Why certified defences hurt robust generalisation

Anonymous Authors[1]

Abstract

Several works have shown that state-of-the-art classifiers are vulnerable to adversarial examples, raising serious concerns for their deployment in safety-critical applications. To address this issue two classes of methods have emerged: empirical defences and certified defences. While the former has good robustness, but no guarantees, the latter sacrifices some robustness in exchange for guarantees. Until now, these two approaches have not been systematically compared in the literature, and a clear understanding of the robustness gap remains elusive. In this paper, we show through extensive empirical evidence that models trained with certified defences suffer from worse accuracy, robustness and fairness than empirical defences. Further, we identify three key factors contributing to the robustness gap between the two approaches and support our arguments with both theoretical and experimental evidence. We hope this serves as a guide to practitioners regarding the use of various defences as well as a motivation for designing future methods.

1. Introduction

Several works have shown that state-of-the-art classifiers are vulnerable to adversarial examples, i.e., imperceptible perturbations to the input that can change the classifier prediction (Biggio et al., 2013; Szegedy et al., 2014). Hence, robustness to adversarial examples has become a crucial design goal when deploying machine learning models in safety critical applications. In real-world scenarios, robustness against many different types of input perturbations may be desired depending on the domain of application. In this paper, we consider the well-studied ℓ_p-ball threat model, where $\mathcal{B}_\epsilon := \{\delta : \|\delta\|_p \leq \epsilon\}$ represents the set of allowed perturbations for some ℓ_p-ball with radius ϵ centred around the origin. Then, for any distribution $\mathcal{D}$, classifier $f_\theta : \mathbb{R}^d \to \mathbb{R}^k$ parameterised by $\theta \in \mathbb{R}^p$, and loss function L, adversarial defences aim to solve the following robust optimisation problem:

$$\min_\theta \mathbf{R}_\epsilon(\theta) := \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\delta\in\mathcal{B}_\epsilon} L(f_\theta(x+\delta), y) \right] \quad (1)$$

We call $\mathbf{R}_\epsilon(\theta)$ the robust error when L is the 0-1 loss function. In practice, as the distribution $\mathcal{D}$ is unknown, we minimise the empirical robust error on a finite dataset D sampled from $\mathcal{D}$. Further, in the case of neural networks, the inner-maximisation is a non-convex optimisation problem and prohibitively hard to solve from a computational perspective (Katz et al., 2017; Weng et al., 2018). Instead, two efficient techniques are widely used to overcome the computational barrier: *empirical* defences that provide a lower bound and *certified* defences that provide an upper bound on the solution.

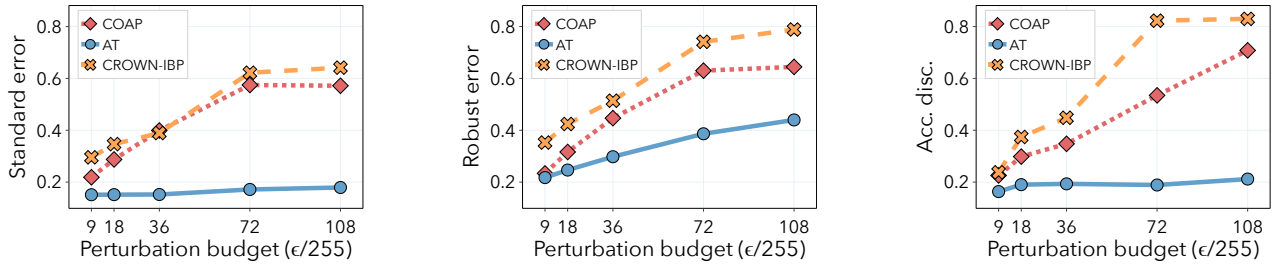
Adversarial training (AT) (Goodfellow et al., 2015; Madry et al., 2018) is one of the most popular empirical defences to date. AT minimises the worst-case empirical loss in Equation (1) by approximately solving the inner-maximisation problem with first-order optimisation methods. However, despite its simplicity and computational efficiency, AT does not provide any robustness guarantees, which are essential in many safety-critical domains.

To address this limitation, there has been significant interest in designing certified defences, i.e., methods for learning neural networks that are *provably* robust to ℓ_p-ball perturbations on the on the test data, allowing a safe upper bound estimate on the robust error. Many recent works have proposed to solve a convex relaxation of the inner-maximisation problem by relaxing the non-convex ReLU constraint sets with convex ones (Wong & Kolter, 2018; Raghunathan et al., 2018; Dvijotham et al., 2018; Zhang et al., 2020). Nevertheless, certified defences based on convex relaxations suffer from an inherent flaw: the upper bound they provide on the robust error is far from being tight (Salman et al., 2019).

While empirical defences have good robustness, but no guarantees, certified defences appear to sacrifice some robustness in exchange for guarantees. However, to the best of our knowledge, a systematic comparison and understanding of the gap in robustness between the two classes of methods is largely missing in the literature.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.



(a) Standard error (b) Robust error (c) Unfairness

Figure 1. Results for ℓ_2-ball perturbations on the CIFAR-10 test set. We compare ResNet architectures trained using state-of-the-art certified defences CROWN-IBP (Zhang et al., 2020; Xu et al., 2020) and COAP (Wong et al., 2018; Wong & Kolter, 2018) against the most popular empirical defence to date AT (Goodfellow et al., 2015; Madry et al., 2018). In Figures 1a and 1b, we plot the standard error and robust error respectively as the perturbation budget increases. In Figure 1c, we plot accuracy discrepancy, i.e. the difference between overall accuracy and worst class accuracy, as the perturbation budget increases. We refer the reader to Section 2 for further details on the models and robust evaluation.

In this paper, we provide the first systematic comparison between empirical and certified defences on popular computer vision datasets. Further, we identify three key factors contributing to the robustness gap between the two approaches and support our arguments with both theoretical and experimental evidence. We hope this serves as a guide to practitioners regarding the use of various defences as well as a motivation for designing future defence methods.

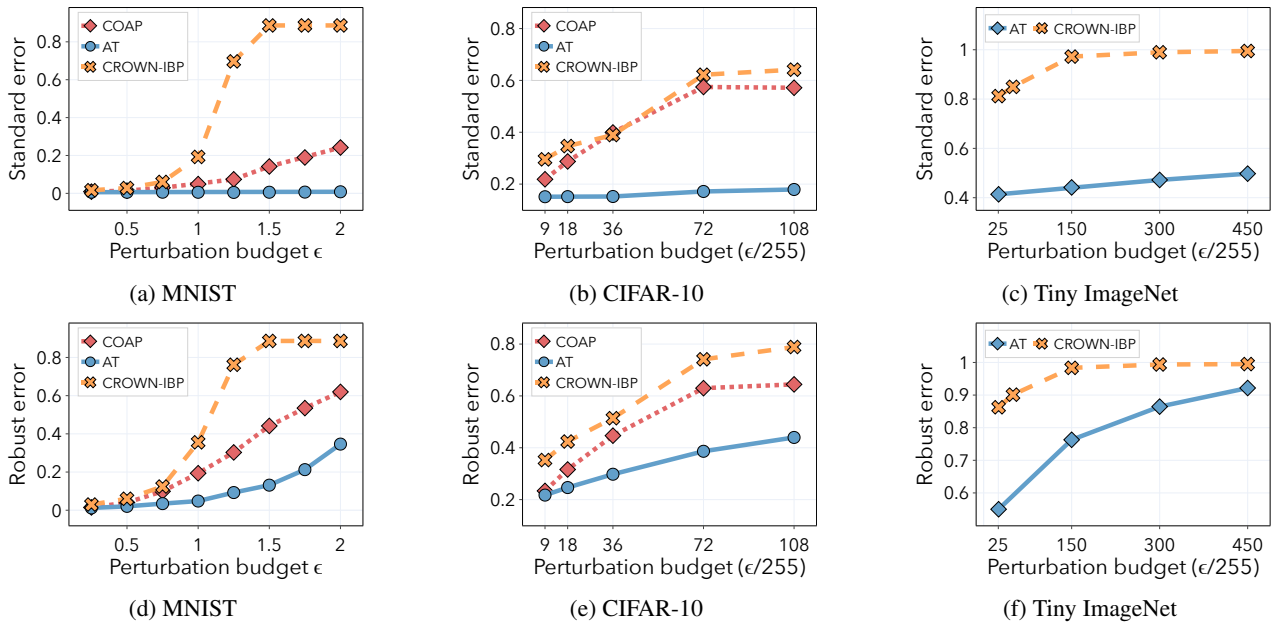
Specifically, our contributions are as follows:

- In Section 2, we show through extensive experimental evidence on image datasets that models trained with certified defences suffer from worse accuracy and robustness than empirical defences (see Figures 1a and 1b).
- In Section 3, we identify three factors driving the robustness gap: (i) the number of active neurons; (ii) the magnitude of the adversarial perturbations compared to the implicit margin of the data; (iii) the alignment between the adversarial perturbations and the "signal" direction. We provide both theoretical and experimental evidence on synthetic and image datasets to motivate these findings.
- In Section 4, we investigate the fairness of empirical and certified defences; we show that certified defences lead to models with worse fairness than empirical defences (see Figure 1c).

1.1. Related work

Deterministic defences A large body of literature has been devoted to developing methods for training neural networks that are provably robust to ℓ_p-ball perturbations. These works include methods based on semidefinite relaxations (Raghunathan et al., 2018), convex relaxations (Wong & Kolter, 2018; Wong et al., 2018), abstract interpretation (Mirman et al., 2018; Singh et al., 2018), and interval bound propagation (Gowal et al., 2019a; Zhang et al., 2020). Salman et al. (2019) unify these different views in a common convex relaxation framework and show that convex relaxations that approximate each ReLU output separately suffer an inherent tightness barrier. In particular, even the optimal convex relaxation cannot obtain tight bounds on the robust error. However, Salman et al. (2019) investigate the tightness of convex relaxations for verification purposes, i.e., for certifying the robustness of already trained models. In contrast, we explore the impact of this looseness on the robustness of models trained using convex relaxations, which to the best of our knowledge has not been systematically studied before. Previous research in this direction, such as the work of Jovanovic et al. (2021), compares the training dynamics of various convex relaxations. Our work, on the other hand, focuses on the comparison between empirical and certified defences.

Probabilistic defences As an alternative to deterministic defences, probabilistic defences give a guarantee of robustness with a certain probability. One of the most popular probabilistic defence is randomised smoothing (Cohen et al., 2019; Lécuyer et al., 2019; Li et al., 2019). The key idea behind this technique is to transform an arbitrary classifier $f_\theta : \mathbb{R}^d \to \mathbb{R}^k$ into a "smoothed" classifier $g_\theta : \mathbb{R}^d \to \mathbb{R}^k$. In particular, for a given data point $x \in \mathbb{R}^d$ and variance σ^2, the smoothed classifier's prediction $g_\theta(x)$ is defined as the most probable prediction by f_θ on the random variable $z \sim \mathcal{N}(x, \sigma^2 I_d)$. Despite its popularity, randomised smoothing significantly increases the computational cost of inference and suffers from several limitations. For example, Mohapatra et al. (2021) investigate the side-effects of randomised smoothing and show that it significantly hurts the disparity in class-wise accuracy; Sun et al. (2022) ob-



(a) MNIST (b) CIFAR-10 (c) Tiny ImageNet

(d) MNIST (e) CIFAR-10 (f) Tiny ImageNet

Figure 2. Results for ℓ_2-ball perturbations on MNIST, CIFAR-10 and Tiny ImageNet test sets. We compare ResNet architectures trained using state-of-the-art certified defences CROWN-IBP (Zhang et al., 2020; Xu et al., 2020) and COAP (Wong et al., 2018; Wong & Kolter, 2018) against the most popular empirical defence to date AT (Goodfellow et al., 2015; Madry et al., 2018). In Figures 2a to 2c we plot the standard error as the perturbation budget increases. In Figures 2d to 2f we plot the robust error as the perturbation budget increases. We omit COAP in Figures 2c and 2f as it does not scale to Tiny ImageNet.

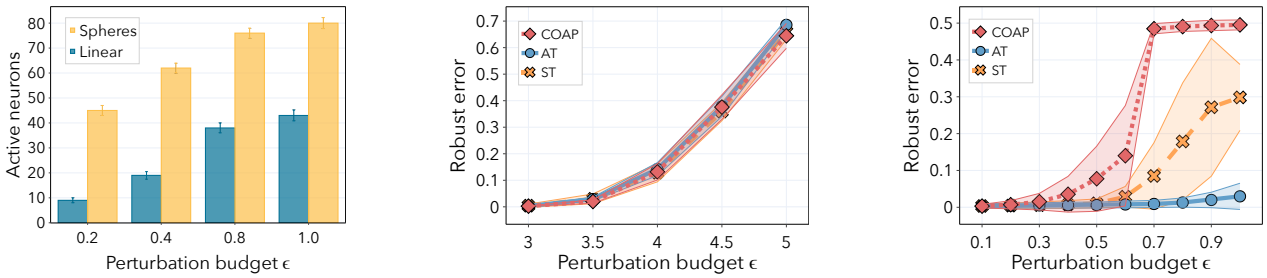
served that randomised smoothing is extremely vulnerable to low-frequency corruptions of the test data. Further, several works have exposed an accuracy-robustness tradeoff related to the smoothness of the classifier (Yang et al., 2020; Blum et al., 2020; Kumar et al., 2020). Compared to convex relaxation-based methods, the drawbacks of randomised smoothing are much better understood, and efforts are being made towards developing new defences to bridge the gap with empirical defences (Nandy et al., 2022). Therefore, we focus our attention in this paper on understanding the limitations of convex relaxation-based methods.

2. Generalisation gap between empirical and certified defences

In this section, we provide a systematic comparison of empirical and certified defences on three real-world computer vision datasets: Tiny ImageNet (Le & Yang, 2015), CIFAR-10 (Krizhevsky, 2009) and MNIST (LeCun et al., 1998). This paper mainly focuses on ℓ_2-ball perturbations. Supplementary experiments with ℓ_∞-ball perturbations, presented in Appendix E, indicate a similar trend. Moreover, we note that certified defences do not scale to larger vision datasets such as ImageNet, hence why we omit them in our experiments.

Methods Among certified defences based on convex relaxations, we consider the convex outer adversarial polytope (COAP) (Wong et al., 2018; Wong & Kolter, 2018), which achieves state-of-the-art certified robustness under ℓ_2-ball perturbations. We also consider CROWN-IBP (Zhang et al., 2018; Xu et al., 2020), which combines the tight convex relaxation CROWN (Zhang et al., 2018) with interval bound propagation (IBP) (Gowal et al., 2019b; Mirman et al., 2018) and achieves state-of-the-art certified robustness under ℓ_∞-ball perturbations. We compare these methods against the most popular empirical defence— adversarial training (AT) (Goodfellow et al., 2015; Madry et al., 2018), which is the de-factor method for adversarial robustness among practitioners.

Models and robust evaluation To reliably evaluate the defences in the ℓ_2-ball perturbations threat model, we use the strongest version of AutoAttack (AA+) (Croce & Hein, 2020). For CIFAR-10, we train a residual network (ResNet) and for MNIST we train a convolutional neural network (CNN). Both architectures were introduced in Wong et al. (2018) as standard benchmarks for certified defences. For Tiny ImageNet, we train a WideResNet along the same lines of Xu et al. (2020). We refer the reader to Appendix D.4 for complete experimental details.



(a) COAP active neurons (b) Linearly separable (c) Concentric spheres

Figure 3. We report mean and standard deviation over 15 seeds. In Figure 3a, we plot the number of neurons in the activation set for the linearly separable and concentric spheres distribution after they have been trained. In Figures 3b and 3c we plot the robust error for standard training (ST), adversarial training (AT) and convex outer adversarial polytope (COAP), when training on the linearly separable and concentric spheres distributions respectively. See Appendix D.2 for complete experimental details.

Results Several studies have shown that adversarial training may lead to an increase in standard error when compared with standard training (Raghunathan et al., 2020; Tsipras et al., 2019; Zhang et al., 2019). We observe the same phenomenon to a much higher degree when using certified defences. In particular, our experimental results show that certified defences not only suffer worse standard error but also worse robust error than adversarial training. Figures 2a and 2b shows that for increasing perturbation budget, the standard error gap between certified (CROWN-IBP, COAP) and empirical defences (AT) increases for both MNIST and CIFAR-10 respectively. Specifically, the gap reaches almost 90% for CROWN-IBP on MNIST when $\epsilon = 1.5$. Figures 2d and 2e show that the robust error gap also increases with increasing perturbation budget for both MNIST and CIFAR-10. In particular, the gap reaches almost 40% for the largest perturbation budgets. Additionally, we observe a significant standard and robust error gap between AT and CROWN-IBP for Tiny ImageNet in Figures 2c and 2f.

Our experiments with popular image datasets and neural network architectures reveal a significant standard and robust generalisation gap between certified and empirical defences. We will explain the reasons for this behaviour, both theoretically and experimentally, in the following section.

3. Underlying factors of the generalisation gap

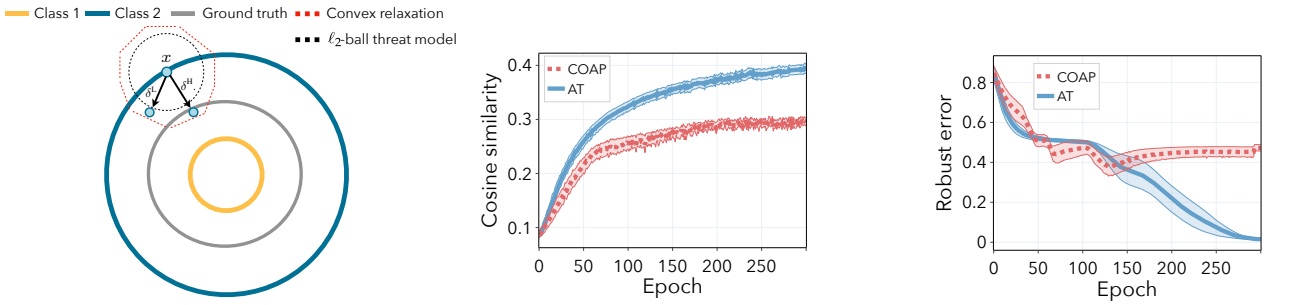
In this section, we investigate when certified defences perform poorly in practice. For the sake of clarity, we focus our attention on COAP, as it showed better robustness than CROWN-IBP when tested on image datasets in Section 2. Nevertheless, we expect that the takeaways from this section will transfer to CROWN-IBP as well.

In particular, we identify three factors underlying the robust generalisation gap: (i) the number of active neurons; (ii) the magnitude of the adversarial perturbations compared to the implicit margin of the data; (iii) the alignment between the adversarial perturbations and the signal direction. We verify these factors using image datasets and two synthetic data distributions: a linearly separable distribution similar to those studied Clarysse et al. (2022); Tsipras et al. (2019), and the concentric spheres distribution studied in Gilmer et al. (2018); Nagarajan & Kolter (2019).

Data and threat models Similar to the previous section, we focus on ℓ_2-ball perturbations of radius ϵ. To sample a data point from the linearly separable distribution with margin $\gamma > 0$, first we sample the label $y \in \{+1, -1\}$ with equal probability. Then, sample $\tilde{x} \in \mathbb{R}^{d-1}$ from a standard normal distribution $\tilde{x} \sim \mathcal{N}\left(0, \sigma^2 I_{d-1}\right)$ and set the covariate vector $x = [\gamma \operatorname{sgn}(y); \tilde{x}]$, where $[;]$ is the concatenation operator. To sample from the concentric spheres distribution with radii $0 < R_1 < R_{-1}$, first draw a binary label $y \in \{+1, -1\}$ with equal probability and then the covariate vector $x \in R^d$ is distributed uniformly on the sphere of radius R_y. Observe that achieving a low test error on the concentric spheres distribution requires a non-linear classifier.

3.1. Factor (i): Active neurons

The first factor we investigate is the number of active neurons for models trained with certified defences. The intuition behind this factor is the following: COAP relaxes the non-convex ReLU constraints for all neurons that activate within the perturbation set, i.e., when there exists a perturbation $\delta \in \mathcal{B}_\epsilon$ for which the input to the neuron crosses 0. Hence, the larger the number of active neurons, the worse the approximation. In Figure 3a we plot the number of active neurons on the concentric spheres and linear distributions as the perturbation budget increases. Observe that the amount is much higher for the concentric spheres than for the linearly separable distribution. This is consistent with the intuition that the complex spherical decision bound-



(a) Signal direction (b) Signal alignment (c) Robust error

Figure 4. We report mean and standard error over 15 seeds. In Figure 4a we plot a 2-dimensional visualisation of the concentric spheres dataset; the black dashed circle represents the ℓ_2-ball threat model and the red dashed polytope illustrates the convex approximation. Observe that for perturbations that are not significantly aligned with the signal direction, e.g. δ^{L}, the perturbed data point cannot cross the decision boundary. On the other hand, for perturbations that are aligned with the signal direction, e.g. δ^{H}, which overlaps with the signal direction, the perturbed data points can cross the decision boundary. In Figure 4b we plot the average cosine similarity between ℓ_2 norm-bounded perturbations on the training data and the signal directed vector. In Figure 4c we plot the robust error for adversarial training (AT) and convex outer adversarial polytope (COAP), for large perturbation budget $\epsilon = 1.0$. We observe that when cosine similarity is high, the gap in robust error between COAP and AT increases. Hence, especially approximations in the signal direction can hurt robust generalisation.

ary requires much more active neurons compared to the linear decision boundary which only needs 1. Moreover, in Figures 3b and 3c, we plot the robust error of standard training (ST), adversarial training (AT), and convex outer adversarial polytope (COAP) on the two distributions. We see that in contrast to the linear setting, COAP has a much higher robust error on the concentric spheres distribution than AT and ST. Overall, these findings indicate that the number of active neurons is a key factor underlying the robust generalisation gap between certified and empirical defences.

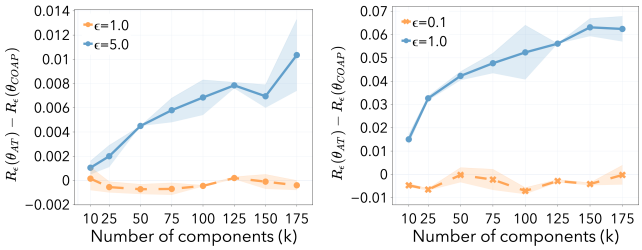
3.2. Factor (ii): Alignment between the adversarial perturbations and the signal direction

The second factor we investigate is the alignment between adversarial perturbations and the "signal" direction. We define the signal direction $s(x, y)$ for a data point (x, y) and a ground truth f^* as the direction along which we can flip the label with minimal perturbation budget. More formally,

$$s(x, y) = \underset{\delta}{\operatorname{argmin}} \underset{\text{s.t. } f^*(x+\epsilon\cdot\delta)\cdot y<0}{\min} \epsilon \tag{2}$$

For a point (x, y) drawn from the concentric spheres distribution, the signal direction is given by $s(x) = y\frac{x}{\|x\|_2}$. Observe that, the convex approximation essentially magnifies the radius of the ℓ_2-ball perturbation, and thus, a significant alignment of the ℓ_2-ball perturbation with the signal direction may result in points crossing the ground truth decision boundary, as illustrated in Figure 4a.

We support this argument with experiments on the concentric spheres dataset. In Figure 4b we plot the cosine similarity between the ℓ_2-ball perturbations computed on the training set and the signal direction, for large perturbation budget. By comparing the signal alignment in figure 4b with the corresponding robust test error in Figure 4c, we can see that during the early stages of training, the ℓ_2-ball perturbations are not aligned with the signal direction and the robust error for COAP is similar to AT. However, as training progresses and the perturbations begin to align with the signal direction, the robust error gap between COAP and AT increases. This provides evidence that as training progresses, ℓ_2-ball perturbations become significantly aligned with the signal direction and the robustness gap worsens.



(a) MNIST (b) CIFAR-10

Figure 5. We report mean and standard deviation over 2 seeds. In Figures 5a and 5b we plot the robust error gap between AT and COAP as the threat model captures more signal direction. We omit Tiny ImageNet as COAP cannot scale to datasets of large size. We refer the reader to Appendix D.5 for complete experimental details.

So far, we have intentionally investigated the factors on very controlled settings to gain a better understanding. Now, we

experiment with more realistic datasets, namely MNIST and CIFAR-10. As it is difficult to model the exact signal direction for image datasets, we propose an approximation of the signal-directed threat model introduced in Section 3.3. Using this, we demonstrate how the robust generalisation gap between AT and COAP gradually increases as the threat model becomes more closely aligned with the signal direction and the perturbation budget ϵ increases.

Below, we describe the approximation for the signal-directed threat model. First, randomly sample k unit vectors s_k and define the threat model as follows:

$$\mathcal{B}_\epsilon(x) = \cup_k \{z_1 = x + s_k\beta \mid |\beta| \leq \epsilon\} \tag{3}$$

As k increases, so does the probability that the signal direction will be significantly aligned with at least one of the k unit vectors. Then, we train both COAP and AT to be robust under this threat model. In Figures 5a and 5b we observe that the robust error gap for large perturbation budget between AT and COAP significantly increases as k increases, for both MNIST and CIFAR-10. This provides further evidence that *perturbations aligned with the signal direction worsen the robustness gap*.

Further, the robust generalisation gap of MNIST is significantly lower than that of CIFAR-10. This is likely due to the fact that the MNIST dataset is almost linearly separable, meaning that the decision boundary between classes is relatively simple and can be approximated by an hyperplane. We already displayed a similar distinction between the linearly separable and the concentric spheres distributions in Figure 3a. In light of these observations, it is not surprising that the robustness gap is less pronounced for MNIST than for CIFAR-10.

Finally, note how the robust generalisation gap is only observed for adversarial perturbations with large magnitude. This quantity plays an important role for robust generalisation, as shown already in Figure 3c, where the gap in robust error between certified and empirical defences increases with larger magnitudes.

3.3. Factor (iii): Magnitude of the adversarial perturbations

The third factor we investigate is the magnitude of the adversarial perturbations. In this section, we present a theoretical result which determines the minimum magnitude required to observe the robust generalisation gap. Further, we corroborate our theoretical result with experimental evidence on synthetic data.

Data and threat model We consider the linearly separable distribution. Note that for this distribution we did not observe a robust generalisation gap in Figure 3b. Since the alignment with the signal direction can increase the robustness gap – as suggested in Section 3.2 – we choose our threat model to maximise the alignment between perturbations and signal direction. In fact, we consider perturbations that are perfectly aligned with the signal direction. Observe that the signal direction, as defined in Equation (2), corresponds to the first index of the input. Hence, we define the set of allowed perturbations

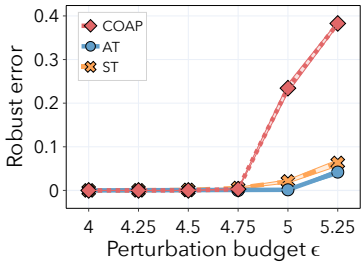


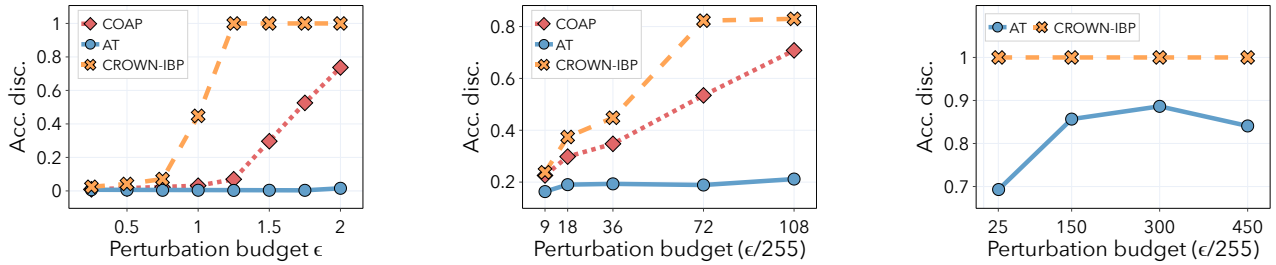
Figure 6. We report mean and standard deviation over 15 seeds. We consider the linearly separable distribution and the signal-directed threat model defined in Equation (4).We plot the robust error for standard training (ST), adversarial training (AT) and convex outer adversarial polytope (COAP) as the perturbation budget ϵ increases. See Appendix D.1 for complete experimental details.

$$\mathcal{B}_\epsilon(x) = \{z_1 = x + e_1\beta \mid |\beta| \leq \epsilon\}, \tag{4}$$

where e_1 is the canonical basis vector of the first coordinate. We refer the reader to Appendix A.1, where we derive an extension of the convex outer adversarial polytope (COAP) (Wong & Kolter, 2018) to this setting.

We are now ready to present our theoretical result for the linearly separable distribution. Perhaps surprisingly, Theorem 1 shows for a simple neural network that, in high dimensions, certified defences (COAP) yield higher robust error than empirical defences (AT) when the perturbation budget is large. Below, we outline our theoretical setting in more detail.

One-step gradient descent We study the early phase of neural network optimisation. Under structural assumptions on the data, it has been proven that one gradient step with sufficiently large learning rate can drastically decrease the training loss (Chatterji et al., 2021) and extract task-relevant features (Frei et al., 2022; Daniely & Malach, 2020). A similar setting was also studied recently in Ba et al. (2022) for the MSE loss in the high-dimensional asymptotic limit. We focus on the classification setting with binary cross-entropy loss.



(a) MNIST (b) CIFAR-10 (c) Tiny ImageNet

Figure 7. Results for ℓ_2-ball perturbations on MNIST, CIFAR-10 and Tiny ImageNet test sets. In Figures 7a to 7c we plot the accuracy discrepancy as the perturbation budget increases. We refer the reader to Appendix D.4 for complete experimental details.

One-neuron neural network Consider the hypothesis class of one-neuron shallow neural networks $f_\theta : \mathbb{R}^d \to \mathbb{R}$,

$$f_\theta(x) = a\,\mathrm{ReLU}\left(\theta^\top x\right) + b \tag{5}$$

where $x, \theta \in \mathbb{R}^d$ and $a, b \in \mathbb{R}$ and the only trainable parameter is θ_1. Note that as our distribution is linearly separable, our hypothesis class includes the ground truth.

Below we state our main theorem.

Theorem 1. *Let $\mathbf{R}_\epsilon$ be the robust risk of f_θ, defined as*

$$\mathbf{R}_\epsilon(\theta) := \mathbb{P}_{(x,y)}\left[\exists z \in \mathcal{B}_\epsilon(x) : y \neq \mathrm{sgn}\left(f_\theta(z)\right)\right].$$

Let $\bar{\theta}$ and $\tilde{\theta}$ be the network parameters after one step of gradient descent with respect to AT and COAP objectives. If the initialised network parameters θ satisfy

$$\frac{\|\theta_{2:d}\|_2}{\|\theta_1\|_2} > \sqrt{\max\left(\frac{(7\epsilon-\gamma)(\gamma+\epsilon)^4}{4\sigma^2(\gamma^2-10\gamma\epsilon+13\epsilon^2)}, \frac{(\gamma+\epsilon)^3}{12\sigma^2\epsilon}\right)}$$

and $\frac{2}{3}\gamma < \epsilon < \gamma$, COAP yields higher robust risk than AT:

$$\mathbf{R}_\epsilon(\tilde{\theta}) > \mathbf{R}_\epsilon(\bar{\theta}).$$

Theorem 1 relies on two main assumptions. The first is an assumption on the data dimensionality and the initialisation of the network parameters θ. For instance, if we initialise the network parameters θ by sampling from a sub-gaussian distribution, then the euclidean norm $\|\theta\|_2$ concentrates around $\sqrt{d}$ with high probability. Hence, the assumption is satisfied when the data dimensionality d is sufficiently high. The second assumption requires that the perturbation budget ϵ is sufficiently close to the implicit margin γ of the data. This provides further evidence that a large magnitude is required to observe the generalisation gap between empirical and certified defences. Moreover, it is consistent with our experimental evidence on image datasets, as the robust generalisation gap significantly worsen for large perturbation budgets.

Synthetic experiments We corroborate our theory with experimental evidence using a one-hidden layer neural network with 100 neurons. In particular, we investigate the effect of perturbation budget ϵ on robust generalisation for three different models: standard training (ST), adversarial training (AT) and convex outer adversarial polytope (COAP). In Figure 6, we plot the robust error as the perturbation budget ϵ increases. We observe that the gap between AT and COAP increases with increasing perturbation budget, which is in agreement with Theorem 1. Overall, these findings suggest that the magnitude of the adversarial perturbations is another key factor underlying the robust generalisation gap. In the next section, we extend our investigation of the gap between certified and empirical defences to another desirable metric, namely fairness.

4. Fairness gap between certified and empirical defences

Xu et al. (2021) observed that empirical defences, such as adversarial training, may induce a large discrepancy of robustness and accuracy among different classes. However, it is not yet understood if the same holds true for certified defences. In this section we examine the fairness of both certified and empirical defences, and to the best of our knowledge, ours is the first study of the fairness gap between the two approaches. In particular, we show that certified defences lead to models with worse fairness than empirical defences. Then, we present a theoretical result which determines the minimum magnitude required to observe the fairness gap, indicating that factor (iii) affects both the robustness and fairness gaps.

We measure fairness by examining the difference between overall accuracy and worst class accuracy and refer to this metric as *accuracy discrepancy* (Buolamwini & Gebru, 2018; Sanyal et al., 2022; Xu et al., 2021). More formally, let $\mathbf{R}(\theta)$ be the standard error of the classifier f_θ and $\mathbf{R}^k(\theta)$ the standard error conditioned on the class label k. The

accuracy discrepancy is measured as follows:

$$\frac{\max_k \mathbf{R}^k(\theta) - \mathbf{R}(\theta)}{1 - \mathbf{R}(\theta)} \tag{6}$$

For MNIST, CIFAR-10 and Tiny ImageNet we observe in Figures 7a to 7c that COAP and CROWN-IBP have a significant larger accuracy discrepancy than AT.

We now present a theoretical result on fairness for the linearly separable distribution. Our theoretical setting is the same as described in Theorem 1, i.e., we study a one neuron neural network after one step of gradient descent. In Theorem 2, we show for a simple neural network that, in high dimensions, certified defences (COAP) yield higher accuracy discrepancy than empirical defences (AT) when the perturbation budget is large.

Theorem 2. *Let $\bar{\theta}$ and $\tilde{\theta}$ be the network parameters after one step of gradient descent with respect to AT and COAP objectives. Then, if the network parameters satisfy the same condition of Theorem 1 and $\frac{2}{3}\gamma < \epsilon < \gamma$, COAP yields higher accuracy discrepancy than AT:*

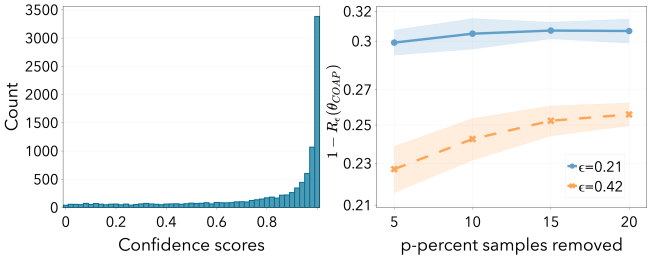
$$\max_k \mathbf{R}^k(\tilde{\theta}) - \mathbf{R}(\tilde{\theta}) > \max_k \mathbf{R}^k(\bar{\theta}) - \mathbf{R}(\bar{\theta}).$$

Theorem 2 relies on the same assumptions of Theorem 1. The first is an assumption on the data dimensionality which we discussed already. The second assumption requires that the perturbation budget ϵ is sufficiently close to the implicit margin γ of the data. This indicates that factor (iii) affects not only the robust generalisation gap, but also the fairness gap.

5. Discussion and future work

In this section, we discuss the implications of our observations for practitioners and for designing future defence methods. In particular, we present a simple experiment where we artificially increase the implicit margin of the data by removing some of the closest training samples to the decision boundary and observe that the robustness of COAP significantly improves.

First, we train a convolutional neural network on the entire CIFAR-10 dataset. Then, we artificially increase the distance between the classes by removing the p-percent of closest samples to the decision boundary of the trained classifier for $p \in [5, 10, 15, 20]$. Since, the distance cannot be analytically calculated for neural networks, we determine the closest samples using the confidence scores, i.e. the softmax probabilities, of the trained classifier to the true class. Hence, we remove the p-percent of points with the lowest confidence scores. In Figure 8a we plot the distribution of the confidence scores for the classifier trained on CIFAR-10, we observe that most of the scores are significantly large.



(a) MNIST (b) CIFAR-10

Figure 8. We report mean and standard deviation over 5 seeds. In Figure 8a we plot the distribution of the confidence scores, i.e. softmax probabilities, for a convolutional neural network classifier trained on CIFAR-10. In Figure 8b we plot COAP robust accuracy as we remove p-percent of the points with lowest confidence scores. We refer the reader to Appendix D.4 for complete experimental details.

To ensure a fair comparison for $p < 20$, we uniformly subsample the dataset such that the number of training samples is the same for all experiments.

We consider the ℓ_2-ball perturbations threat model and we reliably evaluate the robust error using the strongest version of AutoAttack (AA+) (Croce & Hein, 2020). Figure 8b shows that the robust accuracy of COAP increases as we remove a significant percentage of samples close to the decision boundary from the training data. Further, the gain in robust accuracy is more significant when the perturbation budget is large.

The main takeaway from this experiment is that COAP suffers in robust generalisation for datasets where the implicit margin is smaller. Hence, practitioners should consider the implicit margin of the data when selecting the most appropriate defence for their specific situation. A potential approach could be using the standard accuracy as a guide, as it correlates well with the implicit margin of the data. Furthermore, future certified defences could be improved by selectively ignoring datapoints that are close to the decision boundary during the certification process. We leave both of these ideas as interesting directions for future research.

6. Conclusion

In this paper, we show that that models trained with certified defences suffer from worse accuracy and robustness than empirical defences. Further, we are the first to show that certified defences also suffer from worse fairness. Finally, we develop intuition on both synthetic and image datasets for why certified defences hurt generalisation. We believe that shedding light on the robustness gap between empirical and certified defences will not only provide us with a clearer picture of the trade-offs observed in practice but also lead to better approaches for certified robustness.

References

Ba, J., Erdogdu, M. A., Suzuki, T., Wang, Z., Wu, D., and Yang, G. High-dimensional Asymptotics of Feature Learning: How One Gradient Step Improves the Representation, 2022. arXiv:2205.01445.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Srndic, N., Laskov, P., Giacinto, G., and Roli, F. Evasion Attacks against Machine Learning at Test Time. In *Machine Learning and Knowledge Discovery in Databases - European Conference*, 2013.

Blum, A., Dick, T., Manoj, N., and Zhang, H. Random Smoothing Might be Unable to Certify L\(ınfty\) Robustness for High-Dimensional Images. *Journal of Machine Learning Research*, 2020.

Buolamwini, J. and Gebru, T. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, pp. 77–91. PMLR, January 2018. ISSN: 2640-3498.

Chatterji, N. S., Long, P. M., and Bartlett, P. L. When Does Gradient Descent with Logistic Loss Find Interpolating Two-Layer Networks? *Journal of Machine Learning Research*, 2021.

Clarysse, J., Hörrmann, J., and Yang, F. Why adversarial training can hurt robust accuracy, 2022. arXiv:2203.02006.

Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. Certified Adversarial Robustness via Randomized Smoothing. In *Proceedings of the International Conference on Machine Learning*, 2019.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of the International Conference on Machine Learning*, 2020.

Daniely, A. and Malach, E. Learning Parities with Neural Networks. In *Advances in Neural Information Processing Systems*, 2020.

Dvijotham, K., Stanforth, R., Gowal, S., Mann, T. A., and Kohli, P. A Dual Approach to Scalable Verification of Deep Networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2018.

Erdemir, E., Bickford, J., Melis, L., and Aydöre, S. Adversarial Robustness with Non-uniform Perturbations. In *Advances in Neural Information Processing Systems*, 2021.

Frei, S., Chatterji, N. S., and Bartlett, P. L. Random Feature Amplification: Feature Learning and Generalization in Neural Networks, May 2022. arXiv:2202.07626.

Gilmer, J., Metz, L., Faghri, F., Schoenholz, S. S., Raghu, M., Wattenberg, M., and Goodfellow, I. J. Adversarial Spheres. 2018. arXiv: 1801.02774.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and Harnessing Adversarial Examples. In *Proceedings of the International Conference on Learning Representations*, 2015.

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T., and Kohli, P. On the Effectiveness of Interval Bound Propagation for Training Verifiably Robust Models, August 2019a. arXiv:1810.12715 [cs, stat].

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T. A., and Kohli, P. Scalable Verified Training for Provably Robust Image Classification. In *International Conference on Computer Vision*, 2019b.

Jovanovic, N., Balunovic, M., Baader, M., and Vechev, M. T. On the Paradox of Certified Training. February 2021.

Katz, G., Barrett, C. W., Dill, D. L., Julian, K., and Kochenderfer, M. J. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In *Proceedings of the International Conference of Computer Aided Verification*, 2017.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations*, 2015.

Krizhevsky, A. Learning multiple layers of features from tiny images. *citeseer*, 2009.

Kumar, A., Levine, A., Goldstein, T., and Feizi, S. Curse of Dimensionality on Randomized Smoothing for Certifiable Robustness. In *Proceedings of the International Conference on Machine Learning*, 2020.

Le, Y. and Yang, X. S. Tiny imagenet visual recognition challenge. 2015.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE*, (11), 1998.

Li, B., Chen, C., Wang, W., and Carin, L. Certified Adversarial Robustness with Additive Noise. In *Advances in Neural Information Processing Systems*, 2019.

Lécuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. Certified Robustness to Adversarial Examples with Differential Privacy. In *IEEE Symposium on Security and Privacy*, 2019.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In *Proceedings of the International Conference on Learning Representations*, 2018.

Mirman, M., Gehr, T., and Vechev, M. T. Differentiable Abstract Interpretation for Provably Robust Neural Networks. In *Proceedings of the International Conference on Machine Learning*, 2018.

Mohapatra, J., Ko, C.-Y., Weng, L., Chen, P.-Y., Liu, S., and Daniel, L. Hidden Cost of Randomized Smoothing. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2021.

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

Nandy, J., Saha, S., Hsu, W., Lee, M. L., and Zhu, X. X. Towards Bridging the gap between Empirical and Certified Robustness against Adversarial Examples, July 2022. arXiv:2102.05096 [cs].

Raghunathan, A., Steinhardt, J., and Liang, P. Certified Defenses against Adversarial Examples. In *Proceedings of the International Conference on Learning Representations*, 2018.

Raghunathan, A., Xie, S. M., Yang, F., Duchi, J. C., and Liang, P. Understanding and Mitigating the Tradeoff between Robustness and Accuracy. In *Proceedings of the International Conference on Machine Learning*, 2020.

Salman, H., Yang, G., Zhang, H., Hsieh, C.-J., and Zhang, P. A Convex Relaxation Barrier to Tight Robustness Verification of Neural Networks. In *Advances in Neural Information Processing Systems*, 2019.

Sanyal, A., Hu, Y., and Yang, F. How unfair is private learning? In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2022.

Singh, G., Gehr, T., Mirman, M., Püschel, M., and Vechev, M. Fast and Effective Robustness Certification. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Sun, J., Mehra, A., Kailkhura, B., Chen, P.-Y., Hendrycks, D., Hamm, J., and Mao, Z. M. A Spectral View of Randomized Smoothing Under Common Corruptions: Benchmarking and Improving Certified Robustness. In *Computer Vision – ECCV 2022*, volume 13664. 2022.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations*, 2014.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness May Be at Odds with Accuracy. In *Proceedings of the International Conference on Learning Representations*, 2019.

Weng, T., Zhang, H., Chen, H., Song, Z., Hsieh, C., Daniel, L., Boning, D. S., and Dhillon, I. S. Towards fast computation of certified robustness for relu networks. In *Proceedings of the International Conference on Machine Learning*, 2018.

Wong, E. and Kolter, J. Z. Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope. In *Proceedings of the International Conference on Machine Learning*, 2018.

Wong, E., Schmidt, F. R., Metzen, J. H., and Kolter, J. Z. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems*, 2018.

Xu, H., Liu, X., Li, Y., Jain, A. K., and Tang, J. To be Robust or to be Fair: Towards Fairness in Adversarial Training. In *Proceedings of the International Conference on Machine Learning*, 2021.

Xu, K., Shi, Z., Zhang, H., Wang, Y., Chang, K.-W., Huang, M., Kailkhura, B., Lin, X., and Hsieh, C.-J. Automatic Perturbation Analysis for Scalable Certified Robustness and Beyond. In *Advances in Neural Information Processing Systems*, 2020.

Yang, G., Duan, T., Hu, J. E., Salman, H., Razenshteyn, I. P., and Li, J. Randomized Smoothing of All Shapes and Sizes. In *Proceedings of the International Conference on Machine Learning*, pp. 10693–10705, 2020.

Zhang, H., Weng, T.-W., Chen, P.-Y., Hsieh, C.-J., and Daniel, L. Efficient Neural Network Robustness Certification with General Activation Functions. In *Advances in Neural Information Processing Systems*, 2018.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically Principled Trade-off between Robustness and Accuracy. In *Proceedings of the International Conference on Machine Learning*, 2019.

Zhang, H., Chen, H., Xiao, C., Gowal, S., Stanforth, R., Li, B., Boning, D. S., and Hsieh, C.-J. Towards Stable and Efficient Training of Verifiably Robust Neural Networks. In *Proceedings of the International Conference on Learning Representations*, 2020.

A. Certified defences for signal-directed perturbations

A.1. Certified defences for signal-directed perturbations

We now formulate the convex outer adversarial polytope (COAP) (Wong & Kolter, 2018) for adversaries that concentrate all their budget along the signal direction in the input. Our derivation can be seen as an extension of Wong & Kolter (2018); Erdemir et al. (2021).

Network architecture For the sake of clarity, we consider a 2-layers feed-forward ReLU network. However, our formulation can easily be extended to multiple layers. We define $f_\theta(x) : \mathbb{R}^d \to \mathbb{R}^2$ as follows:

$$x \xrightarrow{x+\delta} z_1 \xrightarrow{W_1 z_1 + b_1} \hat{z}_2 \xrightarrow{\mathrm{ReLU}(\cdot)} z_2 \xrightarrow{W_2 z_2 + b_2} \hat{z}_3 \tag{7}$$

where $x \in \mathbb{R}^d$, $z_1 \in \mathcal{B}_\epsilon(x)$, W_1 and W_2 are linear operators, and $\theta = \{W_i, b_i\}_{i=1,2}$ is the set of network parameters.

We define the adversarial polytope $\mathcal{Z}_\epsilon(x)$ as the set of all final-layer activations attainable by perturbing x with some $\tilde{x} \in \mathcal{B}_\epsilon(x)$:

$$\mathcal{Z}_\epsilon(x) = \{f_\theta(\tilde{x}) : \tilde{x} \in \mathcal{B}_\epsilon(x)\}$$

The key idea behind COAP (Wong & Kolter, 2018) is to construct a convex outer bound to this adversarial polytope. Specifically, we relax the ReLU activations $z_2 = \mathrm{ReLU}\,(\hat{z}_2)$ with their convex envelopes:

$$z_2 \geq 0,\ z_2 \geq \hat{z}_2,\ (u - \ell) z_2 \leq u\hat{z}_2 - u\ell$$

where u and ℓ are respectively the pre-activations $\hat{z}_2$ upper and lower bounds. We assume that these bounds are known and provide a closed form solution to compute them in Appendix A.3. Further, let $\tilde{\mathcal{Z}}_\epsilon(x)$ be the outer bound to the adversarial polytope obtained from relaxing the ReLU constraints. Then, given a data point x with known label y, we can formalise the problem of finding an adversarial example with a linear program as follows:

$$\min_{\hat{z}_3} [\hat{z}_3]_y - [\hat{z}_3]_{\bar{y}} = c^\top \hat{z}_3 \qquad \text{s.t. } \hat{z}_3 \in \tilde{\mathcal{Z}}_\epsilon(x) \tag{8}$$

where $\bar{y}$ is the binary negation of y. Note that if we solve this linear program and find that the objective is positive, then we know that no input perturbation within the threat model can misclassify the example. However, solving the linear program in Equation (8) for every example in the dataset is intractable. Therefore, we solve the dual formulation stated in the theorem below.

Theorem 3. *The dual of the linear program* (8) *can be written as*

$$\begin{aligned} \max_{\alpha} \quad & \widetilde{J}_\epsilon\left(x, g_\theta(c, \alpha)\right) \\ \textit{s.t.} \quad & \alpha_j \in [0, 1],\ \forall j \end{aligned}$$

where $\widetilde{J}_\epsilon(x, \nu_1, \nu_2, \nu_3)$ *is equal to*

$$-\sum_{i=1}^{2} \nu_{i+1}^\top b_i + \sum_{j \in \mathcal{I}} \ell_j [\nu_2]_j^+ - \hat{\nu}_1^\top x - \epsilon \left\| [\hat{\nu}_1]_1 \right\|_1$$

and g_θ *is a one-hidden layer neural network given by the equations*

$$\begin{aligned} &\nu_3 = -c \\ &\hat{\nu}_2 = W_2^\top \nu_3 \\ &[\nu_2]_j = 0,\ j \in \mathcal{I}^- \\ &[\nu_2]_j = [\hat{\nu}_2]_j,\ j \in \mathcal{I}^+ \\ &[\nu_2]_j = \tfrac{u_j}{u_j - \ell_j} [\hat{\nu}_2]_j^+ - \alpha_j [\hat{\nu}_2]_j^-,\ j \in \mathcal{I} \\ &\hat{\nu}_1 = W_1^\top \nu_2 \end{aligned}$$

where $\mathcal{I}^-, \mathcal{I}^+$ *and* $\mathcal{I}$ *denote the sets of activations in the hidden layer where* ℓ *and* u *are both negative, both positive or span zero, respectively.*

Further, we take the feasible solution $\alpha_j = \frac{u_j}{u_j - \ell_j}$, as proposed in Wong & Kolter (2018). Hence, we can represent the dual problem as a linear back propagation network, which provides a tractable solution for a lower bound of the primal objective.

A.2. Proof of Theorem 3

Theorem 3. *The dual of the linear program* (8) *can be written as*

$$\begin{array}{ll}\max\limits_{\alpha} & \widetilde{J}_\epsilon\left(x, g_\theta(c,\alpha)\right)\\ \textit{s.t.} & \alpha_j \in [0,1],\ \forall j\end{array}$$

where $\widetilde{J}_\epsilon(x,\nu_1,\nu_2,\nu_3)$ *is equal to*

$$-\sum_{i=1}^{2}\nu_{i+1}^\top b_i + \sum_{j\in\mathcal{I}}\ell_j[\nu_2]_j^+ - \hat{\nu}_1^\top x - \epsilon\left\|[\hat{\nu}_1]_1\right\|_1$$

and g_θ *is a one-hidden layer neural network given by the equations*

$$\begin{array}{l}\nu_3 = -c\\ \hat{\nu}_2 = W_2^\top\nu_3\\ [\nu_2]_j = 0,\ j\in\mathcal{I}^-\\ [\nu_2]_j = [\hat{\nu}_2]_j,\ j\in\mathcal{I}^+\\ [\nu_2]_j = \frac{u_j}{u_j-\ell_j}[\hat{\nu}_2]_j^+ - \alpha_j[\hat{\nu}_2]_j^-,\ j\in\mathcal{I}\\ \hat{\nu_1} = W_1^\top\nu_2\end{array}$$

where $\mathcal{I}^-,\mathcal{I}^+$ *and* $\mathcal{I}$ *denote the sets of activations in the hidden layer where* ℓ *and* u *are both negative, both positive or span zero, respectively.*

Proof. Consider a data point x and let $\tilde{x} = x+\delta$ be the adversarial perturbed data point. First, we explicit all the constraints for the linear program defined in (8):

$$\begin{array}{l}\min\limits_{\hat{z}_3}\ [\hat{z}_3]_y - [\hat{z}_3]_{\bar{y}} = c^\top\hat{z}_3\ , \quad \text{s.t.}\\ x+\delta\in\mathcal{B}_\epsilon(x)\\ z_1 = x+\delta\\ \hat{z}_2 = W_1z_1+b_1\\ \hat{z}_3 = W_2z_2+b_2\\ [z_2]_j = 0,\ \forall j\in\mathcal{I}^-\\ [z_2]_j = [\hat{z}_2]_j,\ \forall j\in\mathcal{I}^+\\ [z_2]_j \geq 0,\ \forall j\in\mathcal{I}\\ [z_2]_j \geq [\hat{z}_2]_j,\ \forall j\in\mathcal{I}\\ \left((u_j-\ell_j)\,[z_2]_j\ -u_j[\hat{z}_2]_j\right) \leq -u_j\ell_j,\ \forall j\in\mathcal{I}\end{array}$$

where $\mathcal{I}^-,\mathcal{I}^+$ and $\mathcal{I}$ denote the sets of activations in the hidden layer where ℓ and u are both negative, both positive, or span zero respectively. In order to compute the dual of this problem, we associate the following Lagrangian variables with each of the constraints:

$$\begin{array}{rl}\hat{z}_2 = W_1z_1+b_1 &\Rightarrow \nu_2\\ \hat{z}_3 = W_2z_2+b_2 &\Rightarrow \nu_3\\ z_1 = x+\delta &\Rightarrow \psi\\ -[z_2]_j \leq 0 &\Rightarrow \mu_j,\ \forall j\in\mathcal{I}\\ [\hat{z}_2]_j - [z_2]_j \leq 0 &\Rightarrow \tau_j,\ \forall j\in\mathcal{I}\\ \left((u_j-\ell_j)\,[z_2]_j - u_j[\hat{z}_2]_j\right) \leq -u_j\ell_j &\Rightarrow \lambda_j,\ \forall j\in\mathcal{I}\end{array}$$

note that we do not define explicit dual variables for $[z_2]_j = 0$ and $[z_2]_j = [\hat{z}_2]_j$ as we can easily eliminate them. We write

the Lagrangian as follows:

$$\begin{aligned}\mathcal{L}(z,\hat{z},\nu,\delta,\lambda,\tau,\mu,\psi) = &-\left(W_1^\top\nu_2+\psi\right)^\top z_1 - \sum_{j\in\mathcal{I}}\left(\mu_j+\tau_j-\lambda_j\left(u_j-\ell_j\right)+\left[W_2^\top\nu_3\right]_j\right)[z_2]_j \\ &+\sum_{j\in\mathcal{I}}\left(\tau_j-\lambda_j u_j+[\nu_2]_j\right)[\hat{z}_2]_j+\left(c+\nu_3\right)^\top\hat{z}_3-\sum_{i=1}^{2}\nu_{i+1}^\top b_i \\ &+\sum_{j\in\mathcal{I}}\lambda_j u_j\ell_j+\psi^\top x+\psi^\top\delta+\sum_{j\in\mathcal{I}^-}[\hat{z}_2]_j[\nu_2]_j \\ &+\sum_{j\in\mathcal{I}^+}[z_2]_j\left([\nu_2]_j-[W_2^\top\nu_3]_j\right) \\ \text{s.t. } &\tilde{x}\in\mathcal{B}_\epsilon(x)\end{aligned}$$

and we take the infimum w.r.t. $z,\hat{z},\delta$:

$$\begin{aligned}\inf_{z,\hat{z},\delta}\mathcal{L}(z,\hat{z},\nu,\delta,\lambda,\tau,\mu,\psi) = &-\inf_{z_2}\sum_{j\in\mathcal{I}}\left(\mu_j+\tau_j-\lambda_j\left(u_j-\ell_j\right)+\left[W_2^\top\nu_3\right]_j\right)[z_2]_j \\ &+\inf_{\hat{z}_2}\sum_{j\in\mathcal{I}}\left(\tau_j-\lambda_j u_j+[\nu_2]_j\right)[\hat{z}_2]_j+\inf_{\hat{z}_3}\left(c+\nu_3\right)^\top z_3-\sum_{i=1}^{2}\nu_{i+1}^\top b_i \\ &+\sum_{j\in\mathcal{I}}\lambda_j u_j\ell_j+\psi^\top x+\inf_{\tilde{x}\in\mathcal{B}_\epsilon(x)}\psi^\top\delta-\inf_{z_1}\left(W_1^\top\nu_2+\psi\right)^\top z_1 \\ &+\inf_{\hat{z}_2}\sum_{j\in\mathcal{I}^-}[\hat{z}_2]_j[\nu_2]_j+\inf_{z_2}\sum_{j\in\mathcal{I}^+}[z_2]_j\left([\nu_2]_j-[W_2^\top\nu_3]_j\right)\end{aligned}$$

Now, we can compute the infimum for the $\psi^\top\delta$ term:

$$\inf_{\tilde{x}\in\mathcal{B}_\epsilon(x)}\psi^\top\delta=\inf_{\|\beta\|_1\le\epsilon}\psi_1\cdot\beta=-\epsilon\cdot\|\psi_1\|_1$$

and since for all the other terms the infimum of a linear function is $-\infty$, except in the special case when it is identically zero, the infimum of $\mathcal{L}(\cdot)$ becomes:

$$\inf_{z,\hat{z},\delta}\mathcal{L}(.)=\begin{cases}-\sum_{i=1}^{2}\nu_{i+1}^\top b_i+\sum_{j\in\mathcal{I}}\lambda_j u_j\ell_j+\psi^\top x-\epsilon\|\psi_1\|_1 & \text{if conditions}\\ -\infty & \text{else}\end{cases}$$

where the conditions to satisfy are:

$$\begin{aligned}&\nu_3=-c\\ &W_1^\top\nu_2=-\psi\\ &[\nu_2]_j=0, j\in\mathcal{I}_i^-\\ &[\nu_2]_j=\left[W_2^\top\nu_3\right]_j, j\in\mathcal{I}_i^+\\ &\left.\begin{array}{ll}\left(u_j-\ell_j\right)\lambda_j-\mu_j-\tau_j & =\left[W_2^\top\nu_3\right]_j\\ [\nu_2]_j & =u_j\lambda_j-\tau_j\end{array}\right\}j\in\mathcal{I}\\ &\lambda,\tau,\mu\ge 0\end{aligned}$$

Thus, we can rewrite the dual problem as follows:

$$
\begin{aligned}
\max_{\nu,\psi,\lambda,\tau,\mu} \; & -\sum_{i=1}^{2} \nu_{i+1}^{\top} b_i + \sum_{j \in \mathcal{I}} \lambda_j u_j \ell_j + \psi^{\top} x - \epsilon \left\| \psi_1 \right\|_1 \\
\text{s.t.} \quad & \nu_3 = -c \\
& W_1^{\top} \nu_2 = -\psi \\
& [\nu_2]_j = 0, j \in \mathcal{I}_i^- \\
& [\nu_2]_j = \left[W_2^{\top} \nu_3 \right]_j, j \in \mathcal{I}_i^+ \\
& \left.
\begin{array}{ll}
(u_j - \ell_j) \lambda_j - \mu_j - \tau_j & = \left[W_2^{\top} \nu_3 \right]_j \\
[\nu_2]_j & = u_j \lambda_j - \tau_j
\end{array}
\right\} j \in \mathcal{I} \\
& \lambda, \tau, \mu \geq 0
\end{aligned}
$$

Note that the dual variable λ corresponds to the upper bounds in the ReLU relaxation, while μ and τ correspond to the lower bounds. By the complementarity property, we know that at the optimal solution, these variables will be zero if the ReLU constraint is non-tight, or non-zero if the ReLU constraint is tight. Since the upper and lower bounds cannot be tight simultaneously, either λ or $\mu + \tau$ must be zero. This means that at the optimal solution to the dual problem we can decompose $[W_2^{\top} \nu_3]_j$ into positive and negative parts since $(u_j - \ell_j)\lambda_j \geq 0$ and $\tau_j + \mu_j \geq 0$:

$$
\begin{aligned}
(u_j - \ell_j)\lambda_j &= [W_2^{\top} \nu_3]_j^+ \\
\tau_j + \mu_j &= [W_2^{\top} \nu_3]_j^-
\end{aligned}
$$

combining this with the constraint $[\nu_2]_j = u_j \lambda_j - \tau_j$ leads to

$$
[\nu_2]_j = \frac{u_j}{u_j - \ell_j} [W_2^{\top} \nu_3]_j^+ - \alpha_j [W_2^{\top} \nu_3]_j^-
$$

for $j \in \mathcal{I}$ and $0 \leq \alpha_j \leq 1$. Hence, we have that:

$$
\lambda_j = \frac{u_j}{u_j - \ell_j} [\hat{\nu}_2]_j^+
$$

Now, we denote $\hat{\nu}_1 = -\psi$ to make our notation consistent, and putting all of this together the dual objective becomes:

$$
\begin{aligned}
-\sum_{i=1}^{2} \nu_{i+1}^{\top} b_i + \sum_{j \in \mathcal{I}} \lambda_j u_j \ell_j + \psi^{\top} x - \epsilon \left\| \psi_1 \right\|_1 &= -\sum_{i=1}^{2} \nu_{i+1}^{\top} b_i + \sum_{j \in \mathcal{I}} \frac{u_j \ell_j}{u_j - \ell_j} [\hat{\nu}_2]_j^+ - \hat{\nu}_1^{\top} x - \epsilon \left\| [\hat{\nu}_1]_1 \right\|_1 \\
&= -\sum_{i=1}^{2} \nu_{i+1}^{\top} b_i + \sum_{j \in \mathcal{I}} \ell_j [\nu_2]_j^+ - \hat{\nu}_1^{\top} x - \epsilon \left\| [\hat{\nu}_1]_1 \right\|_1
\end{aligned}
$$

and the final dual problem:

$$
\begin{aligned}
\max_{\nu,\hat{\nu}} \quad & -\textstyle\sum_{i=1}^{2} \nu_{i+1}^{\top} b_i + \sum_{j \in \mathcal{I}} \ell_j [\nu_2]_j^+ - \hat{\nu}_1^{\top} x - \epsilon \left\| [\hat{\nu}_1]_1 \right\|_1 \\
\text{s.t.} \quad & \nu_3 = -c \\
& \hat{\nu_2} = W_2^{\top} \nu_3 \\
& [\nu_2]_j = 0, \; j \in \mathcal{I}^- \\
& [\nu_2]_j = [\hat{\nu}_2]_j, \; j \in \mathcal{I}^+ \\
& [\nu_2]_j = \tfrac{u_j}{u_j - \ell_j} [\hat{\nu}_2]_j^+ - \alpha_j [\hat{\nu}_2]_j^-, \; j \in \mathcal{I} \\
& \hat{\nu}_1 = W_1^{\top} \nu_2
\end{aligned}
$$

□

A.3. Computing upper and lower bounds on the pre-activations

We address here the problem of obtaining the upper and lower bounds u and ℓ for the pre-activations $\hat{z}$. Specifically, the following proposition gives a closed form solution.

Proposition 4. *Consider the neural network f_θ defined in Equation (7). Let w_1 be the first column of W_1. Then, for a data point x and perturbation budget ϵ, we have the following element-wise bounds on the pre-activation vector $\hat{z}_2$:*

$$\ell \leq \hat{z}_2 \leq u$$

where

$$\ell = W_1 x + b_1 - \epsilon |w_1| \;\; \textit{and} \;\; u = W_1 x + b_1 + \epsilon |w_1|$$

Proof. Given a data point x and perturbation budget ϵ, let $\tilde{x} = x + \delta$ be the perturbed input to the network. First, we find an upper bound the pre-activations values $\hat{z}_2$:

$$\hat{z}_2 = W_1(x + \delta) + b_1 = W_1 x + b_1 + W_1 \delta$$

In particular, we want to solve the following optimisation problem for each component of the pre-activation vector:

$$u_i = \max_{\tilde{x} \in \mathcal{B}_\epsilon(x)} [\hat{z}_2]_i = [W_1 x]_i + [b_1]_i + \max_{\tilde{x} \in \mathcal{B}_\epsilon(x)} [W_1 \delta]_i$$

where u will be the vector containing element-wise upper bounds. Note that $\delta = \beta e_1$, thus the optimisation problem can be rewritten as:

$$\max_{\tilde{x} \in \mathcal{B}_\epsilon(x)} [W_1 \delta]_i = \max_{\|\beta\|_1 \leq \epsilon} \beta \cdot [w_1]_i = \epsilon \cdot \|[w_1]_i\|_1$$

where w_1 is the first column of W_1. The vector of upper bounds will then be:

$$u = W_1 x + b_1 + \epsilon |w_1|$$

Along the same lines, we can derive the vector of lower bounds ℓ:

$$l = W_1 x + b_1 - \epsilon |w_1|$$

□

B. Theoretical results for signal-directed perturbations

In this section, we prove that convex relaxations along the signal direction hurt robust generalisation. We focus on the classification setting with binary cross-entropy loss:

$$L(x, y) = y \log\left(\boldsymbol{\sigma}\left(x\right)\right) + (1 - y) \log\left(1 - \boldsymbol{\sigma}\left(x\right)\right) \tag{9}$$

where $\boldsymbol{\sigma}\left(\cdot\right)$ is the sigmoid function.

First, in Lemma 1 we relate the robust error of the classifier f_θ to the signal parameter θ_1. Specifically, we show that robust error monotonically decreases in θ_1.

Lemma 1. *Let f_θ be the neural network defined in Equation (5) and $\mathcal{B}_\epsilon$ the threat model defined in Equation (4). We define the robust risk $\mathbf{R}_\epsilon$ of f_θ as follows:*

$$\mathbf{R}_\epsilon(\theta) := \mathbb{P}_{(x,y)}\left[\exists z \in \mathcal{B}_\epsilon(x) : y \neq \operatorname{sgn}\left(f_\theta(z)\right)\right]$$

Then, $\mathbf{R}_\epsilon(\theta)$ is monotonically decreasing in θ_1.

Proof. We assume, without loss of generality, that $\theta_1 > 0$, and since a and b are not trainable parameters we must have $a > 0$ and $b < 0$ to solve the classification problem. The robust risk is then equal to:

$$\begin{aligned}
\mathbf{R}_\epsilon(\theta) &:= \mathbb{P}_{(x,y)}\left[\exists z \in \mathcal{B}_\epsilon(x) : y \neq \operatorname{sgn}\left(f_\theta(z)\right)\right] \\
&= \frac{1}{2}\left(\mathbb{P}_x\left[a \operatorname{ReLU}\left(\theta^\top x\right) + b < 0 \mid y = 1\right] + \mathbb{P}_x\left[a \operatorname{ReLU}\left(\theta^\top x\right) + b > 0 \mid y = -1\right]\right)
\end{aligned}$$

further, we can remove the $\mathrm{ReLU}(\cdot)$ using the fact that $\frac{b}{a} < 0$:

$$\begin{aligned}\mathbf{R}_\epsilon(\theta) &= \frac{1}{2}\left(\mathbb{P}_x\left[\{x : \theta^\top x + \frac{b}{a} < 0 \vee \theta^\top x < 0\} \mid y = 1\right] + \mathbb{P}_x\left[\{x : \theta^\top x + \frac{b}{a} > 0 \wedge \theta^\top x > 0\} \mid y = -1\right]\right) \\ &= \frac{1}{2}\left(\mathbb{P}_x\left[\theta^\top x + \frac{b}{a} < 0 \mid y = 1\right] + \mathbb{P}_x\left[\theta^\top x + \frac{b}{a} > 0 \mid y = -1\right]\right) \\ &= \frac{1}{2}\left(\mathbb{P}_x\left[\sum_{i=2}^{d} x_i\theta_i < -\theta_1(\gamma - \epsilon) - \frac{b}{a}\right] + \mathbb{P}_x\left[\sum_{i=2}^{d} x_i\theta_i > \theta_1(\gamma - \epsilon) - \frac{b}{a}\right]\right)\end{aligned}$$

Recall now that for the linearly separable distribution we have $\sum_{i=2}^{d} x_i\theta_i \sim \mathcal{N}\left(0, \sigma^2 \left\|\theta_{2:d}\right\|^2\right)$. Therefore, we can replace the probabilities with the standard normal CDF Φ:

$$\mathbf{R}_\epsilon(\theta) = \frac{1}{2}\left(\Phi\left(-\frac{(\gamma - \epsilon)\theta_1}{\sigma\left\|\theta_{2:d}\right\|_2} - \frac{b}{a\sigma\left\|\theta_{2:d}\right\|_2}\right) + \Phi\left(-\frac{(\gamma - \epsilon)\theta_1}{\sigma\left\|\theta_{2:d}\right\|_2} + \frac{b}{a\sigma\left\|\theta_{2:d}\right\|_2}\right)\right)$$

hence $\mathbf{R}_\epsilon(\theta)$ is monotonically decreasing in θ_1 and the statement follows. □

B.1. Adversarial training

The basic idea behind adversarial training is to update the network parameters according to the following rule:

$$\theta \leftarrow \theta - \frac{\eta}{|D|}\sum_{(x,y)\in D} \nabla_\theta \max_{x+\delta\in\mathcal{B}_\epsilon(x)} L(f_\theta(x+\delta), y)$$

This is usually done by applying some first-order approximation to the maximisation problem. However, for our simplified network we can analytically compute the gradient. First of all, note that when L is the binary cross-entropy loss function we can rewrite the maximisation problem as follows:

$$\max_{x+\delta\in\mathcal{B}_\epsilon(x)} L(f_\theta(x+\delta), y) = L\left(\mathrm{sgn}(y)\overbrace{\min_{x+\delta\in\mathcal{B}_\epsilon(x)} \mathrm{sgn}(y) f_\theta(x+\delta)}^{:=J_\epsilon(x,y)},\ y\right) \tag{10}$$

In particular, if J_ϵ is strictly positive then no adversarial example exists that fools the network. Further, note that this formulation is closely related to the objective considered in Appendix A.2 for the convex outer adversarial polytope. This will be useful when comparing COAP and AT gradients. Below we provide the gradient of the adversarial training objective w.r.t. the network parameters θ.

Proposition 5. *Consider the neural network f_θ defined in Equation (5) and the threat model $\mathcal{B}_\epsilon$ defined in Equation (4). Let L be the binary cross-entropy loss function, as defined in Equation (9). Then, we have:*

$$\begin{aligned}&\nabla_{\theta_1} \max_{x+\delta\in\mathcal{B}_\epsilon(x)} L(f_\theta(x+\delta), y) \\ &= -\mathrm{sgn}(y)\boldsymbol{\sigma}\left(-J_\epsilon(x,y)\right)\begin{cases} a(x_1 - \epsilon\,\mathrm{sgn}(\theta_1))\mathbf{1}\{\ell > 0\} & \text{if } a\,\mathrm{sgn}(y) > 0 \\ a(x_1 + \epsilon\,\mathrm{sgn}(\theta_1))\mathbf{1}\{u > 0\} & \text{if } a\,\mathrm{sgn}(y) < 0 \end{cases}\end{aligned}$$

where $\ell = \theta^\top x - \epsilon|\theta_1|$ and $u = \theta^\top x + \epsilon|\theta_1|$ are respectively lower and upper bounds on the ReLU *inputs.*

Proof. Given a data point x with known label $y \in \{-1, 1\}$, when L is the binary cross-entropy loss function we have:

$$\max_{x+\delta\in\mathcal{B}_\epsilon(x)} L(f_\theta(x+\delta), y) = L\left(\mathrm{sgn}(y)\min_{x+\delta\in\mathcal{B}_\epsilon(x)} \mathrm{sgn}(y) f_\theta(x+\delta),\ y\right)$$

For our simplified network we can analytically compute a closed form solution of the minimisation problem:

$$
\begin{aligned}
J_\epsilon &:= \min_{x+\delta \in \mathcal{B}_\epsilon(x)} \operatorname{sgn}(y)\left(b + a\,\mathrm{ReLU}\left(\theta^\top(x+\delta)\right)\right) \\
&= \begin{cases} \operatorname{sgn}(y)\left(b + a\max(0,\ell)\right) & \text{if } \ a\operatorname{sgn}(y) > 0 \\ \operatorname{sgn}(y)\left(b + a\max(0,u)\right) & \text{if } \ a\operatorname{sgn}(y) < 0 \end{cases} \\
&= \begin{cases} \operatorname{sgn}(y)\left(b + a\max(0,\ell)\right) & \text{if } \ a\operatorname{sgn}(y) > 0 \\ \operatorname{sgn}(y)\left(b + a\max(0,u)\right) & \text{if } \ a\operatorname{sgn}(y) < 0 \end{cases}
\end{aligned}
$$

where $\ell = \theta^\top x - \epsilon|\theta_1|$ and $u = \theta^\top x + \epsilon|\theta_1|$ are respectively lower and upper bounds on the pre-activations. Thus, we can compute the gradients for adversarial training w.r.t the signal parameter:

$$
\frac{\partial}{\partial \theta_1} J_\epsilon = \begin{cases} \operatorname{sgn}(y) a (x_1 - \epsilon \operatorname{sgn}(\theta_1)) \mathbf{1}\{\ell > 0\} & \text{if } \ a\operatorname{sgn}(y) > 0 \\ \operatorname{sgn}(y) a (x_1 + \epsilon \operatorname{sgn}(\theta_1)) \mathbf{1}\{u > 0\} & \text{if } \ a\operatorname{sgn}(y) < 0 \end{cases}
$$

and applying the chain-rule we have:

$$
\begin{aligned}
&\frac{\partial}{\partial \theta_1} L\left(\operatorname{sgn}(y) J_\epsilon, y\right) \\
&= \frac{\partial}{\partial J_\epsilon} L\left(\operatorname{sgn}(y) J_\epsilon, y\right) \cdot \frac{\partial}{\partial \theta_1} J_\epsilon \\
&= \operatorname{sgn}(y)\left[\boldsymbol{\sigma}\left(\operatorname{sgn}(y) J_\epsilon\right) - \mathbf{1}\{y = 1\}\right] \cdot \frac{\partial}{\partial \theta_1} J_\epsilon \\
&= -\operatorname{sgn}(y) \boldsymbol{\sigma}\left(-J_\epsilon\right) \begin{cases} a(x_1 - \epsilon \operatorname{sgn}(\theta_1)) \mathbf{1}\{\ell > 0\} & \text{if } \ a\operatorname{sgn}(y) > 0 \\ a(x_1 + \epsilon \operatorname{sgn}(\theta_1)) \mathbf{1}\{u > 0\} & \text{if } \ a\operatorname{sgn}(y) < 0 \end{cases}
\end{aligned}
$$

where in the last equality we use a known property of the sigmoid function, $\boldsymbol{\sigma}(x) = 1 - \boldsymbol{\sigma}(-x)$. □

B.2. Convex outer adversarial polytope

We now consider the dual approximation $\widetilde{J}_\epsilon$ to the optimisation problem in Equation (10). Note that, for a binary classification problem, we have $c = \operatorname{sgn}(y)$ and the dual objective in Theorem 3 becomes:

$$
\widetilde{J}_\epsilon\left(x, g_\theta(c, \alpha)\right) = \widetilde{J}_\epsilon\left(x, y\right) \tag{11}
$$

where we set α to the dual feasible solution and for the sake of clarity we omit the dependence on the network parameters θ.

We are particularly interested in the data points for which $J_\epsilon(x, y) \neq \widetilde{J}_\epsilon(x, y)$, i.e., when the certified and adversarial training objectives differ. Below, we provide a necessary and sufficient condition to have a mismatch between the two objectives.

Proposition 6. *Consider the neural network f_θ defined in Equation (5) and the threat model $\mathcal{B}_\epsilon$ defined in Equation (4). Let L be the binary cross-entropy loss function, as defined in Equation (9). Further, we define $\ell = \theta^\top x - \epsilon|\theta_1|$ and $u = \theta^\top x + \epsilon|\theta_1|$ respectively as lower and upper bounds on the* ReLU *inputs. Let $\mathcal{I}^\star = \{(x, y) : 0 \in [\ell, u] \wedge a\operatorname{sgn}(y) > 0\}$. Then, for data points in $\mathcal{I}^\star$, we have that AT and COAP gradients differ:*

$$
\nabla_{\theta_1} J_\epsilon(x, y) \neq \nabla_{\theta_1} \widetilde{J}_\epsilon(x, y) \ \ \forall (x, y) \in \mathcal{I}^\star
$$

and COAP gradient is given by:

$$
\begin{aligned}
&\nabla_{\theta_1} L(\operatorname{sgn}(y) \widetilde{J}_\epsilon(x, y), y) \\
&= -\frac{a\operatorname{sgn}(y) \boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon(x, y)\right)}{2\epsilon} \left(\frac{\ell}{\|\theta_1\|_1} (x_1 + \epsilon \operatorname{sgn}(\theta_1)) + u \frac{x_1 \|\theta_1\|_1 - \theta^\top x \operatorname{sgn}(\theta_1)}{\theta_1^2} \right)
\end{aligned}
$$

Further, for data points that are not in $\mathcal{I}^\star$ we have that AT and COAP gradients are equivalent:

$$
\nabla_{\theta_1} J_\epsilon(x, y) = \nabla_{\theta_1} \widetilde{J}_\epsilon(x, y) \ \ \forall (x, y) \notin \mathcal{I}^\star
$$

Proof. For the sake of clarity, we report here the definition of COAP objective from Appendix A.2.

$$\widetilde{J}_\epsilon(x,y) = -\sum_{i=1}^{2} \nu_{i+1}^\top b_i + \sum_{j\in\mathcal{I}} \ell_j [\nu_2]_j^+ - \hat{\nu}_1^\top x - \epsilon \left\| [\hat{\nu}_1]_1 \right\|_1$$

Further, recall that the dual variables ν are given by the following equations:

$$\begin{aligned}
&\nu_3 = -c \\
&\hat{\nu}_2 = W_2^\top \nu_3 \\
&[\nu_2]_j = 0,\ j \in \mathcal{I}^- \\
&[\nu_2]_j = [\hat{\nu}_2]_j,\ j \in \mathcal{I}^+ \\
&[\nu_2]_j = \tfrac{u_j}{u_j - \ell_j} [\hat{\nu}_2]_j^+ - \alpha_j [\hat{\nu}_2]_j^-,\ j \in \mathcal{I} \\
&\hat{\nu}_1 = W_1^\top \nu_2
\end{aligned}$$

where $\mathcal{I}^-, \mathcal{I}^+$ and $\mathcal{I}$ denote the sets of activations in the hidden layer where ℓ and u are both negative, both positive and span zero, respectively.

First, we consider the case when the neuron is always dead, i.e., $\ell < u < 0$. The dual variables are:

$$\begin{aligned}
\nu_3 &= -\operatorname{sgn}(y) \\
\hat{\nu}_2 &= -a \operatorname{sgn}(y) \\
\nu_2 &= 0 \\
\hat{\nu}_1 &= 0
\end{aligned}$$

Hence, AT and COAP objectives are equal in this case:

$$\widetilde{J}_\epsilon = \operatorname{sgn}(y) b = J_\epsilon$$

where the last equality follows from Appendix B.1.

Next, we consider the case when the neuron is always active, i.e., $0 < \ell < u$. The dual variables are:

$$\begin{aligned}
\nu_3 &= -\operatorname{sgn}(y) \\
\hat{\nu}_2 &= -a \operatorname{sgn}(y) \\
\nu_2 &= -a \operatorname{sgn}(y) \\
\hat{\nu}_1 &= -a \operatorname{sgn}(y) \cdot \theta
\end{aligned}$$

and the dual objective becomes:

$$\begin{aligned}
\widetilde{J}_\epsilon &= -\nu_3^\top b - \hat{\nu}_1^\top x - \epsilon \left\| [\hat{\nu}_1]_1 \right\|_1 \\
&= \operatorname{sgn}(y) \left(b + a(\theta^\top x) \right) - \epsilon \| a \operatorname{sgn}(y) \theta_1 \| \\
&= \begin{cases} \operatorname{sgn}(y)\,(b + a\ell) & \text{if } \ a \operatorname{sgn}(y) > 0 \\ \operatorname{sgn}(y)\,(b + au) & \text{if } \ a \operatorname{sgn}(y) < 0 \end{cases} \\
&= J_\epsilon
\end{aligned}$$

where the last equality follows from the fact that $0 < \ell < u$.

Finally, we consider the case when the neuron is in the activation set $\mathcal{I}$, i.e., $\ell < 0 < u$. The dual variables are:

$$\begin{aligned}
\nu_3 &= -\operatorname{sgn}(y) \\
\hat{\nu}_2 &= -a \operatorname{sgn}(y) \\
\nu_2 &= \ -a \operatorname{sgn}(y) \frac{u}{2\epsilon \left\| \theta_1 \right\|_1} \\
\hat{\nu}_1 &= -a \operatorname{sgn}(y) \ \frac{u}{2\epsilon \left\| \theta_1 \right\|_1} \cdot \theta
\end{aligned}$$

Here we have two cases, when $\hat{\nu}_2 > 0$ we can rewrite the dual objective as:

$$\widetilde{J}_\epsilon = \operatorname{sgn}(y)\,(b + au) = J_\epsilon$$

and the two objectives coincide.

When $\nu_2 < 0$ we can rewrite the dual objective as:

$$\widetilde{J}_\epsilon = \operatorname{sgn}(y)\left(b + \frac{au\ell}{2\epsilon\,\|\theta_1\|_1}\right) \neq J_\epsilon$$

Hence, the only case when COAP gradient differs from AT gradient is when $\nu_2 < 0$ and the neuron belongs to the activation set $\mathcal{I}$.

We compute the partial derivative w.r.t. the signal parameter θ_1 in this case, by the chain rule we have:

$$\begin{aligned}
&\frac{\partial}{\partial\theta_1} L\left(\operatorname{sgn}(y)\cdot\widetilde{J}_\epsilon, y\right)\\
&= \frac{\partial}{\partial\widetilde{J}_\epsilon} L\left(\operatorname{sgn}(y)\cdot\widetilde{J}_\epsilon, y\right)\cdot\frac{\partial}{\partial\theta_1}\widetilde{J}_\epsilon\\
&= \operatorname{sgn}(y)\left[\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\cdot\widetilde{J}_\epsilon\right) - \mathbf{1}\{y=1\}\right]\cdot\frac{\partial}{\partial\theta_1}\widetilde{J}_\epsilon\\
&= -\frac{a\operatorname{sgn}(y)\boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon\right)}{2\epsilon}\left(\frac{\ell}{\|\theta_1\|_1}(x_1 + \epsilon\operatorname{sgn}(\theta_1)) + u\frac{x_1\|\theta_1\|_1 - \theta^\top x\operatorname{sgn}(\theta_1)}{\theta_1^2}\right)
\end{aligned}$$

□

B.3. Proof of Theorem 1

Theorem 1. *Let $\mathbf{R}_\epsilon$ be the robust risk of f_θ, defined as*

$$\mathbf{R}_\epsilon(\theta) := \mathbb{P}_{(x,y)}\left[\exists z \in \mathcal{B}_\epsilon(x) : y \neq \operatorname{sgn}\left(f_\theta(z)\right)\right].$$

Let $\bar{\theta}$ and $\tilde{\theta}$ be the network parameters after one step of gradient descent with respect to AT and COAP objectives. If the initialised network parameters θ satisfy

$$\frac{\|\theta_{2:d}\|_2}{\|\theta_1\|_2} > \sqrt{\max\left(\frac{(7\epsilon-\gamma)(\gamma+\epsilon)^4}{4\sigma^2(\gamma^2-10\gamma\epsilon+13\epsilon^2)}, \frac{(\gamma+\epsilon)^3}{12\sigma^2\epsilon}\right)}$$

and $\frac{2}{3}\gamma < \epsilon < \gamma$, COAP yields higher robust risk than AT:

$$\mathbf{R}_\epsilon(\tilde{\theta}) > \mathbf{R}_\epsilon(\bar{\theta}).$$

Proof. First we assume, without loss of generality, that at initialisation $\theta_1 > 0$, and since a and b are not trainable parameters we must have $a > 0$ and $b < 0$ to include the ground truth in our hypothesis class.

Let J_ϵ be the adversarial training inner maximisation as defined in Equation (10). Then, AT solves the following optimisation problem:

$$\min_\theta \mathbb{E}_{(x,y)}\left[L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)J_\epsilon(x,y)\right), y\right)\right]$$

Similarly, let $\widetilde{J}_\epsilon$ be the COAP dual approximation to the inner maximization described in Equation (11). Then, COAP solves the following optimisation problem:

$$\min_\theta \mathbb{E}_{(x,y)}\left[L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\widetilde{J}_\epsilon\right), y\right)\right]$$

Since we are only training the signal parameter θ_1, after one gradient descent step, we have:

$$\left\|\bar{\theta}_{2:d}\right\|_2 = \left\|\tilde{\theta}_{2:d}\right\|_2$$

Further, from Lemma 1 we know that AT yields smaller robust risk than COAP if the following holds:

$$\bar{\theta}_1 > \tilde{\theta}_1 \implies \mathbf{R}_\epsilon(\tilde{\theta}) > \mathbf{R}_\epsilon(\bar{\theta})$$

which, after one step of gradient descent, is equivalent to:

$$\mathbb{E}_{(x,y)}\left[\nabla_{\theta_1} L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y) J_\epsilon(x,y)\right), y\right)\right] < \mathbb{E}_{(x,y)}\left[\nabla_{\theta_1} L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\widetilde{J}_\epsilon(x,y)\right), y\right)\right]$$

Now recall from Theorems 5 and 6 that the gradients of AT and COAP differ only on the set $\mathcal{I}^\star$. In particular, we have that:

$$(x,y) \notin \mathcal{I}^\star \implies \nabla_{\theta_1} L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y) J_\epsilon(x,y)\right), y\right) = \nabla_{\theta_1} L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\widetilde{J}_\epsilon(x,y)\right), y\right) < 0$$

and

$$(x,y) \in \mathcal{I}^\star \implies 0 = \nabla_{\theta_1} L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y) J_\epsilon(x,y)\right), y\right) \neq \nabla_{\theta_1} L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\widetilde{J}_\epsilon(x,y)\right), y\right)$$

Hence, for our purpose we need to show that:

$$\mathbb{E}_{(x,y)}\left[\nabla_{\theta_1} L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\widetilde{J}_\epsilon(x,y)\right), y\right) \mid (x,y) \in \mathcal{I}^\star\right] > 0 \tag{12}$$

Our strategy will be to lower-bound the expectation in Equation (12) with some strictly positive quantity. We define

$$Z = \sum_{i=2}^{d} \theta_i x_i$$

and plug-in the gradient computed in Theorem 6:

$$\begin{aligned}
&\mathbb{E}_{(x,y)}\left[\nabla_{\theta_1} L\left(\operatorname{sgn}(y)\boldsymbol{\sigma}\left(\widetilde{J}_\epsilon(x,y)\right), y\right) \mid (x,y) \in \mathcal{I}^\star\right] \\
&= \mathbb{E}_{(x,y)}\left[\frac{a\boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon(x,y)\right)}{2\epsilon}\left(-\frac{\ell}{\theta_1}(\gamma+\epsilon) + u\frac{\sum_{i=2}^{d} x_i\theta_i}{\theta_1^2}\right) \mid (x,y) \in \mathcal{I}^\star\right] \\
&= \frac{a}{2\theta_1\epsilon}\mathbb{E}_{(x,y)}\left[\boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon(x,y)\right)\left(-\ell(\gamma+\epsilon) + u\frac{Z}{\theta_1}\right) \mid (x,y) \in \mathcal{I}^\star\right] \\
&= \frac{a}{2\theta_1\epsilon}\mathbb{E}_{(x,y)}\left[\boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon(x,y)\right) u\frac{Z}{\theta_1} - \boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon(x,y)\right)\ell(\gamma+\epsilon) \mid (x,y) \in \mathcal{I}^\star\right]
\end{aligned}$$

Now, we observe that Z is always negative on the set $\mathcal{I}^\star$, since we need to satisfy the constraint $\ell < 0 < u$:

$$(x,y) \in \mathcal{I}^\star \implies -\theta_1(\gamma+\epsilon) < \sum_{i=2}^{d} \theta_i x_i < -\theta_1(\gamma-\epsilon) < 0$$

Further, from Appendix B.2 we have:

$$(x,y) \in \mathcal{I}^\star \implies \boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon(x,y)\right) \geq \frac{1}{2}$$

Combining these two observations we can lower-bound the expectation:

$$\begin{aligned}
&\mathbb{E}_{(x,y)}\left[\nabla_{\theta_1} L\left(\operatorname{sgn}(y)\boldsymbol{\sigma}\left(\widetilde{J}_\epsilon(x,y)\right), y\right) \mid (x,y) \in \mathcal{I}^\star\right] \\
&= \frac{a}{2\theta_1\epsilon}\mathbb{E}_{(x,y)}\left[\boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon(x,y)\right) u\frac{Z}{\theta_1} - \boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon(x,y)\right)\ell(\gamma+\epsilon) \mid (x,y) \in \mathcal{I}^\star\right] \\
&\geq \frac{a}{2\theta_1\epsilon}\mathbb{E}_{(x,y)}\left[u\frac{Z}{\theta_1} - \frac{\gamma+\epsilon}{2}\ell \mid (x,y) \in \mathcal{I}^\star\right]
\end{aligned}$$

Now, we need to show that this lower-bound is strictly positive:

$$\mathbb{E}_{(x,y)}\left[u\frac{Z}{\theta_1} - \frac{\gamma+\epsilon}{2}\ell \mid (x,y) \in \mathcal{I}^\star\right] > 0$$

Note that, we can further expand this expression:

$$\begin{aligned}
&\mathbb{E}_{(x,y)}\left[u\frac{Z}{\theta_1} - \frac{\gamma+\epsilon}{2}\ell \mid (x,y) \in \mathcal{I}^\star\right] \\
&= -(\gamma^2-\epsilon^2)\theta_1^2 + (\gamma+\epsilon)\theta_1\mathbb{E}\left[Z \mid (x,y) \in \mathcal{I}^\star\right] + 2\mathbb{E}\left[Z^2 \mid (x,y) \in \mathcal{I}^\star\right]
\end{aligned}$$

Further, $Z \mid (x,y) \in \mathcal{I}^\star$ is distributed as a truncated normal with:

$$\alpha = -\frac{\theta_1(\gamma+\epsilon)}{\sigma\left\|\theta_{2:d}\right\|_2} \text{ and } \beta = -\frac{\theta_1(\gamma-\epsilon)}{\sigma\left\|\theta_{2:d}\right\|_2}$$

Hence, we can plug in the expectations of the truncated normal distribution to obtain the following:

$$\begin{aligned}
&-(\gamma^2-\epsilon^2)\theta_1^2 + \theta_1(\gamma+\epsilon)\mathbb{E}\left[Z \mid (x,y) \in \mathcal{I}^\star\right] + 2\mathbb{E}\left[Z^2 \mid (x,y) \in \mathcal{I}^\star\right] \\
&= -(\gamma^2-\epsilon^2)\theta_1^2 + 2\sigma^2\left\|\theta_{2:d}\right\|_2^2 + \sigma\left\|\theta_{2:d}\right\|_2\theta_1\frac{(\gamma-3\epsilon)\phi(\beta) - (\gamma+\epsilon)\phi(\alpha)}{\Phi(\beta)-\Phi(\alpha)} \\
&\propto -(\gamma^2-\epsilon^2) + 2\sigma^2 r^2 + \sigma r\frac{(\gamma-3\epsilon)\phi(\beta) - (\gamma+\epsilon)\phi(\alpha)}{\Phi(\beta)-\Phi(\alpha)} \\
&= -f(r)
\end{aligned}$$

where we define $r = \frac{\left\|\theta_{2:d}\right\|_2}{\left\|\theta_1\right\|_2}$. Now, under our assumptions, from Lemma 4 we have:

$$f(r) < 0,\ \forall r > \sqrt{\max\left(\frac{(7\epsilon-\gamma)(\gamma+\epsilon)^4}{4\sigma^2(\gamma^2-10\gamma\epsilon+13\epsilon^2)}, \frac{(\gamma+\epsilon)^3}{12\sigma^2\epsilon}\right)}$$

and hence it follows that $\mathbf{R}_\epsilon(\tilde{\theta}) > \mathbf{R}_\epsilon(\bar{\theta})$.

Next, we prove that the accuracy discrepancy is higher for COAP, that is:

$$\max_k \mathbf{R}^k(\tilde{\theta}) - \mathbf{R}(\tilde{\theta}) > \max_k \mathbf{R}^k(\bar{\theta}) - \mathbf{R}(\bar{\theta})$$

As we already showed that $\bar{\theta}_1 > \tilde{\theta}_1$, our strategy will be to prove that the accuracy discrepancy is a decreasing function of θ_1. First, note that for $\theta_1 >$ we have from Lemma 1:

$$\begin{aligned}
\max_k \mathbf{R}^k(\theta) - \mathbf{R}(\theta) &= \mathbf{R}_\epsilon^1(\theta) - \frac{\mathbf{R}_\epsilon^1(\theta) - \mathbf{R}_\epsilon^{-1}(\theta)}{2} \\
&\propto \mathbf{R}_\epsilon^1(\theta) - \mathbf{R}_\epsilon^{-1}(\theta) := a(\theta)
\end{aligned}$$

Next, we take the derivative of $a(\theta)$ w.r.t. θ_1 and show that it is negative:

$$\begin{aligned}
\frac{\partial}{\partial\theta_1}a(\theta) &= \frac{\partial}{\partial\theta_1}\left[\Phi(-c_1\theta_1 + c_2) - \Phi(-c_1\theta_1 - c_2)\right] \\
&= c_1\left[\phi(-c_1\theta_1 - c_2) - \phi(-c_1\theta_1 + c_2)\right] \\
&< 0 \ \ \forall\theta_1, c_1, c_2 > 0
\end{aligned}$$

which concludes the proof. □

C. Auxiliary lemmas

C.1. Upper bound on the exponential function

Lemma 2. *Let $f : \mathbb{R} \to \mathbb{R}$ be the function defined by $f(x) = \exp(x)$. When $x \leq 0$ and n is even we have:*

$$f(x) \leq 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!}$$

Proof. Let $g : (-\infty, 0] \to \mathbb{R}$ be the function defined by

$$g(x) = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} - \exp(x)$$

Since $g(x) \to \infty$ as $x \to -\infty$, g must attain an absolute minimum somewhere on the interval $(-\infty, 0]$. Now, differentiating we have:

- If g has an absolute minimum at 0 , then for all $x, g(x) \geq g(0) = 1 - \exp(0) = 0$, so we are done.
- If g has an absolute minimum at y for some $y < 0$, then $g'(y) = 0$. But differentiating,

  $$g'(y) = 1 + y + \frac{y^2}{2!} + \cdots + \frac{y^{n-1}}{(n-1)!} - \exp(y) = g(y) - \frac{y^n}{n!}.$$

  Therefore, for any x,

  $$g(x) \geq g(y) = \frac{y^n}{n!} + g'(y) = \frac{y^n}{n!} > 0,$$

  since n is even.

□

C.2. Lower bound on the difference of Gaussian CDFs

Lemma 3. *Let $f : \mathbb{R}^2 \to \mathbb{R}$ be the function defined by $f(x, y) = \Phi(y) - \Phi(x)$. When $x < y < 0$ we have:*

$$\phi(0)\left(y - x + \frac{x^3}{6}\right) \leq \Phi(y) - \Phi(x)$$

where Φ and ϕ are respectively the CDF and PDF of the standard Gaussian distribution.

Proof. First, we want to prove that $\frac{2x}{\sqrt{\pi}}$ is a lower bound for the error function $\operatorname{erf}(x)$ when $x \leq 0$. That is, we want to show that $f(x) \geq 0$ where $f : (-\infty, 0] \to \mathbb{R}$ is the function defined by:

$$f(x) = \operatorname{erf}(x) - \frac{2x}{\sqrt{\pi}}$$

Since f is continuous and $f(x) \to \infty$ as $x \to -\infty$, f must attain an absolute minimimum on the interval $(-\infty, 0]$. Now, differentiating we have:

$$f'(x) = \frac{2}{\sqrt{\pi}} \exp(-x^2) - \frac{2}{\sqrt{\pi}}$$

hence f attains an absolute minimum at 0 and we have $f(x) \geq f(0) = 0$.
Next, we show that $\frac{2}{\sqrt{\pi}}(x - x^3/3)$ is an upper bound for $\operatorname{erf}(x)$ when $x \leq 0$. Let $g : (-\infty, 0] \to \mathbb{R}$ the function defined by:

$$g(x) = \frac{2}{\sqrt{\pi}}(x - x^3/3) - \operatorname{erf}(x)$$

Similarly, since g is continuous and $g(x) \to \infty$ as $x \to -\infty$, g must attain an absolute minimimum on the interval $(-\infty, 0]$. Now, differentiating we have:

$$g'(x) = \frac{2}{\sqrt{\pi}}(1 - x^2 - \exp(-x^2))$$

hence g attains an absolute minimum at 0 and we have $g(x) \geq g(0) = 0$.
Now, since $a < b < 0$ we can use the erf bounds derived above:

$$\begin{aligned}\Phi(b) - \Phi(a) &= \frac{1}{2}\left(\mathrm{erf}(b/\sqrt{2}) - \mathrm{erf}(a/\sqrt{2})\right) \\ &\geq \frac{1}{\sqrt{\pi}}\left(\frac{b}{\sqrt{2}} - \frac{a}{\sqrt{2}} + \frac{a^3}{6\sqrt{2}}\right) \\ &= \phi(0)\left(b - a + \frac{a^3}{6}\right)\end{aligned}$$

which concludes the proof. □

C.3. Upper bound on the ratio of Gaussian PDFs and CDFs

Lemma 4. *Suppose $f : \mathbb{R} \to \mathbb{R}$ is defined as follows:*

$$f(r) = \gamma^2 - \epsilon^2 - 2\sigma^2 r^2 - \sigma r \frac{(\gamma - 3\epsilon)\phi(\beta) - (\gamma + \epsilon)\phi(\alpha)}{\Phi(\beta) - \Phi(\alpha)}$$

where $\alpha := -\frac{\gamma+\epsilon}{r\sigma}$, $\beta := -\frac{\gamma-\epsilon}{r\sigma}$, Φ and ϕ are respectively the standard Gaussian CDF and PDF.
Assume that:

$$\frac{5 + 2\sqrt{3}}{13}\gamma < \epsilon < \gamma$$

Then, we have:

$$f(r) < 0, \;\; \forall r > \sqrt{\max\left(\frac{(7\epsilon - \gamma)(\gamma + \epsilon)^4}{4\sigma^2(\gamma^2 - 10\gamma\epsilon + 13\epsilon^2)}, \frac{(\gamma + \epsilon)^3}{12\sigma^2\epsilon}\right)}$$

Proof. We begin by providing a lower bound on the difference of gaussian cdfs. Applying Lemma 3 with $x = \alpha$ and $y = \beta$ we have:

$$\Phi(\beta) - \Phi(\alpha) \geq \left(\frac{2\epsilon}{r\sigma} - \frac{(\gamma + \epsilon)^3}{6\sigma^3 r^3}\right)\phi(0), \;\; \alpha < \beta < 0$$

Next, we can upper-bound f:

$$\begin{aligned}f(r) &\leq \gamma^2 - \epsilon^2 - 2\sigma^2 r^2 - \sigma r \frac{(\gamma - 3\epsilon)\phi(\beta) - (\gamma + \epsilon)\phi(\alpha)}{\left(\frac{2\epsilon}{\sigma r} - \frac{(\gamma+\epsilon)^3}{6\sigma^3 r^3}\right)\phi(0)} \\ &\leq \gamma^2 - \epsilon^2 - 2\sigma^2 r^2 - \sigma^2 r^2 \frac{(\gamma - 3\epsilon)\phi(0) - (\gamma + \epsilon)\phi(\alpha)}{\left(2\epsilon - \frac{(\gamma+\epsilon)^3}{6r^2\sigma^2}\right)\phi(0)} \\ &= \gamma^2 - \epsilon^2 - 2\sigma^2 r^2 - \sigma^2 r^2 \frac{(\gamma - 3\epsilon) - (\gamma + \epsilon)\exp(-\alpha^2/2)}{2\epsilon - \frac{(\gamma+\epsilon)^3}{6\sigma^2 r^2}}\end{aligned}$$

Now, we use the upper-bound for the exponential function from Lemma 2 with $n = 2$:

$$\exp(x) \leq 1 + x - x^2/2, \;\; \forall x \leq 0$$

and substituting it back into our upper-bound for f we get:

$$f(r) \leq \gamma^2 - \epsilon^2 - 2\sigma^2 r^2 - \sigma^2 r^2 \frac{(\gamma - 3\epsilon) - (\gamma + \epsilon)(1 - \frac{(\gamma+\epsilon)^2}{2r^2\sigma^2} + \frac{(\gamma+\epsilon)^4}{8r^4\sigma^4})}{2\epsilon - \frac{(\gamma+\epsilon)^3}{6r^2\sigma^2}}$$

which can be further simplified:

$$
\begin{aligned}
f(r) &\leq \gamma^2 - \epsilon^2 - 2\sigma^2 r^2 - \sigma^2 r^2 \frac{(\gamma - 3\epsilon) - (\gamma + \epsilon)(1 - \frac{(\gamma+\epsilon)^2}{2r^2\sigma^2} + \frac{(\gamma+\epsilon)^4}{8r^4\sigma^4})}{2\epsilon - \frac{(\gamma+\epsilon)^3}{6r^2\sigma^2}} \\
&= \frac{(\gamma - 7\epsilon)(\gamma+\epsilon)^4 + 4r^2\sigma^2(\gamma+\epsilon)(\gamma^2 - 10\gamma\epsilon + 13\epsilon^2)}{4(\gamma+\epsilon)^3 - 48r^2\sigma^2\epsilon} \\
&= u(r)
\end{aligned}
$$

and we have that for $\epsilon > \frac{5+2\sqrt{3}}{13}\gamma$ and $r > \sqrt{\max\left(\frac{(7\epsilon-\gamma)(\gamma+\epsilon)^4}{4\sigma^2(\gamma^2-10\gamma\epsilon+13\epsilon^2)}, \frac{(\gamma+\epsilon)^3}{12\sigma^2\epsilon}\right)}$ the upper bound is negative, i.e. $u(r) < 0$. □

D. Experimental details

D.1. Synthetic experiments with signal-directed perturbations

Below we provide detailed experimental details to reproduce Figure 6.

Data generation For the linearly separable distribution we set $d = 1000,\ n_{\text{train}} = 10^4, n_{\text{test}} = 10^5,\ \gamma = 6$.

Model and hyper-parameters For all the experiments, we use the one hidden layer architecture defined in Equation (7) with 100 neurons. We use PyTorch SGD optimiser and train all networks for 100 epochs. We sweep over the learning rate $\eta \in \{0.1, 0.01, 0.001\}$ and for each perturbation budget, we choose the one that interpolates the training set and minimises robust error on the test set.

Robust evaluation We perform all the attacks to evaluate robust risk at test-time using exact line search; this is computationally tractable since the attacks are directed along one dimension.

Training paradigms For standard training (ST), wee train the network to minimise the cross-entropy loss. For adversarial training (AT) (Madry et al., 2018; Goodfellow et al., 2015), we train the network to minimise the robust binary cross-entropy loss. At each epoch, we compute an exact adversarial example using line search and update the weights using a gradient with respect to this example. For convex outer adversarial polytope (COAP) (Wong & Kolter, 2018; Wong et al., 2018), at each epoch, we compute upper and lower bounds u and ℓ as described in Theorem 4. We then train the network to minimise the upper bound on robust error from Theorem 3.

D.2. Synthetic experiments with ℓ_2-ball perturbations

Below we provide complete experimental details to reproduce Figures 3 and 4.

Data generation For the spheres dataset, we generate a random $x \in \mathbb{R}^d$ where $\|x\|_2$ is either R_1 or R_{-1} , with equal probability assigned to each norm. We associate with each x a label y such that $y = -1$ if $\|x\|_2 = R_{-1}$ and $y = 1$ if $\|x\|_2 = R_1$. We can sample uniformly from this distribution by sampling $z \sim \mathcal{N}(0, I_d)$ and then setting $x = \frac{z}{\|z\|_2} R_{-1}$ or $x = \frac{z}{\|z\|_2} R_1$. For the linearly separable distribution we set $d = 1000,\ n = 50,\ n_{\text{test}} = 10^5,\ \gamma = 6$. For the concentric spheres distribution we set $d = 100,\ , n = 50,\ n_{\text{test}} = 10^5,\ \gamma_{\text{min}} = 1$ and $\gamma_{\text{max}} = 12$.

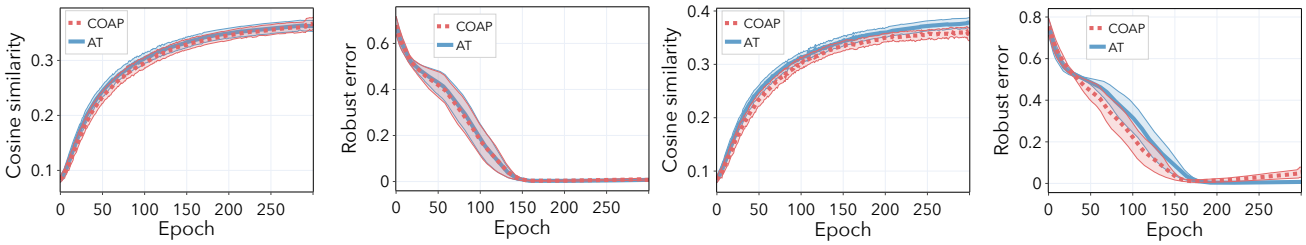
Model and hyper-parameters For all the experiments, we use a MLP architecture with $W = 100$ neurons in each hidden layer and $\mathrm{ReLU}(\cdot)$ activation functions. We use PyTorch SGD optimiser with a momentum of 0.95 and train the network for 150 epochs. We sweep over the learning rate $\eta \in \{0.1, 0.01, 0.001\}$ and for each perturbation budget, we choose the one that minimises robust error on the test set and interpolates the training set.

Robust evaluation We consider ℓ_2-ball perturbations. We evaluate robust error at test-time using Auto-PGD (Croce & Hein, 2020) with 100 iterations and 5 random restarts. We use both the cross-entropy and difference of logits loss to prevent gradient masking. We use the implementation provided in AutoAttack (Croce & Hein, 2020) with minor adjustments to allow for non-image inputs.

Training paradigms For standard training (ST), we train the network to minimise the cross-entropy loss. For adversarial training (AT) (Madry et al., 2018; Goodfellow et al., 2015) we train the network to minimise the robust cross-entropy loss. At each epoch, we search for adversarial examples using Auto-PGD (Croce & Hein, 2020) with a budget of 10 steps and 1 random restart. Then, we update the weights using a gradient with respect to the adversarial examples. For convex outer adversarial polytope (COAP) (Wong & Kolter, 2018; Wong et al., 2018). We train the network to minimise the upper-bound on the robust error. Our implementation is based on the code released by the authors.

D.3. Additional ℓ_2-ball perturbation synthetic experiments with small perturbation budget

For completeness we report here the same plots as in Figures 4b and 4c but for smaller perturbation budgets.



(a) $\epsilon = 0.2$ (b) $\epsilon = 0.2$ (c) $\epsilon = 0.4$ (d) $\epsilon = 0.4$

Figure 9. We report mean and standard error over 15 seeds. Observe that the robust generalisation gap increases with increasing perturbation budget ϵ.

D.4. Image experiments with ℓ_2-ball perturbations

Below we provide complete experimental details to reproduce Figures 2, 7, 8a and 8b.

Model architectures For MNIST, we train the CNN architecture with four convolutional layer and two fully connected layers of 512 units introduced in Wong et al. (2018). We report the architectural details in Table 1. For CIFAR-10, we train the residual network (ResNet) with the same structure used in Wong et al. (2018); we use 1 residual block with 16, 16, 32, and 64 filters. For Tiny ImageNet, we train a WideResNet. Following Xu et al. (2020) we use 3 wide basic blocks with a widen factor of 10.

CNN
CONV 32 $3 \times 3 + 1$
CONV 32 $4 \times 4 + 2$
CONV 64 $3 \times 3 + 1$
CONV 64 $4 \times 4 + 2$
FC 512
FC 512

Table 1. MNIST model architecture. All layers are followed by $\mathrm{ReLU}(\cdot)$ activations. The last fully connected layer is omitted. "CONV k $w \times h + s$" corresponds to a 2D convolutional layer with k filters of size $w \times h$ using a stride of s in both dimensions. "FC n" corresponds to a fully connected layer with n outputs.

Dataset preprocessing For MNIST, we use full 28×28 images without any augmentations and normalisation. For CIFAR-10, we use random horizontal flips and random crops as data augmentation, and normalise images according to per-channel statistics. For Tiny ImageNet, we use random crops of 56×56 and random flips during training. During testing, we use a central 56×56 crop. We also normalise images according to per-channel statistics.

Robust evaluation We consider ℓ_2-ball perturbations. We evaluate the robust error using the most expensive version of AutoAttack (AA+) (Croce & Hein, 2020). Specifically, we include the following attacks: untargeted APGD-CE (5 restarts), untargeted APGD-DLR (5 restarts), untargeted APGD-DLR (5 restarts), Square Attack (5000 queries), targeted APGD-DLR (9 target classes) and targeted FAB (9 target classes).

AT training details For MNIST, we train 100 epochs using Adam optimiser (Kingma & Ba, 2015) with a learning rate of 0.001, momentum of 0.9 and a batch size of 128; we reduce the learning rate by a factor 0.1 at epochs 40 and 80. For CIFAR-10 with ResNet, we train 150 epochs using SGD with a learning rate of 0.05 and a batch size of 128; we reduce the learning rate by a factor 0.1 at epochs 80 and 120. For Tiny Imagenet and CIFAR-10 with Wide-Resnet we train 200 epochs using SGD with a learning rate of 0.1 and a batch size of 512; we reduce the learning rate by a factor 0.1 at epochs 100 and 150. For the inner optimisation of all models and datasets, adversarial examples are generated with 10 iterations of Auto-PGD (Croce & Hein, 2020).

COAP training details We follow the settings proposed by the authors and report them here. For MNIST, we use the Adam optimiser (Kingma & Ba, 2015) with a learning rate of 0.001 and a batch size of 50. We schedule ϵ starting from 0.01 to the desired value over the first 20 epochs, after which we decay the learning rate by a factor of 0.5 every 10 epochs for a total of 60 epochs. For CIFAR-10, we use the SGD optimiser with a learning rate of 0.05 and a batch size of 50. We schedule ϵ starting from 0.001 to the desired value over the first 20 epochs, after which we decay the learning rate by a factor of 0.5 every 10 epochs for a total of 60 epochs. For all datasets and models, we use random projection of 50 dimensions. For all experiments, we use the implementation provided in Wong et al. (2018).

CROWN-IBP training details We follow the settings proposed by the authors and report them here. For MNIST, we train 200 epochs with a batch size of 256. We use Adam optimiser (Kingma & Ba, 2015) and set the learning rate to 5×10^{-4}. We warm up with 10 epochs of regular training, and gradually ramp up ϵ_{train} from 0 to ϵ in 50 epochs. We reduce the learning rate by a factor 0.1 at epoch 130 and 190. For CIFAR-10, we train 2000 epochs with a batch size of 256, and a learning rate of 5×10^{-4}. We warm up for 100 epochs, and ramp-up ϵ for 800 epochs. Learning rate is reduced by a factor 0.1 at epoch 1400 and 1700. For Tiny ImageNet, we train 600 epochs with batch size 128. The first 100 epochs are clean training, then we gradually increase ϵ_{train} with a schedule length of 400. For all datasets, an hyper-parameter β to balance LiRPA bounds and IBP bounds for the output layer is gradually decreased from 1 to 0 (1 for only using LiRPA bounds and 0 for only using IBP bounds), with the same schedule of ϵ. For all experiments, we use the implementation provided in the auto LiRPA library (Xu et al., 2020).

D.5. Image experiments with signal-directed perturbations

Below we provide complete experimental details to reproduce Figures 5a and 5b.

First, we explain our extensions of COAP to the threat model introduced in Equation (3). Rather than deriving the dual problem as in Appendix A.1, we consider the conjugate function view introduced in Wong et al. (2018). In particular, we only have to modify the dual of the input layer to the network. Below we derive the conjugate bound for the signal-directed threat model:

$$\begin{aligned}\sup_{\tilde{x}\in\mathcal{B}(x)} \nu_1^\top \tilde{x} &= \sup_{k,\beta} \nu_1^\top (x + s_k\beta) \\ &= \nu_1^\top x + \epsilon \max_k |\nu_1^\top s_k|\end{aligned}$$

For all experiments, we use the convolutional neural network architecture described in Table 1. Note that it is not possible to scale to ResNet with the threat model in Equation (3), as the random projections trick derived in Wong et al. (2018) is tailored to ℓ_∞ and ℓ_2 threat models.

AT training details For both MNIST and CIFAR-10, we train 20 epochs using Adam optimiser (Kingma & Ba, 2015) with a learning rate of 0.001, momentum of 0.9 and a batch size of 64; we reduce the learning rate by a factor 0.1 at epochs 10. For the inner optimisation of all models and datasets, we solve the exact problem as it is computationally efficient to line-search the maximal perturbation.

COAP training details For both MNIST and CIFAR-10, we use the Adam optimiser (Kingma & Ba, 2015) with a learning rate of 0.001 and a batch size of 64. We schedule ϵ starting from 0.01 to the desired value over the first 3 epochs, after which we decay the learning rate by a factor of 0.5 every 10 epochs. For all datasets and models, we *do not* use random projections. For all experiments, we use the implementation provided in Wong et al. (2018).

E. Additional experiments with ℓ_∞-ball perturbations

We consider ℓ_∞ adversaries and investigate the effect of perturbation budget on robust generalisation. We compare two probable defences, CROWN-IBP (Zhang et al., 2020) and COAP (Wong & Kolter, 2018), with the most effective empirical defence to date, AT (Madry et al., 2018). We plot the robust error on MNIST for increasing perturbation budget in Figure 10. As expected, the robustness of COAP quickly degrades as the perturbation budget increases. Especially for larger model size and perturbation budget, we observe that AT attains much better robust accuracy than COAP. However, for ℓ_∞ adversaries, interval bound propagation methods as CROWN-IBP achieve state of the art performance, outperforming empirical defences.

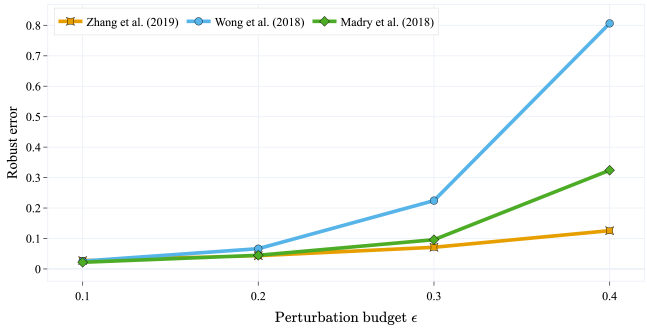


Figure 10. Results for ℓ_∞ adversaries against models trained using provable defenses and adversarial training on MNIST. In (a),(b) we plot the robust error for a CNN network as the perturbation budget ϵ increases.

In Table 2 we report the robust error on CIFAR10 for CNN and ResNet. For all perturbation budgets considered, AT achieves the best robust and standard error. Notably, the standard error of COAP and CROWN-IBP degrades significantly for larger perturbation budgets.

METHOD	MODEL	PERTURBATION BUDGET (ϵ)	STANDARD ERROR	ROBUST ERROR
AT (MADRY ET AL., 2018)	CNN	2/255	17.05%	41.52%
AT (MADRY ET AL., 2018)	RESNET	2/255	**16.58%**	**39.52%**
COAP (WONG & KOLTER, 2018)	CNN	2/255	32.30%	41.23%
COAP (WONG & KOLTER, 2018)	RESNET	2/255	33.67%	42.75%
CROWN-IBP (ZHANG ET AL., 2020)	CNN	2/255	41.67%	52.71%
AT (MADRY ET AL., 2018)	CNN	8/255	25.01%	74.20%
AT (MADRY ET AL., 2018)	RESNET	8/255	**22.59%**	**69.48%**
COAP (WONG & KOLTER, 2018)	CNN	8/255	80.99%	83.54%
COAP (WONG & KOLTER, 2018)	RESNET	8/255	72.93%	77.54%
CROWN-IBP (ZHANG ET AL., 2020)	CNN	8/255	59.91%	71.09%

Table 2. Results for ℓ_∞ adversaries against models trained using provable defenses and adversarial training on CIFAR10.