Certified defences hurt generalisation

Anonymous Author(s)
Affiliation
Address
email

Abstract

In recent years, much work has been devoted to designing certified defences for neural networks, i.e., methods for learning neural networks that are provably robust to certain adversarial perturbations. Due to the non-convexity of the problem, dominant approaches in this area rely on convex approximations, which are inherently loose. In this paper, we question the effectiveness of such approaches for realistic computer vision tasks. First, we provide extensive empirical evidence to show that certified defences suffer not only worse accuracy but also worse robustness and fairness than empirical defences. We hypothesise that the reason for why certified defences suffer in generalisation is (i) the large number of relaxed non-convex constraints and (ii) the strong alignment between the adversarial perturbations and the "signal" direction. We provide a combination of theoretical and experimental evidence to support these hypotheses.

1 Introduction

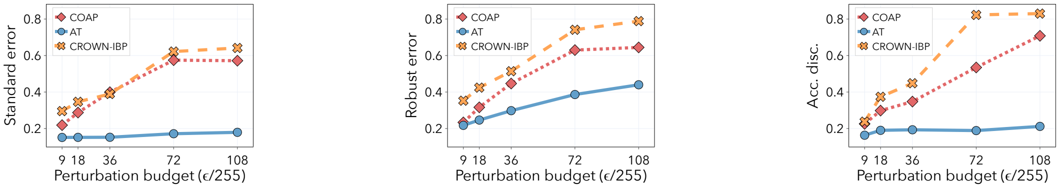
Several works have shown the existence of adversarial examples: imperceptible perturbations to the input can fool state-of-the-art classifiers [2, 22]. Consequently, robustness to adversarial examples has become a crucial design goal for machine learning models. In real-world scenarios, robustness against many different types of input perturbations may be desired depending on the domain of application. Therefore, to build robust models, we must first define a threat model for the adversary. 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.

Once a threat model is defined, we can formalise the problem of building models that are robust to adversarial examples. For any distribution $\mathcal{D}$, neural network model $f_\theta : \mathbb{R}^d \to \mathbb{R}^k$ parameterised by the weights $\theta \in \mathbb{R}^p$, and loss function L, our goal is to solve the following robust optimisation problem:

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

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 [12, 24]. Instead, two efficient techniques are widely used to overcome the computational barrier: *empirical* defences that provide a lower bound on the solution and *certified* defences that provide an upper bound.

Among empirical defences, Adversarial Training (AT) [11, 16] is one of the few that has stood the test of time. AT minimises the worst-case empirical loss in Equation (1) by approximately solving the inner-maximisation problem with first-order gradient-based optimisation methods. However, despite its simplicity and computational efficiency, owing to its heuristic nature, AT is incapable of



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

Figure 1: Results for ℓ_2-adversaries on the CIFAR-10 dataset. We compare ResNet architectures trained using state-of-the-art certified defenses CROWN-IBP [31, 28] and COAP [26, 25] against the most popular empirical defense to date AT [16, 11]. In Figures 1a, 1b and 2g we plot respectively standard error, robust error and accuracy discrepancy as the perturbation budget increases. See Appendix D.3 for complete experimental details.

guaranteeing that no adversarial examples exist. In many safety critical domains, such guarantees are of immense importance.

To address this limitation, recently, there has been significant interests in designing certified defences, i.e., methods for learning neural networks that are *provably* robust to norm-bounded perturbations on the training data. Many recent works [25, 18, 7, 31] have proposed to solve a convex relaxation of the inner-maximisation problem by relaxing the non-convex ReLU constraint sets with convex ones. Despite all of these progresses, 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 due to the looseness of the convex relaxation [20]. In this paper, we argue that the fundamental looseness of convex relaxations hinders the practical effectiveness of current certified defences. In particular, as shown in Figure 1, certified defences suffer significantly worse accuracy, robustness, and fairness on the test data compared to adversarial training. Our contributions are as follows:

- In Section 2, we show that current certified defences hurt accuracy, robustness, and fairness across a range of ℓ_2-ball perturbations on real-world vision datasets like MNIST and CIFAR-10.
- In Section 3, we provide experimental evidence that certified defences hurt generalisation because of (i) the large number of relaxed non-convex constraints and (ii) strong alignment between the adversarial perturbations and the signal direction.

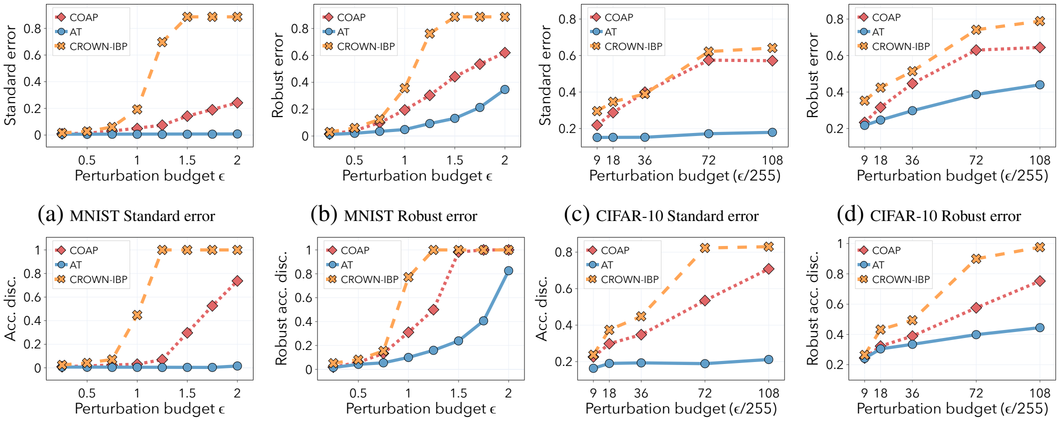
2 Certified defences hurt generalisation on real-world data

In this section, we show that certified defences hurt standard error, robust error, and fairness on two common computer vision datasets: MNIST [15] and CIFAR-10 [14]. Among certified defences, we consider the convex outer adversarial polytope (COAP) [26, 25], which achieves state-of-the-art certified robustness under ℓ_2-ball perturbations. Additionally, we consider CROWN-IBP [30, 28], which uses the tight convex relaxation CROWN [30] and achieves state-of-the-art certified robustness under ℓ_∞-ball perturbations. Among empirical defences, we consider adversarial training (AT) [16, 11], which is one of the most popular and effective defences to date.

Models and robust evaluation We consider the ℓ_2-ball perturbations threat model. To reliably evaluate the robust error, we use the strongest version of AutoAttack (AA+) [5]. For CIFAR-10, we train a residual network (ResNet) and for MNIST we train a large convolutional neural network (CNN). Both architectures were introduced in Wong et al. [26] as standard benchmarks for certified defences.

Certified defences hurt standard and robust error Several studies have shown that adversarial training may lead to an increase in standard error when compared with standard training [19, 23, 29]. We observe the same phenomenon to a much higher degree in certified defences. Our experimental results show that certified defences not only suffer worse standard error but also worse robust error than adversarial training. First, we observe on both MNIST and CIFAR-10 in Figures 2a and 2c, respectively, that for increasing perturbation budget, the standard error gap between certified (CROWN-IBP, COAP) and empirical defences (AT) increases. Secondly, we observe that the robust error gap increases with increasing perturbation budget for both MNIST and CIFAR-10 in Figures 2b and 2d, respectively.

Certified defences hurt fairness Previously, we showed that certified defences hurt both standard and robust generalisation. Taking it one step further, we show that certified defences (CROWN-IBP, COAP) suffer significantly worse fairness than empirical defences (AT). Let $\mathbf{R}(\theta)$ be the standard



(a) MNIST Standard error (b) MNIST Robust error (c) CIFAR-10 Standard error (d) CIFAR-10 Robust error

(e) MNIST Unfairness (f) MNIST Robust unfairness (g) CIFAR-10 Unfairness (h) CIFAR-10 Robust unfairness

Figure 2: Results for ℓ_2-adversaries on MNIST and CIFAR-10 datasets. In Figures 2a, 2b, 2e and 2f we plot respectively the standard error, robust error, accuracy discrepancy and robust accuracy discrepancy for a CNN trained on MNIST, as the perturbation budget ϵ increases. In Figures 2c, 2d, 2g and 2h we plot respectively the standard error, robust error, accuracy discrepancy and robust accuracy discrepancy, for a ResNet trained on CIFAR-10, as the perturbation budget ϵ increases.

error of the classifier f_θ and $\mathbf{R}^k(\theta)$ the standard error conditioned on the class label k. We measure the degree of unfairness as follows: $(\max_k \mathbf{R}^k(\theta) - \mathbf{R}(\theta))(1 - \mathbf{R}(\theta))^{-1}$. Using the terminology in Sanyal et al. [21], we refer to this metric as *accuracy discrepancy*. Similarly, we also consider the discrepancy in robust accuracy, as it was observed in Xu et al. [27] that adversarial defences may induce a large discrepancy of robustness among different classes. We refer to this metric as *robust accuracy discrepancy* where we replace the standard error with the robust error.

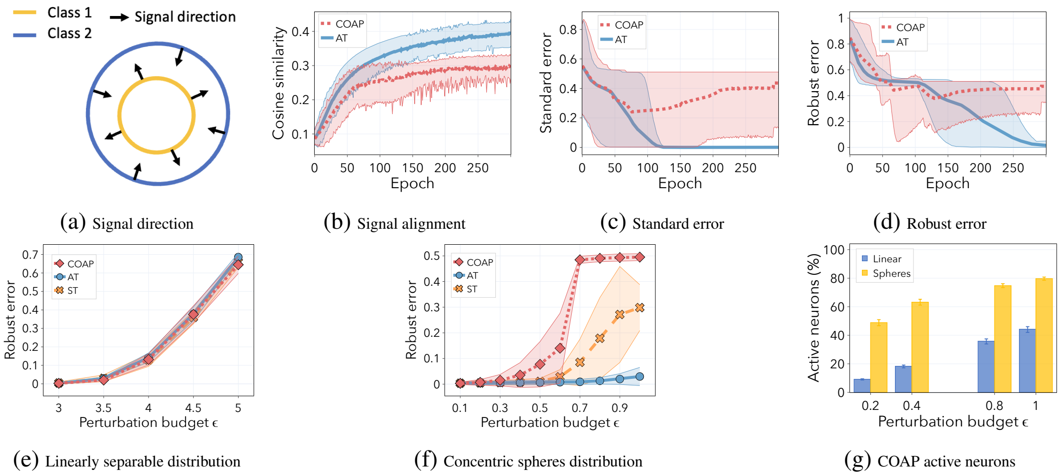
We present our experimental results comparing the fairness of certified and empirical defences. For MNIST, we observe in Figure 2e and 2f that COAP and CROWN-IBP have a significant discrepancy for both standard and robust accuracy. For large perturbations, these methods obtain 100% discrepancy, indicating that their accuracy on the worst class can be as low as 0%. By contrast, AT preserves fairness for both standard and robust accuracy much better. In particular, the discrepancy for standard accuracy is always less than 2% for all perturbation budgets considered. Similarly, for CIFAR-10 AT maintains a constant accuracy discrepancy around 20% for all perturbation budgets considered, whereas for certified defences it steadily increases with the perturbation budget, reaching above 80%. Additionally, for robust accuracy, we observe a discrepancy gap of 35% between the best certified and empirical defences for the largest perturbation budget considered.

3 Developing intuition on synthetic datasets

In this section, we hypothesise that certified defences hurt robust and standard generalisation because of (i) the large number of relaxed non-convex constraints and (ii) strong alignment between the adversarial perturbations and the signal direction. We investigate these hypotheses on more controlled settings. Specifically, we consider two synthetic data distributions: a linearly separable distribution as in Clarysse et al. [4], which is similar to the one in in Nagarajan and Kolter [17], Tsipras et al. [23], and the concentric spheres distribution studied in Gilmer et al. [10], Nagarajan and Kolter [17].

Data and threat models Similarly to the previous section, we focus on ℓ_2-ball perturbations of size ϵ. As for distributions, we consider the linearly separable distribution where first, the label $y \in \{+1, -1\}$ is drawn with equal probability. Then, for some $\gamma > 0$, the covariate vector is created as $x = [\gamma \operatorname{sgn}(y); \tilde{x}]$, where $\tilde{x} \in \mathbb{R}^{d-1}$ is a random vector drawn from a standard normal distribution $\tilde{x} \sim \mathcal{N}\left(0, \sigma^2 I_{d-1}\right)$ and $[;]$ represents concatenation. We sample the concentric spheres dataset as follows; for $0 < R_1 < R_{-1}$, we 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.

In Figure 3e and 3f, we plot the robust error of standard training (ST), adversarial training (AT), and certified training (COAP) on the linear and concentric spheres distributions respectively. We



(a) Signal direction

(b) Signal alignment

(c) Standard error

(d) Robust error

(e) Linearly separable distribution

(f) Concentric spheres distribution

(g) COAP active neurons

Figure 3: We report mean and standard deviation over 15 seeds. In Figures 3e and 3f 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. In Figure 3g, we plot the percentage of neurons in the activation set for the linearly separable and concentric spheres distribution respectively. In Figure 3a we plot a 2-D visualisation of the concentric spheres dataset, the black arrow illustrates the signal direction. In Figure 3b we plot the cosine similarity between ℓ_2-ball perturbations on the training data (average) and the signal directed vector. In Figure 3c and 3d we plot standard and robust error for adversarial training (AT) and convex outer adversarial polytope (COAP).

see that in contrast to the linear setting, COAP has a much higher robust error on the concentric spheres distribution than AT and ST, where the gap increases for increasing perturbation budget ϵ. The intuition for why this happens is two-fold: first of all COAP relaxes the non-convex ReLU constraints for all neurons that activate within the perturbation set, i.e., there exists $\delta \in \mathcal{B}_\epsilon$ for which the input to the neuron equals 0. Hence, the larger the percentage of relaxed neurons, the worse the approximation. This is formally captured by Theorem A.1 in Appendix A. Secondly, the ℓ_2-ball perturbations are significantly aligned with the signal direction, meaning that they effectively reduce the information about the label in the data. Applying an approximation in this direction yields poor generalisation. We prove this in Theorem B.1 in Appendix B for the linearly separable distribution.

COAP relaxes many constraints on the concentric spheres In Figure 3g we empirically show that COAP convexly approximates a large number of constraints when training on the concentric spheres distribution. We plot the percentage of active neurons on the concentric spheres and linear distributions against increasing perturbation budgets: the percentage is much higher for the concentric spheres than for the linearly separable distribution and increases with perturbation budget ϵ. Indeed, the complex spherical decision boundary requires much more active neurons compared to the linear decision boundary which only needs 1 active neuron.

ℓ_2-ball perturbations align with the signal direction We empirically show that ℓ_2-ball perturbations align with the signal direction on the concentric spheres distribution. Note that for a point x drawn from the concentric spheres distribution, the signal direction is given by $y\frac{x}{\|x\|_2}$ (see Figure 3a for a 2D visualization). In Figure 3b, we plot the cosine distance between the ℓ_2-perturbations computed on the training set, and the signal direction. Comparing Figures 3b to 3d, we see that during the early stages of training, the ℓ_2-ball perturbations are not aligned with the signal direction and the robust and standard errors for COAP are similar to AT. However, after some epochs, when the ℓ_2-ball perturbations start to align with the signal direction, both the robust and standard error gaps between COAP and AT increase. This provides evidence that, as training progresses, ℓ_2-ball perturbations become significantly aligned with the signal direction and the generalisation gap worsens.

4 Conclusions

In this paper, we show that certified defences can hurt robustness, accuracy and fairness for realistic datasets and adversarial perturbations. Further, we develop intuition on synthetic datasets for why certified defences hurt generalisation, combining both theoretical and experimental evidence. We believe that understanding the performance gap between empirical and certified defences will lead to better approaches for adversarial robustness.

References

[1] Jimmy Ba, Murat A. Erdogdu, Taiji Suzuki, Zhichao Wang, Denny Wu, and Greg Yang. High-dimensional Asymptotics of Feature Learning: How One Gradient Step Improves the Representation, May 2022. arXiv:2205.01445 [cs, math, stat].

[2] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion Attacks against Machine Learning at Test Time. In *Machine Learning and Knowledge Discovery in Databases - European Conference*, 2013.

[3] Niladri S. Chatterji, Philip M. Long, and Peter L. Bartlett. When Does Gradient Descent with Logistic Loss Find Interpolating Two-Layer Networks? *Journal of Machine Learning Research*, (159), 2021. ISSN 1533-7928.

[4] Jacob Clarysse, Julia Hörrmann, and Fanny Yang. Why adversarial training can hurt robust accuracy, 2022. arXiv:2203.02006.

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

[6] Amit Daniely and Eran Malach. Learning Parities with Neural Networks. In *Advances in Neural Information Processing Systems*, 2020.

[7] Krishnamurthy Dvijotham, Robert Stanforth, Sven Gowal, Timothy A. Mann, and Pushmeet Kohli. A Dual Approach to Scalable Verification of Deep Networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2018.

[8] Ecenaz Erdemir, Jeffrey Bickford, Luca Melis, and Sergül Aydöre. Adversarial Robustness with Non-uniform Perturbations. In *Advances in Neural Information Processing Systems*, 2021.

[9] Spencer Frei, Niladri S. Chatterji, and Peter L. Bartlett. Random Feature Amplification: Feature Learning and Generalization in Neural Networks, May 2022. arXiv:2202.07626 [cs, math, stat].

[10] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S. Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian J. Goodfellow. Adversarial Spheres. *CoRR*, 2018. arXiv: 1801.02774.

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. In *Proceedings of the International Conference on Learning Representations*, 2015.

[12] Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In *Proceedings of the International Conference of Computer Aided Verification*, 2017.

[13] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations*, 2015.

[14] Alex Krizhevsky. Learning multiple layers of features from tiny images. *citeseer*, 2009.

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, (11), 1998.

[16] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks. In *Proceedings of the International Conference on Learning Representations*, 2018.

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

[18] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified Defenses against Adversarial Examples. In *Proceedings of the International Conference on Learning Representations*, 2018.

[19] Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C. Duchi, and Percy Liang. Understanding and Mitigating the Tradeoff between Robustness and Accuracy. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 7909–7919. PMLR, 2020. URL `http://proceedings.mlr.press/v119/raghunathan20a.html`.

[20] Hadi Salman, Greg Yang, Huan Zhang, Cho-Jui Hsieh, and Pengchuan Zhang. A Convex Relaxation Barrier to Tight Robustness Verification of Neural Networks. In *Advances in Neural Information Processing Systems*, 2019.

[21] Amartya Sanyal, Yaxi Hu, and Fanny Yang. How unfair is private learning? In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2022.

[22] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proceedings of the International Conference on Learning Representations*, 2014.

[23] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness May Be at Odds with Accuracy. In *Proceedings of the International Conference on Learning Representations*, 2019.

[24] Tsui-Wei Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane S. Boning, and Inderjit S. Dhillon. Towards fast computation of certified robustness for relu networks. In *Proceedings of the International Conference on Machine Learning*, 2018.

[25] Eric Wong and J. Zico Kolter. Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope. In *Proceedings of the International Conference on Machine Learning*, 2018.

[26] Eric Wong, Frank R. Schmidt, Jan Hendrik Metzen, and J. Zico Kolter. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems*, 2018.

[27] Han Xu, Xiaorui Liu, Yaxin Li, Anil K. Jain, and Jiliang Tang. To be Robust or to be Fair: Towards Fairness in Adversarial Training. In *Proceedings of the International Conference on Machine Learning*, Proceedings of Machine Learning Research, 2021.

[28] Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic Perturbation Analysis for Scalable Certified Robustness and Beyond. In *Advances in Neural Information Processing Systems*, 2020.

[29] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically Principled Trade-off between Robustness and Accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR, 2019. URL `http://proceedings.mlr.press/v97/zhang19p.html`.

[30] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient Neural Network Robustness Certification with General Activation Functions. In *Advances in Neural Information Processing Systems*, 2018.

[31] Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane S. Boning, and Cho-Jui Hsieh. Towards Stable and Efficient Training of Verifiably Robust Neural Networks. In *Proceedings of the International Conference on Learning Representations*, 2020.

A COAP for signal-directed adversaries

In this section, we extend COAP to signal-directed adversaries, our derivation can be seen as an extension of Wong and Kolter [25], Erdemir et al. [8]. We consider the hypothesis class to be the set of one-hidden layer neural networks $f_\theta : \mathbb{R}^d \to \mathbb{R}^2$ with parameters $\theta = \{W_1, b_1, W_2, b_2\}$:

$$x \xrightarrow{x+\delta} z_1 \xrightarrow{W_1 z_1 + b_1} \hat{z}_2 \xrightarrow{\text{ReLU}(\cdot)} z_2 \xrightarrow{W_2 z_2 + b_2} \hat{z}_3 \tag{2}$$

where $x \in \mathbb{R}^d$ and $z_1 \in \mathcal{B}_\epsilon(x)$. We define the adversarial polytope $\mathcal{Z}_\epsilon(x)$ as the set of all final-layer activations attainable by perturbing x with some $\tilde{x} \in \mathcal{B}_\epsilon(x)$:

$$\mathcal{Z}_\epsilon(x) = \{f_\theta(\tilde{x}) : \tilde{x} \in \mathcal{B}_\epsilon(x)\} \tag{3}$$

Our approach will be to construct a convex outer bound on this adversarial polytope: if no adversarial example exists in this outer approximation, then we are guaranteed that no adversarial example exists in the original polytope. We relax the ReLU activations $z = \text{ReLU}\,(\hat{z})$ with their convex envelopes:

$$z \geq 0,\ z \geq \hat{z},\ (u - \ell) z \leq u\hat{z} - u\ell \tag{4}$$

where u and ℓ are respectively the pre-activations $\hat{z}$ upper and lower bounds, for which we provide a closed form solution in Appendix A.1. We define the outer bound on the adversarial polytope we get from relaxing ReLU constraints as $\tilde{\mathcal{Z}}_\epsilon(x)$. Given a sample x with known label y, we can write the linear program formulation of the adversary's problem for our network as follows:

$$\min_{\hat{z}_3}\ [\hat{z}_3]_y - [\hat{z}_3]_{\bar{y}} = c^\top \hat{z}_3 \qquad \text{s.t. }\ \hat{z}_3 \in \tilde{\mathcal{Z}}_\epsilon \tag{5}$$

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 can misclassify the example. Since solving the linear program (5) for every example in the dataset is intractable, we consider the dual formulation and take a feasible solution. In Theorem A.1, we state the dual problem formulation of the linear program with ReLU relaxations (5).

Theorem A.1. *The dual of the linear program* (5) *can be written as*

$$\begin{array}{ll} \max\limits_{\alpha} & \widetilde{J}_\epsilon\left(x, g_\theta(c, \alpha)\right) \\ \textit{s.t.} & \alpha_j \in [0, 1],\ \forall j \end{array} \tag{6}$$

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

$$-\sum_{i=1}^{2} \nu_{i+1}^\top b_i + \sum_{j \in \mathcal{I}} \ell_j [\nu_2]_j^+ - \hat{\nu}_1^\top x - \epsilon \left\| [\hat{\nu}_1]_1 \right\|_1 \tag{7}$$

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

$$\begin{array}{l} \nu_3 = -c \\ \hat{\nu_2} = W_2^\top \nu_3 \\ [\nu_2]_j = 0,\ j \in \mathcal{I}^- \\ [\nu_2]_j = [\hat{\nu}_2]_j\,,\ j \in \mathcal{I}^+ \\ [\nu_2]_j = \frac{u_j}{u_j - \ell_j} [\hat{\nu}_2]_j^+ - \alpha_j [\hat{\nu}_2]_j^-\,,\ j \in \mathcal{I} \\ \hat{\nu_1} = W_1^\top \nu_2 \end{array} \tag{8}$$

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.*

In particular, this theorem states that we can represent the dual problem as a linear back propagation network, which provides a tractable solution for a lower bound on the primal objective. In practice, rather than solving the exact dual problem, we choose the fixed, dual feasible solution: $\alpha_j = \frac{u_j}{u_j - \ell_j}$.

A.1 Computing upper and lower bounds

We address here the problem of obtaining the upper and lower bounds u and ℓ for the pre-activations $\hat{z}$, which so far we have assumed to be known. In Proposition A.2 we give a closed form solution for ℓ and u.

Proposition A.2. *Consider the neural network defined in Equation* (2)*. Let w_1 be the first column of W_1 and x be a given example, then we have the following element-wise bound:*

$$\ell \leq \hat{z}_2 \leq u \tag{9}$$

where

$$\ell = W_1 x + b_1 - \epsilon |w_1|, \; u = W_1 x + b_1 + \epsilon |w_1| \tag{10}$$

Proof. Given an example x, let $\tilde{x} = x + \delta$ be the perturbed input to the network. We want to upper bound the pre-activations values $\hat{z_2}$:

$$\hat{z_2} = W_1(x + \delta) + b_1 = W_1 x + b_1 + W_1 \delta \tag{11}$$

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

$$u_i = \max_{\tilde{x} \in \mathcal{B}_\epsilon(x)} [\hat{z}_2]_i = [W_1 x]_i + [b_1]_i + \max_{\tilde{x} \in \mathcal{B}_\epsilon(x)} [W_1 \delta]_i \tag{12}$$

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:

$$\max_{\tilde{x} \in \mathcal{B}_\epsilon(x)} [W_1 \delta]_i = \max_{\|\beta\|_1 \leq \epsilon} \beta \cdot [w_1]_i = \epsilon \cdot \|[w_1]_i\|_1 \tag{13}$$

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

$$u = W_1 x + b_1 + \epsilon |w_1| \tag{14}$$

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

$$l = W_1 x + b_1 - \epsilon |w_1| \tag{15}$$

□

A.2 Proof of Theorem A.1

Theorem A.1. *The dual of the linear program* (5) *can be written as*

$$\begin{array}{ll} \max\limits_{\alpha} & \widetilde{J}_\epsilon\left(x, g_\theta(c, \alpha)\right) \\ \textit{s.t.} & \alpha_j \in [0, 1], \; \forall j \end{array} \tag{6}$$

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

$$-\sum_{i=1}^{2} \nu_{i+1}^\top b_i + \sum_{j \in \mathcal{I}} \ell_j [\nu_2]_j^+ - \hat{\nu}_1^\top x - \epsilon \left\|[\hat{\nu}_1]_1\right\|_1 \tag{7}$$

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

$$\begin{array}{l} \nu_3 = -c \\ \hat{\nu_2} = W_2^\top \nu_3 \\ [\nu_2]_j = 0, \; j \in \mathcal{I}^- \\ [\nu_2]_j = [\hat{\nu}_2]_j \,, \; j \in \mathcal{I}^+ \\ [\nu_2]_j = \frac{u_j}{u_j - \ell_j} [\hat{\nu}_2]_j^+ - \alpha_j [\hat{\nu}_2]_j^- \,, \; j \in \mathcal{I} \\ \hat{\nu_1} = W_1^\top \nu_2 \end{array} \tag{8}$$

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.

Proof. Given an example x, let $\tilde{x} = x + \delta$ be the perturbed input to the network. First, we explicit all the constraints for the linear program defined in (5):

$$\begin{aligned}
&\min_{\hat{z}_3} \, [\hat{z}_3]_y - [\hat{z}_3]_{\bar{y}} = c^\top \hat{z}_3 \ , \qquad \text{s.t.} \\
&\tilde{x} \in \mathcal{B}_\epsilon(x) \\
&z_1 = \tilde{x} \\
&\hat{z}_2 = W_1 z_1 + b_1 \\
&\hat{z}_3 = W_2 z_2 + b_2 \\
&[z_2]_j = 0, \ \forall j \in \mathcal{I}^- \\
&[z_2]_j = [\hat{z}_2]_j, \ \forall j \in \mathcal{I}^+ \\
&[z_2]_j \geq 0, \ \forall j \in \mathcal{I} \\
&[z_2]_j \geq [\hat{z}_2]_j, \ \forall j \in \mathcal{I} \\
&((u_j - \ell_j) \, [z_2]_j \ - u_j [\hat{z}_2]_j) \leq -u_j \ell_j, \ \forall j \in \mathcal{I}
\end{aligned} \tag{16}$$

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:

$$\begin{aligned}
\hat{z}_2 = W_1 z_1 + b_1 &\Rightarrow \nu_2 \\
\hat{z}_3 = W_2 z_2 + b_2 &\Rightarrow \nu_3 \\
z_1 = x + \delta &\Rightarrow \psi \\
-[z_2]_j \leq 0 &\Rightarrow \mu_j, \ \forall j \in \mathcal{I} \\
[\hat{z}_2]_j - [z_2]_j \leq 0 &\Rightarrow \tau_j, \ \forall j \in \mathcal{I} \\
((u_j - \ell_j) \, [z_2]_j - u_j [\hat{z}_2]_j) \leq -u_j \ell_j &\Rightarrow \lambda_j, \ \forall j \in \mathcal{I}
\end{aligned} \tag{17}$$

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:

$$\begin{aligned}
\mathcal{L}(z, \hat{z}, \nu, \delta, \lambda, \tau, \mu, \psi) = &- \left(W_1^\top \nu_2 + \psi \right)^\top z_1 - \sum_{j \in \mathcal{I}} \Big(\mu_j + \tau_j - \lambda_j \left(u_j - \ell_j \right) + \left[W_2^\top \nu_3 \right]_j \Big) \, [z_2]_j \\
&+ \sum_{j \in \mathcal{I}} \left(\tau_j - \lambda_j u_j + [\nu_2]_j \right) [\hat{z}_2]_j + \left(c + \nu_3 \right)^\top \hat{z}_3 - \sum_{i=1}^{2} \nu_{i+1}^\top b_i \\
&+ \sum_{j \in \mathcal{I}} \lambda_j u_j \ell_j + \psi^\top x + \psi^\top \delta + \sum_{j \in \mathcal{I}^-} [\hat{z}_2]_j [\nu_2]_j \\
&+ \sum_{j \in \mathcal{I}^+} [z_2]_j \left([\nu_2]_j - [W_2^\top \nu_3]_j \right) \\
\text{s.t. } \tilde{x} \in \mathcal{B}_\epsilon(x) &
\end{aligned} \tag{18}$$

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

$$\begin{aligned}
\inf_{z, \hat{z}, \delta} \mathcal{L}(z, \hat{z}, \nu, \delta, \lambda, \tau, \mu, \psi) = &- \inf_{z_2} \sum_{j \in \mathcal{I}} \Big(\mu_j + \tau_j - \lambda_j \left(u_j - \ell_j \right) + \left[W_2^\top \nu_3 \right]_j \Big) \, [z_2]_j \\
&+ \inf_{\hat{z}_2} \sum_{j \in \mathcal{I}} \left(\tau_j - \lambda_j u_j + [\nu_2]_j \right) [\hat{z}_2]_j + \inf_{\hat{z}_3} \left(c + \nu_3 \right)^\top z_3 - \sum_{i=1}^{2} \nu_{i+1}^\top b_i \\
&+ \sum_{j \in \mathcal{I}} \lambda_j u_j \ell_j + \psi^\top x + \inf_{\tilde{x} \in \mathcal{B}_\epsilon(x)} \psi^\top \delta - \inf_{z_1} \left(W_1^\top \nu_2 + \psi \right)^\top z_1 \\
&+ \inf_{\hat{z}_2} \sum_{j \in \mathcal{I}^-} [\hat{z}_2]_j [\nu_2]_j + \inf_{z_2} \sum_{j \in \mathcal{I}^+} [z_2]_j \left([\nu_2]_j - [W_2^\top \nu_3]_j \right)
\end{aligned} \tag{19}$$

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

$$\inf_{\tilde{x} \in \mathcal{B}_\epsilon(x)} \psi^\top \delta = \inf_{\|\beta\|_1 \leq \epsilon} \psi_1 \cdot \beta = -\epsilon \cdot \|\psi_1\|_1 \tag{20}$$

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:

$$\inf_{z,\hat{z},\delta} \mathcal{L}(.) = \begin{cases} -\sum_{i=1}^{2} \nu_{i+1}^\top b_i + \sum_{j\in\mathcal{I}} \lambda_j u_j \ell_j + \psi^\top x - \epsilon \left\|\psi_1\right\|_1 & \text{if conditions} \\ -\infty & \text{else} \end{cases} \tag{21}$$

where the conditions to satisfy are:

$$\begin{aligned} & \nu_3 = -c \\ & W_1^\top \nu_2 = -\psi \\ & [\nu_2]_j = 0, j \in \mathcal{I}_i^- \\ & [\nu_2]_j = \left[W_2^\top \nu_3\right]_j, j \in \mathcal{I}_i^+ \\ & \left.\begin{aligned} (u_j - \ell_j)\,\lambda_j - \mu_j - \tau_j \quad &= \left[W_2^\top \nu_3\right]_j \\ [\nu_2]_j \quad &= u_j \lambda_j - \tau_j \end{aligned}\right\} j \in \mathcal{I} \\ & \lambda, \tau, \mu \geq 0 \end{aligned} \tag{22}$$

Thus, we can rewrite the dual problem as follows:

$$\begin{aligned} & \max_{\nu,\psi,\lambda,\tau,\mu} -\sum_{i=1}^{2} \nu_{i+1}^\top b_i + \sum_{j\in\mathcal{I}} \lambda_j u_j \ell_j + \psi^\top x - \epsilon \left\|\psi_1\right\|_1 \\ & \text{s.t.} \quad \nu_3 = -c \\ & W_1^\top \nu_2 = -\psi \\ & [\nu_2]_j = 0, j \in \mathcal{I}_i^- \\ & [\nu_2]_j = \left[W_2^\top \nu_3\right]_j, j \in \mathcal{I}_i^+ \\ & \left.\begin{aligned} (u_j - \ell_j)\,\lambda_j - \mu_j - \tau_j \quad &= \left[W_2^\top \nu_3\right]_j \\ [\nu_2]_j \quad &= u_j \lambda_j - \tau_j \end{aligned}\right\} j \in \mathcal{I} \\ & \lambda, \tau, \mu \geq 0 \end{aligned} \tag{23}$$

Note that the dual variable λ corresponds to the upper bounds in the convex 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 $\tau_j + \mu_j \geq 0$:

$$\begin{aligned} (u_j - \ell_j)\lambda_j &= [W_2^\top \nu_3]_j^+ \\ \tau_j + \mu_j &= [W_2^\top \nu_3]_j^- \end{aligned} \tag{24}$$

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

$$[\nu_2]_j = \frac{u_j}{u_j - \ell_j}[W_2^\top \nu_3]_j^+ - \alpha_j [W_2^\top \nu_3]_j^- \tag{25}$$

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

$$\lambda_j = \frac{u_j}{u_j - \ell_j}[\hat{\nu}_2]_j^+ \tag{26}$$

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

$$\begin{aligned} -\sum_{i=1}^{2} \nu_{i+1}^\top b_i + \sum_{j\in\mathcal{I}} \lambda_j u_j \ell_j + \psi^\top x - \epsilon \left\|\psi_1\right\|_1 &= -\sum_{i=1}^{2} \nu_{i+1}^\top b_i + \sum_{j\in\mathcal{I}} \frac{u_j \ell_j}{u_j - \ell_j}[\hat{\nu}_2]_j^+ - \hat{\nu}_1^\top x - \epsilon \left\|[\hat{\nu}_1]_1\right\|_1 \\ &= -\sum_{i=1}^{2} \nu_{i+1}^\top b_i + \sum_{j\in\mathcal{I}} \ell_j [\nu_2]_j^+ - \hat{\nu}_1^\top x - \epsilon \left\|[\hat{\nu}_1]_1\right\|_1 \end{aligned} \tag{27}$$

and the final dual problem:

$$\begin{array}{ll} \max\limits_{\nu,\hat{\nu}} & -\sum_{i=1}^{2} \nu_{i+1}^{\top} b_i + \sum_{j\in\mathcal{I}} \ell_j [\nu_2]_j^{+} - \hat{\nu}_1^{\top} x - \epsilon \left\| [\hat{\nu}_1]_1 \right\|_1 \\ \text{s.t.} & \nu_3 = -c \\ & \hat{\nu_2} = W_2^{\top} \nu_3 \\ & [\nu_2]_j = 0,\ j \in \mathcal{I}^- \\ & [\nu_2]_j = \left[\hat{\nu}_2\right]_j,\ j \in \mathcal{I}^+ \\ & [\nu_2]_j = \frac{u_j}{u_j-\ell_j} [\hat{\nu}_2]_j^{+} - \alpha_j [\hat{\nu}_2]_j^{-},\ j \in \mathcal{I} \\ & \hat{\nu_1} = W_1^{\top} \nu_2 \end{array} \tag{28}$$

□

B Approximations along the signal direction hurt generalisation

In this section, we further investigate our hypothesis that certified defences hurt generalisation when adversarial perturbations are aligned with the signal direction. In particular, we study the linearly separable distribution from the previous section and assume that the adversarial attacks concentrate all of their perturbation budget along the signal direction. In Theorem B.1, we prove for a simple neural network that, in high dimensions, certified defences (COAP) yield higher robust error than empirical defences (AT) for large perturbation budgets. We then corroborate our theoretical results with extensive experimental evidence on synthetic data.

Data and threat models We consider the linearly separable distribution described in Section 3. As for the threat model, we consider signal-directed attacks that efficiently concentrate their attack budget on the signal in the input. Since the signal direction corresponds to the first component of the data, we define the set of allowed perturbations as:

$$\mathcal{B}_\epsilon(x) = \{z_1 = x + e_1\beta \mid |\beta| \le \epsilon\} \tag{29}$$

where e_1 is the standard basis vector of the first coordinate. Further, as the original formulation of COAP only allows ℓ_p-adversaries, we provide an extension of COAP that covers our theoretical and experimental setting in Appendix A.

One gradient step training We consider the hypothesis class to be the set of one-neuron shallow neural networks $f_\theta : \mathbb{R}^d \to \mathbb{R}$, defined by:

$$f_\theta(x) = a\,\mathrm{ReLU}\left(\theta^\top x\right) + b \tag{30}$$

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

We study the early phase of neural network optimisation. Under structural assumptions on the data, it has been proved that one gradient step with sufficiently large learning rate can drastically decrease the training loss [3] and extract task-relevant features [9, 6]. A similar setting was also studied recently in Ba et al. [1] for the MSE loss in the high-dimensional asymptotic limit. Here, we focus on the classification setting with binary cross-entropy loss. Below we state our main theorem.

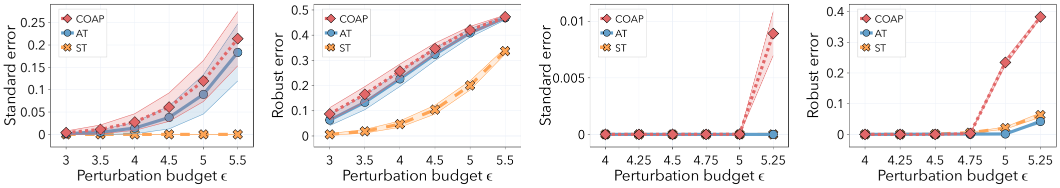
Theorem B.1. *Let $\bar{\theta}$ and $\tilde{\theta}$ be the network parameters after one step of gradient descent with respect to AT and COAP objectives. Let,*

$$\frac{\|\theta_{2:d}\|_2}{\|\theta_1\|_2} > \sqrt{\frac{24\gamma^3}{\sigma^2}} \quad \textit{and} \quad \frac{2}{3}\gamma < \epsilon < \gamma \tag{31}$$

where θ are the network parameters at initialization. Then, COAP yields higher robust risk than AT:

$$\mathbf{R}_\epsilon(\tilde{\theta}) > \mathbf{R}_\epsilon(\bar{\theta}) \tag{32}$$

Theorem B.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 the network parameters are initialised sampling from a standard multivariate gaussian $\theta \sim \mathcal{N}(0, I_d)$, then we know that $\|\theta\|_2$



(a) Small sample size regime (b) Small sample size regime (c) Large sample size regime (d) Large sample size regime

Figure 4: We report mean and standard deviation over 15 seeds. In Figure 4a and 4b we plot respectively the standard and robust errors in the small sample size ($n = 50$) regime for standard training (ST), adversarial training (AT) and convex outer adversarial polytope (COAP) as the perturbation budget ϵ increases. In Figure 4c and 4d we plot respectively the standard and robust errors in the large sample size ($n = 10000$) regime 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.

concentrates around $\sqrt{d}$ with high probability. Hence, the assumption is satisfied when the data dimensionality d is sufficiently high. Further, the second assumption requires that the perturbation budget ϵ is sufficiently close to the separation margin γ. This is consistent with the experimental evidence we presented so far, as the generalisation of certified defences 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 generalisation for three different models: standard training (ST), adversarial training (AT) [16, 11] and convex outer adversarial polytope (COAP) [25, 26]. In Figure 4, we plot robust and standard errors for both small and large sample size regimes as the perturbation budget ϵ increases. The generalisation gap in the small sample size regime between standard and adversarial training was already observed in Clarysse et al. [4] for linear classifiers. Here, we observe a further generalisation gap between AT and COAP in both small and large sample size regimes, which surprisingly worsens in the large sample regime.

C Theoretical results for signal-directed adversaries

We consider a similar generative distribution $\mathbb{P}$ as in [4, 17, 23]: The label $y \in \{+1, -1\}$ is drawn with equal probability and the covariate vector is sampled for an $\gamma > 0$ as $x = [\gamma \operatorname{sgn}(y), \tilde{x}]$, with the random vector $\tilde{x} \in \mathbb{R}^{d-1}$ drawn from a standard normal distribution, $\tilde{x} \sim \mathcal{N}\left(0, \sigma^2 I_{d-1}\right)$. We denote by $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ a dataset of size n i.i.d. drawn from $\mathbb{P}$. We consider the hypothesis class to be the set of one-neuron shallow neural networks $f_\theta : \mathbb{R}^d \to \mathbb{R}$:

$$f_\theta(x) = a \operatorname{ReLU}\left(\theta^\top x\right) + b \tag{33}$$

where $x \in \mathbb{R}^d, \theta \in \mathbb{R}^d, a \in \mathbb{R}, b \in \mathbb{R}$ and the only trainable parameter is θ_1. Moreover, we assume w.l.o.g. that at initialisation $\theta_1 > 0$, and since a and b are not trainable parameters we must have $a > 0$ and $b < 0$ to solve the problem. Note that as our distribution is linearly separable, our hypothesis class includes the ground truth. Further, we consider the binary cross-entropy loss function:

$$L(x, y) = y \log(x) + (1 - y) \log(1 - x) \tag{34}$$

C.1 Adversarial training gradients

Given a sample x with known label $y \in \{-1, 1\}$, we can find the point that minimizes this class by solving the following optimisation problem:

$$J_\epsilon = \min_{\delta} \operatorname{sgn}(y) f_\theta(x + \delta) \quad \text{subject to} \quad x + \delta \in \mathcal{B}_\epsilon(x) \tag{35}$$

For our simplified network we have a closed form solution of this problem:

$$
\begin{aligned}
J_\epsilon &= \min_{x+\delta \in \mathcal{B}_\epsilon(x)} \operatorname{sgn}(y)\left(b + a\,\mathrm{ReLU}\left(\theta^\top (x+\delta)\right)\right) \\
&= \begin{cases} \operatorname{sgn}(y)\left(b + a \max(0, \ell)\right) & \text{if } \ a \operatorname{sgn}(y) > 0 \\ \operatorname{sgn}(y)\left(b + a \max(0, u)\right) & \text{if } \ a \operatorname{sgn}(y) < 0 \end{cases} \qquad (36) \\
&= \begin{cases} \operatorname{sgn}(y)\left(b + a \max(0, \ell)\right) & \text{if } \ \hat{\nu}_2 < 0 \\ \operatorname{sgn}(y)\left(b + a \max(0, u)\right) & \text{if } \ \hat{\nu}_2 > 0 \end{cases}
\end{aligned}
$$

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 weight:

$$
\frac{\partial}{\partial \theta_1} J_\epsilon = \begin{cases} \operatorname{sgn}(y) a (x_1 - \epsilon \operatorname{sgn}(\theta_1)) \mathbf{1}\{\ell > 0\} & \text{if } \ \hat{\nu}_2 < 0 \\ \operatorname{sgn}(y) a (x_1 + \epsilon \operatorname{sgn}(\theta_1)) \mathbf{1}\{u > 0\} & \text{if } \ \hat{\nu}_2 > 0 \end{cases} \tag{37}
$$

and w.r.t. the non-signal weights ($k \geq 2$):

$$
\frac{\partial}{\partial \theta_k} J_\epsilon = \begin{cases} \operatorname{sgn}(y) a x_k \mathbf{1}\{\ell > 0\} & \text{if } \ \hat{\nu}_2 < 0 \\ \operatorname{sgn}(y) a x_k \mathbf{1}\{u > 0\} & \text{if } \ \hat{\nu}_2 > 0 \end{cases} \tag{38}
$$

Finally by the chain-rule we have:

$$
\begin{aligned}
\frac{\partial}{\partial \theta_k} L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y) J_\epsilon\right), y\right) &= \frac{\partial}{\partial J_\epsilon} L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y) J_\epsilon\right), y\right) \cdot \frac{\partial}{\partial \theta_k} J_\epsilon && (39) \\
&= \operatorname{sgn}(y)\left[\boldsymbol{\sigma}\left(\operatorname{sgn}(y) J_\epsilon\right) - \mathbf{1}\{y = 1\}\right] \cdot \frac{\partial}{\partial \theta_k} J_\epsilon && (40) \\
&= -\operatorname{sgn}(y) \boldsymbol{\sigma}\left(-J_\epsilon\right) \begin{cases} a x_k \mathbf{1}\{\ell > 0\} & \text{if } \ \hat{\nu}_2 < 0 \\ a x_k \mathbf{1}\{u > 0\} & \text{if } \ \hat{\nu}_2 > 0 \end{cases} && (41)
\end{aligned}
$$

C.2 COAP gradients

We compute now the dual approximation $\widetilde{J}_\epsilon$ to the optimisation problem (35), as defined in Theorem A.1. In particular we are interested in the cases where $J_\epsilon \neq \widetilde{J}_\epsilon$, that is when the certified and adversarial training objectives differ. First, we consider the case when the neuron is always dead, i.e., $\ell < u < 0$. The dual variables are:

$$
\begin{aligned}
\nu_3 &= -\operatorname{sgn}(y) \\
\hat{\nu}_2 &= -a \operatorname{sgn}(y) \\
\nu_2 &= 0 \\
\hat{\nu}_1 &= 0
\end{aligned} \tag{42}
$$

Hence, there is no mismatch in this case:

$$
\widetilde{J}_\epsilon = \operatorname{sgn}(y) b = J_\epsilon \tag{43}
$$

where the last equality follows from (36).

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

$$
\begin{aligned}
\nu_3 &= -\operatorname{sgn}(y) \\
\hat{\nu}_2 &= -a \operatorname{sgn}(y) \\
\nu_2 &= -a \operatorname{sgn}(y) \\
\hat{\nu}_1 &= -a \operatorname{sgn}(y) \cdot \theta
\end{aligned} \tag{44}
$$

and the dual objective becomes:

$$
\begin{aligned}
\widetilde{J}_\epsilon &= -\nu_3^\top b - \hat{\nu}_1^\top x - \epsilon \left\| [\hat{\nu}_1]_1 \right\|_1 && (45) \\
&= \operatorname{sgn}(y)\left(b + a(\theta^\top x)\right) - \epsilon \| a \operatorname{sgn}(y) \theta_1 \| && (46) \\
&= \begin{cases} \operatorname{sgn}(y)\left(b + a\ell\right) & \text{if } \ a \operatorname{sgn}(y) > 0 \\ \operatorname{sgn}(y)\left(b + au\right) & \text{if } \ a \operatorname{sgn}(y) < 0 \end{cases} && (47) \\
&= J_\epsilon && (48)
\end{aligned}
$$

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:

$$\begin{aligned}
\nu_3 &= -\operatorname{sgn}(y) \\
\hat{\nu}_2 &= -a\operatorname{sgn}(y) \\
\nu_2 &= \ -a\operatorname{sgn}(y)\frac{u}{2\epsilon\left\|\theta_1\right\|_1} \\
\hat{\nu}_1 &= -a\operatorname{sgn}(y)\ \frac{u}{2\epsilon\left\|\theta_1\right\|_1}\cdot\theta
\end{aligned} \tag{49}$$

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

$$\widetilde{J}_\epsilon = \operatorname{sgn}(y)\left(b + au\right) = J_\epsilon \tag{50}$$

hence the dual approximation is tight. When $\nu_2 < 0$ we can rewrite the dual objective as:

$$\widetilde{J}_\epsilon = \operatorname{sgn}(y)\left(b + \frac{au\ell}{2\epsilon\left\|\theta_1\right\|_1}\right) \neq J_\epsilon \tag{51}$$

It follows that the only case when certified training differs from adversarial training 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 weight θ_1 in this case, by the chain rule we have:

$$\frac{\partial}{\partial\theta_1}L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\cdot\widetilde{J}_\epsilon\right), y\right) \tag{52}$$

$$= \frac{\partial}{\partial\widetilde{J}_\epsilon}L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\cdot\widetilde{J}_\epsilon\right), y\right)\cdot\frac{\partial}{\partial\theta_1}\widetilde{J}_\epsilon \tag{53}$$

$$= \operatorname{sgn}(y)\left[\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\cdot\widetilde{J}_\epsilon\right) - \mathbf{1}\{y=1\}\right]\cdot\frac{\partial}{\partial\theta_1}\widetilde{J}_\epsilon \tag{54}$$

$$= -\frac{a\operatorname{sgn}(y)\boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon\right)}{2\epsilon}\left(\frac{\ell}{\left\|\theta_1\right\|_1}(x_1 + \epsilon\operatorname{sgn}(\theta_1)) + u\frac{x_1\left\|\theta_1\right\|_1 - \theta^\top x\operatorname{sgn}(\theta_1)}{\theta_1^2}\right) \tag{55}$$

and finally we compute the partial derivative w.r.t. the non-signal weight $\theta_k (k \geq 2)$:

$$\frac{\partial}{\partial\theta_k}L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\cdot\widetilde{J}_\epsilon\right), y\right) \tag{56}$$

$$= \frac{\partial}{\partial\widetilde{J}_\epsilon}L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\cdot\widetilde{J}_\epsilon\right), y\right)\cdot\frac{\partial}{\partial\theta_k}\widetilde{J}_\epsilon \tag{57}$$

$$= \operatorname{sgn}(y)\left[\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\cdot\widetilde{J}_\epsilon\right) - \mathbf{1}\{y=1\}\right]\cdot\frac{\partial}{\partial\theta_k}\widetilde{J}_\epsilon \tag{58}$$

$$= -\frac{ax_k\operatorname{sgn}(y)\boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon\right)\theta^\top x}{\epsilon\left\|\theta_1\right\|_1} \tag{59}$$

C.3 Auxiliary lemmas

Lemma C.1. *Let f_θ be the neural network defined in Equation* (33)*. We define the robust risk $\mathbf{R}_\epsilon$ of f_θ as follows:*

$$\mathbf{R}_\epsilon(\theta) := \mathbb{P}_{(x,y)}\left[\exists z \in \mathcal{B}_\epsilon(x) : y \neq \operatorname{sgn}\left(f_\theta(z)\right)\right] \tag{60}$$

Then, $\mathbf{R}_\epsilon(\theta)$ is monotonically decreasing in $\left\|\theta_1\right\|_2$.

Proof.

$$\mathbf{R}_\epsilon(\theta) := \mathbb{P}_{(x,y)}\left[\exists z \in \mathcal{B}_\epsilon(x) : y \neq \operatorname{sgn}\left(f_\theta(z)\right)\right] \tag{61}$$

$$= \frac{1}{2}\left(\mathbb{P}_x\left[\theta^\top x < \|b\|_1 \mid x_1 = \gamma - \epsilon\right] + \mathbb{P}_x\left[\theta^\top x > \|b\|_1 \mid x_1 = \epsilon - \gamma\right]\right) \tag{62}$$

$$= \frac{1}{2}\left(\mathbb{P}_x\left[\sum_{i=2}^d x_i\theta_i < -\theta_1(\gamma-\epsilon) + \|b\|_1\right] + \mathbb{P}_x\left[\sum_{i=2}^d x_i\theta_i > \theta_1(\gamma-\epsilon) + \|b\|_1\right]\right) \tag{63}$$

$$= \frac{1}{2}\left(\Phi\left(-\frac{(\gamma-\epsilon)\|\theta_1\|_2}{\sigma\|\theta_{2:d}\|_2} + \frac{\|b\|_1}{\sigma\|\theta_{2:d}\|_2}\right) + \Phi\left(-\frac{(\gamma-\epsilon)\|\theta_1\|_2}{\sigma\|\theta_{2:d}\|_2} - \frac{\|b\|_1}{\sigma\|\theta_{2:d}\|_2}\right)\right) \tag{64}$$

hence $\mathbf{R}_\epsilon(\theta)$ is monotonically decreasing in $\|\theta_1\|_2$ and the statement follows. □

Lemma C.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:*

$$f(x) \leq 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} \tag{65}$$

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

$$g(x) = 1 + x + \frac{x^2}{2!} + \cdots + \frac{x^n}{n!} - \exp(x) \tag{66}$$

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

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

  Therefore, for any x,

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

  since n is even.

□

Lemma C.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:*

$$\phi(0)\left(y - x + \frac{x^3}{6}\right) \leq \Phi(y) - \Phi(x) \tag{67}$$

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:

$$f(x) = \operatorname{erf}(x) - \frac{2x}{\sqrt{\pi}} \tag{68}$$

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:

$$f'(x) = \frac{2}{\sqrt{\pi}}\exp(-x^2) - \frac{2}{\sqrt{\pi}} \tag{69}$$

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:

$$g(x) = \frac{2}{\sqrt{\pi}}(x - x^3/3) - \operatorname{erf}(x) \tag{70}$$

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:

$$g'(x) = \frac{2}{\sqrt{\pi}}(1 - x^2 - \exp(-x^2)) \tag{71}$$

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:

$$\begin{aligned}\Phi(b) - \Phi(a) &= \frac{1}{2}\left(\mathrm{erf}(b/\sqrt{2}) - \mathrm{erf}(a/\sqrt{2})\right) &(72)\\ &\geq \frac{1}{\sqrt{\pi}}\left(\frac{b}{\sqrt{2}} - \frac{a}{\sqrt{2}} + \frac{a^3}{6\sqrt{2}}\right) &(73)\\ &= \phi(0)\left(b - a + \frac{a^3}{6}\right) &(74)\end{aligned}$$

which concludes the proof. $\square$

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

$$f(r) = \gamma^2 - \epsilon^2 - 2\sigma^2 r^2 - \sigma r \frac{(\gamma - 3\epsilon)\phi(\beta) - (\gamma + \epsilon)\phi(\alpha)}{\Phi(\beta) - \Phi(\alpha)} \tag{75}$$

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

$$\frac{5 + 2\sqrt{3}}{13}\gamma < \epsilon < \gamma \tag{76}$$

Then, we have:

$$f(r) < 0, \;\; \forall r > \sqrt{\frac{24\gamma^3}{\sigma^2}} \tag{77}$$

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

$$\Phi(\beta) - \Phi(\alpha) \geq \left(\frac{2\epsilon}{r\sigma} - \frac{(\gamma+\epsilon)^3}{6\sigma^3 r^3}\right)\phi(0), \;\; \alpha < \beta < 0 \tag{78}$$

Next, we can upper-bound f:

$$\begin{aligned}f(r) &\leq \gamma^2 - \epsilon^2 - 2\sigma^2 r^2 - \sigma r \frac{(\gamma - 3\epsilon)\phi(\beta) - (\gamma + \epsilon)\phi(\alpha)}{\left(\frac{2\epsilon}{\sigma r} - \frac{(\gamma+\epsilon)^3}{6\sigma^3 r^3}\right)\phi(0)} &(79)\\ &\leq \gamma^2 - \epsilon^2 - 2\sigma^2 r^2 - \sigma^2 r^2 \frac{(\gamma - 3\epsilon)\phi(0) - (\gamma + \epsilon)\phi(\alpha)}{\left(2\epsilon - \frac{(\gamma+\epsilon)^3}{6r^2\sigma^2}\right)\phi(0)} &(80)\\ &= \gamma^2 - \epsilon^2 - 2\sigma^2 r^2 - \sigma^2 r^2 \frac{(\gamma - 3\epsilon) - (\gamma + \epsilon)\exp(-\alpha^2/2)}{2\epsilon - \frac{(\gamma+\epsilon)^3}{6\sigma^2 r^2}} &(81)\end{aligned}$$

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

$$\exp(x) \leq 1 + x - x^2/2, \;\; \forall x \leq 0 \tag{82}$$

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

$$f(r) \leq \gamma^2 - \epsilon^2 - 2\sigma^2 r^2 - \sigma^2 r^2 \frac{(\gamma - 3\epsilon) - (\gamma + \epsilon)(1 - \frac{(\gamma+\epsilon)^2}{2r^2\sigma^2} + \frac{(\gamma+\epsilon)^4}{8r^4\sigma^4})}{2\epsilon - \frac{(\gamma+\epsilon)^3}{6r^2\sigma^2}} \tag{83}$$

which can be further simplified:

$$f(r) \leq \gamma^2 - \epsilon^2 - 2\sigma^2 r^2 - \sigma^2 r^2 \frac{(\gamma - 3\epsilon) - (\gamma + \epsilon)(1 - \frac{(\gamma+\epsilon)^2}{2r^2\sigma^2} + \frac{(\gamma+\epsilon)^4}{8r^4\sigma^4})}{2\epsilon - \frac{(\gamma+\epsilon)^3}{6r^2\sigma^2}} \tag{84}$$

$$= \frac{(\gamma - 7\epsilon)(\gamma + \epsilon)^4 + 4r^2\sigma^2(\gamma + \epsilon)(\gamma^2 - 10\gamma\epsilon + 13\epsilon^2)}{4(\gamma + \epsilon)^3 - 48r^2\sigma^2\epsilon} \tag{85}$$

$$= u(r) \tag{86}$$

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. $u(r) < 0$. Finally, for the sake of clarity we can further simplify the condition on r:

$$r > \sqrt{\frac{24\gamma^3}{\sigma^2}} > \sqrt{\max\left(\frac{(7\epsilon - \gamma)(\gamma + \epsilon)^4}{4\sigma^2(\gamma^2 - 10\gamma\epsilon + 13\epsilon^2)}, \frac{(\gamma + \epsilon)^3}{12\sigma^2\epsilon}\right)} \tag{87}$$

which concludes the proof. □

Lemma C.5. *Let f_θ the network defined in Equation* (33)*, $\widetilde{J}_\epsilon$ be the COAP training objective, $\boldsymbol{\sigma}(\cdot)$ the sigmoid function and $\mathcal{I}^\star = \{(x, y) : \ell < 0 \wedge u > 0 \wedge y = 1\}$. Assume that:*

$$\frac{\|\theta_{2:d}\|_2}{\|\theta_1\|_2} > \sqrt{\frac{24\gamma^3}{\sigma^2}} \;\; \textit{and} \;\; \frac{5 + 2\sqrt{3}}{13}\gamma < \epsilon < \gamma \tag{88}$$

Then, we have:

$$\mathbb{E}_{(x,y)}\left[\nabla_{\theta_1} L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\widetilde{J}_\epsilon\right), y\right) \mid (x, y) \in \mathcal{I}^\star\right] > 0 \tag{89}$$

Proof. Our strategy will be to lower-bound the expectation in Equation (89) with some strictly positive quantity. We define $Z = \sum_{i=2}^{d} \theta_i x_i$ and plug-in the gradient computed in Appendix C.2:

$$\mathbb{E}_{(x,y)}\left[\nabla_{\theta_1} L\left(\operatorname{sgn}(y)\boldsymbol{\sigma}\left(\widetilde{J}_\epsilon\right), y\right) \mid (x, y) \in \mathcal{I}^\star\right] \tag{90}$$

$$= \mathbb{E}_{(x,y)}\left[\frac{a\boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon\right)}{2\epsilon}\left(-\frac{\ell}{\theta_1}(\gamma + \epsilon) + u\frac{\sum_{i=2}^{d} x_i\theta_i}{\theta_1^2}\right) \mid (x, y) \in \mathcal{I}^\star\right] \tag{91}$$

$$= \frac{a}{2\theta_1\epsilon}\mathbb{E}_{(x,y)}\left[\boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon\right)\left(-\ell(\gamma + \epsilon) + u\frac{Z}{\theta_1}\right) \mid (x, y) \in \mathcal{I}^\star\right] \tag{92}$$

$$= \frac{a}{2\theta_1\epsilon}\mathbb{E}_{(x,y)}\left[\boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon\right)u\frac{Z}{\theta_1} - \boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon\right)\ell(\gamma + \epsilon) \mid (x, y) \in \mathcal{I}^\star\right] \tag{93}$$

Now, we observe that Z is always negative on the set $\mathcal{I}^\star$:

$$(x, y) \in \mathcal{I}^\star \implies -\theta_1(\gamma + \epsilon) < \sum_{i=2}^{d} \theta_i x_i < -\theta_1(\gamma - \epsilon) < 0 \tag{94}$$

since we need to satisfy the constraint $\ell < 0 < u$. Further, from Equation (51) we have:

$$(x, y) \in \mathcal{I}^\star \implies \boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon\right) \geq \frac{1}{2} \tag{95}$$

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

$$\mathbb{E}_{(x,y)}\left[\nabla_{\theta_1} L\left(\operatorname{sgn}(y)\boldsymbol{\sigma}\left(\widetilde{J}_\epsilon\right), y\right) \mid (x, y) \in \mathcal{I}^\star\right] \tag{96}$$

$$= \frac{a}{2\theta_1\epsilon}\mathbb{E}_{(x,y)}\left[\boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon\right)u\frac{Z}{\theta_1} - \boldsymbol{\sigma}\left(-\widetilde{J}_\epsilon\right)\ell(\gamma + \epsilon) \mid (x, y) \in \mathcal{I}^\star\right] \tag{97}$$

$$\geq \frac{a}{2\theta_1\epsilon}\mathbb{E}_{(x,y)}\left[u\frac{Z}{\theta_1} - \frac{\gamma + \epsilon}{2}\ell \mid (x, y) \in \mathcal{I}^\star\right] \tag{98}$$

Hence for our purpose it is enough to show that this lower-bound is strictly positive:

$$\mathbb{E}_{(x,y)}\left[u\frac{Z}{\theta_1} - \frac{\gamma+\epsilon}{2}\ell \mid (x,y) \in \mathcal{I}^\star\right] > 0 \tag{99}$$

we can further expand this expression:

$$\begin{aligned}
&\mathbb{E}_{(x,y)}\left[u\frac{Z}{\theta_1} - \frac{\gamma+\epsilon}{2}\ell \mid (x,y) \in \mathcal{I}^\star\right] && (100)\\
&= -(\gamma^2-\epsilon^2)\theta_1^2 + (\gamma+\epsilon)\theta_1\mathbb{E}\left[Z \mid (x,y)\in\mathcal{I}^\star\right] + 2\mathbb{E}\left[Z^2 \mid (x,y)\in\mathcal{I}^\star\right] && (101)
\end{aligned}$$

Note that $Z \mid (x,y) \in \mathcal{I}^\star$ is a truncated normal with $\alpha = -\frac{\theta_1(\gamma+\epsilon)}{\sigma\|\theta_{2:d}\|_2}$, $\beta = -\frac{\theta_1(\gamma-\epsilon)}{\sigma\|\theta_{2:d}\|_2}$. Hence, we can plug the expectations in and obtain the following:

$$\begin{aligned}
&-(\gamma^2-\epsilon^2)\theta_1^2 + \theta_1(\gamma+\epsilon)\mathbb{E}\left[Z \mid (x,y)\in\mathcal{I}^\star\right] + 2\mathbb{E}\left[Z^2 \mid (x,y)\in\mathcal{I}^\star\right] && (102)\\
&= -(\gamma^2-\epsilon^2)\theta_1^2 + 2\sigma^2\left\|\theta_{2:d}\right\|_2^2 + \sigma\left\|\theta_{2:d}\right\|_2\theta_1\frac{(\gamma-3\epsilon)\phi(\beta)-(\gamma+\epsilon)\phi(\alpha)}{\Phi(\beta)-\Phi(\alpha)} && (103)\\
&\propto -(\gamma^2-\epsilon^2) + 2\sigma^2 r^2 + \sigma r\frac{(\gamma-3\epsilon)\phi(\beta)-(\gamma+\epsilon)\phi(\alpha)}{\Phi(\beta)-\Phi(\alpha)} && (104)\\
&= -f(r) && (105)
\end{aligned}$$

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

$$f(r) < 0,\ \forall r > \sqrt{\frac{24\gamma^3}{\sigma^2}} \tag{106}$$

which concludes the proof. □

C.4 Proof of Theorem B.1

Theorem B.1. *Let $\bar{\theta}$ and $\tilde{\theta}$ be the network parameters after one step of gradient descent with respect to AT and COAP objectives. Let,*

$$\frac{\|\theta_{2:d}\|_2}{\|\theta_1\|_2} > \sqrt{\frac{24\gamma^3}{\sigma^2}} \;\; \textit{and} \;\; \frac{2}{3}\gamma < \epsilon < \gamma \tag{31}$$

where θ are the network parameters at initialization. Then, COAP yields higher robust risk than AT:

$$\mathbf{R}_\epsilon(\tilde{\theta}) > \mathbf{R}_\epsilon(\bar{\theta}) \tag{32}$$

Proof. Let J_ϵ be the adversarial training inner maximisation as defined in Equation (35). Then, AT solves the following optimisation problem:

$$\min_\theta \mathbb{E}_{(x,y)}\left[L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)J_\epsilon\right), y\right)\right] \tag{107}$$

Similarly, let $\widetilde{J}_\epsilon$ be the COAP dual approximation to the inner maximization described in Appendix C.2. Then, COAP solves the following optimisation problem:

$$\min_\theta \mathbb{E}_{(x,y)}\left[L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\widetilde{J}_\epsilon\right), y\right)\right] \tag{108}$$

In what follows, $\bar{\theta}^{(t)}$ refers to the parameter trained with adversarial training at iteration t and $\tilde{\theta}^{(t)}$ to the COAP counterpart. After one step of gradient descent, we have:

$$\left\|\bar{\theta}_{2:d}^{(1)}\right\|_2 = \left\|\tilde{\theta}_{2:d}^{(1)}\right\|_2 \tag{109}$$

since we only train the signal component. Further, from Lemma C.1 we have that adversarial training yields smaller robust risk than certified if the following to holds:

$$\left\|\bar{\theta}_1^{(1)}\right\|_2 > \left\|\tilde{\theta}_1^{(1)}\right\|_2 \tag{110}$$

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

$$\mathbb{E}_{(x,y)}\left[\nabla_{\bar{\theta}_1} L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)J_\epsilon\right), y\right)\right] < \mathbb{E}_{(x,y)}\left[\nabla_{\tilde{\theta}_1} L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\widetilde{J}_\epsilon\right), y\right)\right] \tag{111}$$

In Appendix C.2 and C.1 we compute gradients for both objectives. In particular, we have that the gradients of adversarial and certified training differ only on the set $\mathcal{I}^\star$:

$$(x,y) \notin \mathcal{I}^\star \implies \nabla_{\bar{\theta}_1} L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)J_\epsilon\right), y\right) = \nabla_{\tilde{\theta}_1} L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\widetilde{J}_\epsilon\right), y\right) < 0 \tag{112}$$

and

$$(x,y) \in \mathcal{I}^\star \implies 0 = \nabla_{\bar{\theta}_1} L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)J_\epsilon\right), y\right) \neq \nabla_{\tilde{\theta}_1} L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\widetilde{J}_\epsilon\right), y\right) \tag{113}$$

Hence for our purpose it is enough to show that:

$$\mathbb{E}_{(x,y)}\left[\nabla_{\tilde{\theta}_1} L\left(\boldsymbol{\sigma}\left(\operatorname{sgn}(y)\widetilde{J}_\epsilon\right), y\right) \mid (x,y) \in \mathcal{I}^\star\right] > 0 \tag{114}$$

which is a direct consequence of Lemma C.5. Thus we have that:

$$\left\|\bar{\theta}_1\right\|_2 > \left\|\tilde{\theta}_1\right\|_2 \quad \text{and} \quad \left\|\bar{\theta}_{2:d}\right\|_2 = \left\|\tilde{\theta}_{2:d}\right\|_2 \tag{115}$$

and from Lemma C.1 follows:

$$\mathbf{R}_\epsilon(\tilde{\theta}) > \mathbf{R}_\epsilon(\bar{\theta}) \tag{116}$$

which concludes the proof. □

D Experimental details

D.1 Synthetic experiments with signal-directed adversaries

Below we provide detailed experimental details to reproduce Figure 4. For all the experiments, we use the one hidden layer architecture defined in Equation (2) 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. 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.

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

Standard training. We train the network to minimise the cross-entropy loss.

Adversarial training [16, 11]. 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.

Certified training [25, 26]. At each epoch, we compute upper and lower bounds u and ℓ as described in Proposition A.2. We then train the network to minimize the upper-bound on robust error derived in Theorem A.1.

D.2 Synthetic experiments with ℓ_2 adversaries

Below we provide detailed experimental details to reproduce Figure 3.

For the spheres dataset, we consider a data distribution similar to Gilmer et al. [10] that consists of two concentric spheres in d dimensions: we generate a random $x \in \mathbb{R}$ where $\|x\|_2$ is either γ_{min} or γ_{max} , with equal probability assigned to each norm. We associate with each x a label y such that $y = 0$ if $\|x\|_2 = \gamma_{\text{min}}$ and $y = 1$ if $\|x\|_2 = \gamma_{\text{max}}$. We can sample uniformly from this distribution by sampling $z \sim \mathcal{N}\left(0, I_d\right)$ and then setting $x = \frac{z}{\|z\|_2}\gamma_{\text{min}}$ or $x = \frac{z}{\|z\|_2}\gamma_{\text{max}}$.

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 $\gamma_{\text{max}} = 12$.

For all the experiments, we use the MLP architecture with $W = 100$ neurons in each hidden layer and $\operatorname{ReLU}(\cdot)$ 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 among the classifiers that interpolate the training set. We perform the attacks to evaluate robust risk at test-time using Auto-PGD [5] with 100 iterations and 5 random restarts. We use both the cross-entropy and difference of logits loss to prevent gradient masking. For all attacks we use the implementation provided in AutoAttack [5] with some adjustments to allow for non-image inputs.

Standard training. We train the network to minimise the cross-entropy loss.

Adversarial training [16, 11]. We train the network to minimise the robust cross-entropy loss. At each epoch, we search for adversarial examples using Auto-PGD with a budget of 10 steps and 1 random restart. Then, we update the weights using a gradient with respect to this example.

Certified training [25, 26]. We consider the tightest convex relaxation, i.e. the convex outer adversarial polytope derived in Wong and Kolter [25]. We train the network to minimise their upper-bound on the robust error. Our implementation is based on the code released by the authors.

D.3 Image experiments

Below we provide experimental details to reproduce Figures 2.

For CIFAR-10, we train the residual network (ResNet) with the same structure used in Wong et al. [26]. For MNIST, we train the large convolutional neural network (CNN) architecture (see Table 1) introduced in Wong et al. [26], with four convolutional layer and two fully connected layers of 512 units.

For MNIST we use full 28×28 images without any augmentations and normalization. For CIFAR-10 we use random horizontal flips and random crops as data augmentation, and normalize images according to per-channel statistics.

For all ℓ_p-adversaries considered, we evaluate the robust error using the most expensive version of AutoAttack (AA+) [5] , which includes 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)
- targeted FAB (9 target classes)

AT training details For MNIST, we train 100 epochs using Adam optimiser [13] 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 and 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 the inner optimisation, adversarial examples are generated with 10 iterations of Auto-PGD [5].

COAP training details We follow the settings proposed by the authors and report them here. For MNIST , we use the Adam optimiser 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 experiments, we use random projection of 50 dimensions.

CROWN-IBP training details For MNIST, we train 200 epochs with a batch size of 256. We use Adam optimizer and set 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 [28].

CNN-SMALL	CNN-LARGE
CONV 16 $4 \times 4 + 2$	CONV 32 $3 \times 3 + 1$
CONV 32 $4 \times 4 + 1$	CONV 32 $4 \times 4 + 2$
FC 100	CONV 64 $3 \times 3 + 1$
	CONV 64 $4 \times 4 + 2$
	FC 512
	FC 512

Table 1: Model architectures. 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.