically increasing, therefore, we have

$$\max_{\|\delta\|\leq\epsilon} \ell(\langle w^*, x+\delta\rangle, y) = \phi(\max_{\|\delta\|\leq\epsilon} -y\langle w^*, x+\delta\rangle).$$

Note that,

$$\max_{\|\delta\|\leq\epsilon} -y\langle w^*, x+\delta\rangle = \begin{cases} \max_{\|\delta\|\leq\epsilon} -\langle w^*, x+\delta\rangle = -\min_{\|\delta\|\leq\epsilon}\langle w^*, x+\delta\rangle, & \text{where, } y = 1, \\ \max_{\|\delta\|\leq\epsilon}\langle w^*, x+\delta\rangle, & \text{where, } y = -1. \end{cases}$$

Therefore,

$$\max_{\|\delta\|\leq\epsilon} -y\langle w^*, x+\delta\rangle = \begin{cases} \|w^*\|\epsilon - \langle w^*, x\rangle, & \text{where, } y = 1, \\ \|w^*\|\epsilon + \langle w^*, x\rangle, & \text{where, } y = -1. \end{cases}$$

Hence,

$$\max_{\|\delta\|\leq\epsilon} \ell(\langle w^*, x+\delta\rangle, y) = \phi(\|w^*\|\epsilon - y\langle w^*, x\rangle). \tag{23}$$

Now consider

$$\mathbb{E}_{\tilde{w}\sim\mathcal{N}(0,\sigma^2 I)}\left[\max_{\|\delta\|\leq\epsilon} \ell(< w^* + \tilde{w}, x+\delta >, y)\right].$$

Therefore, by (23) we have,

$$
\begin{aligned}
&\mathbb{E}_{\tilde{w}\sim\mathcal{N}(0,\sigma^2 I)}\left[\max_{\|\delta\|\leq\epsilon} \ell(< w^* + \tilde{w}, x+\delta >, y)\right] = \mathbb{E}_{\tilde{w}\sim\mathcal{N}(0,\sigma^2 I)}\left[\phi(\|w^*+\tilde{w}\|\epsilon - y\langle w^*+\tilde{w}, x\rangle)\right] \\
&\geq \phi(\mathbb{E}_{\tilde{w}\sim\mathcal{N}(0,\sigma^2 I)}\left[\|w^*+\tilde{w}\|\epsilon - y\langle w^*+\tilde{w}, x\rangle\right]) \\
&\geq \phi\left(\|w^*\|\epsilon + \Gamma\epsilon - \mathbb{E}_{\tilde{w}\sim\mathcal{N}(0,\sigma^2 I)}(y\langle w^*+\tilde{w}, x\rangle)\right) \qquad (24) \\
&\geq \phi\left(\|w^*\|\epsilon + \Gamma\epsilon - y\langle w^*, x\rangle\right), \qquad (25)
\end{aligned}
$$

where at the second line, we have used Jensen's inequality and at line (24) we have used Lemma (3), writing Γ for min $\left(\frac{1}{8}\sigma(d-1)\left(\frac{\sigma}{\|w^*\|}\right), \|w^*\|\gamma\right)$. This concludes the proof.

□

C BACKGROUND RESULTS

In this section, for completeness, we present the results from the literature required for the proofs of our theorems.

Theorem 5 (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.

Theorem 6 (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 $Q \in \mathcal{M}(\mathcal{W})$,*

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

Lemma 4 ((Langford and Caruana, 2001)). *Let $t_1, \dots, t_m \sim \mathcal{B}(\lambda)$ be independent Bernoulli variables with $\lambda \in [0, 1]$. Then with probability at least $1-\delta$,*

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

Lemma 5 (Bernstein's Inequality (Boucheron et al., 2003)). *Let $X_1, \dots, \mathcal{X}_n$ 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*

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

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

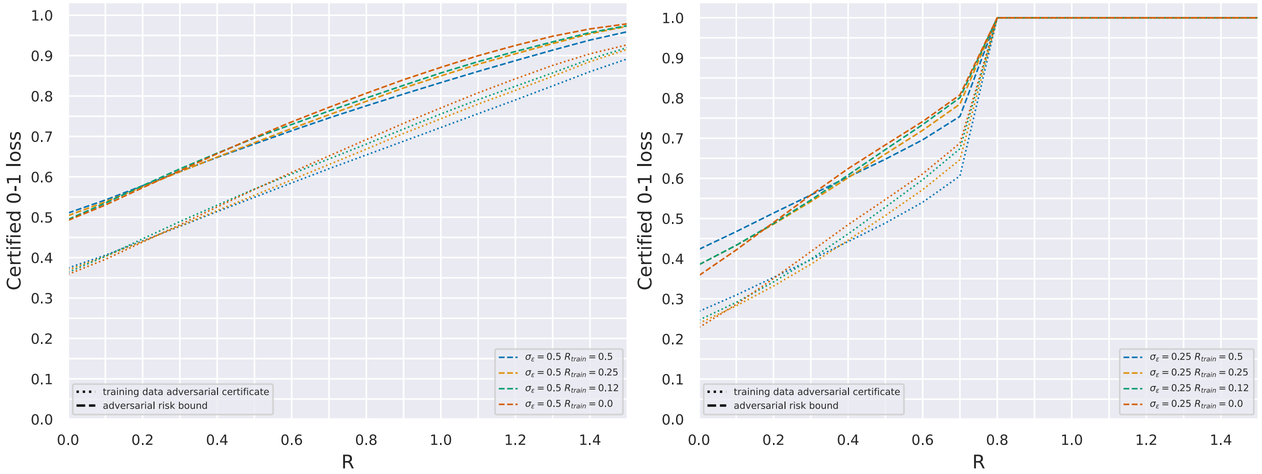
$$\frac{1}{n}\sum_{i=1}^{n} X_i \leq \sqrt{\frac{2\nu\log(\frac{1}{\delta})}{n}} + \frac{\log(\frac{1}{\delta})}{3n} \qquad (26)$$

D ADDITIONAL EXPERIMENTS

In the following sections, we provide ablation studies that focus on sensitive hyperparameters and highlight the usefulness of smoothing and adversarial learning.

D.1 Smoothing And Adversarial Training Improves Model Robustness

Figure 3 in the main paper investigated the effect of smoothing by comparing the training data adversarial certificate for smoothed networks to the training loss under PGD attack for deterministic networks. Here, we instead explore varying the random smoothing and adversarial attack hyperparameters in the training algorithm and plot the training data adversarial certificate and adversarial risk bound. 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 5, 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) $\sigma_\epsilon = 0.5$

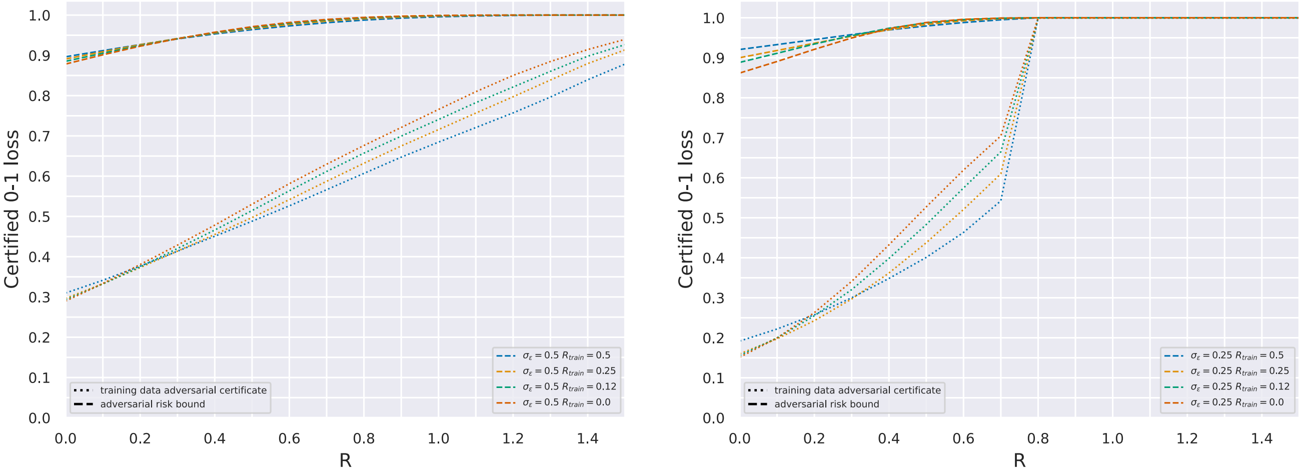
(b) $\sigma_\epsilon = 0.25$

Figure 5: Effects of varying the adversarial capacity and smoothing parameter during training on CIFAR-10. 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.

D.2 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.

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 6). This observation underscores the significance of optimizing the PAC-Bayes bound to computing non-vacuous generalization bounds.



(a) $\sigma_\epsilon = 0.5$

(b) $\sigma_\epsilon = 0.25$

Figure 6: 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.

D.3 Again, No Smoothing Deteriorates Robustness Significantly

In this section, we investigate the effect of smoothing by completely removing smoothing during training; i.e., setting $\sigma_\epsilon = 0$ in line 11 of Algorithm 2. In contrast to Figure 3 in the main paper, however, we 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 7 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.

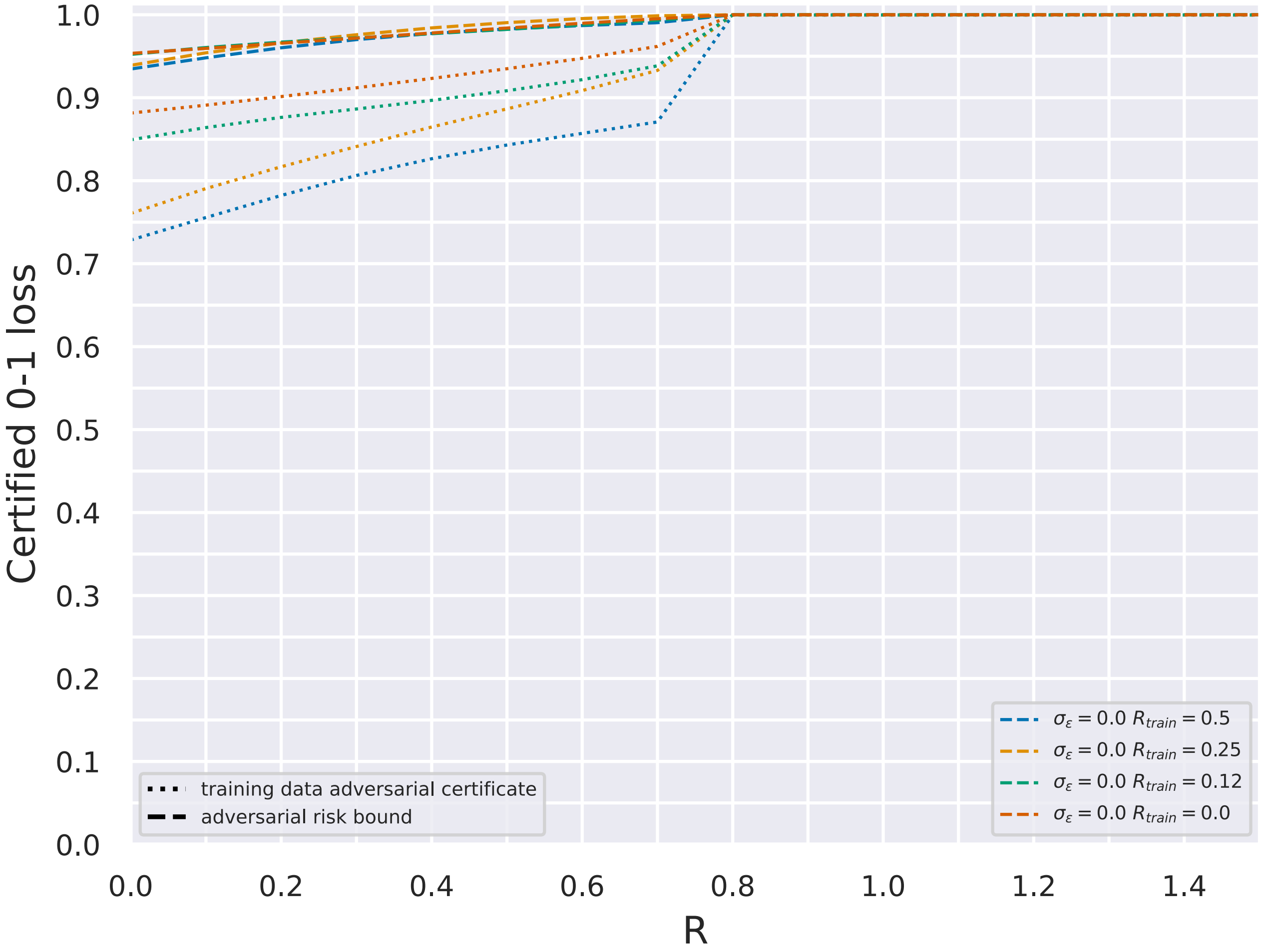


Figure 7: Effects of adversarial smoothing. This figure shows adversarial risk bounds for a DNN trained on CIFAR-10 with no smoothing during training (i.e., $\sigma_\epsilon = 0$) but with smoothing variance $\sigma_{epsilon} = 0.25$ for computing the certificate. Each color corresponds to a network trained with varying attacker capacities (i.e., R_{train}). The figure depicts the 0-1 loss as the attacker capacity during inference (R) increases. The dashed curve represents the risk bound, while the dotted curve represents the empirically certified robustness of the training data.

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 subjected these models to PGD attacks, thereby establishing a lower bound on empirical adversarial risk. See Figure 3 in the main paper.

D.4 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 8 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.

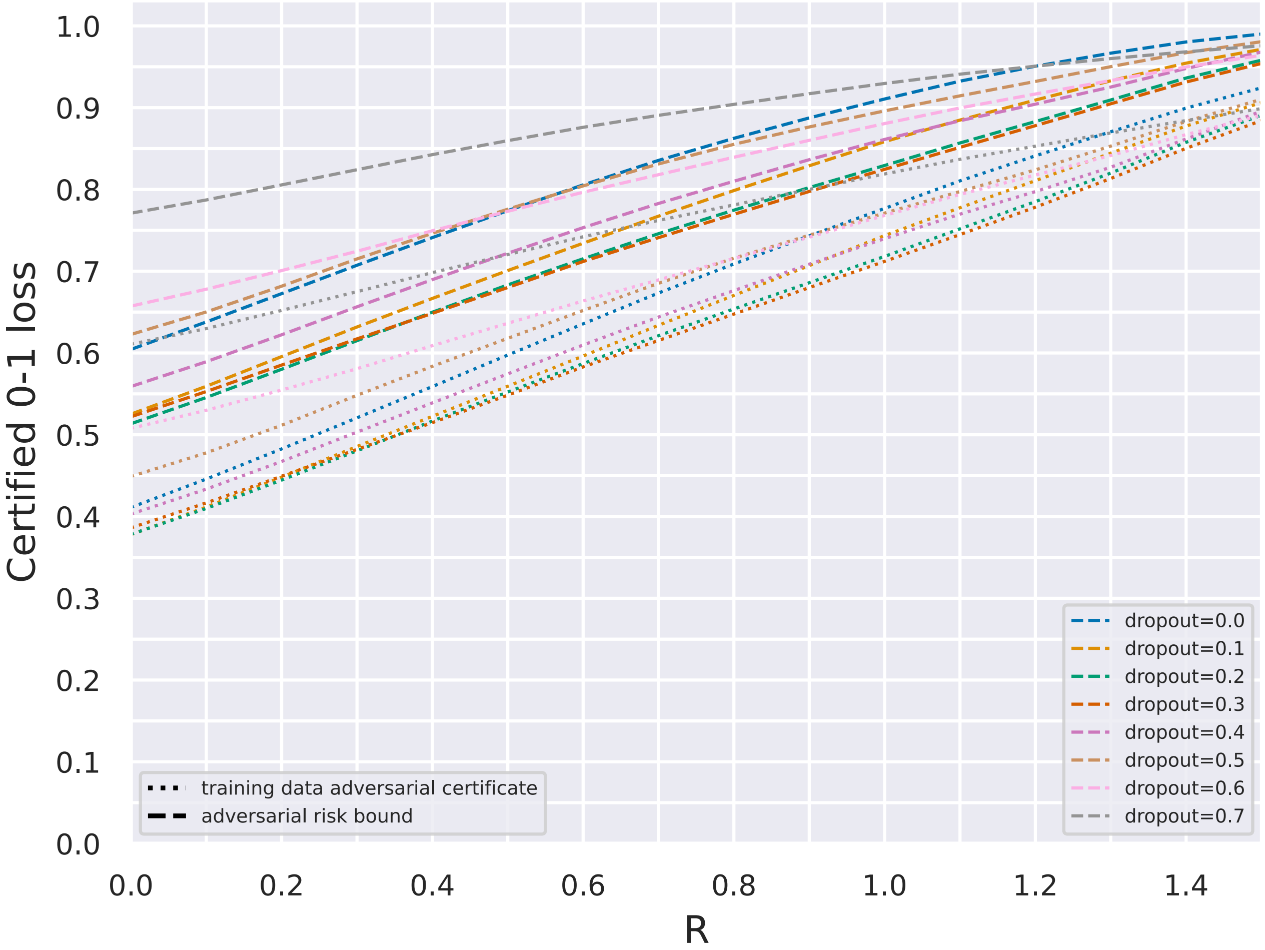


Figure 8: Effect of dropout in prior training. This figure shows adversarial risk bounds for a DNN trained on CIFAR-10. Each color corresponds to a network trained with varying dropout rates during prior training. The figure depicts the 0-1 loss as the attacker capacity during inference (R) increases.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

D.5 Training The Posterior Is Sensitive To The Prior Variance

As shown in Figure 4 in the main paper, 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 9 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.

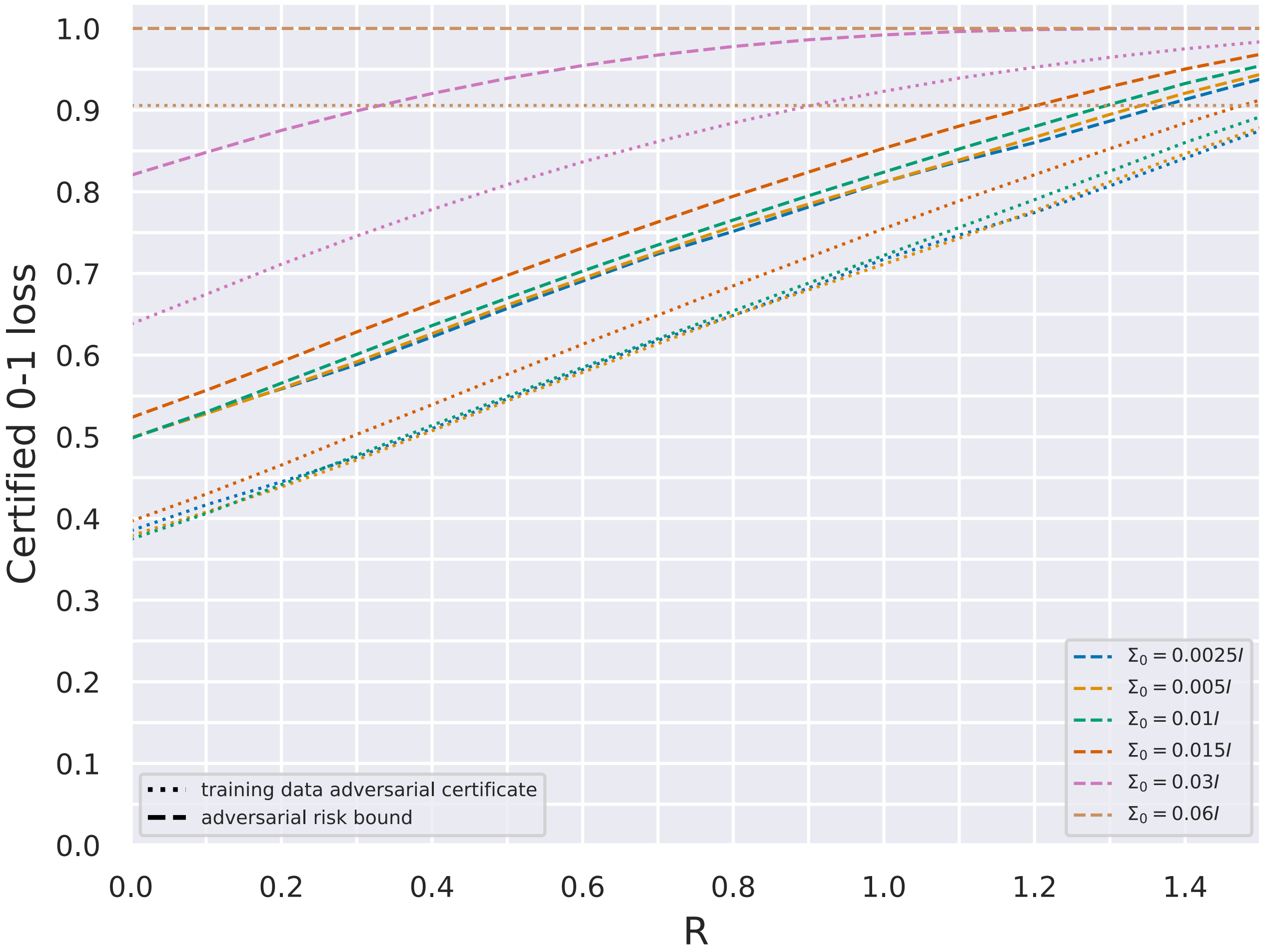


Figure 9: Effect of prior variance. This figure shows adversarial risk bounds for a DNN trained on CIFAR-10. Each color corresponds to a network trained with varying prior covariances Σ_0. The figure depicts the 0-1 loss as the attacker capacity during inference (R) increases. The dashed curve represents the risk bound, while the dotted curve represents the empirically certified robustness of the training data.

E 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 $\mathrm{KL}(p,q) = p \log \frac{p}{q} + (1-p)\log\frac{1-p}{1-q}$, 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:

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

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:

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