

Robust Out-of-Distribution Detection in Deep Classifiers

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
M. Sc. Alexander Meinke
aus Anklam

Tübingen
2023

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der
Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:

24.04.2023

Dekan:

Prof. Dr. Thilo Stehle

1. Berichterstatter:

Prof. Dr. Matthias Hein

2. Berichterstatter:

Prof. Dr. Philipp Hennig

Abstract

Over the past decade, deep learning has gone from a fringe discipline of computer science to a major driver of innovation across a large number of industries. The deployment of such rapidly developing technology in safety-critical applications necessitates the careful study and mitigation of potential failure modes. Indeed, many deep learning models are overconfident in their predictions, are unable to flag out-of-distribution examples that are clearly unrelated to the task they were trained on and are vulnerable to adversarial vulnerabilities, where a small change in the input leads to a large change in the model's prediction. In this dissertation, we study the relation between these issues in deep learning based vision classifiers.

First, we benchmark various methods that have been proposed to enable deep learning methods to detect out-of-distribution examples and we show that a classifier's predictive confidence is well-suited for this task, if the classifier has had access to a large and diverse out-distribution at train time. We theoretically investigate how different out-of-distribution detection methods are related and show that several seemingly different approaches are actually modeling the same core quantities.

In the second part we study the adversarial robustness of a classifier's confidence on out-of-distribution data. Concretely, we show that several previous techniques for adversarial robustness can be combined to create a model that inherits each method's strength while significantly reducing their respective drawbacks. In addition, we demonstrate that the enforcement of adversarially robust low confidence on out-of-distribution data enhances the inherent interpretability of the model by imbuing the classifier with certain generative properties that can be used to query the model for counterfactual explanations for its decisions.

In the third part of this dissertation we will study the problem of issuing mathematically provable certificates for the adversarial robustness of a model's confidence on out-of-distribution data. We develop two different approaches to this problem and show that they have complementary strength and weaknesses. The first method is easy to train, puts no restrictions on the architecture that our classifier can use and provably ensures that the classifier will have low confidence on data very far away. However, it only provides guarantees for very specific types of adversarial perturbations and only for data that is very easy to distinguish from the in-distribution. The second approach works for more commonly studied sets of adversarial perturbations and on much more challenging out-distribution data, but puts heavy restrictions on the architecture that can be used and thus the achievable accuracy. It also does not guarantee low confidence on asymptotically far away data. In the final chapter of this dissertation we show how ideas from both of these techniques can be combined in a way that preserves all of their strengths while inheriting none of their weaknesses. Thus, this thesis outlines how to develop high-performing classifiers that provably know when they do not know.

Kurzfassung

In den letzten zehn Jahren hat sich das Deep Learning von einer Randdisziplin der Informatik zu einer treibenden Kraft der Innovation in vielen Industrien entwickelt. Die Einführung solcher schnell entwickelnder Technologien in sicherheitskritischen Anwendungen erfordert eine sorgfältige Untersuchung und Beseitigung möglicher Ausfallmodi. Tatsächlich sind viele Deep-Learning-Modelle zu konfident in ihren Vorhersagen, sind nicht in der Lage, Beispiele außerhalb der Verteilung zu markieren, die offensichtlich nichts mit der Aufgabe zu tun haben, für die sie trainiert wurden, und besitzen Adversarial-Schwachstellen, bei denen eine kleine Änderung der Eingabe zu einer großen Änderung der Vorhersage des Modells führt. In dieser Dissertation untersuchen wir die Beziehung zwischen diesen Problemen in Deep-Learning Bilderkennungsklassifikatoren.

Zunächst vergleichen wir verschiedene Methoden, die vorgeschlagen wurden, um Deep-Learning-Methoden dazu zu befähigen, Beispiele außerhalb der Verteilung zu erkennen, und zeigen, dass die Vorhersagekonfidenz eines Klassifikators gut geeignet ist, wenn der Klassifikator während des Trainings auf eine große und vielfältige Ausverteilung zugreifen konnte. Wir untersuchen theoriegestützt, wie sich verschiedene Methoden zur Erkennung von Ausverteilungen voneinander unterscheiden und zeigen, dass mehrere scheinbar unterschiedliche Ansätze tatsächlich dieselben Funktionen modellieren.

In dem zweiten Teil untersuchen wir die Angriffsrobustheit der Konfidenz eines Klassifikators bei Ausverteilungsdaten. Konkret zeigen wir, dass mehrere vorherige Techniken zur Angriffsrobustheit kombiniert werden können, um ein Modell zu erstellen, das die Stärken jeder Methode erbt, während deren Nachteile signifikant verringert werden. Zusätzlich zeigen wir, dass die Durchsetzung von angriffsrobusten niedrigen Konfidenzen auf Ausverteilungsdaten die inhärente Interpretierbarkeit des Modells verbessert, indem dem Klassifikator bestimmte generative Eigenschaften verliehen werden, die zur Abfrage des Modells für kontrafaktische Erklärungen seiner Entscheidungen verwendet werden können.

Im dritten Teil dieser Dissertation untersuchen wir das Problem der Ausstellung von mathematisch beweisbaren Zertifikaten für die Angriffsrobustheit der Konfidenz eines Modells auf Ausverteilungsdaten. Wir entwickeln zwei verschiedene Ansätze für dieses Problem und zeigen, dass sie ergänzende Stärken und Schwächen haben. Die erste Methode ist leicht zu trainieren, stellt keine Einschränkungen für die Architektur des Klassifikators dar und stellt beweisbar sicher, dass der Klassifikator eine geringe Konfidenz bei Daten aufweisen wird, die sehr weit entfernt sind. Sie bietet jedoch nur Garantien für sehr spezifische Arten von adversarialen Veränderungen und nur für Daten, die sehr leicht von der In-Verteilung zu unterscheiden sind. Der zweite Ansatz funktioniert für häufiger untersuchte adversariale Veränderungen und für viel herausforderndere Ausverteilungsdaten, setzt jedoch starke Einschränkungen für die

Architektur, die verwendet werden kann, voraus und damit für die erreichbare Genauigkeit. Sie gewährleistet auch keine niedrige Konfidenz auf asymptotisch weit entfernten Daten. Im letzten Kapitel dieser Dissertation zeigen wir, wie Ideen aus beiden dieser Techniken auf eine Weise kombiniert werden können, die alle ihre Stärken bewahrt, während sie keine ihrer Schwächen übernimmt. Auf diese Weise skizziert diese Arbeit, wie leistungsfähige Klassifikatoren entwickelt werden können, die beweisbar wissen, wann sie etwas nicht wissen.

Acknowledgments

This thesis is a major milestone in my life and in this short section I could not possibly cast a net wide enough to capture my gratitude to all the people who have contributed to me reaching it. Of course, professionally the greatest influence has come from my excellent advisor Matthias Hein, who was always patient, kind and supportive, while asking the difficult questions I needed to hear and pushing me to do the best work I could. The endeavor could also not have been possible without my other co-authors Julian Bitterwolf and Maximilian Augustin, whose work was crucial for several of the papers that went into this dissertation. I also want to thank Julian Bitterwolf for all the scientific and non-scientific discussions that we had that did not lead to scientific publications. I am also grateful to Valentyn Boreiko who always offered support, both as a colleague and as a friend. Many thanks should also go to Philipp Hennig and Andreas Geiger who gave both their valuable time and advice when attending my TAC-meetings. I would also like to extend my sincere thanks to my other collaborators Gabriele Carovano and Manuel Arias Chao, both of whom were extremely patient and kind while helping me explore areas of research beyond my comfort zone. I would be remiss in not mentioning Maksym Andriushchenko, who greatly helped me with his technical expertise when I first joined the group.

I also want to sincerely thank the International Max Planck Research School for Intelligent Systems for their support during my PhD thesis. I especially want to thank Leila Masri and Sara Sorce for their amazing dedication to making this graduate program the best that it can be. I am also deeply indebted to the German taxpayers who made it possible for me to freely pursue my research during these years and provided the funds needed for the large compute infrastructure that I was able to tap into. In particular, I acknowledge support from the German Federal Ministry of Education and Research (BMBF) through the Tübingen AI Center and from the Deutsche Forschungsgemeinschaft (DFG).

On a personal level, words cannot express the gratitude I have towards my wife for her unbelievably strong support throughout the entire journey. I simply could not ask for a better partner. Of course, I am deeply grateful to my parents who have helped make all the best things in my life possible and continue to support me to this day. I also want to thank Jean and John Lange who first introduced me to the wonderful world of programming and whose generosity reverberates through my life until today. Lastly, I wish to thank Steve Scoville who was the first person to make me believe that I could become a scientist.

Contents

1	Introduction	1
1.1	OOD Detection	2
1.2	Overconfidence	3
1.2.1	Calibration	4
1.2.2	Asymptotic Overconfidence	5
1.3	Adversarial Robustness	5
1.3.1	Empirical Robustness	6
1.3.2	Certified Robustness	8
1.4	Outline	10
I	Out-of-Distribution Detection	11
2	Benchmarking OOD Detection Methods	13
2.1	Introduction	13
2.2	Baselines	13
2.3	Experiments	15
2.3.1	Training	16
2.3.2	OOD detection performance	17
2.3.3	Results	17
2.4	Conclusion	18
3	Breaking down the Scoring Functions	21
3.1	Introduction	21
3.2	Models for OOD Data and Equivalence of OOD Detection Scores	22
3.3	Bayes-optimal Behaviour of Common OOD Detection Methods	24
3.3.1	OOD detection with methods using unlabeled data	24
3.3.2	OOD detection for methods using labeled data	26
3.3.3	Separate vs shared estimation of $p(i x)$ and $p(y x, i)$	30
3.4	Experiments	31
3.5	Conclusion	34

II	Adversarial Out-of-Distribution Detection	37
4	Adversarial robustness on in-and out-distribution improves explainability	39
4.1	Introduction	39
4.2	RATIO: Robust, Reliable and Explainable Classifier	40
4.2.1	In-Distribution Robustness and Adversarial Training	40
4.2.2	Out-Distribution Robustness and Adversarial Training on OOD	41
4.2.3	RATIO: Robustness via Adversarial Training on In-and Out-distribution	42
4.3	Visual Counterfactual Explanations	43
4.4	Experiments	44
4.4.1	Training	44
4.4.2	Calibration on the in-distribution	47
4.4.3	(Robust) Accuracy on the in-distribution	47
4.4.4	Visual Counterfactual Generation	48
4.4.5	Reliable Detection of (Adversarial) Out-of-Distribution Images	49
4.4.6	Feature Generation on OOD images	49
4.5	Conclusion	49
III	Certifiable Adversarial Out-of-Distribution Detection	57
5	Towards neural networks that provably know when they don't know	59
5.1	Introduction	59
5.2	Certified low confidence far away from the training data	60
5.2.1	A probabilistic model for in- and out-distribution data	60
5.2.2	Maximum likelihood estimation	62
5.2.3	Proofs of close to uniform predictions far away from data	63
5.3	Experiments	68
5.3.1	Training	68
5.3.2	Certified robustness against adversarial noise	69
5.3.3	Generating adversarial noise	71
5.4	Conclusion	73
6	Interval Bound Propagation for robust OOD Detection	75
6.1	Introduction	75
6.2	IBP for OOD	76
6.2.1	Upper bound on the confidence in terms of the logits	77
6.2.2	Quantile-GOOD: trade-off between clean and guaranteed AUC	79
6.3	Experiments	79
6.3.1	Training	80
6.3.2	Evaluation	82
6.3.3	Results	84

6.4	Conclusion	87
7	Provably Robust Detection of Out-of-distribution Data (almost) for free	89
7.1	Introduction	89
7.2	Provably Robust Detection of Out-of-distribution Data	90
7.2.1	Joint Model for OOD Detection and Classification	90
7.2.2	Certiably Robust Binary Discrimination of In- versus Out-Distribution	91
7.2.3	(Semi)-Joint Training of the Final Classifier	92
7.3	Guarantees on Asymptotic Confidence	94
7.4	Experiments	96
7.4.1	Training	96
7.4.2	Evaluation	97
7.4.3	Adversarial Asymptotic Overconfidence	101
7.5	Conclusion	103
8	Conclusion	105
8.1	Summary	105
8.2	Outlook	106
Abbreviations		109
Bibliography		111

Chapter 1

Introduction

In recent years, deep learning has been successfully used in a growing number of applications. With this success, there has also been increasing concern over the safety and reliability of systems that get deployed in situations where mistakes can directly harm humans. For example, consider AI used in the perception system of autonomous vehicles (Grigorescu *et al.*, 2020), the detection of tumors in a medical context (Saba *et al.*, 2020) or the scheduling of maintenance windows for aircraft engines (Chao *et al.*, 2022). In each of these instances, it is not sufficient for the models to have a high performance on a curated test set, but developers must ensure that the systems will not make dangerous mistakes even in potentially unforeseen circumstances. The European Union has even drafted legislature that aims to mandate safety standards for AI systems that operate in “high-risk applications” (EU, 2021).

However, the fuzzy notion of what would make such systems “reliable” is difficult to fully capture. Even if we limit ourselves to only considering classification tasks, reliability could entail many desirable properties, such as domain generalization (Dai and Van Gool, 2018; Geirhos *et al.*, 2018), robustness to random corruptions (Ghosh *et al.*, 2018; Hendrycks and Dietterich, 2019), robustness to adversarially crafted perturbations (Madry *et al.*, 2018; Goyal *et al.*, 2020) and confidence scores that accurately reflect the model’s true uncertainty (Guo *et al.*, 2017).

Additionally, when using a neural network in an open-world setting where any input could potentially occur, inputs to a classifier may not necessarily even belong to the classification task at all. In this case, the system should have the ability to reject samples altogether. Consider, for example, a classifier for different types of skin lesions (Lopez *et al.*, 2017) that gets presented with an unknown type of disease, or an autonomous vehicle that encounters a made-up street sign that does not actually exist. If an input can be flagged as out-of-distribution (OOD), then the system can abstain from making a decision in a safe state and potentially request human intervention. While this task has different names within the literature (Open Category Detection (Liu *et al.*, 2018) or Anomaly Detection (Hendrycks *et al.*, 2019a; Choi *et al.*, 2018)), we will refer to it as OOD detection and it will be the primary focus of this thesis.

Another property that we might require of a reliable OOD detection system is that of adversarial robustness, i.e. an OOD sample which is perturbed in a minimal way, should clearly still be detected as OOD. In this thesis we will see that even if a system seemingly performs well

on the OOD detection task, it might still fail in the case of such adversarially perturbed OOD samples. In this thesis, we will not only discuss the task of OOD detection in deep vision classifiers but additionally study the adversarial robustness of such systems.

1.1 OOD Detection

We will now give a brief outline of how the OOD detection task is defined and measured. First note that it can occasionally be debatable, whether a given sample should even be rejected at all if it is OOD or whether a given classifier should just correctly generalize to the unforeseen sample, as in the task of domain generalization. For example, in the safety-critical scenario of a tumor detector, when shown a scan from a previously unseen machine it could arguably either be desirable to correctly classify the new sample or to outright refuse to make a prediction. In order to avoid this ambiguity, this thesis only deals with OOD in the sense of samples that clearly do not belong to the same classification task at all, e.g. when showing a car to a classifier that is meant to classify different breeds of dogs.

Nonetheless, even with this caveat, providing a rigorous definition for OOD detection is an open problem. Informally though, it refers to the capacity of a given scoring function $h : \mathcal{X} \rightarrow \mathbb{R}$ to distinguish in-distribution (ID) data that is used for the classification task from unseen out-of-distribution (OOD) data (for the purposes of this thesis, the input space is either $\mathcal{X} = \mathbb{R}^d$ or $\mathcal{X} = [0, 1]^d$). We can denote these distributions as $p(x|i) \equiv p(x|\text{in-distribution})$ and $p(x|o) \equiv p(x|\text{out-distribution})$, respectively. We assume that, at test time, we are drawing from a distribution

$$p(x) = p(x|i)p(i) + p(x|o)(1 - p(i)), \quad (1.1)$$

where $p(i)$ is the prior probability of actually sampling from the in-distribution task. On the surface this might appear like its own binary classification task between in- and out-distribution, but, crucially, one should not assume knowledge of the test-out-distribution $p(x|o)$ at train time. Technically, this makes the problem ill-defined as one can always construct out-distributions on which a given scoring function will perform arbitrarily poorly. In principle, one could make additional assumptions such as disjoint support of $p(x|i)$ and $p(x|o)$, but this requires actually knowing the support of the in-distribution, which is highly non-trivial for complicated datasets such as is the case in vision applications. Therefore, in practice, in order to assess a scoring function's OOD detection performance, one can take a set of hand-picked test out-distributions and measure the model's ability to separate each of them from the in-distribution. In this setting, the choice of test out-distributions must be such that there is no semantic overlap between the in-distribution classes and the OOD samples. Different choices of out-distributions may also pose different levels of difficulty to the task. We thus tend to informally think of out-distributions as being either near-OOD or far-OOD, where the former matches the in-distribution in terms of low-level image statistics and can only be detected by looking at the semantics while the latter can differ from the in-distribution in more easily detectable ways.

Note that, in principle, actually separating ID and OOD would require the selection of a threshold. This could be systematically done by taking into account the costs of false positives and false negatives as well as the priors of in- and out-distribution in the deployed setting. Since these are not available to researchers studying standard computer vision datasets, the community tends to report the False Positive Rate (FPR) at a fixed True Positive Rate (TPR) of q , or FPR@ q TPR, for short. A lower FPR@ q TPR corresponds to better OOD detection performance. The quantile that is most often used in research is $q = 0.95$.

Another measure that is commonly used is the area under the receiver-operator characteristic (often called AUROC, but we will simply refer to it as AUC). This means that all possible choices of threshold trace out a curve of TPR vs. FPR and the higher the integral under this curve is, the better the model’s performance. Formally, given a scoring function f , an in-distribution $p(x|i)$ and an out-distribution $p(x|o)$, we can write:

$$\text{AUC}_h(p(x|i), p(x|o)) = \mathbb{E}_{\substack{x \sim p(x|i) \\ z \sim p(z|o)}} \left[\mathbb{1}_{h(x) > h(z)} + \frac{1}{2} \mathbb{1}_{h(x) = h(z)} \right]. \quad (1.2)$$

(Note that we will slightly modify this definition in Chapter 4.) The advantage is that the AUC does not depend on any particular choice of threshold, which is why we will use it as the primary metric throughout this thesis. A downside of using the AUC is that if method A has a higher AUC than method B, it does not guarantee that there is no TPR at which method B has lower FPR than method A.

Occasionally, researchers also report their OOD detection performance using the area under the precision-recall curve. Similarly, to the AUC, it sidesteps the issue of selecting a specific threshold. However, it still relies on the availability of the prior $p(i)$, which we do not wish to assume.

1.2 Overconfidence

Deep neural classifiers can be significantly overconfident in their predictions, both on in-distribution data and out-of-distribution. Here, confidence refers to the classifier’s estimated probability of the predicted class, given the sample. Formally, we refer to the output of a K -class classifier $f : \mathcal{X} \rightarrow \mathbb{R}^K$ as the *logits* and get a probability distribution over the classes via $\hat{p}_f(y|x) = \frac{e^{f_y(x)}}{\sum_k e^{f_k(x)}}$ for $y = 1, \dots, K$. Then we define the confidence as

$$\text{Conf}(f(x)) = \max_{y=1, \dots, K} \hat{p}_f(y|x). \quad (1.3)$$

Throughout the thesis, we denote underlying probabilities/densities with $p(y|x)$ resp. $p(x)$ and the corresponding estimated quantities with $\hat{p}(y|x)$ and $\hat{p}(x)$.

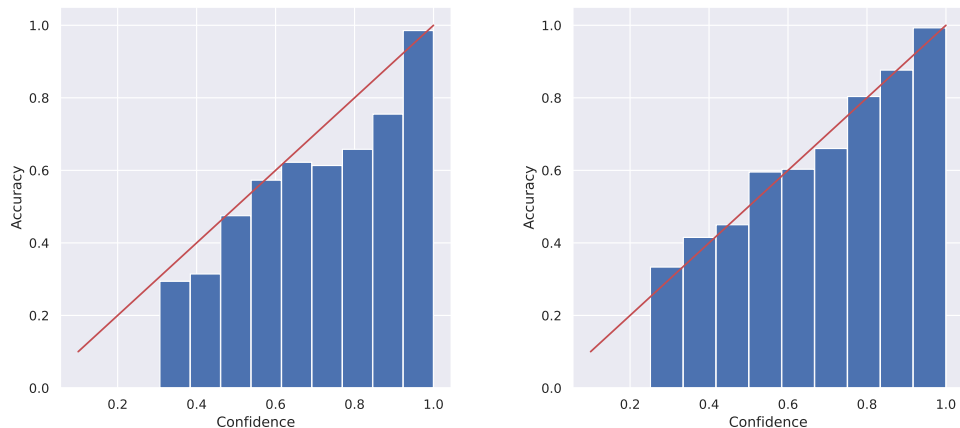


Figure 1.1: **Calibration:** The model predictions are grouped into 10 bins depending on the model’s confidence in its predictions. The accuracy within each bin is shown using the blue bars. An calibrated model’s predictions lie on the red line. On the left we show that a normally trained CIFAR10 model is overconfident, because its accuracy is generally lower than the confidence of each bin. On the right we show that this problem is alleviated after rescaling the logits with a temperature of $T = 1.4$.

1.2.1 Calibration

When saying that a classifier’s confidence is calibrated, we are stating that a predicted confidence of $p\%$ actually corresponds to a prediction that is $p\%$ likely to be correct. In order to actually compute a model’s level of calibration, the most widely used measure is the Expected Calibration Error (ECE) (Naeini *et al.*, 2015). It bins a classifier’s predictions on n data points according to their confidence into M subsets B_m and for each bin computes the difference between the true accuracy on the subset and the confidence on the respective subset, i.e.

$$\text{ECE} = \sum_{m=1}^M \frac{|B_m|}{n} |\text{accuracy}(B_m) - \text{confidence}(B_m)|. \quad (1.4)$$

A small ECE corresponds to better calibration. It has been shown that neural networks tend to produce overconfident and uncalibrated predictions (Guo *et al.*, 2017). A very common mitigation strategy is to element-wise transform the logits via so-called Platt scaling (Platt *et al.*, 1999) using parameters that are chosen on a validation set. This transformation is guaranteed to preserve the classifiers accuracy. The simplest such transformation is known as temperature rescaling, and given a temperature $T \in (0, \infty)$, it leads to a predictive distribution

$$\hat{p}(y|x) = \frac{e^{f_y(x)/T}}{\sum_k^K e^{f_k(x)/T}}. \quad (1.5)$$

A large temperature T leads to drastically reduced confidences. Also see Figure 1.1 for an illustration.

Note that many extensions or modifications of calibration in general and the ECE in particular have been proposed. On the one hand, the above definition only considers calibration after averaging across the entire dataset. However, calibration may be required on subsets of the data or even on individual inputs (Zhao *et al.*, 2020). Additionally, in certain settings the calibration with respect to the top-predicted class may not be sufficient and thus calibration may be required for each class separately (Nixon *et al.*, 2019; Kull *et al.*, 2019). There has also been significant work in ensuring that the estimation of calibration errors is consistent, scalable and has low bias (Vaicenavicius *et al.*, 2019; Roelofs *et al.*, 2022; Zhang *et al.*, 2020b; Kumar *et al.*, 2018; Popordanoska *et al.*, 2022).

1.2.2 Asymptotic Overconfidence

Overconfidence can also be problematic on OOD data, especially when we assume that $\mathcal{X} = \mathbb{R}^d$. The authors of Hein *et al.* (2019) made the following surprising observation: not only is it possible for deep neural classifiers to have high confidence on OOD data, it is in fact provably the case that for most ReLU networks the confidence asymptotically increases to 1, the further one moves away from all training data. The intuition for this is that any ReLU classifier (or more generally, any neural network that uses only piecewise linear non-linearities) can be thought of as a piecewise affine function that is affine when restricted to each polytope in a finite set of convex polytopes that together make up the whole domain \mathbb{R}^d . This means that if one moves very far from all training data, one must eventually end up in an outermost linear region, i.e. moving further along the same direction will never leave the current polytope. Once this happens, whichever logit has the largest slope in this direction will asymptotically be arbitrarily larger than all other logits and thus, the confidence will tend to 1 in this direction. An important caveat here is that this assumes that a unique largest slope exists. If several are tied for largest then the conclusion does not follow. However, unless the neural network is completely constant in a given direction (e.g. because all neurons in some layer have precisely 0 output), this caveat is extremely unlikely to occur. Interestingly, this theorem implies that this problem can really only be fixed by architectural modifications, which we will discuss more in Chapter 5.

1.3 Adversarial Robustness

A remarkable observation about neural networks is the fact that even extremely well-performing models can completely change their predictions when their inputs are only minimally (often imperceptibly) perturbed (Biggio *et al.*, 2013; Szegedy *et al.*, 2014). Many have argued that trustworthy models should not have this property and thus be adversarially robust (Madry *et al.*, 2018; Gowal *et al.*, 2020). This is especially relevant in situations where adversarial actors are able to present arbitrary samples to a model and have a clear incentive to manipu-

late the model’s predictions. For example, this would be the case in AI used for classifying job applicants according to their submitted information where applicants should not be able to use imperceptible or insignificant changes in order to game the system (Harlan and Schnuck, 2021). Besides the intuitively obvious desirability of adversarially robust models there has also been evidence that adversarial robustness helps with transfer learning (Salman *et al.*, 2020) and that it can imply certain interpretability properties (Santurkar *et al.*, 2019) that we will discuss in more detail in this thesis as well.

1.3.1 Empirical Robustness

Similarly to OOD detection, the task of adversarially robust classification is quite hard to define rigorously. The principal issue is that the notion of “inconspicuous” or “imperceptible” changes is hard to formalize. Because of this, the research community tends to use proxies that are known as threat models. A threat model defines a set $\mathcal{T}(x)$ around each point x , in which each element is assumed to be sufficiently similar to x in order to count as “inconspicuous”. An adversarial sample of x with respect to some threat model $\mathcal{T}(x) \subset \mathcal{X}$ is a point $x' \in \mathcal{T}(x)$ such that the decision of the classifier f changes for x' while an oracle would unambiguously associate x' with the class of x . In particular this implies that x' shows no meaningful class-associated features of any other class. Formally, given that y is the correct label of a correctly classified point x , then x' is an adversarial sample with respect to the threat model if

$$\arg \max_{k \neq y} f_k(x') > f_y(x), \quad x' \in [0, 1]^d \cap \mathcal{T}(x). \quad (1.6)$$

Typically, these perturbation sets are defined as l_p -balls

$$\mathcal{T}(x) = B_p(x, \varepsilon) = \{x' \in \mathcal{X} \mid \|x' - x\|_p \leq \varepsilon\} \quad (1.7)$$

of radius ε with $p \in [1, \infty)$ (Madry *et al.*, 2018). The adversarial robustness of a model f is determined by its loss under the worst-case perturbation within a given threat model:

$$\mathbb{E}_{x, y \sim p(x, y)} \max_{x' \in \mathcal{T}(x)} \mathcal{L}(f(x'), \mathbf{e}_y), \quad (1.8)$$

where \mathcal{L} is the chosen loss function and \mathbf{e}_y is the one-hot vector with the y ’s element being 1. For example

$$\mathcal{L}_{\text{CE}}(f(x), \mathbf{y}) = - \sum_{k=1}^K \mathbf{y}_k \log \hat{p}_f(k|x) \quad (1.9)$$

would be the cross-entropy loss. In the commonly chosen 0-1 loss, Eq. (1.8) defines the adversarial error or one minus the adversarially robust accuracy.

Note that for expressive function classes like deep neural networks the inner maximization in Eq. (1.8) is a highly non-convex function and thus it cannot feasibly be evaluated exactly. Therefore, a number of techniques have been developed in order to compute lower and upper

bounds on this term, which lead to the notions of *empirical robustness* and *certified robustness*, respectively. In empirical robustness, one tries to design algorithms which quickly find points with high loss within the neighborhood given by the threat model. The cheapest way is to use a single normalized gradient step, which is known as Fast Gradient Sign Method (FGSM) (Goodfellow *et al.*, 2015). Unfortunately, this often leads to non-robust models in a phenomenon known as “catastrophic overfitting” (Wong *et al.*, 2020) which is why the most successful attack algorithms rely on projected gradient descent (PGD) (Goodfellow *et al.*, 2015; Croce and Hein, 2020b), where many gradient steps are run, and after every step the current point is projected to lie within the threat model. Note that the gradient computation requires that one has access to the model architecture and weights. This setting is known as a white-box attack as opposed to a black-box attack, where only the outputs of the model can be queried. While, from the perspective of Eq. (1.8), there is no mathematically rigorous difference between these two settings, the latter is used in practice in order to capture the notion that someone trying to maliciously craft adversarial samples, does not necessarily have full knowledge of the model’s internal operation. However, typically a model that is robust to black-box attacks but not to white-box attacks would be considered “security through obscurity” and is not adversarially robust.

Even in the white-box setting, sometimes gradient-based attacks might unexpectedly fail to find samples with high loss, when a model obfuscates its gradients somehow (Athalye *et al.*, 2018). In these cases, empirically evaluating Eq. (1.8) may give a false sense of the true robustness of the system and, in fact, many techniques have been reported that were originally claimed to be adversarially robust but were later shown to not confer any meaningful adversarial robustness at all (Carlini and Wagner, 2017b; Mosbach *et al.*, 2018; Tramer *et al.*, 2020). Because of this, a reliable evaluation of a model’s adversarial robustness requires the use of a diverse ensemble of strong attack algorithms, such as AutoAttack (Croce and Hein, 2020b).

Currently, the most successful techniques for obtaining models that are adversarially robust, rely on variants of so-called *adversarial training* (Madry *et al.*, 2018; Zhang *et al.*, 2019; Croce *et al.*, 2021; Rebuffi *et al.*, 2021). Here one solves the following min-max problem:

$$\min_f \mathbb{E}_{x,y \sim p(x,y)} \max_{x' \in \mathcal{T}(x)} \mathcal{L}(f(x'), \mathbf{e}_y), \quad (1.10)$$

where the outer optimization gets approximately solved via SGD (or its variants) and the inner optimization is approximately solved using PGD at every step of SGD. Obviously, this greatly increases the cost of training as compared to simple empirical risk minimization (ERM). Furthermore, adversarial training has been observed to lead to much lower accuracy on the unperturbed test set (called “clean accuracy”) than ERM (Tsipras *et al.*, 2019; Schott *et al.*, 2019; Stutz *et al.*, 2019). Using additional data - either synthetic or real - can somewhat reduce this gap (Uesato *et al.*, 2019; Najafi *et al.*, 2019; Carmon *et al.*, 2019; Alayrac *et al.*, 2019; Hendrycks *et al.*, 2019b).

Nonetheless, adversarial robustness has not been achieved on complex datasets without a

reduction in clean performance, and several works indicate that this issue may be inevitable in certain settings (Tsipras *et al.*, 2019; Ilyas *et al.*, 2019; Zhang *et al.*, 2019), some have suggested to pursue the slightly relaxed task of either correctly classifying or rejecting adversarially perturbed samples (Xu *et al.*, 2017; Pang *et al.*, 2021; Sheikholeslami *et al.*, 2020). However, proposed defenses in this direction have also either been broken (Carlini and Wagner, 2017a; Tramer *et al.*, 2020) or still come at a cost in accuracy (Sheikholeslami *et al.*, 2020; Stutz *et al.*, 2020). In fact, recent work by Tramer (2022) has shown that robust classification can be reduced to robust classification with rejection at a larger radius, thus casting serious doubt on the notion that the latter is truly a simpler task.

1.3.2 Certified Robustness

Despite the success of adversarial training in producing models that are adversarially robust, we stress that any attack-based evaluation can only produce a lower bound on the adversarial error and the true value might always be as large as 1, no matter what an empirical evaluation suggests. Thus, full trustworthiness requires certification methods which can produce upper bounds instead. Generally, certification requires that one is able to make statements about the entire set of outputs into which the set of points in a given adversarial ball get mapped. Given that this set is highly non-convex and complex to describe, one usually has to form supersets on this set which are easier to characterize. There has been a lot of work on deriving such upper bounds that provide good trade-offs between the tightness of the bounds and their ease of computation, generally relying on convex upper bounds or branch and bound techniques (Wong and Kolter, 2018; Wong *et al.*, 2018; Raghunathan *et al.*, 2018b; Salman *et al.*, 2019; Bunel *et al.*, 2020; De Palma *et al.*, 2021).

However, even very sophisticated methods struggle to certify adversarially robust models that use large architectures with practically feasible computational budgets. Instead of certifying pre-trained models, one can thus attempt to specifically train models so that they are more easily certifiable. This line of work has been quite fruitful and in particular, it has shown the counter-intuitive result that simpler certification techniques tend to produce tighter bounds when used during training (Lee *et al.*, 2021b; Jovanović *et al.*, 2022). Concretely, for the popular l_∞ -threat model, very good results have been obtained using so-called Interval Bound Propagation (IBP) (Gowal *et al.*, 2018), which works as follows. We start from an axis-aligned hypercube and we aim to upper bound the set that a ReLU neural net maps this into during a forward pass. The first linear layer deforms the cube into a hyper-parallelepiped. Then the ReLU activations cut off parts of this parallelepiped. A very simple upper bound simply puts another axis-aligned box around the cut parallelepiped, which can then be treated the same way when moving through the subsequent layers. Therefore, IBP only requires a description of the current box and does not need to either look ahead nor look back in the forward pass. Also see Figure 1.2 for an illustration.

In order to mathematically formalize this geometric intuition, consider the feedforward network, $f : \mathbb{R}^d \rightarrow \mathbb{R}^K$, with K classes defined with input $x^{(0)} = x$ and layers $l = 1, \dots, L - 1$

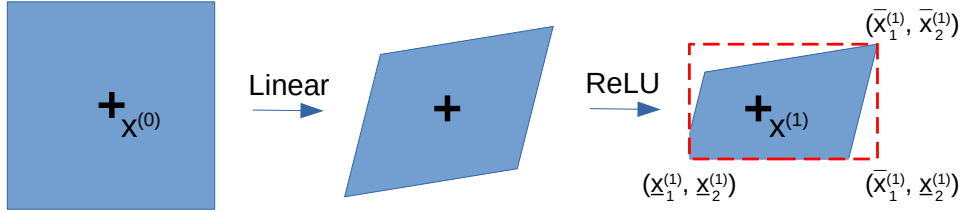


Figure 1.2: **Example of IBP step:** A two-dimensional l_∞ -box around an input $x^{(0)}$ first gets mapped into a parallelepiped by a linear layer. A ReLU layer then cuts off part of this set leaving a more complex shape around $x^{(1)}$. Interval Bound Propagation (IBP) computes an upper bound IBP computes overapproximates this set by the box shown in red.

as

$$x^{(l)} = \sigma^{(l)} \left(W^{(l)} x^{(l-1)} + b^{(l)} \right), \quad f(x) = W^{(L)} x^{(L-1)} + b^{(L)}, \quad (1.11)$$

where $W^{(l)}$ and $b^{(l)}$ are weights and biases and $\sigma^{(l)}$ is either the ReLU or leaky ReLU activation function of layer l . Note that it is possible to generalize IBP to more general activation functions as well. Now IBP can be used recursively at each layer in order to obtain upper and lower bounds $\bar{x}^{(L-1)}$ and $\underline{x}^{(L-1)}$ for the network's outputs:

$$\bar{x}^{(l)} = \sigma \left(W_+^{(l)} \bar{x}^{(l-1)} + W_-^{(l)} \underline{x}^{(l-1)} + b^{(l)} \right), \quad (1.12)$$

$$\underline{x}^{(l)} = \sigma \left(W_+^{(l)} \underline{x}^{(l-1)} + W_-^{(l)} \bar{x}^{(l-1)} + b^{(l)} \right), \quad (1.13)$$

where the indices $+/-$ indicate that we select the positive/negative weights component-wise while setting the negative/positive weights to 0. Thus, the entire bounding procedure requires only two forward passes and is automatically differentiable.

There also exist many other certification techniques beyond the ones that we discussed, e.g. methods that aim to control the Lipschitz-constant of the neural network's learned function, either via regularization (Hein and Andriushchenko, 2017) or by architectural construction (Anil *et al.*, 2019). Most notably though, randomized smoothing (Lecuyer *et al.*, 2019; Li *et al.*, 2019; Cohen *et al.*, 2019) runs many forward passes on samples that are perturbed by Gaussian noise and averages the prediction. The resulting effective classifier can be shown to provably be probabilistically adversarially robust. However, besides only certifying this relaxed notion of adversarial robustness and taking a very long time at verification time, this technique also significantly slows down all inference times. What all the methods for certified adversarial robustness have in common though, is that they generally come at a significant cost in terms of clean accuracy, even more so than empirically adversarially robust methods like adversarial training.

1.4 Outline

This thesis is broadly split into three parts. The first deals with standard OOD detection in vision-based classifiers. As described above, the task consists of recognizing samples that do not belong to the in-distribution classes. In principle, an unlimited number of scoring functions could be used for this task and, indeed, a large number of such functions has been proposed. In Chapter 2 we benchmark various methods and notice that the method that assumes access to a training out-distribution clearly outperforms other baselines. In Chapter 3, we go on to systematize OOD detection methods by defining a notion of equivalence between different scoring functions which we then apply to show that some methods are unexpectedly equivalent in the Bayes optimal limit. We further experimentally compare these approaches to one another and find that no method consistently outperforms the simple use of the classifier’s confidence, if this confidence score gets incentivized to be low on OOD during training time. We therefore continue to focus on this scoring function throughout the remainder of this thesis.

In the second part, we describe the problem of adversarial robustness for confidence-based OOD detection. Concretely, slightly perturbed OOD samples can lead to high confidences in models that otherwise show good OOD detection performance. In Chapter 4, we present work in which we show that adversarial training on in-distribution and out-distribution can lead to interesting synergies in creating more reliable classifiers. We further discuss the connection between adversarially robust confidence estimates on OOD data and the interpretability of models via counterfactual explanations.

Finally, in the third part, we tackle two issues that arise when training classifiers with low confidence on OOD data: i) The asymptotic overconfidence outlined in Section 1.2.2. ii) Similarly to adversarial robustness on the in-distribution, which we describe in Section 1.3, evaluating the adversarial robustness of the confidence scores leads to very difficult optimization problems that empirically often cannot be solved even approximately. We discuss how these issues can both be addressed via the derivation of mathematical guarantees.

Concretely, in Chapter 5 we focus on the first issue by combining a classifier with Gaussian mixture models, which are simple enough to allow us to control their asymptotic behavior, i.e. rather than going to 1, the asymptotic confidence can be shown to converge to uniform prediction across all classes. We additionally show that this approach also implies certain robustness guarantees on uniform noise data, when a specific threat model is selected. In Chapter 6, we show how IBP can be used to address the second issue, if one accepts certain limitations on which architectures one can use for the classification task. We find that it is possible to derive meaningful guarantees on unseen distributions, even if these distributions are very close to the in-distribution. Finally, in Chapter 7, we demonstrate that the advantages of both approaches can actually be combined into a method that provides bounds on unseen out-distributions, as well as provably asymptotically uniform predictions, without incurring any limitations on the architecture used or the final classification accuracy.

Part I

Out-of-Distribution Detection

Chapter 2

Benchmarking OOD Detection Methods

This chapter is based on parts of (Meinke and Hein, 2020) which was published at ICLR 2020. We postpone the presentation of other results from that work to Chapter 5 as they pertain to the detection of adversarially perturbed out-of-distribution samples. I was the first author of the paper and performed all experiments. Besides general guidance, Matthias Hein provided the initial idea and wrote significant parts of the paper.

2.1 Introduction

Perhaps in part because of the difficulty of defining and assessing a method’s OOD detection performance, a large number of methods has been proposed, often with contradicting claims about state-of-the-art performance. Because of this, we start this dissertation by benchmarking several well-known techniques for OOD detection in vision classifiers. We will see that despite some of the original works reporting near perfect performance, when running a fair comparison, the methods often lead to very mixed results. We will show that the strongest and most consistent performance gain over a simple baseline is attained by utilizing a large and diverse training out-distribution.

2.2 Baselines

In this Section, we briefly describe a subset of such methods which we then empirically evaluate in the rest of the chapter.¹

Plain: The confidence of a normally trained classifier is used as a the scoring function for OOD detection, as proposed in (Hendrycks and Gimpel, 2017a). As mentioned in Section 1.2, the confidence is simply defined as the maximum softmax output across classes, i.e. given a logit vector v :

$$\text{Conf}(v) = \max_y \frac{e^{v_y}}{\sum_k e^{v_k}}. \quad (2.1)$$

¹Note that this is based on work done in 2019 and therefore omits many modern baselines.

DE: Deep ensembles (Lakshminarayanan *et al.*, 2017) average the softmax outputs of five models that are adversarially trained via Fast Gradient Sign Method (FGSM) (Goodfellow *et al.*, 2015) with step size $\varepsilon = 0.01$.

MCD: Monte-Carlo Dropout (Gal and Ghahramani, 2016) was originally proposed as a method for uncertainty estimation. The idea is to use dropout layers both at train *and* at test time in order to approximate the use of an ensemble. While they did propose to use this method on classification tasks, they did not precisely specify how the actual uncertainty score ought to be computed from the model’s logits. Following (Shafaei *et al.*, 2019), we take the softmax from 7 forward passes and use the mean of the output for classification and the variance as the scoring function.

EDL: Evidential deep learning (Sensoy *et al.*, 2018) replaces the softmax layer of a neural network and introduces a different loss function that aims to encourage better uncertainty estimates. Concretely, they apply ReLU activations to the logits and treat the resulting values as evidence for each class. If the evidence for all classes is zero, the uncertainty is maximal, i.e. 1. The “belief” in a class y (analogous to confidence in softmax), given a logit vector v is
$$\frac{\text{ReLU}(v_y)}{K + \sum_k^K \text{ReLU}(v_k)}$$
.

GAN: The framework of confidence-calibrated classifiers (Lee *et al.*, 2017) relies on training a generative adversarial network alongside a classifier such that the GAN’s generator is encouraged to generate points close to but not in the in-distribution. On these points one then enforces uniform confidence. We used their provided code to train a VGG this way, as we were unable to adapt the method to a ResNet with an acceptable test error (e.g. test error < 30% on SVHN).

ODIN: The “**O**ut-of-**D**istribution detector for **N**eural networks” (ODIN) (Liang *et al.*, 2018) consists of two parts: a temperature T by which one applies temperature rescaling and a preprocessing step that applies a single FGSM-step (Goodfellow *et al.*, 2015) with step size ε in the direction that increases the model confidence before evaluating the input. In the original paper, the two parameters were calibrated on each test out-distribution. Note that the best OOD detection performance is generally achieved when using very high temperatures (around $T = 1,000$) and thus ODIN classifiers are severely underconfident.

Maha: The approach in Lee *et al.* (2018) is based on computing a class-conditional Mahalanobis distance in feature space and applying an ODIN-like preprocessing step for each layer. The original work then also aggregates Mahalanobis distances from different layers using layer-wise weights. This introduces many hyperparameters which they fit on test out-distributions. Because we do not wish to fit a large number of hyperparameters for each test out-distribution, following Ren *et al.* (2019) we use a single-layer version of the Maha method on our networks’ penultimate layers.

OE: Outlier exposure (Hendrycks *et al.*, 2019a) enforces uniform confidence on a large training OOD dataset. Note that, in principle, many loss functions could be designed that are minimized by uniform confidence across all classes. As in their original paper, and as in (Lee *et al.*, 2017) we use the cross-entropy between the model output and the uniform distribution, i.e. given a set of in-distribution samples $(x_r, y_r)_{r=1}^N$ and training OOD samples $(z_s)_{s=1}^M$, the

loss is:

$$-\frac{1}{N} \sum_{r=1}^N \log(\hat{p}(y_r|x_r)) - \frac{1}{M} \sum_{s=1}^M \frac{1}{K} \sum_{\ell=1}^K \log(\hat{p}(\ell|z_s)). \quad (2.2)$$

We used their provided code to train a model with our chosen architecture.

It is also important to note that OE is closely related to CEDA (Hein *et al.*, 2019), a method that also enforces low confidence on a training out-distribution. The difference is that CEDA used synthetic data instead of natural images. Since their CEDA method did not lead to large gains in OOD detection performance in their paper, we will only consider OE in this chapter.

ACET: Adversarial confidence enhanced training (ACET) (Hein *et al.*, 2019) enforces low confidence on a ball around points from an out-distribution by running adversarial attacks during training. That means the loss it optimizes is

$$-\frac{1}{N} \sum_{r=1}^N \log(\hat{p}(y_r|x_r)) + \frac{1}{M} \sum_{s=1}^M \max_{\ell \in \{1, \dots, K\}} \hat{p}(\ell|z_s). \quad (2.3)$$

We will discuss this method in a lot more detail in Chapter 4. In order to make the comparison with OE more meaningful we use 80M tiny images to draw the seeds rather than smoothed uniform noise as in Hein *et al.* (2019).

Some of the above OOD papers optimize their hyperparameters on a validation set for each out-distribution they test on. However, this leads to different classifiers for each out-distribution dataset which seems unrealistic as we want to have good generic OOD performance and not for a particular dataset. Thus, we keep the comparison realistic and fair by calibrating the hyperparameters of all methods on a subset of 80M tiny images and then evaluating on the other unseen distributions.

2.3 Experiments

We evaluate the OOD detection performance of all methods with the in-distributions MNIST (LeCun *et al.*, 1998), FashionMNIST (Xiao *et al.*, 2017), SVHN (Netzer *et al.*, 2011), CIFAR10 and CIFAR100 (Krizhevsky and Hinton, 2009). For calibrating hyperparameters resp. training, for all OOD methods we use the 80 Million Tiny Images (Torralba *et al.*, 2008) as out-distribution (Hendrycks *et al.*, 2019a) which yields a fair and realistic comparison. Throughout this dissertation, the deep learning framework we use is PyTorch (Paszke *et al.*, 2019). We also make heavy use of NumPy (Harris *et al.*, 2020) and SciPy (Virtanen *et al.*, 2020). Our code is available online.²

²<https://github.com/AlexMeinke/certified-certain-uncertainty>

Table 2.1: AUC (in- versus out-distribution detection based on the respective scoring function) in percent for different OOD detection methods and datasets (higher is better).

		Plain	MCD	EDL	DE	GAN	ODIN	Maha	ACET	OE
MNIST	FMNIST	97.4	93.1	99.3	99.2	99.4	98.7	96.8	100.0	99.9
	EMNIST	89.2	82.0	89.0	92.1	92.8	88.9	91.6	95.0	95.8
	GrCIFAR10	99.7	94.7	99.7	100.0	99.1	99.9	98.7	100.0	100.0
	Noise	100.0	95.2	99.9	100.0	99.3	100.0	97.2	100.0	100.0
	Uniform	95.2	87.9	99.9	97.9	99.9	98.2	100.0	100.0	100.0
FMNIST	MNIST	96.7	82.7	94.5	96.7	99.9	99.0	96.7	96.4	96.3
	EMNIST	97.5	87.3	95.6	97.1	99.9	99.3	97.5	97.6	99.3
	GrCIFAR10	91.0	92.3	84.0	86.1	85.3	93.0	98.2	96.2	100.0
	Noise	97.3	94.0	95.6	97.4	98.9	98.9	98.9	97.8	100.0
	Uniform	96.9	93.3	95.6	98.3	93.2	98.8	99.1	100.0	97.6
SVHN	CIFAR10	95.4	91.9	95.9	97.9	96.8	95.9	97.1	95.2	100.0
	CIFAR100	94.5	91.4	95.6	97.6	96.1	94.8	96.7	94.8	100.0
	LSUN_CR	95.6	92.0	95.3	97.9	99.0	96.5	97.2	97.1	100.0
	Imagenet-Noise	94.7	91.8	95.7	97.7	97.8	95.1	96.8	97.3	100.0
	Noise	96.4	93.1	97.1	98.2	96.2	82.7	98.0	95.8	97.8
	Uniform	96.8	93.1	96.5	95.6	100.0	97.9	97.8	100.0	100.0
CIFAR10	SVHN	95.8	81.9	92.3	90.3	83.9	96.7	91.5	93.7	98.8
	CIFAR100	87.3	78.6	87.3	88.2	82.9	87.5	82.8	86.9	95.3
	LSUN_CR	91.9	81.3	90.8	92.0	89.9	93.3	89.2	91.2	98.6
	Imagenet-Noise	87.5	78.4	88.2	87.7	84.0	88.1	84.1	86.5	94.7
	Noise	96.5	79.9	88.9	90.3	81.8	97.6	94.4	94.8	97.3
	Uniform	96.8	81.0	89.9	96.6	73.0	98.8	100.0	100.0	98.8
CIFAR100	SVHN	78.8	59.2	80.4	83.2	75.9	81.3	77.5	73.9	93.5
	CIFAR10	78.6	58.9	73.3	76.3	69.3	79.5	59.9	77.2	81.6
	LSUN_CR	81.0	59.4	74.2	81.6	79.8	81.4	79.7	78.0	95.4
	Imagenet-Noise	80.8	59.2	76.0	78.2	73.9	81.3	70.8	79.5	83.8
	Noise	73.4	58.7	65.9	67.5	73.6	76.8	90.6	62.9	86.9
	Uniform	93.3	62.0	29.8	36.6	100.0	93.5	94.3	100.0	99.1

2.3.1 Training

For all OOD methods we use LeNet on MNIST and a Resnet18 otherwise. Only for GAN and MCD we do not use resnets. In the case of GAN, the training does not stably train on Resnets due to the batchnorm layers and for MCD, it is unclear where to place dropout layers in a resnet. In both cases we instead use a VGG. Unless specified otherwise we use ADAM on MNIST with a learning rate of $1e-3$ and SGD with learning rate 0.1 for the other datasets. We decrease all learning rates by a factor of 10 after 50, 75 and 90 epochs. Our batch size is 128, the total number of epochs 100 and weight decay is set to $5e-4$.

When training ACET and OE with 80 million tiny images we pick equal batches of in- and

Table 2.2: We report the clean test error (TE) on the in-distribution in % (lower is better). Recall that GAN and MCD use VGG instead of ResNet18.

	Plain	MCD	EDL	DE	GAN	ODIN	Maha	ACET	OE
MNIST	0.5	0.4	0.4	0.4	0.8	0.5	0.9	0.6	0.7
FMNIST	4.8	5.8	5.2	4.9	5.7	4.8	4.8	4.8	5.7
SVHN	2.9	3.9	3.1	2.4	4.2	2.9	2.9	3.2	4.1
CIFAR10	5.6	11.7	7.0	6.7	11.7	5.6	5.6	6.1	4.7
CIFAR100	23.3	45.3	31.1	27.5	43.8	23.3	23.2	25.2	24.7

out-distribution data (corresponding to $p(i) = p(o)$) and concatenate them into batches of size 256. Note that during the 100 epochs only a fraction of the 80 million tiny images are seen and so there is no risk of over-fitting. For training ACET we use a PGD attack with 40 steps that maximize the maximal confidence over the classes. We also employ backtracking and halve the step size whenever the loss does not increase.

Our data augmentation scheme uses random crops with a padding of 2 pixels on MNIST and FMNIST. On SVHN, CIFAR10 and CIFAR100 the padding width is 4 pixels. For SVHN we fill the padding with the value at the boundary and for CIFAR we apply reflection at the boundary pixels. On top of this we include random horizontal flips on CIFAR. For MNIST and FMNIST we generate 60,000 such samples and for SVHN and CIFAR 50,000 samples by drawing from the clean dataset without replacement. During the actual training we use the same data augmentation scheme in a standard fashion.

2.3.2 OOD detection performance

For each dataset and method we report the AUC for the binary classification problem of discriminating in- and out-distribution based on their respective scoring functions. The results are shown in Table 2.1, where the list of datasets we use for OOD detection is also shown. LSUN_CR refers to only the classroom class of LSUN. Imagenet- is a subset of 10,000 resized Imagenet validation images, that have no overlap with CIFAR10/CIFAR100 classes. The noise dataset is obtained as in (Hein *et al.*, 2019) by first shuffling the pixels of the test images in the in-distribution and then smoothing them by a Gaussian filter of random width that is uniformly sampled from $[1, 2.5]$, followed by a rescaling so that the images have full range. GrCIFAR10 refers to the images in CIFAR10 being grayscale and resized to 28x28 and Uniform describes uniform noise over the $[0, 1]^d$ box.

2.3.3 Results

In addition to the OOD detection performance in Table 2.1, we also report each method’s test error on the in-distribution task in Table 2.2. Note that MCD’s and GAN’s low accuracies can be explained by their use of the VGG architecture. Perhaps surprisingly, EDL and DE incur

a significant cost in terms of accuracy. ODIN and Maha have identical or almost identical accuracy compared to Plain. Note that while temperature rescaling is guaranteed to not change the classification, the FGSM pre-processing step could, in principle change it. OE and ACET perform similar or marginally worse compared to Plain.

In terms of OOD detection performance, MCD is worse than the Plain model which confirms the results found in (Leibig *et al.*, 2017) that MCD is not useful for OOD detection. DE outperforms EDL but is not much better than the baseline for CIFAR10 and CIFAR100. The performance of Maha is worse than what has been reported in Lee *et al.* (2018) which can have two reasons. First, we just use their one-layer version where one uses the scores only from the pre-logit layer and second, we do not calibrate hyperparameters for each test set separately but just once on the Tiny Image dataset. Especially on CIFAR10 we find that the results depend strongly on the step size. The results of ACET, GAN and ODIN are mixed but generally outperform the Plain model. OE consistently performs better than the other methods. The gap to other methods is particularly striking where the OOD detection task is arguably the most challenging, i.e. CIFAR10 vs. CIFAR100 and vice-versa. In these cases we can informally consider the OOD data to be near out-distribution, as the image statistics are identical and only semantic information can be used for the detection. Note that ACET and OE are the only methods that incorporate additional data at train time instead of only during the selection of hyperparameters.

2.4 Conclusion

In this chapter we have benchmarked several different OOD detection methods. We clearly see that OE consistently shows the strongest performance without excessively hurting the in-distribution performance. The results also show that some methods that fitted their hyperparameters on the test out-distributions, do not generalize well to unseen out-distributions. This work motivates the further exploration of methods that incorporate unlabeled OOD data during training.

Recent developments: A large number of OOD detection methods have been proposed since we carried out this work, e.g. (Ren *et al.*, 2019; Yu and Aizawa, 2019; Sun *et al.*, 2021; Ming *et al.*, 2022; Lin *et al.*, 2021; Macêdo *et al.*, 2021; Gomes *et al.*, 2022). Some are directly based on outlier exposure, but introduce slight modifications which they claim lead to better performance (Papadopoulos *et al.*, 2021; Liu *et al.*, 2020; Chen *et al.*, 2021; Ming *et al.*, 2022). However, in general, it is difficult to assess if any of the published methods actually outperform the simpler baselines, because they often only report a cherry-picked subset of test out-distributions on which they outperform other methods. Recently a unified benchmark has been introduced that aims to make OOD detection methods more reliable to evaluate (Yang *et al.*, 2022), which may alleviate this problem in the future.

A particularly notable line of work (Fort *et al.*, 2021; Koner *et al.*, 2021) indicates that modern transformers (Dosovitskiy *et al.*, 2021) do not need to be specifically trained in order

for their confidence scores to be strong OOD detectors. In fact, they clearly outperform the OE models in this chapter. While some work suggests, that these promising results on vision transformers only hold when there is substantial overlap between the transformers pre-training set and the test out-distributions (Hendrycks *et al.*, 2022), it nonetheless corroborates the utility of additional unlabeled data found in this chapter.

Regarding ACET we found in later work, that the high accuracy of our ACET models in this chapter (and indeed in the original work (Hein *et al.*, 2019)) was in fact due to them not being fully robust to adversarial attacks on the out-distribution. As we will discuss in Chapter 4, successfully training fully robust ACET models is quite difficult.

It is also important to point out that the dataset 80 million tiny images (Torralba *et al.*, 2008), that we have used as a training out-distribution in this chapter has been retracted by the authors. The reason was that Birhane and Prabhu (2021) showed that the dataset contains offensive class labels and images.

Chapter 3

Breaking down the Scoring Functions

This chapter is based on (Bitterwolf *et al.*, 2022) which we presented at ICML 2022. Julian Bitterwolf and I jointly developed the idea for the paper, formulated the mathematical equivalence for OOD detection scores and its characterization and jointly derived the proofs for equivalences in Section 3.3.1. Julian Bitterwolf performed a majority of the experiments, derived Theorem 2 on his own and was the primary author of the paper. Another equivalence result in (Bitterwolf *et al.*, 2022) due to Julian Bitterwolf is omitted in this chapter. I ran the experiments on Restricted ImageNet. Max Augustin assisted in some of the writing. Matthias Hein provided general guidance and in-depth discussions to the project as well as some of the writing.

3.1 Introduction

In the previous chapter we empirically showed that, for the task of OOD detection, it is useful to assume access to a surrogate out-distribution during training, even if this surrogate OOD is not the same as the OOD data seen at test time. However, clearly, the enforcement of low confidence in a classifier is not the only way one could incorporate such a training OOD. Indeed, a large number of different approaches to OOD detection based on combinations of density estimation, classifier confidence, logit space energy, feature space geometry, behaviour on auxiliary tasks, and other principles has been proposed to tackle this problem. This begs the question if these approaches are related in meaningful ways.

Most work on OOD detection is focused on establishing superior empirical detection performance and provides little theoretical background on either differences or similarities to existing methods. Instead, in this chapter, our goal is to identify, at least for a particular subclass of techniques, whether differences in OOD detection performance are indeed due to a different underlying theoretical principle or whether they are due to the efficiency of different *estimation techniques* for the same underlying scoring function. In some cases, we will see that one can even disentangle the estimation procedure from the scoring function, so that one can simulate several different scoring functions from a single model’s estimated quantities.

In particular, we show that from the perspective of Bayesian decision theory, several established methods are indeed equivalent to a simple binary discriminator between in-distribution

and out-distribution. Differences arise mainly from i) the choice of the training out-distribution, and ii) differences in the estimation procedure. Concretely, the main contributions are:

- We formulate a notion of equivalence among scoring functions and give a simple characterization.
- We show that several OOD detection approaches which optimize an objective that includes predictions on surrogate OOD data are equivalent to the binary discriminator between in- and out-distribution when analyzing the rankings induced by the Bayes optimal classifier/density.
- We theoretically show that density estimation is equivalent to discrimination between the in-distribution and uniform noise which indicates why standard density estimates are not suitable for OOD detection, as has frequently been observed.
- We derive the implicit scoring functions for the confidence loss (Lee *et al.*, 2017) used by Outlier Exposure (Hendrycks *et al.*, 2019a), and for an extra background class for the out-distribution (Thulasidasan *et al.*, 2021). The confidence scoring function turns out not to be equivalent to the “optimal” scoring function of the binary discriminator when training and test out-distributions are the same.
- We show that the combination of a binary discriminator between in- and out-distribution with a standard classifier on the in-distribution, when trained in a shared fashion, yields OOD detection performance competitive with state-of-the-art methods based on surrogate OOD data.

The main aim of this chapter is a better understanding of the key components of different OOD detection methods and the identification of the key properties which lead to SOTA OOD detection performance. All of our findings are supported by extensive experiments on CIFAR10, CIFAR100 and Restricted ImageNet with evaluation on various challenging out-of-distribution test datasets.

3.2 Models for OOD Data and Equivalence of OOD Detection Scores

We first characterize the set of transformations of a scoring function which leaves OOD detection criteria like AUC or FPR invariant. This is important for the analysis later on, since the scoring functions of different methods are in many cases not identical as functions but yield the same OOD detection performance by those criteria. Recall from Eq. (1.2) that the AUC for a scoring function h distinguishing between an in-distribution $p(x|i)$ and an out-distribution

$p(x|o)$ is given by

$$\text{AUC}_h(p(x|i), p(x|o)) = \mathbb{E}_{\substack{x \sim p(x|i) \\ y \sim p(x|o)}} \left[\mathbb{1}_{h(x) > h(y)} + \frac{1}{2} \mathbb{1}_{h(x) = h(y)} \right]. \quad (3.1)$$

For ease of notation we will also refer to the in- and out-distributions as $p_{\text{in}}(x) \equiv p(x|i)$ and $p_{\text{out}}(x) \equiv p(x|o)$. We define an equivalence of scoring functions based on their AUCs and will show that this equivalence implies equality of other employed performance metrics as well.

Definition 1. *Two scoring functions h and g are equivalent and we write $h \cong g$ if*

$$\text{AUC}_h(p(x|i), p(x|o)) = \text{AUC}_g(p(x|i), p(x|o)) \quad (3.2)$$

for all potential distributions $p(x|i)$ and $p(x|o)$.

As the AUC is not dependent on the actual values of h but just on the ranking induced by h one obtains the following characterization of the equivalence of two scoring functions.

Theorem 1. *Two scoring functions h, g are equivalent $h \cong g$ if and only if there exists a strictly monotonously increasing $\phi : \text{range}(g) \rightarrow \text{range}(h)$ such that $h = \phi \circ g$.*

Proof.

- Assume that such a function ϕ exists. Then for any pair x, y we have the logical equivalences $g(x) > g(y) \Leftrightarrow h(x) = \phi(g(x)) > \phi(g(y)) = h(y)$ and $g(x) = g(y) \Leftrightarrow h(x) = \phi(g(x)) = \phi(g(y)) = h(y)$. This directly implies that the AUCs are the same, regardless of the distributions.
- Assume $h \cong g$. For each $a \in \text{range}(g)$, choose some $\hat{a} \in g^{-1}(a)$. For any pair $x, y \in X$, by regarding the Dirac distributions $p_{\text{in}} = \delta_x$ and $p_{\text{out}} = \delta_y$ that are each concentrated on one of the points, we can infer that $h(x) > h(y) \Leftrightarrow \text{AUC}_h(p_{\text{in}}, p_{\text{out}}) = 1 \Leftrightarrow \text{AUC}_g(p_{\text{in}}, p_{\text{out}}) = 1 \Leftrightarrow g(x) > g(y)$ and similarly $h(x) = h(y) \Leftrightarrow g(x) = g(y)$. The latter ensures that the function

$$\phi : \text{range}(g) \rightarrow \text{range}(h) \quad (3.3)$$

$$a \mapsto h(\hat{a}) \quad (3.4)$$

is independent of the choice of \hat{a} and that $h = \phi \circ g$, and the former confirms that ϕ is strictly monotonously increasing.

□

Corollary 1. *The equivalence between scoring functions in Def. 1 is an equivalence relation.*

As described in Chapter 2, besides the AUC, the FPR@qTPR is another important measure of OOD detection performance. The following lemma observes that using the AUC or FPR@qTPR at any value q for defining equivalence, leads to the same partitioning.

Lemma 1. *Two equivalent scoring functions $h \cong g$ have the same FPR@qTPR for any pair of in- and out-distributions $p(x|i), p(x|o)$ and for any chosen TPR q .*

Proof. We know that a function ϕ as in Theorem 1 exists. Then for any pair x, y , we have the logical equivalences $g(x) > g(y) \Leftrightarrow h(x) = \phi(g(x)) > \phi(g(y)) = h(y)$ and $g(x) = g(y) \Leftrightarrow h(x) = \phi(g(x)) = \phi(g(y)) = h(y)$. This directly implies that the FPR@qTPR-values are the same, for any p_{in}, p_{out} and q . \square

In the next section, we use the previous results to show that the Bayes optimal scoring functions of several proposed methods for out-of-distribution detection are equivalent to those of simple binary discriminators.

3.3 Bayes-optimal Behaviour of Common OOD Detection Methods

In the following we will show that the Bayes optimal function of several existing approaches to OOD detection for unlabeled data are equivalent to a binary discriminator between in- and a (training) out-distribution. As the equivalences are based on the Bayes optimal solution, these are asymptotic statements and thus it has to be noted that convergence to the Bayes optimal solution can be infinitely slow and that the methods can have implicit inductive biases. This is why we additionally support our findings with experiments in Section 3.4.

3.3.1 OOD detection with methods using unlabeled data

We first study the case when the labels y from the in-distribution are not used for the purpose of training an OOD detector.

Optimal prediction of a binary discriminator between in- and out-distribution: We consider a binary discriminator between in- and (training) out-distribution, where $\hat{p}_f(i|x)$ is the predicted probability for the in-distribution. Under the assumption that $p(i)$ is the probability for in-distribution samples and using cross-entropy (which in this case is the logistic loss up to a constant global factor of $\log(2)$) the expected loss becomes:

$$\min_f \quad p(i)\mathbb{E}_{x \sim p(x|i)} [-\log \hat{p}_f(i|x)] \quad + \quad p(o)\mathbb{E}_{x \sim p(x|o)} [-\log(1 - \hat{p}_f(i|x))] \quad . \quad (3.5)$$

One can derive that the Bayes optimal classifier minimizing the expected loss has the predictive distribution:

$$\hat{p}_{f^*}(i|x) = \frac{p(x|i)p(i)}{p(x|i)p(i) + p(x|o)p(o)} = p(i|x). \quad (3.6)$$

Thus, if the test out-distribution was identical to the training out-distribution, a binary classifier based on samples from in- and (training) out-distribution would suffice to solve the OOD detection problem optimally.

Equivalence of density estimation and binary discrimination for OOD detection: In this section we further analyze the relationship of common OOD detection approaches with the binary discriminator between in- and out-distribution. We start with density estimators sourced from generative models. A basic approach that is known to yield relatively weak OOD detection performance (Nalisnick *et al.*, 2019; Ren *et al.*, 2019; Xiao *et al.*, 2020) is directly utilizing a model’s estimate for the density $p(x|i)$ at a sample input x . An improved density based approach which uses perturbed in-distribution samples as a surrogate training out-distribution is the Likelihood Ratios method (Ren *et al.*, 2019), which proposes to fit a generative model for both the in- and out-distribution and to use the ratio between the likelihoods output by the two models as a discriminative feature.

We show that with respect to the scoring function, the true in-distribution density $p(x|i)$ is equivalent to the Bayes optimal prediction of a binary discriminator between the in-distribution and uniform noise. Furthermore, the density ratio $\frac{p(x|i)}{p(x|o)}$ is equivalent to the prediction of a binary discriminator between the two distributions on which the respective models used for density estimation have been trained. Because of this equivalence, we argue that the use of binary discriminators is a simple alternative to these methods because of its easier training procedure.

We first prove the more general case of arbitrary likelihood ratios. In the following, we use the abbreviation $\lambda = \frac{p(o)}{p(i)}$ to save space and make the statements more concise.

Lemma 2. Assume p_{in} and p_{out} can be represented by densities and the support of p_{out} covers the whole input domain X . Then $\frac{p(x|i)}{p(x|o)} \cong \frac{p(x|i)}{p(x|i) + \lambda p(x|o)}$ for any $\lambda > 0$.

Proof. The function $\phi : [0, \infty] \rightarrow [0, 1]$ defined by $\phi(x) = \frac{x}{x+\lambda}$ (setting $\phi(\infty) = 1$) fulfills the criterion from Theorem 1 of being strictly monotonously increasing. With

$$\phi\left(\frac{p(x|i)}{p(x|o)}\right) = \frac{\frac{p(x|i)}{p(x|o)}}{\frac{p(x|i)}{p(x|o)} + \lambda \frac{p(x|o)}{p(x|o)}} = \frac{p(x|i)}{p(x|i) + \lambda p(x|o)} \quad (3.7)$$

for $p_{\text{out}} \neq 0$ and $\phi\left(\frac{p(x|i)}{0}\right) = \phi(\infty) = 1 = \frac{p(x|i)}{p(x|i) + \lambda \cdot 0}$, the equivalence follows. \square

This means that the likelihood ratio score of two optimal density estimators is equivalent to the in-distribution probability $\hat{p}_{f^*}(i|x)$ predicted by a binary discriminator and this is true

for any possible ratio of $p(i)$ to $p(o)$. In the experiments below, we show that, indeed, using such a discriminator has similar performance to the likelihood ratios of the different trained generative models.

For the approaches that try to directly use the likelihood of a generative model as a discriminative feature, this means that their objective is equivalent to training a binary discriminator against uniform noise, whose density is $p_{\text{Uniform}}(x) = p(x|o) = 1$ at any x .

Lemma 3. *Assume that p_{in} can be represented by a density. Then $p(x|i) \cong \frac{p(x|i)}{p(x|i)+\lambda}$ for any $\lambda > 0$.*

Proof. This is a special case of Lemma 2, by setting $p(x|o) = 1 = p_{\text{Uniform}}(x)$. □

This provides additional evidence why a purely density based approach for many applications proves to be insufficient as an OOD detection score on the complex image domain: it is not reasonable to assume that a binary discriminator between certain classes of natural images on the one hand and uniform noise on the other hand provides much useful information about images from other classes or even about other nonsensical inputs.

As a side note, one idea that has often been informally suggested to us is that of training a discriminator against a probability distribution that has mass precisely wherever the in-distribution does not have mass. One way of formalizing this under the assumption that p_{in} is bounded would be as follows:

$$p^c = v \cdot (1 - \alpha p_{\text{in}}), \tag{3.8}$$

where $\alpha \in (0, 1)$ is chosen small enough such that $\forall x \in [0, 1]^D : p^c \geq 0$, and $v = \frac{1}{1-\alpha}$ is a normalization constant.

Lemma 4. *Assume that p_{in} can be represented by a density. Then $\frac{p(x|i)}{p(x|i)+\lambda p^c(x)} \cong p(x|i)$ for any $\lambda > 0$.*

Proof. $\frac{p(x|i)}{p(x|i) + \lambda p^c(x)} \cong \frac{p(x|i)}{p^c(x)} = \frac{p(x|i)}{v \cdot (1 - \alpha p(x|i))}$ is strictly monotonically increasing with respect to $p(x|i)$, as its derivative is $\frac{1}{v \cdot (1 - \alpha p(x|i))^2} > 0$; note that the domain of this function is a subset of $[0, \frac{1}{\alpha})$. □

3.3.2 OOD detection for methods using labeled data

We first discuss how one can formulate the OOD problem when one has access to labeled data for the in-distribution and we identify the target distribution of OOD detection using a background/reject class. Then we derive the Bayes optimal classifier of the confidence loss (Lee *et al.*, 2017) as used by the most popular variant of Outlier Exposure (Hendrycks *et al.*, 2019a) (also used in Chapter 2) and discuss the implicit scoring function. In most cases the scoring functions turn out not to be equivalent to $p(i|x)$ (which is optimal if training and

test out-distribution agree) as they integrate additional information from the classification task. Given a joint in-distribution $p(y, x|i)$ (where $y \in \{1, \dots, K\}$ given that we have K labels) for the labeled in-distribution, there are different ways how to come up with a joint distribution for in- and out-distribution. Interestingly, the different encodings used e.g. in training with a background class (Thulasidasan *et al.*, 2021) vs. training a classifier with confidence loss (Lee *et al.*, 2017) together with variants of the employed scoring function lead to methods which unexpectedly can have quite different behavior.

Background class: In this case we just put all out-of-distribution samples into a $K + 1$ -class which is typically called background/reject class (Thulasidasan *et al.*, 2021). The joint distribution then becomes

$$p(y, x) = \begin{cases} p(y, x|i)p(i) & \text{if } y \in \{1, \dots, K\}, \\ p(x|o)p(o) & \text{if } y = K + 1. \end{cases} \quad (3.9)$$

We denote by $p(x|i) = \sum_{y=1}^K p(y, x|i)$ the marginal in-distribution and note that the marginal distribution of the joint distribution of in- and out-distribution is again

$$p(x) = p(x|i)p(i) + p(x|o)p(o).$$

Thus we get the conditional distribution

$$p(y|x) = \begin{cases} p(y|x, i)p(i|x) & \text{if } y \in \{1, \dots, K\}, \\ p(o|x) = 1 - p(i|x) & \text{if } y = K + 1. \end{cases}$$

The Bayes optimal solution of training with a background class using any calibrated loss function $L(f(x), \mathbf{y})$, e.g. the cross-entropy loss (Laptev *et al.*, 2016), then yields a Bayes optimal classifier f^* which has a predictive distribution $\hat{p}_{f^*}(y|x) = p(y|x)$. There are two potential scoring functions that come to mind:

$$s_1(x) = 1 - \hat{p}_{f^*}(y = K + 1|x) \quad \text{and} \quad s_2(x) = \max_{k=1, \dots, K} \hat{p}_{f^*}(k|x) \quad (3.10)$$

The first one, used in Chen *et al.* (2021); Thulasidasan *et al.* (2021), is motivated by the fact that $\hat{p}_{f^*}(y = K + 1|x)$ is directly the predicted probability that the point is from the out-distribution as indeed it holds: $s_1(x) = p(i|x)$ which is the optimal scoring function if training and test out-distribution are equal. On the other hand the maximal predicted probability $\max_{k=1, \dots, K} \hat{p}_{f^*}(k|x)$, which is often employed as a scoring function (Hendrycks and Gimpel, 2017a), becomes for the Bayes optimal classifier

$$s_2(x) = p(i|x) \max_{k=1, \dots, K} p(k|x, i),$$

which is a product of $p(i|x)$ and the maximal conditional probability of some class of the in-distribution; note that s_2 is well defined as $p(i|x)$ is defined if $p(x|o)$ has support everywhere in X and if $p(i|x) > 0$ then also $p(x|i) > 0$. Thus, the scoring function $s_2(x)$ integrates class-specific information in addition to $p(i|x)$ and is therefore less dependent on the chosen training out-distribution. In fact, one can see that s_2 only ranks points high if both the binary discriminator *and* the classifier rank the corresponding point high. However, in the case where training and test out-distribution are identical, this scoring function is not equivalent to $p(i|x)$ and thus introduces a bias in the estimation.

Outlier Exposure (Hendrycks *et al.*, 2019a) with confidence loss (Lee *et al.*, 2017): We analyze the Bayes optimal solution for the confidence loss (Lee *et al.*, 2017) that is used by Outlier Exposure (OE) and show that the associated scoring function can be written, similarly to the scoring function $s_2(x)$ for training with a background class, as a function of $p(i|x)$ and $p(y|x, i)$. Recall that the finite sample loss in Eq. (2.2) is an estimator for the OE training objective with the confidence loss in expectation:

$$\min_f \mathbb{E}_{(x,y) \sim p(x,y|i)} [\mathcal{L}_{\text{CE}}(f(x), \mathbf{e}_y)] + \lambda \mathbb{E}_{x \sim p(x|o)} [\mathcal{L}_{\text{CE}}(f(x), \mathbf{1}/K)], \quad (3.11)$$

where $f(x) \in \mathbb{R}^K$ is the model output as logits, and $\mathbf{1}/K = (\frac{1}{K}, \dots, \frac{1}{K})^T$ is the uniform distribution over the K classes of the in-distribution classification task.

In the following theorem we derive the Bayes optimal predictive distribution for this training objective. Note that we will use $\hat{p}_f(x)[k]$ to denote the k -th component of $\hat{p}_f(x)$ in order to avoid confusion in the index notation.

Theorem 2. *The predictive distribution $\hat{p}_{f^*}(y|x)$ of the Bayes optimal classifier f^* minimizing the expected confidence loss is given for $y \in \{1, \dots, K\}$ as*

$$\hat{p}_{f^*}(y|x) = p(i|x)p(y|x, i) + \frac{1}{K}(1 - p(i|x)). \quad (3.12)$$

Proof. This is the minimization problem

$$\min_{\hat{p}_f(x)} -p(i|x) \cdot \sum_{k=1}^K p_{\text{in}}(k|x) \cdot \log \hat{p}_f(x)[k] - (1 - p(i|x)) \cdot \sum_{k=1}^K \frac{1}{K} \cdot \log \hat{p}_f(x)[k] \quad (3.13)$$

$$\text{subject to } \hat{p}_f(x)[k] \geq 0 \text{ for each } k \in \{1, \dots, K\} \quad (3.14)$$

$$\sum_{k=1}^K \hat{p}_f(x)[k] = 1. \quad (3.15)$$

For $p(i|x) = 0$ or $p_{\text{in}}(k|x) = 0$, the optimalities of the respective terms are easy to show (applying the common conventions for $0 \log 0$), so we assume those to be non-zero. The Lagrange

function of the optimization problem is

$$L(\hat{p}_f(x), \alpha, \beta) = -p(i|x) \cdot \sum_{k=1}^K p_{\text{in}}(k|x) \cdot \log \hat{p}_f(x)[k] - (1 - p(i|x)) \cdot \sum_{k=1}^K \frac{1}{K} \cdot \log \hat{p}_f(x)[k] \quad (3.16)$$

$$- \sum_{k=1}^K \alpha_k \hat{p}_f(x)[k] + \beta \left(-1 + \sum_{k=1}^K \hat{p}_f(x)[k] \right), \quad (3.17)$$

with $\beta \in \mathbb{R}$ and $\alpha \in \mathbb{R}_+^K$. Its first derivative with respect to $\hat{p}_f(x)[k]$ is

$$\begin{aligned} \frac{\partial L}{\partial \hat{p}_f(x)[k]} &= -p(i|x) \cdot p_{\text{in}}(k|x) \frac{1}{\hat{p}_f(x)[k]} - (1 - p(i|x)) \cdot \frac{1}{K} \frac{1}{\hat{p}_f(x)[k]} - \alpha_k + \beta \\ &= -\frac{s^K(x)[k]}{\hat{p}_f(x)[k]} - \alpha_k + \beta. \end{aligned} \quad (3.18)$$

The second derivative is a positive diagonal matrix on the domain, therefore we find the unique minimum by setting Eq. (3.18) to zero, which means

$$\hat{p}_f(x)[k] = \frac{s^K(x)[k]}{\beta - \alpha_k}. \quad (3.19)$$

The dual problem is hence maximizing (with $\alpha_k \geq 0$)

$$q(\alpha, \beta) = -p(i|x) \cdot \sum_{k=1}^K p_{\text{in}}(k|x) \cdot \log \frac{s^K(x)[k]}{\beta - \alpha_k} - (1 - p(i|x)) \cdot \sum_{k=1}^K \frac{1}{K} \cdot \log \frac{s^K(x)[k]}{\beta - \alpha_k} \quad (3.20)$$

$$- \sum_{k=1}^K \alpha_k \frac{s^K(x)[k]}{\beta - \alpha_k} + \beta \left(-1 + \sum_{k=1}^K \frac{s^K(x)[k]}{\beta - \alpha_k} \right) \quad (3.21)$$

$$= \sum_{k=1}^K s^K(x)[k] \left(-\log s^K(x)[k] + \log(\beta - \alpha_k) + \frac{\beta}{\beta - \alpha_k} - \frac{\alpha_k}{\beta - \alpha_k} \right) - \beta; \quad (3.22)$$

here, α only appears in $\log(\beta - \alpha_k)$, so $\alpha = 0$ maximizes the expression. Noting $\sum_{k=1}^K s^K(x)[k] = 1$, what remains is $q^0(\beta) = 1 + \log(\beta) - \sum_{k=1}^K s^K(x)[k] \log s^K(x)[k] - \beta$, which is maximized by $\beta = 1$. This means that the dual optimal pair is $\hat{p}_f(x)[k] = s^K(x)[k]$, $(\beta = 1, \alpha = 0)$. Slater's condition (Boyd *et al.*, 2004) holds since the feasible set of the original problem is the probability simplex. Thus, $\hat{p}_f(x) = s^K(x)$ is indeed primal optimal. \square

Thus the effective scoring function of using the probability of the predicted class as sug-

gested in Hendrycks and Gimpel (2017a); Lee *et al.* (2017); Hendrycks *et al.* (2019a) is

$$\begin{aligned} s_3(x) &= p(i|x) \max_{y=1,\dots,K} p(y|x,i) + \frac{1}{K}(1 - p(i|x)) \\ &= p(i|x) \left[\max_{y=1,\dots,K} p(y|x,i) - \frac{1}{K} \right] + \frac{1}{K}. \end{aligned} \quad (3.23)$$

Note that the term inside the brackets is positive as $\max_{k=1,\dots,K} p(k|x,i) \geq \frac{1}{K}$. Interestingly, the scoring functions s_2 and s_3 are not equivalent even though they look quite similar. In particular, due to the subtraction of $\frac{1}{K}$ the scoring function s_3 puts more emphasis on the classifier than s_2 .

3.3.3 Separate vs shared estimation of $p(i|x)$ and $p(y|x,i)$

So far we have derived that at least from the point of view of the ranking induced by the Bayes optimal solution, OOD detection based on generative methods, likelihood ratios, and the background class formulation with the scoring function s_1 is equivalent to a binary classification problem between in- and out-distribution in order to estimate $p(i|x)$. The differences arise mainly in the choice of the training out-distribution $p(x|o)$: i) uniform for generative resp. density based methods, ii) a synthetic out-distribution for likelihood ratios (Ren *et al.*, 2019) and iii) a proxy of the distribution of all natural images (Hendrycks *et al.*, 2019a; Thulasidasan *et al.*, 2021). On the other hand when labeled data is involved we can additionally train a classifier on the in-distribution in order to estimate $p(y|x,i)$. We will then combine the estimates of $p(i|x)$ and $p(y|x,i)$ according to the three scoring functions derived in the previous section and check if the novel OOD detection methods constructed in this way perform similar to the OOD methods from which we derived the corresponding scoring function i) OOD detection with a background class (Thulasidasan *et al.*, 2021) or ii) using Outlier Exposure (Hendrycks *et al.*, 2019a). This will allow us to differentiate between differences of the employed scoring functions for OOD detection and the estimators for the involved quantities. In this way we foster a more systematic approach to OOD detection.

In the unlabeled case we simply train the binary classifier $\hat{p}_f : [0,1]^d \rightarrow \mathbb{R}$ using logistic/cross entropy loss in a class balanced fashion

$$\min_f \left(-\frac{1}{N} \sum_{r=1}^N \log(\hat{p}_f(i|x_r)) - \frac{\lambda}{M} \sum_{s=1}^M \log(1 - \hat{p}_f(i|z_s)) \right), \quad (3.24)$$

where $(x_r)_{r=1}^N$ and $(z_s)_{s=1}^M$ are samples from the in-distribution and the out-distribution.

In the case where we have labeled in-distribution data, we can additionally solve the classification problem. The obvious approach is to train the binary classifier for estimating $p(i|x)$ and the classifier to estimate $p(y|x,i)$ completely independently. We show in Section 3.4 that this approach does not work as well as allowing the models to share their representations. In fact both tasks benefit from each other. Moreover, in training a neural network using a

background class or with Outlier Exposure (Hendrycks *et al.*, 2019a) we are implicitly using a shared representation for both tasks which improves the results.

Thus, we propose to train the binary discriminator of in-versus out-distribution together with the classifier on the in-distribution jointly. Concretely, we use a neural network with $K + 1$ outputs where the first K outputs represent the classifier and the last output is the logit of the binary discriminator. The resulting shared problem can then be written as

$$\min_f \left(-\frac{1}{N_b} \sum_{r=1}^{N_b} \log \hat{p}_f(i|x_r) - \frac{\lambda}{M} \sum_{s=1}^M \log (1 - \hat{p}_f(i|z_s)) - \frac{1}{N_c} \sum_{t=1}^{N_c} \log \hat{p}_f(y_t|x_t) \right), \quad (3.25)$$

where $\lambda = \frac{p(o)}{p(i)}$ which is typically set to 1 during training in order to get a class-balanced problem. Note that the in-distribution samples $(x_r)_{r=1}^{N_b}$ used to estimate $p(i|x)$ can, in principle, be a super-set of the labeled examples $(x_t, y_t)_{t=1}^{N_c}$ used to train the classifier so that one can potentially integrate unlabeled data - this is an advantage compared to OOD detection with a background class or Outlier Exposure where this is not directly possible. We stress that the loss functions of the classifier and the discriminator act on independent outputs; the functions modelling the two tasks only interact with each other due to the shared network weights up to the final layer. Nevertheless, we see in the next Section 3.4 that training with a shared representation boosts both the classifier and the binary discriminator.

3.4 Experiments

Training: We use CIFAR10, CIFAR100 (Krizhevsky and Hinton, 2009) datasets as in-distribution and OpenImages dataset (Krasin *et al.*, 2017; Kuznetsova *et al.*, 2020) as training out-distribution. The 80 Million Tiny Images (80M) dataset (Torralba *et al.*, 2008) is the de facto standard for training out-distribution aware models that has been adopted by most prior works, but this dataset has been withdrawn by the authors as (Birhane and Prabhu, 2021) pointed out the presence of offensive images. As in the previous chapter, we use the AUC as our evaluation metric for OOD detection performance. For CIFAR, we use the following datasets as our test out-distributions: SVHN, resized LSUN Classroom, Uniform Noise, the respective other CIFAR dataset, 80M, and CelebA (Liu *et al.*, 2015) and Smooth Noise. Note that CelebA does not make sense as out-distribution for CIFAR100, because humans are in fact some of the CIFAR100 classes. For Uniform and Smooth noise, we evaluate 30,080 inputs. We emphasize that, again, none of the listed methods has access to those test distributions during training or for fine-tuning as we try to assess the ability of an OOD aware model to generalize to unseen distributions. The AUC for the OpenImages test set is not part of the Mean AUC, since OpenImages was seen during training.

The binary discriminators (BINDISC) as well as the classifiers with background class (BGC) and the shared binary discriminator+classifier (SHARED) of $p(i|x)$ and $p(y|x, i)$ are trained on the 40-2 Wide Residual Network (Zagoruyko and Komodakis, 2016) architecture with the

Chapter 3 Breaking down the Scoring Functions

Table 3.1: Accuracy on the in-distribution (CIFAR10/CIFAR100/RImgNet) and AUC for test OODs of the different methods with OpenImages/Not Restricted ImageNet as training OOD. Best method on each distribution marked in green and best accuracy / mean AUC boldface.

in-distribution: CIFAR10										
Model	Acc.	Mean	SVHN	LSUN	Uni	Smooth	C100	80M	CeLA	OpenIm
Plain Classi	95.16	91.85	93.52	92.94	97.04	92.84	89.61	91.30	85.70	84.81
Separate BinDisc (s_1)		89.03	96.42	100.00	99.97	99.99	58.60	72.36	95.87	99.99
OE (s_3)	95.06	97.28	98.49	99.99	99.99	99.99	90.03	92.53	99.91	99.43
BGC s_1		95.02	99.48	100.00	99.99	99.95	79.64	86.37	99.74	99.97
BGC s_2	95.21	97.22	98.90	100.00	99.99	99.67	90.47	92.41	99.11	99.73
BGC s_3	95.21	97.21	98.87	100.00	99.98	99.62	90.47	92.41	99.08	99.71
Shared BinDisc (s_1)		92.51	98.77	100.00	99.89	99.93	68.34	80.81	99.80	99.95
Shared Classi	95.28	95.49	96.10	98.60	99.06	96.09	90.09	92.35	96.18	93.57
Shared Combi s_2	95.28	97.26	98.66	100.00	99.93	99.94	89.71	92.84	99.72	99.88
Shared Combi s_3	95.28	97.26	98.62	100.00	99.93	99.94	89.75	92.85	99.71	99.88
in-distribution: CIFAR100										
Model	Acc.	Mean	SVHN	LSUN	Uni	Smooth	C10	80M		OpenIm
Plain Classi	77.16	82.13	82.33	79.13	96.03	81.36	76.14	77.80		75.80
Separate BinDisc (s_1)		84.30	94.68	100.00	99.81	99.64	50.06	61.62		99.98
OE (s_3)	77.19	90.37	89.54	99.98	99.03	99.68	75.95	78.03		99.67
BGC s_1		88.41	97.38	99.99	99.70	99.79	60.51	73.11		99.93
BGC s_2	77.61	90.47	90.50	99.99	99.87	99.75	74.88	77.82		99.64
BGC s_3	77.61	90.46	90.46	99.99	99.88	99.74	74.88	77.82		99.64
Shared BinDisc (s_1)		84.62	97.44	99.99	99.70	99.68	47.82	63.13		99.93
Shared Classi	77.35	82.06	82.72	99.05	72.73	84.14	75.76	77.99		93.54
Shared Combi s_2	77.35	90.74	91.74	99.99	99.59	99.54	75.50	78.10		99.57
Shared Combi s_3	77.35	90.73	91.69	99.99	99.57	99.53	75.50	78.10		99.57
in-distribution: Restricted ImageNet										
Model	Acc.	Mean	Flowers	FGVC	Cars	Smooth	Uni			NotRIN
Plain Classi	96.34	94.96	91.65	92.67	92.46	98.74	99.26			92.38
OE (s_3)	97.10	98.76	96.65	99.75	99.85	97.95	99.58			98.46
BGC s_1		98.61	96.64	99.86	99.97	97.77	98.80			98.67
BGC s_2	97.50	98.66	96.39	99.83	99.96	98.18	98.94			98.69
BGC s_3	97.50	98.66	96.43	99.83	99.96	98.14	98.93			98.68
Shared BinDisc (s_1)		98.26	97.62	99.83	99.94	96.13	97.78			98.71
Shared Classi	97.59	96.93	93.40	96.58	96.53	99.48	98.66			96.10
Shared Combi s_2	97.59	98.54	97.41	99.80	99.93	97.37	98.18			98.72
Shared Combi s_3	97.59	98.58	97.36	99.79	99.92	97.61	98.22			98.71

same training schedule as used in (Hendrycks *et al.*, 2019a) for training their Outlier Exposure(OE) models. This includes averaging the loss over batches that are twice as large for the out-distribution. This way we ensure that the differences do not arise due to differences in the training schedules or other important details but only on the employed objectives. In addition to their standard augmentation and normalization, we apply AutoAugment (Cubuk *et al.*, 2019) without Cutout, and we use $\lambda = 1$ where applicable. Our code is available on github.¹ It builds upon the code of Hendrycks *et al.* (2019a) available at <https://github.com/hendrycks/outlier-exposure> and we use their general architecture and training settings. Concretely, we use 40-2 Wide Residual Network (Zagoruyko and Komodakis, 2016) models with normalization based on the CIFAR training datasets and a dropout rate of 0.3. They are trained for 100 epochs with an initial learning rate of 0.1 that decreases following a cosine annealing schedule. Unless mentioned otherwise, each training step uses a batch of size 128 for the in-distribution and a batch of size 256 for the training out-distribution. The optimizer uses stochastic gradient descent with a Nesterov momentum of 0.9. Weight decay is set to $5e-4$.

In addition to the results for CIFAR10 and CIFAR100 we also run experiments on Restricted ImageNet. Restricted ImageNet, introduced by Tsipras *et al.* (2019), consists of 9 classes, where each individual class is a union of multiple ImageNet (Deng *et al.*, 2009) classes, for example the Restricted ImageNet class 'dog' contains all dog breeds from ImageNet. As Restricted ImageNet only contains animal classes, the union over all its classes does not cover the entire ILSVRC2012 dataset (Russakovsky *et al.*, 2015), which allows us to use the remaining ILSVRC2012 classes as training out-distribution. The corresponding test out-distributions are Flowers (Nilsback and Zisserman, 2008), FGVC Aircraft (Maji *et al.*, 2013), Stanford Cars (Krause *et al.*, 2013), as well as Smooth noise and Uniform noise. Like for the CIFAR experiments, we train a plain classifier, an Outlier Exposure model, a background class model and a shared discriminator/classifier and evaluate them with the different scoring functions. The model is a ResNet50 and we use random cropping and flipping as data augmentation during training.

Results: In Table 3.1 we compare multiple OOD methods: confidence of standard training (PLAIN) and OE, binary discriminator trained without a shared representation with a classifier (SEPARATE BINDISC), classifier with background class (BGC) and the combination of a plain classifier and a binary in-vs-out-distribution classifier with shared representation (SHARED COMBI). As described in Section 3.2, both BGC and SHARED COMBI can be used in combination with different scoring functions. For BGC, we evaluate all three scoring functions s_1 , s_2 and s_3 and for SHARED COMBI we only use s_2 and s_3 as s_1 is equivalent to $p(i|x)$ which is the output of SHARED BINDISC. Additionally, we evaluate OOD detection based on the confidence of the shared classifier (SHARED CLASSI) trained together with SHARED BINDISC.

For CIFAR10 and RImgNet, a first interesting observation is that SHARED CLASSI has

¹https://github.com/j-cb/Breaking_Down_OOD_Detection

remarkably good OOD detection performance; significantly better than a normal classifier (plain) even though it is just trained using normal cross-entropy loss and so the OOD performance is only due to the regularization enforced by the shared representation with SHARED BINDISC. Furthermore, on both CIFAR10 and RImgNet SHARED BINDISC already has good OOD detection performance with mean AUCs of 92.51 and 98.26 respectively, which is further improved by considering scoring functions s_2/s_3 in the combination of SHARED BINDISC and SHARED CLASSI. In all cases the SHARED COMBI models yield both good classification accuracy and mean AUC. Moreover, the results of the classifier with background class (BGC) (Thulasidasan *et al.*, 2021) are interesting. It works very well but the performance depends on the chosen scoring function. Whereas s_1 (output of the background class) is a usable scoring function (mean AUC: 95.02, 88.41, 98.61), the maximum probability over the other classes s_2 (mean AUC: 97.22, 90.47, 98.54) or the combination in terms of s_3 (mean AUC: 97.21, 90.46, 98.66) perform better. In total with the scoring function s_2/s_3 integrating classifier and discriminative information, BGC reaches similar performance to OE (which implicitly also uses s_3 as scoring function).

In general, the differences of the methods are relatively minor both in terms of OOD detection and classification accuracy, where the integration of OOD information generally helps classification accuracy compared to the plain classifier. This is most likely explained by better learned representations, see also Hendrycks *et al.* (2019a); Augustin *et al.* (2020) for similar observations. Overall, as suggested by the theoretical results on the equivalence of the Bayes optimal classifier of OE with the s_3 scoring function of BGC and SHARED COMBI, we observe that even though these methods are derived and in particular trained with quite different objectives, they behave very similarly in our experiments. In total we think that this provides a much better understanding of where differences of OOD methods are coming from. Regarding the question of which method and scoring function should be used for a given application, the experimental results across datasets suggest that their difference is minor and there is no clear best choice.

3.5 Conclusion

In this chapter we have analyzed different ways of utilizing training OOD data for the task of OOD detection of unseen out-distributions at test times. We have introduced a notion of equivalence of scoring functions for OOD detection and theoretically showed that various OOD detection methods can unexpectedly be seen as equivalent to binary discrimination between in- and out-distribution in the limit of infinite data. We empirically compared various methods and showed that, as long as shared representations are used for classification and OOD detector, various scoring functions lead to similar OOD detection performance on average. In particular, no method consistently outperforms the method of using the confidence score of an outlier exposure model that we used in Chapter 2.

Recent developments: Since our very recent publication of (Bitterwolf *et al.*, 2022), there have not been many significant developments in the theory of OOD detection. The authors of Kristiadi *et al.* (2022) compared different ways of incorporating OOD data from Bayesian perspective and also found OE to perform well. In Fang *et al.* (2022) they formalize the ways in which OOD detection is (un)learnable from a theoretical point of view.

Part II

Adversarial Out-of-Distribution Detection

Chapter 4

Adversarial robustness on in-and out-distribution improves explainability

This chapter is based on (Augustin *et al.*, 2020) which we presented at ECCV 2020. Matthias Hein and I jointly came up with the idea of using adversarially robust OOD detection as a mechanism for feature generation based on previous experiments by me. Maximilian Augustin and I jointly worked on experiments, exploring different ways of combining adversarial robustness on in- and out-distribution as well as on reliably evaluating OOD robustness. Ultimately, Maximilian Augustin’s scientific ideas and experimental work were more impactful for this paper than my own. In particular, he developed the final implementation of the training schedules as well as of the counterfactual generation. Matthias Hein provided guidance to the project and significantly helped with the writing.

4.1 Introduction

In the previous chapters we benchmarked a number of different OOD detection techniques on various image classification datasets. The two main takeaways were that

1. assuming access to a large and diverse training out-distribution helps with the task of OOD detection and that
2. using the confidence score of an OE trained model performs no worse than any of the other baselines that we benchmarked.

Despite this, unsurprisingly, OE models are not robust to adversarial perturbations - neither for classification nor for the detection of adversarially perturbed OOD samples. As mentioned in Section 1.3, Adversarial Training (AT) is known to mitigate the former. However, this dissertation is primarily concerned with the latter issue, i.e. the adversarially robust detection of OOD data. Here, Adversarial Confidence Enhanced Training (ACET) as proposed by Hein *et al.* (2019) enforces low confidence in a neighborhood around OOD samples and can be seen as a form of adversarial training on the out-distribution. ACET leads improved OOD detection performance in an adversarial setting and suffers from a smaller loss in clean accuracy compared to AT. Unfortunately, ACET suffers from some significant drawbacks: i) its training

is relatively unstable compared to adversarial training, sometimes leading to models with no adversarial robustness on OOD data whatsoever. ii) Similarly to AT there is a loss in clean accuracy and iii) even when successfully trained, there are no guarantees that the model is as robust as its empirical evaluation implies. iv) The training of ACET (like AT) is over an order of magnitude more expensive than plain training.

In this chapter we will show how to solve issue i) and somewhat mitigate issue ii), while we leave the last two problems to the later chapters of this dissertation. Concretely, we show that combining AT and ACET into a method that we call RATIO (Robustness via Adversarial Training on In- and Out-distribution) inherits the good properties of adversarial training and ACET with significantly reduced negative effects, e.g. we get SOTA l_2 -robustness on CIFAR10 and have better clean accuracy than AT. Crucially, we empirically find that the training instabilities that ACET faces get resolved by this combination with AT. On top of this we get reliable confidence estimates on the out-distribution even in a worst-case scenario. In particular, AT yields highly overconfident predictions on out-distribution images in the absence of class specific features, whereas RATIO only yields highly confident predictions if recognizable features are present. We will also show that this desirable property can be used to produce high-quality visual counterfactual explanations, which demonstrates the utility of achieving adversarially robust low confidence on perturbed OOD samples. In summary, RATIO achieves high clean accuracy, is robust, calibrated and has generative properties which can be used to produce high-quality visual counterfactual explanations.

4.2 RATIO: Robust, Reliable and Explainable Classifier

An ideal model for classification is accurate and calibrated on the in-distribution, reliably has low confidence on out-distribution inputs, is robust to adversarial manipulation and has explainable decisions. To our knowledge there is no model which claims to have all these properties. The closest one we are aware of is the JEM-0 of (Grathwohl *et al.*, 2020) which claims to be robust, detects out-of-distribution samples and has generative properties. They state “JEM does not confidently classify nonsensical images, so instead, [...] natural image properties visibly emerge”. We show that RATIO gets us closer to this ultimate goal and outperforms JEM-0 in all aspects: accuracy, robustness, (worst-case) out-of-distribution detection, and visual counterfactual explanations.

4.2.1 In-Distribution Robustness and Adversarial Training

Recall from Section 1.3 that, typically, adversarial robustness is defined in terms of l_p -based threat models, i.e.

$$\mathcal{T}(x) = B_p(x, \varepsilon) = \{x' \in \mathcal{X} \mid \|x' - x\|_p \leq \varepsilon\}. \quad (4.1)$$

For convenience we also restate the objective of adversarial training for a threat model $\mathcal{T}(x)$:

$$\min_f \mathbb{E}_{(x,y) \sim p_{\text{in}}(x,y)} \left[\max_{x' \in \mathcal{T}(x)} \mathcal{L}_{\text{CE}}(f(x'), \mathbf{e}_y) \right], \quad (4.2)$$

where $p_{\text{in}}(x,y)$ is the training distribution. The community has put emphasis on robustness wrt. l_∞ but there has also been interest in other threat models e.g. l_2 -balls (Tramèr and Boneh, 2019; Rony *et al.*, 2019; Santurkar *et al.*, 2019). In particular, it has been noted that robust models wrt. an l_2 -ball have the property that “adversarial” samples generated within a sufficiently large l_2 -ball tend to have image features of the predicted class (Tsipras *et al.*, 2019; Santurkar *et al.*, 2019; Engstrom *et al.*, 2019a). Thus, they are not really “adversarial” samples in the sense that the true class has changed or is at least ambiguous. Because of these interesting generative properties, we focus on l_2 -based threat models in this chapter.

4.2.2 Out-Distribution Robustness and Adversarial Training on OOD

In the previous chapters, we already investigated the use of a classifier’s confidence as a feature for out-of-distribution detection. However, neural networks are known to have overly high confidence on adversarially perturbed OOD samples (Schott *et al.*, 2019; Hein *et al.*, 2019; Sehwag *et al.*, 2019; Meinke and Hein, 2020). In this section, we finally formalize the notion of an OOD detector being robust via the so-called Worst-Case AUC (WCAUC).¹ This will allow a rigorous comparison of the adversarial OOD detection performances of different methods.

The WCAUC is defined as the minimal AUC one can achieve if each out-distribution sample is allowed to be perturbed to reach maximal confidence within a certain threat model, which in our case is an l_∞ -ball of radius ε . Technically, we will use a slightly modified definition of the AUC than the one used in the previous chapters, in that we remove the equality term. This makes the AUC asymmetric and has been called the “conservative AUC” in Bitterwolf *et al.* (2020). The reason for this modification is that otherwise a trivially constant model would obtain a better worst-case AUC than many other models. This almost never makes any difference in the values one obtains for clean AUCs which is why from here on out, as a slight abuse of notation, we will just call the conservative AUC the AUC. Formally, the AUC and WCAUC of a feature $h : \mathcal{X} \rightarrow \mathbb{R}$ are defined as:

$$\text{AUC}_h(p_1, p_2) = \mathbb{E}_{\substack{x \sim p_1 \\ z \sim p_2}} [\mathbb{1}_{h(x) > h(z)}], \quad \text{WCAUC}_h(p_1, p_2) = \mathbb{E}_{\substack{x \sim p_1 \\ z \sim p_2}} \left[\mathbb{1}_{h(x) > \max_{\|z' - z\|_\infty \leq \varepsilon} h(z')} \right], \quad (4.3)$$

where p_1, p_2 are in- and out-distribution respectively and the indicator function $\mathbb{1}$ returns 1 if the expression in its argument is true and 0 otherwise. Note that an alternative formulation of a worst-case AUC as the worst-case across all samples from the out-distribution would turn out to be uninteresting, since it would necessarily be close to zero even if only a single sample

¹This measure was already used but not named in Hein *et al.* (2019) and in Meinke and Hein (2020).

gets assigned high-confidence, so we do not consider this notion.

Similarly to the robust accuracy, the exact evaluation of the WCAUC is computationally infeasible. Instead we can find an upper bound on the WCAUC by numerically maximizing the confidence using an adversarial attack inside the l_∞ -ball. We call our empirical lower bound the Adversarial AUC (AAUC) and we discuss its computation in Section 4.4.

In order to actually achieve high AAUCs, Hein *et al.* (2019) proposed Adversarial Confidence Enhanced Training (ACET) which enforces low confidence in a neighborhood around the out-distribution samples which can be seen as a form of AT on the out-distribution:

$$\min_f \mathbb{E}_{x,y \sim p_{\text{in}}} \left[\mathcal{L}_{\text{CE}}(f(x), \mathbf{e}_y) \right] + \lambda \mathbb{E}_{z \sim p_{\text{out}}} \left[\max_{\|z' - z\|_2 \leq \varepsilon} \mathcal{L}_{\text{CE}}(f(z'), \mathbf{1}/K) \right], \quad (4.4)$$

where $\mathbf{1}$ is the vector of all ones (outlier exposure (Hendrycks *et al.*, 2019a) has the same objective without the inner maximization for the out-distribution). Different from (Hein *et al.*, 2019) and the version of ACET benchmarked in Chapter 2, we use the same loss for in-and out-distribution, whereas they used the maximal log-confidence over all classes as loss for the out-distribution. In our experience the maximal (log-)confidence is more difficult to optimize, but both losses are minimized by the uniform distribution over the labels. Thus, the difference is rather small and we also denote this version as ACET.

4.2.3 RATIO: Robustness via Adversarial Training on In-and Out-distribution

We propose RATIO: adversarial training on in-and out-distribution. This combination leads to synergy effects where most positive attributes of AT and ACET are fused without having larger drawbacks. The objective of RATIO is given by:

$$\min_f \mathbb{E}_{x,y \sim p_{\text{in}}} \left[\max_{\|x' - x\|_2 \leq \varepsilon_i} \mathcal{L}_{\text{CE}}(f(x'), \mathbf{e}_y) \right] + \lambda \mathbb{E}_{z \sim p_{\text{out}}} \left[\max_{\|z' - z\|_2 \leq \varepsilon_o} \mathcal{L}_{\text{CE}}(f(z'), \mathbf{1}/K) \right], \quad (4.5)$$

where λ has the interpretation of $\frac{p_o}{p_i}$, the probability to see out-distribution p_o and in-distribution p_i samples at test time. Here we have specified an l_2 -threat model for in-and out-distribution but the objective can be adapted to different threat models which could be different for in-and out-distribution. The surprising part of RATIO is that the addition of the out-distribution part can improve the results even on the in-distribution in terms of (robust) accuracy. We hypothesize that the reason is that adversarial training on the out-distribution ensures that non-robust features do not change the confidence of the classifier. This behavior generalizes to the in-distribution and thus ACET (adversarial training on the out-distribution) is also robust on the in-distribution (52.3% robust accuracy for l_2 with $\varepsilon = 0.5$ on CIFAR10). One problem of adversarial training is overfitting on the training set (Rice *et al.*, 2020). Our RATIO has seen more images at training time and while the direct goal is distinct (keeping one-hot prediction on the in-distribution and uniform prediction on out-distribution) both aim at constant behavior of the classifier over the l_2 -ball and thus the effectively increased training size

improves generalization (in contrast to AT, RATIO has its peak robustness at the end of the training). Moreover, RATIO typically only shows high confidence if class-specific features have appeared which we use in the generative process described next.

4.3 Visual Counterfactual Explanations

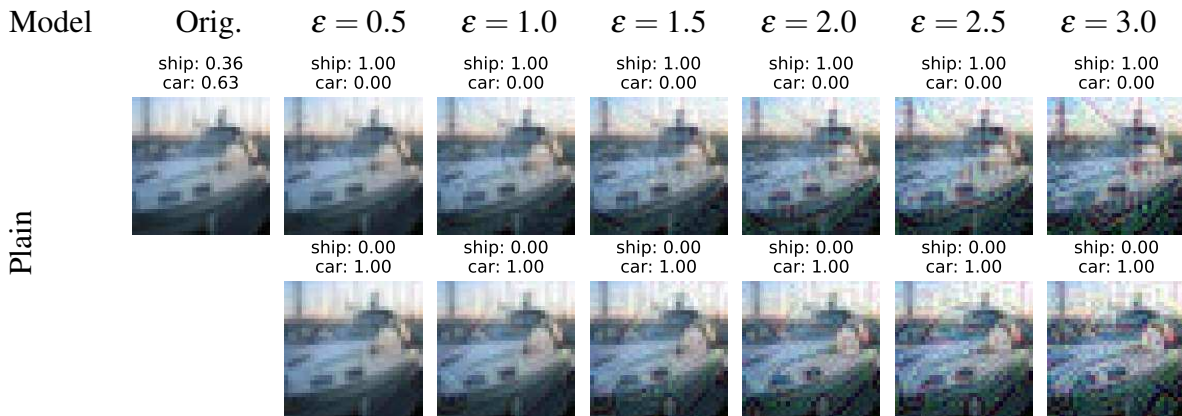


Figure 4.1: Failure of a visual counterfactual for a plain model. The targeted attack immediately produces very high confidence in both classes but instead of class features only high-frequency noise appears because plain models are not robust.

Counterfactual explanations have been proposed in (Wachter *et al.*, 2018) as a tool for making classifier decisions plausible, since humans can also justify decisions via counterfactuals “I would have decided for X, if Y had been true” (Miller, 2019). Other forms are explanations based on image features (Hendricks *et al.*, 2016, 2018). However, changing the decision for image classification in *image space* for non-robust models leads to adversarial samples (Dong *et al.*, 2017) with changes that are visually not meaningful. Thus visual counterfactuals are often based on generative models or restrictions on the space of image manipulation (Samanogouei *et al.*, 2018; Álvaro Parafita and Vitrià, 2019; Chang *et al.*, 2019; Goyal *et al.*, 2019; Zhu *et al.*, 2016; Wang *et al.*, 2018). Robust models wrt. l_2 -adversarial attacks (Tsipras *et al.*, 2019; Santurkar *et al.*, 2019) have been shown to change their decision when class-specific features appear in the image, which is a prerequisite for meaningful counterfactuals (Barocas *et al.*, 2020). RATIO generates better counterfactuals, i.e. the confidence of the counterfactual images obtained by an l_2 -adversarial attack tends to be high only after features of the alternative class have appeared. Especially for out-distribution images the difference to AT is pronounced.

Compared to sensitivity based explanations (Baehrens *et al.*, 2010; Zeiler and Fergus, 2014) or explanations based on feature attributions (Bach *et al.*, 2015) counterfactual explanations have the advantage that the explanation is directly to the decision of the classifier. On the

other hand the counterfactual explanation requires us to specify a metric and a budget for the allowed change of the image which can be done directly in image space or in the latent space of a generative model. However, our goal is that the classifier directly learns what meaningful changes are and we do not want to impose that via a generative model. Thus, we aim at visual counterfactual explanations directly in image space with a fixed budget for changing the image. As the decision changes, features of this class should appear in the image (see Figure 4.2). Normally trained models will not achieve this since non-robust models change their prediction for non-perceptible perturbations (Szegedy *et al.*, 2014), see Figure 4.1. Thus robustness against (l_2 -)adversarial perturbations is a necessary requirement for visual counterfactuals.

A *visual counterfactual* for the original point x classified as $c = \arg \max_{k=1, \dots, K} f_k(x)$, a target class $t \in \{1, \dots, K\}$ and a budget ε is defined as

$$x^{(t)} = \arg \max_{x' \in [0,1]^d, \|x-x'\|_2 \leq \varepsilon} \hat{p}_{f,t}(x'), \quad (4.6)$$

where $\hat{p}_{f,t}(z)$ is the confidence for class t of our classifier for the image z . If $t \neq c$ it answers the counterfactual question of how to use the given budget to change the original input x so that the classifier is most confident in class t . Note that in our definition we include the case where $t = c$, that is we ask how to change the input x classified as c to get even more confident in class c . In Figure 4.2 we illustrate both directions and show how for robust models class specific image features appear when optimizing the confidence of that class. This shows that the optimization of visual counterfactuals can be done directly in image space.

4.4 Experiments

We validate our approach on SVHN (Netzer *et al.*, 2011), CIFAR10/100 (Krizhevsky and Hinton, 2009) and restricted ImageNet (Santurkar *et al.*, 2019). The code can be found online.²

4.4.1 Training

On CIFAR10 we compare RATIO to a pretrained JEM-0 (Grathwohl *et al.*, 2020) and the AT model (Engstrom *et al.*, 2019c) with $\varepsilon = 0.5$ ($M_{0.5}$) (both not available on the other datasets). As an ablation study of RATIO we train a plain model, outlier exposure (OE) (Hendrycks *et al.*, 2019a), ACET (Hein *et al.*, 2019) and AT with $\varepsilon = 0.5$ ($AT_{0.5}$) and $\varepsilon = 0.25$ ($AT_{0.25}$), using the same hyperparameters as for our RATIO training. As out-distribution for SVHN and CIFAR we use 80 million tiny images (Torralba *et al.*, 2008)³ as suggested in (Hendrycks *et al.*, 2019a) and for restricted ImageNet, again, the remaining ImageNet classes.

For our experiments on CIFAR10/100 (Krizhevsky and Hinton, 2009) we use a standard ResNet50 architecture and SGD with Nesterov momentum ($\beta = 0.9$) and a base learning rate

²<https://github.com/M4xim41/InN0utRobustness>

³Note that this work was carried out before the retraction of 80 million tiny images.

Table 4.1: *Summary:* We show clean and robust accuracy in an l_2 -threat model with $\epsilon = 0.5$ and the expected calibration error (ECE). For OOD detection we report the mean of clean and worst case AUC over several out-distributions in an l_2 -threat model with $\epsilon = 1.0$ as well as the mean maximal confidence (MMC) on the out-distributions. In light red we highlight failure cases for certain metrics. Only RATIO-0.25 ($R_{0.25}$) has good performance across all metrics.

CIFAR10	Plain	OE	ACET	$M_{0.5}$	$AT_{0.5}$	$AT_{0.25}$	JEM-0	$R_{0.5}$	$R_{0.25}$
Acc. \uparrow	96.2	96.4	94.1	90.8	90.8	94.0	92.8	91.1	93.5
R. Acc. $_{0.5}$ \uparrow	0.0	0.0	52.3	69.3	70.4	65.0	40.5	73.3	70.5
ECE in % \downarrow	1.0	2.9	2.8	2.6	2.2	2.2	3.9	2.8	2.7
AUC \uparrow	94.2	96.5	94.7	81.8	88.9	92.7	75.0	95.6	95.0
AAUC $_{1.0}$ \uparrow	1.6	8.7	81.9	48.5	57.4	42.0	14.6	83.6	84.3
MMC \downarrow	62.0	31.9	39.1	62.7	55.8	55.2	69.7	31.9	33.9
CIFAR100	Plain	OE	ACET	$AT_{0.5}$	$AT_{0.25}$	$R_{0.5}$	$R_{0.25}$		
Acc. \uparrow	81.5	81.4	-	70.6	75.8	69.2	74.4		
R. Acc. $_{0.5}$ \uparrow	0.0	0.0	-	43.2	37.3	45.6	42.4		
ECE \downarrow	1.2	7.2	-	1.3	1.5	3.2	2.0		
AUC \uparrow	84.0	91.9	-	75.6	79.4	87.0	86.9		
AAUC $_{1.0}$ \uparrow	0.4	14.6	-	29.9	24.8	55.5	54.5		
MMC \downarrow	51.1	21.8	-	45.8	47.1	24.4	31.0		
SVHN	Plain	OE	ACET	$AT_{0.5}$	$AT_{0.25}$	$R_{0.5}$	$R_{0.25}$		
Acc. \uparrow	97.3	97.6	97.8	94.4	96.7	94.3	96.8		
R. Acc. $_{0.5}$ \uparrow	0.9	0.3	28.8	68.1	63.0	68.4	64.8		
ECE in % \downarrow	0.9	0.9	1.6	1.6	0.8	2.0	1.8		
AUC \uparrow	96.9	99.6	99.8	91.0	97.0	99.8	99.9		
AAUC $_{1.0}$ \uparrow	8.5	18.2	96.0	51.1	48.3	97.5	97.5		
MMC \downarrow	61.5	16.3	11.8	67.1	49.1	12.1	11.1		
R.Imagenet	Plain	OE	ACET	$M_{3.5}$	$AT_{3.5}$	$AT_{1.75}$	$R_{3.5}$	$R_{1.75}$	
Acc. \uparrow	96.6	97.2	96.2	90.3	93.5	95.5	93.9	95.5	
R. Acc. $_{3.5}$ \uparrow	0.0	0.0	6.2	47.7	47.7	36.7	49.2	43.0	
ECE \downarrow	0.6	1.8	0.9	0.7	0.9	0.5	0.3	0.7	
AUC \uparrow	92.7	98.9	97.74	83.6	84.3	86.5	97.2	97.8	
AAUC $_{7.0}$ \uparrow	0.0	1.8	87.54	44.2	37.5	16.3	90.9	90.6	
MMC \downarrow	67.9	20.6	34.85	69.2	75.2	81.8	33.6	32.3	

of 0.1 and weight decay of $5e - 4$. Our training schedule spans 220 epochs and we decrease the learning rate by a factor of 10 in epochs 100, 150 and 200. As data augmentation for all our trained CIFAR10 models we use the recommended AutoAugment policy from (Cubuk *et al.*, 2019), including Cutout (DeVries and Taylor, 2017). For SVHN, we use a ResNet18 architecture with a 100 epochs schedule which decreases the learning rate in epochs 50, 75 and 90. The data augmentation scheme consists of input normalization, random cropping and

Cutout.

On restricted ImageNet we adopt the overall training scheme from (Santurkar *et al.*, 2019), including the ResNet50 architecture and data augmentation with a slightly shorter 75 epoch LR schedule which decays the initial LR of 0.1 at epochs 30, 60 and 75 by a factor of 10. We also tested the AutoAugment ImageNet policy, however, found that it performed worse than the simpler transform based on random crops, flips, color jitter and a lighting transformation. Similarly to Chapter 3, we use all the remaining classes from ILSVRC2012 as training out-distribution for OE, RATIO and ACET training.

For adversarial and RATIO training, we use 100% adversarial training on the train distribution, i.e. the model only sees perturbed samples during training. Instead of solving the robust min-max formulation in Eq. (4.2) directly, we use the logits-based loss from (Carlini and Wagner, 2017b) in the inner maximization problem, i.e. for a training sample (x, y) we approximately solve:

$$\max_{x' \in \mathcal{T}(x)} \max_{i \neq y} f_i(x') - f_y(x'). \quad (4.7)$$

To compute z , we use a 7-step PGD with the l_2 -normalized gradient with step size 0.1 and momentum weight 0.9 which returns the point with the highest loss across its trajectory. This deviation from the standard adversarial training scheme of (Madry *et al.*, 2018) is justified by our empirical experience that for this small number of steps the optimization of the logits-based loss even leads to higher cross-entropy loss than optimizing the cross-entropy loss directly. Note that for the actual update of the model we use the gradient of the cross-entropy loss.

The same scheme applies to the inner maximization problem for the adversarial training on the out-distribution (cross-entropy loss to uniform distribution, see also (4.4)) in ACET and RATIO training, where we again use PGD with momentum and a step size of 0.1. We again emphasize that unlike (Hein and Andriushchenko, 2017) who used a smoothed form of noise as out-distribution, we use 80 Million Tiny Images which makes ACET resp. the adversarial training on the out-distribution a substantially harder task. As the radius of the l_2 -threat model on the out-distribution is significantly larger than on the in-distribution we increase the initial number of iterations to 20. For pure ACET training we noticed that even a 20-step attack is often too weak to find an approximate maximum of the inner maximization problem which results in the model gradually becoming less robust. We therefore incrementally increase the number of ACET iterations to 40 by adding 5 steps for each update of the learning rate. However even with those adjustments, pure ACET training on CIFAR10 remains very unstable and reproducibly ends up with a maximum-mean confidence close to 0.1 for CIFAR10 test samples. We therefore use a smaller ResNet18 with a 100 epoch schedule for all ACET experiments where this training scheme can be used without problems. On CIFAR100, we did not successfully train ACET and thus omit the results. We note that RATIO does not suffer from ACET’s stability problems and in this setting the training reliably works.

The threat models are l_2 -balls of radius 0.25 resp. 0.5 on the in-distribution and a l_2 -ball of radius 1.0 on the out-distribution. We use a batch size of 128 for plain and adversarial training and a total batch size of 256 for OE, ACET and RATIO training, i.e. 128 samples from the in-

and 128 samples from the out-distribution.

Only on restricted ImageNet, we noticed an unexpected drop in clean accuracy on the larger RATIO models and therefore we add an additional clean in-distribution loss to the RATIO and AT models. In detail, adversarial training uses a 50/50 scheme with 128 standard and 128 perturbed samples per batch while RATIO uses 128 clean and 128 perturbed samples from the in-distribution and 128 perturbed samples from the out distribution, resulting in a total batch-size of 384. Such a scheme typically improves clean accuracy while reducing the robustness, however we note that our 50/50 AT models are able to compete with Madry’s standard AT model (100% adversarial training) and are thus a fair baseline for RATIO. Also, due to computational complexity, we use a simple 10 step PGD with a stepsize of 1.0 on the out-distribution for Restricted ImageNet.

As adversarial training is prone to overfitting on the training set (Rice *et al.*, 2020), resulting in a loss in robust accuracy on the test set in the last epochs of training, we use the robust accuracy on the test set under the 7 step PGD attack as early stopping criterion (note that the 7-step PGD attack is significantly weaker than what we use later on for evaluation of robustness).

4.4.2 Calibration on the in-distribution

With RATIO we aim for reliable confidence estimates, in particular no overconfident predictions. In order to have comparable confidences for the different models we train, especially when we check visual counterfactuals or feature generation, we first need to “align” their confidences. We do this by minimizing the expected calibration error (ECE) via temperature rescaling (Guo *et al.*, 2017). As explained in Section 1.2.1, this rescaling does not change the classification and thus has no impact on (robust) accuracy and only a minor influence on the (adversarial) AUC values for OOD-detection. For computing the ECE we use 10 bins (note that validation sets are smaller than the test set) and for the evaluation of the final calibration on the test set we use 15 bins. For the finding the temperature we pick 500 geometrically spaced temperatures on the interval $T \in [0.05, 2.71]$ and choose the minimizer for each model. Since $M_{0.5}$ and JEM-0 have used the entire training set and removing data from the test set would make the accuracy values harder to compare, we use the CIFAR10.1 dataset (Recht *et al.*, 2018) for calibration on CIFAR10. On SVHN we use 2000 points from the unused additional data, on CIFAR100 the first 2000 test points and on R.Imagenet we use a random subset of 2000 test points as our validation set. For CIFAR100 and R.Imagenet we omit these 2000 points when testing the OOD performance.

4.4.3 (Robust) Accuracy on the in-distribution

For the adversarial attacks on in- and out-distribution we use Auto-Attack (Croce and Hein, 2020b) which is an ensemble of four attacks, including the black-box Square Attack (Andriushchenko *et al.*, 2020) and three white-box attacks (FAB-attack (Croce and Hein, 2020a) and AUTO-PGD with different losses). For each of the white-box attacks, a budget of 100

iterations and 5 restarts is used and a query limit of 5,000 for Square attack. In Croce and Hein (2020b) they show that Auto-Attack consistently improves the robustness evaluation for a large number of models (including JEM-x).

Using Auto-Attack we evaluate robustness on the full test set for both CIFAR and R. Imagenet and 10,000 test samples for SVHN. Table 4.1 contains the robust l_2 accuracy. On CIFAR10, RATIO achieves significantly higher robust accuracy than AT for l_2 - and l_∞ -attacks. Thus the additional adversarial training on the out-distribution with radius $\epsilon_o = 1$ boosts the robustness on the in-distribution. In particular, $\text{RATIO}_{0.25}$ achieves better l_2 -robustness than $\text{AT}_{0.5}$ and $\text{M}_{0.5}$ at $\approx 2.7\%$ higher clean accuracy. In addition, $\text{R}_{0.5}$ yields new state-of-the-art l_2 -robust accuracy at radius 0.5 (see (Croce and Hein, 2020b) for a benchmark) while having higher test accuracy than $\text{AT}_{0.5}$, $\text{M}_{0.5}$. Interestingly, although ACET is not designed to yield adversarial robustness on the in-distribution, it achieves more than 50% robust accuracy for $\epsilon = 0.5$ and outperforms JEM-0 in all benchmarks. However, as our goal is to have a model which is both robust and accurate, we recommend to use $\text{R}_{0.25}$ for CIFAR10 which has a drop of only 2.6% in test accuracy compared to a plain model while having similar robustness to $\text{M}_{0.5}$ and $\text{AT}_{0.5}$. Similar observations as for CIFAR10 hold for CIFAR100 and for Restricted ImageNet even though for CIFAR100 AT and RATIO suffer a higher loss in accuracy. On SVHN, RATIO outperforms AT in terms of robust accuracy trained with the same l_2 -radius but the effect is less than for CIFAR10. We believe that this is due to the fact that the images obtained from the 80 million tiny image dataset (out-distribution) do not reflect the specific structure of SVHN numbers which makes (adversarial) outlier detection an easier task. This is supported by the fact that ACET achieves better clean accuracy on SVHN than both OE and the plain model while it has worse clean accuracy on CIFAR10.

4.4.4 Visual Counterfactual Generation

We use 500 step Auto-PGD (Croce and Hein, 2020b) for a targeted attack with the objective in (4.6). However, note that this non-convex optimization problem has been shown to be NP-hard (Katz *et al.*, 2017). In Figure 4.2, 4.3 and 4.5 we show generated counterfactuals for all datasets. For CIFAR10 $\text{AT}_{0.5}$ performs very similarly to $\text{RATIO}_{0.25}$ in terms of the emergence of class specific image features. In particular, we often see the appearance of characteristic features such as pointed ears for cats, wheels for cars and trucks, large eyes for both cats and dogs and the antlers for deers. JEM-0 and ACET perform worse but for both of them one observes the appearance of image features. However, particularly the images of JEM-0 have a lot of artefacts. For SVHN, on average, $\text{RATIO}_{0.25}$ performs better than $\text{AT}_{0.25}$ and ACET. It is interesting to note that for both datasets class-specific features already emerge for an l_2 -radius of 1.0. Thus it seems questionable if l_2 -adversarial robustness beyond a radius of 1.0 should be enforced. Due to the larger number of classes, CIFAR100 counterfactuals are of slightly lower quality. For Restricted ImageNet the visual counterfactuals show class-specific features but can often be identified as synthetic due to misaligned features.

4.4.5 Reliable Detection of (Adversarial) Out-of-Distribution Images

We report the adversarial AUC by maximizing the confidence in an l_2 -ball of radius 1.0 (resp. 7.0 for R. ImageNet) around OOD images via Auto-PGD (Croce and Hein, 2020b) with 100 steps and 5 random restarts. Due to computational constraints, instead of using the entire OOD test sets, we use 1024 points from each out-distribution (300 points for LSUN_CR).

The average AUC over all OOD datasets is reported in Tables 4.1. The detailed results are shown in Table 4.2 and Table 4.3. The AT-model of Madry et. al ($M_{0.5}$) perform worse than the plain model even on the average case task. However, we see that with our more aggressive data augmentation this problem is somewhat alleviated ($AT_{0.5}$ and $AT_{0.25}$). As expected ACET, has good worst-case OOD performance but is similar to the plain model for the average case. JEM-0 has bad worst-case AUCs and we cannot confirm the claim that “JEM does not confidently classify nonsensical images” (Grathwohl *et al.*, 2020). Also as expected, OE has state-of-the-art performance on the clean task but has no robustness on the out-distribution, so it fails completely in this regime. Our RATIO models show strong performance on all tasks and even outperform the ACET model which shows that adversarial robustness wrt the in-distribution also helps with adversarial robustness on the out-distribution. On SVHN the average case OOD task is simple enough that several models achieve near perfect AUCs, but again only ACET and our RATIO models manage to retain strong performance in the adversarial setting. The worst-case AUC of AT models is significantly worse than that of ACET and RATIO.

4.4.6 Feature Generation on OOD images

Finally, we test the abilities to generate image features with a targeted attack on OOD images (taken from 80m tiny image dataset resp. ImageNet classes not belonging to R. ImageNet). The setting is similar to the visual counterfactuals. We take some OOD image and then optimize the confidence in the class which is predicted on the OOD image. The results can be found in Figure 4.4 and 4.6. For CIFAR10 all methods are able to generate image features of the class but the predicted confidences are only reasonable for ACET and $RATIO_{0.25}$ whereas $AT_{0.5}$ and JEM-0 are overconfident when no strong class features are visible. This observation generalizes to SVHN and mostly CIFAR100 and restricted Imagenet, i.e. RATIO generally has the best OOD-confidence profile.

4.5 Conclusion

We have shown that adversarial robustness on in-distribution and out-distribution (as a proxy of all natural images) gets us closer to a classifier which is accurate, robust, has reliable confidence estimates and is able to produce visual counterfactual explanations with strong class specific image features. For the usage in safety-critical in systems, it is desirable to achieve these robustness properties in a provable way, which we will explore in the upcoming chapters.

Recent developments: Since the paper’s publication, the finding that enforcing adversarially robust low confidence on OOD data can be stably trained by combining it with adversarial training has been confirmed by (Chen *et al.*, 2022; Lee *et al.*, 2021a), which effectively independently rediscovered RATIO. Furthermore, the generative capabilities of adversarially robust models have been explored further in the context of counterfactual explainability (Boreiko *et al.*, 2022a) and even applied to the medical domain (Boreiko *et al.*, 2022b; Augustin *et al.*, 2022). EBMs have also been improved to have more stable training and higher adversarial robustness on OOD data but have nonetheless failed to outperform RATIO (Yin *et al.*, 2022). The concept of the adversarial AUC has been further extended by Azizmalayeri *et al.* (2022), where they allow in- and out-distribution points to be perturbed simultaneously.

Table 4.2: *OOD performance (CIFAR10, CIFAR100)*: The area under the ROC curve (AUC) on the binary task of separating the in- from the out-distribution based on the confidence. For each dataset the first table shows the average-case AUC and the second ones show the worst-case AUC with a threat model $l_2 = 1.0$ around the out-distribution samples.

CIFAR10									
Av. Case	Plain	OE	ACET	M _{0.5}	AT _{0.5}	AT _{0.25}	JEM-0	R _{0.5}	R _{0.25}
SVHN	96.8	99.4	93.6	91.9	93.5	95.3	89.3	96.5	96.4
CIFAR100	91.6	91.4	90.4	84.3	85.3	89.1	87.6	90.8	91.6
LSUN_CR	95.6	99.6	98.2	89.7	90.7	92.5	91.6	98.0	98.3
Imagenet-Noise	91.6	89.8	91.0	84.8	85.9	89.0	86.7	90.5	91.3
Uni. Noise	94.3	99.3	95.0	93.7	94.9	95.5	83.1	97.8	97.6
Uni. Noise	95.0	99.5	99.9	46.1	83.2	94.6	11.8	99.9	99.9
Worst Case	Plain	OE	ACET	M _{0.5}	AT _{0.5}	AT _{0.25}	JEM-0	R _{0.5}	R _{0.25}
SVHN	0.0	0.6	76.1	57.1	62.0	40.1	7.3	81.3	81.1
CIFAR100	0.0	2.7	69.9	47.9	48.5	31.8	19.2	71.9	73.0
LSUN_CR	0.0	4.0	87.9	52.0	52.8	36.5	20.6	87.3	89.1
Imagenet-Noise	0.0	1.5	72.8	50.6	51.1	36.8	21.2	72.4	73.5
Uni. Noise	0.0	0.0	84.5	62.9	67.9	38.8	16.5	88.9	89.4
Uni. Noise	9.4	43.1	99.9	20.7	62.0	67.8	2.5	99.8	99.8

CIFAR100							
Av. Case	Plain	OE	ACET	AT _{0.5}	AT _{0.25}	R _{0.5}	R _{0.25}
SVHN	86.8	95.8	-	82.2	81.0	83.8	84.5
CIFAR10	81.1	84.3	-	73.0	76.0	71.9	73.2
LSUN_CR	83.1	97.5	-	81.0	80.4	93.6	91.7
Imagenet-Noise	83.9	86.2	-	74.3	77.7	79.6	81.0
Uni. Noise	85.9	87.6	-	84.8	82.3	93.0	91.1
Uni. Noise	73.2	99.7	-	58.5	78.7	99.8	99.6
Worst Case	Plain	OE	ACET	AT _{0.5}	AT _{0.25}	R _{0.5}	R _{0.25}
SVHN	0.0	5.6	-	30.2	20.6	41.4	42.6
CIFAR10	0.0	5.0	-	27.3	18.9	31.3	28.9
LSUN_CR	0.0	5.0	-	30.0	21.3	58.9	59.2
Imagenet-Noise	0.0	4.9	-	31.3	23.3	34.1	31.3
Uni. Noise	0.0	6.2	-	32.6	22.2	68.3	67.5
Uni. Noise	2.5	60.6	-	27.9	42.2	99.2	97.5

Table 4.3: *OOD performance (SVHN, R. ImageNet)*: The area under the ROC curve (AUC) on the binary task of separating the in- from the out-distribution based on the confidence. For each dataset the first table shows the average-case AUC and the second ones show the worst-case AUC with a threat model $l_2 = 1.0$ for SVHN and $l_2 = 7.0$ for R.ImageNet around the out-distribution samples.

		SVHN						
Av. Case	Plain	OE	ACET	AT _{0.5}	AT _{0.25}	R _{0.5}	R _{0.25}	
CIFAR10	95.8	100.0	100.0	88.5	95.7	100.0	100.0	
CIFAR100	95.6	100.0	100.0	87.8	95.5	100.0	100.0	
LSUN_CR	97.1	100.0	100.0	87.3	95.8	100.0	100.0	
Imagenet-	96.2	100.0	100.0	87.9	95.9	100.0	100.0	
Noise	97.2	97.8	99.2	97.5	99.2	99.1	99.5	
Uni. Noise	99.9	100.0	100.0	97.2	99.7	100.0	99.9	
Worst Case	Plain	OE	ACET	AT _{0.5}	AT _{0.25}	R _{0.5}	R _{0.25}	
CIFAR10	0.0	1.3	99.8	43.4	43.7	99.8	99.8	
CIFAR100	0.0	2.5	99.8	42.7	39.5	99.7	99.8	
LSUN_CR	0.0	1.0	99.8	36.6	42.4	99.9	99.9	
Imagenet-	0.0	4.3	99.8	39.4	43.7	99.9	99.9	
Noise	0.0	0.0	76.8	71.2	52.7	85.6	85.7	
Uni. Noise	51.1	99.9	99.9	73.0	67.7	99.9	99.9	
		R.ImageNet						
Av. Case	Plain	OE	ACET	M _{3.5}	AT _{3.5}	AT _{1.75}	R _{3.5}	R _{1.75}
Flowers	90.6	96.2	94.1	74.4	79.8	83.2	91.3	92.8
Food101	91.6	99.3	98.3	79.9	83.8	86.9	98.1	98.7
FGVC	89.6	99.7	98.8	79.8	80.8	81.5	98.8	99.1
Cars	93.9	99.9	99.9	83.5	86.3	90.0	99.8	99.9
Uni. Noise	98.7	99.6	100.0	95.9	88.2	87.4	100.0	100.0
Worst Case	Plain	OE	ACET	M _{3.5}	AT _{3.5}	AT _{1.75}	R _{3.5}	R _{1.75}
SVHN	0.0	1.8	83.4	32.2	35.9	16.1	86.6	86.5
CIFAR10	0.0	1.8	87.7	33.3	32.5	11.9	90.6	90.5
LSUN_CR	0.0	1.8	92.4	39.5	33.0	13.6	95.8	95.7
Imagenet-	0.0	1.8	94.0	44.7	50.0	31.8	97.9	97.9
Uni. Noise	0.0	1.8	100.0	83.6	43.3	16.1	100.0	99.9
















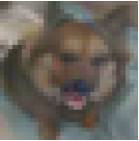
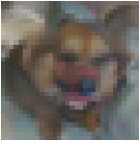










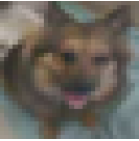


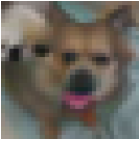









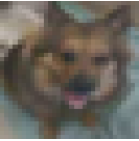
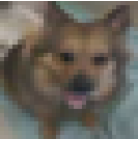
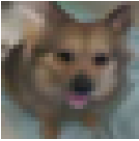









Model	Orig.	$\epsilon = 0.5$	$\epsilon = 1.0$	$\epsilon = 1.5$	$\epsilon = 2.0$	$\epsilon = 2.5$	$\epsilon = 3.0$
ACET	dog: 0.00 cat: 0.98	dog: 0.99 cat: 0.00	dog: 0.99 cat: 0.00	dog: 1.00 cat: 0.00	dog: 1.00 cat: 0.00	dog: 1.00 cat: 0.00	dog: 1.00 cat: 0.00
							
		dog: 0.00 cat: 0.99	dog: 0.00 cat: 1.00	dog: 0.00 cat: 1.00	dog: 0.00 cat: 1.00	dog: 0.00 cat: 1.00	dog: 0.00 cat: 1.00
							
JEM-0	dog: 0.04 cat: 0.71	dog: 0.99 cat: 0.00	dog: 1.00 cat: 0.00	dog: 1.00 cat: 0.00	dog: 1.00 cat: 0.00	dog: 1.00 cat: 0.00	dog: 1.00 cat: 0.00
							
		dog: 0.00 cat: 0.99	dog: 0.00 cat: 0.99	dog: 0.00 cat: 1.00	dog: 0.00 cat: 1.00	dog: 0.00 cat: 1.00	dog: 0.00 cat: 1.00
							
AT-0.50	dog: 0.05 frog: 0.66	dog: 0.46 frog: 0.03	dog: 0.89 frog: 0.01	dog: 0.99 frog: 0.00	dog: 1.00 frog: 0.00	dog: 1.00 frog: 0.00	dog: 1.00 frog: 0.00
							
		dog: 0.01 frog: 0.89	dog: 0.00 frog: 0.99	dog: 0.00 frog: 1.00	dog: 0.00 frog: 1.00	dog: 0.00 frog: 1.00	dog: 0.00 frog: 1.00
							
RATIO-0.25	dog: 0.01 cat: 0.95	dog: 0.97 cat: 0.01	dog: 1.00 cat: 0.00	dog: 1.00 cat: 0.00	dog: 1.00 cat: 0.00	dog: 1.00 cat: 0.00	dog: 1.00 cat: 0.00
							
		dog: 0.00 cat: 0.99	dog: 0.00 cat: 1.00	dog: 0.00 cat: 1.00	dog: 0.00 cat: 1.00	dog: 0.00 cat: 1.00	dog: 0.00 cat: 1.00
							

Figure 4.2: **Visual Counterfactuals (CIFAR10)**: The dog image on the left is misclassified by all models (confidence for true and predicted class are shown). The top row shows visual counterfactuals for the correct class (how to change the image so that it is classified as dog) and the bottom row shows how to change the image in order to increase the confidence in the wrong prediction for different budgets of the l_2 -radius ($\epsilon = 0.5$ to $\epsilon = 3$).

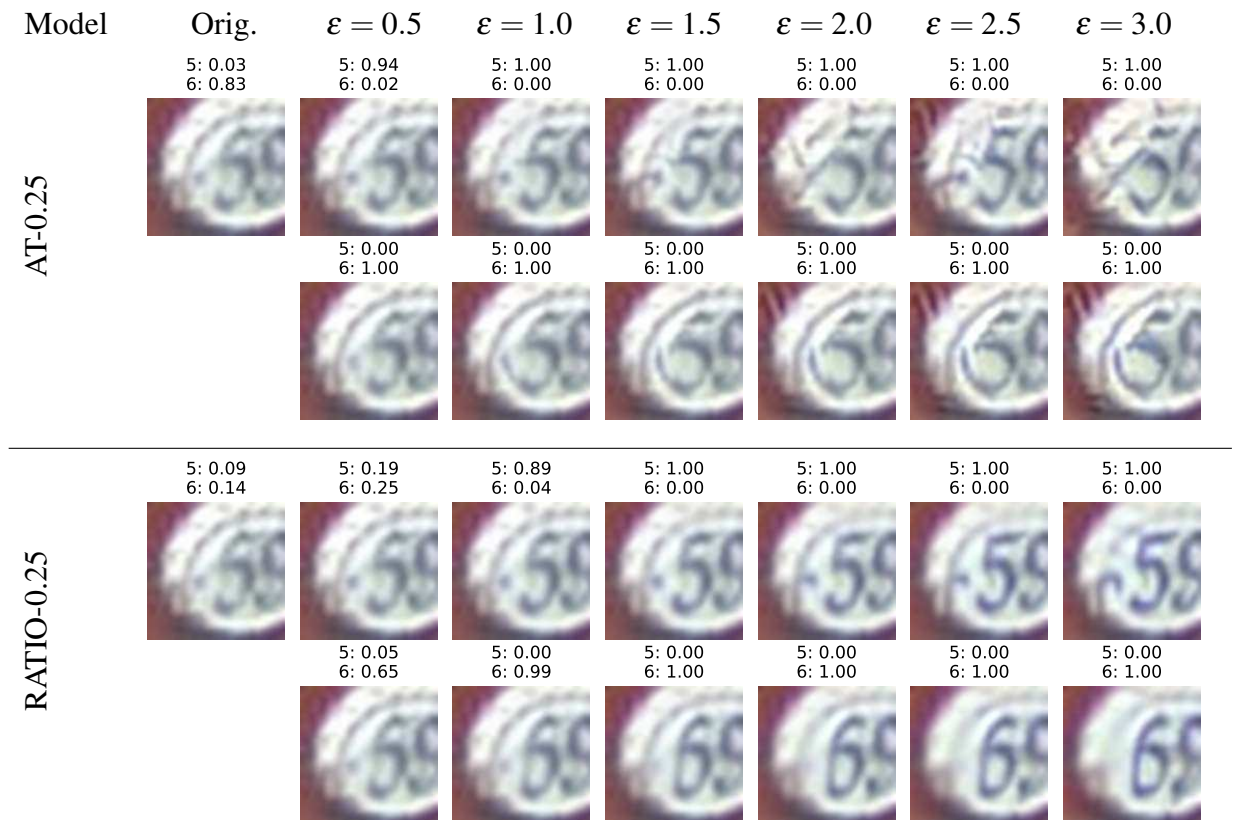


Figure 4.3: **Visual Counterfactuals (SVHN):** The 5 on the left is misclassified by all models. We show counterfactuals for the true class the predicted class (see Figure 4.2). RATIO consistently produces samples with fewer artefacts than AT.

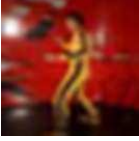

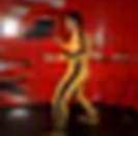
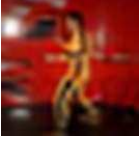
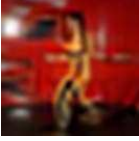
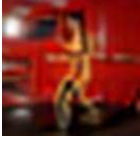
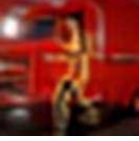
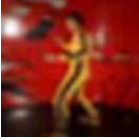
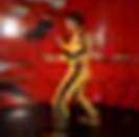
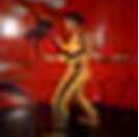
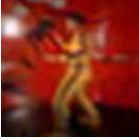
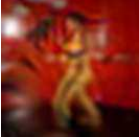
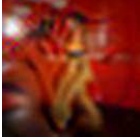

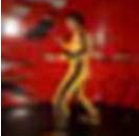

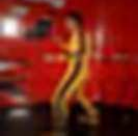
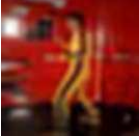
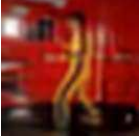
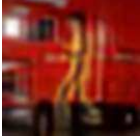

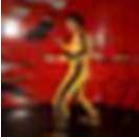
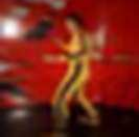
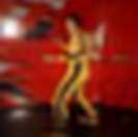
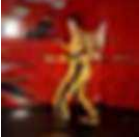
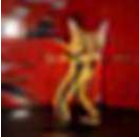
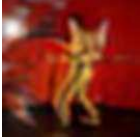
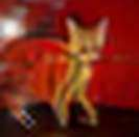
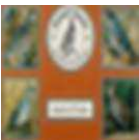


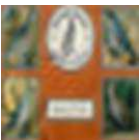
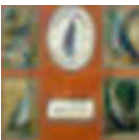
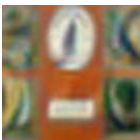
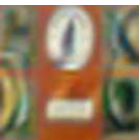
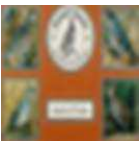






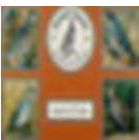



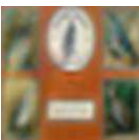
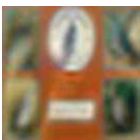

Model	Orig.	$\epsilon = 0.5$	$\epsilon = 1.0$	$\epsilon = 1.5$	$\epsilon = 2.0$	$\epsilon = 2.5$	$\epsilon = 3.0$
ACET	truck - 0.11	truck - 0.16	truck - 0.50	truck - 0.99	truck - 1.00	truck - 1.00	truck - 1.00
							
JEM-0	cat - 0.54	cat - 0.87	cat - 0.95	cat - 0.98	cat - 0.99	cat - 0.99	cat - 1.00
							
AT-0.50	truck - 0.26	truck - 0.84	truck - 0.99	truck - 1.00	truck - 1.00	truck - 1.00	truck - 1.00
							
R-0.25	cat - 0.13	cat - 0.20	cat - 0.34	cat - 0.69	cat - 0.97	cat - 1.00	cat - 1.00
							
ACET	2 - 0.10	2 - 0.10	2 - 0.10	2 - 0.11	2 - 0.57	2 - 0.98	2 - 1.00
							
AT-0.25	2 - 0.99	2 - 1.00	2 - 1.00	2 - 1.00	2 - 1.00	2 - 1.00	2 - 1.00
							
R-0.25	2 - 0.10	2 - 0.10	2 - 0.10	2 - 0.17	2 - 0.84	2 - 0.99	2 - 1.00
							

Figure 4.4: **Feature Generation for out-distribution images (CIFAR10 (top), SVHN (bottom))**: targeted attacks towards the class achieving highest confidence on original image for different budgets of the l_2 -radius ranging from $\epsilon = 0.5$ to $\epsilon = 3$. RATIO-0.25 generates the visually best images and in particular has reasonable confidence values for its decision. While AT-0.5/AT-0.25 generates also good images it is overconfident into the target class.

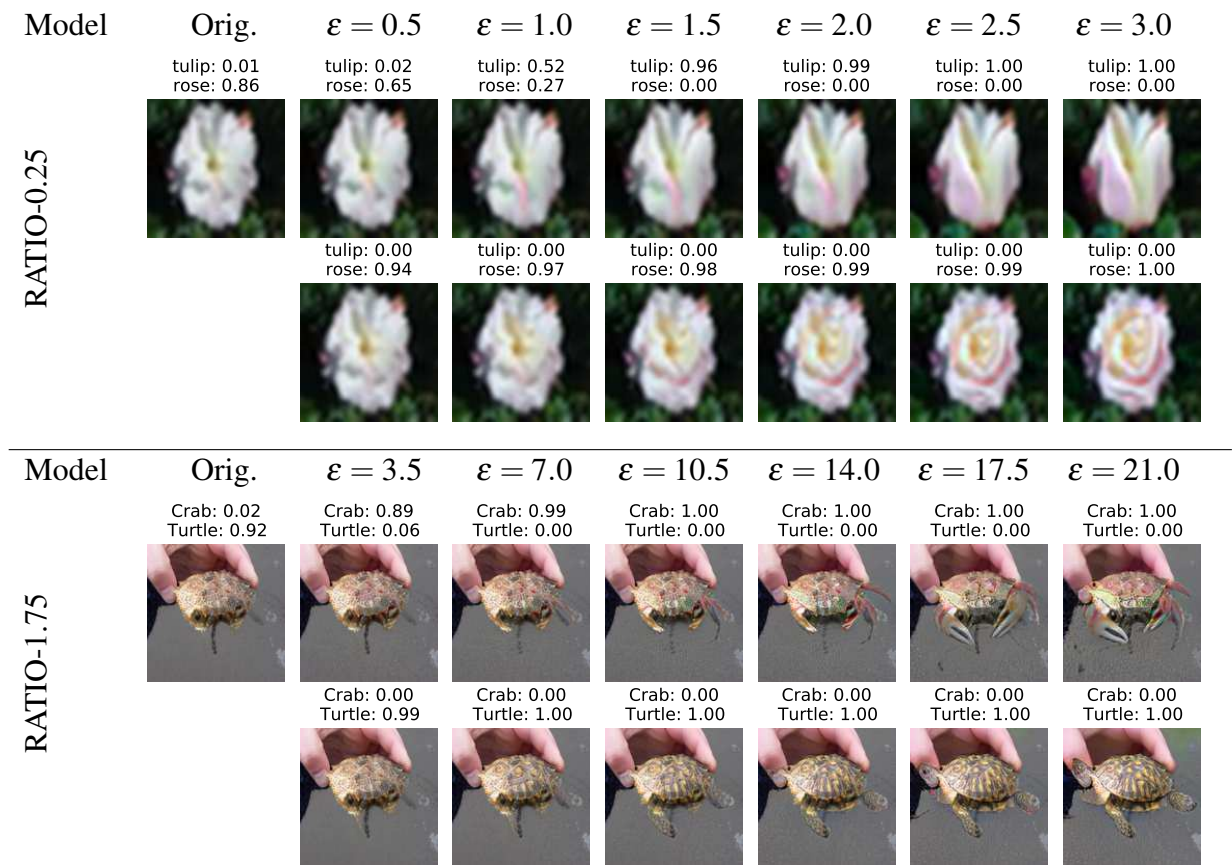


Figure 4.5: **Visual Counterfactuals** top: RATIO-0.25 for CIFAR100 and bottom: RATIO-1.75 for RestrictedImageNet.

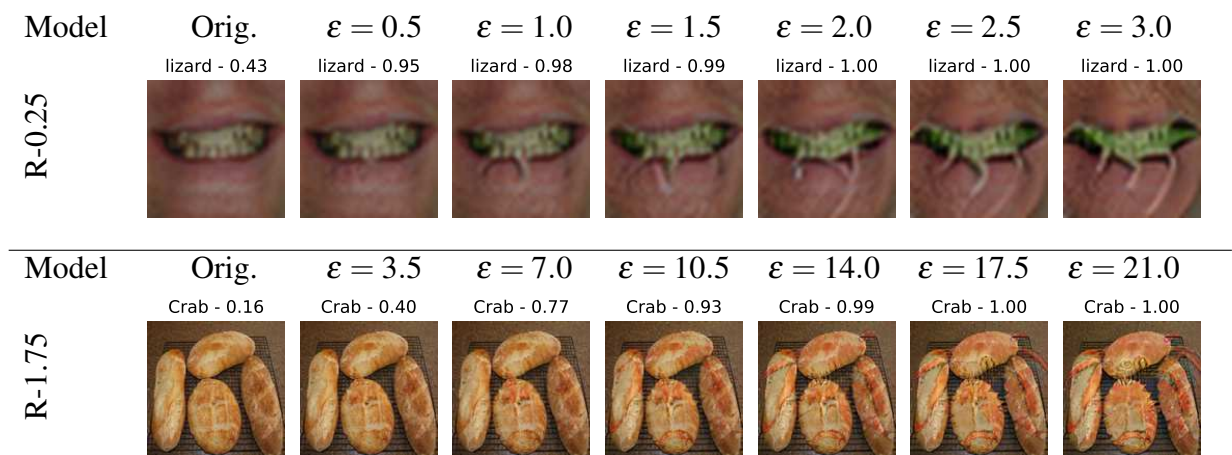


Figure 4.6: **Feature Generation for out-distribution images** top: RATIO-0.25 for CIFAR100 and bottom: RATIO-1.75 for R.ImageNet

Part III

Certifiable Adversarial Out-of-Distribution Detection

Chapter 5

Towards neural networks that provably know when they don't know

This chapter is based on (Meinke and Hein, 2020) (as was Chapter 2) which we presented at ICLR 2020. I was the first author of the paper and performed all experiments. Besides general guidance, Matthias Hein provided the initial idea and wrote significant parts of the paper, including revising the proofs of the theoretical results.

5.1 Introduction

In the previous chapters we have seen that the confidence of a classifier can be a good out-of-distribution detector if the classifier has been exposed to a large and diverse training out-distribution. Furthermore, we saw that, by default, this confidence estimate is not adversarially robust on the out-distribution, but that specifically training for this robustness can alleviate the issue. However, a problem that we have completely neglected thus far is the issue of asymptotic overconfidence that we outlined in Section 1.2.2. Recall that Hein *et al.* (2019) showed that under mild assumptions, piecewise linear classifiers such as ReLU networks provably suffer from overconfident predictions when moving sufficiently far away from the training data in almost any direction. Therefore, fixing this issue of asymptotic overconfidence requires some modification to a classifier's architecture and this will be the main goal of this chapter.

We will demonstrate how to construct a classifier that asymptotically has uniform confidence across all classes and then formally prove this property, see Figure 5.1 for an illustration. The final model can use arbitrary network architectures in its classifier model, without losing performance on either the prediction task on the in-distribution nor the OOD detection performance. Despite our construction being motivated by asymptotic properties that require leaving the image space $\mathcal{X} = [0, 1]^d$, we will show that our model actually additionally provides provable upper bounds on the confidence over whole neighborhoods around points that are indeed valid images. We show that most state-of-the-art OOD detection methods can be fooled by maximizing the confidence in such balls even when starting from uniform noise images, which should be trivial to identify. The central difference from existing OOD-methods is that we use Bayes' law to decompose the model so that we model in-and out-distribution separately. In this framework our algorithm for training neural networks follows directly as

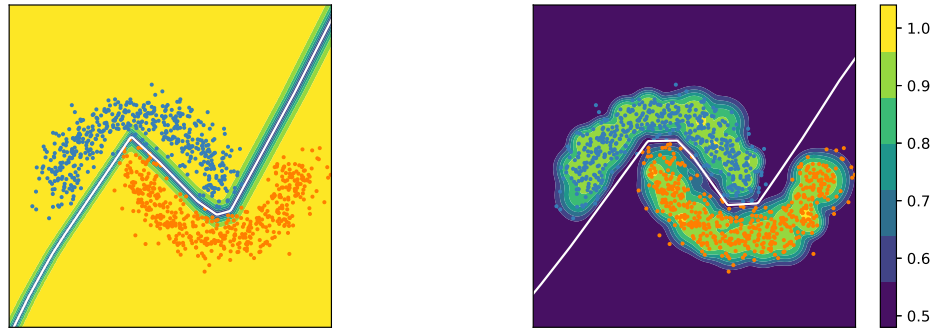


Figure 5.1: **Illustration on toy dataset:** We show the color-coded confidence in the prediction (yellow indicates high confidence $\max_y \hat{p}(y|x) \approx 1$, whereas dark purple regions indicate low confidence $\max_y \hat{p}(y|x) \approx 0.5$) for a normal neural network (left) and our CCU neural network (right). The decision boundary is shown in white which is similar for both models. Our CCU-model retains high-confidence predictions in regions close to the training data, whereas far away from the training the CCU-model outputs close to uniform confidence. In contrast the normal neural network is over-confident everywhere except very close to the decision boundary.

maximum likelihood estimator which is different from the more ad-hoc methods proposed in the literature. The usage of Gaussian mixture models as the density estimator is the essential key to get the desired provable guarantees.

5.2 Certified low confidence far away from the training data

5.2.1 A probabilistic model for in- and out-distribution data

We assume that there exists a joint probability distribution $p(x, y)$ over the in- and out-distribution data and since we are interested in classification, we want to estimate $p(y|x)$. We can represent this via the conditional distributions of the in-distribution $p(y|x, i)$ and out-distribution $p(y|x, o)$:

$$p(y|x) = p(y|x, i)p(i|x) + p(y|x, o)p(o|x) = \frac{p(y|x, i)p(x|i)p(i) + p(y|x, o)p(x|o)p(o)}{p(x|i)p(i) + p(x|o)p(o)}. \quad (5.1)$$

At first it might seem strange to have a conditional distribution $p(y|x, o)$ for out-distribution data, but until now we have made no assumptions about what in- and out-distribution are. A realistic scenario would be that at test time we are presented with instances x from other classes (out-distribution) for which we expect a close to uniform $p(y|x, o)$. Recall that we have already shown in Chapter 3 that a Bayes optimal OE model indeed makes this assumption.

Our model for $\hat{p}(y|x)$ has the same form as $p(y|x)$

$$\hat{p}(y|x) = \frac{\hat{p}(y|x, i)\hat{p}(x|i)\hat{p}(i) + \hat{p}(y|x, o)\hat{p}(x|o)\hat{p}(o)}{\hat{p}(x|i)\hat{p}(i) + \hat{p}(x|o)\hat{p}(o)}. \quad (5.2)$$

Typically, out-distribution data has no relation to the actual task and thus we would like to have uniform confidence over the classes. Therefore in our model we set

$$\hat{p}(y|x, o) = \frac{1}{K} \quad \text{and} \quad \hat{p}(y|x, i) = \frac{e^{f_y(x)}}{\sum_{k=1}^K e^{f_k(x)}}, \quad y \in \{1, \dots, K\}, \quad (5.3)$$

where $f : \mathbb{R}^d \rightarrow \mathbb{R}^K$ is the classifier function (logits). This framework is generic for classifiers trained with the cross-entropy (CE) loss (as the softmax function is the correct link function for the CE loss) and in particular we focus on neural networks. As described in Section 1.2.2, for a ReLU network the classifier function f is componentwise a continuous piecewise affine function and has been shown to produce asymptotically arbitrarily highly confident predictions, i.e. the classifier gets more confident in its predictions the further it moves away from its training data. One of the main goals of our technique is to fix this behavior of neural networks in a provable way.

Note that with the choice of $\hat{p}(y|x, o)$ and non-zero priors for $\hat{p}(i), \hat{p}(o)$, the full model $\hat{p}(y|x)$ can be seen as a calibrated version of $\hat{p}(y|x, i)$, where $\hat{p}(y|x) \approx \hat{p}(y|x, i)$ for inputs with $\hat{p}(x|i) \gg \hat{p}(x|o)$ and $\hat{p}(y|x) \approx \frac{1}{K}$ if $\hat{p}(x|i) \ll \hat{p}(x|o)$. However, note that only the confidence in the prediction $\hat{p}(y|x)$ is affected, the classifier decision is still done according to $\hat{p}(y|x, i)$ as the calibration does not change the ranking. Thus even if the OOD data came from the classification task we would like to solve, the trained classifier’s performance would be unaffected, only the confidence in the prediction would be damped. For the marginal out-distribution $\hat{p}(x|o)$ we will make the same assumption as in most of the previous chapters and use 80 million tiny image dataset (Torralba *et al.*, 2008) as a proxy of all possible images. Thus we estimate the density of $\hat{p}(x|o)$ using this data.

In order to obtain guarantees, the employed generative models for $\hat{p}(x|i)$ and $\hat{p}(x|o)$ have to be chosen in a way that allows one to control predictions far away from the training data. Variational autoencoders (VAEs) (Kingma and Welling, 2014; Rezende *et al.*, 2014), normalizing flows (Dinh *et al.*, 2016; Kingma and Dhariwal, 2018) and generative adversarial networks (GANs) (Goodfellow *et al.*, 2014) are powerful generative models. However, there is no direct way to control the likelihood far away from the training data. Moreover, it has been discovered that VAEs, flows and GANs also predict high likelihoods (Nalisnick *et al.*, 2019; Hendrycks *et al.*, 2019a) far away from the data they are supposed to model as well as adversarial samples (Kos *et al.*, 2017).

For $\hat{p}(x|o)$ and $\hat{p}(x|i)$ we use a Gaussian mixture model (GMM) which is far less powerful than deep learning based models but has the advantage that the density estimates can be

controlled far away from the training data:

$$\hat{p}(x|i) = \sum_{k=0}^{K_i} \alpha_k \exp\left(-\frac{d(x, \mu_k)^2}{2\sigma_k^2}\right), \quad \hat{p}(x|o) = \sum_{l=0}^{K_o} \beta_l \exp\left(-\frac{d(x, \nu_l)^2}{2\theta_l^2}\right), \quad (5.4)$$

where $K_i, K_o \in \mathbb{N}$ are the number of centroids and $d : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is the metric

$$d(x, y) = \|C^{-\frac{1}{2}}(x - y)\|_2, \quad (5.5)$$

with C being a positive definite matrix and

$$\alpha_k = \frac{1}{K_i} \frac{1}{(2\pi\sigma_k^2 \det C)^{\frac{d}{2}}}, \quad \beta_l = \frac{1}{K_o} \frac{1}{(2\pi\theta_l^2 \det C)^{\frac{d}{2}}}. \quad (5.6)$$

We later fix C as the regularized covariance matrix of the in-distribution data (see Section 5.3 for details). Thus one just has to estimate the centroids μ_k, ν_l and the variances σ_k^2, θ_l^2 . The idea of this metric is to use distances adapted to the data-distribution. Note that (5.4) is a properly normalized density in \mathbb{R}^d .

5.2.2 Maximum likelihood estimation

Given models for $\hat{p}(y|x)$ and $\hat{p}(x)$ we effectively have a full generative model and apply maximum likelihood estimation to get the underlying classifier $\hat{p}(y|x, i)$ and the parameters of the Gaussian mixture models $\hat{p}(x|i), \hat{p}(x|o)$. The only free parameter left is the probability $\hat{p}(i), \hat{p}(o)$ which, as in the previous chapters, we compactly write as $\lambda = \frac{\hat{p}(o)}{\hat{p}(i)}$. In our experiments we fix it to $\lambda = 1$. Our loss is therefore:

$$\begin{aligned} \mathbb{E}_{(x,y) \sim p(x,y)} \log(\hat{p}(y,x)) &= \mathbb{E}_{(x,y) \sim p(x,y)} \log(\hat{p}(y|x)) + \log(\hat{p}(x)), \\ &= \mathbb{E}_{(x,y) \sim p(x,y)} \log\left(\frac{\hat{p}(y|x, i)\hat{p}(x|i)\hat{p}(i) + \frac{1}{K}\hat{p}(x|o)\hat{p}(o)}{\hat{p}(x|i)\hat{p}(i) + \hat{p}(x|o)\hat{p}(o)}\right) + \log(\hat{p}(x|i)\hat{p}(i) + \hat{p}(x|o)\hat{p}(o)). \end{aligned} \quad (5.7)$$

In practice, we have to compute empirical expectations from finite training data from the in-distribution $(x_i, y_i)_{i=1}^N$ and out-distribution $(z_j)_{j=1}^M$. Labels for the out-distribution could be generated randomly via $p(y|x, o) = \frac{1}{K}$, but we obtain an unbiased estimator with lower variance by averaging over all classes directly, as was done in Lee *et al.* (2017); Hein *et al.* (2019); Hendrycks *et al.* (2019a). Now we can estimate the classifier f and the mixture model

parameters μ, ν, σ, θ via

$$\arg \max_{f, \mu, \nu, \sigma, \theta} \left\{ \frac{1}{N} \sum_{i=1}^N \log(\hat{p}(y_i|x_i)) + \frac{\lambda}{M} \sum_{j=1}^M \frac{1}{K} \sum_{y=1}^K \log(\hat{p}(y|z_j)) \right. \\ \left. + \frac{1}{N} \sum_{i=1}^N \log(\hat{p}(x_i)) + \frac{\lambda}{M} \sum_{j=1}^M \log(\hat{p}(z_j)) \right\}, \quad (5.8)$$

with

$$\hat{p}(y|x) = \frac{\hat{p}(y|x, i)\hat{p}(x|i) + \frac{\lambda}{K}\hat{p}(x|o)}{\hat{p}(x|i) + \lambda\hat{p}(x|o)} \quad \text{and} \quad \hat{p}(x) = \frac{1}{\lambda + 1} \left(\hat{p}(x|i) + \lambda\hat{p}(x|o) \right). \quad (5.9)$$

Due to the bounds derived in Section 5.2.3, we denote our method by **Certified Certain Uncertainty (CCU)**. Neglecting the terms for $\hat{p}(x)$ we recover OE. The key difference in this approach is that $\hat{p}(y|x) \neq \hat{p}(y|x, i)$ and the estimated densities for in- and out distribution $\hat{p}(x|i)$ and $\hat{p}(x|o)$ lead to a confidence calibration of $\hat{p}(y|x)$, and in turn the fit of the classifier influences the estimation of $\hat{p}(x|i)$ and $\hat{p}(x|o)$. The major advantage of our model is that we can give guarantees on the confidence of the classifier decision far away from the training data.

5.2.3 Proofs of close to uniform predictions far away from data

In this section we provide two types of guarantees on the confidence of a classifier trained according to our model in (5.8). The first one says that the classifier has provably low confidence far away from the training data, where an explicit bound on the minimal distance is provided, and the second provides an upper bound on the confidence in a ball around a given input point. The latter bound resembles robustness guarantees for adversarial samples (Hein and Andriushchenko, 2017; Wong and Kolter, 2018; Raghunathan *et al.*, 2018a; Mirman *et al.*, 2018) and is going to enable us to compute lower bounds on the WCAUC introduced in Eq. (4.3).

We provide our bounds for a more general mixture model which includes our GMM in (5.4) as a special case. To our knowledge, these are the first such bounds for neural networks and thus it is the first modification of a ReLU neural network so that it provably “knows when it does not know” (Hein *et al.*, 2019) in the sense that far away from the training data the predictions are close to uniform over the classes.

Theorem 3. *Let $(x_i, y_i)_{i=1}^N$ be the training set of the in-distribution and let the model for the conditional probability be given as*

$$\forall x \in \mathbb{R}^d, y \in \{1, \dots, K\}, \quad \hat{p}(y|x) = \frac{\hat{p}(y|x, i)\hat{p}(x|i) + \frac{\lambda}{K}\hat{p}(x|o)}{\hat{p}(x|i) + \lambda\hat{p}(x|o)}, \quad (5.10)$$

where $\lambda = \frac{\hat{p}(o)}{\hat{p}(i)} > 0$ and let the model for the marginal density of the in-distribution $\hat{p}(x|i)$ and

out-distribution $p(x|o)$ be given by the generalized GMMs

$$\hat{p}(x|i) = \sum_{k=0}^{K_i} \alpha_k \exp\left(-\frac{d(x, \mu_k)^2}{2\sigma_k^2}\right), \quad \hat{p}(x|o) = \sum_{l=0}^{K_o} \beta_l \exp\left(-\frac{d(x, \nu_l)^2}{2\theta_l^2}\right) \quad (5.11)$$

with $\alpha_k, \beta_l > 0$ and $\mu_k, \nu_l \in \mathbb{R}^d \quad \forall k = 1, \dots, K_i, l = 1, \dots, K_o$ and $d: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}_+$ a metric. Let $z \in \mathbb{R}^d$ and define

$$k^* = \arg \min_{k=1, \dots, K_i} \frac{d(z, \mu_k)}{\sigma_k} \quad i^* = \arg \min_{i=1, \dots, N} d(z, x_i) \quad (5.12)$$

$$l^* = \arg \min_{l=1, \dots, K_o} \beta_l \exp\left(-\frac{d(z, \nu_l)^2}{2\theta_l^2}\right) \quad \Delta = \frac{\theta_{l^*}^2}{\sigma_{k^*}^2} - 1. \quad (5.13)$$

For any $\varepsilon > 0$, if $\min_l \theta_l > \max_k \sigma_k$ and

$$\min_{i=1, \dots, N} d(z, x_i) \geq d(x_{i^*}, \mu_{k^*}) + d(\mu_{k^*}, \nu_{l^*}) \left[\frac{2}{\Delta} + \frac{1}{\sqrt{\Delta}} \right] + \frac{\theta_{l^*}}{\sqrt{\Delta}} \sqrt{\log\left(\frac{K-1}{\varepsilon \lambda} \frac{\sum_k \alpha_k}{\beta_{l^*}}\right)}, \quad (5.14)$$

then it holds for all $y \in \{1, \dots, K\}$ that

$$\hat{p}(y|z) \leq \frac{1}{K} (1 + \varepsilon). \quad (5.15)$$

In particular, if $\min_i d(z, x_i) \rightarrow \infty$, then $\hat{p}(y|z) \rightarrow \frac{1}{K}$.

Proof. The proof essentially hinges on upper bounding $\frac{\hat{p}(z|i)}{\hat{p}(z|o)}$ using the specific properties of the Gaussian mixture model. We note that

$$\hat{p}(y|x) = \frac{\hat{p}(y|x, i) \hat{p}(x|i) + \frac{\lambda}{K} \hat{p}(x|o)}{\hat{p}(x|i) + \lambda \hat{p}(x|o)} = \frac{1}{K} \frac{1 + \frac{K}{\lambda} \frac{\hat{p}(x|i)}{\hat{p}(x|o)}}{1 + \frac{1}{\lambda} \frac{\hat{p}(x|i)}{\hat{p}(x|o)}} \leq \frac{1}{K} \left(1 + \frac{K-1}{\lambda} \frac{\hat{p}(x|i)}{\hat{p}(x|o)} \right) \quad (5.16)$$

The last step holds because the function $g(\xi) = \frac{1+K\xi}{1+\xi}$ is monotonically increasing

$$\frac{\partial g}{\partial \xi} = \frac{K-1}{(1+\xi)^2} \quad \text{and} \quad \frac{\partial^2 g}{\partial \xi^2} = -2 \frac{K-1}{(1+\xi)^3}. \quad (5.17)$$

As the second derivative is negative for $\xi \geq 0$, g is concave for $\xi \geq 0$ and thus

$$\frac{1+K\xi}{1+\xi} = g(\xi) \leq g(0) + \frac{\partial g}{\partial \xi} \Big|_{\xi=0} (\xi - 0) = 1 + (K-1)\xi. \quad (5.18)$$

In order to achieve the required result we need to show that $\frac{K-1}{\lambda} \frac{\hat{p}(x|i)}{\hat{p}(x|o)} \leq \varepsilon$ for x sufficiently far

away from the training data.

We note that

$$\frac{\hat{p}(x|i)}{\hat{p}(x|o)} = \frac{\sum_k \alpha_k \exp\left(-\frac{d(x, \mu_k)^2}{2\sigma_k^2}\right)}{\sum_l \beta_l \exp\left(-\frac{d(x, \nu_l)^2}{2\theta_l^2}\right)} \leq \frac{\max_k \exp\left(-\frac{d(x, \mu_k)^2}{2\sigma_k^2}\right) \sum_k \alpha_k}{\max_l \beta_l \exp\left(-\frac{d(x, \nu_l)^2}{2\theta_l^2}\right)} \quad (5.19)$$

$$= \frac{\sum_k \alpha_k}{\beta_{l^*}} \exp\left(-\frac{d(x, \mu_{k^*})^2}{2\sigma_{k^*}^2} + \frac{d(x, \nu_{l^*})^2}{2\theta_{l^*}^2}\right) \quad (5.20)$$

where $k^* = \arg \min_k \frac{d(x, \mu_k)^2}{2\sigma_k^2}$ and $l^* = \arg \min_l \beta_l \exp\left(-\frac{d(x, \nu_l)^2}{2\theta_l^2}\right)$. Using triangle inequality, $d(x, \nu_{l^*}) \leq d(x, \mu_{k^*}) + d(\mu_{k^*}, \nu_{l^*})$, we get the desired condition as

$$\frac{\sum_k \alpha_k}{\beta_{l^*}} \exp\left(-d(x, \mu_{k^*})^2 \left(\frac{1}{2\sigma_{k^*}^2} - \frac{1}{2\theta_{l^*}^2}\right) + \frac{d(\mu_{k^*}, \nu_{l^*})d(x, \mu_{k^*})}{\theta_{l^*}^2} + \frac{d(\mu_{k^*}, \nu_{l^*})^2}{2\theta_{l^*}^2}\right) \leq \frac{\varepsilon \lambda}{K-1} \quad (5.21)$$

Thus we get with $a = \left(\frac{1}{2\sigma_{k^*}^2} - \frac{1}{2\theta_{l^*}^2}\right)$, $b = \frac{d(\mu_{k^*}, \nu_{l^*})}{\theta_{l^*}^2}$ and $c = \frac{d(\mu_{k^*}, \nu_{l^*})^2}{2\theta_{l^*}^2}$, $d = \log\left(\frac{\varepsilon \lambda}{K-1} \frac{\beta_{l^*}}{\sum_k \alpha_k}\right)$, the quadratic inequality

$$-d(x, \mu_{k^*})^2 a + d(x, \mu_{k^*}) b + \frac{b^2}{2} \leq d, \quad (5.22)$$

where $d < 0$ for sufficiently small ε . We get the solution

$$d(x, \mu_{k^*}) \geq \frac{b}{2a} + \sqrt{\max\left\{0, \frac{c-d}{a} + \frac{b^2}{4a^2}\right\}}. \quad (5.23)$$

It holds, using $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for $a, b > 0$,

$$\frac{b}{2a} + \sqrt{\max\left\{0, \frac{c-d}{a} + \frac{b^2}{4a^2}\right\}} \leq \frac{b}{a} + \sqrt{\frac{c}{a}} + \sqrt{\frac{-d}{a}}. \quad (5.24)$$

One can simplify

$$\frac{b}{a} = 2 \frac{\sigma_{k^*}^2 \theta_{l^*}^2}{\theta_{l^*}^2 - \sigma_{k^*}^2} \frac{d(\mu_{k^*}, \nu_{l^*})}{\theta_{l^*}^2} = 2 \frac{\sigma_{k^*}^2 d(\mu_{k^*}, \nu_{l^*})}{\theta_{l^*}^2 - \sigma_{k^*}^2} = 2 \frac{d(\mu_{k^*}, \nu_{l^*})}{\frac{\theta_{l^*}^2}{\sigma_{k^*}^2} - 1} \quad (5.25)$$

$$\frac{c}{a} = 2 \frac{\sigma_{k^*}^2 \theta_{l^*}^2}{\theta_{l^*}^2 - \sigma_{k^*}^2} \frac{d(\mu_{k^*}, \nu_{l^*})^2}{2\theta_{l^*}^2} = \frac{\sigma_{k^*}^2 d(\mu_{k^*}, \nu_{l^*})^2}{\theta_{l^*}^2 - \sigma_{k^*}^2} = \frac{d(\mu_{k^*}, \nu_{l^*})^2}{\frac{\theta_{l^*}^2}{\sigma_{k^*}^2} - 1} \quad (5.26)$$

Noting that $d(x, \mu_{k^*}) \geq |d(x, x_{i^*}) - d(x_{i^*}, \mu_{k^*})|$ we get that

$$d(x, x_{i^*}) \geq d(x_{i^*}, \mu_{k^*}) + \frac{b}{2a} + \frac{b}{a} + \sqrt{\frac{c}{a}} + \sqrt{\frac{-d}{a}},$$

implies $\frac{K-1}{\lambda} \frac{\hat{p}(x|i)}{\hat{p}(x|o)} \leq \varepsilon$. The last statement follows directly by noting that by assumption $a > 0$ (independently of the choice of l^* and k^*) and $b, c, d(x_{i^*}, \mu_{k^*})$ are bounded as K_i, K_o, N are finite. With $\Delta = \frac{\theta_{l^*}^2}{\sigma_{k^*}^2} - 1$ we can rewrite the required condition as

$$d(x, x_{i^*}) \geq d(x_{i^*}, \mu_{k^*}) + d(\mu_{k^*}, \nu_{l^*}) \left[\frac{2}{\Delta} + \frac{1}{\sqrt{\Delta}} \right] + \frac{\theta_{l^*}}{\sqrt{\Delta}} \sqrt{\log \left(\frac{M-1}{\varepsilon \lambda} \frac{\sum_k \alpha_k}{\beta_{l^*}} \right)}. \quad (5.27)$$

□

Theorem 3 holds for any multi-class classifier which for each input defines a probability distribution over the labels. Given the parameters of the GMM's it quantifies at which distance of an input z to the training set the classifier achieves close to uniform confidence. The theorem holds even if we use ReLU classifiers which in their unmodified form have been shown to produce arbitrarily high confidence far away from the training data (Hein *et al.*, 2019). This is a first step towards neural networks which provably know when they don't know.

In the next corollary, we provide an upper bound on the confidence over a ball around a given data point. This allows to give “confidence guarantees” for a whole volume and thus is much stronger than the usual pointwise evaluation of OOD methods.

Corollary 2. Let $x_0 \in \mathbb{R}^d$ and $R > 0$, then with $\lambda = \frac{\hat{p}(o)}{\hat{p}(i)}$ it holds

$$\max_{d(x, x_0) \leq R} \hat{p}(y|x) \leq \frac{1}{K} \frac{1 + K \frac{b}{\lambda}}{1 + \frac{b}{\lambda}}, \quad (5.28)$$

$$\text{where } b = \frac{\sum_{k=1}^{K_i} \alpha_k \exp\left(-\frac{\max\{d(x_0, \mu_k) - R, 0\}^2}{2\sigma_k^2}\right)}{\sum_{l=1}^{K_o} \beta_l \exp\left(-\frac{(d(x_0, \nu_l) + R)^2}{2\theta_l^2}\right)}.$$

Proof. From the previous section we already know that $\hat{p}(y|x) \leq \frac{1}{K} \frac{1 + K \frac{b}{\lambda}}{1 + \frac{b}{\lambda}}$ as long as $\frac{p(x|i)}{p(x|o)} \leq b$. Now we can separately bound the numerator and denominator within a ball of radius R around

x_0 . For the numerator we have

$$\max_{d(x,x_0) \leq R} \hat{p}(x|i) \leq \sum_{k=1}^{K_i} \alpha_k \max_{d(x,x_0) \leq R} e^{-\frac{d(x,\mu_k)^2}{2\sigma_k^2}} \quad (5.29)$$

$$\begin{aligned} &\leq \sum_{k=1}^{K_i} \alpha_k \exp\left(-\frac{\min_{d(x,x_0) \leq R} d(x,\mu_k)^2}{2\sigma_k^2}\right) \\ &\leq \sum_{k=1}^{K_i} \alpha_k \exp\left(-\frac{(\max\{d(\mu_k, x_0) - R, 0\})^2}{2\sigma_k^2}\right), \end{aligned} \quad (5.30)$$

where we have lower bounded $\min_{d(x,x_0) \leq R} d(x,\mu_k)$ via the reverse triangle inequality

$$\begin{aligned} \min_{d(x,x_0) \leq R} d(x,\mu_k) &\geq \min_{d(x,x_0) \leq R} |d(x_0,\mu_k) - d(x,x_0)|, \\ &\geq \max\left\{\min_{d(x,x_0) \leq R} (d(x_0,\mu_k) - d(x,x_0)), 0\right\}, \\ &\geq \max\{d(x_0,\mu_k) - r, 0\}. \end{aligned} \quad (5.31)$$

The denominator can similarly be bounded via

$$\min_{d(x,x_0) \leq R} \hat{p}(x|o) \geq \sum_{l=1}^{K_o} \beta_l \min_{d(x,x_0) \leq R} e^{-\frac{d(x,\nu_l)^2}{2\theta_l^2}} \quad (5.32)$$

$$\begin{aligned} &\geq \sum_{l=1}^{K_o} \beta_l \exp\left(-\frac{\max_{d(x,x_0) \leq R} d(x,\nu_l)^2}{2\theta_l^2}\right) \\ &\geq \sum_{l=1}^{K_o} \beta_l \exp\left(-\frac{(d(x_0,\nu_l) + R)^2}{2\theta_l^2}\right). \end{aligned} \quad (5.33)$$

With both of these bounds in place the conclusion immediately follows. \square

In Section 5.3 we show that even though OOD methods achieve low confidence on noise images, the maximization of the confidence in a ball around a noise point (adversarial noise) yields high confidence predictions for OOD methods, whereas our classifier has provably low confidence, as certified by Corollary 2. In fact, we can use Corollary 2 to compute lower bounds on the worst-case AUC defined in Eq. (4.3). Recall that the WCAUC with respect to some metric $d(\cdot, \cdot)$ is defined as:

$$\text{WCAUC}_h(p_1, p_2) = \mathbb{E}_{\substack{x \sim p_1 \\ z \sim p_2}} \left[\mathbb{1}_{h(x) > \max_{d(z',z) \leq \epsilon} h(z')} \right], \quad (5.34)$$

where, in the case of CCU, the scoring function is the confidence, i.e. $h(x) = \max_y \hat{p}(y|x)$.

Since we are able to derive upper bounds on the confidence around OOD samples, we can compute lower bounds on the WCAUC, which we call the Guaranteed AUC or GAUC.

5.3 Experiments

We evaluate the worst-case performance of various OOD detection methods within regions for which CCU yields guarantees and by standard OOD using MNIST (LeCun *et al.*, 1998), FashionMNIST (Xiao *et al.*, 2017), SVHN (Netzer *et al.*, 2011), CIFAR10 and CIFAR100 (Krizhevsky and Hinton, 2009) as in-distributions. As baselines we use the same methods that we described in Section 2.2. Recall that for calibrating those methods' hyperparameters, we use the 80 Million Tiny Images (Torralba *et al.*, 2008) dataset as out-distribution for a fair comparison. We show that all baselines yield undesired high confidence predictions in regions around uniform noise points for which CCU can certify low confidence. Our code is available on github.¹

5.3.1 Training

For CCU, unless specified otherwise we use ADAM on MNIST with a learning rate of $1e-3$ and SGD with learning rate 0.1 for the other datasets. The learning rate for the GMM is always set to $1e-5$. We decrease all learning rates by a factor of 10 after 50, 75 and 90 epochs. Our batch size is 128, the total number of epochs 100 and weight decay is set to $5e-4$. We also pick equal batches of in- and out-distribution data (corresponding to $p(i) = p(o)$) and concatenate them into a batches of size 256. Note that during the 100 epochs only a fraction of the 80 million tiny images are seen and so there is no risk of overfitting.

Our data augmentation scheme uses random crops with a padding of 2 pixels on MNIST and FMNIST. On SVHN, CIFAR10 and CIFAR100 the padding width is 4 pixels. For SVHN we fill the padding with the value at the boundary and for CIFAR we apply reflection at the boundary pixels. On top of this we include random horizontal flips on CIFAR. For MNIST and FMNIST we generate 60000 such samples and for SVHN and CIFAR 50000 samples by drawing from the clean dataset without replacement. This augmented data is used to calculate the covariance matrix from (5.35). During the actual training we use the same data augmentation scheme in a standard fashion.

For the Gaussian mixture models, we have to select a specific distance metric. As the Euclidean metric is known to be a relatively bad distance between two images we instead use the distance $d(x, y) = \|C^{-\frac{1}{2}}(x - y)\|$, where C is generated as follows. We calculate the covariance matrix C' on augmented in-distribution samples. Let $(\lambda_i, u_i)_{i=1}^d$ be the eigenvalues/eigenvectors of C' . Then we set

$$C = \sum_{i=1}^d \max\{\lambda_i, 10^{-6} \max_j \lambda_j\} u_i u_i^T, \quad (5.35)$$

¹<https://github.com/AlexMeinke/certified-certain-uncertainty>

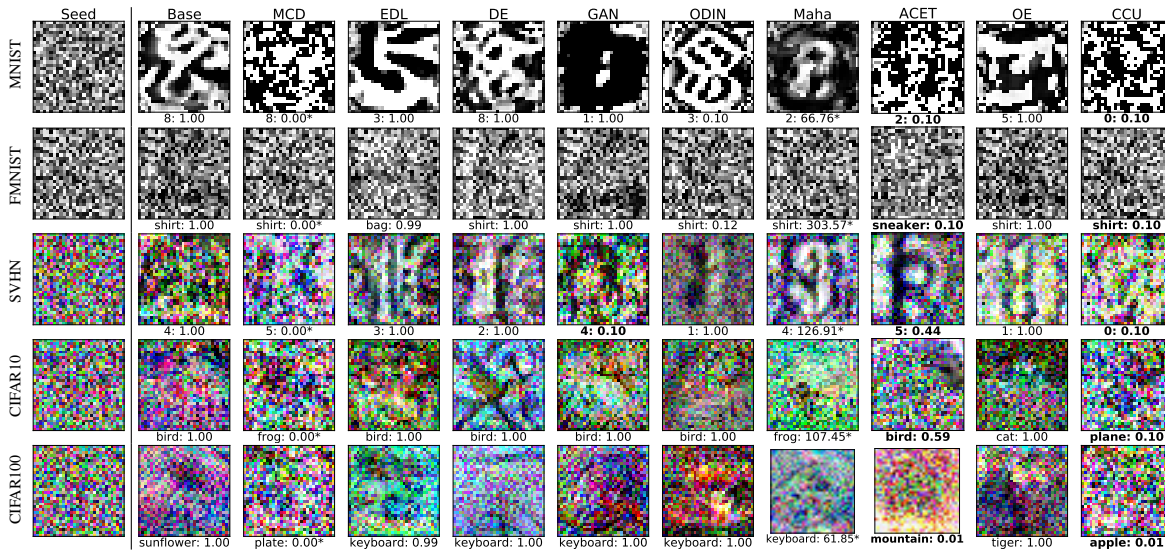


Figure 5.2: **Adversarial Noise:** We maximize the confidence of the OOD methods using PGD in the ball around a uniform noise sample (seed images, left) on which CCU is guaranteed to yield less than $1.1 \frac{1}{K}$ maximal confidence by Corollary 2. For each OOD method we report the image with the highest confidence. Maha and MCD use scores where lower is more confident (indicated by *). If we do *not* find a sample that has higher confidence/lower score than the median of the in-distribution, we highlight this in boldface. All other OOD methods fail on some dataset, see Table 5.1 for a quantitative version. ODIN at high temperatures always returns low confidence, so a value of 0.1 is not informative.

that is we fix a lower bound on the smallest eigenvalue so that C has full rank. In Hendrycks and Gimpel (2017b) a similar metric has been used for detection of adversarial images. We choose $K_i = K_o = 100$ as the number of centroids for the GMMs. We initialize the in-GMM on augmented in-data using the EM algorithm with spherical covariance matrices in the transformed space, as in (5.4). For the out-distribution we use a subset of 20000 points for the initialization. While, initially it holds that $\forall k, l : \sigma_k < \theta_l$, as required in Theorem 3, this is not guaranteed during the optimization of (5.8). Thus, we enforce the constraint during training by setting: $\theta_l \mapsto \max\{\theta_l, 2 \max_k \sigma_k\}$ at every gradient step. Since the “classifier” and “density” terms in (5.8) have very different magnitudes we choose a small learning rate of $1e - 5$ for the parameters in the GMMs. It is also crucial to not apply weight decay to these parameters. The other hyperparameters are chosen as in the base model below.

5.3.2 Certified robustness against adversarial noise

We sample uniform noise images as they are obviously out-distribution for all tasks and, using Corollary 2, certify the largest ball around the uniform noise sample on which CCU attains at most $1.1 \cdot$ uniform confidence, that is 1.1% on CIFAR100 and 11% on all other datasets. Since

Table 5.1: Worst-case performance of different OOD detections methods in neighborhoods around uniform noise points certified by CCU. We report the clean test error (TE) on the in-distribution (GAN and MCD use VGG). The success rate (SR) is the fraction of adversarial noise points for which the confidence/score inside the ball is higher than the median of the in-distribution’s confidence/score. The AUC quantifies detection of adversarial noise versus in-distribution. All values in %.

		Base	MCD	EDL	DE	GAN	ODIN	Maha	ACET	OE	CCU
MNIST	TE	0.5	0.4	0.4	0.4	0.8	0.5	0.9	0.6	0.7	0.6
	SR	100.0	99.0	100.0	100.0	43.5	100.0	100.0	0.0	100.0	0.0
	AUC	1.4	8.6	0.0	7.3	54.4	0.0	11.7	100.0	35.2	100.0
FMNIST	TE	4.8	5.8	5.2	4.9	5.7	4.8	4.8	4.8	5.7	4.9
	SR	100.0	72.5	100.0	100.0	99.0	100.0	100.0	0.0	100.0	0.0
	AUC	0.0	47.1	0.0	0.4	39.5	0.0	18.8	100.0	35.7	100.0
SVHN	TE	2.9	3.9	3.1	2.4	4.2	2.9	2.9	3.2	4.1	3.0
	SR	100.0	73.5	100.0	100.0	0.0	100.0	100.0	3.0	100.0	0.0
	AUC	0.0	34.1	0.0	0.0	100.0	0.0	0.0	96.5	0.0	100.0
CIFAR10	TE	5.6	11.7	7.0	6.7	11.7	5.6	5.6	6.1	4.7	5.8
	SR	100.0	90.5	100.0	100.0	100.0	100.0	100.0	0.0	100.0	0.0
	AUC	0.0	23.9	0.0	0.0	25.3	0.0	0.0	99.9	0.0	100.0
CIFAR100	TE	23.3	45.3	31.1	27.5	43.8	23.3	23.2	25.2	24.7	25.9
	SR	100.0	100.0	100.0	100.0	89.5	100.0	100.0	3.5	100.0	0.0
	AUC	0.1	17.3	0.0	0.2	15.3	0.0	0.0	95.8	2.5	100.0

Corollary 2 does not explicitly give a radius, one has to numerically invert the bound. Note that the bound

$$b(R) = \frac{\sum_{k=1}^{K_i} \alpha_k \exp\left(-\frac{\max\{d(x_0, \mu_k) - R, 0\}^2}{2\sigma_k^2}\right)}{\sum_{l=1}^{K_o} \beta_l \exp\left(-\frac{(d(x_0, \nu_l) + R)^2}{2\theta_l^2}\right)} \quad (5.36)$$

is monotonically increasing in R . Thus, for a given sample x_0 one can fix a desired bound $\max_{d(x, x_0) \leq R} \hat{p}(x|i) \leq \frac{1}{K} \nu$, where $\nu \in (1, K)$ and then uniquely solve

$$b(R) = \frac{\nu - 1}{K - \nu} \lambda \quad (5.37)$$

for R via bisection. This radius \hat{R} will then represent the maximal radius, that one can certify using Corollary 2. The presumption is, of course, that for $R = 0$ one has a sufficiently low bound in the first place, i.e. that a solution exists. In our experiments on uniform noise we did not encounter a single counterexample to this assumption. However, for more difficult OOD samples, we did not find any points that were certifiable in this way. Note that, in principle, it could be possible that the certified balls are in a sense too large, i.e. that they contain training or test images. In the following section we will show that this is not the case.

5.3.3 Generating adversarial noise

We construct adversarial noise samples for all OOD methods by maximizing the respective scoring function via a PGD attack with 500 steps and 50 random restarts on this ball. We begin with a step size of 3 and for each of the 50 restarts we randomly initialize at some point in the ellipsoid. Whenever a gradient step successfully decreases the losses we increase the step size by a factor of 1.1. Whenever the loss increases instead we use backtracking and decrease the step size by a factor of 2. We apply normal PGD using the l_2 -norm in the transformed space to ensure that we stay on the ellipsoid and after each gradient step we transform back into the original space to project onto the box $[0, 1]^d$. The result is not guaranteed to lie within the ellipsoid so after the 500 steps we use the alternating projection algorithm (Bauschke and Borwein, 1996) for 10 steps which is guaranteed to converge to a point in the intersection of the ellipsoid and the box because both of these sets are convex.

In Table 5.1 we show the results of running this attack on the different models. We use 200 noise images and we report clean test error on the in-distribution, the success rate (SR) (fraction of adversarial noise points for which the OOD detection score inside the ball is higher than the median of the in-distribution’s score) and the AUC for the separation of the generated adversarial noise images and the in-distribution based on score. By construction (see Corollary 2) our method provably makes no overconfident predictions but we nevertheless run the attack on CCU as well. We note that only CCU performs perfectly on this task for all datasets - all other OOD methods fail at least on one datasets, most of them on all. We also see that ACET achieves very robust performance which may be expected as it does some kind of ad-

Table 5.2: Lowest confidence that CCU attains on the test set (in percent) as well as total number of test points on which confidence is lower than our imposed bound of $\frac{1.1}{K}$.

	$\min p(y x)$	$\# < \frac{1.1}{M}$	$\% < \frac{1.1}{M}$
MNIST	33.08	0	0
FMNIST	28.77	0	0
SVHN	10.02	20	0.08
CIFAR10	10.01	529	5.29
CIFAR100	1.03	130	1.30

versarial training for OOD detection. Nevertheless high-confidence adversarial noise images for ACET can be found on SVHN, CIFAR10 and CIFAR100 and ACET has no guarantees. We illustrate the generated adversarial noise images for all methods in Figure 5.2.

As one can observe in Figure 5.2 the images which maximize the confidence in the certified ball around the uniform noise image are sometimes quite far away from the original noise image. As CCU certifies low confidence (the maximal confidence is less than $1.1 \times \frac{1}{K}$ - so the predicted probability distribution over the classes is very close to the uniform distribution) over the whole ball, it is a natural question what these balls look like and what kind of images they contain. In particular, it is in general not desired that the certified balls contain images from the training and test set. For each dataset we certified balls around 200 uniform noise images and for each of the certified balls we check if it contains training or test images of the corresponding dataset. We found that even though the certified balls are large, not a single training or test image was contained in any of them. This justifies the use of our proposed threat model.

A different potential problem for our defined threat model could be that our threshold of $\frac{1.1}{K}$ for the certification is too high and that many predictions on the test set have confidence less than this threshold. For this purpose, in Table 5.2, we report the smallest predicted confidence of CCU on the test set T , that is

$$\min_{x \in T} \max_{y \in \{1, \dots, K\}} \hat{p}(y|x), \quad (5.38)$$

for each dataset and the total number of test samples where the confidence is below $\frac{1.1}{K}$. While for MNIST and FMNIST, this never happens, and for SVHN this is negligible (less than 0.1% of the test set), for CIFAR10 and CIFAR100 this happens in 5.3% resp. 1.3% of the cases.

In theory, this could impair our AUC value for the detection of adversarial noise. However, in practice our bound for the confidence is quite conservative as the bound is only tight in very specific configurations of the centroids of the Gaussian mixture model which are unlikely to happen for any practical dataset, meaning that the actual maximal confidence in the certified region is typically significantly lower. In fact the AUC values of CCU are always 100% which means that for all 200 certified balls the maximal value of the confidence of CCU in any of these balls (found by our PGD attack algorithm) is lower than the minimal confidence of

all predictions on the test set as reported in Table 5.2. On the other hand assuming a worst case scenario in the sense that we assume that the upper bound of the maximal confidence is attained in all 200 certified balls, then the (certified) AUC value would be: 99.92% for SVHN, 94.71% for CIFAR10, and 98.70% for CIFAR100. Note that this theoretical lower bound on our performance is still better than all other models’ empirical performance on this task, as reported in Table 5.1 on both CIFAR10 and CIFAR100, and only marginally below the perfect AUC of ACET and GAN on SVHN.

In order to ensure that CCU does not degrade our clean OOD detection performance, we also evaluate it on the same test out-distributions described in Chapter 2. We show these results in Table 5.3 where, for the reader’s convenience, we have also repeated the results of all baselines from Chapter 2. The table shows that CCU does not perform worse than OE, thus enabling us to give non-trivial guarantees without harming clean performance. In fact, comparing Table 5.1 and Table 5.3 we see that most models perform well when evaluating on uniform noise but fail when finding the worst case in a small neighborhood around the noise point.

5.4 Conclusion

In this chapter we have shown how to provably solve the issue of asymptotically overconfident predictions by combining our classifier with Gaussian mixture models. This change in architecture then also allowed us to derive guarantees for adversarially robust detection of OOD samples in a very specific threat model. We showed that these guarantees are, in fact, non-trivial, because they contain points that all of our baselines’ scoring functions would assign high scores to. Crucially, CCU’s guarantees did not degrade either our model’s OOD detection performance nor its accuracy.

Recent developments: The issue of asymptotic overconfidence has been further studied in Kristiadi *et al.* (2020), where the authors use a Bayesian approach to achieve asymptotically low confidence. Furthermore, in Kristiadi *et al.* (2021) they show how to mitigate far-away overconfidence in Bayesian neural networks. A different approach has been suggested in Liu *et al.* (2022), where they incorporate spectral normalization and a Gaussian process layer into deep classifiers which they prove also achieves asymptotically uniform confidence far from all training data. Finally, there has been more work on worst-case OOD detection, which we will discuss in the following chapters.

Table 5.3: Extension of Table 2.1. All numbers except for CCU are just repeated for the reader's convenience. AUC (in- versus out-distribution detection based on confidence/score) in percent for different OOD methods and datasets (higher is better). CCU's OOD performance is generally comparable to OE's.

		Base	MCD	EDL	DE	GAN	ODIN	Maha	ACET	OE	CCU
MNIST	FMNIST	97.4	93.1	99.3	99.2	99.4	98.7	96.8	100.0	99.9	99.9
	EMNIST	89.2	82.0	89.0	92.1	92.8	88.9	91.6	95.0	95.8	92.0
	GrCIFAR10	99.7	94.7	99.7	100.0	99.1	99.9	98.7	100.0	100.0	100.0
	Noise	100.0	95.2	99.9	100.0	99.3	100.0	97.2	100.0	100.0	100.0
	Uniform	95.2	87.9	99.9	97.9	99.9	98.2	100.0	100.0	100.0	100.0
FMNIST	MNIST	96.7	82.7	94.5	96.7	99.9	99.0	96.7	96.4	96.3	97.8
	EMNIST	97.5	87.3	95.6	97.1	99.9	99.3	97.5	97.6	99.3	99.5
	GrCIFAR10	91.0	92.3	84.0	86.1	85.3	93.0	98.2	96.2	100.0	100.0
	Noise	97.3	94.0	95.6	97.4	98.9	98.9	98.9	97.8	100.0	100.0
	Uniform	96.9	93.3	95.6	98.3	93.2	98.8	99.1	100.0	97.6	100.0
SVHN	CIFAR10	95.4	91.9	95.9	97.9	96.8	95.9	97.1	95.2	100.0	100.0
	CIFAR100	94.5	91.4	95.6	97.6	96.1	94.8	96.7	94.8	100.0	100.0
	LSUN_CR	95.6	92.0	95.3	97.9	99.0	96.5	97.2	97.1	100.0	100.0
	Imagenet-Noise	94.7	91.8	95.7	97.7	97.8	95.1	96.8	97.3	100.0	100.0
	Uniform	96.8	93.1	96.5	95.6	100.0	97.9	97.8	100.0	100.0	100.0
CIFAR10	SVHN	95.8	81.9	92.3	90.3	83.9	96.7	91.5	93.7	98.8	98.2
	CIFAR100	87.3	78.6	87.3	88.2	82.9	87.5	82.8	86.9	95.3	94.2
	LSUN_CR	91.9	81.3	90.8	92.0	89.9	93.3	89.2	91.2	98.6	98.2
	Imagenet-Noise	87.5	78.4	88.2	87.7	84.0	88.1	84.1	86.5	94.7	93.3
	Uniform	96.5	79.9	88.9	90.3	81.8	97.6	94.4	94.8	97.3	97.0
CIFAR100	SVHN	96.8	81.0	89.9	96.6	73.0	98.8	100.0	100.0	98.8	100.0
	SVHN	78.8	59.2	80.4	83.2	75.9	81.3	77.5	73.9	93.5	94.2
	CIFAR10	78.6	58.9	73.3	76.3	69.3	79.5	59.9	77.2	81.6	80.2
	LSUN_CR	81.0	59.4	74.2	81.6	79.8	81.4	79.7	78.0	95.4	95.9
	Imagenet-Noise	80.8	59.2	76.0	78.2	73.9	81.3	70.8	79.5	83.8	81.4
Uniform	73.4	58.7	65.9	67.5	73.6	76.8	90.6	62.9	86.9	94.6	
Uniform	93.3	62.0	29.8	36.6	100.0	93.5	94.3	100.0	99.1	100.0	

Chapter 6

Interval Bound Propagation for robust OOD Detection

This chapter is based on (Bitterwolf *et al.*, 2020) which was published at NeurIPS 2020. Both Julian Bitterwolf and I independently came up with the idea of using interval bound propagation for robust OOD detection. Julian Bitterwolf carried out the many experiments that were needed to find a stable training schedule for the method and he was the primary author of the paper. Matthias Hein provided general guidance and in-depth discussions. Matthias Hein and I assisted in the writing of the paper. I also carried out the evaluations of the adversarial AUCs for all models and investigated the properties of the method on MNIST vs. EMNIST.

6.1 Introduction

In the previous chapter we have seen that combining Gaussian mixture models and a classifier in a specific way not only guarantees that the confidence of the joint model will be low far away from the training data, but also implies concrete guarantees on the adversarial robustness the model’s confidence on OOD data. We also showed that these guarantees were actually meaningful because the baseline models produced overconfident predictions within the regions that our CCU could certify. Unfortunately, there were still many limitations: i) The certificates could only be obtained around uniform noise points but not on more challenging out-distributions that are more similar to the in-distribution. ii) The threat model that we could certify was highly non-standard. In a sense, we basically cherry-picked the threat model to be precisely what we could certify rather than certifying an existing threat model. In this chapter we will present a technique that overcomes these limitations, by dropping our requirements that the classifier’s network architecture be arbitrary and that the model’s confidence be provably uniform far from the training data.

Thus, we aim to provide worst-case OOD guarantees not only for noise but also for images from related but different image classification tasks. For this purpose we use the techniques from interval bound propagation (IBP) (Gowal *et al.*, 2018) to derive a provable upper bound on the maximal confidence of the classifier in an l_∞ -ball of radius ε around a given point. By minimizing this bound on the out-distribution using our training scheme GOOD (Guaranteed Out-Of-distribution Detection) we arrive at the first models which have guaranteed low confi-

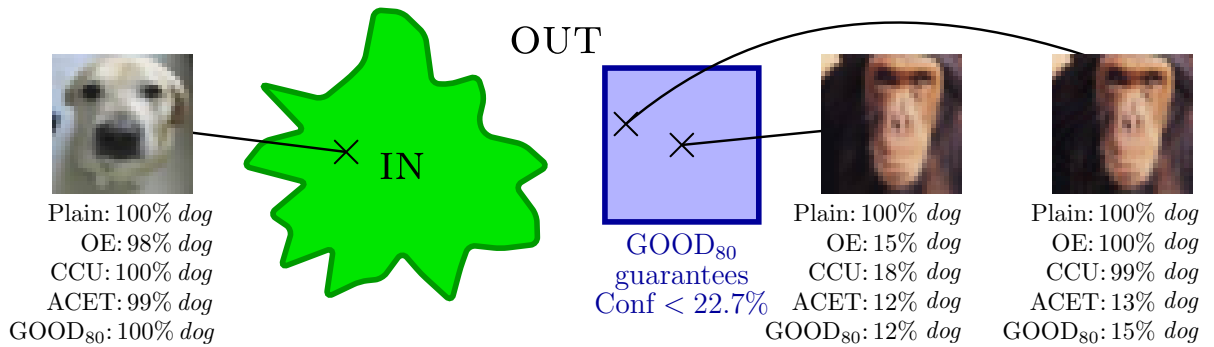


Figure 6.1: **Overconfident predictions on out-distribution inputs.** **Left:** On the in-distribution CIFAR10 all methods have similar high confidence on the image of a *dog*. **Middle:** For the out-distribution image of a *chimpanzee* from CIFAR100 the plain model is overconfident while an out-distribution aware method like OE produces low confidence. **Right:** When maximizing the confidence inside the l_∞ -ball of radius 0.01 around the *chimpanzee* image (for the OE model), OE as well as CCU become overconfident (right image). ACET and our GOOD₈₀ perform well in having empirical low confidence, but only GOOD₈₀ guarantees that the confidence in the l_∞ -ball of radius 0.01 around the *chimpanzee* image (middle image) is less than 22.7% for any class (note that 10% corresponds to maximal uncertainty as CIFAR10 has 10 classes).

dence even on near out-distributions; e.g., we get state-of-the-art results on separating letters from EMNIST from digits in MNIST even though the digit classifier has never seen any images of letters at training time. In particular, the guarantees for the training out-distribution generalize to other out-distribution datasets. In contrast to classifiers which have certified adversarial robustness on the in-distribution, GOOD has the desirable property of achieving provable guarantees for OOD detection with almost no loss in accuracy on the in-distribution task compared to plain models with the same architecture, even on datasets like CIFAR10.

6.2 IBP for OOD

Our goal is to minimize the confidence of the classifier not only on the out-distribution images themselves but in a whole neighborhood around them. For this purpose, we first derive bounds on the maximal confidence on some l_∞ -ball around a given point. In certified adversarial robustness, IBP (Gowal *et al.*, 2018) currently leads to the best guarantees for deterministic classifiers under the l_∞ -threat model. While other methods for deriving guarantees yield tighter bounds (Wong and Kolter, 2018; Mirman *et al.*, 2018), they are not easily scalable and, when optimized, the bounds given by IBP have been shown to be very tight (Gowal *et al.*, 2018).

Recall from Section 1.3.2 that Interval Bound Propagation (IBP) (Gowal *et al.*, 2018) provides entrywise lower and upper bounds $\underline{x}^{(l)}$ resp. $\bar{x}^{(l)}$ for the output $x^{(l)}$ of the l -th layer of an

L -layer neural network, given upper and lower bounds on the previous layer's outputs:

$$\bar{x}^{(l)} = \sigma \left(W_+^{(l)} \bar{x}^{(l-1)} + W_-^{(l)} \underline{x}^{(l-1)} + b^{(l)} \right), \quad (6.1)$$

$$\underline{x}^{(l)} = \sigma \left(W_+^{(l)} \underline{x}^{(l-1)} + W_-^{(l)} \bar{x}^{(l-1)} + b^{(l)} \right). \quad (6.2)$$

The recursion starts by assuming that the input x is varied in the l_∞ -ball of radius ε , i.e. $\bar{x}^{(0)} = x + \varepsilon \mathbf{1}$ and $\underline{x}^{(0)} = x - \varepsilon \mathbf{1}$. Note that the derivation in (Gowal *et al.*, 2018) is slightly different, but the bounds are the same. The forward propagation of the bounds is of similar nature as a standard forward pass and back-propagation wrt the weights is straightforward.

6.2.1 Upper bound on the confidence in terms of the logits

The log confidence of the model at x can be written as

$$\log \text{Conf}(x) = \max_{k=1, \dots, K} \log \frac{e^{f_k(x)}}{\sum_{l=1}^K e^{f_l(x)}} = \max_{k=1, \dots, K} -\log \sum_{l=1}^K e^{f_l(x) - f_k(x)}. \quad (6.3)$$

Note that each $\underline{x}^{(l)}$, $\bar{x}^{(l)}$ and $x^{(l)}$ is actually a function of $x = x^{(0)}$ which we will write explicitly in the derivation below. We assume that the last layer is affine: $f(x) = W^{(L)} \cdot x^{(L-1)} + b^{(L)}$. We calculate the upper bounds of all K^2 logit differences as:

$$\begin{aligned} \max_{\|x' - x\|_\infty \leq \varepsilon} f_k(x') - f_\ell(x') &= \max_{\|x' - x\|_\infty \leq \varepsilon} W_{k,:}^{(L)} \cdot x^{(L-1)}(x') + b_k^{(L)} - W_{\ell,:}^{(L)} \cdot x^{(L-1)}(x') - b_\ell^{(L)} \\ &= \max_{\|x' - x\|_\infty \leq \varepsilon} (W_{k,:}^{(L)} - W_{\ell,:}^{(L)}) \cdot x^{(L-1)}(x') + b_k^{(L)} - b_\ell^{(L)} \\ &\leq \left(W_{k,:}^{(L)} - W_{\ell,:}^{(L)} \right)_+ \cdot \bar{x}^{(L-1)}(x) \\ &\quad + \left(W_{k,:}^{(L)} - W_{\ell,:}^{(L)} \right)_- \cdot \underline{x}^{(L-1)}(x) + b_k^{(L)} - b_\ell^{(L)} \\ &=: \overline{f_k(x) - f_\ell(x)} \end{aligned} \quad (6.4)$$

where $W_{k,:}^{(L)}$ denotes the k -th row of $W^{(L)}$. Note that this upper bound on the logit difference can be negative and is zero for $\ell = k$. Using this upper bound on the logit difference in Equation (6.3), we obtain an upper bound on the log confidence:

$$\max_{\|x' - x\|_\infty \leq \varepsilon} \log \text{Conf}(x') \leq \max_{k=1, \dots, K} -\log \sum_{\ell=1}^K e^{-\overline{f_k(x) - f_\ell(x)}}. \quad (6.5)$$

We use the bound in (6.5) to evaluate the guarantees on the confidences for given out-distribution datasets. However, minimizing it directly during training leads to numerical problems, especially at the beginning of training, when the upper bounds $\overline{f_k(x) - f_\ell(x)}$ are very large for

$\ell \neq k$, which makes training numerically infeasible. Instead, we rather upper bound the log confidence again by bounding the sum inside the negative log from below with K times its lowest term:

$$\begin{aligned} \max_{k=1,\dots,K} -\log \sum_{\ell=1}^K e^{-\overline{f_k(x)-f_\ell(x)}} &\leq \max_{k=1,\dots,K} -\log \left(K \cdot \min_{\ell=1,\dots,K} e^{-\overline{f_k(x)-f_\ell(x)}} \right) \\ &= \max_{k,\ell=1,\dots,K} \overline{f_k(x)-f_\ell(x)} - \log K. \end{aligned} \quad (6.6)$$

While this bound can considerably differ from the potentially tighter bound of Equation (6.5), it is often quite close as one term in the sum dominates the others. Moreover, both bounds have the same global minimum when all logits are equal over the l_∞ -ball. We omit the constant $\log K$ in the following as it does not matter for training.

Unfortunately, the direct minimization of the upper bound in (6.6) is still difficult, in particular for more challenging in-distribution datasets like SVHN and CIFAR10, as the bound $\max_{k,\ell=1,\dots,K} \overline{f_k(x)-f_\ell(x)}$ can be several orders of magnitude larger than the in-distribution loss. Therefore, we use the logarithm of this quantity. However, we also want to have a more fine-grained optimization when the upper bound becomes small in the later stage of the training. Thus we define the Confidence Upper Bound loss \mathcal{L}_{CUB} for an OOD input as

$$\mathcal{L}_{\text{CUB}}(x; \varepsilon) := \log \left(\frac{\left(\max_{k,\ell=1,\dots,K} \overline{f_k(x)-f_\ell(x)} \right)^2}{2} + 1 \right), \quad (6.7)$$

where we have omitted the implicit dependence on ε on the right-hand side. Note that for small a $\log(\frac{a^2}{2} + 1) \approx \frac{a^2}{2}$ and thus we achieve the more fine-grained optimization with an l_2 -type of loss in the later stages of training which tries to get all upper bounds small. The overall objective of fully applied Guaranteed OOD Detection training (GOOD₁₀₀) is the minimization of

$$\frac{1}{N} \sum_{i=1}^N \mathcal{L}_{\text{CE}}(x_i, y_i) + \frac{\kappa}{M} \sum_{j=1}^M \mathcal{L}_{\text{CUB}}(z_j; \varepsilon), \quad (6.8)$$

where, as in previous chapters, $(x_i, y_i)_{i=1}^N$ is the in-distribution training set and $(z_j)_{j=1}^M$ the out-distribution. The hyper-parameter κ determines the relative magnitude of the two loss terms. During training we slowly increase this value and ε in order to further stabilize the training with GOOD.

6.2.2 Quantile-GOOD: trade-off between clean and guaranteed AUC

Training models by minimizing (6.8) means that the classifier gets severely punished if *any* training OOD input receives a high confidence upper bound. If OOD inputs exist to which the classifier already assigns high confidence without even considering the worst case, e.g. as these inputs share features with the in-distribution, it makes little sense to enforce low confidence guarantees. Later in the experiments we show that for difficult tasks like CIFAR10 this can happen. In such cases the normal AUC for OOD detection gets worse as the high loss of the out-distribution part effectively leads to low confidence on a significant part of the in-distribution which is clearly undesirable.

Hence, for OOD inputs x which are not clearly distinguishable from the in-distribution, it is preferable to just have the “normal” loss $\mathcal{L}_{\text{CUB}}(z_j; 0)$ without considering the worst case. We realize this by enforcing the loss with the guaranteed upper bounds on the confidence just on some quantile of the easier OOD inputs, namely the ones with the lowest guaranteed out-distribution loss $\mathcal{L}_{\text{CUB}}(z; \varepsilon)$. We first order the OOD training set by the potential loss $\mathcal{L}_{\text{CUB}}(z; \varepsilon)$ of each sample in ascending order π , that is $\mathcal{L}_{\text{CUB}}(z_{\pi_1}) \leq \mathcal{L}_{\text{CUB}}(z_{\pi_2}) \leq \dots \leq \mathcal{L}_{\text{CUB}}(z_{\pi_M})$. We then apply the loss $\mathcal{L}_{\text{CUB}}(z; \varepsilon)$ to the lower quantile q of the points (the ones with the smallest loss $\mathcal{L}_{\text{CUB}}(z; \varepsilon)$) and take $\mathcal{L}_{\text{CUB}}(z; 0)$ for the remaining samples, which means no worst-case guarantees on the confidence are enforced:

$$\frac{1}{N} \sum_{i=1}^N \mathcal{L}_{\text{CE}}(x_i, y_i) + \frac{\kappa}{M} \sum_{j=1}^{\lfloor q \cdot M \rfloor} \mathcal{L}_{\text{CUB}}(z_{\pi_j}; \varepsilon) + \frac{\kappa}{M} \sum_{j=\lfloor q \cdot M \rfloor + 1}^M \mathcal{L}_{\text{CUB}}(x_{\pi_j}; 0). \quad (6.9)$$

During training we do this ordering on each batch consisting of out-distribution images. On CIFAR10, where the out-distribution dataset 80M Tiny Images is closer to the in-distribution, the quantile GOOD-loss allows us to choose the trade-off between clean and guaranteed AUC for OOD detection, similar to the trade-off between clean and robust accuracy in adversarial robustness.

6.3 Experiments

We provide experimental results for image recognition tasks with MNIST (LeCun *et al.*, 1998), SVHN (Netzer *et al.*, 2011) and CIFAR10 (Krizhevsky and Hinton, 2009) as in-distribution datasets. We first discuss the training details, hyperparameters and evaluation before we present the results of GOOD and competing methods. For the training out-distribution, we use 80 Million Tiny Images (80M) (Torralba *et al.*, 2008) As usual, all methods get the same out-distribution for training and we are *neither* training *nor* adapting hyperparameters for each OOD dataset separately as in some previous work. At <https://gitlab.com/Bitterwolf/GOOD> you can find the exact implementation.

6.3.1 Training

For all experiments, we use deep convolutional neural networks consisting of convolutional, affine and ReLU layers. For MNIST, we use the large architecture from (Gowal *et al.*, 2018), and for SVHN and CIFAR10 a similar but deeper and wider model. The layer structure is laid out in Table 6.1. Data augmentation is applied to both in- and out-distribution images during training. For MNIST we use random crops to size 28×28 with padding 4 and for SVHN and CIFAR10 random crops with padding 4 as well as the quite aggressive augmentation AutoAugment (Cubuk *et al.*, 2019). Additionally, we apply random horizontal flips for CIFAR10.

Table 6.1: Model architectures used for MNIST (L), SVHN (XL) and CIFAR-10 (XL) experiments. Each convolutional and non-final affine layer is followed by a ReLU activation. All convolutions use a kernel size of 3, a padding of 1, and stride of 1, except for the third convolution which has stride=2.

L	XL
Conv2d(64)	Conv2d(128)
Conv2d(64)	Conv2d(128)
Conv2d(128) _{s=2}	Conv2d(256) _{s=2}
Conv2d(128)	Conv2d(256)
Conv2d(128)	Conv2d(256)
Linear(512)	Linear(512)
Linear(10)	Linear(512)
	Linear(10)

As radii for the l_∞ -perturbation model on the out-distribution we use $\varepsilon = 0.3$ for MNIST, $\varepsilon = 0.03$ for SVHN and $\varepsilon = 0.01$ for CIFAR10 (note that $0.01 > \frac{2}{255} \approx 0.0078$). The chosen $\varepsilon = 0.01$ for CIFAR10 is so small that the changes are hardly visible (see Figure 7.1). As parameter κ for the trade-off between cross-entropy loss and the GOOD regularizer in (6.8) and (6.9), we set $\kappa = 0.3$ for MNIST and $\kappa = 1$ for SVHN and CIFAR10.

In order to explore the potential trade-off between the separation of in- and out-distribution for clean and perturbed out-distribution inputs (clean AUCs vs guaranteed AUCs - see below), we train GOOD models for different quantiles $q \in [0, 1]$ in (6.9) which we denote as GOOD_Q in the following. Here, $Q = 100q$ is the percentage of out-distribution training samples for which we minimize the guaranteed upper bounds on the confidence of the neural network in the l_∞ -ball of radius ε around the out-distribution point during training. Note that GOOD_{100} corresponds to (6.8) where we minimize the guaranteed upper bound on the worst-case confidence for all out-distribution samples, whereas GOOD_0 can be seen as a variant of OE or CEDA. A training batch consists of 128 in- and 128 out-distribution samples.

As is the case with IBP training (Gowal *et al.*, 2018) for certified adversarial robustness, we have observed that the inclusion of IBP bounds can make the training unstable or cause it to fail completely. This can happen for our GOOD training despite the logarithmic damping in the \mathcal{L}_{CUB} loss in (6.7). Thus, in order to further stabilize the training similar to (Gowal *et al.*, 2018), we use linear ramp up schedules for ε and κ . For the MNIST experiments, we use as optimizer SGD with 0.9 Nesterov momentum, with an initial learning rate of $\frac{0.005}{128}$ that is divided by 5 after 50, 100, 200, 300 and 350 epochs, with a total number of 420 training epochs. Weight decay (l_2) is set to 0.05 for MNIST and 0.005 for SVHN and CIFAR-10. For the GOOD, CEDA and OE runs, the first two epochs only use in-distribution \mathcal{L}_{CE} ; over the next 100 epochs, the value of κ is ramped up linearly from zero to its final value of 0.3 for GOOD/OE and 1.0 for CEDA, where it stays for the remaining 318 epochs. The ε value in the \mathcal{L}_{CUB} loss for GOOD is also increased linearly, starting at epoch 10 and reaching its final value of 0.3 on epoch 130. CCU is trained using the publicly available code from (Meinke and Hein, 2020), where we modify the architecture, learning rate schedule and data augmentation to be the same as OE. The initial learning rate for the Gaussian mixture models is $1e - 5/\text{batchsize}$ and gets dropped at the same epochs as the neural network learning rate. Our more aggressive data augmentation implies that our underlying Mahalanobis metric is not the same as they used in (Meinke and Hein, 2020). The ACET model for MNIST is warmed up with two epochs on the in-distribution only, then four with $\kappa = 1.0$ and $\varepsilon = 0$, and the full ACET loss with $\kappa = 1.0$ and $\varepsilon = 0.3$ for the remaining epochs. The reason why we chose a smaller κ of 0.3 for the MNIST GOOD runs is that considering the large ε for which guarantees are enforced, training with higher κ values makes training unstable without improving any validation results.

For the SVHN and CIFAR-10 baseline models, we used the ADAM optimizer (Kingma and Ba, 2014) with initial learning rate $\frac{0.01}{128}$ for SVHN and $\frac{0.1}{128}$ for CIFAR-10 that was divided by 5 after 30 and 100 epochs, with a total number of 420 training epochs. For OE, κ is increased linearly from zero to one between epochs 60 and 360. The same holds for CCU which again uses the same hyperparameters as OE. Again, ACET is warmed up with two in-distribution-only and four OE epochs. Then it is trained with $\kappa = 1.0$ and $\varepsilon = 0.03/0.01$ (SVHN/CIFAR-10), with a shorter training time of 100 epochs (the same number as used in (Hein *et al.*, 2019)).

In line with the experiences reported in (Gowal *et al.*, 2018) and (Zhang *et al.*, 2020a), for GOOD training on SVHN and CIFAR-10 longer training schedules with slower ramping up of the \mathcal{L}_{CUB} loss are necessary, as adding the out-distribution loss defined in Equation (6.7) to the training objective at once will overwhelm the in-distribution cross-entropy loss and cause the model to collapse to uniform predictions for all inputs, without recovery. In order to reduce warm-up time, we use a pre-trained CEDA model for initialization and train for 900 epochs. The learning rate is $1e-4$ in the beginning and is divided by 5 after epochs 450, 750 and 850. Due to the pre-training, we begin training with a small κ and already start with non-zero ε after epoch 4. Then, ε is increased linearly to its final value of 0.03 for SVHN and 0.01 for CIFAR-10, which is reached at epoch 204. Simultaneously, κ is increased linearly with a virtual starting point at epoch -2 to its final value of 1.0 at epoch 298.

6.3.2 Evaluation

We compare a normally trained model (Plain), the state-of-the-art OOD detection method OE, CEDA (Hein *et al.*, 2019) and ACET. Recall from Chapter 2 that CEDA works very similarly to OE, so we omit it in the figures for better readability. The ε -radii for the l_∞ -balls are the same for ACET and GOOD. For CCU we use the same models as in Chapter 5.

For each method, we compute the test accuracy on the in-distribution task, and for various out-distribution datasets (not seen during training) we report the area under the receiver operating characteristic curve (AUC) as a measure for the separation of in- from out-distribution samples based on the predicted confidences on the test sets. For the accuracy, AUC and GAUC evaluations in Table 6.2 the test splits of each (non-noise) dataset were used, with the following numbers of samples: 10,000 for MNIST, FashionMNIST, CIFAR-10, CIFAR-100 and Uniform Noise; 20,800 for EMNIST Letters; 26,032 for SVHN; 300 for LSUN Classroom. Due to the computational cost of the employed attacks, the AAUC values are based on subsets of 1000 samples for each dataset. As OOD evaluation sets we use FashionMNIST (Xiao *et al.*, 2017), the Letters of EMNIST (Cohen *et al.*, 2017), grayscale CIFAR10, and Uniform Noise for MNIST, and CIFAR100 (Krizhevsky and Hinton, 2009), CIFAR10/SVHN, LSUN Classroom (Yu *et al.*, 2015), and Uniform Noise for SVHN/CIFAR10.

AAUC: We are particularly interested in the worst case OOD detection performance of all methods under the l_∞ -perturbation model for the out-distribution. For this purpose, we compute the **adversarial AUC (AAUC)** and the **guaranteed AUC (GAUC)**. These AUCs are based on the maximal confidence in the l_∞ -ball of radius ε around each out-distribution image. As in Chapter 4, for the adversarial AUC, we compute a lower bound on the maximal confidence in the l_∞ -ball by using Auto-PGD (Croce and Hein, 2020b) for maximizing the confidence of the classifier inside the intersection of the l_∞ -ball and the image domain $[0, 1]^d$. Since Auto-PGD has been designed for finding adversarial samples around the in-distribution, we change the objective of Auto-PGD to be the confidence of the classifier. We use Auto-PGD with 500 steps and 5 random restarts which is a quite strong attack. By default, the random initialization is drawn uniformly from the ε -ball. However, we found that for MNIST the attack very often got stuck for our GOOD models, because a large random perturbation of size 0.3 would move the sample directly into a region of the input space where the model is completely flat and thus no gradients are available (in this sense adversarial attacks on OOD inputs are more difficult than usual adversarial attacks on the in-distribution). We instead use a modified version of the attack for MNIST which starts within short distance of the original point. Thus, as initialization we use a random perturbation from $[-0.01, 0.01]^d$ (note that for our evaluation on CIFAR10, this choice coincides with the default settings).

Nevertheless, for MNIST most out-distribution points lie in regions where the predictions of our GOOD models are flat, i.e. the gradients are exactly zero. Because of this, Auto-PGD is unable to effectively explore the search space around those points. Thus, for MNIST we created another adaptive attack which partially circumvents these issues. First, we use an initialization scheme that mitigates lack of gradients by increasing the contrast as much

as the threat model allows. All pixel values x_i that lie above $1 - \varepsilon$ get set to $x_i = 1$ and all values $x_i \leq 1 - \varepsilon$ get set to $\max\{0, x_i - \varepsilon\}$. In our experience these points are more likely to yield gradients, so we use them as initialization for a 200-step PGD attack with backtracking, adaptive step size selection and momentum of 0.9. Concretely, we use a step size of 0.1, and whenever a PGD step does not increase the confidence we backtrack and halve the step size. After every successful gradient step we multiply the step size by 1.1. Using backtracking and adaptive step size is necessary because otherwise one can easily step into regions where gradient information is no longer available.

Additionally, to further mitigate the problem of gradient-masking at initialization, for each model we use the final best points of all other models and use those as starting points for the same monotone PGD as described before. We use the sample-wise worst-case confidence to compute the final AAUC. Especially CEDA displays much higher apparent robustness if one omits the transfer attacks. Surprisingly, in this respect CEDA behaves very differently from OE, even though they pursue very similar objectives during training.

We report the per-sample worst-case across attacks. Note that despite our effort of developing strong adaptive attacks which are specific for our robust OOD detection scenario, it might still be that the AAUC of some methods is overestimated. This again shows how important it is to get provable guarantees.

GAUC: For the guaranteed AUC, we compute an upper bound on the confidence in the intersection of the l_∞ -ball with the image domain $[0, 1]^d$ via IBP using (6.5) for the full test set. These worst case/guaranteed confidences for the out-distributions are then used for the AUC computation.

The story is much more complicated for CCU, which we introduced in the previous Chapter. Recall that CCU’s bounds do hold on such far-away datasets, but do not generalize to inputs relatively close to the in distribution, like for example CIFAR-10 vs. CIFAR-100. Moreover, even in the regime where CCU yields meaningful guarantees, they are given in terms of a data-dependent Mahalanobis distance rather than the l_∞ -distance. Nonetheless, due to norm equivalences, one can, in principle, still extract l_∞ -guarantees from CCU. We evaluate the CCU guarantees as follows. We use Corollary 2 which states that for a CCU model that is written as

$$\hat{p}(y|x) = \frac{\hat{p}(y|x, i)\hat{p}(x|i) + \frac{1}{K}\hat{p}(x|o)}{\hat{p}(x|i) + \hat{p}(x|o)} \quad (6.10)$$

with $\hat{p}(y|x, i)$ being the softmax output of a neural network and $\hat{p}(x|i)$ and $\hat{p}(x|o)$ Gaussian mixture models for in- and out-distribution, one can bound the confidence in a certain neighborhood around any point $x \in \mathbb{R}^d$ via

$$\max_{d_M(\hat{x}, x) \leq R} p(y|x) \leq \frac{1}{K} \frac{1 + Kb(x, R)}{1 + b(x, R)}. \quad (6.11)$$

Here $b : \mathbb{R}^d \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is a positive function that increases monotonically in the radius R and

that depends on the parameters of the Gaussian mixture models (details in (Meinke and Hein, 2020)). The metric $d_M : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}_+$ that we used for the CCU model is given as

$$d_M(x, y) = \|C^{-\frac{1}{2}}(x - y)\|, \quad (6.12)$$

where C is a regularized version of the covariance matrix, calculated on the augmented in-distribution data. Note that this Mahalanobis metric is strongly equivalent to the metric induced by the l_2 -norm and consequently to the metric induced by the l_∞ -norm. By computing the equivalence constants between these metrics we can extract the l_∞ -guarantees that are implicit in the CCU model. Geometrically speaking, we compute the size of an ellipsoid (its shape determined by the eigenvalues of C) that is large enough to fit a cube inside it with a radius given by our threat model $r = 0.3$ or $r = 0.01$, respectively. Via norm equivalences one has

$$d_M(x, y) \leq \sqrt{\lambda_1} d_2(x, y) \leq \sqrt{d\lambda_1} d_\infty(x, y) \leq \sqrt{d\lambda_1} r, \quad (6.13)$$

where λ_1 is the largest eigenvalue of C . This means that the confidence upper bounds from (6.11) on a Mahalanobis-ball of radius $R = (d\lambda)^{\frac{1}{2}}r$ automatically apply to an l_∞ -ball of radius r . However, the covariance matrix C is highly ill-conditioned, which means that λ_1 is fairly high. On top of that, in high dimensions \sqrt{d} is big as well so that in practice the required radius R becomes too large for CCU to certify meaningful guarantees. Even on uniform noise, the upper bounds were larger than the highest confidence on the in-distribution test set, with the consequence that there are no lower-bounds on the AAUC. However, we want to stress that at least for uniform noise the lack of guarantees of CCU is due to the incompatibility of the threat models used in this chapter and in Chapter 5.

6.3.3 Results

In Table 6.2 we present the results on all datasets.

GOOD is provably better than OE/CEDA with regard to worst case OOD detection. We note that for almost all OOD datasets GOOD achieves non-trivial GAUCs. Thus the guarantees generalize from the training out-distribution 80M to the test OOD datasets. For the easier in-distributions MNIST and SVHN, which are more clearly separated from the out-distribution, the overall best results are achieved for GOOD₁₀₀. For CIFAR10, the clean AUCs of GOOD₁₀₀ are low even when compared to plain training. Arguably the best trade-off for CIFAR10 is achieved by GOOD₈₀. Note that the guaranteed AUC (GAUC) of these models is always better than the adversarial AUC (AAUC) of OE/CEDA (except for EMNIST). Thus it is fair to say that the worst-case OOD detection performance of GOOD is provably better than that of OE/CEDA. As expected, ACET yields good AAUCs but has no guarantees. We already discussed the failure of CCU to produce guarantees in the section above. It is notable that GOOD₁₀₀ has close to perfect guaranteed OOD detection performance for MNIST on CIFAR10/uniform noise and for SVHN on **all** out-distribution datasets.

Table 6.2: Accuracies as well as AUC, adversarial AUC (AAUC) and guaranteed AUC (GAUC) values for the MNIST, SVHN and CIFAR10 in-distributions with respect to several unseen out-distributions. The radii of the l_∞ -ball for the worst case OOD detection are 0.3 on MNIST and 0.01 on SVHN/CIFAR10. The GAUC of GOOD₁₀₀ on MNIST/SVHN resp. GOOD₈₀ on CIFAR10 is better than the corresponding AAUC of OE and CEDA on almost all OOD datasets (except EMNIST).

in: MNIST $\epsilon = 0.3$													
Method	Acc.	FashionMNIST			EMNIST Letters			CIFAR10			Uniform Noise		
		AUC	AAUC	GAUC	AUC	AAUC	GAUC	AUC	AAUC	GAUC	AUC	AAUC	GAUC
Plain	99.4	98.0	34.2	0.0	88.0	31.4	0.0	98.8	36.6	0.0	99.1	36.5	0.0
CEDA	99.4	99.9	82.1	0.0	92.6	52.8	0.0	100.0	95.1	0.0	100.0	100.0	0.0
OE	99.4	99.9	76.8	0.0	92.7	50.9	0.0	100.0	92.4	0.0	100.0	100.0	0.0
ACET	99.4	100.0	98.4	0.0	95.9	61.5	0.0	100.0	99.3	0.0	100.0	100.0	0.0
CCU	99.5	100.0	76.6	0.0	92.9	3.1	0.0	100.0	98.9	0.0	100.0	100.0	0.0
GOOD ₀	99.5	99.9	82.3	0.0	92.9	55.0	0.0	100.0	94.7	0.0	100.0	100.0	0.0
GOOD ₄₀	99.0	99.8	88.0	29.1	95.7	56.6	0.0	100.0	97.7	65.2	100.0	100.0	100.0
GOOD ₈₀	99.1	99.8	90.3	55.5	97.9	63.1	3.4	100.0	98.4	94.7	100.0	100.0	100.0
GOOD ₁₀₀	98.7	100.0	96.5	78.0	99.0	53.8	3.3	100.0	99.9	99.4	100.0	100.0	100.0

in: SVHN $\epsilon = 0.03$													
Method	Acc.	CIFAR100			CIFAR10			LSUN Classroom			Uniform Noise		
		AUC	AAUC	GAUC	AUC	AAUC	GAUC	AUC	AAUC	GAUC	AUC	AAUC	GAUC
Plain	95.5	94.9	11.3	0.0	95.2	11.1	0.0	95.7	14.1	0.0	99.4	57.9	0.0
CEDA	95.3	99.9	63.9	0.0	99.9	68.7	0.0	99.9	80.7	0.0	99.9	99.3	0.0
OE	95.5	100.0	60.2	0.0	100.0	62.5	0.0	100.0	77.3	0.0	100.0	98.2	0.0
ACET	96.0	100.0	99.4	0.0	100.0	99.5	0.0	100.0	99.8	0.0	99.9	96.3	0.0
CCU	95.7	100.0	52.5	0.0	100.0	56.8	0.0	100.0	72.1	0.0	100.0	100.0	0.0
GOOD ₀	97.0	100.0	61.0	0.0	100.0	60.0	0.0	100.0	60.8	0.0	100.0	82.5	0.0
GOOD ₄₀	96.3	99.5	81.6	46.0	99.5	85.0	50.6	99.5	95.1	55.7	99.5	99.5	99.4
GOOD ₈₀	96.3	100.0	93.5	87.7	100.0	95.3	91.3	100.0	98.8	96.7	100.0	100.0	99.7
GOOD ₁₀₀	96.3	99.6	97.7	97.3	99.7	98.4	98.1	99.9	99.2	98.9	100.0	99.9	99.8

in: CIFAR10 $\epsilon = 0.01$													
Method	Acc.	CIFAR100			SVHN			LSUN Classroom			Uniform Noise		
		AUC	AAUC	GAUC	AUC	AAUC	GAUC	AUC	AAUC	GAUC	AUC	AAUC	GAUC
Plain	90.1	84.3	13.0	0.0	87.7	10.6	0.0	88.9	13.6	0.0	90.8	56.4	0.0
CEDA	88.6	91.8	31.9	0.0	97.9	25.7	0.0	98.9	53.9	0.0	97.3	70.5	0.0
OE	90.7	92.4	11.0	0.0	97.6	3.7	0.0	98.9	20.0	0.0	98.7	75.7	0.0
ACET	89.3	90.7	74.5	0.0	96.6	88.0	0.0	98.3	91.2	0.0	99.7	98.9	0.0
CCU	91.6	93.0	23.3	0.0	97.1	14.8	0.0	99.3	38.2	0.0	100.0	100.0	0.0
GOOD ₀	89.8	92.9	22.5	0.0	97.0	12.8	0.0	98.3	48.4	0.0	96.3	95.6	0.0
GOOD ₄₀	89.5	89.6	38.2	24.8	95.4	38.0	24.9	96.0	62.0	27.4	92.1	89.9	89.8
GOOD ₈₀	90.1	85.6	48.2	42.3	94.0	41.4	38.0	93.3	66.9	55.2	95.8	95.4	95.3
GOOD ₁₀₀	90.1	70.0	54.7	54.2	75.5	58.9	56.9	75.2	61.5	61.0	99.5	99.2	99.0

GOOD achieves certified OOD performance with almost no loss in accuracy. While there is a small drop in clean accuracy for MNIST, on SVHN, with 96.3% GOOD₁₀₀ surprisingly has a better clean accuracy than all competing methods. On CIFAR10, GOOD₈₀ achieves an accuracy of 90.1% which is better than ACET and only slightly worse than CCU and OE. This is remarkable as we are not aware of any model with certified *adversarial robustness on the in-distribution* which gets even close to this range; e.g. IBP (Gowal *et al.*, 2018) achieves an accuracy of 85.2% on SVHN with $\epsilon = 0.01$ (we have 96.3%), on CIFAR10 with $\epsilon = \frac{2}{255}$ they get 71.2% (we have 90.1%). Previous certified methods had even worse clean accuracy. Since a significant loss in prediction performance is usually not acceptable, certified methods have not yet had much practical impact. Thus we think it is an encouraging and interesting observation that properties different from adversarial robustness like worst-case out-of-distribution detection can be certified without suffering much in accuracy. In particular, it is quite surprising that certified methods can be trained effectively with aggressive data augmentation like AutoAugment.

Trade-off between clean and guaranteed AUC via Quantile-GOOD. As discussed above, for the CIFAR10 experiments, our training out-distribution contains images from in-distribution classes. This seems to be the reason why GOOD₁₀₀ suffers from a significant drop in clean AUC, as the only way to ensure small loss \mathcal{L}_{CUB} , if in- and out-distribution can partially not be distinguished, is to reduce also the confidence on the in-distribution. This conflict is resolved via GOOD₈₀ which both has better clean AUCs. It is an interesting open question if similar trade-offs can also be useful for certified adversarial robustness.

EMNIST: distinguishing letters from digits without ever having seen letters. GOOD₁₀₀ achieves an excellent AUC of 99.0% for the letters of EMNIST which is, up to our knowledge, state-of-the-art. Indeed, an AUC of 100% should not be expected as even for humans some letters like *i* and *l* are indistinguishable from digits. This result is quite remarkable as GOOD₁₀₀ has never seen letters during training. Moreover, as the AUC just distinguishes the separation of in- and out-distribution based on the confidence, we provide the mean confidence on all datasets in Figure 6.2 we show some samples from EMNIST together with their prediction/confidences for all models. GOOD₁₀₀ has a mean confidence of 98.4% on MNIST but only 27.1% on EMNIST in contrast to ACET with 75.0%, OE 87.9% and Plain 91.5%. This shows that while the AUC’s of ACET and OE are good for EMNIST, these methods are still highly overconfident on EMNIST. Only GOOD₁₀₀ produces meaningful higher confidences on EMNIST, when the letter has clear features of the corresponding digit.

We see that GOOD₁₀₀ produces low confidences for most letters when they show no digit-specific features. Interestingly it even rejects some letters that could easily be mistaken for digits by humans (“o”). The mean confidence values of the same selection of MNIST models for each letter of the alphabet for EMNIST are plotted in Figure 6.3. We observe that the mean confidence often aligns with the intuitive likeness of a letter with some digit: GOOD₁₀₀ has the highest mean confidence on the letter inputs “i” and “l”, which in many cases do look



























Plain	9: 63.6	8: 97.9	6: 87.2	0: 100.0	2: 60.0	4: 75.6	9: 100.0	4: 100.0	1: 100.0
OE	9: 79.7	0: 26.8	0: 78.8	0: 100.0	0: 69.4	4: 80.9	9: 100.0	4: 100.0	1: 100.0
CCU	4: 96.4	8: 31.8	6: 90.6	0: 100.0	2: 81.8	4: 86.8	9: 100.0	4: 100.0	1: 100.0
ACET	4: 79.6	0: 38.4	0: 61.6	0: 99.7	2: 55.7	4: 94.4	9: 100.0	4: 98.4	1: 99.9
GOOD ₁₀₀	8: 10.0	8: 11.4	0: 16.1	8: 10.0	8: 44.7	4: 24.1	9: 57.2	8: 10.0	1: 95.9
									
Plain	5: 99.2	6: 82.1	6: 100.0	4: 99.9	7: 99.8	0: 100.0	4: 89.7	8: 90.8	8: 97.4
OE	5: 98.7	6: 67.8	6: 100.0	4: 98.8	7: 99.5	0: 100.0	1: 65.9	8: 97.5	8: 96.2
CCU	5: 96.2	8: 52.7	6: 100.0	4: 89.0	7: 99.0	0: 99.9	1: 78.7	8: 80.2	8: 78.5
ACET	5: 79.4	6: 37.6	6: 99.7	4: 48.6	7: 58.8	0: 98.8	1: 48.0	8: 87.5	8: 69.0
GOOD ₁₀₀	5: 12.1	8: 10.0	6: 10.9	4: 21.2	8: 10.0	8: 10.0	1: 20.4	8: 28.1	8: 28.6
									
Plain	5: 100.0	4: 97.6	0: 90.0	4: 99.8	6: 74.8	4: 76.9	4: 99.8	2: 100.0	
OE	5: 100.0	4: 85.1	0: 79.5	4: 99.5	4: 68.0	4: 65.2	4: 100.0	2: 100.0	
CCU	5: 100.0	4: 98.8	0: 99.7	4: 92.7	4: 72.6	4: 66.0	4: 100.0	2: 99.7	
ACET	5: 99.9	4: 82.8	0: 87.1	4: 96.9	0: 60.8	6: 50.9	4: 99.8	2: 81.8	
GOOD ₁₀₀	5: 10.1	4: 50.2	0: 11.2	4: 12.1	8: 10.0	8: 10.0	4: 74.8	8: 10.0	
									

Figure 6.2: Random samples from all letters in the out-distribution dataset EMNIST. The predictions and confidences of all methods trained on MNIST are shown on top. GOOD₁₀₀ is the only method which is **not** overconfident (e.g. “H”) unless the letter is indistinguishable from a digit (e.g. “I”).

like the digit “1”. Again, the confidence of GOOD₁₀₀ on the letter “o”, which even humans often cannot distinguish from a digit “0”, is generally low. On the other hand, “y” receives a surprisingly high confidence, compared to other letters, so we conclude that GOOD₁₀₀ uses different features than humans in order to achieve its impressive performance on EMNIST.

6.4 Conclusion

In this chapter we have shown that IBP can be used to achieve certifiably adversarially robust detection of OOD data. This allowed us to certify the common l_∞ -threat model (instead of the custom one for CCU) and to certify regions around points drawn from challenging out-distributions (instead of just uniform noise like for CCU). Remarkably, we could see that these guarantees generalize not just to unseen test samples but even unseen test distributions. We also saw evidence that this can be achieved without any loss in clean accuracy and only a mild loss in clean OOD detection performance.

Recent developments: The work of Berrada *et al.* (2021b,a) showed that non-zero GAUCs for ACET models under an l_∞ -threat could also be proved, although for smaller ϵ than considered here. The authors of (Yoon *et al.*, 2022) have proposed to use generative models to define

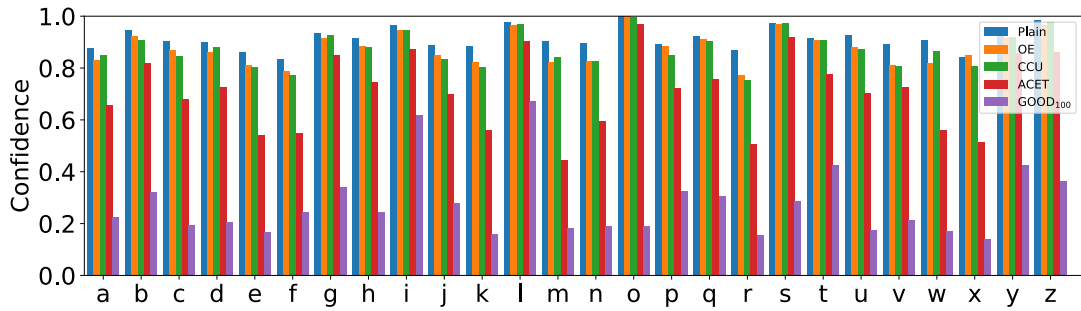


Figure 6.3: Mean confidence of different models across the classes of EMNIST-Letters. GOOD₁₀₀ only has high mean confidence on letters that can easily be mistaken for digits.

adversarial distributions which allow for semantic perturbations as well. They found that under their affine threat model (rotation, translation, shear, scaling), GOOD did not perform very well in the worst case.

Chapter 7

Provably Robust Detection of Out-of-distribution Data (almost) for free

This chapter is based on (Meinke *et al.*, 2022) which we presented at NeurIPS 2022. I came up with the idea for the paper while Julian Bitterwolf and I were experimenting with improvements to GOOD training. I ran the experiments, developed the theoretical results and was the main author of the paper. Julian Bitterwolf assisted in the writing. Matthias Hein provided guidance to the project, helped in writing it and significantly improved the formulation and proof of Theorem 4.

7.1 Introduction

We will briefly summarize our findings from the previous two chapters. In Chapter 5 we have seen that CCU enables us to not only achieve provably low confidence far away from the training data, but also implies guarantees on the adversarial robustness of this low confidence. It did so without imposing any restrictions on the network architecture, but unfortunately could only give robustness guarantees for uniform noise and only within a highly non-standard threat model. On the other hand, we showed in Chapter 6 that we can indeed use interval bound propagation to achieve provably adversarially robust low confidence even on difficult out-distribution like CIFAR10 vs. CIFAR100. This, however, came at the price of using very restricted architectures for the classifier as well as highly complex training losses and schedules.

In this chapter we will show how we can combine ideas from the previous two chapters into ProoD (Provable out-of-Distribution) which merges a certified binary discriminator for in-versus out-distribution with a classifier for the in-distribution task in a principled fashion into a joint classifier. This combines the advantages of CCU and GOOD without suffering from their respective downsides. In particular, ProoD simultaneously achieves the following:

- Guaranteed adversarially robust OOD detection via certified upper bounds on the confidence in l_∞ -balls around OOD samples.
- Additionally, it provably prevents the asymptotic overconfidence of deep neural networks.

Table 7.1: ProoD combines desirable properties of existing (adversarially robust) OOD detection methods. It has high test accuracy and standard OOD detection performance (as OE) and has worst-case guarantees if the out-distribution samples are adversarially perturbed in an l_∞ -neighborhood to maximize the confidence (see Section 7.4.2). Similar to CCU, it avoids the problem of asymptotic overconfidence far away from the training data.

	OE	CCU	ACET	GOOD	ProoD
High accuracy	✓	✓	(✓)		✓
High clean OOD detection performance	✓	✓	✓		✓
Adv. OOD l_∞ -robustness			(✓)	✓	✓
Adv. OOD l_∞ -certificates				✓	✓
Provably not asympt. overconfident		✓			✓

- It can be used with arbitrary architectures and has no loss in prediction performance and standard OOD detection performance.

Thus, we get provable guarantees for adversarially robust OOD detection, fix the asymptotic overconfidence (almost) for free as we have (almost) no loss in prediction and standard OOD detection performance. See Table 7.1 for a qualitative summary of ProoD’s properties in comparison to the models that we have studied in this thesis so far.

7.2 Provably Robust Detection of Out-of-distribution Data

7.2.1 Joint Model for OOD Detection and Classification

The initial setup is similar to CCU’s in Chapter 5. In our joint model we assume that there exists an in- and out-distribution where the out-distribution samples are unrelated to the in-distribution task. Thus, we can formally write the conditional distribution on the input as

$$\hat{p}(y|x) = \hat{p}(y|x, i)\hat{p}(i|x) + \hat{p}(y|x, o)\hat{p}(o|x), \quad (7.1)$$

where $\hat{p}(i|x)$ is the conditional distribution that sample x belongs to the in-distribution and $\hat{p}(y|x, i)$ is the conditional distribution for the in-distribution. We assume that OOD samples are unrelated and thus maximally un-informative to the in-distribution task, i.e. we fix $\hat{p}(y|x, o) = \frac{1}{K}$, so that the classifier can be written as

$$\hat{p}(y|x) = \hat{p}(y|x, i)\hat{p}(i|x) + \frac{1}{K}(1 - \hat{p}(i|x)). \quad (7.2)$$

We train the binary classifier $\hat{p}(i|x)$ in a certified robust fashion wrt. an l_∞ -threat model so that even adversarially manipulated OOD samples are detected. In order to avoid confusion with the multi-class classifier, we will refer to $\hat{p}(i|x)$ as a binary discriminator. In an l_∞ -ball of radius ε around $x \in \mathbb{R}^d$ and for all y we get the upper bound on the confidence of the final

classifier in Eq. (7.2):

$$\max_{\|x'-x\|_\infty \leq \epsilon} \hat{p}(y|x') \leq \max_{\|x'-x\|_\infty \leq \epsilon} \hat{p}(i|x') + \frac{1}{K}(1 - \hat{p}(i|x')) = \frac{K-1}{K} \max_{\|x'-x\|_\infty \leq \epsilon} \hat{p}(i|x') + \frac{1}{K}, \quad (7.3)$$

where we have used that $p(y|x, i) \leq 1 \forall x, y$, so we can defer the certification “work” to the binary discriminator. Using a particular constraint on the weights of the binary discriminator, we get similar asymptotic properties as for CCU but additionally get certified adversarial robustness for close out-distribution samples as with GOOD. In contrast to GOOD, this comes without architectural limits on test accuracy or non-adversarial OOD detection performance since in our model the neural network used for the in-distribution classification task $\hat{p}(y|x, i)$ is independent of the binary discriminator. Thus, we have the advantage that the classifier can use arbitrary deep neural networks and is not constrained to certifiable networks. We call our approach **Provable out-of-Distribution detector (ProoD)** and visualize its components in Figure 7.1. The intuitive idea of why ProoD can achieve adversarially robust OOD detection without loss in clean OOD detection can be explained with the behavior of the predicted probability distribution provided in Equation (7.2).

- **For clean OOD:** the classifier $\hat{p}(y|x, i)$ (trained similar to Outlier Exposure) already enforces low confidence on out-of-distribution points and thus irrespective of the values $\hat{p}(i|x)$, the resulting output of $\hat{p}(y|x)$ will be close to uniform as well and thus ProoD performs similar to Outlier Exposure.
- **For adversarial OOD:** the classifier confidence $\max_y \hat{p}(y|x, i)$ is potentially corrupted but now the binary discriminator $\hat{p}(i|x)$ kicks in and ensures that the resulting prediction $\hat{p}(y|x)$ is close to uniform.

This explains why the combination of certified discriminator and classifier works much better than the individual parts and the use of this “redundancy” is the key idea of ProoD.

7.2.2 Certifiably Robust Binary Discrimination of In- versus Out-Distribution

The first goal is to get a certifiably adversarially robust OOD detector $\hat{p}(i|x)$. We train this binary discriminator independently of the overall classifier as the training schedules for certified robustness are incompatible with the standard training schedules of normal classifiers. For this binary classification problem we use a logistic model $\hat{p}(i|x) = \frac{1}{1+e^{-g(x)}}$, where $g: \mathbb{R}^d \rightarrow \mathbb{R}$ are logits of a neural network (we denote the weights and biases of g by W_g and b_g in order to differentiate it from the classifier f introduced in the next paragraph). Let $(x_r, y_r)_{r=1}^N$ be our in-distribution training data (we use the class encoding +1 for the in-distribution and -1 for the out-distribution) and $(z_s)_{s=1}^M$ be our training out-distribution data. Then the optimization

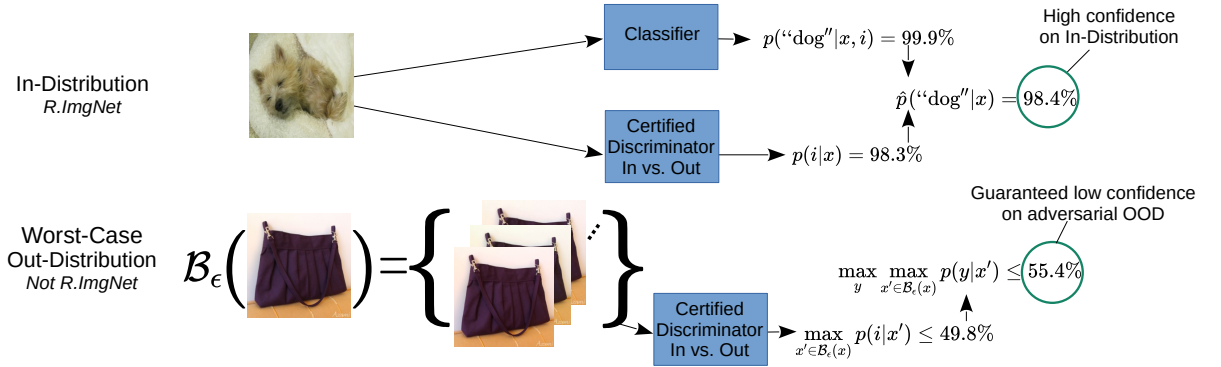


Figure 7.1: **ProoD's Architecture:** Combining the output of a classifier and a certified discriminator, see Eq. (7.1), we achieve high confidence on the in-distribution sample of a dog (R.ImgNet). The certified discriminator, see Eq. (7.3), yields an upper bound on the confidence in a ℓ_∞ -neighborhood of the shown OOD sample not belonging to any classes of R.ImgNet. ProoD achieves provable guarantees on adversarial OOD detection without loss in accuracy or clean OOD detection.

problem associated to the binary classification problem becomes:

$$\min_{\substack{g \\ W_g^{(L_g)} < 0}} \frac{1}{N} \sum_{r=1}^N \log\left(1 + e^{-g(x_r)}\right) + \frac{1}{M} \sum_{s=1}^M \log\left(1 + e^{\bar{g}(z_s)}\right), \quad (7.4)$$

where we minimize over the parameters of the neural network g under the constraint that the weights of the output layer $W_g^{(L_g)}$ are componentwise negative and $\bar{g}(z) \geq \max_{u \in B_p(z, \epsilon)} g(u)$ is an upper bound on the output of g around OOD samples for a given l_p -threat model $B_p(z, \epsilon) = \{u \in [0, 1]^d \mid \|u - z\|_p \leq \epsilon\}$. As in the previous chapter, we consider an l_∞ -threat model. This upper bound could, in principle, be computed using any certification technique but we will again use IBP. Note that this is not standard adversarial training for a binary classification problem as here we have an asymmetric situation: we want to be (certifiably) robust to adversarial manipulation on the out-distribution data but *not* on the in-distribution and thus the upper bound is only used for out-distribution samples. The negativity of the output layer's weights $W_g^{(L_g)}$ is enforced by using the parameterization $(W_g^{(L_g)})_j = -e^{h_j}$ componentwise and optimizing over h_j . In Section 7.3 we show how the negativity of $W_g^{(L_g)}$ allows us to control the asymptotic behavior of the joint classifier.

7.2.3 (Semi)-Joint Training of the Final Classifier

Given the certifiably robust model $\hat{p}(i|x)$ for the binary classification task between in- and out-distribution, we need to determine the final predictive distribution $\hat{p}(y|x)$ in Eq. (7.1). On top of the provable OOD performance that we get from Eq. (7.3), we also want to achieve SOTA

performance on unperturbed OOD data. In principle we could independently train a model for the predictive in-distribution task $\hat{p}(y|x, i)$, e.g. using outlier exposure (OE) (Hendrycks *et al.*, 2019a) or any other state-of-the-art OOD detection method and simply combine it with our $\hat{p}(i|x)$. While this does lead to models with high OOD performance that also have guarantees, it completely ignores the interaction between $\hat{p}(i|x)$ and $\hat{p}(y|x, i)$ during training. Instead we propose to train $\hat{p}(y|x, i)$ by optimizing our final predictive distribution $\hat{p}(y|x)$. Note that in order to retain the guarantees of $\hat{p}(i|x)$ we only train the parameters of the neural network $f: \mathbb{R}^d \rightarrow \mathbb{R}^K$ and need to keep $\hat{p}(i|x)$ resp. g fixed. Because g stays fixed we call this semi-joint training. We use OE (Hendrycks *et al.*, 2019a) for training $\hat{p}(y|x)$ with the cross-entropy loss and use the softmax-function in order to obtain the predictive distribution $\hat{p}_f(y|x, i) = \frac{e^{f_{y(x)}}}{\sum_k e^{f_k(x)}}$ from f :

$$\begin{aligned} & \min_f -\frac{1}{N} \sum_{r=1}^N \log(\hat{p}(y_r|x_r)) - \frac{1}{M} \sum_{s=1}^M \frac{1}{K} \sum_{l=1}^K \log(\hat{p}(l|z_s)) \\ &= \min_f -\frac{1}{N} \sum_{r=1}^N \log\left(\hat{p}_f(y_r|x_r, i)\hat{p}(i|x_r) + \frac{1}{K}(1 - \hat{p}(i|x_r))\right) \\ & \quad - \frac{1}{M} \sum_{s=1}^M \frac{1}{K} \sum_{l=1}^K \log\left(\hat{p}_f(l|z_s, i)\hat{p}(i|z_s) + \frac{1}{K}(1 - \hat{p}(i|z_s))\right), \end{aligned} \quad (7.5)$$

where the first term is the standard cross-entropy loss on the in-distribution but now for our joint model for $\hat{p}(y|x)$ and the second term enforces uniform confidence on out-distribution samples.

The loss in Eq. (7.4) implicitly weighs the in-distribution and worst-case out-distribution equally, which amounts to the assumption $p(i) = \frac{1}{2} = p(o)$. This highly conservative choice simplifies training the binary discriminator but may not reflect the expected frequency of OOD samples at test time and in effect means that $\hat{p}(i|x)$ tends to be quite low. This typically yields good guaranteed AUCs but can have a negative impact on the standard out-distribution performance. In order to better explore the trade-off of guaranteed and standard OOD detection, we repeat the above semi-joint training with different shifts of the offset parameter in the output layer

$$b' = b_g^{(L_g)} + \Delta, \quad (7.6)$$

where $\Delta \geq 0$ leads to increasing $\hat{p}(i|x)$. This shift has a direct interpretation in terms of the probabilities $p(i)$ and $p(o)$. Under the assumption that our binary discriminator g is perfect, that is

$$p(i|x) = \frac{p(x|i)p(i)}{p(x|i)p(i) + p(x|o)p(o)} = \frac{1}{1 + e^{-g(x)}}, \quad (7.7)$$

then it holds that $e^{g(x)} = \frac{p(x|i)p(i)}{p(x|o)p(o)}$. A change of the prior probabilities $\tilde{p}(i)$ and $\tilde{p}(o)$ without

changing $p(x|i)$ and $p(x|o)$ then corresponds to a novel classifier

$$e^{\tilde{g}(x)} = \frac{p(x|i)\tilde{p}(i)}{p(x|o)\tilde{p}(o)} = \frac{p(x|i)p(i)}{p(x|o)p(o)} \frac{p(o)\tilde{p}(i)}{p(i)\tilde{p}(o)} = e^{g(x)} e^{\Delta} \quad (7.8)$$

with $\Delta = \log\left(\frac{p(o)\tilde{p}(i)}{p(i)\tilde{p}(o)}\right)$. Note that $\tilde{p}(i) > p(i)$ corresponds to positive shifts. In practice, this parameter can be chosen based on the priors for a particular application. Since no such priors are available in our case we determine a suitable shift by evaluating on the training out-distribution (see Section 7.4.2). Note that we explicitly do not train the shift parameter since this way the guarantees would get lost as the classifier implicitly learns a large Δ in order to maximize the confidence on the in-distribution, thus converging to a normal outlier exposure-type classifier without any guarantees.

7.3 Guarantees on Asymptotic Confidence

In this section we show that, like CCU but unlike GOOD, our specific construction provably avoids the issue of asymptotic overconfidence that was pointed out in (Hein *et al.*, 2019). Note that the resulting guarantee (as stated in Theorem 4) is different from and in addition to the robustness guarantees discussed in the previous section (see Eq. (7.3)). The previous section dealt with providing confidence upper bounds on neighborhoods around OOD samples whereas this section deals with ensuring that a classifier’s confidence decreases asymptotically as one moves away from all training data.

As briefly mentioned in Section 1.2.2, a ReLU neural network $f : \mathbb{R}^d \rightarrow \mathbb{R}^K$ as defined in Eq. (1.11) using ReLU or leaky ReLU as activation functions, potential max-or average pooling and skip connection yields a piece-wise affine function (Arora *et al.*, 2018), i.e. there exists a finite set of polytopes $Q_r \subset \mathbb{R}^d$ with $r = 1, \dots, R$ such that $\cup_{r=1}^R Q_r = \mathbb{R}^d$ and f restricted to each of the polytopes is an affine function. Since there are only finitely many polytopes some of them have to extend to infinity and on these ones the neural network is essentially an affine classifier. This fact has been used in (Hein *et al.*, 2019) to show that ReLU networks are almost always asymptotically overconfident in the sense that if one moves to infinity the confidence of the classifier approaches 1 (instead of converging to the desirable $1/K$ as in these regions the classifier has never seen any data). The following theorem now shows that, in contrast to standard ReLU networks, our proposed joint classifier gets provably less confident in its decisions as one moves away from the training data which is a desirable property of any reasonable classifier.

The following result of (Hein *et al.*, 2019) basically says that as one moves to infinity by upscaling a vector one eventually ends up in a polytope which extends to infinity. We use this in the proof of our Theorem.

Lemma 5 (Hein *et al.* (2019)). *Let $\{Q_r\}_{r=1}^R$ be the set of convex polytopes on which a ReLU-network $f : \mathbb{R}^d \rightarrow \mathbb{R}^K$ is an affine function, that is for every $k \in \{1, \dots, R\}$ and $x \in Q_k$ there*

exists $V^k \in \mathbb{R}^{K \times d}$ and $c^k \in \mathbb{R}^K$ such that $f(x) = V^k x + c^k$. For any $x \in \mathbb{R}^d$ with $x \neq 0$ there exists $\alpha \in \mathbb{R}$ and $t \in \{1, \dots, R\}$ such that $\beta x \in Q_t$ for all $\beta \geq \alpha$.

Theorem 4. Let $x \in \mathbb{R}^d$ with $x \neq 0$ and let $g : \mathbb{R}^d \rightarrow \mathbb{R}$ be the ReLU-network of the binary discriminator (with the last activation being a non-leaky ReLU). Denote by $\{Q_r\}_{r=1}^R$ the finite set of polytopes on which g is affine (exists by Lemma 5). Denote by Q_t the polytope such that $\beta x \in Q_t$ for all $\beta \geq \alpha$ and let $x^{(L-1)}(z) = Uz + d$ with $U \in \mathbb{R}^{n_{L-1} \times d}$ and $d \in \mathbb{R}^{n_{L-1}}$ be the output of the pre-logit layer of g for $z \in Q_t$. If $Ux \neq 0$, then $\lim_{\beta \rightarrow \infty} \hat{p}(y|\beta x) = \frac{1}{K}$.

Proof. We note that with a similar argument as in the derivation of (7.3) it holds

$$\hat{p}(y|\beta x) \leq \hat{p}(i|\beta x) + \frac{1}{K}(1 - \hat{p}(i|\beta x)) = \frac{K-1}{K}\hat{p}(i|\beta x) + \frac{1}{K} \quad (7.9)$$

We note that for all $\beta \geq \alpha$ it holds $\beta x \in Q_t$ so that

$$\hat{p}(i|\beta x) = \frac{1}{1 + e^{-g(\beta x)}} = \frac{1}{1 + e^{\langle W_g^{(L_g)}, U\beta x + d \rangle + b_g^{(L_g)}}}.$$

As $x_i^{(L-1)}(x) \geq 0$ for all $x \in \mathbb{R}^d$ it has to hold $(\beta Ux + d)_i \geq 0$ for all $\beta \geq \alpha$ and $i = 1, \dots, n_{L-1}$. This implies that $(Ux)_i \geq 0$ for all $i = 1, \dots, n_{L-1}$ and since $Ux \neq 0$ there has to exist at least one component i^* such that $(Ux)_{i^*} > 0$. Moreover, $W_g^{(L_g)}$ has strictly negative components and thus for all $\beta \geq \alpha$ it holds

$$g(\beta x) = \langle W_g^{(L_g)}, U\beta x + d \rangle + b_g^{(L_g)} = \beta \langle W_g^{(L_g)}, Ux \rangle + \langle W_g^{(L_g)}, d \rangle + b_g^{(L_g)}.$$

As $\langle W_g^{(L_g)}, Ux \rangle < 0$ we get $\lim_{\beta \rightarrow \infty} g(x) = -\infty$ and thus

$$\lim_{\beta \rightarrow \infty} \hat{p}(i|\beta x) = 0.$$

Plugging this into (7.9) yields the result. \square

In Section 7.4.3 we show that the condition $Ux \neq 0$ is not restrictive, as this property holds in all cases where we checked it. The negativity condition on the weights $W_g^{(L_g)}$ of the output layer of the in-vs. out-distribution discriminator g is crucial for the proof. This may seem restrictive, but we did not encounter any negative influence of this constraint on test accuracy, guaranteed or standard OOD detection performance. Thus, the asymptotic guarantees come essentially for free.

Table 7.2: **Architecture:** The architectures that are used for the binary discriminators. Each convolutional layer and the penultimate fully connected layers are directly followed by ReLUs.

CIFAR	R.ImgNet
Conv2d(3, 128)	Conv2d(3, 128)
Conv2d(128, 256) _{s=2}	AvgPool(2)
Conv2d(256, 256)	Conv2d(128, 256) _{s=2}
AvgPool(2)	AvgPool(2)
FC(16384, 128)	Conv2d(256, 256)
FC(128, 1)	AvgPool(2)
	FC(50176, 128)
	FC(128, 1)

7.4 Experiments

We provide experiments using CIFAR10, CIFAR100 and Restricted Imagenet (R.ImgNet) as in-distributions. Like in most previous chapters, we use OpenImages as training OOD for CIFAR. For R.ImgNet we again use the ILSVRC2012 train images that do not belong to R.ImgNet as training out-distribution (NotR.ImgNet).

7.4.1 Training

Binary Training We train the binary discriminator between in-and out-distribution using the loss in Eq. (7.4) with the bounds over an l_∞ -ball of radius $\varepsilon = 0.01$ for the out-distribution, thus using the same threat model as in the previous chapter. We use relatively shallow CNNs with only 5 layers plus pooling layers. The architecture is shown in Table 7.2.

Similarly to the previous chapter, we use long training schedules, running Adam for 1000 epochs, with an initial learning rate of $1e-4$ that we decrease by a factor of 5 on epochs 500, 750 and 850 and with a batch size of 128 from the in-distribution and 128 from the out-distribution (for R.ImgNet: 50 epochs with drops at 25, 35, 45, batch sizes 32). In order to avoid large losses we also use a simple ramp up schedule for the ε used in IBP and we downweight the out-distribution loss during the initial phase of training by a scalar κ . Both ε and κ are increased linearly from 0 to their final values (0.01 and 1, respectively) over the first 300 epochs (for R.ImgNet over the first 25 epochs). Compared to the training of GOOD which sometimes fails, we found that training of the binary discriminator is very stable and even 100 epochs on CIFAR would be sufficient, but we found that longer training lead to slightly better results. Weight decay is set to $5e-4$, but is disabled for the weights in the final layer. As data augmentation we use AutoAugment (Cubuk *et al.*, 2019) for CIFAR and simple 4 pixel crops and reflections on R.ImgNet. The strict negativity of the weights leads to a negative bias of g which can cause problems at an early stage if the $b_g^{(L_g)}$ is initialized at 0 and thus we choose 3 as initialization.

Semi-Joint Training For the classifier we use a ResNet18 architecture on CIFAR and a ResNet50 on R.ImgNet. Note that the architecture of our binary discriminator is over an order of magnitude smaller than the one CIFAR model in GOOD (11MB instead of 135MB) and thus the memory overhead for the binary discriminator is less than a third of that of the classifier. On CIFAR we train for 100 epochs using SGD with momentum of 0.9 and a learning rate of 0.1 that drops by a factor of 10 on epochs 50, 75 and 90 (on R.ImgNet 75 epochs with drops at 30 and 60). For all datasets we train using a batch size of 128 (plus 128 out-distribution samples in the case of OE). In order to fit batches of 128 in-distribution samples and 128 out-distribution samples on R.ImgNet we had to train using 4 V100 GPUs in parallel. Because of batch normalization in multi-GPU training it is important to not simply stack the batches but to interlace in- and out-distribution samples.

As discussed in Section 7.2.1, when training the binary discriminator one implicitly assumes that in- and (worst-case) out-distribution samples are equally likely. It seems very unlikely that one would be presented with such a large number of OOD samples in practice but as discussed in Section 7.2.1, we can adjust the weight of the losses after training the discriminator (but before training the classifier) by shifting the bias $b_g^{(L_g)}$ in the output layer of the binary discriminator. We train several ProoD models for binary shifts in $\{0, 1, 2, 3, 4, 5, 6\}$ and then evaluate the AUC and guaranteed AUC (see 7.4.2) on a subset of the training out-distribution OpenImages (resp. NotR.ImgNet). For all bias shifts we use the same fixed provably trained binary discriminator and only train the classifier part. As our goal is to have provable guarantees with minimal or no loss on the standard OOD detection task, among all solutions which have better AUC than OE we choose the one with the highest guaranteed AUC on OpenImages (on CIFAR10/CIFAR100) respective NotR.ImgNet (on R.ImgNet). If none of the solutions has better AUC than OE on the training out-distribution we take the one with the highest AUC. We show the trade-off curves for the example in Figure 7.2.

7.4.2 Evaluation

Adversarial AUC For the evaluation of the Adversarial AUC, we basically use the same setup as in Chapter 6 with only minor modifications. First of all, we also add SquareAttack (Andriushchenko *et al.*, 2020) with 5000 queries in order to be more sure that gradient masking is not degrading our attack’s performance. Fortunately, we find that SquareAttack never outperforms our gradient based method, which indicates that the optimization works well. Secondly, on R.ImgNet we do use APGD and rely only on our custom PGD, because APGD would run out of memory on R.ImgNet. This is also not a bottleneck as our custom PGD outperforms APGD on CIFAR in almost all cases anyway.

Baselines We compare to a normally trained baseline (Plain) and outlier exposure (OE), both trained using the same architecture and hyperparameters as the classifier in ProoD. For methods that pursue adversarial robustness on the out-distribution, we compare to ACET and to the method of Adversarial Training using informative Outlier Mining (ATOM) that was proposed

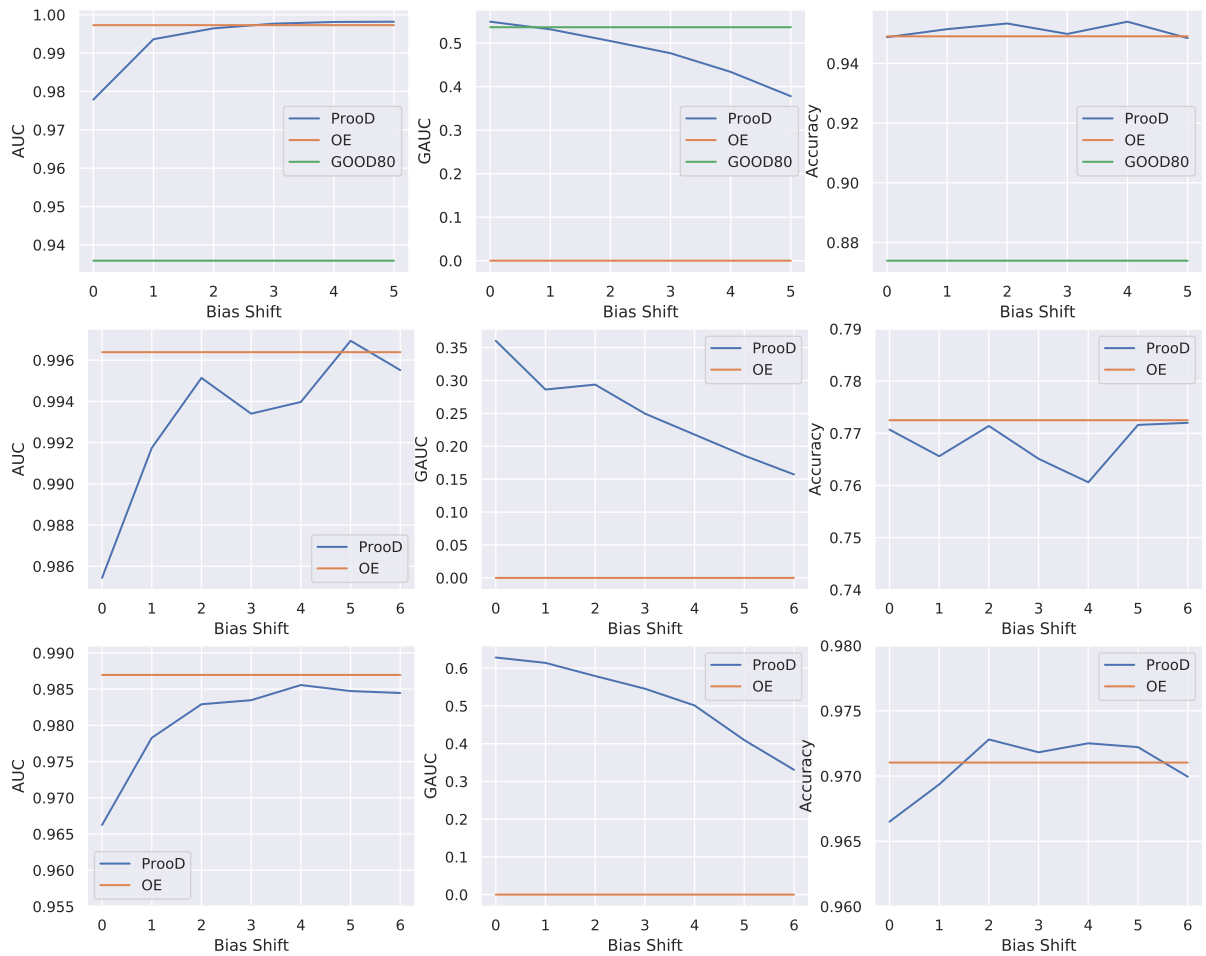


Figure 7.2: **Bias selection for CIFAR100 and R.ImgNet:** Using CIFAR10 (top), CIFAR100 (middle) and R.ImgNet (bottom) as the in-distribution and the test set of OpenImages as OOD (or NotR.ImgNet respectively) AUC, GAUC and test accuracy as a function of the bias shift Δ (see Eq. (7.6)).

in Chen *et al.* (2021) and that uses an out-class as opposed to the confidence score. Note that the authors originally claimed that ATOM produced near-perfect adversarially robust OOD detection against large threat models at no loss in accuracy. However, for both ATOM and ACET we found the pre-trained models by Chen *et al.* (2021) to have far worse adversarial OOD detection performance than they claimed so we retrained their models using our architecture, threat model and training out-distribution with their original code (for CIFAR10/100). Running these adversarial training procedures on ImageNet resolution is infeasibly expensive. For GOOD we also retrain using OpenImages as training OOD dataset. Since they are only available on CIFAR10, we tried to train models on CIFAR100 using and same hyperparameters and schedules as we used for CIFAR10. This only lead to models with accuracy below 25%, so we do not include these models in our evaluation. Since, in Chapter 6, we already

Table 7.3: **OOD performance:** For all models we report accuracy on the test set of the in-distribution and AUCs, guaranteed AUCs (GAUC), adversarial AUCs (AAUC) for different test out-distributions. The radius of the l_∞ -ball for the adversarial manipulations of the OOD data is $\varepsilon = 0.01$ for all datasets. The bias shift Δ that was used for ProoD is shown for each in-distribution. The AAUCs and GAUCs for ProoD tend to be very close, indicating remarkably tight certification bounds. Models with accuracy drop of $> 3\%$ relative to the model with highest accuracy are grayed out. Of the remaining models, we highlight the best OOD detection performance.

CIFAR10	Acc	CIFAR100			SVHN			LSUN_CR			Smooth		
		AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	95.01	90.0	0.0	0.7	93.8	0.0	0.3	93.1	0.0	0.5	98.0	0.0	0.7
OE	94.91	91.1	0.0	0.9	97.3	0.0	0.0	100.0	0.0	2.7	99.9	0.0	1.5
ATOM	93.63	78.3	0.0	21.7	94.4	0.0	24.1	79.8	0.0	20.1	99.5	0.0	73.2
ACET	93.43	86.0	0.0	4.0	99.3	0.0	4.6	89.2	0.0	3.7	99.9	0.0	40.2
GOOD ₈₀ *	87.39	76.7	47.1	57.1	90.8	43.4	76.8	97.4	70.6	93.6	96.2	72.9	89.9
GOOD ₁₀₀ *	86.96	67.8	48.1	49.7	62.6	34.9	36.3	84.9	74.6	75.6	87.0	76.1	78.1
ProoD-Disc	-	62.9	57.1	57.8	72.6	65.6	66.4	78.1	71.5	72.3	59.2	49.7	50.4
ProoD $\Delta=3$	94.99	89.8	46.1	46.8	98.3	53.3	54.1	100.0	58.3	59.7	99.9	38.2	38.8

CIFAR100	Acc	CIFAR10			SVHN			LSUN_CR			Smooth		
		AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	77.38	77.7	0.0	0.4	81.9	0.0	0.2	76.4	0.0	0.3	86.6	0.0	0.4
OE	77.25	77.4	0.0	0.2	92.3	0.0	0.0	100.0	0.0	0.7	99.5	0.0	0.5
ATOM	68.32	78.3	0.0	50.3	91.1	0.0	67.0	95.9	0.0	75.6	98.2	0.0	80.7
ACET	73.02	73.0	0.0	1.4	97.8	0.0	0.7	75.8	0.0	2.6	99.9	0.0	12.8
ProoD-Disc	-	56.1	52.1	52.3	61.0	58.2	58.4	70.4	66.9	67.1	29.6	26.4	26.5
ProoD $\Delta=5$	77.16	76.6	17.3	17.4	91.5	19.7	19.8	100.0	22.5	23.1	98.9	9.0	9.0

R.ImgNet	Acc	Flowers			FGVC			Cars			Smooth		
		AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	96.34	92.3	0.0	0.5	92.6	0.0	0.0	92.7	0.0	0.1	98.9	0.0	8.6
OE	97.10	96.9	0.0	0.2	99.7	0.0	0.4	99.9	0.0	1.8	98.0	0.0	1.9
ProoD-Disc	-	81.5	76.8	77.3	92.8	89.3	89.6	90.7	86.9	87.3	81.0	74.0	74.8
ProoD $\Delta=4$	97.25	96.9	57.5	58.0	99.8	67.4	67.9	99.9	65.7	66.2	98.6	52.7	53.5

*Uses different architecture of classifier, see ‘‘Baselines’’ in Section 7.4.2.

showed that CCU does not provide benefits over OE on OOD data that is not very far from the in-distribution (e.g. uniform noise) we do not include it as baseline. Similarly, we do not include RATIO, partially because it uses an l_2 threat model, but mostly because we are not

Table 7.4: **Generalization to Larger ε** : We evaluate all CIFAR models in Table 7.3 using an $\varepsilon = \frac{8}{255}$, and thus an unseen threat model. The provable methods GOOD and ProoD generalize surprisingly well, while neither ATOM nor ACET display any generalization to the larger threat model.

CIFAR10	Acc	CIFAR100			SVHN			LSUN_CR			Smooth		
		AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
ProoD-Disc	-	62.9	44.1	46.1	72.6	52.5	57.1	78.1	56.3	58.9	59.2	34.9	37.2
ProoD $\Delta=3$	94.99	89.8	39.2	41.0	98.3	46.9	50.8	100.0	50.2	52.7	99.9	30.4	30.6
CIFAR100	Acc	CIFAR10			SVHN			LSUN_CR			Smooth		
		AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
ProoD-Disc	-	56.1	41.1	43.1	61.0	50.5	51.8	70.4	57.5	58.8	29.6	20.9	20.8
ProoD $\Delta=5$	76.51	76.6	13.7	14.1	91.5	16.9	16.9	100.0	18.1	18.2	98.9	8.1	8.1
R.ImgNet	Acc	Flowers			FGVC			Cars			Smooth		
		AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
ProoD-Disc	-	81.5	60.4	61.4	92.8	78.0	80.8	90.7	76.3	79.2	81.0	47.3	53.7
ProoD $\Delta=4$	97.25	96.9	42.8	45.0	99.8	57.0	59.4	99.9	56.0	58.7	98.6	31.6	36.3

interested in in-distribution robustness and instead wish to study robust OOD detection at *no loss* in accuracy. We also evaluate the OOD-performance of the provable binary discriminator (ProoD-Disc) that we trained for ProoD. Note that this is not a classifier and is included only for reference. All results are in Table 7.3.

Results ProoD achieves non-trivial GAUCs on all datasets. As was also observed in (Bitterwolf *et al.*, 2020), this shows that the IBP guarantees not only generalize to unseen samples but even to unseen distributions. In Table 7.4 we show that they even generalize to the much larger threat model $\varepsilon = 8/255$. In general, the gap between our GAUCs and AAUCs is extremely small. This shows that the seemingly simple IBP bounds can be remarkably tight, as has been observed in other works (Gowal *et al.*, 2018; Jovanović *et al.*, 2022). It also shows that there would be very little benefit in applying stronger verification techniques like (Cheng *et al.*, 2017; Katz *et al.*, 2017; Dathathri *et al.*, 2020) in ProoD. Similarly, it demonstrates the strengths of our attack as there provably does not exist an attack that could lower the AAUCs on our ProoD model by more than 1.4% on any of the out-distributions. The bounds are also much tighter than for GOOD, which is likely due to the fact that for GOOD the confidence is much harder to optimize during an attack because it involves maximizing the confidence in an essentially random class.

For CIFAR10, on 3 out of 4 out-distributions ProoD’s GAUCs are higher than ATOM’s and ACET’s AAUCs, i.e. our model’s *provable* adversarial robustness exceeds the SOTA methods’ *empirical* adversarial robustness in these cases. Note that this is *not* due to our retraining, be-

cause the authors’ pre-trained models perform even more poorly (as shown in Table 7.5). On CIFAR100, ProoD’s guarantees are weaker and ATOM produces strong AAUCs. However, we observe that training both ACET and ATOM can produce inconsistent results, i.e. sometimes almost no robustness is achieved. For the successfully trained robust ATOM model on CIFAR100 we observe drastically reduced accuracy. Due to the difficulty in attacking these models, it is not impossible that a more sophisticated attack could produce even lower AAUCs. Combined with the fact that both ACET and ATOM rely on expensive adversarial training procedures we argue that using ProoD is preferable in practice.

On CIFAR10, we see that ProoD’s GAUCs are comparable to, if slightly worse than the ones of both GOOD₈₀ and GOOD₁₀₀. Note that although the presented GOOD models are retrained, the same observations hold true when comparing to the pre-trained models (see Table 7.5). However, we want to point out that ProoD achieves this while retaining both high accuracy and OOD performance, both of which are lacking for GOOD. It is also noteworthy that the GOOD models’ memory footprints are over twice as large as ProoD’s. Generally, for ProoD the accuracy is comparable to OE and the OOD performance is similar or marginally worse. Thus ProoD shows that it is possible to achieve certifiable adversarial robustness on the out-distribution while keeping very good prediction and OOD detection performance. Note that all methods struggle on separating CIFAR10 and CIFAR100 when using OpenImages as training OOD.

To the best of our knowledge with R.ImgNet we provide the first worst case OOD guarantees on high-resolution images. The GAUCs are higher than on CIFAR, indicating that meaningful certificates on higher resolution are more achievable on this task than one might expect. FGVC and Cars may seem simple to separate from the animals in R.ImgNet but this cannot be said for Flowers which are difficult to provably distinguish from images of insects on flowers.

In summary, ProoD achieves our goal of maintaining high accuracy and clean OOD detection performance while providing provably adversarially robust OOD detection. In fact, out of all the methods that do not significantly impair the in-distribution accuracy, ProoD is the only method providing such guarantees as well while simultaneously having the highest empirical robustness. Also note that for applications where adversarial robustness on the in-distribution is desired despite the induced reduction in accuracy, one can combine our ProoD model with a robustly trained classifier.

7.4.3 Adversarial Asymptotic Overconfidence

We also empirically evaluate the asymptotic behavior of ProoD as compared to models that do not benefit from an asymptotic guarantee like Theorem 4. Concretely, we take different models that were trained on CIFAR10 and evaluate their confidence on different CIFAR100 samples. For each sample x we track the confidence, $\max_k \hat{p}(k|x)$, along a trajectory in a uniform noise direction $x + \alpha n$, where $n \in [-0.5, 0.5]^d$ and $\alpha \geq 0$. The mean confidence across 100 such trajectories is shown on the left side of Figure 7.3. Even the models that produce low confidences on the original OOD sample asymptotically converge to maximal

Table 7.5: **Training with 80M Tiny Images:** We repeat the evaluation from Table 7.3 for models that were trained using 80M Tiny Images as out-distribution instead of OpenImages. Plain is identical to before and is just repeated for the reader’s convenience. For ATOM and ACET we compare to pre-trained models from (Chen *et al.*, 2021). Note that these models show almost no robustness on CIFAR100 - despite the far stronger claims in (Chen *et al.*, 2021). Models with accuracy drop of $> 3\%$ relative to the model with highest accuracy are grayed out. Of the remaining models, we highlight the best OOD detection performance. Note that the conclusions from Table 7.3 still hold, which indicates that our method is robust to changes in the choice of training out-distribution.

CIFAR10	Acc	CIFAR100			SVHN			LSUN_CR			Smooth		
		AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
ATOM	95.20	93.7	0.0	14.4	99.6	0.0	8.6	99.7	0.0	40.0	99.6	0.0	18.8
ACET	91.48	91.2	0.0	80.5	95.3	0.0	87.6	98.9	0.0	95.0	99.9	0.0	98.3
ProoD-Disc	-	67.4	61.0	61.7	73.2	65.5	66.4	78.0	72.2	72.7	82.3	71.5	72.9
ProoD $\Delta=3$	95.47	96.0	41.9	43.9	99.5	48.8	49.4	99.6	47.6	53.1	99.7	55.8	57.0

CIFAR100	Acc	CIFAR10			SVHN			LSUN_CR			Smooth		
		AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
ATOM	75.06	64.3	0.0	0.2	93.6	0.0	0.2	97.5	0.0	9.3	98.5	0.0	15.0
ACET	74.43	79.8	0.0	0.2	90.2	0.0	0.0	96.0	0.0	2.1	92.9	0.0	0.3
ProoD-Disc	-	53.8	50.3	50.4	73.1	69.8	69.9	68.1	63.8	64.0	67.2	63.8	63.9
ProoD $\Delta=1$	76.79	80.5	23.1	23.2	93.7	33.9	34.0	97.2	29.6	30.4	98.9	29.7	31.3

confidence far away. The only exceptions here are GOOD and ProoD and only ProoD can guarantee that the confidence cannot converge to 1.

However, even though the architecture provably prevents arbitrarily overconfident predictions and Theorem 4 ensures that most directions will indeed converge to uniform, it is, in principle, possible to find directions where the confidence $\hat{p}(i|x)$ remains constant if the condition $Ux \neq 0$ in Theorem 4 is not satisfied. We attempted to find such directions by running the following type of attack. We start from a random point $x \in [-0.5, 0.5]^d$ that we project onto a sphere of radius 100. We now run gradient descent (for 20000 steps), maximizing $g(x)$ while projecting onto the sphere at each step (unnormalized gradients with step size 0.1 for the first 10000 steps and 0.01 for the last 10000 steps). We then increase the radius to 1000 and run an additional 20000 steps with step size 0.1. We rescale the resulting direction vector down to an l_∞ -ball of norm 1 and compute the confidence $\hat{p}(i|x)$ as a function of the scaling in the adversarial directions. We show the resulting scale-wise *maximum* over 100 adversarial directions in Figure 7.3. Note that even the worst-case over 100 adversarially found directions decays to 0 asymptotically, thus empirically confirming the practical utility of Theorem 4. Note that the value of $\hat{p}(i|x)$ converging to 0 implies that the confidence of the ProoD model $\hat{p}(y|x)$ converges to 10%.

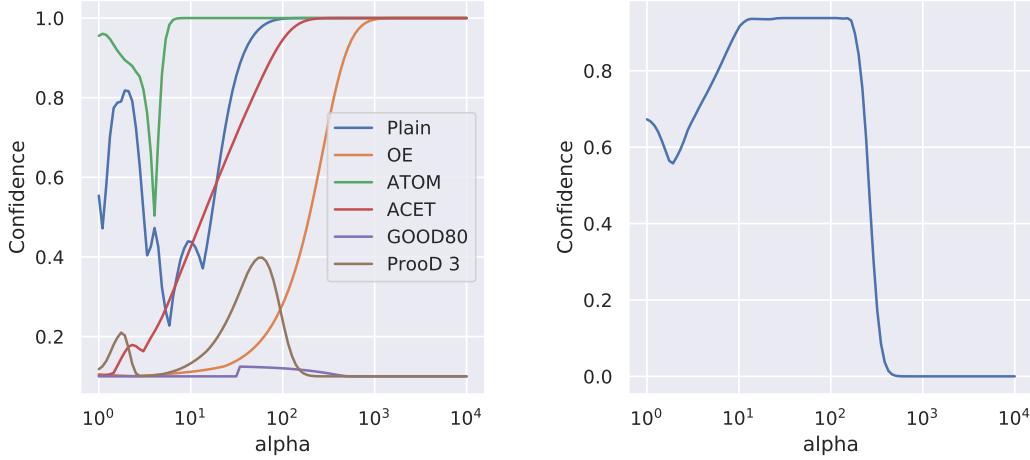


Figure 7.3: **Left, Asymptotic confidence:** We plot the mean confidence in the predicted in-distribution class for different models as one moves away from CIFAR100 samples along the trajectories $x + \alpha n$, where $n \in [-0.5, 0.5]^d$ and $\alpha \geq 0$. Only GOOD and ProoD converge to uniform confidence. **Right, Adversarial asymptotic confidence:** We try to find adversarial directions in which ProoD remains at a constant high confidence, as opposed to converging to low confidence. We plot the *maximum* of $\hat{p}(i|x)$ across 100 adversarially chosen directions as one moves further in these directions by factors of α . Note that $\hat{p}(i|x) \rightarrow 0$ implies $\hat{p}(y|x) \rightarrow \frac{1}{K}$.

In Figure 7.3 GOOD also stands out as having low confidence in all directions that we studied. This is because in all the asymptotic regions that we looked at, the pre-activations of the penultimate layer are all negative. If one moves outward and these pre-activations only get more negative in all directions far away from the data, the confidence does, in fact, remain low. Unfortunately, it also leads to gradients that are precisely zero, which is why the same attack can not be applied here. However, there is no guarantee that GOOD does not also get in some direction asymptotically overconfident.

7.5 Conclusion

We have demonstrated how to combine a provably adversarially robust binary discriminator between in- and out-distribution with a standard classifier in order to simultaneously achieve high accuracy, high clean OOD detection performance as well as certified adversarially robust OOD detection. Thus, we have combined the best properties of the methods CCU and GOOD from the previous chapters with only a small increase in total model size and only a single hyperparameter. This suggests that certifiable adversarial robustness on the out-distribution (as opposed to the in-distribution) is indeed possible without losing accuracy. We further showed how in our model simply enforcing negativity in the final weights of the discriminator fixes the problem of asymptotic overconfidence in ReLU classifiers. Training ProoD models is simple and stable and thus ProoD provides OOD guarantees that come (almost) for free.

Chapter 8

Conclusion

8.1 Summary

In this thesis we have discussed different aspects of the adversarially robust detection of out-of-distribution examples in deep learning based vision classifiers.

In Chapter 1 we gave a general overview of OOD detection. We stated the definition of OOD detection task as the rejection of samples that do not belong to any of the classes in the classification task in question and we summarized the performance main metrics that are used. We briefly described the issues of overconfidence in neural networks. Concretely, we first summarized how over- or underconfidence in a neural network can be measured via the Expected Calibration Error and secondly how prior work has shown that ReLU networks lead to overconfident predictions far from the training data. Finally we gave a brief introduction into the issues related to adversarial robustness, especially the difficulty of reliably evaluating defenses and how certifiable robustness can solve this issue. We specifically focused on summarizing Interval Bound Propagation for the reader.

Part I of this dissertation studied the “clean” OOD detection, i.e. the detection of unperturbed OOD samples. In Chapter 2 we showed our empirical results from the paper Meinke and Hein (2020), where we benchmarked a variety of OOD detection methods that all claimed state-of-the-art performance at the time. Concretely, we compared 9 different methods using 5 different in-distributions and 6 test out-distributions for each and showed that only Outlier Exposure consistently outperformed the other methods. Our main takeaway was that the assumption of using a large and diverse, but unlabeled out-distribution at train time was indeed helpful for the task of OOD detection.

In Chapter 3 presented our results from Bitterwolf *et al.* (2022) where we studied the question of how such an out-distribution should best be incorporated into the model in order to lead to the best performance. We broke down the different scoring functions implicitly or explicitly used by various methods and theoretically showed that some of these were actually equivalent in the Bayes’ limit of infinite data. We then empirically verified that theoretically equivalent methods indeed behaved similarly in practice, and found that, empirically, no method consistently outperformed Outlier Exposure.

Part II of this dissertation dealt with the detection of adversarially perturbed OOD samples. Concretely, in Chapter 4, we discussed Augustin *et al.* (2020) which dealt with the problems

that arise when using adversarial training for achieving adversarially robust low confidence on perturbed OOD samples (i.e. ACET). We showed that ACET suffered from training instabilities that could be resolved by combining it with adversarial training on the in-distribution. We then went on to show that this combination of losses leads to other desirable properties, i.e. higher adversarial robustness at a given loss in clean accuracy and the ability to generate visual counterfactuals using Projected Gradient Descent in image space. These visual counterfactuals were then shown to be qualitatively more realistic than the corresponding images for adversarial training, indicating the importance of adversarially robust low confidences on OOD samples.

In Part III we took on the issue of obtaining certificates for the adversarial robustness of a model’s confidence on OOD data. First, in Chapter 5, we presented more results from our paper Meinke and Hein (2020). We developed the method of Certified Certain Uncertainty which used Gaussian mixture models to ensure that our classifier’s confidence would provably converge to uniform across all classes far from the training data. We theoretically showed that this construction also implied certificates on the adversarial robustness of our confidence estimates in a very specific threat model around uniform noise images. We showed that, despite uniform noise being seemingly trivial to detect, all of our 9 baseline models would fail to detect points that our CCU model’s certificates could provably exclude. Furthermore, we compared the clean accuracy and clean OOD detection performance of CCU to our baselines and found that CCU did not lead to any degradation in performance relative to OE.

In Chapter 6 we presented our work in (Bitterwolf *et al.*, 2020) which showed that Interval Bound Propagation could be used to train classifiers that have certifiably low confidence around unseen out-distributions under the l_∞ -threat model. We particularly showed how the use of IBP leads to very unstable training dynamics and crucially how to technically overcome these obstacles. While this also restricted our choice of architecture to relatively shallow networks, we showed that when comparing to other training methods with the same architecture our method GOOD did not lead to any drop in accuracy.

Finally, in Chapter 7 we described how our work in (Meinke *et al.*, 2022) combined ideas from both CCU and GOOD into ProoD in order to obtain the desirable properties of both methods without their respective drawbacks. In particular, we showed that our method simultaneously achieved simple and stable training using arbitrary architectures for the classifier, with strong guarantees on the adversarial robustness on the confidence on OOD data while losing no accuracy whatsoever and only minimally affecting the clean OOD detection performance. On top of this, we showed that our construction also solved the issue of asymptotic overconfidence. Empirically we could also demonstrate that our method could effortlessly scale to tasks with many classes (CIFAR100) or with higher resolutions (R.ImgNet).

8.2 Outlook

The topic of OOD detection has recently received a lot of attention from the research community, with countless contradicting claims about state-of-the-art performance (Ren *et al.*, 2019;

Hsu *et al.*, 2020; Yu and Aizawa, 2019; Berglind *et al.*, 2022; Sun *et al.*, 2021; Ming *et al.*, 2022; Lin *et al.*, 2021; Macêdo *et al.*, 2021; Gomes *et al.*, 2022; Papadopoulos *et al.*, 2021; Liu *et al.*, 2020; Chen *et al.*, 2021). Some OOD studies turn out to not be reproducible as was shown, e.g. by Tajwar *et al.* (2021); Meinke and Hein (2020) and thus it is unclear how much progress the community has made on clean OOD detection at a fixed amount of training data in the past few years. A stronger set of diverse benchmarks may help to alleviate this issue and, in fact, a unified benchmark has recently been proposed (Yang *et al.*, 2021, 2022). Interestingly, this newly proposed benchmark suggests that, contrary to the findings of our work here, additional data may not always be necessary for high OOD detection performance. In my view, more work on reproducing old methods on new benchmarks is needed for more conclusive evidence on this.

That being said, promising results in clean OOD detection have emerged from large pre-trained transformer models (Fort *et al.*, 2021; Koner *et al.*, 2021) that implicitly leverage vastly more data than most other methods. If reliable benchmarks indeed show that large pre-training datasets effectively solve the OOD detection problem, then the community should move on to different problems. Safety-critical OOD detection is still relevant in somewhat understudied data modalities where it is unclear if pre-training and transfer learning can be employed as successfully as in computer vision, e.g. predictive maintenance tasks (Biggio *et al.*, 2021). In these cases, it is possible that synthetic data will play a much larger role.

However, even in the case of Natural Language Processing, where pre-training is hugely successful, there are many unexplored connections to OOD detection as developed in the vision community. For example, while OOD detection in NLP has been studied a lot, the idea of adversarially robust OOD detection needs to be investigated more in NLP models. This is especially relevant at a time where NLP models, that were trained on vast amounts of general text data, are fine-tuned and publicly deployed for a specific subtask. It should be possible to restrict such models to only respond to user requests in their specific use case, e.g. a customer service bot should not be giving medical advice. Detecting such OOD queries and responses requires the consideration of adversarially robust OOD detection as users have been known to successfully use prompt injection to “jail-break” large language models through cleverly crafted queries that attempt to evade detection (Perez and Ribeiro, 2022).

Nonetheless, currently adversarially robust detection of OOD data suffers from the same issue as adversarial robustness in general, which is the difficulty of defining and optimizing threat models that more closely align with human notions of a distance metric. Several alternatives to l_p -models have been proposed and studied (Engstrom *et al.*, 2019b; Wong and Kolter, 2021; Brown *et al.*, 2017; Zajac *et al.*, 2019; Stutz *et al.*, 2021), but widely adopted, realistic threat models do not yet exist.

Unfortunately, even for the simple threat models of l_p -balls, adversarially robust OOD detection is currently lacking unified tools for the assessment of robustness. In this thesis, we saw that gradient masking on OOD data can occur much more easily than on in-distribution data and that AutoAttack and APGD in particular are not optimally suited for the evaluation of AAUCs. Thus, the development of an analogue for AutoAttack and RobustBench (Croce *et al.*, 2021) would greatly benefit the community.

Another line of work that could be transferred from adversarial robustness on the in-distribution is related to the stability of training. For adversarial training it is known that both adversarial overfitting (Rice *et al.*, 2020) as well as catastrophic overfitting (Wong *et al.*, 2020) can lead to non-robust models. Remedies for both of these issues have been developed in the context of adversarial training, which make adversarial training both fast and stable. As we discussed in this thesis, successfully running adversarial training on OOD data is much more difficult than on the in-distribution data and generally requires even more steps so this would be very valuable for methods like ACET or RATIO.

It is noteworthy that a lot of progress has been made on certifiable robustness. Especially interesting is the observation that derandomized smoothing can produce state-of-the-art performance when off-the-shelf classifiers and diffusion-based denoisers are combined (Carlini *et al.*, 2022). This suggests that Sutton’s bitter lesson might even apply to certified robustness, i.e. that rather than focusing on more cleverly designed algorithms and architectures, the community should focus on improving their access and use of data (Sutton, 2019; Geirhos *et al.*, 2022). At any rate, it would be very interesting to see if more scalable approaches can be leveraged for certifiably adversarially robust detection of OOD as well.

For all methods that we developed in this dissertation, we made the assumption that some training out-distribution was available. Many alternative approaches without this assumption exist for the case of clean OOD detection. However, in the adversarial setting, it is unclear how for RATIO, GOOD or ProoD could be adapted to not need OOD data at all. Of course one could use synthetic data instead but a lot of work is needed to find out how to best do this. Also self-supervised pre-training methods that do not explicitly differentiate between what would later be in- and out-distribution during fine tuning have been shown to allow for pre-training that leads to adversarially robust methods (Gowal *et al.*, 2021). It would be very interesting to see if these methods could also be used to provide certifiable robustness or even certifiable OOD detection.

An entirely different avenue of research could be to rethink the type of robustness one can try to certify. The robustness guarantees discussed throughout this dissertation were computed around points that had to be sampled from natural distributions. Ideally, it would be possible to not only issue such point-wise robustness guarantees but also certify good OOD detection performance against entire families of distributions. Unfortunately, even specifying such families beyond trivial noise distributions likely requires deep learning models in and of itself, thus making it difficult to fully evaluate the quality of the threat model. There has been some work in this direction (Berrada *et al.*, 2021a; Yoon *et al.*, 2022) but a lot more work is needed to get to potentially useful guarantees.

Abbreviations

AAUC	Adversarial AUC
APGD	Auto Projected Gradient Descent
AUC	Area Under receiver-operator Curve
CCU	Certified Certain Uncertainty
CEDA	Confidence Enhancing Data Augmentation
CNN	Convolutional Neural Network
DE	Deep Ensembles
ECE	Expected Calibration Error
EDL	Evidential Deep Learning
FGSM	Fast Gradient Sign Method
FPR	False Positive Rate
GAUC	Guaranteed AUC
GOOD	Guaranteed Out-Of-distribution Detection
IBP	Interval Bound Propagation
MCD	Monte-Carlo Dropout
NLP	Natural Language Processing
ODIN	Out-of-Distribution Detector for Neural Networks
OE	Outlier Exposure
OOD	Out-Of-Distribution
PGD	Projected Gradient Descent
ProoD	Provable out-of-distribution Detection
RATIO	Robustness via Adversarial Training on In- and Out-distribution
SOTA	State-Of-The-Art
SR	Success Rate
TE	Test Error
TPR	True Positive Rate

Bibliography

- Alayrac, J.-B., Uesato, J., Huang, P.-S., Fawzi, A., Stanforth, R., and Kohli, P. (2019). Are labels required for improving adversarial robustness? In *NeurIPS*.
- Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. (2020). Square attack: a query-efficient black-box adversarial attack via random search. In *ECCV*.
- Anil, C., Lucas, J., and Grosse, R. (2019). Sorting out lipschitz function approximation. In *ICML*. PMLR.
- Arora, R., Basuy, A., Mianjyz, P., and Mukherjee, A. (2018). Understanding deep neural networks with rectified linear unit. In *ICLR*.
- Athalye, A., Carlini, N., and Wagner, D. (2018). Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*.
- Augustin, M., Meinke, A., and Hein, M. (2020). Adversarial robustness on in-and out-distribution improves explainability. In *ECCV*.
- Augustin, M., Boreiko, V., Croce, F., and Hein, M. (2022). Diffusion visual counterfactual explanations. In *NeurIPS*.
- Azizmalayeri, M., Moakar, A. S., Zarei, A., Zohrabi, R., Manzuri, M. T., and Rohban, M. H. (2022). Your out-of-distribution detection method is not robust! In *NeurIPS*.
- Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS One*, **10**(7), e0130140.
- Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. (2010). How to explain individual classification decisions. *Journal of Machine Learning Research (JMLR)*, **11**, 1803–1831.
- Barocas, S., Selbst, A. D., and Raghavan, M. (2020). The hidden assumptions behind counterfactual explanations and principal reasons. In *FAT*.
- Bauschke, H. H. and Borwein, J. M. (1996). On projection algorithms for solving convex feasibility problems. *SIAM Review*, **38**, 367–426.

- Berglind, F., Temam, H., Mukhopadhyay, S., Das, K., Sajol, M. S. I., Kumar, S., and Kallurupalli, K. (2022). Xood: Extreme value based out-of-distribution detection for image classification. *arXiv:2208.00629*.
- Berrada, L., Dathathri, S., Dvijotham, K., Stanforth, R., Bunel, R., Uesato, J., Goyal, S., Kumar, M. P., *et al.* (2021a). Make sure you're unsure: A framework for verifying probabilistic specifications. In *NeurIPS*.
- Berrada, L., Dathathri, S., Stanforth, R., Bunel, R., Uesato, J., Goyal, S., Kumar, M. P., *et al.* (2021b). Verifying probabilistic specifications with functional lagrangians. *arXiv:2102.09479v1*.
- Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. (2013). Evasion attacks against machine learning at test time. In *ECML/PKDD*.
- Biggio, L., Wieland, A., Chao, M. A., Kastanis, I., and Fink, O. (2021). Uncertainty-aware prognosis via deep gaussian process. *IEEE Access*.
- Birhane, A. and Prabhu, V. U. (2021). Large image datasets: A pyrrhic win for computer vision? In *WACV*, pages 1537–1547.
- Bitterwolf, J., Meinke, A., and Hein, M. (2020). Certifiably adversarially robust detection of out-of-distribution data. In *NeurIPS*.
- Bitterwolf, J., Meinke, A., Augustin, M., and Hein, M. (2022). Breaking down out-of-distribution detection: Many methods based on ood training data estimate a combination of the same core quantities. In *ICML*.
- Boreiko, V., Augustin, M., Croce, F., Berens, P., and Hein, M. (2022a). Sparse visual counterfactual explanations in image space. In *DAGM German Conference on Pattern Recognition*. Springer.
- Boreiko, V., Ilanchezian, I., Ayhan, M. S., Müller, S., Koch, L. M., Faber, H., Berens, P., and Hein, M. (2022b). Visual explanations for the detection of diabetic retinopathy from retinal fundus images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer.
- Boyd, S., Boyd, S. P., and Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.
- Brown, T. B., Mané, D., Roy, A., Abadi, M., and Gilmer, J. (2017). Adversarial patch. *arXiv:1712.09665*.
- Bunel, R., Mudigonda, P., Turkaslan, I., Torr, P., Lu, J., and Kohli, P. (2020). Branch and bound for piecewise linear neural network verification. *JMLR*, **21**(2020).

- Carlini, N. and Wagner, D. (2017a). Adversarial examples are not easily detected: Bypassing ten detection methods. In *ACM Workshop on Artificial Intelligence and Security*.
- Carlini, N. and Wagner, D. (2017b). Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*.
- Carlini, N., Tramer, F., Kolter, J. Z., *et al.* (2022). (certified!!) adversarial robustness for free! *arXiv:2206.10550*.
- Carmon, Y., Ragunathan, A., Schmidt, L., Duchi, J. C., and Liang, P. S. (2019). Unlabeled data improves adversarial robustness. In *NeurIPS*.
- Chang, C.-H., Creager, E., Goldenberg, A., and Duvenaud, D. (2019). Explaining image classifiers by counterfactual generation. In *ICLR*.
- Chao, M. A., Kulkarni, C., Goebel, K., and Fink, O. (2022). Fusing physics-based and deep learning models for prognostics. *Reliability Engineering & System Safety*, **217**, 107961.
- Chen, J., Li, Y., Wu, X., Liang, Y., and Jha, S. (2021). Informative outlier matters: Robustifying out-of-distribution detection using outlier mining. In *ECML*.
- Chen, J., Li, Y., Wu, X., Liang, Y., and Jha, S. (2022). Robust out-of-distribution detection for neural networks. In *The AAAI-22 Workshop on Adversarial Machine Learning and Beyond*.
- Cheng, C.-H., Nührenberg, G., and Ruess, H. (2017). Maximum resilience of artificial neural networks. In *International Symposium on Automated Technology for Verification and Analysis*.
- Choi, H., Jang, E., and Alemi, A. A. (2018). Waic, but why? generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*.
- Cohen, G., Afshar, S., Tapson, J., and Van Schaik, A. (2017). Emnist: Extending mnist to handwritten letters. In *IJCNN*.
- Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. (2019). Certified adversarial robustness via randomized smoothing. In *NeurIPS*.
- Croce, F. and Hein, M. (2020a). Minimally distorted adversarial examples with a fast adaptive boundary attack. In *ICML*.
- Croce, F. and Hein, M. (2020b). Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*.
- Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., and Hein, M. (2021). Robustbench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

- Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In *CVPR*.
- Dai, D. and Van Gool, L. (2018). Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3819–3824. IEEE.
- Dathathri, S., Dvijotham, K., Kurakin, A., Raghunathan, A., Uesato, J., Bunel, R., Shankar, S., Steinhardt, J., Goodfellow, I., Liang, P., *et al.* (2020). Enabling certification of verification-agnostic networks via memory-efficient semidefinite programming. In *NeurIPS*.
- De Palma, A., Bunel, R., Desmaison, A., Dvijotham, K., Kohli, P., Torr, P. H., and Kumar, M. P. (2021). Improved branch and bound for neural network verification via lagrangian decomposition. *arXiv:2104.06718*.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *CVPR*.
- DeVries, T. and Taylor, G. W. (2017). Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.
- Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2016). Density estimation using real nvp. *arXiv:1605.08803*.
- Dong, Y., Su, H., Zhu, J., and Bao, F. (2017). Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv preprint, arXiv:1708.05493*.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., *et al.* (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.
- Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Tran, B., and Madry, A. (2019a). Adversarial robustness as a prior for learned representations. *arXiv:1906.00945*.
- Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., and Madry, A. (2019b). Exploring the landscape of spatial robustness. In *International conference on machine learning*.
- Engstrom, L., Ilyas, A., Santurkar, S., and Tsipras, D. (2019c). Robustness (python library).
- EU (2021). Proposal for a regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts. *COM(2021) 206 final*.
- Fang, Z., Li, Y., Lu, J., Dong, J., Han, B., and Liu, F. (2022). Is out-of-distribution detection learnable? In *NeurIPS*.

- Fort, S., Ren, J., and Lakshminarayanan, B. (2021). Exploring the limits of out-of-distribution detection. *NeurIPS*.
- Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*.
- Geirhos, R., Temme, C. R., Rauber, J., Schütt, H. H., Bethge, M., and Wichmann, F. A. (2018). Generalisation in humans and deep neural networks. In *NeurIPS*.
- Geirhos, R., Narayanappa, K., Mitzkus, B., Thieringer, T., Bethge, M., Wichmann, F. A., and Brendel, W. (2022). The bittersweet lesson: data-rich models narrow the behavioural gap to human vision. *Journal of Vision*.
- Ghosh, S., Shet, R., Amon, P., Hutter, A., and Kaup, A. (2018). Robustness of deep convolutional neural networks for image degradations. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2916–2920. IEEE.
- Gomes, E. D. C., Alberge, F., Duhamel, P., and Piantanida, P. (2022). Igeood: An information geometry approach to out-of-distribution detection. In *ICLR*.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *NeurIPS*.
- Goodfellow, I. J., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *ICLR*.
- Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T., and Kohli, P. (2018). On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv:1810.12715*.
- Gowal, S., Qin, C., Uesato, J., Mann, T., and Kohli, P. (2020). Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*.
- Gowal, S., Huang, P.-S., van den Oord, A., Mann, T., and Kohli, P. (2021). Self-supervised adversarial robustness for the low-label, high-data regime. In *ICLR*.
- Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., and Lee, S. (2019). Counterfactual visual explanations. In *ICML*.
- Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. (2020). Your classifier is secretly an energy based model and you should treat it like one. *ICLR*.
- Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, **37**(3), 362–386.

- Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. (2017). On calibration of modern neural networks. In *ICML*.
- Harlan, E. and Schnuck, O. (2021). Objective or biased: On the questionable use of artificial intelligence for job applications. *Bayerischer Rundfunk*.
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*.
- Hein, M. and Andriushchenko, M. (2017). Formal guarantees on the robustness of a classifier against adversarial manipulation. In *NeurIPS*.
- Hein, M., Andriushchenko, M., and Bitterwolf, J. (2019). Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *CVPR*.
- Hendricks, L. A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., and Darrell, T. (2016). Generating visual explanations. In *ECCV*.
- Hendricks, L. A., Hu, R., Darrell, T., and Akata, Z. (2018). Grounding visual explanations. In *ECCV*.
- Hendrycks, D. and Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*.
- Hendrycks, D. and Gimpel, K. (2017a). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ICLR*.
- Hendrycks, D. and Gimpel, K. (2017b). Early methods for detecting adversarial images. In *ICLR Workshop Track Proceedings*.
- Hendrycks, D., Mazeika, M., and Dietterich, T. (2019a). Deep anomaly detection with outlier exposure. In *ICLR*. <https://github.com/hendrycks/outlier-exposure>.
- Hendrycks, D., Lee, K., and Mazeika, M. (2019b). Using pre-training can improve model robustness and uncertainty. In *ICML*.
- Hendrycks, D., Basart, S., Mazeika, M., Mostajabi, M., Steinhardt, J., and Song, D. (2022). Scaling out-of-distribution detection for real-world settings. *ICML*.
- Hsu, Y.-C., Shen, Y., Jin, H., and Kira, Z. (2020). Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *CVPR*.

- Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. (2019). Adversarial examples are not bugs, they are features. *NeurIPS*.
- Jovanović, N., Balunović, M., Baader, M., and Vechev, M. (2022). On the paradox of certified training. *TMLR*.
- Katz, G., Barrett, C., Dill, D., Julian, K., and Kochenderfer, M. (2017). Reluplex: An efficient smt solver for verifying deep neural networks. In *CAV*.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*.
- Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *ICLR*.
- Koner, R., Sinhamahapatra, P., Roscher, K., Günnemann, S., and Tresp, V. (2021). Oodformer: Out-of-distribution detection transformer. *British Machine Vision Conference*.
- Kos, J., Fischer, I., and Song, D. (2017). Adversarial examples for generative models. In *ICLR Workshop*.
- Krasin, I., Duerig, T., Alldrin, N., Ferrari, V., Abu-El-Haija, S., Kuznetsova, A., Rom, H., Uijlings, J., Popov, S., Kamali, S., Mallocci, M., Pont-Tuset, J., Veit, A., Belongie, S., Gomes, V., Gupta, A., Sun, C., Chechik, G., Cai, D., Feng, Z., Narayanan, D., and Murphy, K. (2017). Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from <https://storage.googleapis.com/openimages/web/index.html>*.
- Krause, J., Stark, M., Deng, J., and Fei-Fei, L. (2013). 3d object representations for fine-grained categorization. In *ICCV vision workshop*.
- Kristiadi, A., Hein, M., and Hennig, P. (2020). Being Bayesian, Even Just a Bit, Fixes Overconfidence in ReLU Networks. In *ICML*.
- Kristiadi, A., Hein, M., and Hennig, P. (2021). An infinite-feature extension for bayesian relu nets that fixes their asymptotic overconfidence. *NeurIPS*.
- Kristiadi, A., Hein, M., and Hennig, P. (2022). Being a bit frequentist improves bayesian neural networks. In *International Conference on Artificial Intelligence and Statistics*. PMLR.
- Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, Citeseer.

- Kull, M., Perello Nieto, M., Kängsepp, M., Silva Filho, T., Song, H., and Flach, P. (2019). Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *NeurIPS*.
- Kumar, A., Sarawagi, S., and Jain, U. (2018). Trainable calibration measures for neural networks from kernel mean embeddings. In *ICML*. PMLR.
- Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Mallocci, M., Kolesnikov, A., *et al.* (2020). The open images dataset v4. *International Journal of Computer Vision*.
- Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*.
- Laptev, D., Savinov, N., Buhmann, J., and Pollefeys, M. (2016). TI-pooling: Transformation-invariant pooling for feature learning in convolutional neural networks. In *CVPR*.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86**(11), 2278–2324.
- Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. (2019). Certified robustness to adversarial examples with differential privacy. In *IEEE Symposium on Security and Privacy (SP)*.
- Lee, K., Lee, H., Lee, K., and Shin, J. (2017). Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv:1711.09325*.
- Lee, K., Lee, K., Lee, H., and Shin, J. (2018). A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*.
- Lee, S., Park, C., Lee, H., Yi, J., Lee, J., and Yoon, S. (2021a). Removing undesirable feature contributions using out-of-distribution data. In *ICLR*.
- Lee, S., Lee, W., Park, J., and Lee, J. (2021b). Towards better understanding of training certifiably robust models against adversarial examples. *NeurIPS*.
- Leibig, C., Allken, V., Ayhan, M. S., Berens, P., and Wahl, S. (2017). Leveraging uncertainty information from deep neural networks for disease detection. *Scientific Reports*, **7**.
- Li, B., Chen, C., Wang, W., and Carin, L. (2019). Certified adversarial robustness with additive noise. In *NeurIPS*.
- Liang, S., Li, Y., and Srikant, R. (2018). Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*.
- Lin, Z., Roy, S. D., and Li, Y. (2021). Mood: Multi-level out-of-distribution detection. In *CVPR*.

- Liu, J. Z., Padhy, S., Ren, J., Lin, Z., Wen, Y., Jerfel, G., Nado, Z., Snoek, J., Tran, D., and Lakshminarayanan, B. (2022). A simple approach to improve single-model deep uncertainty via distance-awareness. *arXiv:2205.00403*.
- Liu, S., Garrepalli, R., Dietterich, T., Fern, A., and Hendrycks, D. (2018). Open category detection with PAC guarantees. In *PMLR*.
- Liu, W., Wang, X., Owens, J., and Li, Y. (2020). Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*.
- Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *ICCV*.
- Lopez, A. R., Giro-i Nieto, X., Burdick, J., and Marques, O. (2017). Skin lesion classification from dermoscopic images using deep learning techniques. In *2017 13th IASTED international conference on biomedical engineering (BioMed)*, pages 49–54. IEEE.
- Macêdo, D., Ren, T. I., Zanchettin, C., Oliveira, A. L. I., and Ludermit, T. (2021). Entropic out-of-distribution detection. In *2021 International Joint Conference on Neural Networks (IJCNN)*.
- Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Valdu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *ICLR*.
- Maji, S., Rahtu, E., Kannala, J., Blaschko, M., and Vedaldi, A. (2013). Fine-grained visual classification of aircraft. *arXiv:1306.5151*.
- Meinke, A. and Hein, M. (2020). Towards neural networks that provably know when they don't know. In *ICLR*.
- Meinke, A., Bitterwolf, J., and Hein, M. (2022). Provably robust detection of out-of-distribution data (almost) for free. In *NeurIPS*.
- Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, **267**, 1 – 38.
- Ming, Y., Fan, Y., and Li, Y. (2022). Poem: Out-of-distribution detection with posterior sampling. In *ICML*.
- Mirman, M., Gehr, T., and Vechev, M. (2018). Differentiable abstract interpretation for provably robust neural networks. In *ICML*.
- Mosbach, M., Andriushchenko, M., Trost, T., Hein, M., and Klakow, D. (2018). Logit pairing methods can fool gradient-based attacks. In *NeurIPS 2018 Workshop on Security in Machine Learning*.

- Naeni, M. P., Cooper, G., and Hauskrecht, M. (2015). Obtaining well calibrated probabilities using bayesian binning. In *AAAI*.
- Najafi, A., Maeda, S.-i., Koyama, M., and Miyato, T. (2019). Robustness to adversarial perturbations in learning from incomplete data. In *NeurIPS*.
- Nalisnick, E., Matsukawa, A., Whye Teh, Y., Gorur, D., and Lakshminarayanan, B. (2019). Do deep generative models know what they don't know? In *ICLR*.
- Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *AISTATS*.
- Nilsback, M.-E. and Zisserman, A. (2008). Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*.
- Nixon, J., Dusenberry, M. W., Zhang, L., Jerfel, G., and Tran, D. (2019). Measuring calibration in deep learning. In *CVPR Workshops*.
- Pang, T., Zhang, H., He, D., Dong, Y., Su, H., Chen, W., Zhu, J., and Liu, T.-Y. (2021). Adversarial training with rectified rejection. *arXiv:2105.14785*.
- Papadopoulos, A.-A., Rajati, M. R., Shaikh, N., and Wang, J. (2021). Outlier exposure with confidence control for out-of-distribution detection. *Neurocomputing*, **441**, 138–150.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035.
- Perez, F. and Ribeiro, I. (2022). Ignore previous prompt: Attack techniques for language models. *arXiv:2211.09527*.
- Platt, J. *et al.* (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*.
- Popordanoska, T., Sayer, R., and Blaschko, M. B. (2022). A consistent and differentiable lp canonical calibration error estimator. In *NeurIPS*.
- Raghunathan, A., Steinhardt, J., and Liang, P. (2018a). Certified defenses against adversarial examples. In *ICLR*.
- Raghunathan, A., Steinhardt, J., and Liang, P. S. (2018b). Semidefinite relaxations for certifying robustness to adversarial examples. *Advances in Neural Information Processing Systems*, **31**.

- Rebuffi, S.-A., Gowal, S., Calian, D. A., Stimberg, F., Wiles, O., and Mann, T. A. (2021). Data augmentation can improve robustness. *Advances in Neural Information Processing Systems*, **34**, 29935–29948.
- Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. (2018). Do cifar-10 classifiers generalize to cifar-10? arXiv:1806.00451.
- Ren, J., Liu, P. J., Fertig, E., Snoek, J., Poplin, R., DePristo, M. A., Dillon, J. V., and Lakshminarayanan, B. (2019). Likelihood ratios for out-of-distribution detection. In *NeurIPS*.
- Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *ICML*.
- Rice, L., Wong, E., and Kolter, J. Z. (2020). Overfitting in adversarially robust deep learning. In *ICML*.
- Roelofs, R., Cain, N., Shlens, J., and Mozer, M. C. (2022). Mitigating bias in calibration error estimation. In *International Conference on Artificial Intelligence and Statistics*. PMLR.
- Rony, J., Hafemann, L. G., Oliveira, L. S., Ayed, I. B., Sabourin, R., and Granger, E. (2019). Decoupling direction and norm for efficient gradient-based L2 adversarial attacks and defenses. In *CVPR*.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*.
- Saba, T., Mohamed, A. S., El-Affendi, M., Amin, J., and Sharif, M. (2020). Brain tumor detection using fusion of hand crafted and deep learning features. *Cognitive Systems Research*, **59**, 221–230.
- Salman, H., Yang, G., Zhang, H., Hsieh, C.-J., and Zhang, P. (2019). A convex relaxation barrier to tight robustness verification of neural networks. *NeurIPS*.
- Salman, H., Ilyas, A., Engstrom, L., Kapoor, A., and Madry, A. (2020). Do adversarially robust imagenet models transfer better? *NeurIPS*.
- Samangouei, P., Saeedi, A., Nakagawa, L., and Silberman, N. (2018). Explaingan: Model explanation via decision boundary crossing transformations. In *ECCV*.
- Santurkar, S., Tsipras, D., Tran, B., Ilyas, A., Engstrom, L., and Madry, A. (2019). Computer vision with a single (robust) classifier. In *NeurIPS*.
- Schott, L., Rauber, J., Bethge, M., and Brendel, W. (2019). Towards the first adversarially robust neural network model on mnist. In *ICLR*.

- Sehwag, V., Bhagoji, A. N., Song, L., Sitawarin, C., Cullina, D., Chiang, M., and Mittal, P. (2019). Better the devil you know: An analysis of evasion attacks using out-of-distribution adversarial examples. *preprint, arXiv:1905.01726*.
- Sensoy, M., Kaplan, L., and Kandemir, M. (2018). Evidential deep learning to quantify classification uncertainty. In *NeurIPS*.
- Shafaei, A., Schmidt, M., and Little, J. (2019). A Less Biased Evaluation of Out-of-distribution Sample Detectors. In *BMVC*.
- Sheikholeslami, F., Lotfi, A., and Kolter, J. Z. (2020). Provably robust classification of adversarial examples with detection. In *ICLR*.
- Stutz, D., Hein, M., and Schiele, B. (2019). Disentangling adversarial robustness and generalization. In *CVPR*.
- Stutz, D., Hein, M., and Schiele, B. (2020). Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *ICML*. PMLR.
- Stutz, D., Chandramoorthy, N., Hein, M., and Schiele, B. (2021). Bit error robustness for energy-efficient dnn accelerators. *Proceedings of Machine Learning and Systems*.
- Sun, Y., Guo, C., and Li, Y. (2021). React: Out-of-distribution detection with rectified activations. *NeurIPS*.
- Sutton, R. (2019). The bitter lesson. *Incomplete Ideas (blog)*.
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks. In *ICLR*.
- Tajwar, F., Kumar, A., Xie, S. M., and Liang, P. (2021). No true state-of-the-art? ood detection methods are inconsistent across datasets. In *ICML UDL Workshop 2021*.
- Thulasidasan, S., Thapa, S., Dhaubhadel, S., Chennupati, G., Bhattacharya, T., and Bilmes, J. (2021). An effective baseline for robustness to distributional shift. *arXiv: 2105.07107*.
- Torralba, A., Fergus, R., and Freeman, W. T. (2008). 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*.
- Tramer, F. (2022). Detecting adversarial examples is (nearly) as hard as classifying them. In *ICML*. PMLR.
- Tramer, F., Carlini, N., Brendel, W., and Madry, A. (2020). On adaptive attacks to adversarial example defenses. *NeurIPS*.

- Tramèr, F. and Boneh, D. (2019). Adversarial training and robustness for multiple perturbations. In *NeurIPS*.
- Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. (2019). Robustness may be at odds with accuracy. In *International Conference on Learning Representations*.
- Uesato, J., Alayrac, J.-B., Huang, P.-S., Stanforth, R., Fawzi, A., and Kohli, P. (2019). Are labels required for improving adversarial robustness? In *NeurIPS*.
- Vaicenavicius, J., Widmann, D., Andersson, C., Lindsten, F., Roll, J., and Schön, T. (2019). Evaluating model calibration in classification. In *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., *et al.* (2020). Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*.
- Wachter, S., Mittelstadt, B., and Russell, C. (2018). Counterfactual explanations without opening the black box: automated decisions and the GDPR. *Harvard Journal of Law and Technology*, **31**(2), 841–887.
- Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., and Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*.
- Wong, E. and Kolter, J. Z. (2018). Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML*.
- Wong, E. and Kolter, J. Z. (2021). Learning perturbation sets for robust machine learning. In *ICLR*.
- Wong, E., Schmidt, F., Metzen, J. H., and Kolter, J. Z. (2018). Scaling provable adversarial defenses. In *NeurIPS*.
- Wong, E., Rice, L., and Kolter, J. Z. (2020). Fast is better than free: Revisiting adversarial training. In *ICLR*.
- Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. preprint, arXiv:1708.07747.
- Xiao, Z., Yan, Q., and Amit, Y. (2020). Likelihood regret: An out-of-distribution detection score for variational auto-encoder. In *NeurIPS*.
- Xu, W., Evans, D., and Qi, Y. (2017). Feature squeezing: Detecting adversarial examples in deep neural networks. In *Network and Distributed System Security Symposium*.
- Yang, J., Zhou, K., Li, Y., and Liu, Z. (2021). Generalized out-of-distribution detection: A survey. *arXiv:2110.11334*.

- Yang, J., Wang, P., Zou, D., Zhou, Z., Ding, K., Peng, W., Wang, H., Chen, G., Li, B., Sun, Y., Du, X., Zhou, K., Zhang, W., Hendrycks, D., Li, Y., and Liu, Z. (2022). Openood: Benchmarking generalized out-of-distribution detection. *NeurIPS*.
- Yin, X., Li, S., and Rohde, G. K. (2022). Learning energy-based models with adversarial training. In *ECCV*. Springer.
- Yoon, S., Choi, J., Lee, Y., Noh, Y.-K., and Park, F. C. (2022). Evaluating out-of-distribution detectors through adversarial generation of outliers. *arXiv preprint arXiv:2208.10940*.
- Yu, F., Zhang, Y., Song, S., Seff, A., and Xiao, J. (2015). Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, **abs/1506.03365**.
- Yu, Q. and Aizawa, K. (2019). Unsupervised out-of-distribution detection by maximum classifier discrepancy. In *ICCV*.
- Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. In *BMVC*, pages 87.1–87.12.
- Zajac, M., Zołna, K., Rostamzadeh, N., and Pinheiro, P. O. (2019). Adversarial framing for image and video classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *ECCV*.
- Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. (2019). Theoretically principled trade-off between robustness and accuracy. In *ICML*.
- Zhang, H., Chen, H., Xiao, C., Gowal, S., Stanforth, R., Li, B., Boning, D., and Hsieh, C.-J. (2020a). Towards stable and efficient training of verifiably robust neural networks. In *ICLR*.
- Zhang, J., Kailkhura, B., and Han, T. Y.-J. (2020b). Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *ICML*. PMLR.
- Zhao, S., Ma, T., and Ermon, S. (2020). Individual calibration with randomized forecasting. In *ICML*. PMLR.
- Zhu, J.-Y., Krähenbühl, P., Shechtman, E., and Efros, A. A. (2016). Generative visual manipulation on the natural image manifold. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *ECCV*.
- Álvaro Parafita and Vitrià, J. (2019). Explaining visual models by causal attribution. In *ICCV Workshop on XCAI*.