# How robust accuracy suffers from certified training with convex relaxations

Piersilvio De Bartolomeis[1], Jacob Clarysse[1], Amartya Sanyal[2], and Fanny Yang[1]

[1]*Department of Computer Science, ETH Zürich*
[2]*Max Planck Institute for Intelligent Systems, Tübingen*

**Abstract**

Adversarial attacks pose significant threats to deploying state-of-the-art classifiers in safety-critical applications. Two classes of methods have emerged to address this issue: empirical defences and certified defences. Although certified defences come with robustness guarantees, empirical defences such as adversarial training enjoy much higher popularity among practitioners. In this paper, we systematically compare the standard and robust error of these two robust training paradigms across multiple computer vision tasks. We show that in most tasks and for both $\ell_\infty$-ball and $\ell_2$-ball threat models, certified training with convex relaxations suffers from worse standard and robust error than adversarial training. We further explore how the error gap between certified and adversarial training depends on the threat model and the data distribution. In particular, besides the perturbation budget, we identify as important factors the shape of the perturbation set and the implicit margin of the data distribution. We support our arguments with extensive ablations on both synthetic and image datasets.

## 1 Introduction

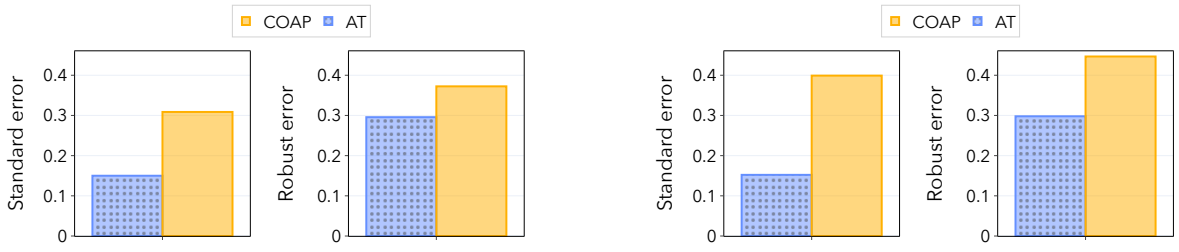
State-of-the-art classifiers are known to be vulnerable to adversarial perturbations of the input. These perturbations, whether perceptible or imperceptible, can manipulate the input in such a way as to drastically alter the classifier's predictions [1, 35]. Hence, robustness to such *adversarial attacks* has become a crucial design goal when deploying machine learning models in safety-critical applications.

More precisely, for any given distribution $\mathcal{D}$ and loss function $L$, we aim to find a classifier $f_\theta : \mathbb{R}^d \to \mathbb{R}^k$ parameterised by $\theta \in \mathbb{R}^p$ that minimises the *robust loss*

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

the threat model $\mathcal{B}_\epsilon := \{\delta : \|\delta\|_p \leq \epsilon\}$ is a bounded $\ell_p$-ball centred at the origin, and $\mathbf{R}_\epsilon(\theta)$ denotes the robust error when $L$ is the 0-1 loss. The main challenge in solving the optimisation problem presented in Equation (1) is that, when $\theta$ parameterises a neural network, the inner maximisation becomes a non-convex optimisation problem and computationally intractable [11, 37]. In order to overcome the computational barrier, two methods of approximation have been widely discussed in the literature so far: *empirical* defences such as adversarial training and *certified* defences, including randomised smoothing and convex relaxations.

Adversarial training (AT) [8, 20] is one of the most popular empirical defences to date: it minimises the empirical robust loss in Equation (1) by approximately solving the inner maximisation with iterative first-order optimisation methods. Although adversarial training is favoured for its simplicity and computational efficiency, it lacks the robustness guarantees that are essential in safety-critical applications. In particular, even though




(a) $\ell_\infty$-ball (b) $\ell_2$-ball

Figure 1: Standard and robust error of adversarial (dotted bars) and certified training (solid bars) on the CIFAR-10 test set. Models were trained for robustness against: (a) $\ell_\infty$-ball perturbations with radius $\epsilon_\infty = 1/255$, and (b) $\ell_2$-ball perturbations with radius $\epsilon_2 = 36/255$. We report the best performing certified training method among many convex relaxations (FAST-IBP [32], IBP [9], CROWN-IBP [40, 43] and COAP [38, 39]). We refer the reader to Section 2 for further details on the models and robust evaluation.

empirical evaluation tools like AutoAttack [5] can provide non-trivial lower bounds of the robust error, it is worth noting that these lower bounds might substantially underestimate the true robust error.

To address this limitation, certified defences train neural networks for which it is possible to obtain non-trivial upper bounds of the robust error. In this paper, we focus our attention on *certified training*, i.e. the widely studied class of certified defences based on convex relaxations. The key idea underlying certified training methods is to solve a convex relaxation of the optimisation problem in Equation (1), by replacing the non-convex ReLU constraint sets with larger convex sets [6, 28, 38].

While certified training offers robustness guarantees, adversarial training is frequently the preferred choice among practitioners. Beyond computational efficiency issues, this preference stems from the suboptimal standard and robust error achieved with this class of certified defences [21]. As we discuss in Section 5, so far, the drawbacks of certified training have received considerably less attention in the literature than other types of defences, such as randomised smoothing [2, 14, 23, 34]. Further, the most prominent works on certified training [9, 24, 32, 38, 40, 43] do not include a direct comparison of their methods with adversarial training, nor are there any existing explanations for the error gap between these two paradigms of robust training.

In this paper, we fill this void in the literature and systematically compare the robust and standard error of certified and adversarial training across three widely adopted computer vision datasets. More specifically, our contributions can be summarised as follows:

- In Section 2, we show that, in most tasks and for both $\ell_\infty$-ball and $\ell_2$-ball threat models, certified training suffers from worse standard and robust error than adversarial training (e.g. see Figures 1a and 1b).
- In Section 3, we explore how the error gap between certified and adversarial training depends on the
  (i) **Threat model** – via the perturbation budget and the shape of the perturbation set.
  (ii) **Data distribution** – via the implicit margin of the data, which denotes the minimum distance in feature space between any two data points from different classes.
- In Section 4, we propose a possible explanation for the error gap between certified and adversarial training. Specifically, through a series of ablation studies and illustrations, we show that the above factors influence the number of *unstable neurons*, which in turn affects the error gap.

## 2 Systematic comparison between adversarial and certified training

In this section, we systematically compare the standard and robust error of adversarial and certified training, for both $\ell_2$-ball and $\ell_\infty$-ball threat models. We consider three widely adopted computer vision datasets: Tiny ImageNet [15], CIFAR-10 [13] and MNIST [16]. We note that certified training with convex relaxations does not scale to larger datasets such as ImageNet, hence we omit them from our comparison.

### 2.1 Experimental setup

**Adversarial defences** Among certified training methods, we focus on the most popular ones and cover a wide range of convex relaxations techniques. In particular, we consider:

- Convex outer adversarial polytope (COAP) [38, 39], which uses the DeepZ relaxation [33] and achieves state-of-the-art certified robustness under $\ell_2$-ball perturbations.
- Interval bound propagation (IBP) [9], which uses the Box relaxation [22].
- Fast IBP [32], which is a computationally more efficient version of IBP and achieves state-of-the-art certified robustness under $\ell_\infty$-ball perturbations.
- CROWN-IBP [40], which combines the tight convex relaxation CROWN [41] with IBP.

We compare the certified training methods against the most popular empirical defence to date, adversarial training (AT) [8, 20], which is the go-to approach for adversarial robustness among practitioners.

**Models and evaluation** For CIFAR-10, we train a residual network (ResNet) and for MNIST we train a vanilla convolutional neural network (CNN). Both architectures were introduced in Wong et al. [39] as standard benchmarks for certified training. For Tiny ImageNet, we train the WideResNet introduced in Xu et al. [40]. We refer the reader to Appendix A.1 for complete experimental details.

We evaluate the performance of these models according to standard and robust error. More precisely, standard error is defined as in Equation (1) by setting $L$ to the 0-1 loss function and $\epsilon = 0$. As an approximation for the population quantity, we compute the empirical standard and robust error on the test set. Further, since exact evaluation of the robust error is computationally infeasible, we evaluate the models with AutoAttack (AA+) [5], which is widely considered to be one of the most reliable empirical evaluation tools to date.

Our threat models in this section correspond to $\ell_2$-ball and $\ell_\infty$-ball perturbations. For the sake of clarity, we present our experimental results with a single perturbation budget for each dataset and threat model, as reported in Table 1 (caption). Nevertheless, we provide additional experimental results with a wide range of perturbation budgets in Appendix B. Note that we intentionally choose small perturbation budgets for $\ell_2$-ball perturbations, as larger budgets result in trivial standard and robust error for certified training.

### 2.2 Standard and robust error gap

We present the results of our comparison in Table 1. Generally, for both $\ell_2$-ball and $\ell_\infty$-ball threat models, we observe that certified training suffers from worse standard and robust error than adversarial training. An interesting exception to this pattern is found with the MNIST dataset and the $\ell_\infty$-ball threat model, where the best convex relaxation achieves smaller robust error than adversarial training.

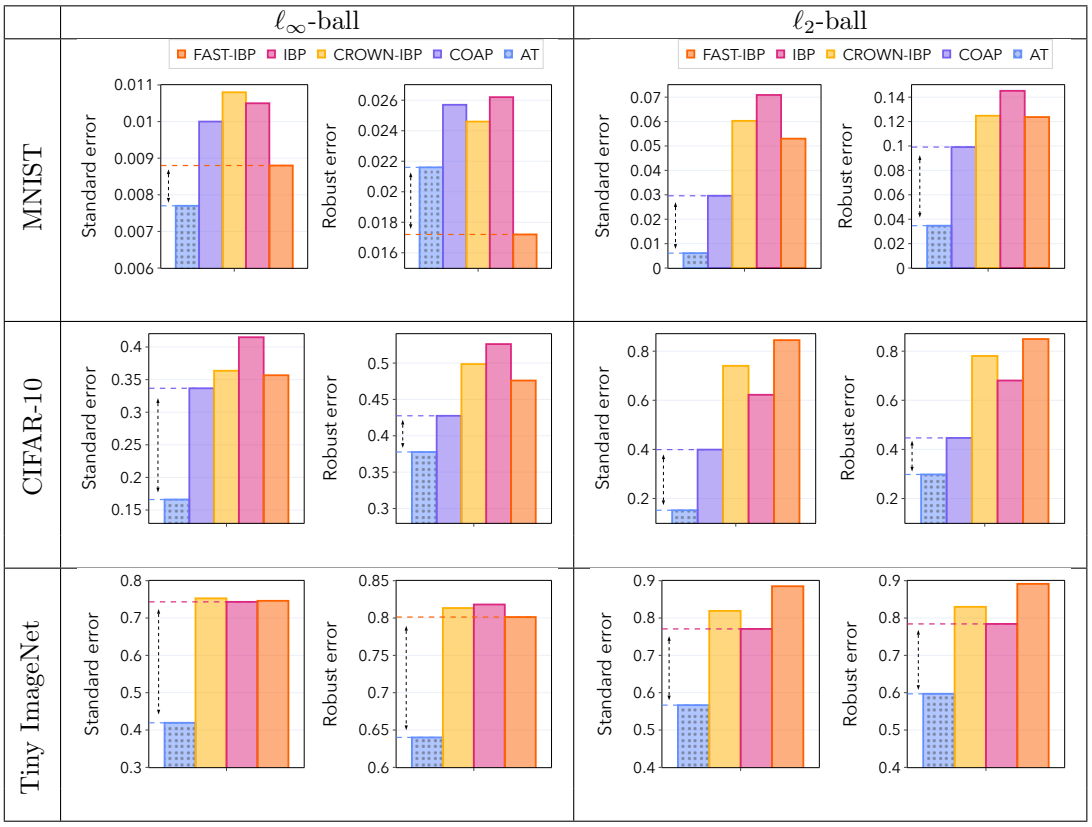



Table 1: Standard and robust error of adversarial (dotted bar) and certified training (solid bar, tightest to loosest convex relaxation from left to right) on the MNIST ($\epsilon_\infty = 0.1, \epsilon_2 = 0.75$), CIFAR-10 ($\epsilon_\infty = 2/255, \epsilon_2 = 36/255$) and Tiny ImageNet ($\epsilon_\infty = 1/255, \epsilon_2 = 5/255$) test sets.

**Discussion** For MNIST, in Table 1 (first row) we observe a gap in standard error of up to 2% between adversarial and certified training, under both $\ell_2$-ball and $\ell_\infty$-ball perturbations. Similarly, for robust error under $\ell_2$-ball perturbations, we observe a 6% gap between the best convex relaxation and AT. However, under $\ell_\infty$-ball perturbations the best convex relaxation achieves lower robust error than AT.

For CIFAR-10, in Table 1 (second row) we observe a standard error gap of up to 25% between the best convex relaxation and AT, under both $\ell_2$-ball and $\ell_\infty$-ball perturbations. Unlike MNIST, for robust error, we observe a gap of up to 15%, under both threat models.

For Tiny ImageNet, in Table 1 (third row) we observe a significant standard error gap of up to 35% between the best convex relaxation and AT, under both $\ell_2$-ball and $\ell_\infty$-ball perturbations. Similarly, we observe a significant robust error gap for both threat models, of up to 20%. Note that COAP does not scale to Tiny ImageNet, and hence we omit it from our comparison.

We observe that even though COAP is not traditionally recognised as a state-of-the-art certified training method under $\ell_\infty$-ball perturbations, given its subpar performance for larger perturbation budgets, it surprisingly surpasses all other convex relaxations for smaller perturbation budgets on the CIFAR-10 test set. Additionally, it generally performs better under $\ell_2$-ball perturbations for both MNIST and CIFAR, marking it as an effective convex relaxation in these settings.

**Tight convex relaxations are better for $\ell_2$-ball perturbations** We observe a significantly more pronounced standard and robust error gap for $\ell_2$-ball perturbations compared to $\ell_\infty$-ball perturbations. Surprisingly, we find that the paradox of certified training [10] — the notion that loose interval-based training often yields better performance than tighter relaxations — does not hold for $\ell_2$-ball perturbations and for $\ell_\infty$-ball perturbations when the perturbation budget is small (e.g. CIFAR-10 with $\epsilon_\infty = \frac{2}{255}$). Instead, we notice that tighter convex relaxations actually enhance performance in this setting. Notably, COAP [38, 39], which is the tightest among the convex relaxations considered, stands out as the best-performing method against $\ell_2$-ball perturbations on both MNIST and CIFAR-10 test sets. Conversely, IBP and FAST-IBP, which are the loosest convex relaxation considered, emerge as the least effective.

# 3 Which factors influence the error gap?

In this section, we explore how the standard and robust error gap between certified and adversarial training depends on the threat model and the data distribution. Besides the perturbation budget $\epsilon$, we find that the shape of the perturbation set plays a crucial role: the more *aligned* it is with the shortest path towards the (robust Bayes optimal) decision boundary, the larger the error gap. Furthermore, the natural distribution of the data also affects the phenomenon via the *implicit margin* $\gamma$: a small minimum distance between two classes in feature space leads to a large error gap. We discuss the intuition underlying these factors in Section 4.

## 3.1 Experimental setup

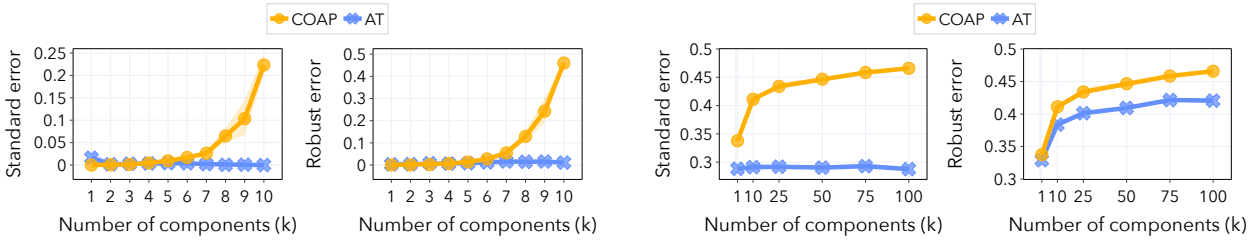
**Datasets** We use one synthetic dataset and one image dataset to study the effect of the aforementioned factors on the error gap. For our controlled synthetic setting, we consider the concentric spheres distribution studied in Gilmer et al. [7], Nagarajan and Kolter [25]. To sample from the concentric spheres distribution with radii $0 < R_0 < R_1$, we first draw a binary label $y \in \{0, 1\}$ with equal probability, and then a covariate vector $x \in R^d$ uniformly from the sphere of radius $R_y$. Note that the implicit margin of the concentric spheres distribution is given by $\gamma := R_1 - R_0$. For image datasets, we present our ablations on CIFAR-10 but observe similar trends on MNIST and provide additional experiments in Appendix C.2.

**Models and robust evaluation** For all CIFAR-10 experiments, we train a vanilla convolutional neural network (CNN) as specified in Wong et al. [39]. For all concentric spheres experiments, we train a multilayer perceptron with $W = 100$ neurons and one hidden layer. For simplicity of exposition, we compare adversarial training with COAP throughout this section. We choose COAP as a representative for certified training since it outperforms the other convex relaxations under $\ell_2$-ball perturbations. We refer the reader to Appendix A for complete experimental details.

For the ablations on the perturbation budget and the margin, we specifically concentrate on $\ell_2$-ball perturbations, as the phenomena we aim to investigate are more prominent in this context. Nevertheless, we observe similar trends for $\ell_\infty$-ball perturbations, and we provide additional experiments in Appendix C.1. As for the previous section, we evaluate the empirical robust error for $\ell_2$-ball perturbations using AutoAttack (AA+) [5].

## 3.2 Factor (i): Shape of the perturbation set

The first factor we investigate is the shape of the perturbation set. In particular, we study the alignment of the perturbation set with the shortest path to the (robust Bayes optimal) decision boundary, which we call the *signal direction*.




(a) Concentric spheres

(b) CIFAR-10

Figure 2: Ablations for the signal aligned threat model $\mathcal{B}_{\epsilon,k}$ defined in Equation (2): (a) We report mean and standard error over 5 runs for standard and robust error on the concentric spheres dataset ($n = 500, d = 10, \gamma = 20$, $\epsilon = 5.0$). (b) We report mean and standard error over 3 runs for standard and robust error on the CIFAR-10 dataset ($\epsilon = 2.5$).

More formally, we define the signal direction $s(x, y)$ for a data point $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and the robust Bayes optimal classifier $f^* : \mathcal{X} \to \mathcal{Y}$, as the direction along the shortest path to the decision boundary

$$s(x, y) := \operatorname*{argmin}_{\delta : \|\delta\|_2 = 1} \left\{ \min_{\epsilon \geq 0} \epsilon \ : \ f^*(x + \epsilon \cdot \delta) \neq y \right\}.$$

Then, given a data distribution $\mathcal{D}$ and a threat model $\mathcal{B}_\epsilon$, we define the alignment of the perturbation set as

$$\text{Alignment}(\mathcal{B}_\epsilon) := \mathbb{E}_{(x,y)\sim\mathcal{D}} \sup_{\delta \in \mathcal{B}_\epsilon} \left[ \frac{s(x, y)^\top \delta}{\|s(x, y)\|_2 \|\delta\|_2} \right].$$

**Perturbation sets with different degrees of alignment** Throughout this section, we intervene on the alignment of the perturbation set by restricting the threat model to $k$ random orthogonal directions. More formally, we randomly sample $k$ orthogonal unit vectors $\delta_j$ and define the nested threat model as

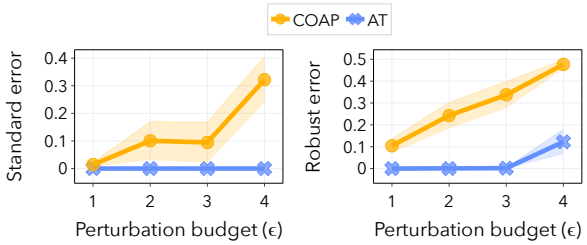
$$\mathcal{B}_{\epsilon,k} = \cup_{j=1}^{k} \{\delta_j \beta \mid |\beta| \leq \epsilon\}, \tag{2}$$

with $\mathcal{B}_{\epsilon,1} \subset \mathcal{B}_{\epsilon,2} \ldots$. It is important to note that as $k$ increases, there are more choices for the direction of the adversarial perturbation and hence Alignment$(\mathcal{B}_{\epsilon,k})$ increases with $k$ even though the perturbation budget $\epsilon$ stays constant. Furthermore, unlike $\ell_p$-ball perturbations, we can evaluate the empirical robust error exactly for this threat model with a line-search along the $k$ random orthogonal directions.

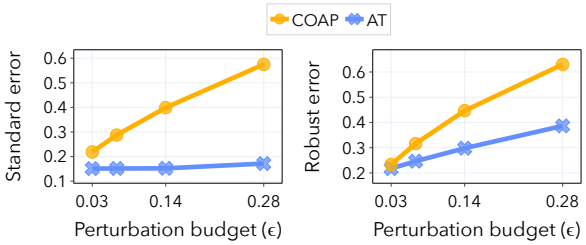
**Ablations** We now present a series of ablations on the number of components $k$ to show that stronger alignment indeed correlates with a wider standard and robust error gap between COAP and AT. In particular, in Figure 2a we observe for the concentric spheres distribution that stronger alignment yields a gap of up to 25% for standard error and 45% for robust error. In Figure 2b, we observe for CIFAR-10 a standard error gap of up to 15% and robust error gap of up to 5%, for the largest number of components.

## 3.3 Factor (ii): Perturbation budget

The second factor we investigate is the perturbation budget $\epsilon$ of the threat model. This quantity is directly related to the tightness of convex relaxations. In particular, Wong and Kolter [38] show that the COAP relaxation is very tight for small $\epsilon$, while it becomes loose for larger values of $\epsilon$. Although it is well-known




(a) Concentric spheres




(b) CIFAR-10

Figure 3: Ablations for the $\ell_2$-ball threat model: (a) We report mean and standard error over 5 runs for standard and robust error on the concentric spheres dataset ($n = 500, d = 10, \gamma = 20$). (b) Standard and robust error for COAP and AT on the CIFAR-10 dataset.

that a larger perturbation budget results in worse certified training performance, the same also holds for adversarial training and it is unknown whether the error gap between the two increases.

**Ablations** In Figures 3a and 3b we show that indeed, a larger perturbation budget correlates with a wider gap for both standard and robust error. We observe that for both concentric spheres and CIFAR-10 the error gap monotonically increases as a function of the perturbation budget. Additionally, it is worth noting that this increase is not a consequence of a limited sample size. COAP training can indeed fail when the perturbation budget is significant, even given a large sample size. This is evident in Figure 3a for the case of concentric spheres, where the sample size is $n = 500$ and the data dimensionality is $d = 10$.

### 3.4 Factor (iii): Margin of the data distribution

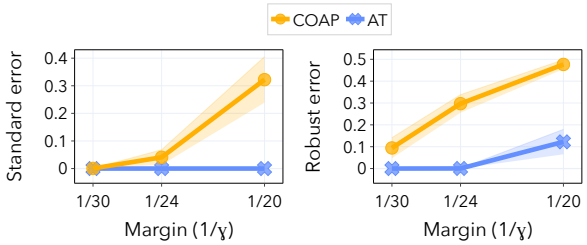
The third factor we investigate is the implicit margin of the data distribution, i.e. the minimum distance in feature space between any two data points of different classes.

More formally, we define the margin $\gamma$ for a dataset $D = \{(x_i, y_i)\}_{i=1}^n$ as

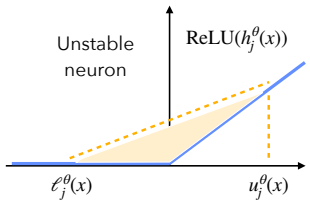
$$\gamma := \min_{i,j} \|x_i - x_j\|_2 \quad \text{s.t.} \quad i \neq j \ , \ y_i \neq y_j.$$

In any robust classification task, this factor plays a pivotal role. When the implicit margin is small, the task becomes inherently more challenging, leading to potential failures in both adversarial and certified training. However, given that certified training essentially relies on an over-approximation of the perturbation set, it can be posited that it is particularly susceptible to failure for small margins. Below, we present as evidence an ablation study on the concentric spheres dataset.

**Ablations** We empirically show that a smaller margin correlates with a wider standard and robust error gap between AT and COAP. As intervening on the implicit margin for image datasets such as CIFAR-10 is not feasible, we focus our ablations on the concentric spheres dataset instead. In particular, in Figure 4a, we observe that the standard and robust error gap steadily increases as a function of the inverse margin, reaching up to 30% and 35%, respectively.




(a) Concentric spheres




(b)

Figure 4: (a) Ablations for the $\ell_2$-ball threat model: We report mean and standard error over 5 runs for standard and robust error on the concentric spheres dataset ($n = 500, d = 10$, $\epsilon = 4.0$). (b) Conceptual illustration of the COAP convex relaxation for neurons in the unstable state.

# 4 The role of unstable neurons

In this section, we attempt to explain how the previously identified factors influence the standard and robust error gap. Specifically, we discuss how they affect the error gap via their impact on the number of *unstable neurons*, a quantity which plays a crucial role in the performance of certified training with convex relaxations.

**Definition of unstable neurons** For a neural network $f_\theta$ with $m$ total number of neurons, we use $h_j^\theta(x)$ to denote the pre-activation value for an input $x$ and a neuron $j$. Similarly, we define lower and upper bounds for this value under perturbations $\left[\ell_j^\theta(x), u_j^\theta(x)\right]$, where $\ell_j^\theta(x) \leq h_j^\theta(x+\delta) \leq u_j^\theta(x)$ for all allowed perturbations $\delta \in \mathcal{B}_\epsilon$. Formally, we define *inactive neurons* for an input $x$, as all neurons $j$ with non-positive pre-activation upper bounds $u_j^\theta(x) \leq 0$, i.e. they are always inactive regardless of input perturbations. Similarly, *active neurons* have non-negative pre-activation lower bounds $\ell_j^\theta(x) \geq 0$, i.e. they are always active. In contrast, *unstable* neurons have uncertain activation states given different input perturbations, i.e. $\ell_j^\theta(x) \leq 0 \leq u_j^\theta(x)$. Given a neural network $f_\theta$ and a dataset $D = \{(x_i, y_i)\}_{i=1}^n$, we define the number of unstable neurons as

$$\text{UnstNeur}(f_\theta) = \frac{1}{mn} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbb{I}\{\ell_j^\theta(x_i) \leq 0 \leq u_j^\theta(x_i)\}. \tag{3}$$

**Unstable neurons and convex relaxations** A key property of certified training with convex relaxations is the tightness of the over-approximation compared to the original perturbation set. Since convex relaxations are much looser for unstable neurons compared to active or inactive neurons (exemplarily illustrated for the COAP convex relaxation [39] in Figure 4b), the number of unstable neurons is a good indicator for quantifying the looseness of the over-approximation of the perturbation set.

Further, existing works [17, 24, 32] suggest that the number of unstable neurons directly affects the performance of certified training. Intuitively, the looser the over-approximation during training, the greater the susceptibility to noise becomes. In particular, noise can be introduced during training when the over-approximated perturbation set extends over the (robust Bayes optimal) decision boundary, essentially causing over-regularization. Hence, it is natural to expect that an increased number of unstable neurons during training might cause a larger error gap between certified and adversarial training. However, it is impossible to verify this hypothesis by only intervening on the number of unstable neurons in a neural network while keeping all other factors fixed.

For the above reasons, we study how the factors identified in Section 3 affect the total number of unstable neurons during training, i.e. $\sum_{t=1}^{T} \text{UnstNeur}(f_{\theta^t})$ where $f_{\theta^t}$ is the neural network at epoch $t$ and $T$ is the total

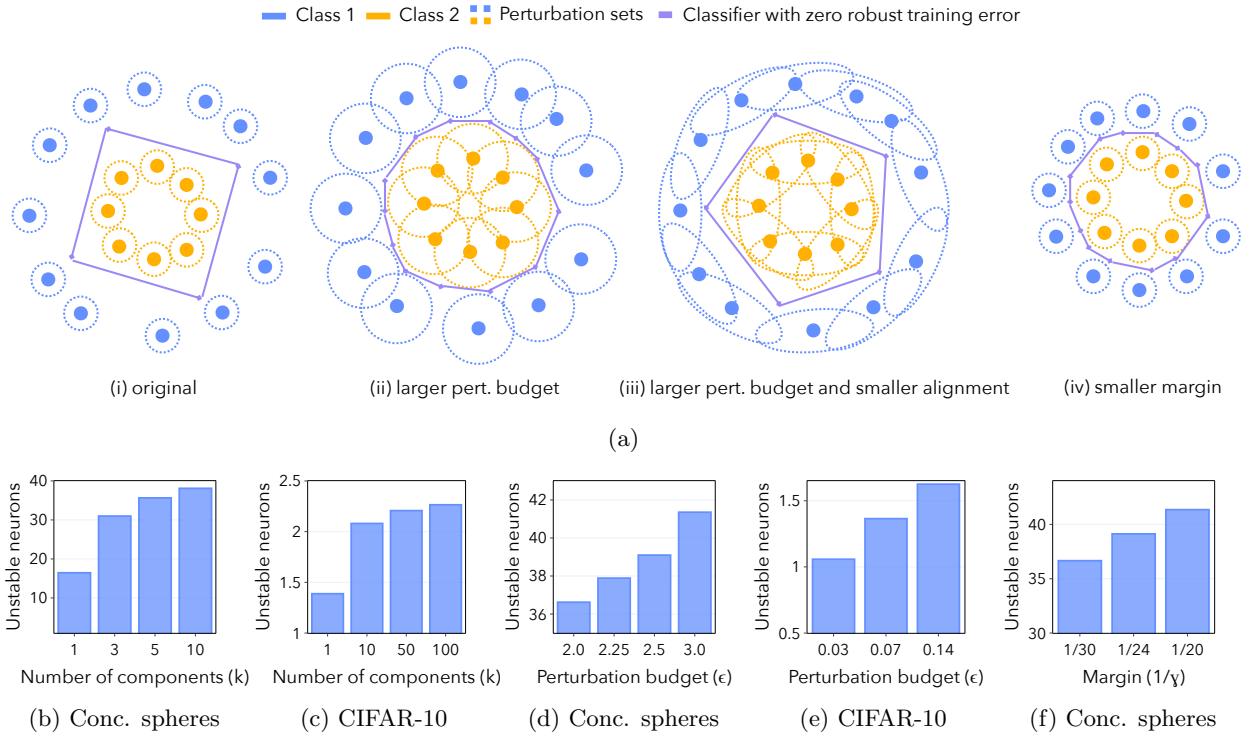



Figure 5: (a) Conceptual illustration of how the three factors act on the number of unstable neurons. (b – f) Ablations for the signal aligned and the $\ell_2$-ball threat models: (b,d,f) Concentric spheres dataset ($\gamma = 20, n = 500, d = 10, \epsilon_2 = 5.0, \epsilon_{\text{signal}} = 3.0$). (c,e) CIFAR-10 dataset ($\epsilon_{\text{signal}} = 2.5$).

number of training epochs. In particular, we first provide empirical evidence that the three factors correlate with a higher number of unstable neurons. Then, we provide an intuitive explanation with illustrations on a simple realisation of the concentric spheres distribution.

**How the factors affect unstable neurons** In Figures 5b to 5f, we can observe how for both the CIFAR-10 and the concentric spheres datasets, the number of unstable neurons increases with larger alignment, larger perturbation budget and smaller margin.

We interpret these findings as follows: both smaller margin and larger perturbation budget increase the model complexity of the decision boundary with good robust training accuracy (similar to an argument made by Nakkiran [26]). In particular, they require a larger number of piecewise linear regions, as also argued in Shah et al. [31]. This intuition is illustrated in Figure 5a: for a fixed perturbation set, both smaller margin (iv) and larger perturbation budget (ii) require a larger number of piecewise linear functions to approximate the decision boundary. The increased number of linear regions, together with training points close to the boundary, then results in an increase in the number of unstable neurons. Further, for less aligned perturbation sets with the same margin and perturbation budget, i.e. contrasting (iii) and (ii), the learning task is simplified, and fewer piecewise regions are required. As a result, fewer neurons become unstable.

# 5 Related work

**Limitations of certified training with convex relaxations** Certified training with convex relaxations hinges on over-approximating the potential output range of each neuron for the perturbed versions of any input point. While this over-approximation allows for tractable computation of an upper bound on the robust error, it also introduces an inherent looseness that impacts both robust training and evaluation. For example, Salman et al. [30] investigate the tightness of convex relaxations for verification purposes, i.e. for certifying the robustness of already trained models, and show that even the best existing convex relaxation provides only very loose upper bounds on the robust error.

The effect of the tightness of convex relaxations on certified training has recently been studied in the context of the so-called paradox of certified training [10], i.e. when training with tighter relaxations leads to worse certified robustness. Lee et al. [17] show empirically that tighter convex relaxations affect the smoothness of the loss function, which in turn impacts the performance of certified training. Instead of comparing convex relaxations with different tightness, we discuss how tightness affects the performance of certified training *in the context* of varying other factors related to the threat model and the data distribution.

**Limitations of randomised smoothing** As an alternative to convex relaxations, randomised smoothing gives robustness guarantees with a certain probability [4, 18, 19]. Despite its popularity, randomised smoothing also suffers from several limitations beyond an increased computational cost. For example, several works have exposed an accuracy-robustness tradeoff [2, 14]. Further, randomised smoothing can significantly hurt the disparity in class-wise accuracy [23] and is extremely vulnerable to low-frequency corruptions of the test data [34]. In general, compared to certified training with convex relaxations, the drawbacks of randomised smoothing are much better understood, and efforts are being made towards developing new defences to bridge the gap with adversarial training [27]. Therefore, we focus our attention in this paper on certified training with convex relaxations.

**Limitations of adversarial training** Not only certified training comes at the price of reduced test performance - the same has been reported for adversarial training. For example, it has been well-studied that adversarial training often results in a trade-off between robust and standard accuracy; that is, the standard accuracy of adversarially trained models often decreases even though the robust accuracy increases (see, e.g. [29, 36, 42]). Attempts to provide an explanation for consistent perturbations, i.e. perturbations that do not change the true label, have so far focused on the small sample size regime. For example, Raghunathan et al. [29] prove that for small sample sizes, adversarial training can increase standard error even in the absence of noise. Further, most related to our work, Clarysse et al. [3] recently prove that adversarial training with perturbations aligned with the signal direction can even increase the robust error. In contrast to this line of work, our experiments focus on the large-sample regime and compare certified with adversarial training.

# 6 Conclusion

In this paper, we show that certified training with convex relaxations suffers from worse standard error and robust error than adversarial training. Further, we are the first to provide a systematic comparison of these two robust training paradigms across multiple datasets and threat models. In doing so, we explore three important factors which are correlated with a wider standard and robust error gap between certified and adversarial training. We believe that shedding light on this error gap 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.

## Acknowledgements

PDB was supported by the Hasler Foundation grant number 21050 and AS acknowledges partial support from the ETH AI Center postdoctoral fellowship.

## References

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

[2] A. Blum, T. Dick, N. Manoj, and H. Zhang. Random Smoothing Might be Unable to Certify $L_\infty$-Robustness for High-Dimensional Images. *Journal of Machine Learning Research*, 2020.

[3] J. Clarysse, J. Hörrmann, and F. Yang. Why adversarial training can hurt robust accuracy. In *Proceedings of the International Conference on Learning Representations*, 2023.

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

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

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

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

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

[10] N. Jovanović, M. Balunovic, M. Baader, and M. Vechev. On the paradox of certified training. *Transactions on Machine Learning Research*, 2022.

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

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

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

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

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

[16] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[17] S. Lee, W. Lee, J. Park, and J. Lee. Towards better understanding of training certifiably robust models against adversarial examples. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

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

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

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

[21] Y. Mao, M. N. Müller, M. Fischer, and M. Vechev. Taps: Connecting certified and adversarial training, 2023. arXiv: 2305.04574.

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

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

[24] M. N. Müller, F. Eckert, M. Fischer, and M. Vechev. Certified training: Small boxes are all you need. In *Proceedings of the International Conference on Learning Representations*, 2023.

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

[26] P. Nakkiran. Adversarial robustness may be at odds with simplicity. *arXiv preprint arXiv:1901.00532*, 2019.

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

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

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

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

[31] H. Shah, K. Tamuly, A. Raghunathan, P. Jain, and P. Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

[32] Z. Shi, Y. Wang, H. Zhang, J. Yi, and C.-J. Hsieh. Fast certified robust training with short warmup. In *Advances in Neural Information Processing Systems*, 2021.

[33] G. Singh, T. Gehr, M. Mirman, M. Püschel, and M. Vechev. Fast and Effective Robustness Certification. In *Advances in Neural Information Processing Systems*, 2018.

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

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

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

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

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

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

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

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

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

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

# A Experimental Details

## A.1 Image experiments with $\ell_2$-ball and $\ell_\infty$-ball threat models

Below we provide complete experimental details to reproduce: Tables 1, 4 and 5 and figs. 3b and 5e.

**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. [39]. We report the architectural details in Table 2. For CIFAR-10, we train the residual network (ResNet) with the same structure used in Wong et al. [39]; we use 1 residual block with 16, 16, 32, and 64 filters. For Tiny ImageNet, we train a WideResNet as in Xu et al. [40], using three wide basic blocks with a wide 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 2: MNIST model architecture. All layers are followed by 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 $n$" is a fully connected layer with $n$ outputs.

**Dataset preprocessing** For MNIST, we use full $28 \times 28$ 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 $56 \times 56$ and random flips during training. During testing, we use a central $56 \times 56$ crop. We also normalise images according to per-channel statistics.

**Robust evaluation** We consider $\ell_2$-ball perturbations. We evaluate the robust error using the most expensive version of AutoAttack (AA+) [5]. 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 [12] 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 of 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 of 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 [5].

**COAP training details** We follow the settings proposed by the authors and report them here. For MNIST, we use the Adam optimiser [12] with a learning rate of 0.001 and a batch size of 50. We schedule $\epsilon$ 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 $\epsilon$ 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. [39].

**CROWN-IBP and 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 [12] and set the learning rate to $5 \times 10^{-4}$. We warm up with 10 epochs of regular training and gradually ramp up $\epsilon_{\text{train}}$ from 0 to $\epsilon$ in 50 epochs. We reduce the learning rate by a factor of 0.1 at epochs 130 and 190. For CIFAR-10, we train 2000 epochs with a batch size of 256 and a learning rate of $5 \times 10^{-4}$. We warm up for 100 epochs and ramp-up $\epsilon$ for 800 epochs. The learning rate is reduced by a factor of 0.1 at epochs 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 $\epsilon_{\text{train}}$ with a schedule length of 400. For all datasets, a hyper-parameter $\beta$ 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 $\epsilon$. For all experiments, we use the implementation provided in the auto LiRPA library [40].

**FAST-IBP training details** We follow the settings proposed by the authors and report them here. Further, we modify the architecture to add batch normalisation at each layer, as suggested by the authors. Models are trained with Adam optimiser [12] with an initial learning rate of $5 \times 10^{-4}$, and there are two milestones where the learning rate decays by 0.2. We determine the milestones for learning rate decay according to the training schedule and the total number of epochs, as shown in Table 3. The gradient clipping threshold is set to 10.0. We train the models using a batch size of 256 on MNIST and 128 on CIFAR-10 and TinyImageNet. The tolerance value $\tau$ in our warmup regularization is fixed to 0.5.

During the warmup stage, after training with $\epsilon = 0$ for a number of epochs, the perturbation radius $\epsilon$ is gradually increased from 0 until the target perturbation radius $\epsilon_{\text{target}}$ , during the $0 < \epsilon < \epsilon_{\text{target}}$ phase. Specifically, during the first 25% epochs of the $\epsilon$ increasing stage, $\epsilon$ is increased exponentially, and after that $\epsilon$ is increased linearly. In this way, $\epsilon$ remains relatively small and increases relatively slowly during the beginning to stabilize training. We use the SmooothedScheduler in the autoLiRPA library as the scheduler for $\epsilon$, similarly adopted by Xu et al. [40].

| Dataset | Total epochs | Decay-1 | Decay-2 |
|---|---|---|---|
| MNIST | 70 | 50 | 60 |
| CIFAR-10 | 160 | 120 | 140 |
| TinyImageNet | 80 | 60 | 70 |

Table 3: Milestones for learning rate decay when the different total number of epochs are used. "Decay-1" and "Decay-2" denote the two milestones, respectively, when the learning rate decays by a factor of 0.2.

## A.2 Image experiments with signal aligned threat model

Below we provide complete experimental details to reproduce: Figures 2b, 5c and 8a. First, we present our extension of COAP to the threat model introduced in Equation (2). Rather than deriving the dual problem as in Wong and Kolter [38], we consider the conjugate function view introduced in Wong et al. [39]. 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-aligned threat model:

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

For all experiments, we use the convolutional neural network architecture described in Table 2. Note that it is not possible to scale to ResNet with the threat model in Equation (2), as the random projections trick derived in Wong et al. [39] is tailored to $\ell_\infty$-ball and $\ell_2$-ball threat models.

**AT training details** For both MNIST and CIFAR-10, we train 20 epochs using Adam optimiser [12] 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 [12] with a learning rate of 0.001 and a batch size of 64. We schedule $\epsilon$ 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. [39].

### A.3 Synthetic experiments with signal-aligned, $\ell_2$-ball and $\ell_\infty$-ball threat models

Below we provide complete experimental details to reproduce Figures 2a, 3a, 4a, 5b, 5d and 5f.

**Data generation** For the spheres dataset, we generate a random $x \in \mathbb{R}^d$ where $\|x\|_2$ is either $R_0$ or $R_1$ , with equal probability assigned to each norm. We associate with each $x$ a label $y$ such that $y = -1$ if $\|x\|_2 = R_0$ 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 $x = \frac{z}{\|z\|_2} R_0$ or $x = \frac{z}{\|z\|_2} R_1$. For all experiments with the concentric spheres distribution, we set $d = 10, n = 500, n_{\text{test}} = 10^4$.

**Model and hyper-parameters** For all the experiments, we use an MLP architecture with $W = 100$ neurons and one hidden layer and 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.

**Robust evaluation** We evaluate robust error at test-time using Auto-PGD [5] with 100 iterations and 5 random restarts. We use both the cross-entropy and the difference of logit loss to prevent gradient masking. We use the implementation provided in AutoAttack [5] 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) [8, 20], we train the network to minimise the robust cross-entropy loss. At each epoch, we search for adversarial examples using Auto-PGD [5] 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) [38, 39], we train the network to minimise the upper bound on the robust error. Our implementation is based on the code released by the authors.

# B Experimental comparison of certified and adversarial training

This section provides a more complete evaluation of certified and adversarial training, including a wide range of perturbation budgets.

**MNIST** Similar to Section 2, we observe that for the $\ell_\infty$-ball threat model, FAST-IBP achieves the best robust error across all perturbation budgets, while AT delivers the best standard error. On the other hand, for the $\ell_2$-ball threat model, AT achieves both the best robust and standard error across all perturbation budgets.

| Dataset | Perturbation budget | Method | Robust error | Standard error |
|---|---|---|---|---|
| MNIST | $\epsilon_\infty = 0.1$ | AT | 0.021 | **0.008** |
| | | COAP | 0.026 | 0.010 |
| | | CROWN-IBP | 0.024 | 0.010 |
| | | IBP | 0.026 | 0.010 |
| | | FAST-IBP | **0.017** | 0.009 |
| MNIST | $\epsilon_\infty = 0.2$ | AT | 0.045 | **0.008** |
| | | COAP | 0.066 | 0.026 |
| | | CROWN-IBP | 0.050 | 0.014 |
| | | IBP | 0.060 | 0.014 |
| | | FAST-IBP | **0.032** | 0.010 |
| MNIST | $\epsilon_\infty = 0.3$ | AT | 0.095 | **0.008** |
| | | COAP | 0.224 | 0.110 |
| | | CROWN-IBP | 0.075 | 0.017 |
| | | IBP | 0.091 | 0.021 |
| | | FAST-IBP | **0.058** | 0.015 |
| MNIST | $\epsilon_\infty = 0.4$ | AT | 0.321 | **0.008** |
| | | COAP | 0.806 | 0.737 |
| | | CROWN-IBP | 0.120 | 0.025 |
| | | IBP | 0.147 | 0.033 |
| | | FAST-IBP | **0.104** | 0.024 |
| MNIST | $\epsilon_2 = 0.5$ | AT | **0.020** | **0.006** |
| | | COAP | 0.040 | 0.014 |
| | | CROWN-IBP | 0.059 | 0.027 |
| | | IBP | 0.068 | 0.032 |
| | | FAST-IBP | 0.044 | 0.022 |
| MNIST | $\epsilon_2 = 0.75$ | AT | **0.035** | **0.006** |
| | | COAP | 0.100 | 0.030 |
| | | CROWN-IBP | 0.125 | 0.060 |
| | | IBP | 0.145 | 0.070 |
| | | FAST-IBP | 0.124 | 0.053 |
| MNIST | $\epsilon_2 = 1.0$ | AT | **0.048** | **0.006** |
| | | COAP | 0.193 | 0.049 |
| | | CROWN-IBP | 0.355 | 0.192 |
| | | IBP | 0.539 | 0.432 |
| | | FAST-IBP | 0.361 | 0.175 |
| MNIST | $\epsilon_2 = 1.25$ | AT | **0.092** | **0.005** |
| | | COAP | 0.302 | 0.074 |
| | | CROWN-IBP | 0.760 | 0.696 |
| | | IBP | 0.806 | 0.794 |
| | | FAST-IBP | 0.724 | 0.652 |

Table 4: Results on MNIST for both $\ell_2$-ball and $\ell_\infty$-ball threat models.

**CIFAR-10** We observe that for both $\ell_\infty$-ball and $\ell_2$-ball threat models, AT consistently achieves the best standard and robust error, with the only exception being $\epsilon = 8/255$, where FAST-IBP slightly outperforms AT in terms of robust error.

| Dataset | Perturbation budget | Method | Robust error | Standard error |
|---|---|---|---|---|
| CIFAR-10 | $\epsilon_\infty = 1/255$ | AT | **0.296** | **0.150** |
| | | COAP | 0.372 | 0.309 |
| | | CROWN-IBP | 0.425 | 0.331 |
| | | IBP | 0.459 | 0.356 |
| | | FAST-IBP | 0.384 | 0.290 |
| CIFAR-10 | $\epsilon_\infty = 2/255$ | AT | **0.378** | **0.166** |
| | | COAP | 0.427 | 0.336 |
| | | CROWN-IBP | 0.498 | 0.363 |
| | | IBP | 0.526 | 0.415 |
| | | FAST-IBP | 0.476 | 0.357 |
| CIFAR-10 | $\epsilon_\infty = 4/255$ | AT | **0.506** | **0.193** |
| | | COAP | 0.609 | 0.502 |
| | | CROWN-IBP | 0.640 | 0.548 |
| | | IBP | 0.614 | 0.487 |
| | | FAST-IBP | 0.585 | 0.449 |
| CIFAR-10 | $\epsilon_\infty = 8/255$ | AT | 0.698 | **0.239** |
| | | COAP | 0.775 | 0.729 |
| | | CROWN-IBP | 0.673 | 0.539 |
| | | IBP | 0.708 | 0.606 |
| | | FAST-IBP | **0.675** | 0.546 |
| CIFAR-10 | $\epsilon_2 = 9/255$ | AT | **0.217** | **0.151** |
| | | COAP | 0.233 | 0.219 |
| | | CROWN-IBP | 0.521 | 0.491 |
| | | IBP | 0.457 | 0.424 |
| | | FAST-IBP | 0.661 | 0.647 |
| CIFAR-10 | $\epsilon_2 = 18/255$ | AT | **0.246** | **0.151** |
| | | COAP | 0.316 | 0.287 |
| | | CROWN-IBP | 0.721 | 0.690 |
| | | IBP | 0.566 | 0.515 |
| | | FAST-IBP | 0.763 | 0.753 |
| CIFAR-10 | $\epsilon_2 = 36/255$ | AT | **0.298** | **0.152** |
| | | COAP | 0.447 | 0.399 |
| | | CROWN-IBP | 0.780 | 0.740 |
| | | IBP | 0.680 | 0.622 |
| | | FAST-IBP | 0.849 | 0.844 |
| CIFAR-10 | $\epsilon_2 = 72/255$ | AT | **0.386** | **0.172** |
| | | COAP | 0.630 | 0.574 |
| | | CROWN-IBP | 0.900 | 0.900 |
| | | IBP | 0.772 | 0.740 |
| | | FAST-IBP | 0.900 | 0.900 |

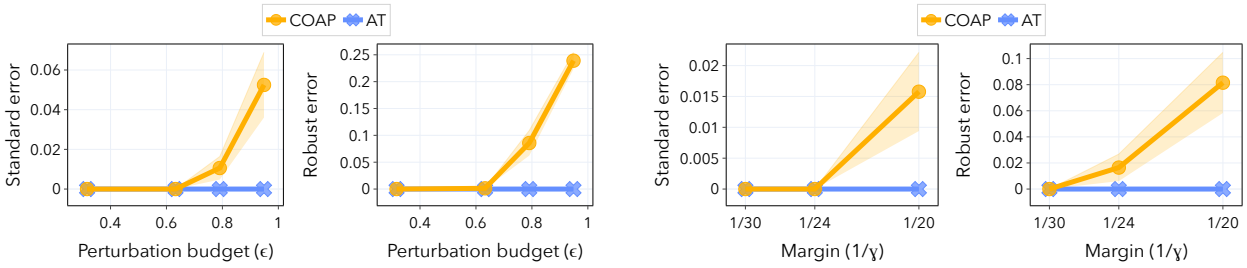
Table 5: Results on CIFAR-10 for both $\ell_2$-ball and $\ell_\infty$-ball threat models

# C Additional factor ablations

For the ablations on the perturbation budget and the margin, we focused on $\ell_2$-ball perturbations in Sections 3.3, 3.4 and 4, as the phenomena we aim to investigate are more prominent in this context. We present similar results for the $\ell_\infty$-ball threat model. Additionally, while our alignment ablations in Section 3.2 focused on CIFAR-10 for clarity, we will now illustrate similar trends for MNIST.

## C.1 $\ell_\infty$-ball threat model on concentric spheres

First, we observe in Figures 6a and 6b that for the concentric spheres distribution, the standard and robust error gap increases with the perturbation budget and the inverse margin.




(a) (b)

Figure 6: Results for $\ell_\infty$-ball threat model on the concentric spheres dataset. (a) Standard and robust error for COAP and AT as the perturbation budget $\epsilon$ of the threat model increases ($\gamma = 20, n = 500, d = 10$). (b) Standard and robust error for COAP and AT as the inverse margin increases ($\epsilon \approx 1, n = 500, d = 10$).

Further, we verify that unstable neurons, i.e. neurons with uncertain activation states given different input perturbations, are similarly affected by the factors as for the $\ell_2$-ball threat model. In particular, Figures 7a and 7b shows that unstable neurons increase steadily with the perturbation budget and the inverse margin.

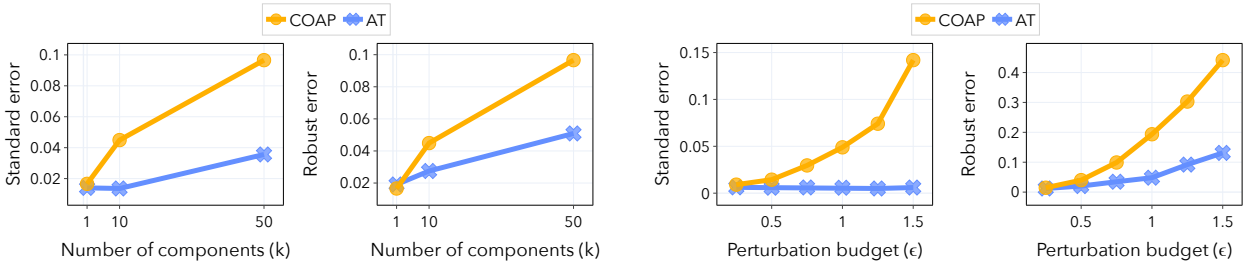

(a) (b)

Figure 7: Ablations on perturbation budget and margin for the $\ell_\infty$-ball threat model. (a) Results on concentric spheres dataset ($\gamma = 20, n = 500, d = 10$). (b) Results on concentric spheres dataset ($\epsilon \approx 1, n = 500, d = 10$).

## C.2 $\ell_2$-ball and signal aligned threat models on MNIST

In addition to the ablation study conducted with CIFAR-10, presented in Section 3.2, we conduct similar ablations using MNIST.

The results presented here align well with the trends identified on CIFAR-10, reinforcing our findings. In particular, we observe in Figures 8a and 8b that the standard and robust error gap increases with the perturbation budget and the alignment.




(a) (b)

Figure 8: Results for $\ell_2$-ball and signal aligned threat models on MNIST. (a) Standard and robust error for COAP and AT ($\epsilon = 20$) as the number of components $k$ of the threat model increases. (b) Standard and robust error for COAP and AT as the perturbation budget $\epsilon$ increases.

Further, we verify that unstable neurons are similarly affected by the factors as for CIFAR-10. In particular, Figures 7a and 7b shows that unstable neurons increase steadily with the alignment and the perturbation budget.


(a) (b)

Figure 9: Ablations for alignment and perturbation budget on MNIST. (a) Results for the signal aligned threat model ($\epsilon = 20$). (b) Results for the $\ell_2$-ball threat model.