ASYMMETRIC CERTIFIED ROBUSTNESS VIA FEATURE-CONVEX NEURAL NETWORKS

Samuel Pfrommer,* Brendon G. Anderson,* Julien Piet, and Somayeh Sojoudi
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA 94720
{sam.pfrommer,bganderson,piet,sojoudi}@berkeley.edu
*Equal contribution

ABSTRACT

Recent works have introduced input-convex neural networks (ICNNs) as learning models with advantageous training, inference, and generalization properties linked to their convex structure. In this paper, we propose a novel *feature-convex neural network* (FCNN) architecture as the composition of an ICNN with a Lipschitz feature map in order to achieve adversarial robustness. We consider the asymmetric binary classification setting with one "sensitive" class, and for this class we prove deterministic, closed-form, and easily-computable certified robust radii for arbitrary ℓ_p-norms. We theoretically justify the use of these models by characterizing their decision region geometry, extending the universal approximation theorem for ICNN regression to the classification setting, and proving a lower bound on the probability that such models perfectly fit even unstructured uniformly distributed data in sufficiently high dimensions. Experiments on Malimg malware classification as well as subsets of MNIST, CIFAR-10, and ImageNet-scale datasets show that FCNNs can attain orders of magnitude larger certified ℓ_1-radii than competing methods while maintaining substantial ℓ_2- and ℓ_∞-radii.

1 INTRODUCTION

Although neural networks achieve state-of-the-art performance across a range of machine learning tasks, researchers have shown that they can be highly sensitive to adversarial inputs that are maliciously designed to fool the model (Biggio et al., 2013; Szegedy et al., 2014; Nguyen et al., 2015). For example, the works Eykholt et al. (2018) and Liu et al. (2019) show that small physical and digital alterations of vehicle traffic signs can cause image classifiers to fail. In safety-critical applications of neural networks, such as autonomous driving (Bojarski et al., 2016; Wu et al., 2017) and medical diagnostics (Amato et al., 2013; Yadav & Jadhav, 2019), this sensitivity to adversarial inputs is clearly unacceptable.

A line of heuristic defenses against adversarial inputs has been proposed, only to be defeated by stronger attack methods (Carlini & Wagner, 2017; Kurakin et al., 2017; Athalye et al., 2018; Uesato et al., 2018; Madry et al., 2018). This has led researchers to develop certifiably robust methods that provide a provable guarantee of safe performance. The strength of such certificates can be highly dependent on network architecture; general off-the-shelf models tend to have large Lipschitz constants, leading to loose Lipschitz-based robustness guarantees (Hein & Andriushchenko, 2017; Fazlyab et al., 2019; Yang et al., 2020b). Consequently, lines of work that impose certificate-amenable structures onto networks have been popularized, e.g., randomized smoothing-based networks (Li et al., 2019; Cohen et al., 2019; Zhai et al., 2020; Yang et al., 2020a; Anderson & Sojoudi, 2022) and ReLU networks that are certified using convex optimization and mixed-integer programming (Wong & Kolter, 2018; Weng et al., 2018; Raghunathan et al., 2018; Anderson et al., 2020; Ma & Sojoudi, 2021). Both of these method families incur serious computational challenges: randomized smoothing typically requires the classification of thousands of randomly perturbed samples per input, while optimization-based solutions scale poorly to large networks.

Despite the moderate success of these certifiable classifiers, conventional assumptions in the literature are unnecessarily restrictive for most practical adversarial settings. Specifically, most works consider a multiclass setting where certificates are desired for inputs of any class. By contrast, many real-world adversarial attacks involve a binary setting with only one *sensitive class* that must be made robust to adversarial perturbations. Consider the representative problem of spam classification; a malicious adversary crafting a spam email will always attempt to fool the classifier toward the "not-spam" class—never conversely (Kuchipudi et al., 2020). Similar logic applies for a range of applications, including malware detection (Grosse et al., 2017), malicious network traffic filtering (Sadeghzadeh et al., 2021), fake news and social media bot detection (Cresci et al., 2021), hate speech removal (Grolman et al., 2022), insurance claims filtering (Finlayson et al., 2019), and financial fraud detection (Cartella et al., 2021).

These applications motivate us to introduce a narrower, asymmetric robustness problem and develop a novel classifier architecture to address this challenge.

1.1 Problem statement and contributions

This work considers the problem of *asymmetric robustness certification*. Specifically, we assume a classification setting wherein one class is "sensitive" and seek to certify that, if some input is classified into this sensitive class, then adversarial perturbations of sufficiently small magnitude cannot change the prediction.

To tackle the asymmetric robustness certification problem and attain state-of-the-art certified radii, we propose *feature-convex neural networks*, and achieve the following contributions in doing so:

1. We provide easily-computable class 1 certified robust radii for feature-convex classifiers with respect to arbitrary ℓ_p-norms.
2. We characterize the decision region geometry of feature-convex classifiers, extend the universal approximation theorem for input-convex ReLU neural networks to the classification setting, and show that, in high dimensions, feature-convex classifiers can perfectly fit even unstructured, uniformly distributed datasets.
3. We evaluate against several baselines on Malimg malware classification (Nataraj et al., 2011), MNIST 3-8 (LeCun, 1998), CIFAR-10 cats-dogs (Krizhevsky et al., 2009), and Kaggle cats-dogs (Kaggle, 2016) and show that our classifiers yield state-of-the-art certified robust radii.

1.2 Related works

Certified adversarial robustness. Two of the most popular approaches for generating robustness certificates are randomized smoothing and convex optimization. Randomized smoothing, popularized by Lecuyer et al. (2019); Li et al. (2019); Cohen et al. (2019), uses the expected prediction of a model when subjected to Gaussian input noise. These works derive ℓ_2-norm balls around inputs on which the smoothed classifier remains constant, but suffer from nondeterminism and high computational burden. Follow-up works generalize randomized smoothing to certify input regions defined by different metrics, e.g., Wasserstein, ℓ_1-, and ℓ_∞-norms (Levine & Feizi, 2020; Teng et al., 2020; Yang et al., 2020a). Other works focus on enlarging the certified regions by optimizing the smoothing distribution (Zhai et al., 2020; Eiras et al., 2021; Anderson et al., 2022), incorporating adversarial training into the base classifier (Salman et al., 2019; Zhang et al., 2020), and employing dimensionality reduction at the input (Pfrommer et al., 2022). Convex optimization-based certificates seek to derive a convex over-approximation of the set of possible outputs when the input is subject to adversarial perturbations, and show that this over-approximation is safe. Various over-approximations have been proposed, e.g., based on linear programming and bounding (Wong & Kolter, 2018; Weng et al., 2018), semidefinite programming (Raghunathan et al., 2018), and partitioned linear and semidefinite programming (Anderson et al., 2020; Ma & Sojoudi, 2021), but it is generally the case that these convex relaxations are either computationally burdensome or too loose when employed on large-scale models. In this paper, we exploit the convex structure of input-convex neural networks to directly derive closed-form robustness certificates for our proposed architecture.

Input-convex neural networks. Input-convex neural networks, popularized by Amos et al. (2017), are a class of parameterized models whose input-output mapping is convex (in at least a subset of the input variables). In Amos et al. (2017), the authors develop tractable methods to learn an input-convex neural network $f\colon \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}$ and show that utilizing it for the convex optimization-based inference $x \mapsto \arg\min_{y\in\mathbb{R}^n} f(x,y)$ yields state-of-the-art results in a variety of domains. Subsequent works propose novel applications of input-convex neural networks in areas such as optimal control and reinforcement learning (Chen et al., 2019; Zeng et al., 2022), optimal transport (Makkuva et al., 2020), and optimal power flow (Chen et al., 2020; Zhang et al., 2021b). Other works have generalized input-convex networks to input-invex networks (Nesterov et al., 2022) and global optimization networks (Zhao et al., 2022) so as to maintain the benign optimization properties of input-convexity. The authors of Siahkamari et al. (2022) present algorithms for efficiently learning convex functions, while Chen et al. (2019); Kim & Kim (2022) derive universal approximation theorems for input-convex neural networks in the convex regression setting. The work Sivaprasad et al. (2021) shows that input-convex neural networks do not suffer from overfitting, and generalize better than multilayer perceptrons on common benchmark datasets. In this work, we incorporate input-convex neural networks as a part of our overall feature-convex architecture, and we leverage convexity properties to derive our novel robustness guarantees.

1.3 NOTATIONS

The natural numbers and real numbers are denoted by $\mathbb{N}$ and $\mathbb{R}$, respectively. The $d \times d$ identity matrix is written as $I_d \in \mathbb{R}^{d\times d}$, and the identity map on $\mathbb{R}^d$ is denoted by $\mathrm{Id}\colon x \mapsto x$. For $A \in \mathbb{R}^{n\times d}$, we define $|A| \in \mathbb{R}^{n\times d}$ by $|A|_{ij} = |A_{ij}|$ for all i, j, and we write $A \geq 0$ if and only if $A_{ij} \geq 0$ for all i, j. The ℓ_p-norm on $\mathbb{R}^d$ is given by $\|\cdot\|_p\colon x \mapsto (|x_1|^p + \cdots + |x_d|^p)^{1/p}$ for $p \in [1,\infty)$ and by $\|\cdot\|_p\colon x \mapsto \max\{|x_1|,\ldots,|x_d|\}$ for $p = \infty$. The dual norm of $\|\cdot\|_p$ is denoted by $\|\cdot\|_{p,*}$. The convex hull of a set $X \subseteq \mathbb{R}^d$ is denoted by $\mathrm{conv}(X)$. The subdifferential of a convex function $g\colon \mathbb{R}^d \to \mathbb{R}$ at $x \in \mathbb{R}^d$ is denoted by $\partial g(x)$. If $\epsilon\colon \Omega \to \mathbb{R}^d$ is a random variable on a probability space $(\Omega, \mathcal{B}, \mathbb{P})$ and P is a predicate defined on $\mathbb{R}^d$, then we write $\mathbb{P}(P(\epsilon))$ to mean $\mathbb{P}(\{\omega \in \Omega : P(\epsilon(\omega))\})$. Lebesgue measure on $\mathbb{R}^d$ is denoted by m. We define $\mathrm{ReLU}\colon \mathbb{R} \to \mathbb{R}$ as $\mathrm{ReLU}(x) = \max\{0, x\}$, and if $x \in \mathbb{R}^d$, $\mathrm{ReLU}(x)$ denotes $(\mathrm{ReLU}(x_1),\ldots,\mathrm{ReLU}(x_d))$. For a function $\varphi\colon \mathbb{R}^d \to \mathbb{R}^q$ and $p \in [1,\infty]$, we define $\mathrm{Lip}_p(\varphi) = \inf\{K \geq 0 : \|\varphi(x) - \varphi(x')\|_p \leq K\|x - x'\|_p \text{ for all } x, x' \in \mathbb{R}^d\}$, and if $\mathrm{Lip}_p(\varphi) < \infty$ we say that φ is Lipschitz continuous with constant $\mathrm{Lip}_p(\varphi)$ (with respect to the ℓ_p-norm).

2 FEATURE-CONVEX CLASSIFIERS

Let $d, q \in \mathbb{N}$ and $p \in [1,\infty]$ be fixed, and consider the task of classifying inputs from a subset of $\mathbb{R}^d$ into a fixed set of classes $\mathcal{Y} \subseteq \mathbb{N}$. In what follows, we restrict to the binary setting where $\mathcal{Y} = \{1, 2\}$ and class 1 is the sensitive class for which we desire robustness certificates (Section 1). In Appendix A, we briefly discuss possible avenues to generalize our framework to multiclass settings using one-versus-all and sequential classification methodologies.

We now formally define the classifiers considered in this work.

Definition 1. Let $f\colon \mathbb{R}^d \to \{1, 2\}$ be defined by

$$f(x) = \begin{cases} 1 & \text{if } g(\varphi(x)) > 0, \\ 2 & \text{if } g(\varphi(x)) \leq 0, \end{cases}$$

for some $\varphi\colon \mathbb{R}^d \to \mathbb{R}^q$ and some $g\colon \mathbb{R}^q \to \mathbb{R}$. Then f is said to be a *feature-convex classifier* if the *feature map* φ is Lipschitz continuous with constant $\mathrm{Lip}_p(\varphi) < \infty$ and g is a convex function.

We denote the class of all feature-convex classifiers by $\mathcal{F}$. Furthermore, for $q = d$, the subclass of all feature-convex classifiers with $\varphi = \mathrm{Id}$ is denoted by $\mathcal{F}_{\mathrm{Id}}$.

As we will see in Section 3.1, defining our classifiers using the composition of a convex classifier with a Lipschitz feature map enables the fast computation of certified regions in the input space. This naturally arises from the global underestimation of convex functions by first-order Taylor approximations. Since sublevel sets of such g are restricted to be convex, the feature map φ is included to increase the representation power and practical performance of our architecture (see Appendix B for

a motivating example). In practice, we find that it suffices to choose φ to be a "simple" map with a small closed-form Lipschitz constant. For example, in our experiments that follow with $q = 2d$, we choose $\varphi(x) = (x - \mu, |x - \mu|)$ with a constant channel-wise dataset mean μ, yielding $\mathrm{Lip}_1(\varphi) \leq 2$, $\mathrm{Lip}_2(\varphi) \leq \sqrt{2}$, and $\mathrm{Lip}_\infty(\varphi) \leq 1$. Although this particular choice of φ is convex, the function g need not be monotone, and therefore the composition $g \circ \varphi$ is nonconvex in general. The prediction and certification of feature-convex classifiers are illustrated in Figure 1.

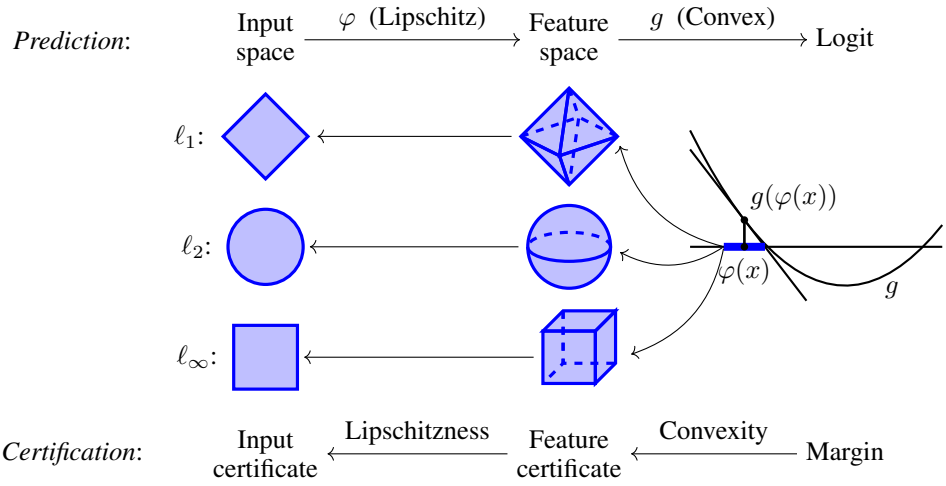


Figure 1: Illustration of feature-convex classifiers and their robustness certification. Since g is convex, it can be globally underapproximated by its tangent plane at $\varphi(x)$, yielding certified sets for all norm balls in the higher-dimensional feature space. Lipschitzness of φ then yields appropriately scaled certificates in the original input space.

In practice, we implement feature-convex classifiers using parameterizations of g, which we now make explicit. Following Amos et al. (2017), we instantiate g as a neural network with nonnegative weight matrices and nondecreasing convex nonlinearities. Specifically, we consider ReLU nonlinearities, which is not restrictive, as our universal approximation result in Theorem 2 proves.

Definition 2. A *feature-convex* ReLU *neural network* is a function $\hat{f} \colon \mathbb{R}^d \to \{1, 2\}$ defined by

$$\hat{f}(x) = \begin{cases} 1 & \text{if } \hat{g}(\varphi(x)) > 0, \\ 2 & \text{if } \hat{g}(\varphi(x)) \leq 0, \end{cases}$$

with $\varphi \colon \mathbb{R}^d \to \mathbb{R}^q$ Lipschitz continuous with constant $\mathrm{Lip}_p(\varphi) < \infty$ and $\hat{g} \colon \mathbb{R}^q \to \mathbb{R}$ defined by

$$\begin{aligned} x^{(1)} &= \mathrm{ReLU}\left(A^{(1)} x^{(0)} + b^{(1)}\right), \\ x^{(l)} &= \mathrm{ReLU}\left(A^{(l)} x^{(l-1)} + b^{(l)} + C^{(l)} x^{(0)}\right), \ l \in \{2, 3, \ldots, L-1\}, \\ \hat{g}(x^{(0)}) &= A^{(L)} x^{(L-1)} + b^{(L)} + C^{(L)} x^{(0)}, \end{aligned}$$

for some $L \in \mathbb{N}$, $L > 1$, and some consistently sized matrices $A^{(l)}, C^{(l)}$ and vectors $b^{(l)}$ satisfying $A^{(l)} \geq 0$ for all $l \in \{2, 3, \ldots, L\}$.

Going forward, we denote the class of all feature-convex ReLU neural networks by $\hat{\mathcal{F}}$. Furthermore, if $q = d$, the subclass of all feature-convex ReLU neural networks with $\varphi = \mathrm{Id}$ is denoted by $\hat{\mathcal{F}}_{\mathrm{Id}}$, which corresponds to the input-convex ReLU neural networks proposed in Amos et al. (2017).

For every $\hat{f} \in \hat{\mathcal{F}}$, it holds that $\hat{g}$ is a convex function due to the rules for composition and nonnegatively weighted sums of convex functions (Boyd & Vandenberghe, 2004, Section 3.2), and therefore $\hat{\mathcal{F}} \subseteq \mathcal{F}$ and $\hat{\mathcal{F}}_{\mathrm{Id}} \subseteq \mathcal{F}_{\mathrm{Id}}$. The "passthrough" weights $C^{(l)}$ were originally included by Amos et al. (2017) to improve the practical performance of the architecture. In some of our more challenging experiments that follow, we remove these passthrough operations and instead add residual identity

mappings between hidden layers, which also preserves convexity. We note that the transformations defined by $A^{(l)}$ and $C^{(l)}$ can be taken to be convolutions, which are nonnegatively weighted linear operations and thus preserve convexity (Amos et al., 2017).

3 CERTIFICATION AND ANALYSIS OF FEATURE-CONVEX CLASSIFIERS

We begin by deriving asymmetric robustness certificates for our feature-convex classifier in Section 3.1. In Section 3.2, we introduce convexly separable sets and theoretically analyze the clean performance of our classifiers through this lens. Namely, we show that there exists a feature-convex classifier with $\varphi = \mathrm{Id}$ that perfectly classifies the CIFAR-10 cats-dogs training dataset, which we show is unsurprising by proving that feature-convex classifiers can perfectly fit high-dimensional uniformly distributed data with high probability. Proofs are deferred to the appendix.

3.1 CERTIFIED ROBUSTNESS GUARANTEES

In this section, we address the asymmetric certified robustness problem by providing class 1 robustness certificates for feature-convex classifiers $f \in \mathcal{F}$. Such robustness corresponds to proving the absence of false negatives in the case that class 1 represents positives and class 2 represents negatives. For example, if in a malware detection setting class 1 represents malware and class 2 represents non-malware, the following certificate gives a lower bound on the magnitude of the malware file alteration needed in order to misclassify the file as non-malware.

Theorem 1. *Let $f \in \mathcal{F}$ be as in Definition 1 and let $x \in f^{-1}(\{1\}) = \{x' \in \mathbb{R}^d : f(x') = 1\}$. If $v(\varphi(x)) \in \mathbb{R}^q$ is a nonzero subgradient of the convex function g at $\varphi(x)$, then $f(x + \delta) = 1$ for all $\delta \in \mathbb{R}^d$ such that*

$$\|\delta\|_p < r(x) \coloneqq \frac{g(\varphi(x))}{\mathrm{Lip}_p(\varphi)\|v(\varphi(x))\|_{p,*}}$$

Remark 1. For $f \in \mathcal{F}$ and $x \in f^{-1}(\{1\})$, a subgradient $v(\varphi(x)) \in \mathbb{R}^q$ of g always exists at $\varphi(x)$, since the subdifferential $\partial g(\varphi(x))$ is a nonempty closed bounded convex set, as g is a finite convex function on all of $\mathbb{R}^q$—see Theorem 23.4 in Rockafellar (1970) and the discussion thereafter. Furthermore, if f is not a constant classifier, such a subgradient $v(\varphi(x))$ must necessarily be nonzero, since, if it were zero, then $g(y) \geq g(\varphi(x)) + v(\varphi(x))^\top (y - \varphi(x)) = g(\varphi(x)) > 0$ for all $y \in \mathbb{R}^q$, implying that f identically predicts class 1, which is a contradiction. Thus, the certified radius given in Theorem 1 is always well-defined in practical settings.

Theorem 1 is derived from the fact that a convex function is globally underapproximated by any tangent plane. The nonconstant terms in Theorem 1 afford an intuitive interpretation: the radius scales proportionally to the confidence $g(\varphi(x))$ and inversely with the input sensitivity $\|v(\varphi(x))\|_{p,*}$. In practice, the subgradient $v(\varphi(x))$ is easily evaluated as the Jacobian of g at $\varphi(x)$ using standard automatic differentiation packages. This provides fast, deterministic class 1 certificates for any ℓ_p-norm without modification of the feature-convex network's training procedure or architecture.

3.2 REPRESENTATION POWER CHARACTERIZATION

We restrict our analysis to the class $\mathcal{F}_{\mathrm{Id}}$ of feature-convex classifiers with an identity feature map. This can be equivalently considered as the class of classifiers for which the input-to-logit map g is convex. We therefore refer to models in $\mathcal{F}_{\mathrm{Id}}$ as *input-convex classifiers*. While the feature map φ is useful in boosting the practical performance of our classifiers, the theoretical results in this section suggest that there is significant potential in using input-convex classifiers as a standalone solution.

Classifying convexly separable sets. We begin by introducing the notion of convexly separable sets, which are intimately related to decision regions representable by the class $\mathcal{F}_{\mathrm{Id}}$.

Definition 3. Let $X_1, X_2 \subseteq \mathbb{R}^d$. The ordered pair (X_1, X_2) is said to be *convexly separable* if there exists a nonempty closed convex set $X \subseteq \mathbb{R}^d$ such that $X_2 \subseteq X$ and $X_1 \subseteq \mathbb{R}^d \setminus X$.

Notice that it may be the case that a pair (X_1, X_2) is convexly separable yet the pair (X_2, X_1) is not. Although low-dimensional intuition may cause concerns regarding the convex separability of sets of binary-labeled data, we will soon see in Theorem 4 that, even for relatively unstructured

data distributions, binary datasets are actually convexly separable in high dimensions with high probability. We now show that convexly separable datasets possess the property that they may always be perfectly fit by input-convex classifiers.

Proposition 1. *For any nonempty closed convex set $X \subseteq \mathbb{R}^d$, there exists $f \in \mathcal{F}_{\mathrm{Id}}$ such that $X = f^{-1}(\{2\}) = \{x \in \mathbb{R}^d : f(x) = 2\}$. In particular, this shows that if (X_1, X_2) is a convexly separable pair of subsets of $\mathbb{R}^d$, then there exists $f \in \mathcal{F}_{\mathrm{Id}}$ such that $f(x) = 1$ for all $x \in X_1$ and $f(x) = 2$ for all $x \in X_2$.*

We also show that the converse of Proposition 1 holds: the geometry of the decision regions of classifiers in $\mathcal{F}_{\mathrm{Id}}$ can be characterized as consisting of a convex set and its complement.

Proposition 2. *Let $f \in \mathcal{F}_{\mathrm{Id}}$. The decision region under f associated to class 2, namely $X := f^{-1}(\{2\}) = \{x \in \mathbb{R}^d : f(x) = 2\}$, is a closed convex set.*

Note that this is not necessarily true for our more general feature-convex architectures with $\varphi \neq \mathrm{Id}$. We continue our theoretical analysis of input-convex classifiers by extending the universal approximation theorem for regressing upon real-valued convex functions (given in Chen et al. (2019)) to the classification setting. In particular, Theorem 2 below shows that any input-convex classifier $f \in \mathcal{F}_{\mathrm{Id}}$ can be approximated arbitrarily well on any compact set by ReLU neural networks with nonnegative weights. Here, "arbitrarily well" means that the set of inputs where the neural network prediction differs from that of f can be made to have arbitrarily small Lebesgue measure.

Theorem 2. *For any $f \in \mathcal{F}_{\mathrm{Id}}$, any compact convex subsets X of $\mathbb{R}^d$, and any $\epsilon > 0$, there exists $\hat{f} \in \hat{\mathcal{F}}_{\mathrm{Id}}$ such that $m(\{x \in X : \hat{f}(x) \neq f(x)\}) < \epsilon$.*

An extension of the proof of Theorem 2 combined with Proposition 1 yields that input-convex ReLU neural networks can perfectly fit convexly separable pairs of sampled data.

Theorem 3. *If (X_1, X_2) is a convexly separable pair of finite subsets of $\mathbb{R}^d$, then there exists $\hat{f} \in \hat{\mathcal{F}}_{\mathrm{Id}}$ such that $\hat{f}(x) = 1$ for all $x \in X_1$ and $\hat{f}(x) = 2$ for all $x \in X_2$.*

Theorems 2 and 3 theoretically justify the particular parameterization in Definition 2 for learning feature-convex classifiers to fit convexly separable data.

Empirical convex separability. Interestingly, we find empirically that high-dimensional image training data is convexly separable. We illustrate this in Appendix D by attempting to reconstruct a CIFAR-10 cat image from a convex combination of the dogs and vice versa; the error is always significantly positive and image reconstruction is visually poor. This observation, combined with Theorem 3, immediately yields the following result.

Corollary 1. *There exists $\hat{f} \in \hat{\mathcal{F}}_{\mathrm{Id}}$ such that $\hat{f}$ achieves perfect training accuracy for the unaugmented CIFAR-10 cats versus dogs dataset.*

The gap between this theoretical guarantee and our practical performance is large; without the feature map, our CIFAR-10 cats-dogs classifier achieves just 73.4% training accuracy (Table 2). While high training accuracy may not necessarily imply strong test set performance, Corollary 1 demonstrates that the typical deep learning paradigm of overfitting to the training dataset is attainable and that there is at least substantial room for improvement in the design and optimization of input-convex classifiers (Nakkiran et al., 2021). We leave the challenge of overfitting to the CIFAR-10 cats-dogs training data with an input-convex classifier as an open research problem for the field.

Convex separability in high dimensions. We conclude by investigating *why* the convex separability property that allows for Corollary 1 might hold for natural image datasets. We argue that dimensionality facilitates this phenomenon by showing that data is easily separated by some $f \in \hat{\mathcal{F}}_{\mathrm{Id}}$ when d is sufficiently large. In particular, although it may seem restrictive to rely on models in $\hat{\mathcal{F}}_{\mathrm{Id}}$ with convex class 2 decision regions, we show in Theorem 4 below that even uninformative data distributions that are seemingly difficult to classify may be fit by such models with high probability as the dimensionality of the data increases.

Theorem 4. *Consider $M, N \in \mathbb{N}$. Let $X_1 = \{x^{(1)}, \ldots, x^{(M)}\} \subseteq \mathbb{R}^d$ and $X_2 = \{y^{(1)}, \ldots, y^{(N)}\} \subseteq \mathbb{R}^d$ be samples with all elements $x_k^{(i)}, y_l^{(j)}$ drawn independently and identically*

from the uniform probability distribution on $[-1, 1]$. *Then, it holds that*

$$\mathbb{P}\left((X_1, X_2) \text{ is convexly separable}\right) \geq 1 - \left(1 - \frac{1}{2^{MN}}\right)^d. \tag{1}$$

In particular, the probability that $\hat{\mathcal{F}}_{\mathrm{Id}}$ *contains an input-convex* ReLU *neural network that classifies all* $x^{(i)}$ *into class* 1 *and all* $y^{(j)}$ *into class* 2 *converges linearly to* 1 *as* $d \to \infty$.

Although the uniformly distributed data in Theorem 4 is unrealistic in practice, the result demonstrates that the class $\hat{\mathcal{F}}_{\mathrm{Id}}$ of input-convex ReLU neural networks has sufficient complexity to fit even the most unstructured data in high dimensions. Despite this ability, researchers have found that current input-convex neural networks tend to not overfit in practice, yielding small generalization gaps relative to conventional neural networks (Sivaprasad et al., 2021). Achieving the typical modern deep learning paradigm of overfitting to the training dataset with input-convex networks is an exciting open challenge (Nakkiran et al., 2021).

4 Experiments

We first describe our baseline methods, feature-convex architecture, and class accuracy balancing procedure. Our results are then reported across a variety of datasets, ranging from simple MNIST 3-8 and malware classification to more challenging CIFAR-10 and ImageNet-scale cats-dogs classification. Further experimental setup details are deferred to Appendix E.

Baseline methods. We consider several state-of-the-art randomized and deterministic baselines. For all datasets, we evaluate the randomized smoothing certificates of Yang et al. (2020a) for the Gaussian, Laplacian, and uniform distributions, trained with noise augmentation. We also evaluate, when applicable, deterministic certified methods for each norm ball. These include the splitting-noise ℓ_1-certificates from Levine & Feizi (2021), the orthogonality-based ℓ_2-certificates from Trockman & Kolter (2021), and the ℓ_∞-distance-based ℓ_∞-certificates from Zhang et al. (2021a). We note that the last two deterministic methods are not evaluated on the large-scale Malimg and Kaggle datasets due to their prohibitive runtime; furthermore, the ℓ_∞-distance net was unable to significantly surpass the performance of a random classifier on the CIFAR-10 cats-dogs dataset and therefore is only included in the MNIST 3-8 experiment.

Feature-convex architecture. Our simple experiments (MNIST 3-8 and Malimg) require no feature map ($\varphi = \mathrm{Id}$); for both cats-dogs classification tasks, we let our feature map be the concatenation $\varphi(x) = (x - \mu, |x - \mu|)$, where μ is the channel-wise dataset mean (e.g., size 3 for an RGB image) broadcasted to the appropriate dimensions. Our MNIST architecture then consists of a simple two hidden layer input-convex multilayer perceptron with $(n_1, n_2) = (200, 50)$ hidden features, ReLU nonlinearities, and passthrough weights. For all other datasets, we use various instantiations of a convex ConvNet where successive layers have a constant number of channels and image size. This allows for the addition of identity residual connections to each convolution and lets us remove the passthrough connections altogether. Convexity is enforced by projecting relevant weights onto the nonnegative orthant after each epoch and similarly constraining BatchNorm γ parameters to be positive. We initialize positive weight matrices to be drawn uniformly from the interval $[0, \epsilon]$, where $\epsilon = 0.003$ for linear weights and $\epsilon = 0.005$ for convolutional weights. Jacobian regularization is also used to improve our certified radii (Hoffman et al., 2019).

Class accuracy balancing. Since we consider *asymmetric* certified robustness, care must be taken to ensure a fair comparison of class 1 certificates. Indeed, a constant classifier that always outputs class 1 would achieve perfect class 1 accuracy and infinite class 1 certified radii—yet it would not be a particularly interesting classifier as its accuracy on class 2 inputs would be poor. We therefore post-process the decision threshold of each classifier such that the clean class 1 and class 2 accuracies are equivalent, allowing for a direct comparison of the certification performance for class 1.

4.1 Datasets

We now introduce the various datasets considered in this work. The first two (MNIST 3-8 and Malimg) are relatively simple classification problems where near-perfect classification accuracy is attainable; the Malimg dataset falls in this category despite containing relatively large images. We

then discuss two cats-versus-dogs classification problems at different image scales. Data augmentation details are deferred to Appendix E.4.

MNIST 3-8. For our MNIST binary classification problem, we choose the problem of distinguishing between 3 and 8 (LeCun, 1998). These were selected as 3 and 8 are generally more visually similar and challenging to distinguish than other digit pairs. Images are 28×28 pixels and grayscale.

Malimg. Our malware classification experiments use grayscale, bytewise encodings of raw malware binaries Nataraj et al. (2011). Each image pixel corresponds to one byte of data, in the range of 0–255, and successive bytes are added horizontally from left to right on the image until wrapping at some predetermined width. We use the extracted malware images from the seminal dataset Nataraj et al. (2011), padding and cropping images to be 512×512. Note that licensing concerns generally prevent the distribution of "clean" executable binaries; as this work is focused on providing a general approach to robust classification, in the spirit of reproducibility we instead report classification results between different kinds of malware. Namely, we distinguish between malware from the most numerous "Allaple.A" class (2949 samples) and an identically-sized random subset of all other 24 malware classes.

CIFAR-10 cats-dogs. We take as our two CIFAR-10 classes the cat and dog classes as they are relatively difficult to distinguish (Giuste & Vizcarra, 2020; Liu & Mukhopadhyay, 2018; Ho-Phuoc, 2018). Other classes (e.g., ships) are typically easier to classify since large background features (e.g., blue water) are strongly correlated with the target label. Samples are 32×32 RGB images.

Kaggle cats-dogs. To enable comparisons with the CIFAR-10 cats-dogs dataset, we selected an ImageNet-scale cats-dogs dataset available on Kaggle (Kaggle, 2016). This dataset contains $25,000$ images of cats and dogs which we crop to 224×224 pixels. While ImageNet also contains cat and dog images, these classes are unbalanced and subdivided by species. ImageNet's extraneous classes also add significant unnecessary size.

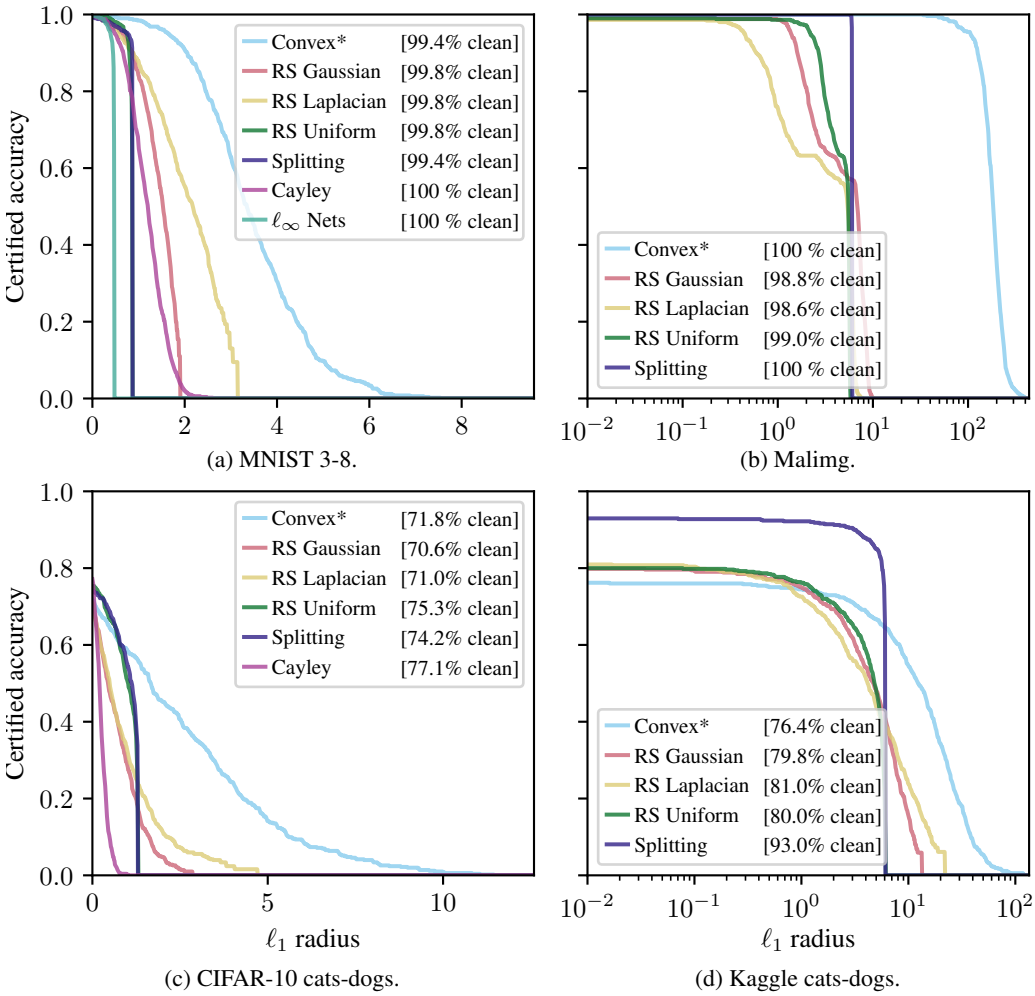
4.2 Discussion

Experimental results for ℓ_1-norm balls are reported in Figure 2, where our feature-convex classifier radii outperform all other baselines across all datasets. Due to space constraints, we defer the corresponding plots for ℓ_2- and ℓ_∞-norm balls to Appendix F.

For the MNIST 3-8 and Malimg datasets (Figures 2a and 2b), all methods achieve high clean test accuracy, even with $\varphi = \mathrm{Id}$ for the feature-convex classifiers. Our method scales exceptionally well with the dimensionality of the input, with orders of magnitude improvement over state-of-the-art certificates for the Malimg dataset. The Malimg certificates in particular have an interesting concrete interpretation. As each pixel corresponds to one byte in the original malware file, an ℓ_1-certificate of radius r provides a robustness certificate for up to r bytes in the file. Namely, even if a malware designer were to arbitrarily change r malware bytes, they would be unable to fool our classifier into returning a false negative. This may not have an immediate practical impact as small semantic changes (e.g., reordering unrelated instructions) could induce large ℓ_p-norm shifts. However, as randomized smoothing was extended from pixel-space to semantic transformations (Li et al., 2021), we expect that similar extensions can produce practical certifiably robust malware classifiers.

The two cats-dogs classification experiments provide an interesting study on the impact of dimensionality for similar tasks (Figures 2c and 2d). As with the simple experiments, the benefits of our method are especially pronounced for the larger 224×224 Kaggle images. We also note that the clean accuracy of our method is 4.6% improved in absolute terms for the higher-dimension images. This matches our theoretical result in Theorem 4, which finds that convex separability is easier to achieve in high dimensions. Nevertheless, Corollary 1 and the discussion in Appendix G.3 suggest that significant performance gains are still attainable for the CIFAR-10 experiment.

While our method produces nontrivial robustness certificates for ℓ_2- and ℓ_∞-norms (Appendix F), it offers the largest improvement for ℓ_1-certificates in high dimensions. This is likely due to the characteristics of the subgradient dual norm factor in the denominator of Theorem 1. The dual of the ℓ_1-norm is the ℓ_∞-norm, which selects the largest magnitude element in the gradient of the output logit with respect to the input pixels. As the input image scales, it is natural for the classifier to become less dependent on any one specific pixel, shrinking the denominator in Theorem 1. Conversely, when certifying for the ℓ_∞-norm, one must evaluate the ℓ_1-norm of the gradient, which scales pro-



(a) MNIST 3-8. (b) Malimg.

(c) CIFAR-10 cats-dogs. (d) Kaggle cats-dogs.

Figure 2: Class 1 certified radii curves for the ℓ_1-norm. Note the log-scale on the rightmost plots.

portionally to the input size. Nevertheless, we find in Appendix F that our ℓ_2- and ℓ_∞-radii are generally on the same order as those of the baselines while maintaining speed and deterministism.

Unlike randomized smoothing-based methods, our method's certificates are almost immediate, requiring just one forwards pass and one backwards pass through the network. For our largest inputs from the 512×512 Malimg dataset, the certification procedure requires fewer than 10 milliseconds per sample on our hardware. This is substantially faster than the runtime for randomized smoothing, which scales from several seconds per CIFAR-10 image to minutes for an ImageNet image (Cohen et al., 2019). Our method has the added benefit of being completely deterministic in both certification and prediction.

Ablation tests examining the impact of Jacobian regularization, the feature map φ, and data augmentation are included in Appendix G.

5 Conclusion

This work introduces the problem of asymmetric certified robustness, which we show naturally applies to a number of practical adversarial settings. We define feature-convex classifiers in this setting and theoretically characterize their representation power from geometric, approximation theoretic, and statistical lenses. Closed-form certified robust radii for the sensitive class are proven for arbitrary ℓ_p-norms, and we find that our ℓ_1 robustness certificates in particular substantially outperform

those of prior state-of-the-art methods. We also show theoretically that significant performance improvements should be realizable for natural image datasets such as CIFAR-10 cats-versus-dogs. Possible directions for future research include bridging the gap between the theoretical power of feature-convex models and their numerical implementation, exploring more sophisticated choices of the feature map φ, and developing multiclass extensions.

REFERENCES

Filippo Amato, Alberto López, Eladia María Peña-Méndez, Petr Vaňhara, Aleš Hampl, and Josef Havel. Artificial neural networks in medical diagnosis, 2013.

Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *International Conference on Machine Learning*, pp. 146–155. PMLR, 2017.

Brendon G. Anderson and Somayeh Sojoudi. Certified robustness via locally biased randomized smoothing. In *Learning for Dynamics and Control*. PMLR, 2022.

Brendon G. Anderson, Ziye Ma, Jingqi Li, and Somayeh Sojoudi. Tightened convex relaxations for neural network robustness certification. In *Proceedings of the 59th IEEE Conference on Decision and Control*, 2020.

Brendon G. Anderson, Samuel Pfrommer, and Somayeh Sojoudi. Towards optimal randomized smoothing: A semi-infinite linear programming approach. In *ICML Workshop on Formal Verification of Machine Learning*, 2022.

MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.*, 2019. URL `http://docs.mosek.com/9.0/toolbox/index.html`.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pp. 274–283. PMLR, 2018.

Dimitri Bertsekas. *Nonlinear Programming*. Athena Scientific, 3rd edition, 2016.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 387–402. Springer, 2013.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pp. 3–14, 2017.

Francesco Cartella, Orlando Anunciacao, Yuki Funabiki, Daisuke Yamaguchi, Toru Akishita, and Olivier Elshocht. Adversarial attacks for tabular data: Application to fraud detection and imbalanced data. *arXiv preprint arXiv:2101.08030*, 2021.

Yize Chen, Yuanyuan Shi, and Baosen Zhang. Optimal control via neural networks: A convex approach. In *International Conference on Learning Representations*, 2019.

Yize Chen, Yuanyuan Shi, and Baosen Zhang. Data-driven optimal voltage regulation using input convex neural networks. *Electric Power Systems Research*, 189:106741, 2020.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pp. 1310–1320. PMLR, 2019.

Stefano Cresci, Marinella Petrocchi, Angelo Spognardi, and Stefano Tognazzi. Adversarial machine learning for protecting against online manipulation. *IEEE Internet Computing*, 26(2):47–52, 2021.

Francisco Eiras, Motasem Alfarra, M Pawan Kumar, Philip HS Torr, Puneet K Dokania, Bernard Ghanem, and Adel Bibi. ANCER: Anisotropic certification via sample-wise volume maximization. *arXiv preprint arXiv:2107.04570*, 2021.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1625–1634, 2018.

Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Samuel G Finlayson, John D Bowers, Joichi Ito, Jonathan L Zittrain, Andrew L Beam, and Isaac S Kohane. Adversarial attacks on medical machine learning. *Science*, 363(6433):1287–1289, 2019.

Felipe O Giuste and Juan C Vizcarra. Cifar-10 image classification using feature ensembles. *arXiv preprint arXiv:2002.03846*, 2020.

Edita Grolman, Hodaya Binyamini, Asaf Shabtai, Yuval Elovici, Ikuya Morikawa, and Toshiya Shimizu. Hateversarial: Adversarial attack against hate speech detection algorithms on twitter. In *Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*, pp. 143–152, 2022.

Kathrin Grosse, Nicolas Papernot, Praveen Manoharan, Michael Backes, and Patrick McDaniel. Adversarial examples for malware detection. In *European symposium on research in computer security*, pp. 62–79. Springer, 2017.

Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. *Advances in neural information processing systems*, 30, 2017.

Tien Ho-Phuoc. Cifar10 to compare visual recognition performance between deep neural networks and humans. *arXiv preprint arXiv:1811.07270*, 2018.

Judy Hoffman, Daniel A Roberts, and Sho Yaida. Robust learning with jacobian regularization. *arXiv preprint arXiv:1908.02729*, 2019.

Kaggle. Dogs vs. cats redux: Kernels edition. 2016. URL `https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition`.

Jinrae Kim and Youdan Kim. Parameterized convex universal approximators for decision-making problems. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Bhargav Kuchipudi, Ravi Teja Nannapaneni, and Qi Liao. Adversarial machine learning for spam filters. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, pp. 1–6, 2020.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2017.

Yann LeCun. The MNIST database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *IEEE Symposium on Security and Privacy*, pp. 656–672. IEEE, 2019.

Alexander Levine and Soheil Feizi. Wasserstein smoothing: Certified robustness against Wasserstein adversarial attacks. In *International Conference on Artificial Intelligence and Statistics*, pp. 3938–3947. PMLR, 2020.

Alexander J Levine and Soheil Feizi. Improved, deterministic smoothing for l_1 certified robustness. In *International Conference on Machine Learning*, pp. 6254–6264. PMLR, 2021.

Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Linyi Li, Maurice Weber, Xiaojun Xu, Luka Rimanic, Bhavya Kailkhura, Tao Xie, Ce Zhang, and Bo Li. Tss: Transformation-specific smoothing for robustness certification. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 535–557, 2021.

Aishan Liu, Xianglong Liu, Jiaxin Fan, Yuqing Ma, Anlan Zhang, Huiyuan Xie, and Dacheng Tao. Perceptual-sensitive gan for generating adversarial patches. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1028–1035, 2019.

Qun Liu and Supratik Mukhopadhyay. Unsupervised learning using pretrained cnn and associative memory bank. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 01–08. IEEE, 2018.

Ziye Ma and Somayeh Sojoudi. A sequential framework towards an exact SDP verification of neural networks. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–8. IEEE, 2021.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Ashok Makkuva, Amirhossein Taghvaei, Sewoong Oh, and Jason Lee. Optimal transport mapping via input convex neural networks. In *International Conference on Machine Learning*, pp. 6672–6681. PMLR, 2020.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

Lakshmanan Nataraj, Sreejith Karthikeyan, Gregoire Jacob, and Bangalore S Manjunath. Malware images: visualization and automatic classification. In *Proceedings of the 8th international symposium on visualization for cyber security*, pp. 1–7, 2011.

Vitali Nesterov, Fabricio Arend Torres, Monika Nagy-Huber, Maxim Samarin, and Volker Roth. Learning invariances with generalised input-convex neural networks. *arXiv preprint arXiv:2204.07009*, 2022.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 427–436, 2015.

Samuel Pfrommer, Brendon G. Anderson, and Somayeh Sojoudi. Projected randomized smoothing for certified adversarial robustness. *Preprint*, 2022. URL `https://brendon-anderson.github.io/files/publications/pfrommer2022projected.pdf`.

Aditi Raghunathan, Jacob Steinhardt, and Percy S Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In *Advances in Neural Information Processing Systems*, pp. 10877–10887, 2018.

R Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

Amir Mahdi Sadeghzadeh, Saeed Shiravi, and Rasool Jalili. Adversarial network traffic: Towards evaluating the robustness of deep-learning-based network traffic classification. *IEEE Transactions on Network and Service Management*, 18(2):1962–1976, 2021.

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Ali Siahkamari, Durmus Alp Emre Acar, Christopher Liao, Kelly L Geyer, Venkatesh Saligrama, and Brian Kulis. Faster algorithms for learning convex functions. In *International Conference on Machine Learning*, pp. 20176–20194. PMLR, 2022.

Sarath Sivaprasad, Ankur Singh, Naresh Manwani, and Vineet Gandhi. The curious case of convex neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 738–754. Springer, 2021.

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

Jiaye Teng, Guang-He Lee, and Yang Yuan. ℓ_1 adversarial robustness certificates: A randomized smoothing approach. *Preprint*, 2020. URL `https://openreview.net/forum?id=H1lQIgrFDS`.

Asher Trockman and J Zico Kolter. Orthogonalizing convolutional layers with the cayley transform. *arXiv preprint arXiv:2104.07167*, 2021.

Jonathan Uesato, Brendan O'Donoghue, Pushmeet Kohli, and Aaron Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, pp. 5025–5034. PMLR, 2018.

Lily Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon. Towards fast computation of certified robustness for ReLU networks. In *International Conference on Machine Learning*, pp. 5276–5285. PMLR, 2018.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pp. 5286–5295. PMLR, 2018.

Bichen Wu, Forrest Iandola, Peter H. Jin, and Kurt Keutzer. SqueezeDet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 129–137, 2017.

Samir S Yadav and Shivajirao M Jadhav. Deep convolutional neural network based medical image classification for disease diagnosis. *Journal of Big Data*, 6(1):1–18, 2019.

Greg Yang, Tony Duan, J Edward Hu, Hadi Salman, Ilya Razenshteyn, and Jerry Li. Randomized smoothing of all shapes and sizes. In *International Conference on Machine Learning*, pp. 10693–10705. PMLR, 2020a.

Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Russ R Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. *Advances in neural information processing systems*, 33:8588–8601, 2020b.

Fancheng Zeng, Guanqiu Qi, Zhiqin Zhu, Jian Sun, Gang Hu, and Matthew Haner. Convex neural networks based reinforcement learning for load frequency control under denial of service attacks. *Algorithms*, 15(2):34, 2022.

Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. MACER: Attack-free and scalable robust training via maximizing certified radius. In *International Conference on Learning Representations*, 2020.

Bohang Zhang, Du Jiang, Di He, and Liwei Wang. Boosting the certified robustness of l-infinity distance nets. *arXiv preprint arXiv:2110.06850*, 2021a.

Dinghuai Zhang, Mao Ye, Chengyue Gong, Zhanxing Zhu, and Qiang Liu. Black-box certification with randomized smoothing: A functional optimization based framework. In *Advances in Neural Information Processing Systems*, volume 33, pp. 2316–2326, 2020.

Ling Zhang, Yize Chen, and Baosen Zhang. A convex neural network solver for dcopf with generalization guarantees. *IEEE Transactions on Control of Network Systems*, 2021b.

Sen Zhao, Erez Louidor, and Maya Gupta. Global optimization networks. In *International Conference on Machine Learning*, pp. 26927–26957. PMLR, 2022.

A CLASSIFICATION FRAMEWORK GENERALIZATION

While outside the scope of our work, we note that there are two natural ways to extend our approach to a multiclass setting with one sensitive class. Let $\mathcal{Y} = \{1, 2, \ldots, c\}$, with class 1 being the sensitive class for which we aim to generate certificates.

One approach involves a two-step architecture, where a feature-convex classifier first distinguishes between the sensitive class 1 and all other classes $\{2, \ldots, c\}$ and an arbitrary second classifier distinguishes between the classes $\{2, \ldots, c\}$. The first classifier could then be used to generate class 1 certificates, as described in Section 3.1.

Alternatively, we could define g to map directly to c output logits, with the first logit convex in the input and the other logits concave in the input. Concavity can be easily achieved by negating the output of a convex network. Let the ith output logit then be denoted as g_i and consider an input where the classifier predicts class 1 (i.e., $g_1(\varphi(x)) \geq g_i(\varphi(x))$ for all $i \geq 2$); since the difference of a convex and a concave function is convex, we can generate a certificate for the nonnegativity of each convex decision function $g_1(\varphi(x)) - g_i(\varphi(x))$. Minimizing these certificates over all $i \geq 2$ yields a robustness certificate for the sensitive class.

Note that g mapping to 2 or more logits, all convex in the input, would not yield any tractable certificates. This is because the classifier decision function would now be the difference of two convex functions and have neither convex nor concave structure. We therefore choose to instantiate our binary classification networks with a single convex output logit for clarity.

B FEATURE MAP MOTIVATION

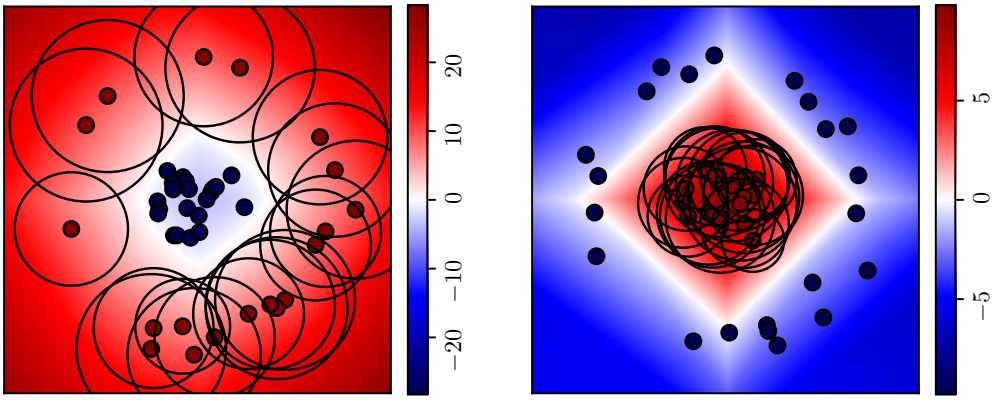


Figure 3: Experiments demonstrating the role of the feature map $\varphi = (x, |x|)$ in $\mathbb{R}^2$, with the output logit shaded. Certified radii from our method are shown as black rings. (a) Certifying the outer class (red points). This is possible using an input-convex classifier as a convex sublevel set contains the inner class (blue points). (b) Certifying the inner class (red points). This would not be possible with $\varphi = \mathrm{Id}$ as there is no convex set containing the outer class (blue points) but excluding the inner. The feature map φ enables this by permitting convex separability in the higher dimensional space. Note that though the shaded output logit is not convex in the input we still generate certificates.

This section examines the importance of the feature map φ with a low-dimensional example. Consider the binary classification setting where one class $X_2 \subseteq \mathbb{R}^d$ is clustered around the origin and the other class is $X_1 \subseteq \mathbb{R}^d$ surrounds it in a ring. Here, the pair (X_1, X_2) is convexly separable as a circular decision boundary surrounding X_2 would be convex (Figure 3a). Note that the reverse pair (X_2, X_1) is *not* convexly separable, as there does not exist a convex set containing X_1 but excluding X_2. A standard input-convex classifier with $\varphi = \mathrm{Id}$ would therefore be unable to discriminate between the classes in this direction (Proposition 2); i.e., we would be able to learn a classifier than generates certificates for points in X_1, but not X_2.

This problem is addressed with by choosing the feature map to be the simple concatenation $\varphi(x) = (x, |x|)$ mapping from $\mathbb{R}^d$ to $\mathbb{R}^q = \mathbb{R}^{2d}$, with associated Lipschitz constants $\mathrm{Lip}_1(\varphi) \leq 2$, $\mathrm{Lip}_2(\varphi) \leq \sqrt{2}$, and $\mathrm{Lip}_\infty(\varphi) \leq 1$. In this augmented feature space, X_1 and X_2 are convexly separable in both directions, as they are each contained in a convex set (specifically, a half-plane) whose complement contains the other class. We are now able to learn a classifier which takes X_2 as the sensitive class for which certificates are required (Figure 3b). This parallels the motivation of the support vector machine "kernel trick," where inputs are augmented to a higher-dimensional space wherein the data is linearly separable (instead of convexly separable as in our case).

C Proofs for Section 3 (Certification and analysis of feature-convex classifiers)

Theorem 1. *Let $f \in \mathcal{F}$ be as in Definition 1 and let $x \in f^{-1}(\{1\}) = \{x' \in \mathbb{R}^d : f(x') = 1\}$. If $v(\varphi(x)) \in \mathbb{R}^q$ is a nonzero subgradient of the convex function g at $\varphi(x)$, then $f(x + \delta) = 1$ for all $\delta \in \mathbb{R}^d$ such that*

$$\|\delta\|_p < r(x) \coloneqq \frac{g(\varphi(x))}{\mathrm{Lip}_p(\varphi)\|v(\varphi(x))\|_{p,*}}$$

Proof. Suppose that $v(\varphi(x)) \in \mathbb{R}^q$ is a nonzero subgradient of g at $\varphi(x)$, so that $g(y) \geq g(\varphi(x)) + v(\varphi(x))^\top (y - \varphi(x))$ for all $y \in \mathbb{R}^q$. Let $\delta \in \mathbb{R}^d$ be such that $\|\delta\|_p < r(x)$. Then it holds that

$$\begin{aligned} g(\varphi(x + \delta)) &\geq g(\varphi(x)) + v(\varphi(x))^\top (\varphi(x + \delta) - \varphi(x)) \\ &\geq g(\varphi(x)) - \|v(\varphi(x))\|_{p,*} \|\varphi(x + \delta) - \varphi(x)\|_p \\ &\geq g(\varphi(x)) - \|v(\varphi(x))\|_{p,*} \mathrm{Lip}_p(\varphi) \|\delta\|_p \\ &> 0, \end{aligned}$$

so indeed $f(x + \delta) = 1$. □

Lemma 1. *For any nonempty closed convex set $X \subseteq \mathbb{R}^d$, there exists a convex function $g \colon \mathbb{R}^d \to \mathbb{R}$ such that $X = g^{-1}((-\infty, 0]) = \{x \in \mathbb{R}^d : g(x) \leq 0\}$.*

Proof. Let $X \subseteq \mathbb{R}^d$ be a nonempty closed convex set. We take the distance function $g = d_X$ defined by $d_X(x) = \inf_{y \in X} \|y - x\|_2$. Since X is closed and $y \mapsto \|y - x\|_2$ is coercive for all $x \in \mathbb{R}^d$, it holds that $y \mapsto \|y - x\|_2$ attains its infimum over X (Bertsekas, 2016, Proposition A.8). Let $x^{(1)}, x^{(2)} \in \mathbb{R}^d$ and let $\theta \in [0, 1]$. Then there exist $y^{(1)}, y^{(2)} \in X$ such that $g(x^{(1)}) = \|y^{(1)} - x^{(1)}\|_2$ and $g(x^{(2)}) = \|y^{(2)} - x^{(2)}\|_2$. Since X is convex, it holds that $\theta y^{(1)} + (1-\theta) y^{(2)} \in X$, and therefore

$$\begin{aligned} g(\theta x^{(1)} + (1 - \theta) x^{(2)}) &= \inf_{y \in X} \|y - (\theta x^{(1)} + (1 - \theta) x^{(2)})\|_2 \\ &\leq \|\theta y^{(1)} + (1 - \theta) y^{(2)} - (\theta x^{(1)} + (1 - \theta) x^{(2)})\|_2 \\ &\leq \theta \|y^{(1)} - x^{(1)}\|_2 + (1 - \theta) \|y^{(2)} - x^{(2)}\|_2 \\ &= \theta g(x^{(1)}) + (1 - \theta) g(x^{(2)}). \end{aligned}$$

Hence, $g = d_X$ is convex. Since $X = \{x \in \mathbb{R}^d : \inf_{y \in X} \|y - x\|_2 = 0\} = \{x \in \mathbb{R}^d : d_X(x) = 0\} = \{x \in \mathbb{R}^d : d_X(x) \leq 0\} = \{x \in \mathbb{R}^d : g(x) \leq 0\}$ by nonnegativity of d_X, the lemma holds. □

Proposition 1. *For any nonempty closed convex set $X \subseteq \mathbb{R}^d$, there exists $f \in \mathcal{F}_{\mathrm{Id}}$ such that $X = f^{-1}(\{2\}) = \{x \in \mathbb{R}^d : f(x) = 2\}$. In particular, this shows that if (X_1, X_2) is a convexly separable pair of subsets of $\mathbb{R}^d$, then there exists $f \in \mathcal{F}_{\mathrm{Id}}$ such that $f(x) = 1$ for all $x \in X_1$ and $f(x) = 2$ for all $x \in X_2$.*

Proof. Let $X \subseteq \mathbb{R}^d$ be a nonempty closed convex set. By Lemma 1, there exists a convex function $g \colon \mathbb{R}^d \to \mathbb{R}$ such that $X = \{x \in \mathbb{R}^d : g(x) \leq 0\}$. Define $f \colon \mathbb{R}^d \to \{1, 2\}$ by $f(x) = 1$ if $g(x) > 0$ and $f(x) = 2$ if $g(x) \leq 0$. Clearly, it holds that $f \in \mathcal{F}_{\mathrm{Id}}$. Furthermore, for all $x \in X$ it holds that $g(x) \leq 0$, implying that $f(x) = 2$ for all $x \in X$. Conversely, if $x \in \mathbb{R}^d$ is such that $f(x) = 2$, then $g(x) \leq 0$, implying that $x \in X$. Hence, $X = \{x \in \mathbb{R}^d : f(x) = 2\}$.

If (X_1, X_2) is a convexly separable pair of subsets of $\mathbb{R}^d$, then there exists a nonempty closed convex set $X \subseteq \mathbb{R}^d$ such that $X_2 \subseteq X$ and $X_1 \subseteq \mathbb{R}^d \setminus X$, and therefore there exists $f \in \mathcal{F}_{\mathrm{Id}}$ such that $X_2 \subseteq X = f^{-1}(\{2\})$ and $X_1 \subseteq \mathbb{R}^d \setminus X = f^{-1}(\{1\})$, implying that indeed $f(x) = 1$ for all $x \in X_1$ and $f(x) = 2$ for all $x \in X_2$. □

Proposition 2. *Let $f \in \mathcal{F}_{\mathrm{Id}}$. The decision region under f associated to class 2, namely $X \coloneqq f^{-1}(\{2\}) = \{x \in \mathbb{R}^d : f(x) = 2\}$, is a closed convex set.*

Proof. For all $x \in \mathbb{R}^d$, it holds that $f(x) = 2$ if and only if $g(x) \leq 0$. Since $f \in \mathcal{F}_{\mathrm{Id}}$, g is convex, and hence, $X = \{x \in \mathbb{R}^d : g(x) \leq 0\}$ is a (nonstrict) sublevel set of a convex function and is therefore a closed convex set. □

In order to apply the universal approximation results in Chen et al. (2019), we now introduce their parameterization of input-convex ReLU neural networks. Note that it imposes the additional constraint that the first weight matrix $A^{(1)}$ is elementwise nonnegative.

Definition 4. Define $\tilde{\mathcal{F}}_{\mathrm{Id}}$ to be the class of functions $\tilde{f} \colon \mathbb{R}^d \to \{1, 2\}$ given by

$$\tilde{f}(x) = \begin{cases} 1 & \text{if } \tilde{g}(x) > 0, \\ 2 & \text{if } \tilde{g}(x) \leq 0, \end{cases}$$

with $\tilde{g} \colon \mathbb{R}^d \to \mathbb{R}$ given by

$$\begin{aligned} x^{(1)} &= \mathrm{ReLU}\left(A^{(1)}x + b^{(1)}\right), \\ x^{(l)} &= \mathrm{ReLU}\left(A^{(l)}x^{(l-1)} + b^{(l)} + C^{(l)}x\right),\ l \in \{2, 3, \ldots, L-1\}, \\ \tilde{g}(x) &= A^{(L)}x^{(L-1)} + b^{(L)} + C^{(L)}x, \end{aligned}$$

for some $L \in \mathbb{N}$, $L > 1$, and some consistently sized matrices $A^{(1)}, C^{(1)}, \ldots, A^{(L)}, C^{(L)}$, all of which have nonnegative elements, and some consistently sized vectors $b^{(1)}, \ldots, b^{(L)}$.

The following preliminary lemma relates the class $\hat{\mathcal{F}}_{\mathrm{Id}}$ from Definition 2 to the class $\tilde{\mathcal{F}}_{\mathrm{Id}}$ above.

Lemma 2. *It holds that $\tilde{\mathcal{F}}_{\mathrm{Id}} \subseteq \hat{\mathcal{F}}_{\mathrm{Id}}$.*

Proof. Let $\tilde{f} \in \tilde{\mathcal{F}}_{\mathrm{Id}}$. Then certainly $A^{(l)} \geq 0$ for all $l \in \{2, 3, \ldots, L\}$, so indeed $\tilde{f} \in \hat{\mathcal{F}}_{\mathrm{Id}}$. Hence, $\tilde{\mathcal{F}}_{\mathrm{Id}} \subseteq \hat{\mathcal{F}}_{\mathrm{Id}}$. □

Theorem 1 in Chen et al. (2019) shows that a Lipschitz convex function can be approximated within an arbitrary tolerance. We now provide a technical lemma adapting Theorem 1 in Chen et al. (2019) to show that convex functions can be *underapproximated* within an arbitrary tolerance on a compact convex subset.

Lemma 3. *For any convex functions $g \colon \mathbb{R}^d \to \mathbb{R}$, any compact convex subsets X of $\mathbb{R}^d$, and any $\epsilon > 0$, there exists $\hat{f} \in \hat{\mathcal{F}}_{\mathrm{Id}}$ such that $\hat{g}(x) < g(x)$ for all $x \in X$ and $\sup_{x \in X}(g(x) - \hat{g}(x)) < \epsilon$.*

Proof. Let $g \colon \mathbb{R}^d \to \mathbb{R}$ be a convex function, let X be a compact convex subset of $\mathbb{R}^d$, and let $\epsilon > 0$. Since $g - \epsilon/2$ is a real-valued convex function on $\mathbb{R}^d$ (and hence is proper), its restriction to the closed and bounded set X is Lipschitz continuous (Rockafellar, 1970, Theorem 10.4), and therefore Lemma 2 together with Theorem 1 in Chen et al. (2019) gives that there exists $\hat{f} \in \tilde{\mathcal{F}}_{\mathrm{Id}} \subseteq \hat{\mathcal{F}}_{\mathrm{Id}}$ such that $\sup_{x \in X} |(g(x) - \epsilon/2) - \hat{g}(x)| < \epsilon/2$. Thus, for all $x \in X$,

$$\begin{aligned} g(x) - \hat{g}(x) &= \left(g(x) - \frac{\epsilon}{2}\right) - \hat{g}(x) + \frac{\epsilon}{2} \\ &> \left(g(x) - \frac{\epsilon}{2}\right) - \hat{g}(x) + \sup_{y \in X} \left|\left(g(y) - \frac{\epsilon}{2}\right) - \hat{g}(y)\right| \\ &\geq \left(g(x) - \frac{\epsilon}{2}\right) - \hat{g}(x) + \left|\left(g(x) - \frac{\epsilon}{2}\right) - \hat{g}(x)\right| \\ &\geq 0. \end{aligned}$$

Furthermore,

$$\begin{aligned}\sup_{x\in X}\left(g(x)-\hat{g}(x)\right) &= \sup_{x\in X}\left|g(x)-\hat{g}(x)\right| \\ &= \sup_{x\in X}\left|\left(g(x)-\frac{\epsilon}{2}\right)-\hat{g}(x)+\frac{\epsilon}{2}\right| \\ &\leq \sup_{x\in X}\left|\left(g(x)-\frac{\epsilon}{2}\right)-\hat{g}(x)\right|+\frac{\epsilon}{2} \\ &< \epsilon,\end{aligned}$$

which proves the lemma. □

We leverage Lemma 3 to construct a uniformly converging sequence of underapproximating functions.

Lemma 4. *For all $f \in \mathcal{F}_{\mathrm{Id}}$ and all compact convex subsets X of $\mathbb{R}^d$, there exists a sequence $\{\hat{f}_n \in \hat{\mathcal{F}}_{\mathrm{Id}} : n \in \mathbb{N}\} \subseteq \hat{\mathcal{F}}_{\mathrm{Id}}$ such that $\hat{g}_n(x) < \hat{g}_{n+1}(x) < g(x)$ for all $x \in X$ and all $n \in \mathbb{N}$ and $\hat{g}_n$ converges uniformly to g on X as $n \to \infty$.*

Proof. Let $f \in \mathcal{F}_{\mathrm{Id}}$ and let X be a compact convex subset of $\mathbb{R}^d$. Let $\{\epsilon_n > 0 : n \in \mathbb{N}\}$ be a sequence such that $\epsilon_{n+1} < \epsilon_n$ for all $n \in \mathbb{N}$ and $\epsilon_n \to 0$ as $n \to \infty$. Such a sequence clearly exists, e.g., by taking $\epsilon_n = 1/n$ for all $n \in \mathbb{N}$. Now, for all $n \in \mathbb{N}$, the function $g - \epsilon_{n+1}$ is convex, and therefore by Lemma 3 there exists $\hat{f}_n \in \hat{\mathcal{F}}_{\mathrm{Id}}$ such that $\hat{g}_n(x) < g(x) - \epsilon_{n+1}$ for all $x \in X$ and $\sup_{x\in X}\left((g(x)-\epsilon_{n+1})-\hat{g}_n(x)\right) < \epsilon_n - \epsilon_{n+1}$. Fixing such $\hat{f}_n, \hat{g}_n$ for all $n \in \mathbb{N}$, we see that $\sup_{x\in X}\left((g(x)-\epsilon_{n+2})-\hat{g}_{n+1}(x)\right) < \epsilon_{n+1} - \epsilon_{n+2}$, which implies that

$$\hat{g}_{n+1}(x) > g(x) - \epsilon_{n+1} > \hat{g}_n(x)$$

for all $x \in X$, which proves the first inequality. The second inequality comes from the fact that $\hat{g}_{n+1}(x) < g(x) - \epsilon_{n+2} < g(x)$ for all $x \in X$. Finally, since $g(x) - \hat{g}_n(x) > \epsilon_{n+1} > 0$ for all $x \in X$ and all $n \in \mathbb{N}$, we see that

$$\sup_{x\in X}\left|g(x)-\hat{g}_n(x)\right| = \sup_{x\in X}\left(g(x)-\hat{g}_n(x)\right) < \epsilon_n \to 0 \text{ as } n \to \infty,$$

which proves that $\lim_{n\to\infty}\sup_{x\in X}\left|g(x)-\hat{g}_n(x)\right| = 0$, so indeed $\hat{g}_n$ converges uniformly to g on X as $n \to \infty$. □

With all the necessary lemmas in place, we now present our main theoretical results.

Theorem 2. *For any $f \in \mathcal{F}_{\mathrm{Id}}$, any compact convex subsets X of $\mathbb{R}^d$, and any $\epsilon > 0$, there exists $\hat{f} \in \hat{\mathcal{F}}_{\mathrm{Id}}$ such that $m(\{x \in X : \hat{f}(x) \neq f(x)\}) < \epsilon$.*

Proof. Let $f \in \mathcal{F}_{\mathrm{Id}}$ and let X be a compact convex subset of $\mathbb{R}^d$. By Lemma 4, there exists a sequence $\{\hat{f}_n \in \hat{\mathcal{F}}_{\mathrm{Id}} : n \in \mathbb{N}\} \subseteq \hat{\mathcal{F}}_{\mathrm{Id}}$ such that $\hat{g}_n(x) < \hat{g}_{n+1}(x) < g(x)$ for all $x \in X$ and all $n \in \mathbb{N}$ and $\hat{g}_n$ converges uniformly to g on X as $n \to \infty$. Fix this sequence.

For all $n \in \mathbb{N}$, define

$$E_n = \{x \in X : \hat{f}_n(x) \neq f(x)\},$$

i.e., the set of points in X for which the classification under $\hat{f}_n$ does not agree with that under f. Since $\hat{g}_n(x) < g(x)$ for all $x \in X$ and all $n \in \mathbb{N}$, we see that

$$\begin{aligned}E_n &= \{x \in X : \hat{g}_n(x) > 0 \text{ and } g(x) \leq 0\} \cup \{x \in X : \hat{g}_n(x) \leq 0 \text{ and } g(x) > 0\} \\ &= \{x \in X : \hat{g}_n(x) \leq 0 \text{ and } g(x) > 0\}.\end{aligned}$$

Since g is a real-valued convex function on $\mathbb{R}^d$, it is continuous (Rockafellar, 1970, Corollary 10.1.1), and therefore $g^{-1}((0,\infty)) = \{x \in \mathbb{R}^d : g(x) > 0\}$ is measurable. Similarly, $\hat{g}_n^{-1}((-\infty, 0]) = \{x \in \mathbb{R}^d : \hat{g}_n(x) \leq 0\}$ is also measurable for all $n \in \mathbb{N}$ since $\hat{g}_n$ is continuous. Furthermore, X is measurable as it is compact. Therefore, E_n is measurable for all $n \in \mathbb{N}$. Now, since $\hat{g}_n(x) < \hat{g}_{n+1}(x)$ for all $x \in X$ and all $n \in \mathbb{N}$, it holds that $E_{n+1} \subseteq E_n$ for all $n \in \mathbb{N}$. It is clear that to prove the result, it suffices to show that $\lim_{n\to\infty} m(E_n) = 0$. Therefore, if we show

that $m\left(\bigcap_{n\in\mathbb{N}} E_n\right) = 0$, then the fact that $m(E_1) \leq m(X) < \infty$ together with Lebesgue measure's continuity from above yields that $\lim_{n\to\infty} m(E_n) = 0$, thereby proving the result.

It remains to be shown that $m\left(\bigcap_{n\in\mathbb{N}} E_n\right) = 0$. To this end, suppose for the sake of contradiction that $\bigcap_{n\in\mathbb{N}} E_n \neq \emptyset$. Then there exists $x \in \bigcap_{n\in\mathbb{N}} E_n$, meaning that $g(x) > 0$ and $\hat{g}_n(x) \leq 0$ for all $n \in \mathbb{N}$. Thus, for this $x \in X$, we find that $\limsup_{n\to\infty} \hat{g}_n(x) \leq 0 < g(x)$, which contradicts the fact that $\hat{g}_n$ uniformly converges to g on X. Therefore, it must be that $\bigcap_{n\in\mathbb{N}} E_n = \emptyset$, and thus $m\left(\bigcap_{n\in\mathbb{N}} E_n\right) = 0$, which concludes the proof. □

Theorem 3. *If (X_1, X_2) is a convexly separable pair of finite subsets of $\mathbb{R}^d$, then there exists $\hat{f} \in \hat{\mathcal{F}}_{\mathrm{Id}}$ such that $\hat{f}(x) = 1$ for all $x \in X_1$ and $\hat{f}(x) = 2$ for all $x \in X_2$.*

Proof. Throughout this proof, we denote the complement of a set $Y \subseteq \mathbb{R}^d$ by $Y^c = \mathbb{R}^d \setminus Y$.

Suppose that $X_1 = \{x^{(1)}, \ldots, x^{(M)}\} \subseteq \mathbb{R}^d$ and $X_2 = \{y^{(1)}, \ldots, y^{(N)}\} \subseteq \mathbb{R}^d$ are such that (X_1, X_2) is convexly separable. Then, by definition of convex separability, there exists a nonempty closed convex set $X' \subseteq \mathbb{R}^d$ such that $X_2 \subseteq X'$ and $X_1 \subseteq \mathbb{R}^d \setminus X'$. Let $X = X' \cap \mathrm{conv}(X_2)$. Since $X_2 \subseteq X'$ and both sets X' and $\mathrm{conv}(X_2)$ are convex, the set X is nonempty and convex. By finiteness of X_2, the set $\mathrm{conv}(X_2)$ is compact, and therefore by closedness of X', the set X is compact and hence closed.

By Proposition 1, there exists $f \in \mathcal{F}_{\mathrm{Id}}$ such that $f^{-1}(\{2\}) = X$. Since $\mathrm{conv}(X_1 \cup X_2)$ is compact and convex, Lemma 4 gives that there exists a sequence $\{\hat{f}_n \in \hat{\mathcal{F}}_{\mathrm{Id}} : n \in \mathbb{N}\} \subseteq \hat{\mathcal{F}}_{\mathrm{Id}}$ such that $\hat{g}_n(x) < \hat{g}_{n+1}(x) < g(x)$ for all $x \in \mathrm{conv}(X_1 \cup X_2)$ and all $n \in \mathbb{N}$ and $\hat{g}_n$ converges uniformly to g on $\mathrm{conv}(X_1 \cup X_2)$ as $n \to \infty$. Fix this sequence.

Let $x \in X_2$. Then, since $X_2 \subseteq X'$ and $X_2 \subseteq \mathrm{conv}(X_2)$, it holds that $x \in X' \cap \mathrm{conv}(X_2) = X = f^{-1}(\{2\})$, implying that $f(x) = 2$ and hence $g(x) \leq 0$. Since $\hat{g}_n(x) < g(x)$ for all $n \in \mathbb{N}$, this shows that $\hat{f}_n(x) = 2$ for all $n \in \mathbb{N}$. On the other hand, let $i \in \{1, \ldots, M\}$ and consider $x = x^{(i)} \in X_1$. Since $X_1 \subseteq \mathbb{R}^d \setminus X' = \mathbb{R}^d \cap (X')^c \subseteq \mathbb{R}^d \cap (X' \cap \mathrm{conv}(X_2))^c = \mathbb{R}^d \cap X^c = \mathbb{R}^d \cap f^{-1}(\{1\})$, it holds that $f(x) = 1$ and thus $g(x) > 0$. Suppose for the sake of contradiction that $\hat{f}_n(x) = 2$ for all $n \in \mathbb{N}$. Then $\hat{g}_n(x) \leq 0$ for all $n \in \mathbb{N}$. Therefore, for this $x \in X_1$, we find that $\limsup_{n\to\infty} \hat{g}_n(x) \leq 0 < g(x)$, which contradicts the fact that $\hat{g}_n$ uniformly converges to g on $\mathrm{conv}(X_1 \cup X_2)$. Therefore, it must be that there exists $n_i \in \mathbb{N}$ such that $\hat{f}_{n_i}(x) = 1$, and thus $\hat{g}_{n_i}(x) > 0$. Since $\hat{g}_n(x) < \hat{g}_{n+1}(x)$ for all $n \in \mathbb{N}$, this implies that $\hat{g}_n(x) > 0$ for all $n \geq n_i$. Hence, $\hat{f}_n(x) = \hat{f}_n(x^{(i)}) = 1$ for all $n \geq n_i$.

Let $n^\star$ be the maximum of all such n_i, i.e., $n^\star = \max\{n_i : i \in \{1, \ldots, M\}\}$. Then the above analysis shows that $\hat{f}_{n^\star}(x) = 2$ for all $x \in X_2$ and that $\hat{f}_{n^\star}(x) = 1$ for all $x \in X_1$. Since $\hat{f}_{n^\star} \in \hat{\mathcal{F}}_{\mathrm{Id}}$, the claim has been proven. □

Theorem 4. *Consider $M, N \in \mathbb{N}$. Let $X_1 = \{x^{(1)}, \ldots, x^{(M)}\} \subseteq \mathbb{R}^d$ and $X_2 = \{y^{(1)}, \ldots, y^{(N)}\} \subseteq \mathbb{R}^d$ be samples with all elements $x_k^{(i)}, y_l^{(j)}$ drawn independently and identically from the uniform probability distribution on $[-1, 1]$. Then, it holds that*

$$\mathbb{P}\left((X_1, X_2) \text{ is convexly separable}\right) \geq 1 - \left(1 - \frac{1}{2^{MN}}\right)^d. \tag{1}$$

In particular, the probability that $\hat{\mathcal{F}}_{\mathrm{Id}}$ contains an input-convex ReLU *neural network that classifies all $x^{(i)}$ into class 1 and all $y^{(j)}$ into class 2 converges linearly to 1 as $d \to \infty$.*

Proof. For notational convenience, let $P \geq 0$ be the probability of interest:

$$P = \mathbb{P}\left((X_1, X_2) \text{ is convexly separable}\right).$$

Suppose that there exists a coordinate $k \in \{1, 2, \ldots, d\}$ such that $x_k^{(i)} < y_k^{(j)}$ for all pairs $(i, j) \in \{1, 2, \ldots, M\} \times \{1, 2, \ldots, N\}$ and that $a := \min\{y_k^{(1)}, \ldots, y_k^{(N)}\} < \max\{y_k^{(1)}, \ldots, y_k^{(N)}\} =: b$. Then, let $X = \{x \in \mathbb{R}^d : x_k \in [a, b]\}$. That is, X is the extrusion of the convex hull of the projections $\{y_k^{(1)}, \ldots, y_k^{(N)}\}$ along all remaining coordinates. The set X is a nonempty closed

convex set, and it is clear by our supposition that $X_2 \subseteq X$ and $X_1 \subseteq \mathbb{R}^d \setminus X$. Therefore, the supposition implies that (X_1, X_2) is convexly separable, and thus

$$\begin{aligned}
P &\geq \mathbb{P}\big(\text{there exists } k \in \{1,2,\dots,d\} \text{ such that } x_k^{(i)} < y_k^{(j)} \text{ for all pairs } (i,j) \\
&\quad \text{and that } \min\{y_k^{(1)},\dots,y_k^{(N)}\} < \max\{y_k^{(1)},\dots,y_k^{(N)}\}\big) \\
&= 1 - \mathbb{P}\big(\text{for all } k \in \{1,2,\dots,d\}, \text{ it holds that } x_k^{(i)} \geq y_k^{(j)} \text{ for some pair } (i,j) \\
&\quad \text{or that } \min\{y_k^{(1)},\dots,y_k^{(N)}\} = \max\{y_k^{(1)},\dots,y_k^{(N)}\}\big) \\
&= 1 - \prod_{k=1}^{d} \mathbb{P}\left(x_k^{(i)} \geq y_k^{(j)} \text{ for some pair } (i,j) \text{ or } \min\{y_k^{(1)},\dots,y_k^{(N)}\} = \max\{y_k^{(1)},\dots,y_k^{(N)}\}\right),
\end{aligned}$$

where the final equality follows from the independence of the coordinates of the samples. Since $\min\{y_k^{(1)},\dots,y_k^{(N)}\} < \max\{y_k^{(1)},\dots,y_k^{(N)}\}$ almost surely, we find that

$$\begin{aligned}
P &\geq 1 - \prod_{k=1}^{d} \Bigg(\mathbb{P}(x_k^{(i)} \geq y_k^{(j)} \text{ for some pair } (i,j)) \\
&\qquad + \mathbb{P}(\min\{y_k^{(1)},\dots,y_k^{(N)}\} = \max\{y_k^{(1)},\dots,y_k^{(N)}\}) \Bigg) \\
&= 1 - \prod_{k=1}^{d} \mathbb{P}(x_k^{(i)} \geq y_k^{(j)} \text{ for some pair } (i,j)) \\
&= 1 - \prod_{k=1}^{d} \Big(1 - \mathbb{P}(x_k^{(i)} < y_k^{(j)} \text{ for all pairs } (i,j))\Big) \\
&= 1 - \prod_{k=1}^{d} \left(1 - \prod_{(i,j)\in\{1,\dots,M\}\times\{1,\dots,N\}} \mathbb{P}(x_k^{(i)} < y_k^{(j)})\right) \\
&= 1 - \prod_{k=1}^{d} \left(1 - \prod_{(i,j)\in\{1,\dots,M\}\times\{1,\dots,N\}} \frac{1}{2}\right) \\
&= 1 - \left(1 - \frac{1}{2^{MN}}\right)^d,
\end{aligned}$$

which proves (1). The linear convergence to 1 as $d \to \infty$ of the probability of $\hat{\mathcal{F}}_{\mathrm{Id}}$ containing a classifier that classifies all $x^{(i)}$ into class 1 and all $y^{(j)}$ into class 2 follows immediately from Theorem 3. □

D CIFAR-10 CATS-VERSUS-DOGS CONVEX SEPARABILITY

In order to establish that the cat and dog images in CIFAR-10 are convexly separable, we experimentally attempt to reconstruct an image from one class using a convex combination of all images in the other class (without augmentation such as random crops, flips, etc.). Namely, if x is drawn from one class and $y^{(1)}, \ldots, y^{(N)}$ represent the entirety of the other class, we form the following optimization problem:

$$\begin{aligned} \underset{\alpha \in \mathbb{R}^N}{\text{minimize}} \quad & \left\| x - \sum_{j=1}^{N} \alpha_j y^{(j)} \right\|_2 \\ \text{subject to} \quad & \alpha \geq 0, \\ & \sum_{j=1}^{N} \alpha_j = 1. \end{aligned}$$

The reverse experiment for the other class follows similarly. We solve the optimization using MOSEK (ApS, 2019), and report the various norms of $x - \sum_{j=1}^{N} \alpha_j y^{(j)}$ in Figure 4. Reconstruction accuracy is generally very poor, with no reconstruction achieving better than an ℓ_1-error of 52. A typical reconstructed image is shown in Figure 5.

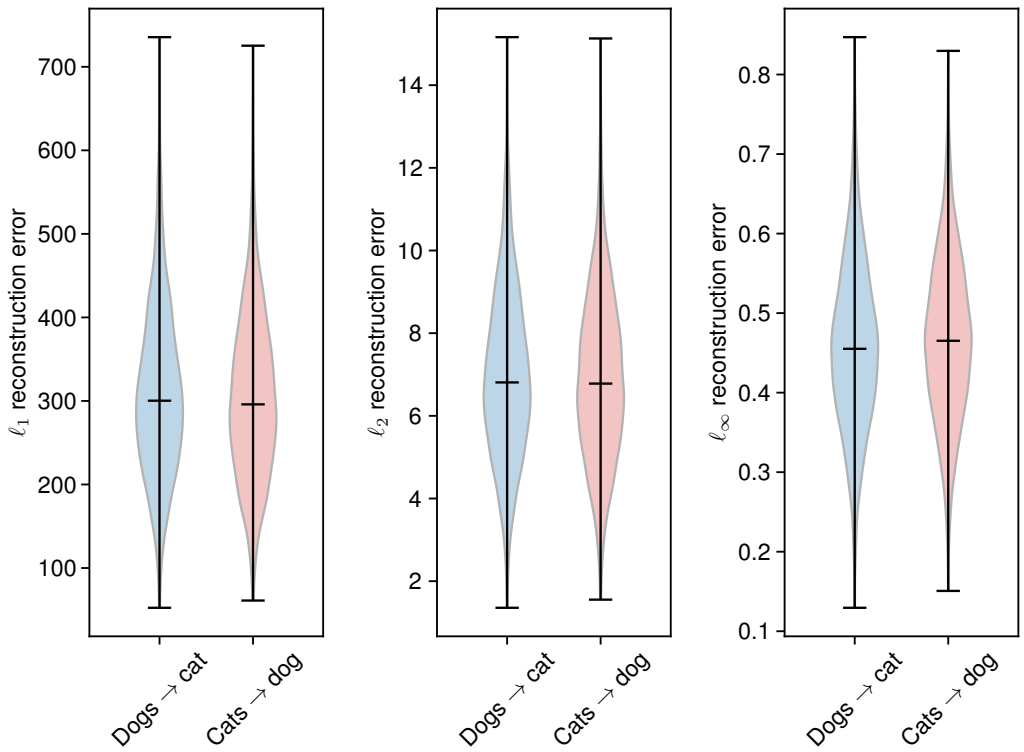


Figure 4: Reconstructing CIFAR-10 cat and dog images as convex combinations. The label "Dogs → cat" indicates that a cat image was attempted to be reconstructed as a convex combination of all $5,000$ dog images.

Figure 5: Reconstructing a CIFAR-10 cat image (left) from a convex combination of dog images (right). The reconstruction error norms are 294.57, 6.65, and 0.38 for the ℓ_1-, ℓ_2-, and ℓ_∞-norms, respectively. These are typical, as indicated by Figure 4.

E EXPERIMENTAL SETUP

We include a detailed exposition of our experimental setup in this section, beginning with general details on our choice of epochs and batch size. We then discuss baseline methods, architecture choices for our method, class balancing, and data processing.

Epochs and batch size. For the MNIST 3-8 experiments, we used 30 epochs for all methods. This was increased to 50 epochs for the Malimg dataset, 150 epochs for CIFAR-10 cats-dogs (besides the smoothing baselines which were given 600 epochs), and 225 epochs for Kaggle cats-dogs. The batch size was 64 for all datasets besides the 512×512 Malimg dataset, where it was lowered to 32.

Hardware. All experiments were conducted on a single Ubuntu 20.04 instance with an Nvidia RTX A6000 GPU. Complete reproduction of the experiments takes approximately 0.08 GPU-years.

E.1 BASELINE METHODS

We provide additional details on each of the baseline methods below.

Randomized smoothing. Since the certification runtime of randomized smoothing is large, especially for the 512×512 pixel Malimg images, we evaluate the randomized smoothing classifiers over 10^4 samples and project the certified radius to 10^5 samples by scaling the number fed into the Clopper-Pearson confidence interval, as described in Cohen et al. (2019) (we exempt the small MNIST 3-8 dataset and evaluate the full 10^5 samples). This allows for a representative and improved certified accuracy curve while dramatically reducing the method's runtime. We take an initial guess for the certification class with $n_0 = 100$ samples and set the incorrect prediction tolerance parameter $\alpha = 0.001$. Our MNIST base classifier was a two-hidden layer ReLU multilayer perceptron with $(n_1, n_2) = (200, 50)$ hidden features to maintain consistency with the other MNIST methods. For CIFAR-10 we use a depth-40 Wide ResNet and for Kaggle we use a ResNet-50 architecture, mirroring the choices from Cohen et al. (2019); Yang et al. (2020a). To improve computational efficiency on the large Malimg images we use a ResNet-18. All networks were trained using SGD with an initial learning rate of 0.1, Nesterov momentum of 0.9, weight decay of 10^{-4}, and cosine annealing scheduling as described in Yang et al. (2020a). We set the smoothing noise parameter $\sigma = 0.75$ for the MNIST 3-8 and CIFAR-10 cats-dogs experiments; for the higher-resolution Kaggle cats-dogs and Malimg experiments we increase the noise to $\sigma = 3.5$, matching the highest noise level examined in Levine & Feizi (2021).

Splitting noise. As this method is a deterministic derivative of randomized smoothing, it avoids the many aforementioned hyperparameter choices. We use the same architectures described above for the other randomized smoothing experiments.

Cayley convolutions. To maintain consistency, we use a two-hidden layer multilayer perceptron with $(n_1, n_2) = (200, 50)$ hidden features, CayleyLinear layers, and GroupSort activations for the MNIST experiment. For the CIFAR-10 experiment, we use the ResNet9 architecture implementation from Trockman & Kolter (2021). Following the authors' suggestions, we trained these networks using Adam with a learning rate of 0.001.

ℓ_∞-distance nets. As the architecture of the ℓ_∞-distance net (Zhang et al., 2021a) is substantially different from traditional architectures, we use the authors' 5-layer MNIST architecture and 6-layer CIFAR-10 architecture with 5120 neurons per hidden layer. Unfortunately, the classification accuracy on the CIFAR-10 cats-dogs experiment remained near 50% throughout training. This was not the case when we tested easier classes, such as planes-versus-cars, where large features (e.g., blue sky) can be used to discriminate. We therefore only include this model in the MNIST experiments, and use the training procedure directly from the aforementioned paper's codebase.

E.2 Convex ConvNet architecture and training

The convex ConvNet architecture consists of a sequence of convolutional layers, BatchNorms, and ReLU nonlinearities. The first convolutional layer is unconstrained, as the composition of a convex function with an affine function is still convex (Amos et al., 2017). All subsequent convolutions and the final linear readout layer are uniformly initialized from some small positive weight interval ($[0, 0.003]$ for linear weights, $[0, 0.005]$ for convolutional weights) and projected to have nonnegative weights after each gradient step. We found this heuristic initialization choice helps to stabilize network training, as standard Kaiming initialization assumptions are violated when weights are constrained to be nonnegative instead of normally distributed with mean zero. More principled weight initialization strategies for this architecture would form an exciting area of future research. Before any further processing, inputs into the network are fed into an initial BatchNorm; this enables flexibility with different feature augmentation maps.

Since the first convolutional layer is permitted negative weights, we generally attain better performance by enlarging the first convolution kernel size (see Table 1). For subsequent convolutions, we set the stride to 1, the input and output channel counts to the output channel count from the first convolution, and the padding to half the kernel size, rounded down. This ensures that the output of each of these deeper convolutions has equivalent dimension to its input, allowing for an identity residual connection across each convolution. If $C_i(z)$ is a convolutional operation on a hidden feature z, this corresponds to evaluating $C_i(z) + z$ instead of just $C_i(z)$. The final part of the classifier applies MaxPool and BatchNorm layers before a linear readout layer with output dimension 1. See Figure 6 for a diagram depicting an exemplar convex ConvNet instantiation.

For training, we use a standard binary cross entropy loss, optionally augmented with a Jacobian regularizer (Hoffman et al., 2019) scaled by $\lambda > 0$. As our certified radii in Theorem 1 vary inversely to the norm of the Jacobian, this regularization helps boost our certificates at a minimal loss in clean accuracy. We choose $\lambda = 0.1$ for Malimg, $\lambda = 0.01$ for Kaggle and CIFAR-10 cats-dogs, and $\lambda = 0.0001$ for MNIST 3-8. Further ablation tests studying the impact of regularization are reported in Appendix G. All feature-convex networks were trained using SGD with a learning rate of 0.001, momentum 0.9, and exponential learning rate decay with $\gamma = 0.99$.

Dataset	Features	Depth	C_1 size	C_1 stride	C_1 dilation	$C_{2,\ldots}$ size	Pool
Malimg	32	4	21	2	1	3	4
CIFAR-10	16	5	11	1	1	3	1
Kaggle	32	5	15	1	2	3	8

Table 1: Convex ConvNet architecture parameters. C_1 denotes the first convolution, with $C_{2,\ldots}$ denoting all subsequent convolutions. The "Features" column denotes the number of output features of C_1, which is held fixed across $C_{2,\ldots}$. The "Pool" column refers to the size of the final MaxPool window before the linear readout layer. The MNIST architecture is a simple multilayer perceptron and is therefore not listed here.

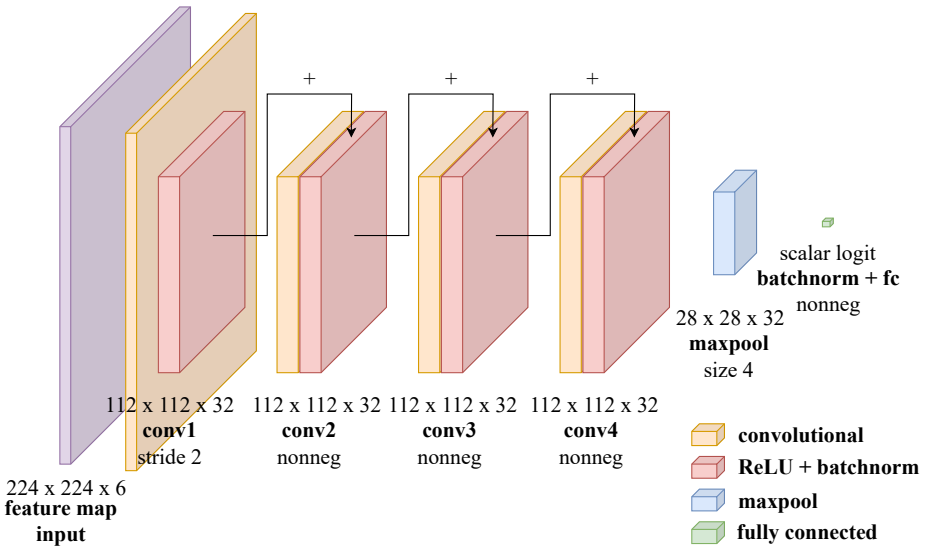


Figure 6: An example convex ConvNet of depth 4 with a C_1 stride of 2, pool size of 4, and 224×224 images. There are 6 input channels from the output of the feature map $\varphi\colon x \mapsto (x - \mu, |x - \mu|)$.

E.3 Class accuracy balancing

As discussed in Section 4, a balanced class 1 and class 2 test accuracy is essential for a fair comparison of different methods. For methods where the output logits can be directly balanced, this is easily accomplished by computing the ROC curve and choosing the threshold that minimizes $|\mathrm{TPR} - (1 - \mathrm{FPR})|$. This includes both our feature-convex classifiers with one output logit and the Cayley orthogonalization and ℓ_∞-net architectures with two output logits.

Randomized smoothing classifiers are more challenging as the relationship between the base classifier threshold and the smoothed classifier prediction is indirect. We address this using an iterative balancing procedure. Namely, on each iteration, the classifier's prediction routine is executed over the test dataset and the "error" between the class 1 accuracy and the class 2 accuracy is computed. The base classifier decision threshold is then shifted proportionally to the error, and the procedure is continued until the error magnitude drops below 1%.

E.4 Data processing

For consistency with Zhang et al. (2021a), we augment the MNIST training data with 1-pixel padding and random cropping. The CIFAR-10 dataset is augmented with 3-pixel edge padding, horizontal flips, and random cropping. We similarly randomly crop and flip the Kaggle cats-dogs dataset, with scaling bounds $[0.5, 1.0]$. The Malimg dataset is augmented with 20-pixel padding and random 512×512 cropping.

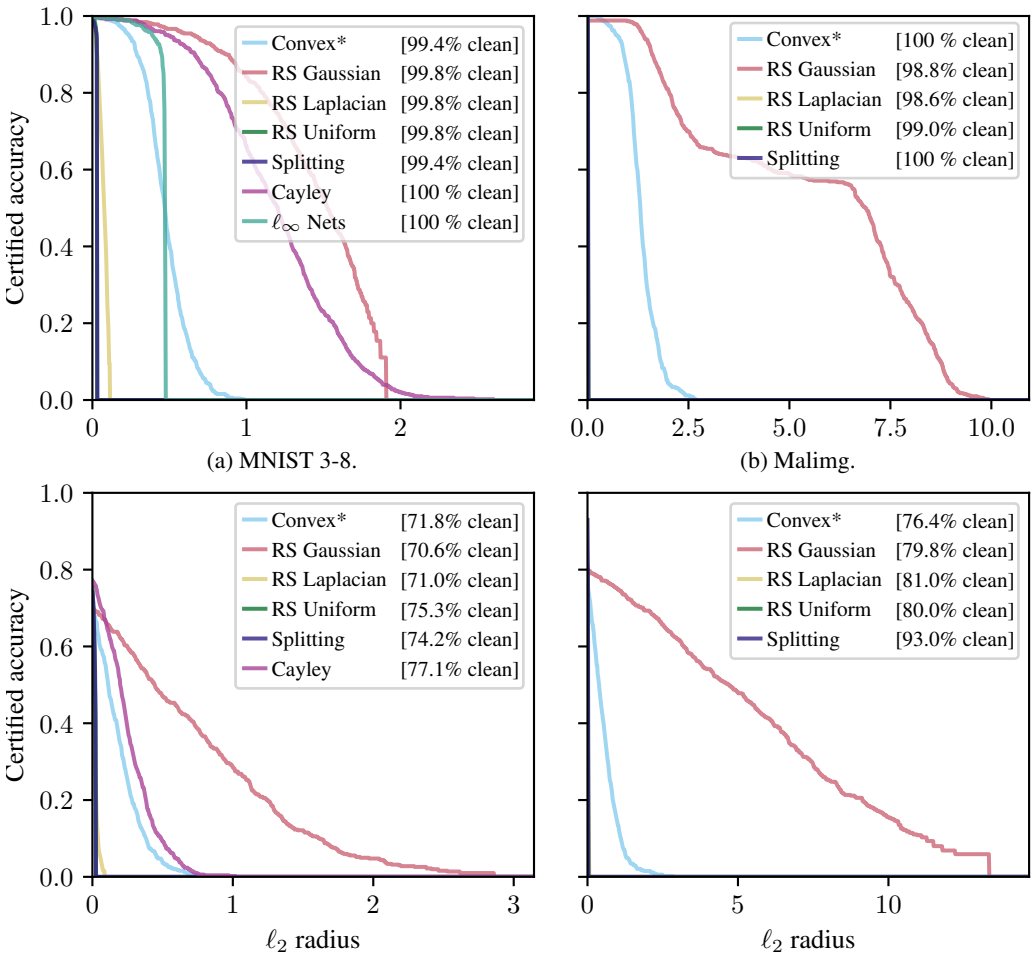
For CIFAR-10 and MNIST, we use the preselected test sets. For Malimg and Kaggle cats-dogs we hold out a random 20% and 10% test dataset, respectively, although this may not be entirely used during testing. The training set is further subdivided by an 80%-20% validation split. For all experiments, we use the first 1000 test samples to evaluate our methods.

F ℓ_2- and ℓ_∞-certified radii

This section reports the counterpart to Figure 2 for the ℓ_2- and ℓ_∞-norms. Across all experiments, we attain substantial ℓ_2 and ℓ_∞ radii without relying on computationally expensive sampling schemes or nondeterminism. Methods that certify to another norm $\|\cdot\|_p$ are converted to ℓ_q-radii at a factor of 1 if $p > q$ or $d^{1/p-1/q}$ otherwise.

Certified ℓ_2-radii are reported in Figure 7. Note that some methods' certificates scale poorly in high dimensions and are thus omitted in the Malimg and Kaggle cats-dogs plots. Our ℓ_2-radii are moderate, generally half the size of those produced by Gaussian randomized smoothing on low-resolution datasets. Higher resolution images prove more challenging—see the dimensionality discussion in Section 4.2.

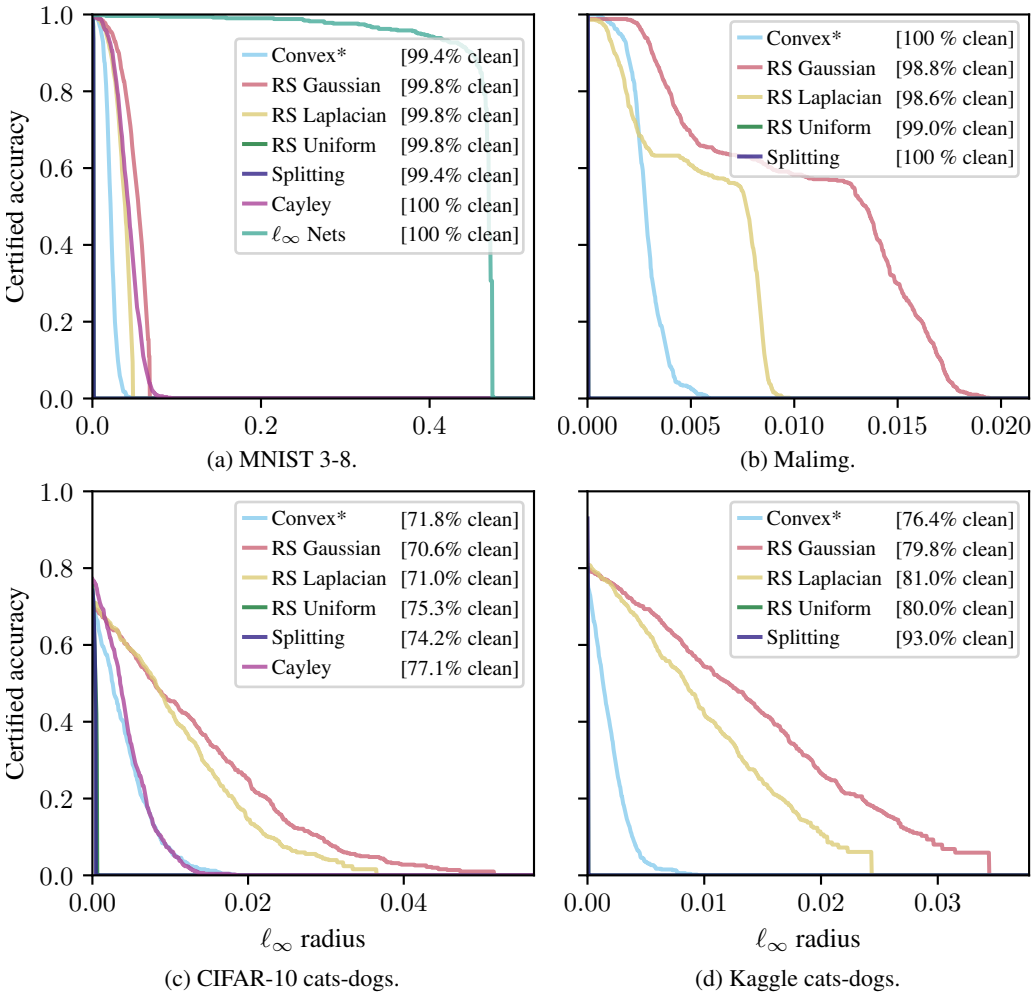
Certified ℓ_∞-radii are reported in Figure 8. For the MNIST 3-8 experiment, the ℓ_∞-distance nets produce exceptional certified radii, nearly achieving the theoretical maximum certified radii of 0.5; this is a logical upper bound as it suffices to perturb the entire image to a uniform gray. Compared to randomized smoothing, our ℓ_∞-radii seem to scale similarly for the higher-dimensional Malimg and Kaggle cats-dogs datasets.



(a) MNIST 3-8. (b) Malimg.

(c) CIFAR-10 cats-dogs. (d) Kaggle cats-dogs.

Figure 7: Class 1 certified radii curves for the ℓ_2-norm. Methods with negligible radii not shown.



(a) MNIST 3-8. (b) Malimg.

(c) CIFAR-10 cats-dogs. (d) Kaggle cats-dogs.

Figure 8: Class 1 certified radii curves for the ℓ_∞-norm. Methods with negligible radii not shown.

G Ablation tests

We conduct a series of ablation tests on the CIFAR-10 cats-dogs dataset, examining the impact of regularization, feature maps, and data augmentation.

G.1 Regularization

Figure 9 examines the impact of Jacobian regularization over a range of regularization scaling factors λ, with $\lambda = 0$ corresponding to no regularization. Clean accuracy is minimally affected, while increasing λ generally enlarges the certified radii. Further increases in λ yield minimal additional benefits.

G.2 Feature map

In this section, we investigate the importance of the feature map φ. Figure 10 compares our standard feature-convex classifier with $\varphi(x) = (x-\mu, |x-\mu|)$ against an equivalent architecture with $\varphi = \mathrm{Id}$. Note that the initial layer in the convex ConvNet is a BatchNorm, so even with $\varphi = \mathrm{Id}$, features still get normalized before being passed into the convolutional architecture. We perform this experiment across both the standard cats-dogs experiment (cats are certified) in the main text and the reverse dogs-cats experiment (dogs are certified).

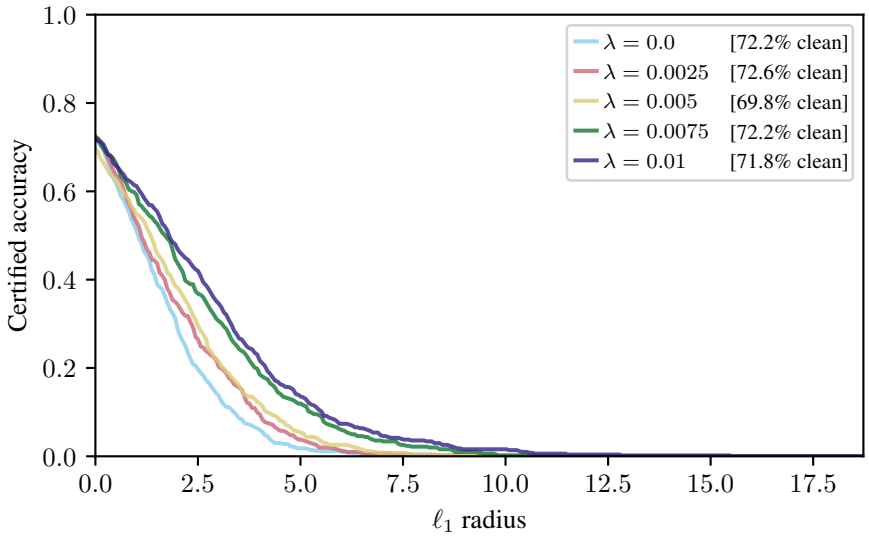


Figure 9: Impact of the Jacobian regularization parameter λ on CIFAR-10 cats-dogs classification.

As expected, the clean accuracies for both datasets are lower for $\varphi = \mathrm{Id}$, while the certified radii are generally larger due to the Lipschitz scaling factor in Theorem 1. Interestingly, while the standard φ produces comparable performance in both experiments, the identity feature map classifier is more effective in the dogs-cats experiment, achieving around 7% greater clean accuracy. This reflects the observation that convex separability is an asymmetric condition and suggests that feature maps can mitigate this concern.

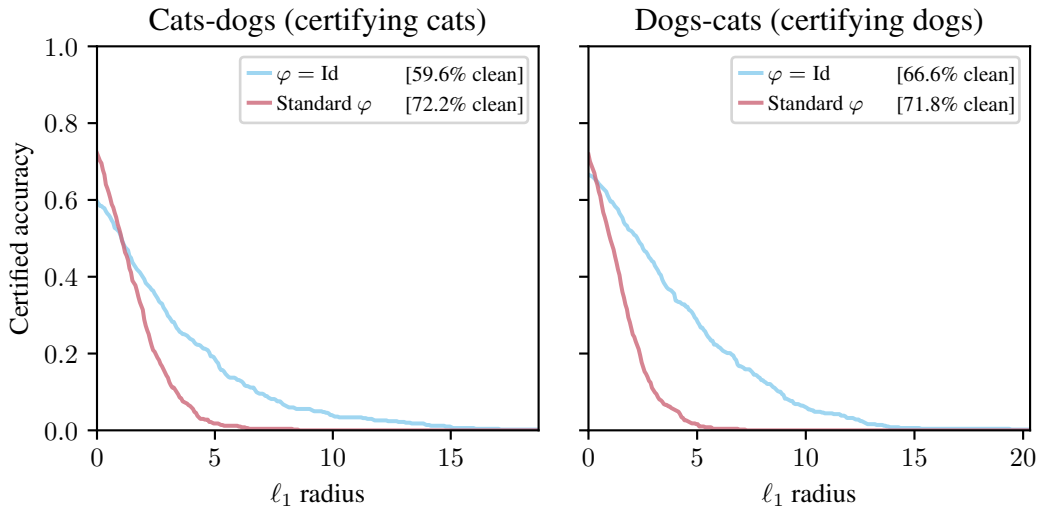


Figure 10: (a) Certification performance with cats as class 1 and dogs as class 2. (b) Certification performance with dogs as class 1 and cats as class 2.

G.3 Unaugmented accuracies

Table 2 summarizes the experimental counterpart to Section 3.2. Namely, Corollary 1 proves that there exists an input-convex classifier ($\varphi = \mathrm{Id}$) that achieves perfect training accuracy on the CIFAR-10 cats-dogs dataset with no dataset augmentations (random crops, flips, etc.). Our prac-

tical experiments are far from achieving this theoretical guarantee, with just 73.4% accuracy for cats-dogs and 77.2% for dogs-cats. Improving the practical performance of input-convex classifiers to match their theoretical capacity is an exciting area of future research.

Table 2: CIFAR-10 accuracies with no feature augmentation ($\varphi = \mathrm{Id}$) and no input augmentation.

Class 1-class 2 data	Training accuracy	Test accuracy (balanced)
Cats-dogs	73.4%	57.3%
Dogs-cats	77.2%	63.9%