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

Alexander Meinke [*1] **Julian Bitterwolf** [1] **Matthias Hein** [1]

Abstract

Deep neural networks are known to produce highly overconfident predictions on out-of-distribution (OOD) data and even if trained to be non-confident on OOD data one can still adversarially manipulate OOD data so that the classifer again assigns high confidence to the manipulated samples. In this paper we propose a novel method where from first principles we combine a certifiable OOD detector with a standard classifier into an OOD aware classifier. In this way we achieve the best of two worlds: certifiably adversarially robust OOD detection, even for OOD samples close to the in-distribution, without loss in prediction accuracy and close to state-of-the-art OOD detection performance for non-manipulated OOD data. Moreover, due to the particular construction our classifier provably avoids the asymptotic overconfidence problem of standard neural networks.

1. Introduction

Many approaches have been proposed for OOD detection, (Hendrycks & Gimpel, 2017; Liang et al., 2018; Lee et al., 2018a;b; Hendrycks et al., 2019; Ren et al., 2019; Hein et al., 2019; Meinke & Hein, 2020; Chen et al., 2020; Papadopoulos et al., 2021) and one of the currently best performing methods enforces low confidence during training on a large and diverse set of out-distribution images (Hendrycks et al., 2019) which leads to strong separation of in- and out-distribution based on the confidence of the classifier. Crucially, this also generalizes to novel test out-distributions. However, current OOD detection methods are vulnerable to adversarial manipulations, i.e. small adversarial modifications of OOD inputs lead to large confidence of the classifier on the manipulated samples (Nguyen et al., 2015; Hein et al., 2019; Sehwag et al., 2019). While different methods for adversarially robust OOD detection have been proposed (Hein et al., 2019; Sehwag et al., 2019; Meinke & Hein, 2020; Chen et al., 2020; Bitterwolf et al., 2020) there is little work on *provably* robust OOD detection (Meinke & Hein, 2020; Bitterwolf et al., 2020; Kopetzki et al., 2020; Berrada et al., 2021).

In (Meinke & Hein, 2020) they append density estimators based on Gaussian mixture models for in- and out-distribution to the softmax layer, so they can guarantee that the classifier shows decreasing confidence as one moves away from the training data. However, for close in-distribution inputs this approach yields no guarantee as the Gaussian mixture models are not powerful enough for complex image classification tasks. In (Kristiadi et al., 2020a;b) similar asymptotic guarantees are derived for Bayesian neural networks but without any robustness guarantees. In (Kopetzki et al., 2020) they apply randomized smoothing to obtain guarantees wrt l_2-perturbations for Dirchlet-based models (Malinin & Gales, 2018; 2019; Sensoy et al., 2018) which already show quite some gap in terms of AUC-ROC to SOTA OOD detection methods even without attacks. Interval bound propagation (IBP) (Gowal et al., 2018; Mirman et al., 2018; Zhang et al., 2020; Jovanović et al., 2021) has been shown to be one of the most effective techniques in certified adversarial robustness on the in-distribution when applied during training. In (Bitterwolf et al., 2020) they use IBP to compute upper bounds on the confidence in an l_∞-neighborhood of the input and minimize these upper bounds on the training out-distribution. This yields classifiers with pointwise guarantees for robust OOD detection even for "close" out-distribution inputs which generalize to novel OOD test distributions. However, the employed architectures of the neural network are restricted to rather shallow networks as otherwise the bounds of IBP are loose. Thus, they obtain a classification accuracy which is far away from the state-of-the-art, e.g. 91% on CIFAR-10, and thus the approach does not scale to more complex tasks like ImageNet. Moreover, one does not get any guarantees on the asymptotic behavior far from the data.

In this paper we propose a framework which merges a certified binary classifier for in-versus out-distribution with a classifier for the in-distribution task in a principled fashion into a joint classifier which combines the advantages of (Meinke & Hein, 2020) and (Bitterwolf et al., 2020) but does not suffer from the downsides of the respective ap-

[*]Equal contribution [1]Department of Computer Science, University of Tübingen, Germany. Correspondence to: Alexander Meinke <alexander.meinke@uni-tuebingen.de>.

Presented at the ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning.,

proaches. In particular, our method simultaneously achieves the following properties: 1) Point-wise l_∞-robust OOD detection with guarantees similar to (Bitterwolf et al., 2020). 2) It provably prevents the asymptotic overconfidence of deep neural networks. 3) 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 robust OOD detection, fix the asymptotic overconfidence (almost) for free as we have (almost) no loss in prediction and standard OOD detection performance.

2. Provably Robust Detection of Out-of-distribution Data

In the following we consider feedforward networks for classification, $f : \mathbb{R}^d \to \mathbb{R}^K$, with K classes. We get a probability distribution over the classes via $\hat{p}(y|x) = \frac{e^{f_y(x)}}{\sum_k^K e^{f_k(x)}}$ for $y = 1, \dots, K$. We define the confidence as $\mathrm{Conf}(f(x)) = \max_{y=1,\dots,K} \hat{p}(y|x)$ and assume that only piece-wise linear non-linearities are used.

2.1. Joint Model for OOD Detection and Classification

In our joint model we assume that there exists an in- and out-distribution where we assume that out-distribution samples are unrelated to the in-distribution task. Thus we can formally write the conditional distribution on the input as

$$\hat{p}(y|x) = \hat{p}(y|x, i)\hat{p}(i|x) + \hat{p}(y|x, o)\hat{p}(o|x), \qquad (1)$$

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 $\hat{p}(y|x, o) = \frac{1}{K}$. In (Meinke & Hein, 2020) they further decomposed $\hat{p}(i|x) = \frac{\hat{p}(x|i)\hat{p}(i)}{\hat{p}(x)}$ and used Gaussian mixture models to estimate $\hat{p}(x|i)$ with fixed $\hat{p}(i) = \hat{p}(o) = \frac{1}{2}$. Instead in this paper we directly learn $\hat{p}(i|x)$ which results in a binary classification problem and we train this binary classifier in a certified robust fashion wrt an l_∞-threat model so that even adversarially manipulated OOD samples are detected. Around $x \in \mathbb{R}^d$ we have the upper bound

$$\max_{\|u-x\|_\infty \leq \epsilon} \hat{p}(y|u) \leq \frac{K-1}{K} \max_{\|u-x\|_\infty \leq \epsilon} \hat{p}(i|u) + \frac{1}{K}, \quad (2)$$

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 in (Meinke & Hein, 2020) but additionally get certified adversarial robustness for close out-distribution samples as in (Bitterwolf et al., 2020). In contrast to (Bitterwolf et al., 2020) this comes without loss in test accuracy or non-adversarial OOD detection performance as 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 **Pr**ovable **o**ut-**o**f-**D**istribution detector (ProoD).

Certifiably Robust Binary Discrimination of In- versus Out-Distribution The first goal is to get a certifiably 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 $g : \mathbb{R}^d \to \mathbb{R}$ are logits of a neural network (we denote the weights and biases of g by W_g and b_g in order to discriminate 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 out-distribution) and $(z_s)_{s=1}^M$ be our training out-distribution data. Then the optimization problem associated to the binary classification problem becomes:

$$\min_{\substack{g \\ W_g^{(L_g)} < 0}} \frac{1}{N} \sum_{r=1}^{N} \log\left(1+e^{-g(x_r)}\right) + \frac{1}{M} \sum_{s=1}^{M} \log\left(1+e^{\bar{g}(z_s)}\right), \qquad (3)$$

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 $B_p(z, \epsilon) = \{u \in [0, 1]^d \mid \|u - z\|_p \leq \epsilon\}$. In this paper we always use an l_∞-threat model. This upper bound could, in principle, be computed using any certification technique but we will use interval bound propagation (IBP) since it is simple, fast and has been shown to produce SOTA results (Gowal et al., 2018). 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 wrt 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 constraint on the weights of the output layer $W_g^{(L_g)}$ is enforced by using the parameterization $(W_g^{(L_g)})_j = -e^{h_j}$ componentwise and optimizing over h_j. Later on, the negativity of $W_g^{(L_g)}$ allows us to control the asymptotic behavior of the joint classifier, see Section 3.

While in (Bitterwolf et al., 2020) they also used IBP to upper bound the confidence of the classifier this resulted in a bound that took into account all $\mathcal{O}(K^2)$ logit differences between all classes. In contrast, our loss in Eq. (3) is significantly

simpler as we just have a binary classification problem and therefore only need a single bound. Thus, our approach easily scales to tasks with a large number of classes and training the binary discriminator with IBP turns out to be significantly more stable than the approach in (Bitterwolf et al., 2020) and does not require many additional tricks.

(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. (1). On top of the provable OOD performance that we get from Eq. (2), 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 (Hendrycks et al., 2019) 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 $f : \mathbb{R}^d \to \mathbb{R}^K$ and need to keep $\hat{p}(i|x)$ resp. g fixed. Because g stays fixed we call this semi-joint training. We use outlier exposure (Hendrycks et al., 2019) 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:

$$\min_f -\frac{1}{N}\sum_{r=1}^{N} \log\left(\hat{p}(y_r|x_r)\right) - \frac{1}{M}\sum_{s=1}^{M}\frac{1}{K}\sum_{l=1}^{K}\log\left(\hat{p}(l|z_s)\right) \tag{4}$$

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 is the outlier exposure term which enforces uniform confidence on out-distribution samples. In App. D we show that semi-joint training does, in fact, lead to stronger guarantees than separate training.

The loss in Eq. (3) 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 $b' = b_g^{(L_g)} + \Delta$, where $\Delta \geq 0$ leads to increasing $\hat{p}(i|x)$. This shift may appear post-hoc, but it actually has a direct interpretation in terms of the probabilities $p(i)$ and $p(o)$ which we explore in App. G.

3. Guarantees on Asymptotic Confidence

We note that a ReLU neural network 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 $1/K$ as in these regions the classifier has never seen any data). The following theorem now shows that, opposite 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 desireable for any reasonable classifier.

Theorem 1. *Let $x \in \mathbb{R}^d$ with $x \neq 0$ and let $g : \mathbb{R}^d \to \mathbb{R}$ be the ReLU-network of the binary discriminator and denote by $\{Q_r\}_{r=1}^R$ the finite set of polytopes such that g is affine on these polytopes (exists by Lemma 3.1 in (Hein et al., 2019)). 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 $\lim_{\beta\to\infty} \hat{p}(y|\beta x) = \frac{1}{K}$.*

See the proof in App. E. In App. C we see that the condition $Ux \neq 0$ is not restrictive, as in all cases we tested this property it holds for our joint classifier (see Figure 1 for an illustration of the asymptotic confidence of ProoD in comparison to other models). 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. While one might think that this condition is restrictive, 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.

4. Experiments

We compare our ProoD in terms of accuracy and clean and adversarial OOD performance on several datasets with the main competitors. We provide experiments on CIFAR10, CIFAR100 (in the appendix) (Krizhevsky & Hinton, 2009) and Restricted Imagenet (R.ImgNet) (Tsipras et al., 2018). Training details are described in App. F[1].

[1]Code at github.com/AlexMeinke/Provable-OOD-Detection.

4.1. Evaluation

Setup For OOD evaluation for CIFAR10 we use the test sets from CIFAR100 and SVHN (Netzer et al., 2011). For R.ImgNet we use Flowers (Nilsback & Zisserman, 2008) and FGVC Aircraft (Maji et al., 2013) (more in the Appendix). Since the computation of adversarial AUCs (next paragraph) requires computationally expensive adversarial attacks, we restrict the evaluation on the out-distribution to a fixed subset of 1000 images for the CIFAR experiments and 400 for the R.ImgNet models. We still use the entire test set for the in-distribution.

Guaranteed and Adversarial AUC We use the confidence of the classifier as the feature to discriminate between in- and out-distribution samples. While in standard OOD detection one uses the area under the receiver-operator characteristic (AUC) to measure discrimination of in- from out-distribution, in (Bitterwolf et al., 2020) they introduced the worst-case AUC (WCAUC) which is defined as the minimal AUC one can achieve if all out-distribution samples are allowed to be perturbed to reach maximal confidence within a certain threat model, which in our case is an l_∞-ball of radius ϵ. The AUC and WAUC are then defined as:

$$\mathrm{AUC}_f(p_1, p_2) = \mathop{\mathbb{E}}_{\substack{x\sim p_1\\ z\sim p_2}} \left[\mathbb{1}_{\mathrm{Conf}(x)>\mathrm{Conf}(z)}\right],$$

$$\mathrm{WCAUC}_f(p_1, p_2) = \mathop{\mathbb{E}}_{\substack{x\sim p_1\\ z\sim p_2}} \left[\mathbb{1}_{\mathrm{Conf}(x)>\max\limits_{\|u-z\|_\infty\leq\epsilon} \mathrm{Conf}(u)}\right],$$

where p_1, p_2 are in-resp. out-distribution and with slight abuse of notation the indicator function $\mathbb{1}$ returns 1 if the expression in its argument is true and 0 otherwise. Since the exact evaluation of the WCAUC is computationally infeasible we compute an upper bound on the WCAUC, the adversarial AUC (AAUC), by maximizing the confidence using an adversarial attack inside the l_∞-ball and we compute a lower bound on the WCAUC, the guaranteed AUC (GAUC), by using bounds on the confidence inside the l_∞-ball via IBP. Gradient obfuscation (Papernot et al., 2017; Athalye et al., 2018) poses a significant challenge for the evaluation of AAUCs so we employ an ensemble of strong attacks that we discuss in App. B.

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 our ProoD. For GOOD we use the publicly available models from (Bitterwolf et al., 2020). 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 so it is included simply for reference.

Results All results are shown in Table 1 and an extended version can be found in App. A. 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 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., 2021). 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) to our ProoD. The bounds are also much tighter than for GOOD, which is likely due to the fact that the confidence on GOOD is much harder to optimize during an attack because it involves maximizing the confidence in an essentially random class. To the best of our knowledge with R.ImgNet we provide the first worst case OOD guarantees on high-resolution images. The fact that our GAUCs are comparable to those on CIFAR10 indicates that meaningful certification on higher resolution is more achievable on this task than one might expect.

Table 1. **OOD performance:** We report accuracy 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 $\epsilon = 0.01$ for all datasets. The bias shift Δ that was used for ProoD is shown for each in-distribution.

In: CIFAR10		CIFAR100			SVHN		
	Acc	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	95.01	90.0	0.0	0.6	93.8	0.0	0.1
OE	95.53	96.1	0.0	6.0	99.2	0.0	0.4
GOOD_{80}	90.13	87.2	42.5	63.9	94.2	37.5	67.4
ProoD-Disc	-	67.4	61.0	61.7	73.2	65.5	66.4
ProoD $\Delta=3$	95.47	96.0	41.9	43.9	99.5	48.8	49.4
In: R.ImgNet		**Flowers**			**FGVC**		
	Acc	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	96.34	92.3	0.0	0.5	92.6	0.0	0.0
OE	97.10	96.9	0.0	0.2	99.7	0.0	0.4
ProoD-Disc	-	81.5	76.8	77.3	92.8	89.3	89.6
ProoD $\Delta=4$	97.25	96.9	57.5	58.0	99.8	67.4	67.9

5. Conclusion

We have demonstrated how to combine a provably robust binary discriminator between in- and out-distribution with a standard classifier in order to simultaneously achieve high accuracy, high OOD detection performance as well as worst-case OOD guarantees that are comparable to previous works. We further showed how our model fixes the problem of asymptotic overconfidence in ReLU classifiers. We described how to train these networks simply and stably and thus we provide OOD guarantees (almost) for free.

Acknowledgements

The authors acknowledge support from the German Federal Ministry of Education and Research (BMBF) through the Tübingen AI Center (FKZ: 01IS18039A) and from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy (EXC-number 2064/1, Project number 390727645), as well as from the DFG TRR 248 (Project number389792660). The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Alexander Meinke. We also thank Maximilian Augustin for helpful advice.

References

Arora, R., Basuy, A., Mianjyz, P., and Mukherjee, A. Understanding deep neural networks with rectified linear unit. In *ICLR*, 2018.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

Augustin, M., Meinke, A., and Hein, M. Adversarial robustness on in-and out-distribution improves explainability. In *ECCV*, 2020.

Berrada, L., Dathathri, S., Stanforth, R., Bunel, R., Uesato, J., Gowal, S., Kumar, M. P., et al. Verifying probabilistic specifications with functional lagrangians. *arXiv:2102.09479*, 2021.

Birhane, A. and Prabhu, V. U. Large image datasets: A pyrrhic win for computer vision? In *WACV*, 2021.

Bitterwolf, J., Meinke, A., and Hein, M. Certifiably adversarially robust detection of out-of-distribution data. *NeurIPS*, 2020.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017a.

Carlini, N. and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In *ACM Workshop on Artificial Intelligence and Security*, 2017b.

Chen, J., Li, Y., Wu, X., Liang, Y., and Jha, S. Informative outlier matters: Robustifying out-of-distribution detection using outlier mining. *preprint, arXiv:2006.15207*, 2020.

Cheng, C.-H., Nührenberg, G., and Ruess, H. Maximum resilience of artificial neural networks. In *International Symposium on Automated Technology for Verification and Analysis*, 2017.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *CVPR*, 2019.

Dathathri, S., Dvijotham, K., Kurakin, A., Raghunathan, A., Uesato, J., Bunel, R., Shankar, S., Steinhardt, J., Goodfellow, I., Liang, P., et al. Enabling certification of verification-agnostic networks via memory-efficient semidefinite programming. In *NeurIPS*, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T., and Kohli, P. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv:1810.12715*, 2018.

Hein, M., Andriushchenko, M., and Bitterwolf, J. Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *CVPR*, 2019.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.

Hendrycks, D., Mazeika, M., and Dietterich, T. Deep anomaly detection with outlier exposure. In *ICLR*, 2019.

Jovanović, N., Balunović, M., Baader, M., and Vechev, M. Certified defenses: Why tighter relaxations may hurt training? *arXiv:2102.06700*, 2021.

Katz, G., Barrett, C., Dill, D., Julian, K., and Kochenderfer, M. Reluplex: An efficient smt solver for verifying deep neural networks. In *CAV*, 2017.

Kopetzki, A.-K., Charpentier, B., Zügner, D., Giri, S., and Günnemann, S. Evaluating robustness of predictive uncertainty estimation: Are dirichlet-based models reliable? *arXiv:2010.14986*, 2020.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *ICCV vision workshop*, 2013.

Kristiadi, A., Hein, M., and Hennig, P. Being Bayesian, Even Just a Bit, Fixes Overconfidence in ReLU Networks. In *ICML*, 2020a.

Kristiadi, A., Hein, M., and Hennig, P. Fixing asymptotic uncertainty of bayesian neural networks with infinite relu features. *arXiv:2010.02709*, 2020b.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al. The open images dataset v4. *International Journal of Computer Vision*, 2020.

Lee, K., Lee, H., Lee, K., and Shin, J. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *ICLR*, 2018a.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018b.

Liang, S., Li, Y., and Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Valdu, A. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Maji, S., Rahtu, E., Kannala, J., Blaschko, M., and Vedaldi, A. Fine-grained visual classification of aircraft. *arXiv:1306.5151*, 2013.

Malinin, A. and Gales, M. Predictive uncertainty estimation via prior networks. In *NeurIPS*, 2018.

Malinin, A. and Gales, M. Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness. In *NeurIPS*, 2019.

Meinke, A. and Hein, M. Towards neural networks that provably know when they don't know. In *ICLR*, 2020.

Mirman, M., Gehr, T., and Vechev, M. Differentiable abstract interpretation for provably robust neural networks. In *ICML*, 2018.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Nguyen, A., Yosinski, J., and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 2008.

Papadopoulos, A.-A., Rajati, M. R., Shaikh, N., and Wang, J. Outlier exposure with confidence control for out-of-distribution detection. *Neurocomputing*, 2021.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against machine learning. In *ACM ASIACCS*, 2017.

Ren, J., Liu, P. J., Fertig, E., Snoek, J., Poplin, R., DePristo, M. A., Dillon, J. V., and Lakshminarayanan, B. Likelihood ratios for out-of-distribution detection. In *NeurIPS*, 2019.

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. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015.

Sehwag, V., Bhagoji, A. N., Song, L., Sitawarin, C., Cullina, D., Chiang, M., and Mittal, P. Better the devil you know: An analysis of evasion attacks using out-of-distribution adversarial examples. *preprint, arXiv:1905.01726*, 2019.

Sensoy, M., Kaplan, L., and Kandemir, M. Evidential deep learning to quantify classification uncertainty. In *NeurIPS*, 2018.

Torralba, A., Fergus, R., and Freeman, W. T. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 30(11):1958–1970, 2008.

Tramer, F., Carlini, N., Brendel, W., and Madry, A. On adaptive attacks to adversarial example defenses. *arXiv:2002.08347*, 2020.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. In *ICLR*, 2018.

Yu, F., Zhang, Y., Song, S., Seff, A., and Xiao, J. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015.

Zhang, H., Chen, H., Xiao, C., Gowal, S., Stanforth, R., Li, B., Boning, D., and Hsieh, C.-J. Towards stable and efficient training of verifiably robust neural networks. In *ICLR*, 2020.

A. Results on all Datasets

We present an extension of Table 1 that additionally includes CIFAR100 as in-distribution as well as additional test out-distributions; specifically, the classroom category of downscaled LSUN (Yu et al., 2015) (LSUN_CR), Stanford Cars (Krause et al., 2013) as well as smooth noise as suggested in (Hein et al., 2019) and described in App. F. We also show additional baselines on CIFAR10. For both ATOM and ACET we use the publicly available pre-trained Densenets from (Chen et al., 2020) (note that these are only available for CIFAR10/100). Since CCU was already shown to not provide benefits over OE on OOD data that is not very far from the in-distribution (e.g. uniform noise) (Meinke & Hein, 2020; Bitterwolf et al., 2020) we do not include it as a baseline.

ProoD's GAUCs are higher than the AAUCs of ATOM on every evaluated dataset and thus ProoD is *provably better* at the task than ATOM. This may seem surprising because the authors of (Chen et al., 2020) claimed far stronger adversarial OOD performance on the much harder threat model of $\epsilon = 8/255 > 0.03$ (compared to our $\epsilon = 0.01$). For example, they report an AAUC of 83.78% on CIFAR10 vs. SVHN compared to the 8.6% that we find within the weaker threat model. Similar failures of adversarial robustness on the in the in-distribution are common in the literature (Tramer et al., 2020; Carlini & Wagner, 2017a;b; Athalye et al., 2018; Croce & Hein, 2020) and this result shows emphatically why certified adversarial OOD robustness is so important as empirical evaluations can be unreliable. We can see a similar issue with ACET. On CIFAR10 it appears that it is quite robust to our attacks, but on CIFAR100 it fails completely with AAUCs below even those of OE. Evidently the training on CIFAR100 failed, but only very strong attacks can show this. Interestingly, only the non-robust ACET model has no drop in accuracy, corroborating the findings in (Augustin et al., 2020).

On CIFAR10 we see that ProoD's GAUCs are comparable to, if slightly worse than the ones of GOOD_{80} and strictly worse than the GAUCs of GOOD_{100}. 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, the accuracy and OOD performance of ProoD are comparable to OE. On CIFAR100 the accuracy and the clean AUC against CIFAR10 are somewhat smaller than for OE, by 0.5% and 3.4% respectively. Together with the failure of GOOD to train at all, the failure of ACET to train robustly and the low clean AUCs of ATOM against CIFAR10 (13.4% worse than plain) this may indicate that obtaining OOD robustness on a task with this many classes is very challenging.

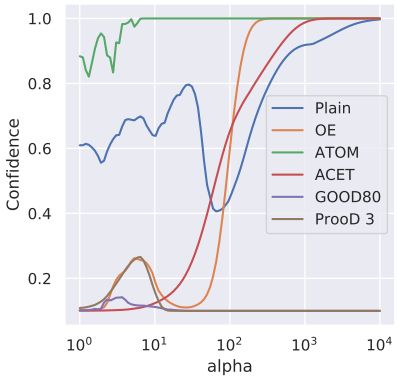

Figure 1. **Asymptotic confidence:** We plot the mean confidence in the predicted in-distribution class for different models as one moves away from the center of the box $[0,1]^d$ by α multiples for 10 fixed vectors n_i (adversarially found for ATOM, see App. C for details). Plain, OE and ACET become asymptotically overconfident as expected by the result of (Hein et al., 2019). However, as shown in Theorem 1, under mild assumptions our ProoD asymptotically converges to uniform confidence. GOOD_{80} (Bitterwolf et al., 2020) also converges to uniform confidence along these directions even though no guarantee was shown in (Bitterwolf et al., 2020).

B. Adversarial Attacks

We employ an ensemble of different versions of projected gradient descent (PGD) (Madry et al., 2018). We use APGD (Croce & Hein, 2020) (except on RImgNet, due to a memory leak) with 500 iterations and 5 random restarts. We also use a 200-step PGD attack with momentum of 0.9 and backtracking that starts with a step size of 0.1 which is halved every time a gradient step does not increase the confidence and gets multiplied by 1.1 otherwise. This PGD is applied to different starting points: i) a decontrasted version of the image, i.e. the point that minimizes the l_∞-distance to the grey image $0.5 \cdot \vec{1}$ within the threat model, ii) 3 uniformly drawn samples from the threat model and iii) 3 versions of the original image perturbed by Gaussian noise with $\sigma = 10^{-4}$ and then clipped to the threat model. We always clip to the box $[0,1]^d$ at each step of the attack. Using different types of starting points is crucial for strong attacks on these OOD points, as some models have precisely 0 gradients on many OOD samples.

C. Adversarial Asymptotic Overconfidence

According to the authors of (Hein et al., 2019), under mild conditions, we should expect to find asymptotic overconfidence in all ReLU networks and almost all directions. For Plain, OE and ACET we did indeed observe this. However, the theorem in (Hein et al., 2019) does not apply to ATOM since it uses an out-class. It can therefore happen that ATOM only ever gets more confident in the out-class

Table 2. **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 $\epsilon = 0.01$ for all datasets. The bias shift Δ that was used for ProoD is shown for each in-distribution. ProoD provides guarantees that are strictly better than ATOM's empirical performance on the adversarial task, all while retaining the same accuracy as Plain and OE. The AAUCs and GAUCs for ProoD tend to be very close, indicating remarkably tight certification bounds. Note that ATOM and ACET have lower accuracy on CIFAR100, because their architecture is smaller.

In: CIFAR10		CIFAR100			SVHN			LSUN_CR			Smooth		
	Acc	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	95.01	90.0	0.0	0.6	93.8	0.0	0.1	93.1	0.0	0.5	98.2	0.0	0.6
OE	95.53	96.1	0.0	6.0	99.2	0.0	0.4	99.5	0.0	15.2	99.0	0.0	11.3
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
GOOD_{80}	90.13	87.2	42.5	63.9	94.2	37.5	67.4	93.3	55.2	83.6	95.3	57.3	88.5
GOOD_{100}	90.14	70.7	54.5	55.0	74.9	56.3	56.6	75.2	61.0	61.6	81.4	66.6	67.5
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
In: CIFAR100		CIFAR10			SVHN			LSUN_CR			Smooth		
	Acc	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	77.38	77.7	0.0	0.3	81.9	0.0	0.2	76.4	0.0	0.3	88.8	0.0	0.5
OE	77.28	83.9	0.0	0.8	92.8	0.0	0.1	97.4	0.0	4.6	97.6	0.0	0.9
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
In: R.ImgNet		Flowers			FGVC			Cars			Smooth		
	Acc	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

as we move further away from the training data. Empirically, this does appear to be true for the majority of directions. However, as shown in Figure 1, it is possible to find directions in which ATOM becomes arbitrarily overconfident in an in-distribution class. The way we found these directions is as follows: 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 2000 steps), minimizing $f_{K+1}(x) - \max_{k \in \{1,\dots K\}} f_k(x)$ while projecting onto the sphere at each step (unnormalized gradients with step size 0.1 for the first 1000 steps and 0.01 for the last 1000 steps). We then increase the radius of the sphere to 1000 and run an additional 2000 steps with a step size of 0.1. The directions that receive high confidence in an in-distribution class by ATOM tend to also remain in-distribution when scaled up by arbitrarily large factors. The mean confidence over 10 such directions is shown in Figure 1.

It is interesting to ask if similar directions can also be found for ProoD. Of course, the architecture provably prevents arbitrarily overconfident predictions and Theorem 1 ensures that most directions will indeed converge to uniform, but 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 1 is not satisfied. We attempted to find such directions by running an attack similar to the one described above. But even using as many as 50000 iterations using various schedules for the radius and the step size, we were unable to find directions where the confidence did not become uniform asymptotically.

In Figure 1 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.

D. Separate Training for ProoD

In Section 2.1 we describe semi-joint training of $\hat{p}(y|x)$. However, as pointed out in that section, it is possible to separately train a certifiable binary discriminator $\hat{p}(i|x)$ and an OOD aware classifier $\hat{p}(y|x,i)$ and to then simply combine them via Eq. (1). We call this method of separate training ProoD-S and evaluate it by using an OE trained model for $\hat{p}(y|x,i)$. We show the results in Table 3, where we repeat the results for OE and ProoD for the reader's convenience. Note that OE and ProoD-S must always have the same accuracy on the in-distribution since they use the same model for classification.

We see that the AUCs of ProoD-S are almost identical to those of OE. Even without any loss in performance ProoD-S manages to provide non-trivial GAUCs. However, as one would expect, the semi-jointly trained ProoD provides stronger guarantees at similar clean performance. Nonetheless, this post-hoc method of adding some amount of certifiability to an existing system may be interesting in applications where retraining a deployed model from scratch is infeasible.

E. Proof of Theorem 1

Theorem 1. *Let $x \in \mathbb{R}^d$ with $x \neq 0$ and let $g : \mathbb{R}^d \to \mathbb{R}$ be the ReLU-network of the binary discriminator and denote by $\{Q_r\}_{r=1}^R$ the finite set of polytopes such that g is affine on these polytopes (exists by Lemma 3.1 in (Hein et al., 2019)). 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 $\lim_{\beta \to \infty} \hat{p}(y|\beta x) = \frac{1}{K}$.*

Proof. We note that with a similar argument as in the derivation of (2) it holds

$$\hat{p}(y|\beta x) \leq \hat{p}(i|\beta x) \tag{5}$$

$$+ \frac{1}{K}\big(1 - \hat{p}(i|\beta x)\big) = \frac{K-1}{K}\hat{p}(i|\beta x) + \frac{1}{K}. \tag{6}$$

For all $\beta \geq \alpha$ it holds that $\beta x \in Q_t$ so that

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

As $x_i^{(L-1)}(x) \geq 0$ for all $x \in \mathbb{R}^d$ it has to hold $(\beta U x + d)_i \geq 0$ for all $\beta \geq \alpha$ and $i = 1, \ldots, n_{L-1}$. This implies that $(Ux)_i \geq 0$ for all $i = 1, \ldots, n_{L-1}$ 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

$$g(\beta x) = \left\langle W_g^{(L_g)}, U\beta x + d \right\rangle + b_g^{(L_g)} \tag{7}$$

$$= \beta \left\langle W_g^{(L_g)}, Ux \right\rangle + \left\langle W_g^{(L_g)}, d \right\rangle + b_g^{(L_g)}. \tag{8}$$

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

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

Plugging this into (5) yields the result. □

F. Experimental Details

Datasets We use CIFAR10 and CIFAR100 (Krizhevsky & Hinton, 2009) (MIT license), SVHN (Netzer et al., 2011) (free for non-commercial use), LSUN (Yu et al., 2015) (no license), the ILSVRC2012 split of ImageNet (Deng et al., 2009; Russakovsky et al., 2015) (free for non-commercial use), FGVC-Aircraft (Maji et al., 2013) (free for non-commercial use), Stanford Cars (Krause et al., 2013) (free for non-commercial use), OpenImages v4 (Kuznetsova et al., 2020) (images have a CC BY 2.0 license), Oxford 102 Flower (Nilsback & Zisserman, 2008) (no license) as well as 80 million tiny images (Torralba et al., 2008) (no license given, see also App. H). For the train/test splits we use the standard splits, except on 80M Tiny Images where we treat a random but fixed subset of 1000 images in the first 1,000,000 as our test set. For all datasets that get used as a test out-distribution we use a random but fixed subset of 1000 images.

Binary Training We train the binary discriminator between in-and out-distribution using the loss in Eq. (3) with the bounds over an l_∞-ball of radius $\epsilon = 0.01$ for the out-distribution following (Bitterwolf et al., 2020). For the training out-distribution, we follow previous work and use 80M Tiny Images (Torralba et al., 2008) for CIFAR10 and CIFAR100. There have been concerns over the use of this dataset (Birhane & Prabhu, 2021) because of offensive class labels. We emphasize that we do not use any of the class labels. Since all prior work used this dataset for the sake of comparison we also use it. However, we also perform our experiments using a downscaled version of OpenImages (Kuznetsova et al., 2020) as training out-distribution in App. H and we encourage the community to use those models and values for future comparisons. For R.ImgNet we use the ILSVRC2012 train images that do not belong to R.ImgNet as training out-distribution (NotR.ImgNet).

The architecture that we use for the binary discriminator is relatively shallow (5 linear layers). The architecture is shown in Table 4. Similarly to (Zhang et al., 2020; Bitterwolf et al., 2020), we use long training schedules, running

Table 3. **Separate Training:** Addendum to Table 1 showing the AUCs, GAUCs and AAUCs of ProoD-S on all datasets. The accuracy must always be identical to that of OE and the clean AUCs are also very similar to those of OE. The guarantees are strictly weaker than those provided by the semi-jointly trained ProoD.

In: CIFAR10		CIFAR100			SVHN			LSUN_CR			Smooth		
	Acc	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
OE	95.53	96.1	0.0	6.0	99.2	0.0	0.4	99.5	0.0	15.2	99.0	0.0	11.3
ProoD-S $\Delta=3$	95.53	96.2	28.5	31.8	99.2	33.4	34.7	99.5	32.3	41.0	99.0	31.5	39.7
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
In: CIFAR100		**CIFAR10**			**SVHN**			**LSUN_CR**			**Smooth**		
	Acc	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
OE	77.28	83.9	0.0	0.8	92.8	0.0	0.1	97.4	0.0	4.6	97.6	0.0	0.9
ProoD-S $\Delta=1$	77.28	83.8	17.8	18.0	93.0	26.7	26.8	97.4	22.9	23.8	97.6	22.9	23.1
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
In: R.ImgNet		**Flowers**			**FGVC**			**Cars**			**Smooth**		
	Acc	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
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-S $\Delta=4$	97.10	96.9	50.1	50.7	99.7	59.7	60.6	99.9	57.9	58.9	98.0	40.8	42.3
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

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 (Bitterwolf et al., 2020) which sometimes fails, we found that training of the binary classifier 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. All binary classifiers were trained on single 2080Ti GPUs, managed on a SLURM cluster. Overall, the training of a provable discriminator takes around 16h on CIFAR and 44h on R.ImgNet (wall clock time including evaluations and logging on each epoch).

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 in (Bitterwolf et al., 2020) (11MB instead of 135MB) and thus the the memory overhead for the binary discriminator is less than a third of that of the classifier. As discussed in Section 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 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 4.1) on a subset of the training out-distribution 80M Tiny Images (resp. NotR.ImgNet). As our goal is to have provable guarantees with minimal or no loss on the standard OOD detection task, we choose among all solutions which have better AUC than outlier exposure (OE) (Hendrycks et al., 2019) the one with the highest guaranteed AUC on 80M Tiny Images (on CIFAR10/CIFAR100) resp. 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 (which never happens).

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). The CIFAR experiments were run on single 2080Ti GPUs. This takes about 4h20min in wall clock time. 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. The wall clock time was around 15h for the semi-joint training on R.ImgNet.

Table 4. **Architecture:** The architectures that are used for the binary discriminators. Each convolutional layer is directly followed by a ReLU.

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)

G. Bias Shift

Under the assumption that our binary discriminator g is perfect, that is

$$p(i|x) = \frac{p(x|i)p(i)}{p(x|i)p(i) + p(x|o)p(o)} \tag{9}$$

$$= \frac{1}{1 + \frac{p(x|o)p(o)}{p(x|i)p(i)}} \tag{10}$$

$$= \frac{1}{1 + e^{-g(x)}}, \tag{11}$$

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

$$e^{\tilde{g}(x)} = \frac{p(x|i)\tilde{p}(i)}{p(x|o)\tilde{p}(o)} \tag{12}$$

$$= \frac{p(x|i)p(i)}{p(x|o)p(o)} \frac{p(o)\tilde{p}(i)}{p(i)\tilde{p}(o)} \tag{13}$$

$$= e^{g(x)}e^{\Delta}, \quad \text{with} \quad \Delta = \log\left(\frac{p(o)\tilde{p}(i)}{p(i)\tilde{p}(o)}\right). \tag{14}$$

Note that $\tilde{p}(i) > p(i)$ corresponds to positive shifts. In a practical setting, this parameter can be chosen based on the priors for the 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 4.1 for details. Please 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. This way the classifier would converge to a normal outlier exposure-type classifier without any guarantees.

H. OpenImages as Training Out-Distribution

The 80M Tiny Images dataset has been retracted by the authors due to concerns over offensive class labels (Birhane & Prabhu, 2021). Since all prior work used this dataset, we used the dataset in order to compare ProoD's performance to prior baselines. However, we support the decision of the community to move away from the use of 80M Tiny Images, so we also trained our CIFAR models using a downscaled version of OpenImages v4 (Kuznetsova et al., 2020) as a training out-distribution. We encourage the community to use the results in Table 5 for future comparisons.

I. False Positive Rates

Since in a practical setting a threshold for OOD detection ultimately has to be chosen, it can be interesting to study the false positive rate at a fixed threshold. It is relatively standard to pick the false positive rate at 95% true positive rate (called FPR in Table 6), where low values are desirable. We show the results for all methods and datasets in Table 6. Although ProoD has similarly good performance as OE on this task, it still fails to give non-trivial guarantees. Achieving stronger bounds on the worst-case FPR is an interesting goal for future work.

J. Additional Datasets

In addition to the results shown in Table 1, it is interesting to study how ProoD performs on uniform noise as well as the test set of out-distribution it was trained on. We show the results in Table 7. As in Table 1 the clean performance of ProoD is comparable to that of OE. On CIFAR10, GOOD$_{100}$ achieves almost perfect GAUC against uniform noise, which ProoD unfortunately does not reach.

K. Error Bars

In order to be mindful of our resource consumption we restrict the computation of error bars to our experiments on CIFAR10. We rerun our experiments using the same hyperparameters 5 times. We compute the mean and the standard deviations for our models for all metrics shown in Table 8. The results are shown in Table 8. We see that the fluctuations across different runs are indeed rather small. Furthermore, the clean performance of OE and ProoD show no significant discrepancies.

Table 5. **Training with OpenImages:** We repeat the evaluation from Table 1 for models that were trained using OpenImages v4 as out-distribution instead of 80M Tiny Images. Plain is identical to before and is just repeated for the reader's convenience. Note that the conclusions from the main paper still hold, which indicates that our method is robust to changes in the exact choice of training out-distribution.

In: CIFAR10		CIFAR100			SVHN			LSUN_CR			Smooth		
	Acc	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	95.01	90.0	0.0	0.6	93.8	0.0	0.1	93.1	0.0	0.5	98.2	0.0	0.6
OE	95.42	91.1	0.0	0.5	97.8	0.0	0.2	100.0	0.0	5.2	100.0	0.0	4.1
ProoD-Disc	-	62.9	57.1	57.8	72.6	65.6	66.4	78.1	71.5	72.3	59.5	50.0	50.7
ProoD $\Delta=3$	95.26	90.0	45.2	45.9	97.6	52.4	53.2	100.0	57.4	58.9	99.9	37.6	38.4
In: CIFAR100		CIFAR10			SVHN			LSUN_CR			Smooth		
	Acc	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	77.38	77.7	0.0	0.3	81.9	0.0	0.2	76.4	0.0	0.3	88.8	0.0	0.5
OE	77.86	77.5	0.0	0.3	89.0	0.0	0.1	100.0	0.0	0.2	99.2	0.0	0.2
ProoD-Disc	-	56.1	52.1	52.3	61.0	58.2	58.4	70.4	66.9	67.1	30.3	27.0	27.1
ProoD $\Delta=1$	77.40	75.6	30.7	30.8	86.7	34.9	35.0	100.0	40.0	40.1	99.1	16.1	16.2

Table 6. **False Positive Rates:** For all models we report accuracy on the the test of the in-distribution and the false positive rate at 95% true positive rate (FPR) (smaller is better). We also show the adversarial FPR (AFPR) and the guaranteed FPR (GFPR) for different test out-distributions. The radius of the l_∞-ball for the adversarial manipulations of the OOD data is $\epsilon = 0.01$ for all datasets. The bias shift Δ that was used for ProoD is shown for each in-distribution. ProoD struggles to give non-trivial guarantees for the FPR@95% on most datasets. However, different from GOOD or ProoD-Disc, the clean performance is generally as good as that of OE.

In: CIFAR10		CIFAR100			SVHN			LSUN_CR			Smooth		
	Acc	FPR	GFPR	AFPR	FPR	GFPR	AFPR	FPR	GFPR	AFPR	FPR	GFPR	AFPR
Plain	95.01	56.3	100.0	100.0	40.7	100.0	100.0	46.7	100.0	100.0	9.2	100.0	100.0
OE	95.53	20.7	100.0	99.6	1.7	100.0	100.0	1.3	100.0	99.7	0.0	100.0	96.5
ATOM	95.20	26.5	100.0	88.8	0.8	100.0	94.0	0.3	100.0	66.0	0.0	100.0	99.2
ACET	91.48	39.2	100.0	67.1	29.3	100.0	63.6	5.0	100.0	29.7	0.4	100.0	10.8
GOOD_{80}	90.13	48.9	96.3	69.0	33.5	99.9	65.2	43.0	100.0	59.3	31.8	100.0	50.6
GOOD_{100}	90.14	78.7	85.7	84.8	81.2	87.5	86.9	91.3	96.3	96.3	68.4	85.6	84.4
ProoD-Disc	-	78.5	83.5	83.1	69.2	79.2	78.1	86.3	93.7	92.7	82.4	93.0	91.2
ProoD $\Delta=3$	95.47	22.2	100.0	99.4	1.5	100.0	100.0	1.3	100.0	99.0	0.0	100.0	100.0
In: CIFAR100		CIFAR10			SVHN			LSUN_CR			Smooth		
	Acc	FPR	GFPR	AFPR	FPR	GFPR	AFPR	FPR	GFPR	AFPR	FPR	GFPR	AFPR
Plain	77.38	80.1	100.0	100.0	77.3	100.0	100.0	79.0	100.0	100.0	60.4	100.0	100.0
OE	77.28	73.6	100.0	100.0	40.0	100.0	100.0	14.0	100.0	100.0	12.6	100.0	100.0
ATOM	75.06	88.9	100.0	99.9	37.7	100.0	100.0	8.7	100.0	98.0	0.0	100.0	100.0
ACET	74.43	79.1	100.0	100.0	53.5	100.0	100.0	21.3	100.0	100.0	49.4	100.0	100.0
ProoD-Disc	-	97.9	98.4	98.4	86.7	89.3	89.1	96.0	98.0	97.7	99.2	99.2	99.2
ProoD $\Delta=1$	76.79	77.7	100.0	100.0	36.4	100.0	100.0	15.3	100.0	100.0	1.6	100.0	100.0
In: R.ImgNet		Flowers			FGVC			Cars			Smooth		
	Acc	FPR	GFPR	AFPR	FPR	GFPR	AFPR	FPR	GFPR	AFPR	FPR	GFPR	AFPR
Plain	96.34	55.2	100.0	100.0	48.2	100.0	100.0	75.2	100.0	100.0	0.0	100.0	100.0
OE	97.10	18.2	100.0	100.0	0.2	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0
ProoD-Disc	-	59.2	65.2	65.0	51.0	67.8	66.5	51.7	63.7	62.3	100.0	100.0	100.0
ProoD $\Delta=4$	97.25	18.5	100.0	100.0	0.5	100.0	100.0	0.0	100.0	100.0	0.0	100.0	100.0

Table 7. **Additional Datasets:** We show the AUC, AAUC and GAUC for all models on uniform noise and on the test set of the train out-distribution.

In: CIFAR10		Uniform			Tiny Images		
	Acc	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	95.01	98.0	0.0	82.9	91.7	0.0	0.7
OE	95.53	99.5	0.0	88.0	98.7	0.0	11.1
ATOM	92.33	99.9	0.0	99.9	98.7	0.0	37.4
ACET	91.48	99.9	0.0	99.9	97.1	0.0	92.1
GOOD_{80}	90.13	95.8	95.3	95.5	92.4	56.1	78.4
GOOD_{100}	90.14	99.5	99.0	99.2	81.5	68.1	68.1
ProoD-Disc	-	99.7	99.6	99.6	80.1	75.5	75.7
ProoD $\Delta = 3$	95.47	99.2	80.4	90.1	98.5	52.8	56.0
In: CIFAR100		**Uniform**			**Tiny Images**		
	Acc	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	77.38	82.2	0.0	53.0	81.7	0.0	0.9
OE	77.28	95.8	0.0	64.1	92.2	0.0	4.0
ATOM	71.72	100.0	0.0	100.0	91.9	0.0	11.5
ACET	74.43	99.4	0.0	97.5	91.9	0.0	3.8
ProoD-Disc	-	98.9	98.8	98.8	68.7	65.0	65.4
ProoD $\Delta = 1$	76.79	97.4	57.2	76.1	92.8	32.6	33.0
In: R.ImgNet		**Uniform**			**NotR.ImgNet**		
	Acc	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	96.34	99.3	0.0	74.9	91.7	0.0	0.2
OE	97.10	99.6	0.0	84.6	98.7	0.0	1.2
ProoD-Disc		99.7	99.2	99.3	73.6	69.9	69.9
ProoD $\Delta = 4$	97.25	99.8	79.7	95.2	98.6	50.1	51.3

Table 8. **Error Bars:** We show the mean and standard deviation σ of all metrics for our CIFAR10 models across 5 runs. The tolerances for ProoD's clean performance are very small and yet the differences in clean performance between OE ProoD are not significant.

In: CIFAR10		CIFAR100			SVHN			LSUN_CR			Smooth		
	Acc	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC	AUC	GAUC	AAUC
Plain	94.91	90.0	0.0	0.6	93.9	0.0	0.1	93.4	0.0	0.7	96.7	0.0	1.2
Plain σ	0.16	0.1	0.0	0.1	1.2	0.0	0.0	0.3	0.0	0.2	2.1	0.0	0.5
OE	95.56	96.1	0.0	7.6	99.4	0.0	0.4	99.6	0.0	16.7	99.6	0.0	4.3
OE σ	0.04	0.1	0.0	1.5	0.1	0.0	0.2	0.1	0.0	3.5	0.3	0.0	3.7
ProoD-Disc	-	67.7	61.6	62.2	75.5	68.6	69.3	76.5	70.4	70.9	87.2	77.7	78.8
ProoD-Disc σ	-	0.7	0.7	0.7	1.4	1.7	1.5	1.4	1.7	1.7	3.6	4.3	4.3
ProoD $\Delta = 3$	95.60	96.0	42.2	44.1	99.4	48.6	49.2	99.6	47.1	52.0	99.8	55.2	57.0
ProoD $\Delta = 3$ σ	0.11	0.1	0.8	0.8	0.1	0.6	0.6	0.1	1.5	1.9	0.1	2.9	3.4